

Corruption_ML_model

May 1, 2024

1 Corruption Prediction using the World Bank Indicators

```
[101]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import os
import xgboost as xgb
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from matplotlib import gridspec
from matplotlib import style
from prophet import Prophet
import statsmodels
from tqdm import tqdm
from sklearn.metrics import r2_score, mean_squared_error, make_scorer
from sklearn.model_selection import train_test_split, KFold
from xgboost import XGBRegressor
from statsmodels.nonparametric.smoothers_lowess import lowess
import pycountry
import matplotlib.cm as cm
from matplotlib.colors import Normalize
from matplotlib.colorbar import ColorbarBase
from statsmodels.tsa.arima.model import ARIMA
from xgboost import plot_tree
from keras.models import Sequential
from keras.layers import Dense
import tensorflow_decision_forests as tfdf
from tensorflow.keras.utils import plot_model
import pydotplus as pydot
import graphviz
import networkx as nx
import visualkeras
```

```
from xgboost import plot_importance
```

2 Model 1

```
[102]: df = pd.read_csv('WDI_data.csv')
df = df.drop(columns=['CPI_EST'])
# Make corruption more intuitive with higher values indicating more corruption
df['CC_EST'] = (-1)*df['CC_EST']
df['CC_EST'] = (df['CC_EST'] + 2.5) * 20
# Drop years 1960-2011 if column 'year' is less than 2012
df_dr = df[df['year'] >= 2012]
```

```
[103]: # Create a new dataframe with only iso2c, country, and CC_EST
df_corr = df_dr[['iso2c', 'country', 'year', 'CC_EST']]
```

```
[104]: df_corr
```

```
[104]:      iso2c   country   year    CC_EST
 52       AD     Andorra  2012  24.789383
 53       AD     Andorra  2013  24.929030
 54       AD     Andorra  2014  25.585828
 55       AD     Andorra  2015  26.963785
 56       AD     Andorra  2016  26.808882
...
13729     ZW     Zimbabwe 2018  74.920013
13730     ZW     Zimbabwe 2019  75.423806
13731     ZW     Zimbabwe 2020  75.759847
13732     ZW     Zimbabwe 2021  75.071001
13733     ZW     Zimbabwe 2022  75.102789
```

[2398 rows x 4 columns]

```
[105]: # Filter the data for the years 2012 and 2022
df_filtered = df_corr[df_corr['year'].isin([2012, 2022])]

# Calculate the linear differences over time for each given country
change_by_country = df_filtered.pivot_table(index='iso2c', columns='year',
                                             values='CC_EST', aggfunc='first')

# Calculate the change for each country
change_by_country['avg_change'] = (change_by_country[2022] - change_by_country[2012]) / 10

# Calculate the mean and median CC_EST values across all years for each country
mean_cc_est_by_country = df_filtered.groupby('iso2c')['CC_EST'].mean()
median_cc_est_by_country = df_filtered.groupby('iso2c')['CC_EST'].median()
```

```

# Merge with the original DataFrame to get the country names
df_trend = pd.merge(change_by_country['avg_change'], mean_cc_est_by_country, ↵
    ↵left_index=True, right_index=True, how='left')
df_trend = pd.merge(df_trend, median_cc_est_by_country, left_index=True, ↵
    ↵right_index=True, how='left')

# Reset index and rename columns
df_trend.reset_index(inplace=True)
df_trend.rename(columns={'CC_EST_x': 'mean_CC_EST', 'CC_EST_y': ↵
    ↵'median_CC_EST'}, inplace=True)

# Include country names
df_trend = pd.merge(df_trend, df_filtered[['iso2c', 'country']]. ↵
    ↵drop_duplicates(), on='iso2c', how='left')

print(df_trend)

```

	iso2c	avg_change	mean_CC_EST	median_CC_EST	country
0	AD	-0.019346	24.692656	24.692656	Andorra
1	AE	0.000799	26.889273	26.889273	United Arab Emirates
2	AF	-0.493193	76.141493	76.141493	Afghanistan
3	AG	1.899853	34.288647	34.288647	Antigua and Barbuda
4	AL	-0.741708	61.866050	61.866050	Albania
..
199	XK	-0.779124	59.171521	59.171521	Kosovo
200	YE	0.841619	79.383066	79.383066	Yemen, Rep.
201	ZA	0.271189	55.039358	55.039358	South Africa
202	ZM	0.475517	58.206411	58.206411	Zambia
203	ZW	-0.253328	76.369429	76.369429	Zimbabwe

[204 rows x 5 columns]

```

[106]: # Max and min values
# Find the index of the maximum and minimum mean_CC_EST values
most_corrupt_idx = df_trend['mean_CC_EST'].idxmax()
least_corrupt_idx = df_trend['mean_CC_EST'].idxmin()

most_corrupt_country = df_trend.loc[most_corrupt_idx, 'country']
least_corrupt_country = df_trend.loc[least_corrupt_idx, 'country']

print("The most corrupt country (on average between 2012 and 2022) is", ↵
    ↵most_corrupt_country)
print("The least corrupt country (on average between 2012 and 2022) is", ↵
    ↵least_corrupt_country)
print("The country that became more corrupt (rose the most) on average between ↵
    ↵2012 and 2022 is", df_trend.loc[df_trend['avg_change'].idxmax(), 'country'])

```

```
print("The country that became less corrupt (fell the most) on average between\u
↳2012 and 2022 is", df_trend.loc[df_trend['avg_change'].idxmin(), 'country'])
```

The most corrupt country (on average between 2012 and 2022) is Somalia
The least corrupt country (on average between 2012 and 2022) is Denmark
The country that became more corrupt (rose the most) on average between 2012 and 2022 is Antigua and Barbuda
The country that became less corrupt (fell the most) on average between 2012 and 2022 is Seychelles

```
[107]: style.use('default')
df_trend = df_trend.sort_values('avg_change', ascending=True)
fig, ax = plt.subplots(figsize=(20, 10))

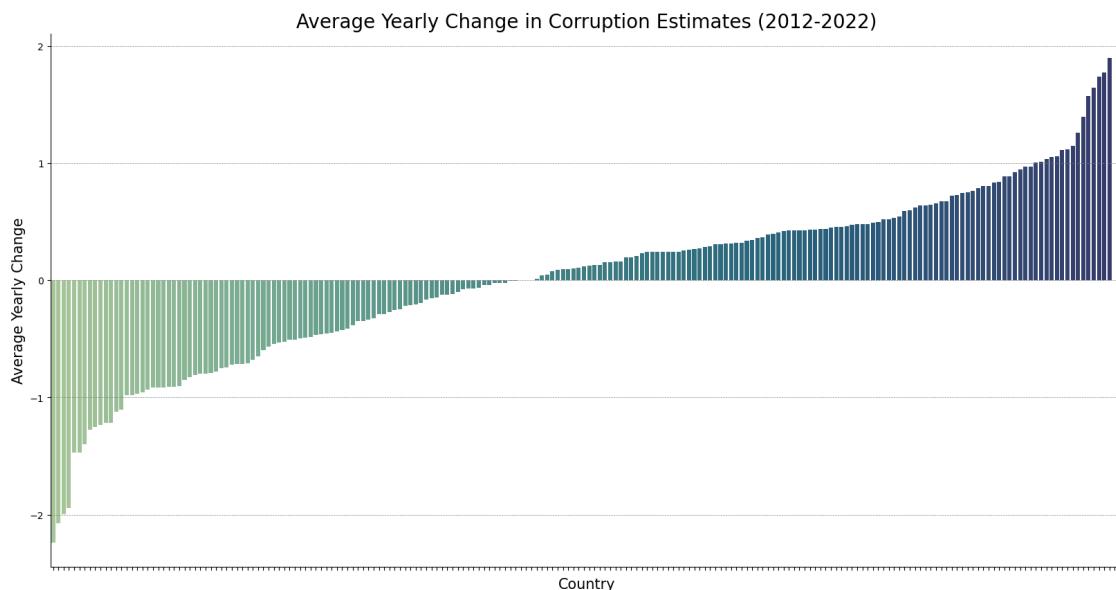
sns.barplot(x='country', y='avg_change', data=df_trend, ax=ax, palette =✉
↳'crest')

ax.set_title('Average Yearly Change in Corruption Estimates (2012-2022)',✉
↳fontsize=20)
ax.set_xlabel('Country', fontsize=15)
ax.set_ylabel('Average Yearly Change', fontsize=15)

ax.set_xticklabels([])
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

ax.grid(color='gray', linestyle='--', linewidth=0.5, axis='y')

plt.show()
```



```
[108]: # Mean and standard deviation of the average yearly change in corruption
       ↪estimates between 2012 and 2022
mean_change = df_trend['avg_change'].mean()
std_change = df_trend['avg_change'].std()

print("The mean average yearly change in corruption estimates between 2012 and
       ↪2022 is", mean_change)
print("The standard deviation of the average yearly change in corruption
       ↪estimates between 2012 and 2022 is", std_change)
```

The mean average yearly change in corruption estimates between 2012 and 2022 is 0.01970628684158408

The standard deviation of the average yearly change in corruption estimates between 2012 and 2022 is 0.7312920855366155

```
[109]: # List the top 10 countries that became more corrupt on average between 2012
       ↪and 2022
print("Top 10 countries that became more corrupt on average between 2012 and
       ↪2022:")
print(df_trend[['country', 'avg_change']].sort_values('avg_change', ↪
       ↪ascending=False).head(10))

# List the top 10 countries that became less corrupt on average between 2012
       ↪and 2022
print("Top 10 countries that became less corrupt on average between 2012 and
       ↪2022:")
print(df_trend[['country', 'avg_change']].sort_values('avg_change', ↪
       ↪ascending=True).head(10))
```

Top 10 countries that became more corrupt on average between 2012 and 2022:

	country	avg_change
3	Antigua and Barbuda	1.899853
101	Cayman Islands	1.777005
97	St. Kitts and Nevis	1.741485
44	Cyprus	1.643386
195	Virgin Islands (U.S.)	1.571842
126	Malta	1.393783
184	Turkiye	1.257980
37	Chile	1.150622
171	South Sudan	1.119872
174	Syrian Arab Republic	1.113265

Top 10 countries that became less corrupt on average between 2012 and 2022:

	country	avg_change
160	Seychelles	-2.236412
151	Palau	-2.070680
186	Tuvalu	-1.993962

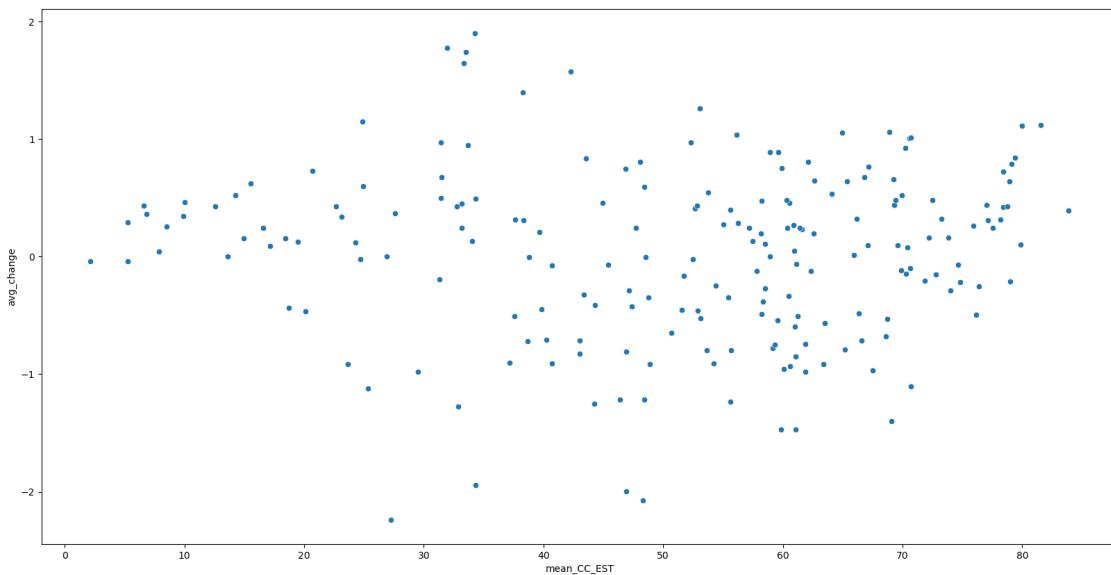
```

8      American Samoa   -1.941585
102     Kazakhstan    -1.471929
21      Benin        -1.467326
6       Angola       -1.400920
23     Brunei Darussalam -1.272374
139     Nauru        -1.252146
5       Armenia      -1.232073

```

```
[110]: # Plot average yearly change against mean CC_EST
fig, ax = plt.subplots(figsize=(20, 10))
sns.scatterplot(x='mean_CC_EST', y='avg_change', data=df_trend, ax=ax)
```

```
[110]: <Axes: xlabel='mean_CC_EST', ylabel='avg_change'>
```



```
[111]: # Plot the mean corruption estimates
plt.figure(figsize=(20, 10))

# Use seaborn's histplot function to plot the histogram
# Set the color, edgecolor and linewidth for the bars
sns.histplot(df_trend['mean_CC_EST'], bins=30, kde=True, color='#00688B',
             edgecolor='#00688B', linewidth=1)

# Calculate the mean and median of the mean corruption estimates
mean_est = df_trend['mean_CC_EST'].mean()
median_est = df_trend['mean_CC_EST'].median()

# Add a vertical line at the mean
plt.axvline(mean_est, color='#00688B', linestyle='--', linewidth=2)
```

```

# Add a vertical line at the median
plt.axvline(median_est, color='#00688B', linestyle='--', linewidth=2)

# Set the labels and title with larger fonts
plt.xlabel('Mean Corruption Estimate', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Mean Corruption Estimates', fontsize=16)

plt.xlim(0,100)

# Set the grid style
plt.grid(axis='y', linestyle='--', alpha=0.7)

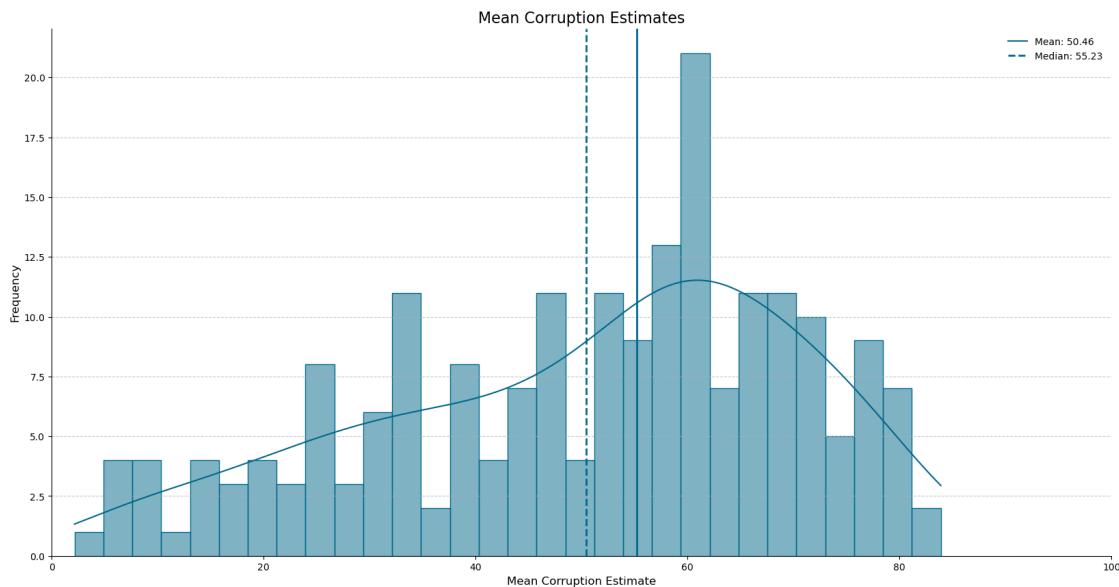
# Add a legend for mean and median
plt.legend([f'Mean: {mean_est:.2f}', f'Median: {median_est:.2f}'],  

           frameon=False)

# Remove top and right spines
sns.despine()

# Show the plot
plt.show()

```



```

[112]: df_correl = pd.read_csv('correlations.csv')
# Rank in order of correlation with CC_EST
df_correl = df_correl.sort_values('correlation', ascending=False)

```

```
df_correl
```

```
[112]:      variable  correlation
 942      DT_NFL_UNWT_CD    0.907982
 557      SE_LPV_PRIM     0.714894
 810      SE_LPV_PRIM_MA   0.714519
 727      SE_LPV_PRIM_FE   0.714091
 96       SE_LPV_PRIM_LD   0.703578
...
        ...
1091      LP_LPI_CUST_XQ   -0.812534
537       NY_ADJ_NNTY_PC_KD -0.813347
460       SI_SPR_PCAP     -0.841848
463       SI_SPR_PC40     -0.846424
77       IQ_CPA_TRAN_XQ   -0.859551
```

[1449 rows x 2 columns]

```
[113]: # Remove rows in df_dr that have missing values in CC_EST column
df_dr = df_dr.dropna(subset=['CC_EST'])

# Filter to only show correlations greater than or equal to +- 0.5
df_correl_5 = df_correl[abs(df_correl['correlation'])] >= 0.5]
# Filter to only show correlations greater than or equal to +- 0.6
df_correl_6 = df_correl[abs(df_correl['correlation'])] >= 0.6]
# Filter to only show correlations greater than or equal to +- 0.7
df_correl_7 = df_correl[abs(df_correl['correlation'])] >= 0.7]
# Filter to only show correlations greater than or equal to +- 0.8
df_correl_8 = df_correl[abs(df_correl['correlation'])] >= 0.8]
# Filter to only show correlations greater than or equal to +- 0.9
df_correl_9 = df_correl[abs(df_correl['correlation'])] >= 0.9]

print(f"The number of correlations with an absolute value greater than or equal
      to 0.5 is", len(df_correl_5))
print(f"The number of correlations with an absolute value greater than or equal
      to 0.6 is", len(df_correl_6))
print(f"The number of correlations with an absolute value greater than or equal
      to 0.7 is", len(df_correl_7))
print(f"The number of correlations with an absolute value greater than or equal
      to 0.8 is", len(df_correl_8))
print(f"The number of correlations with an absolute value greater than or equal
      to 0.9 is", len(df_correl_9))
```

The number of correlations with an absolute value greater than or equal to 0.5
is 253

The number of correlations with an absolute value greater than or equal to 0.6
is 128

The number of correlations with an absolute value greater than or equal to 0.7
is 54

```
The number of correlations with an absolute value greater than or equal to 0.8
is 7
The number of correlations with an absolute value greater than or equal to 0.9
is 1
```

I can then use these differing correlationary values to build a predictive model of corruption whilst optimising for computational efficiency simultaneously.

```
[114]: # Merge together corruption values and the largest correlators
# Create variables that are a list of the variables that are highly correlated
# with CC_EST
vars_5 = list(df_correl_5['variable'])
vars_6 = list(df_correl_6['variable'])
vars_7 = list(df_correl_7['variable'])
vars_8 = list(df_correl_8['variable'])
vars_9 = list(df_correl_9['variable'])

# Merge the corruption values with the variables that are highly correlated
# with CC_EST
df_5 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_5]
df_6 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_6]
df_7 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_7]
df_8 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_8]
df_9 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_9]
```

```
[115]: # Build random forest model to predict corruption from highly correlated
# variables
# Simplify notation:
x5 = df_5.iloc[:,4:].values
x6 = df_6.iloc[:,4:].values
x7 = df_7.iloc[:,4:].values
x8 = df_8.iloc[:,4:].values
x9 = df_9.iloc[:,4:].values

y = df_dr['CC_EST'].values
```

```
[116]: style.use('default')

def custom_cross_val_score(model, X, y, cv):
    kf = KFold(n_splits=cv, shuffle=True, random_state=0)
    scores = []

    for train_index, val_index in kf.split(X):
        x_train, x_val = X[train_index], X[val_index]
        y_train, y_val = y[train_index], y[val_index]

        model.fit(x_train, y_train, eval_set=[(x_val, y_val)], verbose=False)
        y_pred = model.predict(x_val)
```

```

        scores.append(r2_score(y_val, y_pred))

    return np.array(scores)

datasets = [(x5, y), (x6, y), (x7, y), (x8, y), (x9, y)]

r2_train_vals = []
r2_tests = []
rmse_train_vals = []
rmse_tests = []
mean_cv_scores = []

# Loop over the datasets
for i, (x, y) in enumerate(datasets, start=5):
    # Split the data into training+validation and testing sets
    x_train_val, x_test, y_train_val, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

    # Initialize the model with L1 regularization (alpha is the regularization parameter)
    model = XGBRegressor(objective='reg:squarederror', random_state=0, alpha=0.1, early_stopping_rounds=10)

    # After fitting the model
    # After fitting the model
    model.fit(x_train_val, y_train_val, eval_set=[(x_test, y_test)], verbose=False)

    # Get feature importances
    importances = model.feature_importances_

    # Sort features by importance
    indices = np.argsort(importances)[::-1]

    # Print the feature ranking
    print("Feature ranking:")

    for f in range(x.shape[1]):
        print(f"{f + 1}. feature {indices[f]} ({importances[indices[f]]})")

    # You can limit the number of features printed by replacing `x.shape[1]` with the desired number

    # Make predictions
    y_train_val_pred = model.predict(x_train_val)
    y_test_pred = model.predict(x_test)

```

```

# Calculate R^2 scores and round to 5 decimal places
r2_train_val = round(r2_score(y_train_val, y_train_val_pred), 5)
r2_test = round(r2_score(y_test, y_test_pred), 5)

# Calculate RMSE and round to 5 decimal places
rmse_train_val = round(np.sqrt(mean_squared_error(y_train_val, y_train_val_pred)), 5)
rmse_test = round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 5)

# Perform cross-validation and calculate mean score
cv_scores = custom_cross_val_score(model, x_train_val, y_train_val, cv=5)
mean_cv_score = round(np.mean(cv_scores), 5)

r2_train_vals.append(r2_train_val)
r2_tests.append(r2_test)
rmse_train_vals.append(rmse_train_val)
rmse_tests.append(rmse_test)
mean_cv_scores.append(mean_cv_score)

print(f"Model with correlation >= 0.{i}:")
print(f"Training+Validation R^2: {r2_train_val}, RMSE: {rmse_train_val}")
print(f"Testing R^2: {r2_test}, RMSE: {rmse_test}")
print(f"Mean cross-validation score: {mean_cv_score}\n")

# Create a scatter plot for the actual vs predicted values
abs_diffs = np.abs(y_test - y_test_pred)

# Create a DataFrame with the actual values and absolute differences
df = pd.DataFrame({'Actual': y_test, 'AbsDifference': abs_diffs})

# Define the bins for the actual values
bins = np.linspace(0, 100, 50)

# Calculate the mid-points of the bins
bin_midpoints = bins[:-1] + np.diff(bins) / 2

# Create a new column for the binned actual values
df['ActualBin'] = pd.cut(df['Actual'], bins, labels=bin_midpoints)

# Group by the binned actual values and calculate the variance of the
# absolute differences for each group
var_abs_diffs = df.groupby('ActualBin')['AbsDifference'].var()

# Create a scatter plot for the actual vs predicted values
fig = plt.figure(figsize=(10, 10))
gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1])
ax0 = plt.subplot(gs[0])

```

```

ax0.scatter(y_test, y_test_pred, alpha=0.7, color="#00688B", s=20)
ax0.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color="#00688B", lw=3)
ax0.set_xlim([0, 100])
ax0.set_ylabel('Predicted', fontsize=14)
ax0.set_title(f'Actual vs Predicted Values and Variance of Absolute Differences (Correlation >= 0.{i})', fontsize=16)
ax0.grid(True, color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

# Hide the right and top spines
ax0.spines['right'].set_visible(False)
ax0.spines['top'].set_visible(False)

# Only show ticks on the left and bottom spines
ax0.yaxis.set_ticks_position('left')
ax0.xaxis.set_ticks_position('bottom')

# Create a line plot for the binned actual values vs variance of the absolute differences
ax1 = plt.subplot(gs[1])

# Apply LOESS to smooth the variance curve
smoothed = lowess(var_abs_diffs, var_abs_diffs.index, frac=0.5)
index, data = zip(*smoothed)
ax1.plot(index, data, color="#00688B")

ax1.set_ylim([0, 15])
ax1.set_xlim([0, 100])
ax1.set_ylabel('Variance of Abs. Diff.', fontsize=14)
ax1.set_xticks([])

# Hide the right and top spines
ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)

# Only show ticks on the left and bottom spines
ax1.yaxis.set_ticks_position('left')
ax1.xaxis.set_ticks_position('bottom')

ax1.grid(axis='y', color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

plt.tight_layout()
plt.show()

```

Feature ranking:

1. feature 199 (0.3441309630870819)

2. feature 210 (0.23138484358787537)
3. feature 252 (0.11535115540027618)
4. feature 186 (0.01702813431620598)
5. feature 237 (0.013964777812361717)
6. feature 142 (0.013950027525424957)
7. feature 203 (0.011279524303972721)
8. feature 136 (0.010657941922545433)
9. feature 183 (0.01058618538081646)
10. feature 246 (0.00918436236679554)
11. feature 58 (0.008346294984221458)
12. feature 59 (0.007700021378695965)
13. feature 189 (0.007487301714718342)
14. feature 129 (0.00732423085719347)
15. feature 141 (0.006930687930434942)
16. feature 179 (0.005710081662982702)
17. feature 48 (0.005422129761427641)
18. feature 15 (0.005320514086633921)
19. feature 200 (0.005264010746032)
20. feature 150 (0.005106520839035511)
21. feature 98 (0.005003555212169886)
22. feature 122 (0.004956114571541548)
23. feature 174 (0.004833898041397333)
24. feature 20 (0.00476043438538909)
25. feature 31 (0.004652796313166618)
26. feature 219 (0.004637097008526325)
27. feature 169 (0.004278087522834539)
28. feature 206 (0.004080058075487614)
29. feature 30 (0.004039145540446043)
30. feature 132 (0.003971410449594259)
31. feature 171 (0.003869465086609125)
32. feature 120 (0.0035399298649281263)
33. feature 231 (0.003293973160907626)
34. feature 167 (0.003277714131399989)
35. feature 27 (0.0030536942649632692)
36. feature 95 (0.0026647155173122883)
37. feature 140 (0.002565942704677582)
38. feature 32 (0.002426720689982176)
39. feature 243 (0.0023525867145508528)
40. feature 162 (0.0021935408003628254)
41. feature 75 (0.0021625703666359186)
42. feature 201 (0.00215712352655828)
43. feature 198 (0.002140918280929327)
44. feature 115 (0.0020970392506569624)
45. feature 128 (0.002060073660686612)
46. feature 103 (0.0018604060169309378)
47. feature 82 (0.0018514996627345681)
48. feature 202 (0.0018514511175453663)
49. feature 185 (0.001807143329642713)

50. feature 47 (0.0017388425767421722)
51. feature 180 (0.0016392107354477048)
52. feature 207 (0.001629773760214448)
53. feature 172 (0.0016215266659855843)
54. feature 24 (0.0014824168756604195)
55. feature 156 (0.0014612438390031457)
56. feature 104 (0.0014590926002711058)
57. feature 56 (0.0013772668316960335)
58. feature 161 (0.0012830663472414017)
59. feature 41 (0.0012614066945388913)
60. feature 165 (0.0012070187367498875)
61. feature 125 (0.0011429657461121678)
62. feature 119 (0.0011292498093098402)
63. feature 87 (0.0010881648631766438)
64. feature 133 (0.0010379055747762322)
65. feature 34 (0.0009937261929735541)
66. feature 178 (0.0009781945263966918)
67. feature 227 (0.0009553525014780462)
68. feature 16 (0.0009503473993390799)
69. feature 127 (0.0009486611234024167)
70. feature 61 (0.0009341167169623077)
71. feature 137 (0.0009217206388711929)
72. feature 64 (0.0009134260471910238)
73. feature 130 (0.0008871010504662991)
74. feature 29 (0.0008848977158777416)
75. feature 153 (0.0008636629208922386)
76. feature 152 (0.0008475551148876548)
77. feature 191 (0.0008469861932098866)
78. feature 184 (0.0008359444909729064)
79. feature 22 (0.0008310644188895822)
80. feature 250 (0.0007915589376352727)
81. feature 188 (0.0007896274328231812)
82. feature 135 (0.0007785715279169381)
83. feature 60 (0.0007008419488556683)
84. feature 68 (0.000695106340572238)
85. feature 109 (0.0006670821458101273)
86. feature 55 (0.0006337789818644524)
87. feature 13 (0.0005979274865239859)
88. feature 195 (0.0005903344135731459)
89. feature 92 (0.0005829762667417526)
90. feature 67 (0.0005751053686253726)
91. feature 241 (0.0005681756883859634)
92. feature 12 (0.0005631537642329931)
93. feature 168 (0.0005451080505736172)
94. feature 248 (0.0005373795866034925)
95. feature 126 (0.000509825476910919)
96. feature 37 (0.0004976686323061585)
97. feature 131 (0.00048822275130078197)

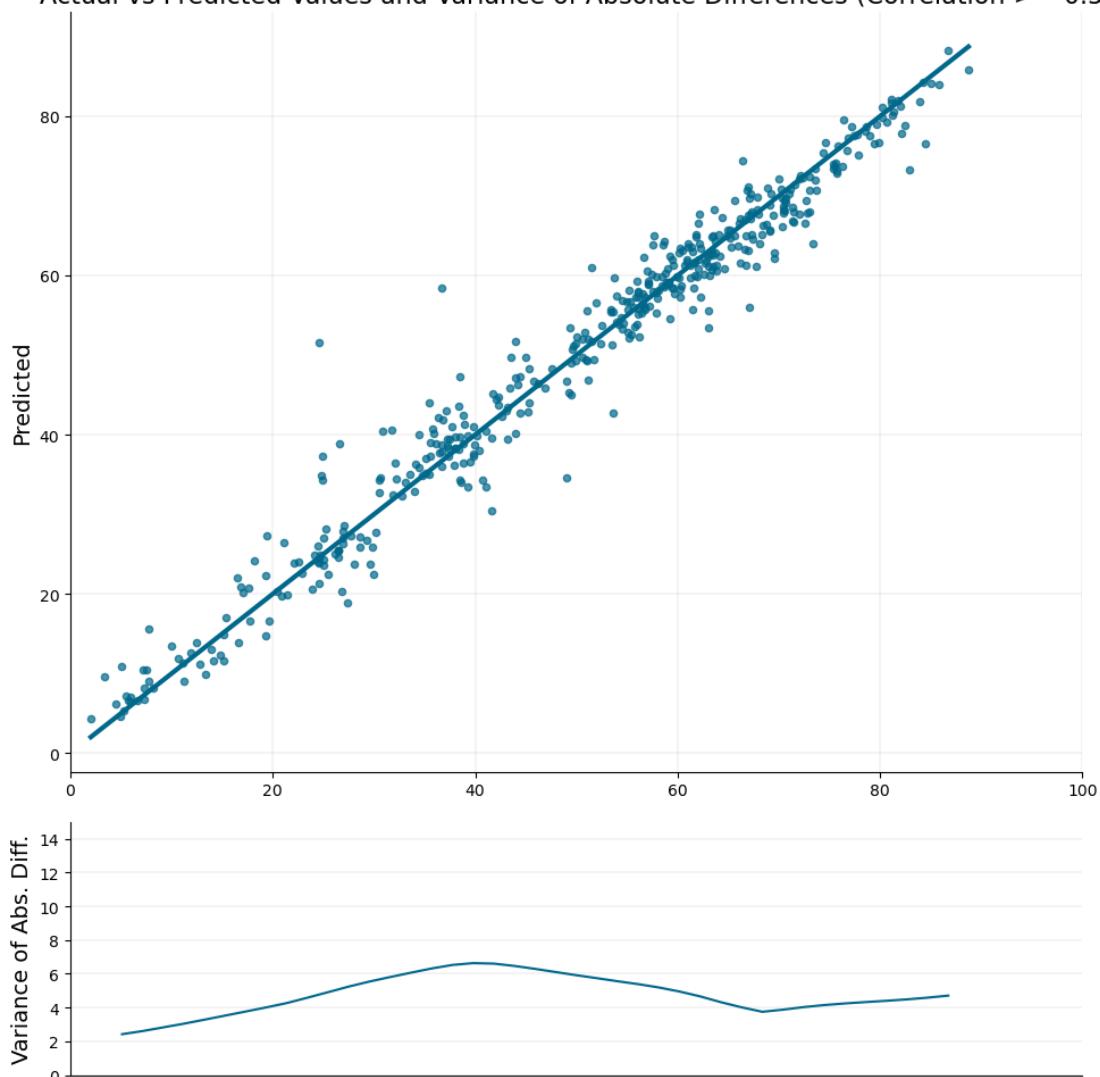
98. feature 159 (0.0004687319742515683)
99. feature 18 (0.00046522534103132784)
100. feature 102 (0.00046300076064653695)
101. feature 76 (0.00045744315139018)
102. feature 229 (0.0004399557947181165)
103. feature 46 (0.0004163759294897318)
104. feature 154 (0.0003972953127231449)
105. feature 86 (0.0003922555479221046)
106. feature 117 (0.0003793713403865695)
107. feature 239 (0.0003724454145412892)
108. feature 116 (0.00036602321779355407)
109. feature 57 (0.00034538115141913295)
110. feature 134 (0.00033913605147972703)
111. feature 144 (0.0003312210028525442)
112. feature 42 (0.0003219239297322929)
113. feature 148 (0.00031464180210605264)
114. feature 40 (0.0003135628649033606)
115. feature 177 (0.00031111514545045793)
116. feature 139 (0.0003067078359890729)
117. feature 160 (0.00030579938902519643)
118. feature 72 (0.00030361913377419114)
119. feature 163 (0.00029753748094663024)
120. feature 110 (0.0002943875442724675)
121. feature 151 (0.0002892186457756907)
122. feature 147 (0.00028917353483848274)
123. feature 247 (0.00027855465305037796)
124. feature 62 (0.0002692743146326393)
125. feature 196 (0.00026806804817169905)
126. feature 25 (0.000264921021880582)
127. feature 107 (0.0002629832015372813)
128. feature 74 (0.00025868689408525825)
129. feature 79 (0.000256554689258337)
130. feature 118 (0.0002552580845076591)
131. feature 26 (0.0002464327262714505)
132. feature 124 (0.00023895702906884253)
133. feature 66 (0.00023352718562819064)
134. feature 35 (0.00022536172764375806)
135. feature 158 (0.0002249151875730604)
136. feature 19 (0.00022358006390277296)
137. feature 187 (0.00021966702479403466)
138. feature 145 (0.0002085063752019778)
139. feature 176 (0.00020674978441093117)
140. feature 4 (0.00020464070257730782)
141. feature 39 (0.00019982705998700112)
142. feature 242 (0.00019346170302014798)
143. feature 23 (0.00019010910182259977)
144. feature 218 (0.00018109574739355594)
145. feature 111 (0.00017976659000851214)

146. feature 173 (0.0001723152818158269)
147. feature 90 (0.00017199719150085002)
148. feature 54 (0.00017125993326772004)
149. feature 113 (0.00016842447803355753)
150. feature 212 (0.00016485252126585692)
151. feature 170 (0.0001602077973075211)
152. feature 245 (0.00015568104572594166)
153. feature 211 (0.00015314552001655102)
154. feature 51 (0.00015273351164069027)
155. feature 175 (0.00013836969446856529)
156. feature 99 (0.00013297764235176146)
157. feature 53 (0.0001290155341848731)
158. feature 96 (0.00012052930105710402)
159. feature 52 (0.00011645397171378136)
160. feature 80 (0.00011612851085374132)
161. feature 45 (0.00011467323929537088)
162. feature 155 (0.00010689994087442756)
163. feature 85 (0.0001063461895682849)
164. feature 106 (0.00010537758498685434)
165. feature 204 (0.00010422895138617605)
166. feature 249 (0.00010394361743237823)
167. feature 50 (9.601299825590104e-05)
168. feature 84 (9.424517338629812e-05)
169. feature 232 (8.905595313990489e-05)
170. feature 17 (8.444210834568366e-05)
171. feature 9 (8.393337338929996e-05)
172. feature 5 (8.292553684441373e-05)
173. feature 11 (8.252855332102627e-05)
174. feature 83 (7.750095392111689e-05)
175. feature 88 (7.691430073464289e-05)
176. feature 44 (7.629996252944693e-05)
177. feature 157 (7.369340164586902e-05)
178. feature 146 (7.158853259170428e-05)
179. feature 112 (6.968402885831892e-05)
180. feature 100 (6.835614476585761e-05)
181. feature 223 (6.669055437669158e-05)
182. feature 234 (6.187924009282142e-05)
183. feature 97 (5.980397327220999e-05)
184. feature 36 (5.94133889535442e-05)
185. feature 77 (5.907815648242831e-05)
186. feature 166 (5.286864688969217e-05)
187. feature 190 (5.130211866344325e-05)
188. feature 33 (5.097526081954129e-05)
189. feature 143 (4.979063669452444e-05)
190. feature 192 (4.972963506588712e-05)
191. feature 21 (4.969849032931961e-05)
192. feature 71 (4.8702393542043865e-05)
193. feature 220 (4.837946835323237e-05)

194. feature 49 (4.721228469861671e-05)
195. feature 28 (4.715786053566262e-05)
196. feature 205 (4.655495285987854e-05)
197. feature 91 (4.551088932203129e-05)
198. feature 73 (4.5096709072822705e-05)
199. feature 236 (4.309223004383966e-05)
200. feature 228 (4.237949542584829e-05)
201. feature 78 (4.174164496362209e-05)
202. feature 238 (4.0440329030388966e-05)
203. feature 121 (3.9336737245321274e-05)
204. feature 209 (3.869632200803608e-05)
205. feature 114 (3.838496922980994e-05)
206. feature 235 (3.694710903801024e-05)
207. feature 182 (3.487982394290157e-05)
208. feature 93 (3.3877717214636505e-05)
209. feature 240 (3.200723949703388e-05)
210. feature 221 (3.123346687061712e-05)
211. feature 230 (2.9431308576022275e-05)
212. feature 43 (2.844043410732411e-05)
213. feature 108 (2.767677142401226e-05)
214. feature 149 (2.6208390409010462e-05)
215. feature 251 (2.567703995737247e-05)
216. feature 2 (2.2847902073408477e-05)
217. feature 81 (2.220953319920227e-05)
218. feature 89 (2.1273108359309845e-05)
219. feature 63 (2.09961726795882e-05)
220. feature 94 (2.048303394985851e-05)
221. feature 197 (1.9177039575879462e-05)
222. feature 217 (1.8516513591748662e-05)
223. feature 244 (1.8001173884840682e-05)
224. feature 65 (1.7724905774230137e-05)
225. feature 208 (1.729766336211469e-05)
226. feature 6 (1.7147061953437515e-05)
227. feature 164 (1.6098494597827084e-05)
228. feature 14 (1.5415647794725373e-05)
229. feature 1 (1.3290084098116495e-05)
230. feature 38 (1.3157126886653714e-05)
231. feature 101 (1.3132137610227801e-05)
232. feature 214 (1.3082737496006303e-05)
233. feature 70 (1.288781004404882e-05)
234. feature 224 (1.0706388820835855e-05)
235. feature 10 (1.0699781341827475e-05)
236. feature 181 (1.0388048394815996e-05)
237. feature 215 (9.160081390291452e-06)
238. feature 213 (6.807796580687864e-06)
239. feature 0 (5.111327936901944e-06)
240. feature 225 (4.512893156061182e-06)
241. feature 123 (1.1736859732991434e-06)

```
242. feature 193 (0.0)
243. feature 194 (0.0)
244. feature 105 (0.0)
245. feature 138 (0.0)
246. feature 8 (0.0)
247. feature 7 (0.0)
248. feature 233 (0.0)
249. feature 216 (0.0)
250. feature 3 (0.0)
251. feature 69 (0.0)
252. feature 222 (0.0)
253. feature 226 (0.0)
Model with correlation >= 0.5:
Training+Validation R^2: 0.99968, RMSE: 0.35587
Testing R^2: 0.96505, RMSE: 3.77319
Mean cross-validation score: 0.95452
```

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.5)



Feature ranking:

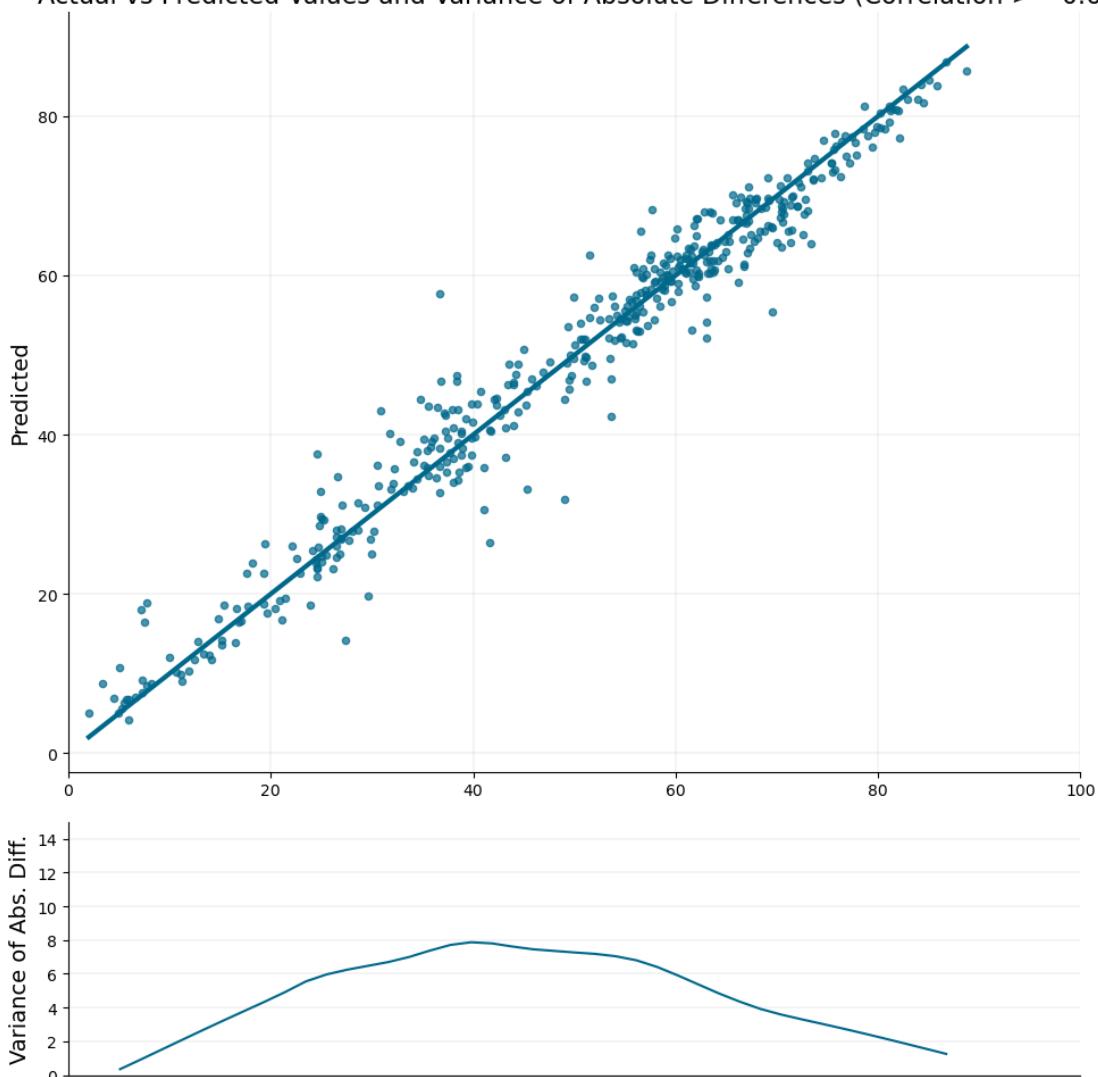
1. feature 74 (0.31771016120910645)
2. feature 85 (0.18727819621562958)
3. feature 127 (0.1559630185365677)
4. feature 32 (0.026648659259080887)
5. feature 112 (0.025778453797101974)
6. feature 20 (0.021845590323209763)
7. feature 94 (0.01952669583261013)
8. feature 64 (0.017185378819704056)
9. feature 70 (0.011316444724798203)
10. feature 78 (0.01081069465726614)
11. feature 57 (0.010012955404818058)
12. feature 31 (0.008948569186031818)

13. feature 54 (0.008236341178417206)
14. feature 49 (0.007733013480901718)
15. feature 50 (0.007290168199688196)
16. feature 122 (0.006851824000477791)
17. feature 69 (0.005982161499559879)
18. feature 81 (0.005447397939860821)
19. feature 29 (0.005414977204054594)
20. feature 73 (0.005399579647928476)
21. feature 63 (0.005092828534543514)
22. feature 44 (0.004977512173354626)
23. feature 106 (0.004775524139404297)
24. feature 24 (0.00475804228335619)
25. feature 66 (0.004689306020736694)
26. feature 25 (0.0044582500122487545)
27. feature 121 (0.004212925676256418)
28. feature 71 (0.004133296199142933)
29. feature 60 (0.003959030844271183)
30. feature 118 (0.003936504013836384)
31. feature 75 (0.0038321763277053833)
32. feature 62 (0.0036408863961696625)
33. feature 41 (0.0034151917789131403)
34. feature 26 (0.003081787843257189)
35. feature 114 (0.003070663893595338)
36. feature 55 (0.0026908242143690586)
37. feature 36 (0.002665147418156266)
38. feature 35 (0.0024645659141242504)
39. feature 47 (0.002309795469045639)
40. feature 15 (0.0022584819234907627)
41. feature 123 (0.0022542085498571396)
42. feature 52 (0.00218269694596529)
43. feature 11 (0.0020200933795422316)
44. feature 115 (0.001959689659997821)
45. feature 87 (0.001914836815558374)
46. feature 40 (0.001897975686006248)
47. feature 48 (0.0018208209658041596)
48. feature 102 (0.0018006728496402502)
49. feature 53 (0.001798510318621993)
50. feature 68 (0.0017848375719040632)
51. feature 43 (0.0017411555163562298)
52. feature 12 (0.0017253487603738904)
53. feature 46 (0.0017227537464350462)
54. feature 77 (0.0017118050018325448)
55. feature 58 (0.0017022638348862529)
56. feature 22 (0.0016550786094740033)
57. feature 96 (0.001615671208128333)
58. feature 105 (0.0015449728816747665)
59. feature 33 (0.0015023780288174748)
60. feature 42 (0.0014944967115297914)

61. feature 27 (0.001486034132540226)
62. feature 86 (0.0013817090075463057)
63. feature 82 (0.0013467512326315045)
64. feature 76 (0.0013449807884171605)
65. feature 59 (0.001202919171191752)
66. feature 116 (0.001092569320462644)
67. feature 51 (0.0009320040117017925)
68. feature 13 (0.0008829956059344113)
69. feature 30 (0.0008685712236911058)
70. feature 61 (0.00079933280358091)
71. feature 56 (0.0006371669005602598)
72. feature 83 (0.0005882136174477637)
73. feature 113 (0.0005777640617452562)
74. feature 17 (0.0005777503829449415)
75. feature 124 (0.0005328395636752248)
76. feature 39 (0.0004936057957820594)
77. feature 79 (0.0004783498588949442)
78. feature 9 (0.00046340166591107845)
79. feature 117 (0.00045829705777578056)
80. feature 103 (0.00041815961594693363)
81. feature 67 (0.0003846377949230373)
82. feature 18 (0.0003819888224825263)
83. feature 19 (0.0003712064935825765)
84. feature 34 (0.00036914058728143573)
85. feature 80 (0.00035868349368683994)
86. feature 89 (0.0003542428894434124)
87. feature 111 (0.00032860960345715284)
88. feature 72 (0.0003058721194975078)
89. feature 23 (0.00030180011526681483)
90. feature 97 (0.00029032453312538564)
91. feature 16 (0.0002775304892566055)
92. feature 28 (0.00026267702924087644)
93. feature 14 (0.00026254242402501404)
94. feature 109 (0.0002308694674866274)
95. feature 107 (0.0002206421340815723)
96. feature 92 (0.00021653017029166222)
97. feature 37 (0.00021610195108223706)
98. feature 84 (0.00020134223450440913)
99. feature 119 (0.00019268215692136437)
100. feature 95 (0.00016082289221230894)
101. feature 65 (0.0001548499712953344)
102. feature 38 (0.00015360195538960397)
103. feature 101 (0.00015075673582032323)
104. feature 126 (0.00013583428517449647)
105. feature 6 (0.00012228774721734226)
106. feature 104 (0.0001076135304174386)
107. feature 98 (0.00010185241262661293)
108. feature 4 (0.00010168009612243623)

```
109. feature 5 (9.867953485809267e-05)
110. feature 99 (9.639330528443679e-05)
111. feature 93 (9.114253771258518e-05)
112. feature 45 (9.055052942130715e-05)
113. feature 125 (8.838870417093858e-05)
114. feature 90 (8.196311682695523e-05)
115. feature 88 (7.867952808737755e-05)
116. feature 7 (7.60700786486268e-05)
117. feature 110 (7.243661093525589e-05)
118. feature 10 (6.393803050741553e-05)
119. feature 120 (5.9841226175194606e-05)
120. feature 3 (5.763019362348132e-05)
121. feature 108 (5.7393288443563506e-05)
122. feature 1 (1.9879564206348732e-05)
123. feature 2 (1.949218312802259e-05)
124. feature 21 (7.09394043951761e-06)
125. feature 0 (6.996914635237772e-06)
126. feature 8 (0.0)
127. feature 100 (0.0)
128. feature 91 (0.0)
Model with correlation >= 0.6:
Training+Validation R^2: 0.99961, RMSE: 0.39421
Testing R^2: 0.96229, RMSE: 3.9194
Mean cross-validation score: 0.94662
```

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.6)



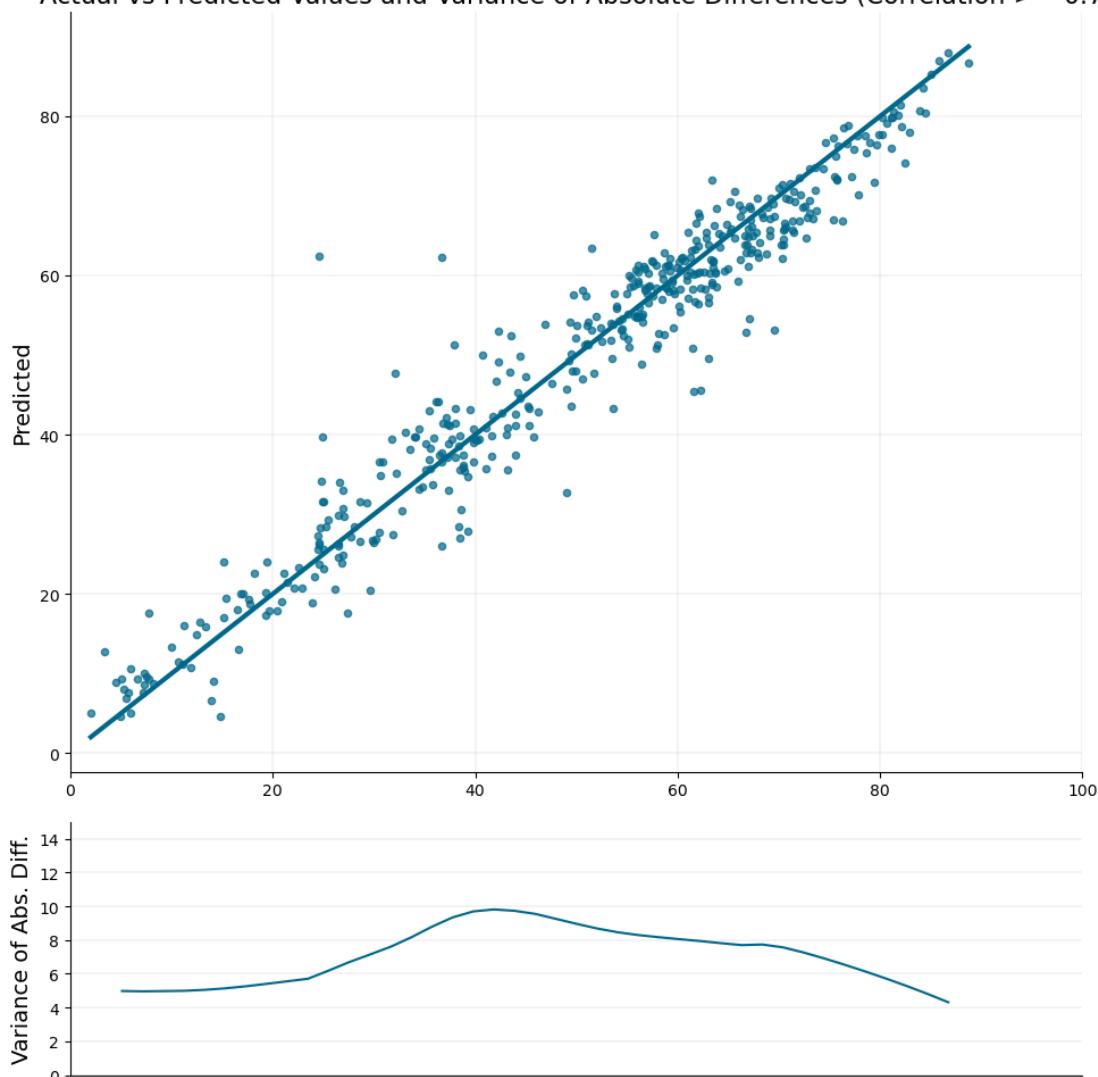
Feature ranking:

1. feature 11 (0.3659200966358185)
2. feature 53 (0.28489449620246887)
3. feature 38 (0.11513756960630417)
4. feature 20 (0.02071014791727066)
5. feature 28 (0.014015117660164833)
6. feature 31 (0.013600725680589676)
7. feature 48 (0.013204297982156277)
8. feature 13 (0.010585211217403412)
9. feature 47 (0.009999927133321762)
10. feature 32 (0.009939268231391907)
11. feature 49 (0.009131794795393944)
12. feature 12 (0.008918261155486107)

```
13. feature 41 (0.008812271058559418)
14. feature 40 (0.007948868907988071)
15. feature 34 (0.007729346863925457)
16. feature 8 (0.007671836297959089)
17. feature 50 (0.007223219610750675)
18. feature 45 (0.006975828669965267)
19. feature 44 (0.006220669951289892)
20. feature 42 (0.006059700157493353)
21. feature 7 (0.00591591140255332)
22. feature 19 (0.005635147448629141)
23. feature 33 (0.004717874340713024)
24. feature 37 (0.0038628443144261837)
25. feature 30 (0.00358963874168694)
26. feature 43 (0.003124205395579338)
27. feature 22 (0.0030850954353809357)
28. feature 52 (0.0028595642652362585)
29. feature 6 (0.0028463548514992)
30. feature 16 (0.0021189525723457336)
31. feature 5 (0.0021099888253957033)
32. feature 46 (0.0019200618844479322)
33. feature 17 (0.001764087239280343)
34. feature 25 (0.0017198780551552773)
35. feature 51 (0.0016804642509669065)
36. feature 39 (0.0016485248925164342)
37. feature 2 (0.0015894847456365824)
38. feature 27 (0.001539153279736638)
39. feature 10 (0.0015252049779519439)
40. feature 35 (0.001436354941688478)
41. feature 18 (0.0013349942164495587)
42. feature 23 (0.0011333520524203777)
43. feature 14 (0.0010921345092356205)
44. feature 24 (0.0010713303927332163)
45. feature 15 (0.001060275943018496)
46. feature 9 (0.0009235357865691185)
47. feature 3 (0.0009167752577923238)
48. feature 36 (0.0008743382059037685)
49. feature 21 (0.0006285422714427114)
50. feature 29 (0.0005353413871489465)
51. feature 1 (0.0003807509783655405)
52. feature 4 (0.0003074120613746345)
53. feature 26 (0.0002749039267655462)
54. feature 0 (7.89538025856018e-05)

Model with correlation >= 0.7:
Training+Validation R^2: 0.99447, RMSE: 1.48281
Testing R^2: 0.93749, RMSE: 5.0465
Mean cross-validation score: 0.90218
```

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.7)



Feature ranking:

1. feature 6 (0.4924904704093933)
2. feature 1 (0.3603402078151703)
3. feature 2 (0.05750051885843277)
4. feature 3 (0.041945330798625946)
5. feature 5 (0.0327630490064621)
6. feature 4 (0.01222801674157381)
7. feature 0 (0.002732437802478671)

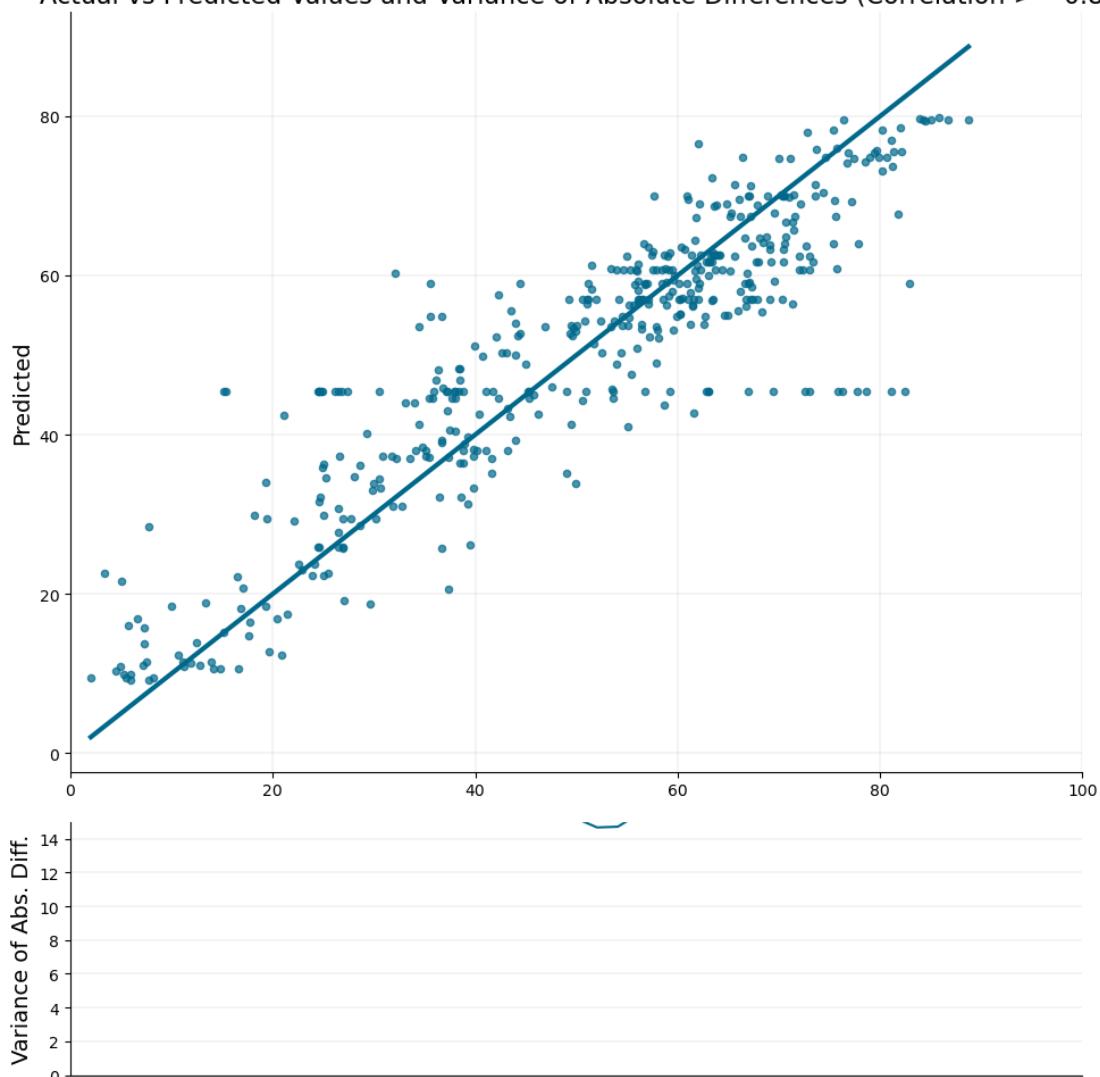
Model with correlation ≥ 0.8 :

Training+Validation R^2 : 0.81193, RMSE: 8.64535

Testing R^2 : 0.79896, RMSE: 9.04992

Mean cross-validation score: 0.75062

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.8)



Feature ranking:

1. feature 0 (1.0)

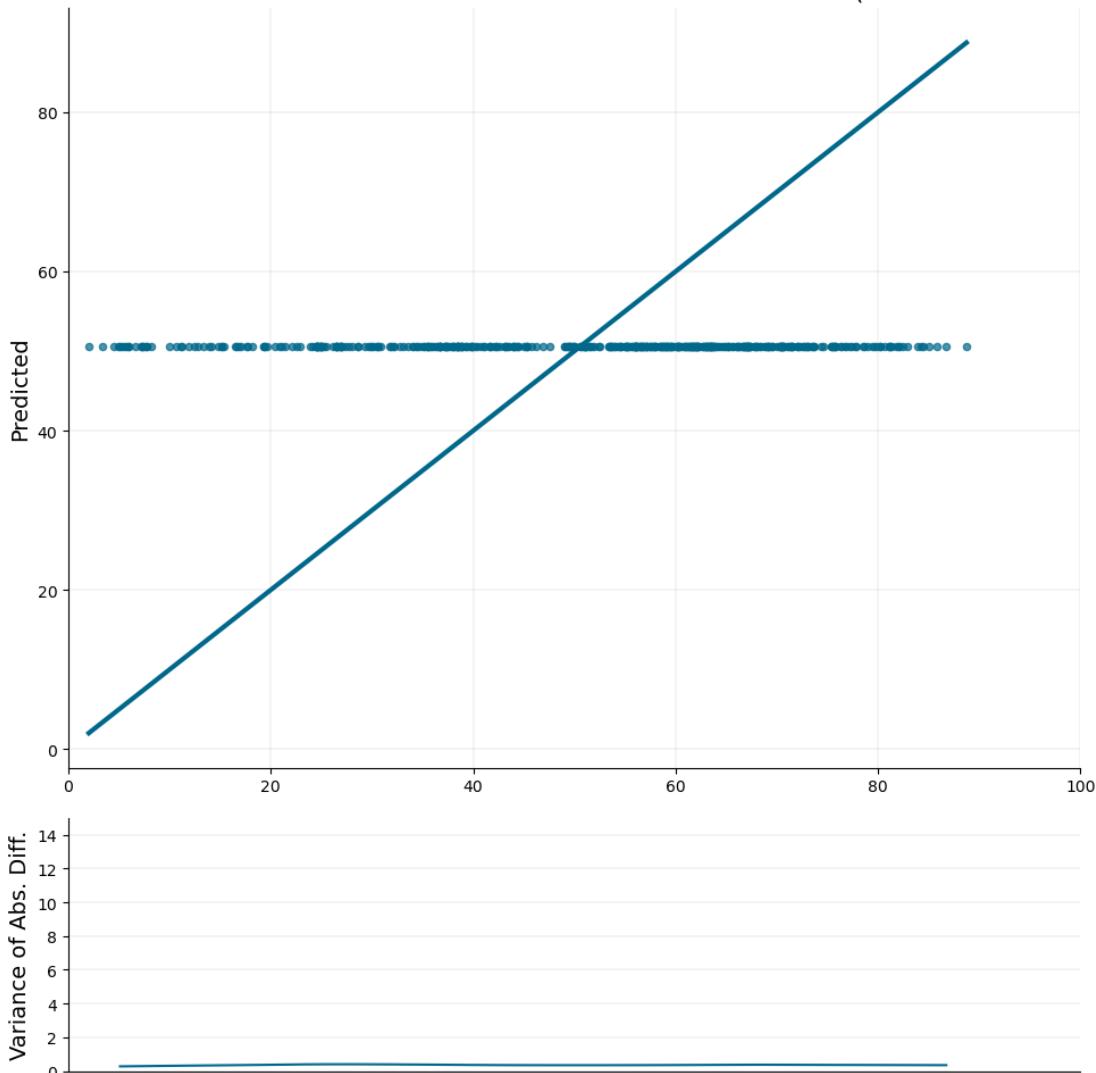
Model with correlation ≥ 0.9 :

Training+Validation $R^2: 0.00111$, RMSE: 19.92421

Testing $R^2: -0.00041$, RMSE: 20.18809

Mean cross-validation score: -0.00175

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.9)



```
[117]: # Assuming you have lists of feature names for each dataset
feature_names_list = [vars_5, vars_6, vars_7, vars_8, vars_9]

# Loop over the datasets
for i, ((x, y), feature_names) in enumerate(zip(datasets, feature_names_list), start=5):
    # The rest of your code...
    # Split the data into training+validation and testing sets
    x_train_val, x_test, y_train_val, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```

```

# Initialize the model with L1 regularization (alpha is the regularization parameter)
model = XGBRegressor(objective='reg:squarederror', random_state=0, alpha=0.1, early_stopping_rounds=10)

# Fit the model
model.fit(x_train_val, y_train_val, eval_set=[(x_test, y_test)], verbose=False)

# Get feature importances
importances = model.feature_importances_

# Sort features by importance
indices = np.argsort(importances)[::-1]

# Print the feature ranking
print("Feature ranking:")

for f in range(x.shape[1]):
    print(f" {f + 1}. feature {feature_names[indices[f]]} {importances[indices[f]]}")

# You can limit the number of features printed by replacing `x.shape[1]` with the desired number

# Make predictions
y_train_val_pred = model.predict(x_train_val)
y_test_pred = model.predict(x_test)

# Calculate R^2 scores and round to 5 decimal places
r2_train_val = round(r2_score(y_train_val, y_train_val_pred), 5)
r2_test = round(r2_score(y_test, y_test_pred), 5)

# Calculate RMSE and round to 5 decimal places
rmse_train_val = round(np.sqrt(mean_squared_error(y_train_val, y_train_val_pred)), 5)
rmse_test = round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 5)

# Perform cross-validation and calculate mean score
cv_scores = custom_cross_val_score(model, x_train_val, y_train_val, cv=5)
mean_cv_score = round(np.mean(cv_scores), 5)

r2_train_vals.append(r2_train_val)
r2_tests.append(r2_test)
rmse_train_vals.append(rmse_train_val)
rmse_tests.append(rmse_test)

```

```

mean_cv_scores.append(mean_cv_score)

print(f"Model with correlation >= 0.{i}:")
print(f"Training+Validation R^2: {r2_train_val}, RMSE: {rmse_train_val}")
print(f"Testing R^2: {r2_test}, RMSE: {rmse_test}")
print(f"Mean cross-validation score: {mean_cv_score}\n")

# Create a scatter plot for the actual vs predicted values
abs_diffs = np.abs(y_test - y_test_pred)

# Create a DataFrame with the actual values and absolute differences
df = pd.DataFrame({'Actual': y_test, 'AbsDifference': abs_diffs})

# Define the bins for the actual values
bins = np.linspace(0, 100, 50)

# Calculate the mid-points of the bins
bin_midpoints = bins[:-1] + np.diff(bins) / 2

# Create a new column for the binned actual values
df['ActualBin'] = pd.cut(df['Actual'], bins, labels=bin_midpoints)

# Group by the binned actual values and calculate the variance of the
absolute differences for each group
var_abs_diffs = df.groupby('ActualBin')['AbsDifference'].var()

# Create a scatter plot for the actual vs predicted values
fig = plt.figure(figsize=(10, 10))
gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1])
ax0 = plt.subplot(gs[0])
ax0.scatter(y_test, y_test_pred, alpha=0.7, color='#00688B', s=20)
ax0.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='grey', lw=3)
ax0.set_xlim([0, 100])
ax0.set_ylabel('Predicted', fontsize=14)
ax0.set_title(f'Actual vs Predicted Values and Variance of Absolute'
Differences (Correlation >= 0.{i})', fontsize=16)
ax0.grid(True, color='grey', linestyle='-', linewidth=0.25, alpha=0.5)

# Hide the right and top spines
ax0.spines['right'].set_visible(False)
ax0.spines['top'].set_visible(False)

# Only show ticks on the left and bottom spines
ax0.yaxis.set_ticks_position('left')
ax0.xaxis.set_ticks_position('bottom')

```

```

# Create a line plot for the binned actual values vs variance of the absolute differences
ax1 = plt.subplot(gs[1])

# Apply LOESS to smooth the variance curve
smoothed = lowess(var_abs_diffs, var_abs_diffs.index, frac=0.5)
index, data = zip(*smoothed)
ax1.plot(index, data, color='#00688B')

ax1.set_ylim([0, 15])
ax1.set_xlim([0, 100])
ax1.set_ylabel('Variance of Abs. Diff.', fontsize=14)
ax1.set_xticks([])

# Hide the right and top spines
ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)

# Only show ticks on the left and bottom spines
ax1.yaxis.set_ticks_position('left')
ax1.xaxis.set_ticks_position('bottom')

ax1.grid(axis='y', color='grey', linestyle='-', linewidth=0.25, alpha=0.5)

plt.tight_layout()
plt.show()

```

Feature ranking:

1. feature NY_GDP_PCAP_CD (0.3441309630870819)
2. feature NY_GDP_PCAP_KD (0.23138484358787537)
3. feature IQ_CPA_TRAN_XQ (0.11535115540027618)
4. feature SL_EMP_WORK_MA_ZS (0.01702813431620598)
5. feature IQ_CPA_PROP_XQ (0.013964777812361717)
6. feature SP_POP_5054_FE_5Y (0.013950027525424957)
7. feature SL_GDP_PCAP_EM_KD (0.011279524303972721)
8. feature FD_AST_PRVT_GD_ZS (0.010657941922545433)
9. feature SL_SRV_EMPL_ZS (0.01058618538081646)
10. feature NY_GNP_PCAP_KD (0.00918436236679554)
11. feature SH_XPD_OOPC_CH_ZS (0.008346294984221458)
12. feature SH_STA_STNT_ME_ZS (0.007700021378695965)
13. feature IT_NET_BBND_P2 (0.007487301714718342)
14. feature SH_STA_BASS_RU_ZS (0.00732423085719347)
15. feature FS_AST_PRVT_GD_ZS (0.006930687930434942)
16. feature SH_XPD_GHED_GD_ZS (0.005710081662982702)
17. feature SL_AGR_EMPL_FE_ZS (0.005422129761427641)
18. feature SL_EMP_VULN_FE_ZS (0.005320514086633921)
19. feature SH_XPD_OOPC_PC_CD (0.005264010746032)
20. feature IQ_CPA_PADM_XQ (0.005106520839035511)

21. feature SE_TER_ENRR_FE (0.005003555212169886)
22. feature SH_STA_BASS_ZS (0.004956114571541548)
23. feature SL_SRV_EMPL_FE_ZS (0.004833898041397333)
24. feature SL_EMP_SELF_MA_ZS (0.00476043438538909)
25. feature SL_UEM_NEET_FE_ZS (0.004652796313166618)
26. feature NY_GNP_PCAP_CD (0.004637097008526325)
27. feature SP_POP_5054_MA_5Y (0.004278087522834539)
28. feature IC_BUS_DFRN_XQ (0.004080058075487614)
29. feature SL_AGR_EMPL_MA_ZS (0.004039145540446043)
30. feature SH_H20_BASW_ZS (0.003971410449594259)
31. feature SP_DYN_T065_MA_ZS (0.003869465086609125)
32. feature SP_POP_6064_FE_5Y (0.0035399298649281263)
33. feature IQ_CPA_PUBS_XQ (0.003293973160907626)
34. feature SP_POP_5559_MA_5Y (0.003277714131399989)
35. feature SP_POP_1014_MA_5Y (0.0030536942649632692)
36. feature SH_H20_BASW_UR_ZS (0.0026647155173122883)
37. feature SH_XPD_PVTD_PC_CD (0.002565942704677582)
38. feature SL_AGR_EMPL_ZS (0.002426720689982176)
39. feature NY_GNP_PCAP_PP_KD (0.0023525867145508528)
40. feature SH_STA_SMSS_UR_ZS (0.0021935408003628254)
41. feature SN_ITK_DEFC_ZS (0.0021625703666359186)
42. feature GB_XPD_RSDV_GD_ZS (0.00215712352655828)
43. feature SH_H20_SMDW_ZS (0.002140918280929327)
44. feature SH_H20_SMDW_RU_ZS (0.0020970392506569624)
45. feature FB_ATM_TOTL_P5 (0.002060073660686612)
46. feature IC_BUS_NDNS_ZS (0.0018604060169309378)
47. feature SP_RUR_TOTL_ZS (0.0018514996627345681)
48. feature SH_STA_SMSS_ZS (0.0018514511175453663)
49. feature SE_SEC_ENRR_FE (0.001807143329642713)
50. feature SH_SGR_CRSK_ZS (0.0017388425767421722)
51. feature SE_SEC_ENRR_MA (0.0016392107354477048)
52. feature NY_GDP_PCAP_PP_KD (0.001629773760214448)
53. feature SP_POP_65UP_MA_ZS (0.0016215266659855843)
54. feature SP_POP_0014_TO_ZS (0.0014824168756604195)
55. feature NV_IND_EMPL_KD (0.0014612438390031457)
56. feature IQ_CPA_IRAI_XQ (0.0014590926002711058)
57. feature NV_AGR_TOTL_ZS (0.0013772668316960335)
58. feature SP_POP_4549_MA_5Y (0.0012830663472414017)
59. feature SP_POP_0004_FE_5Y (0.0012614066945388913)
60. feature NV_SRV_TOTL_ZS (0.0012070187367498875)
61. feature SP_POP_5559_FE_5Y (0.0011429657461121678)
62. feature NV_MNF_TECH_ZS_UN (0.0011292498093098402)
63. feature SP_POP_1564_MA_ZS (0.0010881648631766438)
64. feature SP_DYN_T065_FE_ZS (0.0010379055747762322)
65. feature SP_POP_0014_FE_ZS (0.0009937261929735541)
66. feature FS_AST_DOMS_GD_ZS (0.0009781945263966918)
67. feature PA_NUS_PPPC_RF (0.0009553525014780462)
68. feature SL_EMP_VULN_ZS (0.0009503473993390799)

69. feature IQ_SPI_OVRL (0.0009486611234024167)
70. feature SL_UEM_NEET_ZS (0.0009341167169623077)
71. feature SH_H20_BASW_RU_ZS (0.0009217206388711929)
72. feature SH_DYN_NCOM_FE_ZS (0.0009134260471910238)
73. feature SP_POP_7579_FE_5Y (0.0008871010504662991)
74. feature SP_POP_0004_MA_5Y (0.0008848977158777416)
75. feature SE_SEC_NENR_FE (0.0008636629208922386)
76. feature SE_PRE_ENRR_FE (0.0008475551148876548)
77. feature SH_STA_SMSS_RU_ZS (0.0008469861932098866)
78. feature SE_SEC_ENRR (0.0008359444909729064)
79. feature SL_EMP_VULN_MA_ZS (0.0008310644188895822)
80. feature SI_SPR_PCAP (0.0007915589376352727)
81. feature SP_DYN_LE00_MA_IN (0.0007896274328231812)
82. feature FM_AST_PRVT_GD_ZS (0.0007785715279169381)
83. feature SP_DYN_TFRT_IN (0.0007008419488556683)
84. feature SE_PRM_ENRL_TC_ZS (0.000695106340572238)
85. feature SH_H20_SMDW_UR_ZS (0.0006670821458101273)
86. feature EN_ATM_PM25_MC_T3_ZS (0.0006337789818644524)
87. feature SL_EMP_SELF_FE_ZS (0.0005979274865239859)
88. feature FB_CBK_BRWR_P3 (0.0005903344135731459)
89. feature SE_PRM CUAT_MA_ZS (0.0005829762667417526)
90. feature SH_DYN_MORT_FE (0.0005751053686253726)
91. feature NV_SRV_EMPL_KD (0.0005681756883859634)
92. feature SL_EMP_SELF_ZS (0.0005631537642329931)
93. feature SP_POP_7579_MA_5Y (0.0005451080505736172)
94. feature LP_LPI_CUST_XQ (0.0005373795866034925)
95. feature FB_CBK_DPTR_P3 (0.000509825476910919)
96. feature SP_POP_0509_FE_5Y (0.0004976686323061585)
97. feature SE_TER CUAT_BA_FE_ZS (0.00048822275130078197)
98. feature IQ_SPI_PIL4 (0.0004687319742515683)
99. feature SH_ANM_ALLW_ZS (0.00046522534103132784)
100. feature SH_TBS_DTEC_ZS (0.00046300076064653695)
101. feature SP_POP_2024_FE_5Y (0.00045744315139018)
102. feature FX_OWN_TOTL_YG_ZS (0.0004399557947181165)
103. feature SP_DYN_IMRT_IN (0.0004163759294897318)
104. feature SP_POP_6569_MA_5Y (0.0003972953127231449)
105. feature SE_TER_ENRR (0.0003922555479221046)
106. feature SP_POP_7074_FE_5Y (0.0003793713403865695)
107. feature NY_ADJ_NNTY_PC_CD (0.0003724454145412892)
108. feature SH_XPD_GHED_GE_ZS (0.00036602321779355407)
109. feature SP_ADO_TFRT (0.00034538115141913295)
110. feature SH_XPD_PVTD_PP_CD (0.00033913605147972703)
111. feature SL_SRV_EMPL_MA_ZS (0.0003312210028525442)
112. feature SP_POP_1519_FE_5Y (0.0003219239297322929)
113. feature SE_PRE_ENRR (0.00031464180210605264)
114. feature SP_POP_DPND_YG (0.0003135628649033606)
115. feature SP_DYN_LE00_FE_IN (0.00031111514545045793)
116. feature SP_POP_4549_FE_5Y (0.0003067078359890729)

117. feature SP_POP_7074_MA_5Y (0.00030579938902519643)
118. feature SP_DYN_AMRT_MA (0.00030361913377419114)
119. feature SP_POP_80UP_MA_5Y (0.00029753748094663024)
120. feature SP_POP_6569_FE_5Y (0.0002943875442724675)
121. feature SP_POP_65UP_FE_ZS (0.0002892186457756907)
122. feature EG_CFT_ACCS_ZS (0.00028917353483848274)
123. feature NE_CON_PRVT_PC_KD (0.00027855465305037796)
124. feature SH_DYN_MORT_MA (0.0002692743146326393)
125. feature EG_USE_ELEC_KH_PC (0.00026806804817169905)
126. feature SP_POP_0509_MA_5Y (0.000264921021880582)
127. feature SP_POP_4044_FE_5Y (0.0002629832015372813)
128. feature SI_POV_LMIC (0.00025868689408525825)
129. feature IC_EXP_TMDC (0.000256554689258337)
130. feature IQ_SPI_PIL5 (0.0002552580845076591)
131. feature SH_DYN_NMRT (0.0002464327262714505)
132. feature SE_TER CUAT BA ZS (0.00023895702906884253)
133. feature SH_DYN_1519 (0.00023352718562819064)
134. feature SP_POP_1519_MA_5Y (0.00022536172764375806)
135. feature DC_ODA_TLDC_GN_ZS (0.0002249151875730604)
136. feature SH_ANM_NPRG_ZS (0.00022358006390277296)
137. feature NY_GDP_PCAP_PP_CD (0.00021966702479403466)
138. feature SE_SEC_NENR_MA (0.0002085063752019778)
139. feature EG_CFT_ACCS_RU_ZS (0.00020674978441093117)
140. feature SE_LPV_PRIM_LD (0.00020464070257730782)
141. feature SP_DYN_CBRT_IN (0.00019982705998700112)
142. feature LP_LPI_LOGS_XQ (0.00019346170302014798)
143. feature SP_POP_0014_MA_ZS (0.00019010910182259977)
144. feature SH_XPD_GHED_PC_CD (0.00018109574739355594)
145. feature SE_SEC CUAT LO FE ZS (0.00017976659000851214)
146. feature SH_XPD_GHED_CH_ZS (0.0001723152818158269)
147. feature EG_CFT_ACCS_UR_ZS (0.00017199719150085002)
148. feature SH_SGR_IRSK_ZS (0.00017125993326772004)
149. feature SP_POP_4044_MA_5Y (0.00016842447803355753)
150. feature FS_AST_DOMO_GD_ZS (0.00016485252126585692)
151. feature SH_UHC_SRVS_CV_XD (0.0001602077973075211)
152. feature LP_LPI_INFR_XQ (0.00015568104572594166)
153. feature IT_MLT_MAIN_P2 (0.00015314552001655102)
154. feature SP_DYN_IMRT_FE_IN (0.00015273351164069027)
155. feature SP_POP_80UP_FE_5Y (0.00013836969446856529)
156. feature SE_SEC CUAT UP FE ZS (0.00013297764235176146)
157. feature SI_POV_UMIC_GP (0.0001290155341848731)
158. feature SE_PRM_PRS5_MA_ZS (0.00012052930105710402)
159. feature SH_STA_TRAF_P5 (0.00011645397171378136)
160. feature IC_IMP_TMBC (0.00011612851085374132)
161. feature SP_DYN_IMRT_MA_IN (0.00011467323929537088)
162. feature IT_NET_USER_ZS (0.00010689994087442756)
163. feature SE_SEC_CMPT_LO_FE_ZS (0.0001063461895682849)
164. feature SE_PRM CUAT ZS (0.00010537758498685434)

165. feature LP_LPI_ITRN_XQ (0.00010422895138617605)
166. feature NY_ADJ_NNTY_PC_KD (0.00010394361743237823)
167. feature SI_POV_MDIM_XQ (9.601299825590104e-05)
168. feature SE_SEC_CUAT_LO_MA_ZS (9.424517338629812e-05)
169. feature SH_XPD_CHEX_PP_CD (8.905595313990489e-05)
170. feature SH_PRG_ANEM (8.444210834568366e-05)
171. feature SI_POV_UMIC (8.393337338929996e-05)
172. feature SI_POV_MDIM_17 (8.292553684441373e-05)
173. feature SI_POV_MDIM (8.252855332102627e-05)
174. feature SP_URB_TOTL_IN_ZS (7.750095392111689e-05)
175. feature SE_TER_CUAT_DO_MA_ZS (7.691430073464289e-05)
176. feature SH_STA_AIRP_P5 (7.629996252944693e-05)
177. feature SP_POP_6064_MA_5Y (7.369340164586902e-05)
178. feature SE_SEC_NENR (7.158853259170428e-05)
179. feature SE_PRM_CUAT_FE_ZS (6.968402885831892e-05)
180. feature SE_TER_CUAT_MS_FE_ZS (6.835614476585761e-05)
181. feature HD_HCI_OVRL_UB (6.669055437669158e-05)
182. feature LP_LPI_TRAC_XQ (6.187924009282142e-05)
183. feature SE_SEC_CUAT_LO_ZS (5.980397327220999e-05)
184. feature SP_POP_1014_FE_5Y (5.94133889535442e-05)
185. feature SE_XPD_PRIM_ZS (5.907815648242831e-05)
186. feature SH_XPD_OOPC_PP_CD (5.286864688969217e-05)
187. feature SH_SGR_PROC_P5 (5.130211866344325e-05)
188. feature SH_ANM_CHLD_ZS (5.097526081954129e-05)
189. feature SP_POP_DPND_OL (4.979063669452444e-05)
190. feature NY_GNP_PCAP_PP_CD (4.972963506588712e-05)
191. feature SI_POV_MDIM_FE (4.969849032931961e-05)
192. feature SH_STA_STNT_FE_ZS (4.8702393542043865e-05)
193. feature HD_HCI_OVRL_UB_MA (4.837946835323237e-05)
194. feature SH_STA_AIRP_MA_P5 (4.721228469861671e-05)
195. feature SN_ITK_MSFI_ZS (4.715786053566262e-05)
196. feature SH_MED_NUMW_P3 (4.655495285987854e-05)
197. feature SE_PRM_PRS5_ZS (4.551088932203129e-05)
198. feature IC_FRM_CORR_ZS (4.5096709072822705e-05)
199. feature FX_OWN_TOTL_40_ZS (4.309223004383966e-05)
200. feature FX_OWN_TOTL_OL_ZS (4.237949542584829e-05)
201. feature SN_ITK_SVFI_ZS (4.174164496362209e-05)
202. feature FX_OWN_TOTL_PL_ZS (4.0440329030388966e-05)
203. feature SH_DTH_NCOM_ZS (3.9336737245321274e-05)
204. feature LP_LPI_TIME_XQ (3.869632200803608e-05)
205. feature EG_USE_PCAP_KG_OE (3.838496922980994e-05)
206. feature FX_OWN_TOTL_FE_ZS (3.694710903801024e-05)
207. feature SP_DYN_LE00_IN (3.487982394290157e-05)
208. feature SE_SEC_CUAT_UP_ZS (3.3877717214636505e-05)
209. feature SP_POP_SCIE_RD_P6 (3.200723949703388e-05)
210. feature SH_XPD_CHEX_PC_CD (3.123346687061712e-05)
211. feature SH_XPD_GHED_PP_CD (2.9431308576022275e-05)
212. feature SH_STA_AIRP_FE_P5 (2.844043410732411e-05)

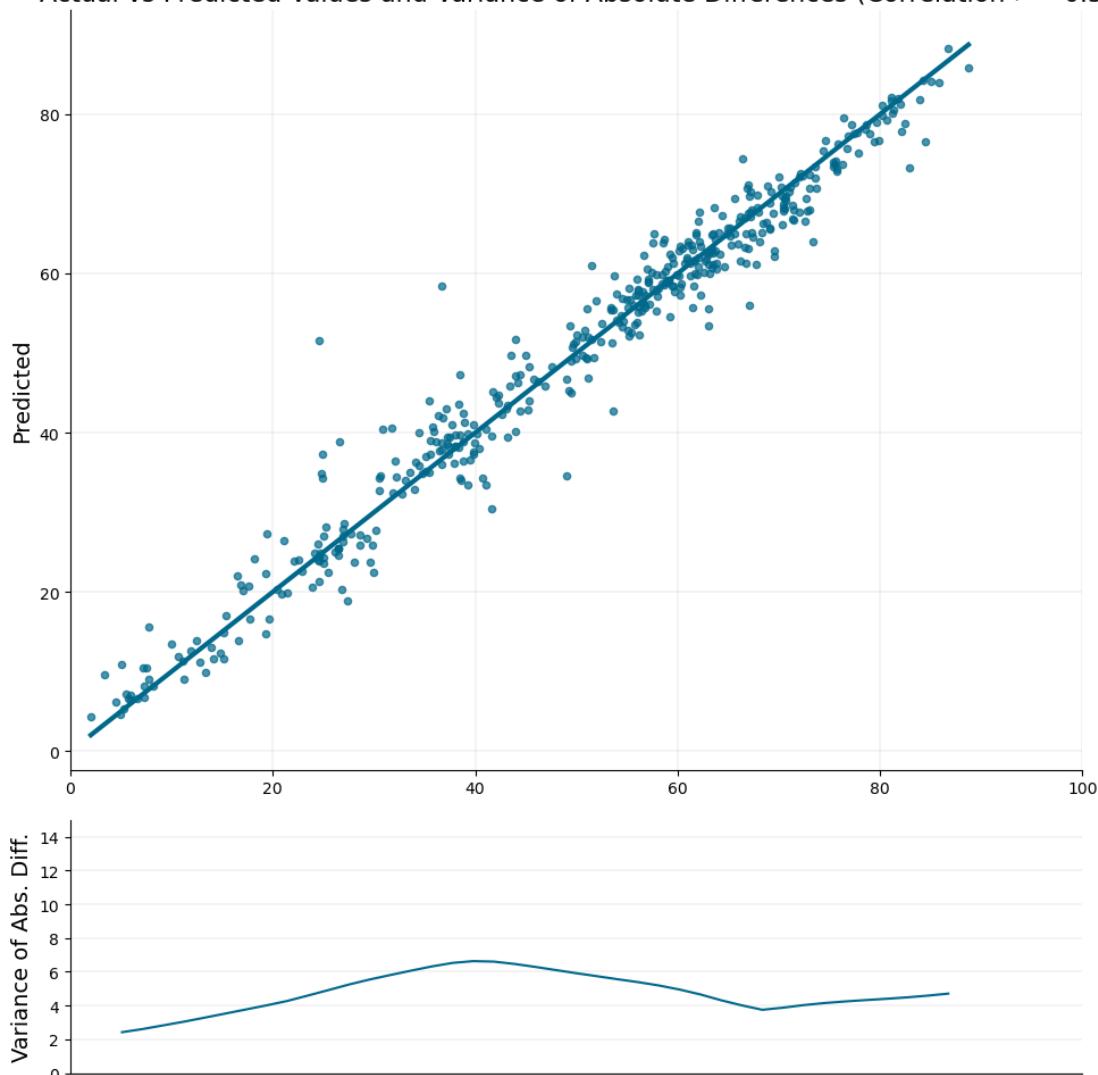
```

213. feature SE_TER CUAT BA MA ZS (2.767677142401226e-05)
214. feature SE_PRE ENRR MA (2.6208390409010462e-05)
215. feature SI_SPR PC40 (2.567703995737247e-05)
216. feature SE_LPV PRIM MA (2.2847902073408477e-05)
217. feature SH_DTH COMM ZS (2.220953319920227e-05)
218. feature SP_REG_BIRTH_ZS (2.1273108359309845e-05)
219. feature SH_DYN_MORT (2.09961726795882e-05)
220. feature SE_PRM_PRS5_FE_ZS (2.048303394985851e-05)
221. feature SP_POP_TECH_RD_P6 (1.9177039575879462e-05)
222. feature FX_OWN_TOTL_MA_ZS (1.8516513591748662e-05)
223. feature LP_LPI_OVRL_XQ (1.8001173884840682e-05)
224. feature SH_STA_STNT_ZS (1.7724905774230137e-05)
225. feature FX_OWN_TOTL_SO_ZS (1.729766336211469e-05)
226. feature SE_LPV_PRIM_LD_MA (1.7147061953437515e-05)
227. feature SP_POP_65UP_TO_ZS (1.6098494597827084e-05)
228. feature SI_POV_MDIM_MA (1.5415647794725373e-05)
229. feature SE_LPV_PRIM (1.3290084098116495e-05)
230. feature IC_FRM_BRIB_ZS (1.3157126886653714e-05)
231. feature SE_TER CUAT DO ZS (1.3132137610227801e-05)
232. feature HD_HCI_OVRL_UB_FE (1.3082737496006303e-05)
233. feature SH_STA_STNT_MA_ZS (1.288781004404882e-05)
234. feature HD_HCI_OVRL_LB_MA (1.0706388820835855e-05)
235. feature IC_BUS_EASE_XQ (1.0699781341827475e-05)
236. feature DC_ODA_TOTL_GN_ZS (1.0388048394815996e-05)
237. feature HD_HCI_OVRL_LB_FE (9.160081390291452e-06)
238. feature FX_OWN_TOTL_60_ZS (6.807796580687864e-06)
239. feature DT_NFL_UNWT_CD (5.111327936901944e-06)
240. feature HD_HCI_OVRL (4.512893156061182e-06)
241. feature SE_TER CUAT MS ZS (1.1736859732991434e-06)
242. feature SL_EMP_WORK_FE_ZS (0.0)
243. feature SL_EMP_WORK_ZS (0.0)
244. feature SE_TER CUAT DO FE ZS (0.0)
245. feature SE_TER CUAT MS MA ZS (0.0)
246. feature SI_POV_MDIM_17_XQ (0.0)
247. feature SE_LPV_PRIM_LD_FE (0.0)
248. feature FX_OWN_TOTL_ZS (0.0)
249. feature HD_HCI_OVRL_FE (0.0)
250. feature SE_LPV_PRIM_FE (0.0)
251. feature IC_FRM_METG_ZS (0.0)
252. feature HD_HCI_OVRL_MA (0.0)
253. feature HD_HCI_OVRL_LB (0.0)

Model with correlation >= 0.5:
Training+Validation R^2: 0.99968, RMSE: 0.35587
Testing R^2: 0.96505, RMSE: 3.77319
Mean cross-validation score: 0.95452

```

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.5)



Feature ranking:

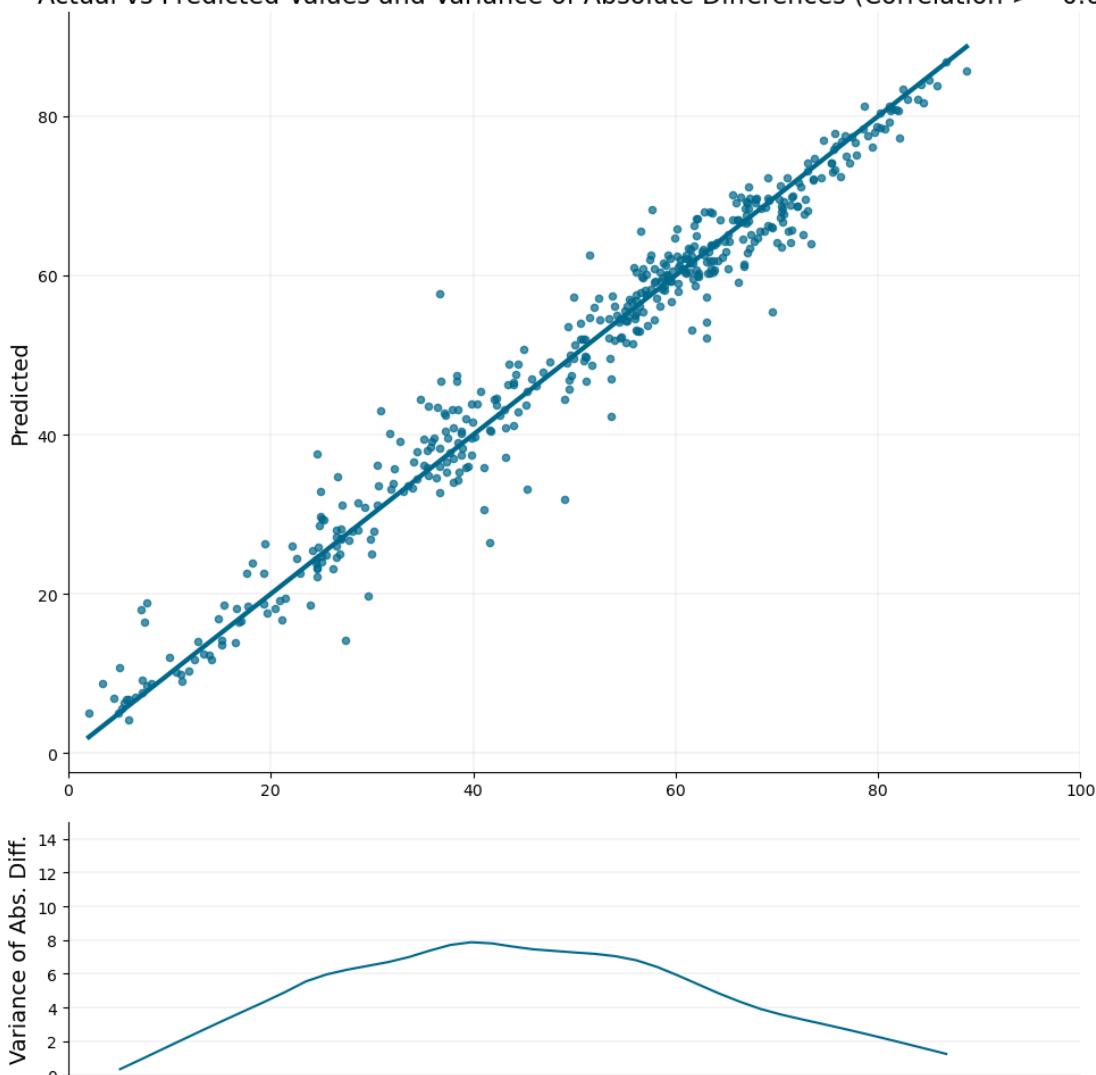
1. feature NY_GDP_PCAP_CD (0.31771016120910645)
2. feature NY_GDP_PCAP_KD (0.18727819621562958)
3. feature IQ_CPA_TRAN_XQ (0.1559630185365677)
4. feature SL_AGR_EMPL_ZS (0.026648659259080887)
5. feature IQ_CPA_PROP_XQ (0.025778453797101974)
6. feature SL_EMP_SELF_MA_ZS (0.021845590323209763)
7. feature NY_GNP_PCAP_CD (0.01952669583261013)
8. feature IT_NET_BBND_P2 (0.017185378819704056)
9. feature FB_CBK_BRWR_P3 (0.011316444724798203)
10. feature SL_GDP_PCAP_EM_KD (0.01081069465726614)
11. feature SP_DYN_LE00_IN (0.010012955404818058)
12. feature SL_UEM_NEET_FE_ZS (0.008948569186031818)

13. feature SH_XPD_GHED_GD_ZS (0.008236341178417206)
14. feature SL_SRV_EMPL_FE_ZS (0.007733013480901718)
15. feature SP_POP_80UP_FE_5Y (0.007290168199688196)
16. feature NE_CON_PRVT_PC_KD (0.006851824000477791)
17. feature SL_EMP_WORK_ZS (0.005982161499559879)
18. feature IC_BUS_DFRN_XQ (0.005447397939860821)
19. feature SP_POP_0004_MA_5Y (0.005414977204054594)
20. feature SH_H20_SMDW_ZS (0.005399579647928476)
21. feature SP_DYN_LE00_MA_IN (0.005092828534543514)
22. feature SP_POP_5054_MA_5Y (0.004977512173354626)
23. feature IQ_CPA_PUBS_XQ (0.004775524139404297)
24. feature SP_POP_0014_TO_ZS (0.00475804228335619)
25. feature SH_STA_SMSS_RU_ZS (0.004689306020736694)
26. feature SP_POP_0509_MA_5Y (0.0044582500122487545)
27. feature NY_GNP_PCAP_KD (0.004212925676256418)
28. feature EG_USE_ELEC_KH_PC (0.004133296199142933)
29. feature SE_SEC_ENRR_FE (0.003959030844271183)
30. feature NY_GNP_PCAP_PP_KD (0.003936504013836384)
31. feature SH_XPD_OOPC_PC_CD (0.0038321763277053833)
32. feature NY_GDP_PCAP_PP_CD (0.0036408863961696625)
33. feature SH_XPD_OOPC_PP_CD (0.0034151917789131403)
34. feature SH_DYN_NMRT (0.003081787843257189)
35. feature NY_ADJ_NNTY_PC_CD (0.003070663893595338)
36. feature SE_SEC_ENRR_MA (0.0026908242143690586)
37. feature SP_POP_1014_FE_5Y (0.002665147418156266)
38. feature SP_POP_1519_MA_5Y (0.0024645659141242504)
39. feature SP_POP_65UP_MA_ZS (0.002309795469045639)
40. feature SL_EMP_VULN_FE_ZS (0.0022584819234907627)
41. feature LP_LPI_CUST_XQ (0.0022542085498571396)
42. feature SP_DYN_LE00_FE_IN (0.00218269694596529)
43. feature SI_POV_MDIM (0.0020200933795422316)
44. feature SP_POP_SCIE_RD_P6 (0.001959689659997821)
45. feature FS_AST_DOMO_GD_ZS (0.001914836815558374)
46. feature SP_POP_DPND_YG (0.001897975686006248)
47. feature SH_XPD_GHED_CH_ZS (0.0018208209658041596)
48. feature PA_NUS_PPPC_RF (0.0018006728496402502)
49. feature FS_AST_DOMS_GD_ZS (0.001798510318621993)
50. feature SL_EMP_WORK_FE_ZS (0.0017848375719040632)
51. feature SP_POP_7579_MA_5Y (0.0017411555163562298)
52. feature SL_EMP_SELF_ZS (0.0017253487603738904)
53. feature SP_DYN_TO65_MA_ZS (0.0017227537464350462)
54. feature SH_STA_SMSS_ZS (0.0017118050018325448)
55. feature SL_SRV_EMPL_ZS (0.0017022638348862529)
56. feature SL_EMP_VULN_MA_ZS (0.0016550786094740033)
57. feature SH_XPD_CHEX_PC_CD (0.001615671208128333)
58. feature SH_XPD_GHED_PP_CD (0.0015449728816747665)
59. feature SH_ANM_CHLD_ZS (0.0015023780288174748)
60. feature SP_POP_5559_MA_5Y (0.0014944967115297914)

61. feature SP_POP_1014_MA_5Y (0.001486034132540226)
62. feature IT_MLT_MAIN_P2 (0.0013817090075463057)
63. feature NY_GDP_PCAP_PP_KD (0.0013467512326315045)
64. feature GB_XPD_RSDV_GD_ZS (0.0013449807884171605)
65. feature SE_SEC_ENRR (0.001202919171191752)
66. feature NV_SRV_EMPL_KD (0.001092569320462644)
67. feature EG_CFT_ACCS_RU_ZS (0.0009320040117017925)
68. feature SL_EMP_SELF_FE_ZS (0.0008829956059344113)
69. feature SL_AGR_EMPL_MA_ZS (0.0008685712236911058)
70. feature SL_EMP_WORK_MA_ZS (0.00079933280358091)
71. feature DC_ODA_TOTL_GN_ZS (0.0006371669005602598)
72. feature FX_OWN_TOTL_SO_ZS (0.0005882136174477637)
73. feature FX_OWN_TOTL_PL_ZS (0.0005777640617452562)
74. feature SH_PRG_ANEM (0.0005777503829449415)
75. feature NY_ADJ_NNTY_PC_KD (0.0005328395636752248)
76. feature SP_DYN_CBRT_IN (0.0004936057957820594)
77. feature LP_LPI_ITRN_XQ (0.0004783498588949442)
78. feature SI_POV_UMIC (0.00046340166591107845)
79. feature LP_LPI_LOGS_XQ (0.00045829705777578056)
80. feature FX_OWN_TOTL_DL_ZS (0.00041815961594693363)
81. feature NY_GNP_PCAP_PP_CD (0.0003846377949230373)
82. feature SH_ANM_ALLW_ZS (0.0003819888224825263)
83. feature SH_ANM_NPRG_ZS (0.0003712064935825765)
84. feature SP_POP_0014_FE_ZS (0.00036914058728143573)
85. feature SH_MED_NUMW_P3 (0.00035868349368683994)
86. feature HD_HCI_OVRL_UB_FE (0.0003542428894434124)
87. feature FX_OWN_TOTL_40_ZS (0.00032860960345715284)
88. feature SP_POP_TECH_RD_P6 (0.0003058721194975078)
89. feature SP_POP_0014_MA_ZS (0.00030180011526681483)
90. feature HD_HCI_OVRL_MA (0.00029032453312538564)
91. feature SL_EMP_VULN_ZS (0.0002775304892566055)
92. feature SN_ITK_MSFI_ZS (0.00026267702924087644)
93. feature SI_POV_MDIM_MA (0.00026254242402501404)
94. feature LP_LPI_TRAC_XQ (0.0002308694674866274)
95. feature SH_XPD_CHEX_PP_CD (0.0002206421340815723)
96. feature FX_OWN_TOTL_MA_ZS (0.00021653017029166222)
97. feature SP_POP_0509_FE_5Y (0.00021610195108223706)
98. feature LP_LPI_TIME_XQ (0.00020134223450440913)
99. feature LP_LPI_OVRL_XQ (0.00019268215692136437)
100. feature HD_HCI_OVRL_UB_MA (0.00016082289221230894)
101. feature SH_SGR_PROC_P5 (0.0001548499712953344)
102. feature IC_FRM_BRIB_ZS (0.00015360195538960397)
103. feature HD_HCI_OVRL_LB (0.00015075673582032323)
104. feature SI_SPR_PC40 (0.00013583428517449647)
105. feature SE_LPV_PRIM_LD_MA (0.00012228774721734226)
106. feature FX_OWN_TOTL_YG_ZS (0.0001076135304174386)
107. feature HD_HCI_OVRL_UB (0.00010185241262661293)
108. feature SE_LPV_PRIM_LD (0.00010168009612243623)

```
109. feature SI_POV_MDIM_17 (9.867953485809267e-05)
110. feature HD_HCI_OVRL_LB_MA (9.639330528443679e-05)
111. feature SH_XPD_GHED_PC_CD (9.114253771258518e-05)
112. feature SH_UHC_SRVS_CV_XD (9.055052942130715e-05)
113. feature SI_SPR_PCAP (8.838870417093858e-05)
114. feature HD_HCI_OVRL_LB_FE (8.196311682695523e-05)
115. feature FX_OWN_TOTL_60_ZS (7.867952808737755e-05)
116. feature SE_LPV_PRIM_LD_FE (7.60700786486268e-05)
117. feature FX_OWN_TOTL_FE_ZS (7.243661093525589e-05)
118. feature IC_BUS_EASE_XQ (6.393803050741553e-05)
119. feature LP_LPI_INFR_XQ (5.9841226175194606e-05)
120. feature SE_LPV_PRIM_FE (5.763019362348132e-05)
121. feature FX_OWN_TOTL_ZS (5.7393288443563506e-05)
122. feature SE_LPV_PRIM (1.9879564206348732e-05)
123. feature SE_LPV_PRIM_MA (1.949218312802259e-05)
124. feature SI_POV_MDIM_FE (7.09394043951761e-06)
125. feature DT_NFL_UNWT_CD (6.996914635237772e-06)
126. feature SI_POV_MDIM_17_XQ (0.0)
127. feature HD_HCI_OVRL (0.0)
128. feature HD_HCI_OVRL_FE (0.0)
Model with correlation >= 0.6:
Training+Validation R^2: 0.99961, RMSE: 0.39421
Testing R^2: 0.96229, RMSE: 3.9194
Mean cross-validation score: 0.94662
```

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.6)



Feature ranking:

1. feature NY_GDP_PCAP_KD (0.3659200966358185)
2. feature IQ_CPA_TRAN_XQ (0.28489449620246887)
3. feature IQ_CPA_PROP_XQ (0.11513756960630417)
4. feature NY_GNP_PCAP_CD (0.02071014791727066)
5. feature PA_NUS_PPPC_RF (0.014015117660164833)
6. feature SH_XPD_GHED_PP_CD (0.013600725680589676)
7. feature NE_CON_PRVT_PC_KD (0.013204297982156277)
8. feature FS_AST_DOMO_GD_ZS (0.010585211217403412)
9. feature NY_GNP_PCAP_KD (0.009999927133321762)
10. feature IQ_CPA_PUBS_XQ (0.009939268231391907)
11. feature LP_LPI_CUST_XQ (0.009131794795393944)
12. feature IT_MLT_MAIN_P2 (0.008918261155486107)

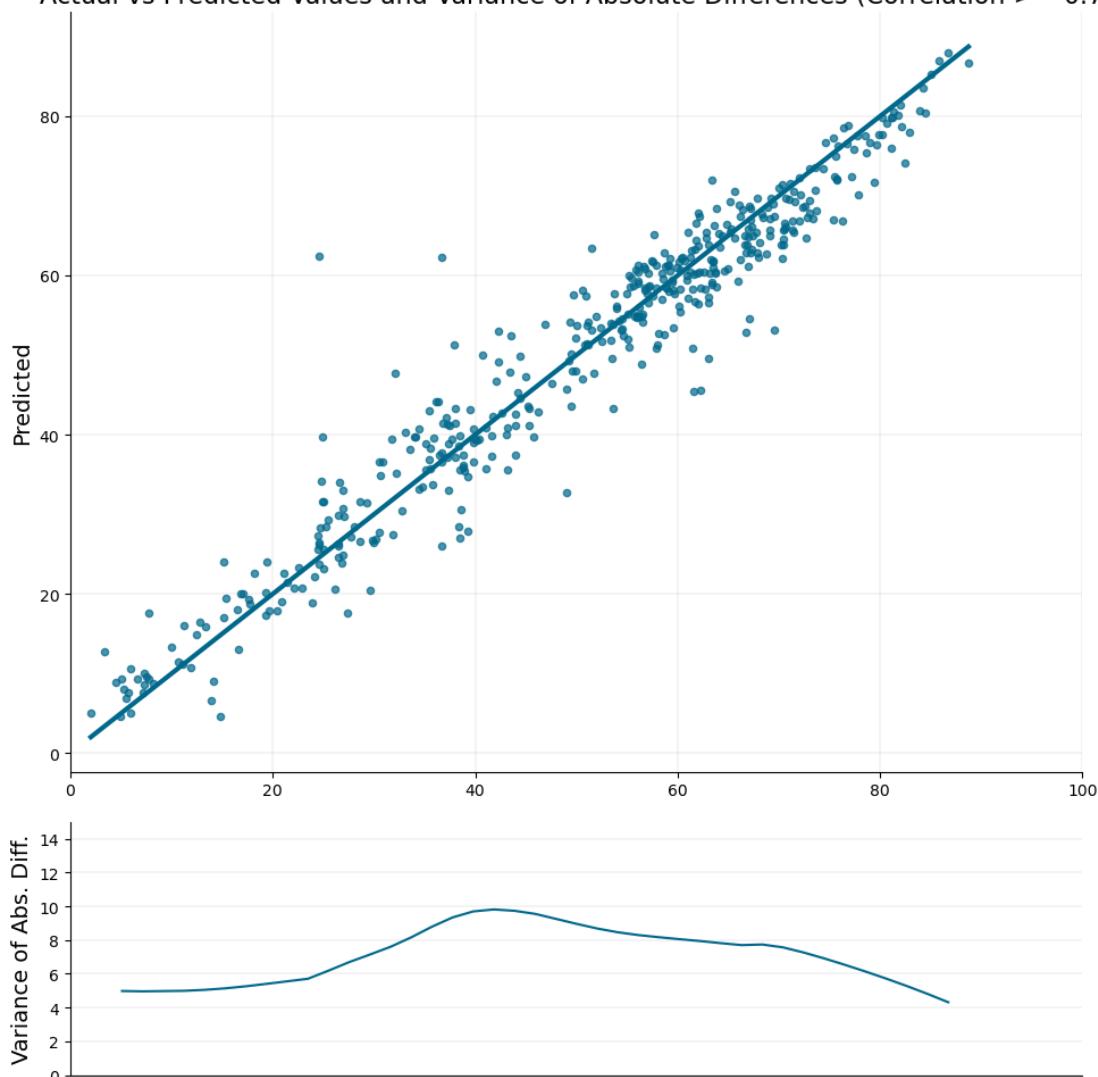
```

13. feature SP_POP_SCIE_RD_P6 (0.008812271058559418)
14. feature NY_ADJ_NNTY_PC_CD (0.007948868907988071)
15. feature FX_OWN_TOTL_ZS (0.007729346863925457)
16. feature NY_GDP_PCAP_PP_KD (0.007671836297959089)
17. feature NY_ADJ_NNTY_PC_KD (0.007223219610750675)
18. feature LP_LPI_OVRL_XQ (0.006975828669965267)
19. feature NY_GNP_PCAP_PP_KD (0.006220669951289892)
20. feature NV_SRV_EMPL_KD (0.006059700157493353)
21. feature IC_BUS_DFRN_XQ (0.00591591140255332)
22. feature SH_XPD_GHED_PC_CD (0.005635147448629141)
23. feature SH_XPD_CHEX_PP_CD (0.004717874340713024)
24. feature FX_OWN_TOTL_40_ZS (0.0038628443144261837)
25. feature FX_OWN_TOTL_YG_ZS (0.00358963874168694)
26. feature LP_LPI_LOGS_XQ (0.003124205395579338)
27. feature SH_XPD_CHEX_PC_CD (0.0030850954353809357)
28. feature SI_SPR_PC40 (0.0028595642652362585)
29. feature SE_LP_V_PRIM_LD_MA (0.0028463548514992)
30. feature HD_HCI_OVRL_LB_FE (0.0021189525723457336)
31. feature SI_POV_MDIM_17 (0.0021099888253957033)
32. feature LP_LPI_INFR_XQ (0.0019200618844479322)
33. feature HD_HCI_OVRL_FE (0.001764087239280343)
34. feature HD_HCI_OVRL_LB_MA (0.0017198780551552773)
35. feature SI_SPR_PCAP (0.0016804642509669065)
36. feature FX_OWN_TOTL_PL_ZS (0.0016485248925164342)
37. feature SE_LP_V_PRIM_MA (0.0015894847456365824)
38. feature HD_HCI_OVRL_LB (0.001539153279736638)
39. feature LP_LPI_TIME_XQ (0.0015252049779519439)
40. feature LP_LPI_TRAC_XQ (0.001436354941688478)
41. feature FX_OWN_TOTL_MA_ZS (0.0013349942164495587)
42. feature HD_HCI_OVRL_MA (0.0011333520524203777)
43. feature FX_OWN_TOTL_60_ZS (0.0010921345092356205)
44. feature HD_HCI_OVRL_UB (0.0010713303927332163)
45. feature HD_HCI_OVRL_UB_FE (0.001060275943018496)
46. feature FX_OWN_TOTL_SO_ZS (0.0009235357865691185)
47. feature SE_LP_V_PRIM_FE (0.0009167752577923238)
48. feature FX_OWN_TOTL_FE_ZS (0.0008743382059037685)
49. feature HD_HCI_OVRL_UB_MA (0.0006285422714427114)
50. feature FX_OWN_TOTL_OL_ZS (0.0005353413871489465)
51. feature SE_LP_V_PRIM (0.0003807509783655405)
52. feature SE_LP_V_PRIM_LD (0.0003074120613746345)
53. feature HD_HCI_OVRL (0.0002749039267655462)
54. feature DT_NFL_UNWT_CD (7.89538025856018e-05)

Model with correlation >= 0.7:
Training+Validation R^2: 0.99447, RMSE: 1.48281
Testing R^2: 0.93749, RMSE: 5.0465
Mean cross-validation score: 0.90218

```

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.7)



Feature ranking:

1. feature IQ_CPA_TRAN_XQ (0.4924904704093933)
2. feature NE_CON_PRVT_PC_KD (0.3603402078151703)
3. feature LP_LPI_CUST_XQ (0.05750051885843277)
4. feature NY_ADJ_NNTY_PC_KD (0.041945330798625946)
5. feature SI_SPR_PC40 (0.0327630490064621)
6. feature SI_SPR_PCAP (0.01222801674157381)
7. feature DT_NFL_UNWT_CD (0.002732437802478671)

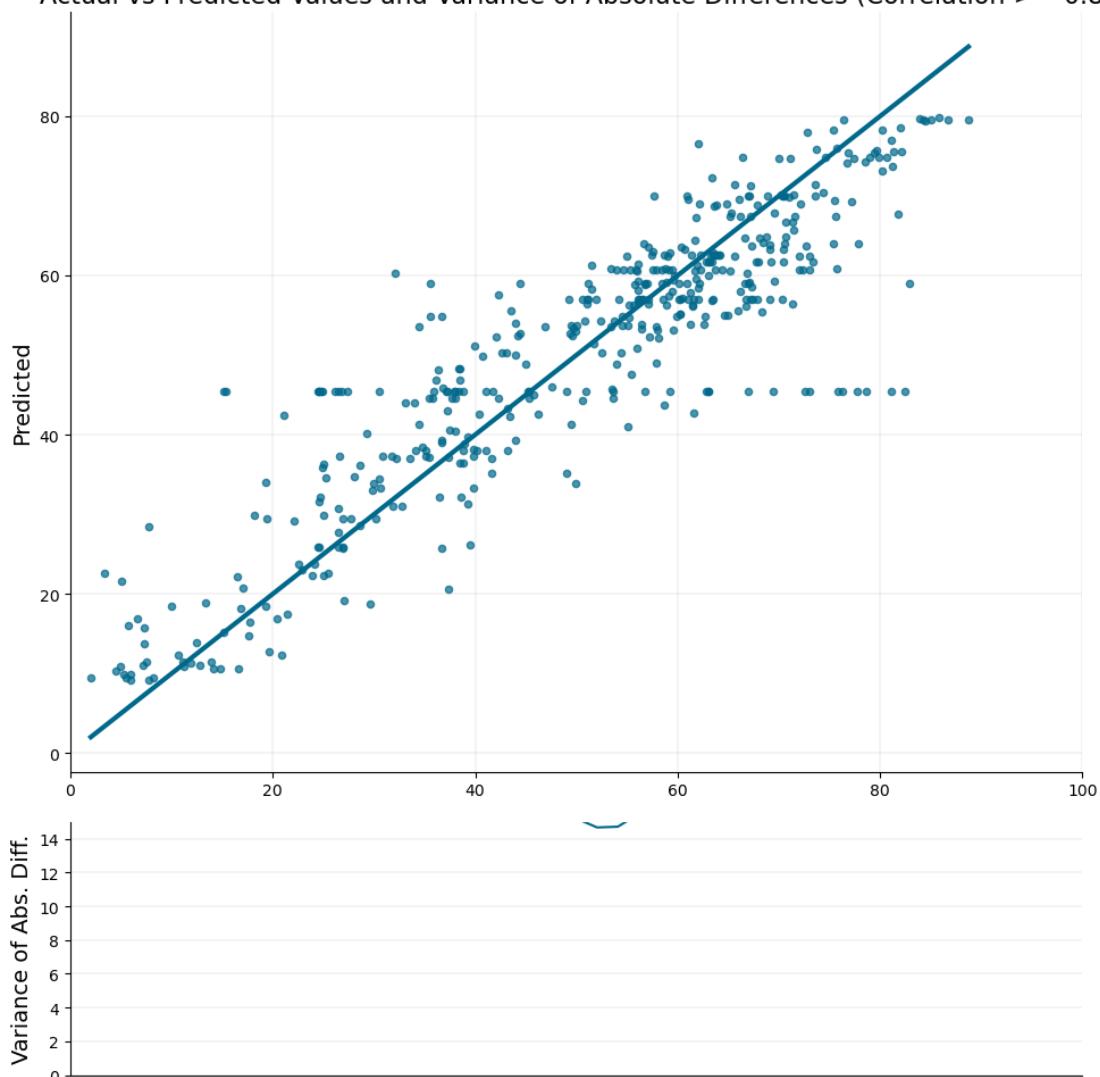
Model with correlation ≥ 0.8 :

Training+Validation R^2 : 0.81193, RMSE: 8.64535

Testing R^2 : 0.79896, RMSE: 9.04992

Mean cross-validation score: 0.75062

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.8)



Feature ranking:

1. feature DT_NFL_UNWT_CD (1.0)

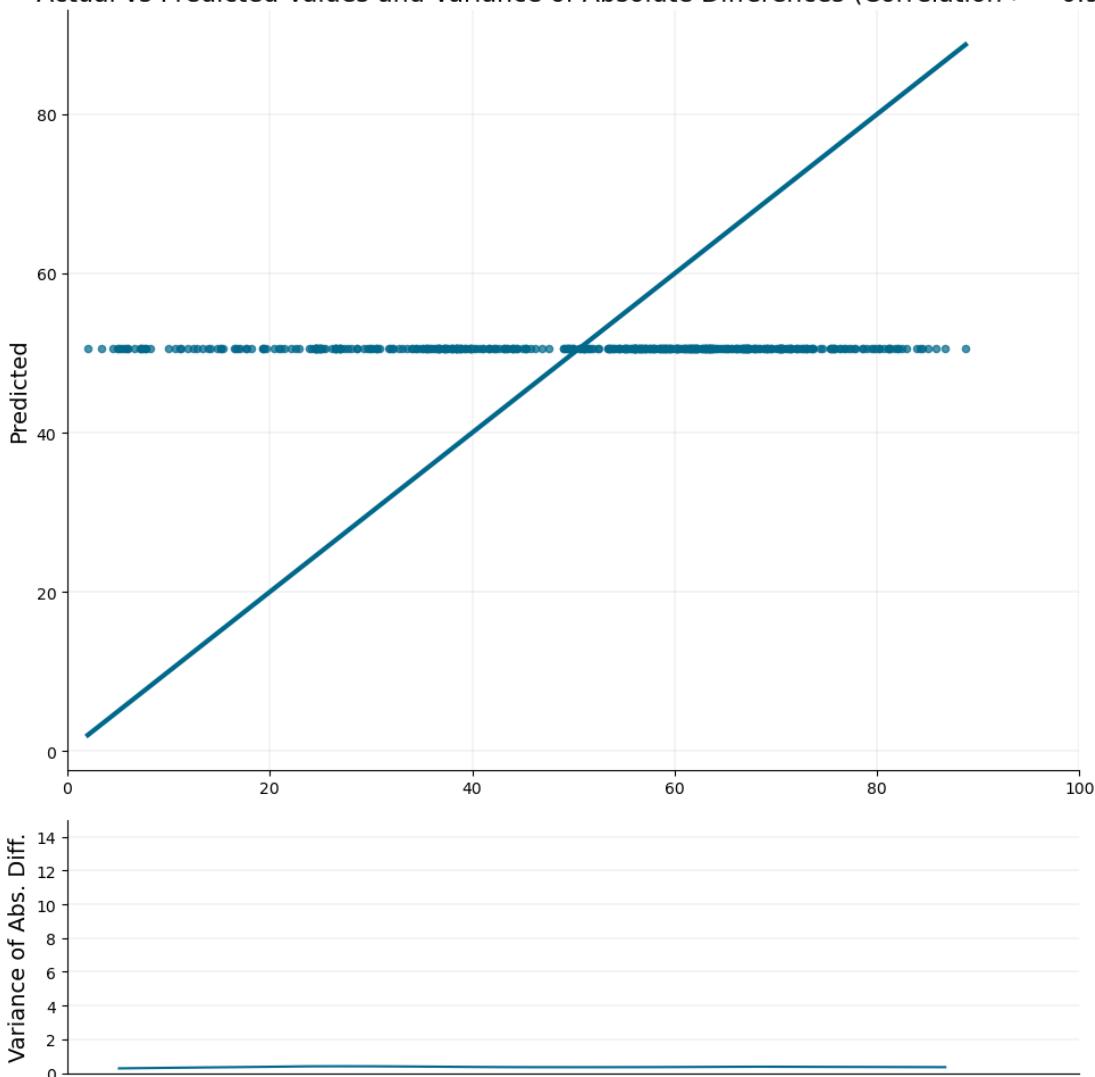
Model with correlation ≥ 0.9 :

Training+Validation R²: 0.00111, RMSE: 19.92421

Testing R²: -0.00041, RMSE: 20.18809

Mean cross-validation score: -0.00175

Actual vs Predicted Values and Variance of Absolute Differences (Correlation ≥ 0.9)



```
[118]: sns.set_theme()

def custom_cross_val_score(model, X, y, cv):
    kf = KFold(n_splits=cv, shuffle=True, random_state=0)
    scores = []

    for train_index, val_index in kf.split(X):
        x_train, x_val = X[train_index], X[val_index]
        y_train, y_val = y[train_index], y[val_index]

        model.fit(x_train, y_train, eval_set=[(x_val, y_val)], verbose=False)
        y_pred = model.predict(x_val)
        scores.append(r2_score(y_val, y_pred))
```

```

    return np.array(scores)

datasets = [(x5, y), (x6, y), (x7, y), (x8, y), (x9, y)]

feature_importances_dfs = {}

# Prepare a list of variable lists
vars_list = [vars_5, vars_6, vars_7, vars_8, vars_9]

r2_train_vals = []
r2_tests = []
rmse_train_vals = []
rmse_tests = []
mean_cv_scores = []

# Loop over the datasets
for i, (x, y) in enumerate(datasets, start=5):
    # Split the data into training+validation and testing sets
    x_train_val, x_test, y_train_val, y_test = train_test_split(x, y, □
    ↵test_size=0.2, random_state=0)

    # Initialize the model with L1 regularization (alpha is the regularization parameter)
    model = XGBRegressor(objective='reg:squarederror', random_state=0, alpha=0.1, early_stopping_rounds=10)

    # Fit the model
    model.fit(x_train_val, y_train_val, eval_set=[(x_test, y_test)], □
    ↵verbose=False)

    # Create a DataFrame with the feature importances
    feature_importances = pd.DataFrame({
        'Feature': vars_list[i-5],
        'Importance': model.feature_importances_
    })

    # Sort the DataFrame by importance in descending order
    feature_importances = feature_importances.sort_values(by='Importance', □
    ↵ascending=False)

    # Store the DataFrame in the dictionary
    feature_importances_dfs[f'feature_importances_{i}'] = feature_importances

# Make predictions
y_train_val_pred = model.predict(x_train_val)
y_test_pred = model.predict(x_test)

```

```

# Calculate R^2 scores and round to 5 decimal places
r2_train_val = round(r2_score(y_train_val, y_train_val_pred), 5)
r2_test = round(r2_score(y_test, y_test_pred), 5)

# Calculate RMSE and round to 5 decimal places
rmse_train_val = round(np.sqrt(mean_squared_error(y_train_val, y_train_val_pred)), 5)
rmse_test = round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 5)

# Perform cross-validation and calculate mean score
cv_scores = custom_cross_val_score(model, x_train_val, y_train_val, cv=5)
mean_cv_score = round(np.mean(cv_scores), 5)

r2_train_vals.append(r2_train_val)
r2_tests.append(r2_test)
rmse_train_vals.append(rmse_train_val)
rmse_tests.append(rmse_test)
mean_cv_scores.append(mean_cv_score)

print(f"Model with correlation >= 0.{i}:")
print(f"Training+Validation R^2: {r2_train_val}, RMSE: {rmse_train_val}")
print(f"Testing R^2: {r2_test}, RMSE: {rmse_test}")
print(f"Mean cross-validation score: {mean_cv_score}\n")

# Print the sorted feature importances
print(f"Feature importances for model with correlation >= 0.{i}:")
print(feature_importances)
print("\n")

```

Model with correlation >= 0.5:
 Training+Validation R^2: 0.99968, RMSE: 0.35587
 Testing R^2: 0.96505, RMSE: 3.77319
 Mean cross-validation score: 0.95452

Feature importances for model with correlation >= 0.5:

	Feature	Importance
199	NY_GDP_PCAP_CD	0.344131
210	NY_GDP_PCAP_KD	0.231385
252	IQ_CPA_TRAN_XQ	0.115351
186	SL_EMP_WORK_MA_ZS	0.017028
237	IQ_CPA_PROP_XQ	0.013965
..
216	HD_HCI_OVRL_FE	0.000000
3	SE_LPV_PRIM_FE	0.000000
105	SE_TER CUAT DO FE ZS	0.000000
222	HD_HCI_OVRL_MA	0.000000
193	SL_EMP_WORK_FE_ZS	0.000000

[253 rows x 2 columns]

Model with correlation >= 0.6:
Training+Validation R^2: 0.99961, RMSE: 0.39421
Testing R^2: 0.96229, RMSE: 3.9194
Mean cross-validation score: 0.94662

Feature importances for model with correlation >= 0.6:

	Feature	Importance
74	NY_GDP_PCAP_CD	0.317710
85	NY_GDP_PCAP_KD	0.187278
127	IQ_CPA_TRAN_XQ	0.155963
32	SL_AGR_EMPL_ZS	0.026649
112	IQ_CPA_PROP_XQ	0.025778
..
21	SI_POV_MDIM_FE	0.000007
0	DT_NFL_UNWT_CD	0.000007
100	HD_HCI_OVRL	0.000000
91	HD_HCI_OVRL_FE	0.000000
8	SI_POV_MDIM_17_XQ	0.000000

[128 rows x 2 columns]

Model with correlation >= 0.7:
Training+Validation R^2: 0.99447, RMSE: 1.48281
Testing R^2: 0.93749, RMSE: 5.0465
Mean cross-validation score: 0.90218

Feature importances for model with correlation >= 0.7:

	Feature	Importance
11	NY_GDP_PCAP_KD	0.365920
53	IQ_CPA_TRAN_XQ	0.284894
38	IQ_CPA_PROP_XQ	0.115138
20	NY_GNP_PCAP_CD	0.020710
28	PA_NUS_PPPC_RF	0.014015
31	SH_XPD_GHED_PP_CD	0.013601
48	NE_CON_PRVT_PC_KD	0.013204
13	FS_AST_DOMO_GD_ZS	0.010585
47	NY_GNP_PCAP_KD	0.010000
32	IQ_CPA_PUBS_XQ	0.009939
49	LP_LPI_CUST_XQ	0.009132
12	IT_MLT_MAIN_P2	0.008918
41	SP_POP_SCIE_RD_P6	0.008812
40	NY_ADJ_NNTY_PC_CD	0.007949
34	FX_OWN_TOTL_ZS	0.007729

8	NY_GDP_PCAP_PP_KD	0.007672
50	NY_ADJ_NNTY_PC_KD	0.007223
45	LP_LPI_OVRL_XQ	0.006976
44	NY_GNP_PCAP_PP_KD	0.006221
42	NV_SRV_EMPL_KD	0.006060
7	IC_BUS_DFRN_XQ	0.005916
19	SH_XPD_GHED_PC_CD	0.005635
33	SH_XPD_CHEX_PP_CD	0.004718
37	FX_OWN_TOTL_40_ZS	0.003863
30	FX_OWN_TOTL_YG_ZS	0.003590
43	LP_LPI_LOGS_XQ	0.003124
22	SH_XPD_CHEX_PC_CD	0.003085
52	SI_SPR_PC40	0.002860
6	SE_LP_V_PRIM_LD_MA	0.002846
16	HD_HCI_OVRL_LB_FE	0.002119
5	SI_POV_MDIM_17	0.002110
46	LP_LPI_INFR_XQ	0.001920
17	HD_HCI_OVRL_FE	0.001764
25	HD_HCI_OVRL_LB_MA	0.001720
51	SI_SPR_PCAP	0.001680
39	FX_OWN_TOTL_PL_ZS	0.001649
2	SE_LP_V_PRIM_MA	0.001589
27	HD_HCI_OVRL_LB	0.001539
10	LP_LPI_TIME_XQ	0.001525
35	LP_LPI_TRAC_XQ	0.001436
18	FX_OWN_TOTL_MA_ZS	0.001335
23	HD_HCI_OVRL_MA	0.001133
14	FX_OWN_TOTL_60_ZS	0.001092
24	HD_HCI_OVRL_UB	0.001071
15	HD_HCI_OVRL_UB_FE	0.001060
9	FX_OWN_TOTL_SO_ZS	0.000924
3	SE_LP_V_PRIM_FE	0.000917
36	FX_OWN_TOTL_FE_ZS	0.000874
21	HD_HCI_OVRL_UB_MA	0.000629
29	FX_OWN_TOTL_OL_ZS	0.000535
1	SE_LP_V_PRIM	0.000381
4	SE_LP_V_PRIM_LD	0.000307
26	HD_HCI_OVRL	0.000275
0	DT_NFL_UNWT_CD	0.000079

Model with correlation >= 0.8:
 Training+Validation R^2: 0.81193, RMSE: 8.64535
 Testing R^2: 0.79896, RMSE: 9.04992
 Mean cross-validation score: 0.75062

Feature importances for model with correlation >= 0.8:

Feature	Importance
---------	------------

```

6      IQ_CPA_TRAN_XQ    0.492490
1  NE_CON_PRVT_PC_KD    0.360340
2      LP_LPI_CUST_XQ    0.057501
3  NY_ADJ_NNTY_PC_KD    0.041945
5      SI_SPR_PC40    0.032763
4      SI_SPR_PCAP    0.012228
0      DTNFL_UNWT_CD    0.002732

```

```

Model with correlation >= 0.9:
Training+Validation R^2: 0.00111, RMSE: 19.92421
Testing R^2: -0.00041, RMSE: 20.18809
Mean cross-validation score: -0.00175

```

```

Feature importances for model with correlation >= 0.9:
      Feature  Importance
0  DTNFL_UNWT_CD        1.0

```

```
[119]: # How to view each dataframe of the variable by its respective importance in predicting corruption
feature_importances_dfs['feature_importances_5']
```

```

[119]:          Feature  Importance
 199      NY_GDP_PCAP_CD    0.344131
 210      NY_GDP_PCAP_KD    0.231385
 252      IQ_CPA_TRAN_XQ    0.115351
 186      SL_EMP_WORK_MA_ZS    0.017028
 237      IQ_CPA_PROP_XQ    0.013965
 ...
 ...
 216      HD_HCI_OVRL_FE    0.000000
 3       SE_LPV_PRIM_FE    0.000000
 105     SE_TER CUAT DO FE_ZS    0.000000
 222      HD_HCI_OVRL_MA    0.000000
 193     SL_EMP_WORK_FE_ZS    0.000000

```

[253 rows x 2 columns]

```
[120]: df = pd.read_csv('WDI_data.csv')
df = df.drop(columns=['CPI_EST'])
# Make corruption more intuitive with higher values indicating more corruption
df['CC_EST'] = (-1)*df['CC_EST']
df['CC_EST'] = (df['CC_EST'] + 2.5) * 20
# Drop years 1960-2011 if column 'year' is less than 2012
df_dr = df[df['year'] >= 2012]
```

```
[121]: numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()

# Calculate the correlation of each numeric feature with CC_EST
correlations = df[numeric_cols].corr()['CC_EST']

# Keep only features with a correlation greater than 0.5
vars_full = correlations[correlations.abs() > 0.5].index.tolist()

# Remove 'iso3c', 'iso3n', 'year', and 'CC_EST' from vars_full if they're included
vars_full = [var for var in vars_full if var not in ['iso3c', 'iso3n', 'year', 'CC_EST', 'country']]

# Get the unique country codes
countries = df['iso2c'].unique()

datasets = [(x5, y)]

feature_importances_dfs = {}

# Prepare a list of variable lists
vars_list = [vars_5]

# Create a new dataframe to store the results
df_new = pd.DataFrame()

r2_train_vals = []
r2_tests = []
rmse_train_vals = []
rmse_tests = []
mean_cv_scores = []

# feature_names_list = [x.columns.tolist() for x, y in datasets]

for i, ((x, y), feature_names) in enumerate(zip(datasets, vars_list), start=5):
    # Use vars_full as model_features
    model_features = [feature for feature in vars_full if feature not in ['iso2c', 'country', 'year', 'CC_EST']]

    x_train_val, x_test, y_train_val, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

    # Initialize the model with L1 regularization (alpha is the regularization parameter)
    model = XGBRegressor(objective='reg:squarederror', random_state=0, alpha=0.1, early_stopping_rounds=10)
```

```

# Fit the model
model.fit(x_train_val, y_train_val, eval_set=[(x_test, y_test)], verbose=False)

# Create a DataFrame with the feature importances
feature_importances = pd.DataFrame({
    'Feature': vars_list[i-5],
    'Importance': model.feature_importances_
})

# Backcast CC_EST to 1960 using the trained model
for country in tqdm(countries):
    # Get the data for the current country and create a copy of it
    df_country = df[df['iso2c'] == country].copy()

    # Add a new feature for the previous year's CC_EST value
    df_country['CC_EST_prev'] = df_country.groupby('iso2c')['CC_EST'].shift()

    # Sort the data in descending order of the year
    df_country = df_country.sort_values('year', ascending=False)

    # Get the latest year in the data
    latest_year = df_country['year'].max()

    # Skip the current country if all its 'year' values are NaN
    if np.isnan(latest_year):
        continue

    latest_year = int(latest_year)

    # Iterate from the latest year to 1960
    for year in range(latest_year, 1959, -1):
        # Check if CC_EST for the current year is NaN
        if df_country.loc[df_country['year'] == year, 'CC_EST'].isna().any():

            # Prepare the features for prediction
            X = df_country[df_country['year'] == year][vars_5]

            X = X.fillna()

            if X.shape[1] != len(vars_5):
                extra_features = set(X.columns) - set(vars_5)
                X = X.drop(columns=extra_features)

            if X.shape[1] != len(vars_5):

```

```

        raise ValueError(f"Expected {len(vars_5)} features, but got {X.shape[1]}")

    # Make predictions
    y_pred = model.predict(X)

    # Update the data with the predicted value
    df_country.loc[df_country['year'] == year, 'CC_EST'] = y_pred

    # Append the data for the current country to the new dataframe
    df_new = pd.concat([df_new, df_country])

```

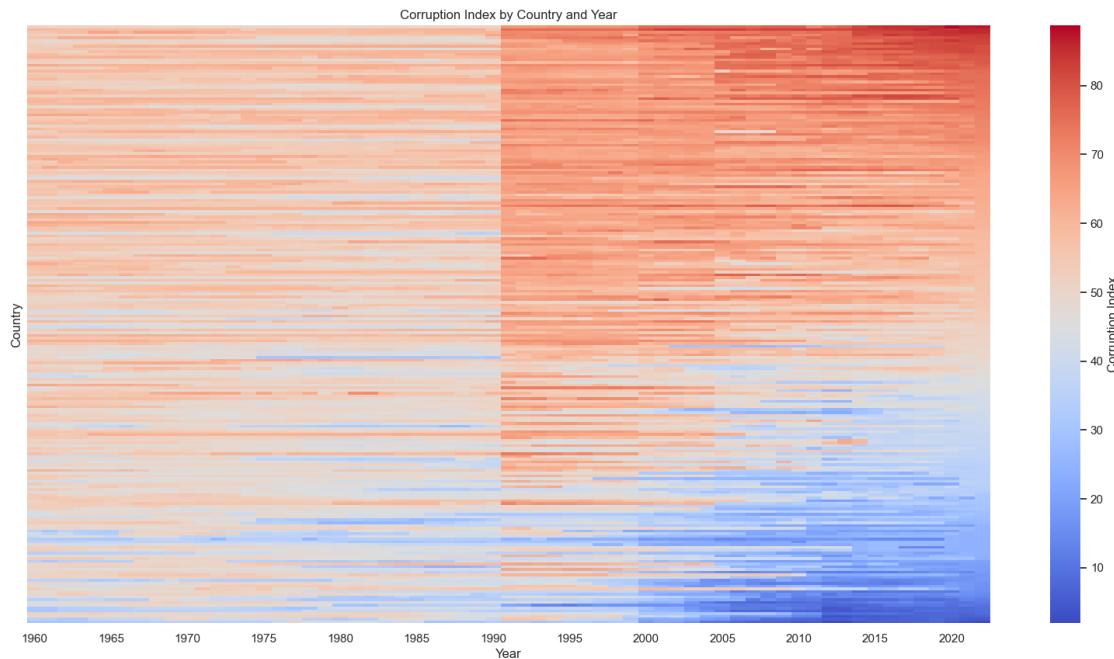
100% | 218/218 [01:58<00:00, 1.84it/s]

```

[122]: # Graphically illustrate the generale moves of country and corruption levels over time
# Create a pivot table of the data for the heatmap
df_heatmap = df_new.pivot(index='country', columns='year', values='CC_EST')
df_heatmap = df_heatmap.sort_values(by=2022, ascending=False)

# Create a heatmap of the data, dont show country labels, and 1970 to 2022
plt.figure(figsize=(20, 10))
sns.heatmap(df_heatmap, cmap='coolwarm', cbar_kws={'label': 'Corruption Index'}, xticklabels=5, yticklabels=False)
plt.xlim(0, 63)
plt.title('Corruption Index by Country and Year')
plt.xlabel('Year')
plt.ylabel('Country')
plt.show()

```



```
[127]: # Get the underlying Booster object
booster = model.get_booster()

# Print out a text representation of the first tree
print(booster.get_dump()[0])
```

AttributeError Cell In[127], line 2 <code>1 # Get the underlying Booster object</code> <code>----> 2 booster = model.get_booster()</code> <code>4 # Print out a text representation of the first tree</code> <code>5 print(booster.get_dump()[0])</code>	Traceback (most recent call last) <code>AttributeError: 'RandomForestModel' object has no attribute 'get_booster'</code>
---	--

```
[128]: style.use('default')

# Plot the distribution of the aggregate forecasted corruption levels
# Calculate the mean and median of the forecasted corruption levels
mean_forecast = df_new.groupby('year')['CC_EST'].mean()
median_forecast = df_new.groupby('year')['CC_EST'].median()

# Plot the distribution of the forecasted corruption levels
plt.figure(figsize=(20, 10))
```

```

# Use seaborn's histplot function to plot the histogram
# Set the color, edgecolor and linewidth for the bars
sns.histplot(df_new['CC_EST'], bins=20, kde=True, color='#00688B',
             edgecolor='#00688B', linewidth=1)

# Calculate the mean and median of the forecasted corruption levels
mean_est = df_new['CC_EST'].mean()
median_est = df_new['CC_EST'].median()

# Add a vertical line at the mean
plt.axvline(mean_est, color='#00688B', linestyle='--', linewidth=2)

# Add a vertical line at the median
plt.axvline(median_est, color='#00688B', linestyle='-', linewidth=2)

# Set the labels and title with larger fonts
plt.xlabel('Forecasted Corruption Index', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Forecasted Corruption Index', fontsize=16)
plt.xlim(0,100)

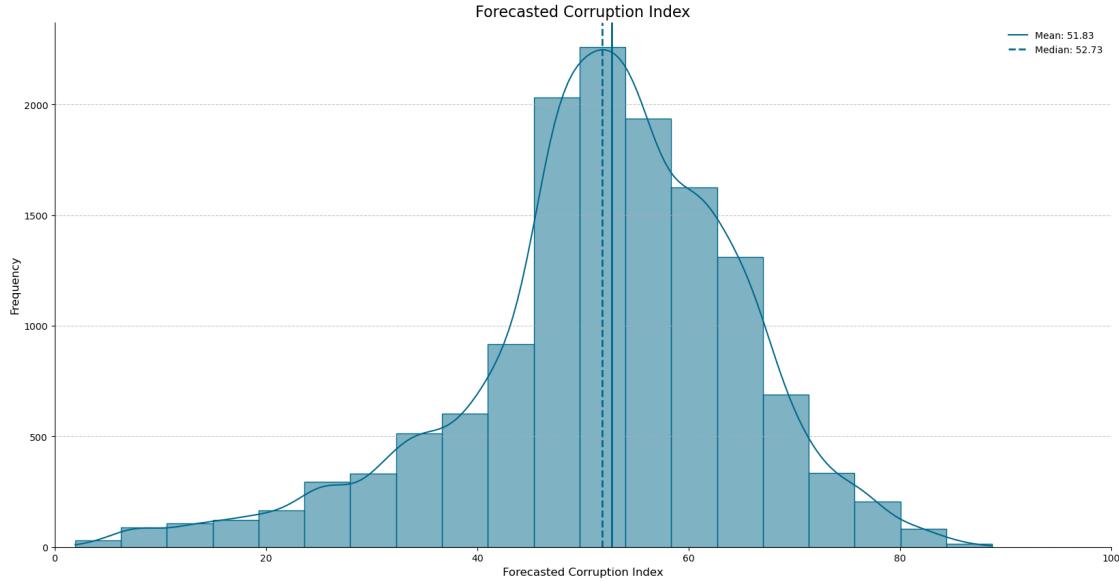
# Set the grid style
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Add a legend for mean and median
plt.legend([f'Mean: {mean_est:.2f}', f'Median: {median_est:.2f}'], □
           frameon=False)

# Remove top and right spines
sns.despine()

# Show the plot
plt.show()

```

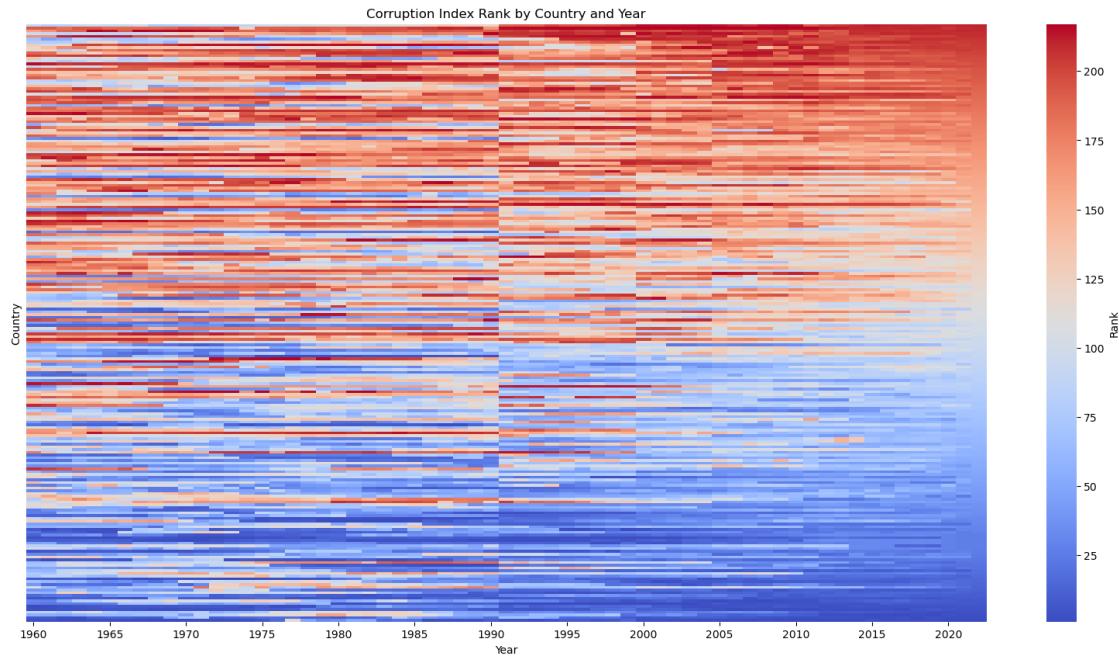


```
[129]: # Give a relative rank of CC_EST for each country in each year
df_new['CC_EST_rank'] = df_new.groupby('year')['CC_EST'].rank(pct=True)

# Give a rank of CC_EST for each country in each year as integers
df_new['CC_EST_rank_int'] = df_new.groupby('year')['CC_EST'].
    ↪rank(method='dense', ascending=True)

# Plot changes in country ranks over time
# Create a pivot table of the data for the heatmap
df_heatmap = df_new.pivot(index='country', columns='year',
    ↪values='CC_EST_rank_int')
df_heatmap = df_heatmap.sort_values(by=2022, ascending=False)

# Create a heatmap of the data, don't show country labels, and 1970 to 2022,
    ↪ranking from lowest to highest in 2022
plt.figure(figsize=(20, 10))
sns.heatmap(df_heatmap, cmap='coolwarm', cbar_kws={'label': 'Rank'},
    ↪xticklabels=5, yticklabels=False)
plt.xlim(0, 63)
plt.title('Corruption Index Rank by Country and Year')
plt.xlabel('Year')
plt.ylabel('Country')
plt.show()
```



3 Model 2

```
[130]: df = pd.read_csv('WDI_data.csv')
df = df.drop(columns=['CPI_EST'])
# Make corruption more intuitive with higher values indicating more corruption
df['CC_EST'] = (-1)*df['CC_EST']
df['CC_EST'] = (df['CC_EST'] + 2.5) * 20
# Drop years 170-2011 if column 'year' is less than 2012
df_dr = df[df['year'] >= 2012]
# Make years integers
df['year'] = df['year'].astype(int)

[131]: # Create a new dataframe with only iso2c, country, and CC_EST
df_corr = df_dr[['iso2c', 'country', 'year', 'CC_EST']]

# Filter the data for the years 2012 and 2022
df_filtered = df_corr[df_corr['year'].isin([2012, 2022])]

[132]: # Calculate the linear differences over time for each given country
change_by_country = df_filtered.pivot_table(index='iso2c', columns='year',
                                             values='CC_EST', aggfunc='first')

# Calculate the change for each country
change_by_country['avg_change'] = (change_by_country[2012] - change_by_country[2012]) / 10
```

```

# Calculate the mean and median CC_EST values across all years for each country
mean_cc_est_by_country = df_filtered.groupby('iso2c')['CC_EST'].mean()
median_cc_est_by_country = df_filtered.groupby('iso2c')['CC_EST'].median()

# Merge with the original DataFrame to get the country names
df_trend = pd.merge(change_by_country['avg_change'], mean_cc_est_by_country, □
    ↪left_index=True, right_index=True, how='left')
df_trend = pd.merge(df_trend, median_cc_est_by_country, left_index=True, □
    ↪right_index=True, how='left')

# Reset index and rename columns
df_trend.reset_index(inplace=True)
df_trend.rename(columns={'CC_EST_x': 'mean_CC_EST', 'CC_EST_y': □
    ↪'median_CC_EST'}, inplace=True)

# Include country names
df_trend = pd.merge(df_trend, df_filtered[['iso2c', 'country']]. □
    ↪drop_duplicates(), on='iso2c', how='left')

print(df_trend)

```

	iso2c	avg_change	mean_CC_EST	median_CC_EST	country
0	AD	0.0	24.692656	24.692656	Andorra
1	AE	0.0	26.889273	26.889273	United Arab Emirates
2	AF	0.0	76.141493	76.141493	Afghanistan
3	AG	0.0	34.288647	34.288647	Antigua and Barbuda
4	AL	0.0	61.866050	61.866050	Albania
..
199	XK	0.0	59.171521	59.171521	Kosovo
200	YE	0.0	79.383066	79.383066	Yemen, Rep.
201	ZA	0.0	55.039358	55.039358	South Africa
202	ZM	0.0	58.206411	58.206411	Zambia
203	ZW	0.0	76.369429	76.369429	Zimbabwe

[204 rows x 5 columns]

```

[133]: # Max and min values
# Find the index of the maximum and minimum mean_CC_EST values
most_corrupt_idx = df_trend['mean_CC_EST'].idxmax()
least_corrupt_idx = df_trend['mean_CC_EST'].idxmin()

most_corrupt_country = df_trend.loc[most_corrupt_idx, 'country']
least_corrupt_country = df_trend.loc[least_corrupt_idx, 'country']

print("The most corrupt country (on average between 2012 and 2022) is", □
    ↪most_corrupt_country)

```

```

print("The least corrupt country (on average between 2012 and 2022) is", df_trend['country'].idxmin())
print("The country that became more corrupt (rose the most) on average between 2012 and 2022 is", df_trend.loc[df_trend['avg_change'].idxmax(), 'country'])
print("The country that became less corrupt (fell the most) on average between 2012 and 2022 is", df_trend.loc[df_trend['avg_change'].idxmin(), 'country'])

```

The most corrupt country (on average between 2012 and 2022) is Somalia
The least corrupt country (on average between 2012 and 2022) is Denmark
The country that became more corrupt (rose the most) on average between 2012 and 2022 is Andorra
The country that became less corrupt (fell the most) on average between 2012 and 2022 is Andorra

```
[134]: df_correl = pd.read_csv('correlations.csv')
# Rank in order of correlation with CC_EST
df_correl = df_correl.sort_values('correlation', ascending=False)
df_correl
```

	variable	correlation
942	DT_NFL_UNWT_CD	0.907982
557	SE_LPV_PRIM	0.714894
810	SE_LPV_PRIM_MA	0.714519
727	SE_LPV_PRIM_FE	0.714091
96	SE_LPV_PRIM_LD	0.703578
...
1091	LP_LPI_CUST_XQ	-0.812534
537	NY_ADJ_NNTY_PC_KD	-0.813347
460	SI_SPR_PCAP	-0.841848
463	SI_SPR_PC40	-0.846424
77	IQ_CPA_TRAN_XQ	-0.859551

[1449 rows x 2 columns]

```
[135]: # Remove rows in df_dr that have missing values in CC_EST column
#df_dr = df[df['year'] >= 2012]
df_dr = df_dr.dropna(subset=['CC_EST'])

df_correl_0 = df_correl[abs(df_correl['correlation']) >= 0.0]
# Filter to only show correlations greater than or equal to +- 0.5
df_correl_5 = df_correl[abs(df_correl['correlation']) >= 0.5]
# Filter to only show correlations greater than or equal to +- 0.6
df_correl_6 = df_correl[abs(df_correl['correlation']) >= 0.6]
# Filter to only show correlations greater than or equal to +- 0.7
df_correl_7 = df_correl[abs(df_correl['correlation']) >= 0.7]
# Filter to only show correlations greater than or equal to +- 0.8
df_correl_8 = df_correl[abs(df_correl['correlation']) >= 0.8]
```

```

# Filter to only show correlations greater than or equal to +- 0.9
df_correl_9 = df_correl[abs(df_correl['correlation'])] >= 0.9]

print(f"The number of correlations with an absolute value greater than or equal to 0.5 is", len(df_correl_5))
print(f"The number of correlations with an absolute value greater than or equal to 0.6 is", len(df_correl_6))
print(f"The number of correlations with an absolute value greater than or equal to 0.7 is", len(df_correl_7))
print(f"The number of correlations with an absolute value greater than or equal to 0.8 is", len(df_correl_8))
print(f"The number of correlations with an absolute value greater than or equal to 0.9 is", len(df_correl_9))

```

The number of correlations with an absolute value greater than or equal to 0.5 is 253

The number of correlations with an absolute value greater than or equal to 0.6 is 128

The number of correlations with an absolute value greater than or equal to 0.7 is 54

The number of correlations with an absolute value greater than or equal to 0.8 is 7

The number of correlations with an absolute value greater than or equal to 0.9 is 1

```
[136]: # Merge together corruption values and the largest correlators
# Create variables that are a list of the variables that are highly correlated with CC_EST
vars_0 = list(df_correl_0['variable'])
vars_5 = list(df_correl_5['variable'])
vars_6 = list(df_correl_6['variable'])
vars_7 = list(df_correl_7['variable'])
vars_8 = list(df_correl_8['variable'])
vars_9 = list(df_correl_9['variable'])

# Merge the corruption values with the variables that are highly correlated with CC_EST
df_0 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_0]
df_5 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_5]
df_6 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_6]
df_7 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_7]
df_8 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_8]
df_9 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_9]
# Build random forest model to predict corruption from highly correlated variables
# Simplify notation:
x0 = df_0.iloc[:,2:].values
```

```

x5 = df_5.iloc[:,2:].values
x6 = df_6.iloc[:,2:].values
x7 = df_7.iloc[:,2:].values
x8 = df_8.iloc[:,2:].values
x9 = df_9.iloc[:,2:].values

y = df_dr['CC_EST'].values

```

```

[137]: '''
def backcast_cc_est_iteratively(df, df_filtered, target_col='CC_EST', ↴
    max_lead_years=2):
    """Backcasts a target column for each country and year,
    using a model trained on a fixed recent time period.
    """
    predictions = []

    # Training Phase
    train_start_year = 2022
    train_end_year = 2015
    training_data = df_filtered[(df_filtered['year'] <= train_start_year) & ↴
        (df_filtered['year'] >= train_end_year)]

    # Select Features (Customize!)
    features = [col for col in training_data.columns
        if col not in [target_col, 'iso2c', 'country', 'year']]

    # Handle Missing Values in 'CC_EST'
    training_data = training_data.dropna(subset=['CC_EST'])

    X_train = training_data[features]
    y_train = training_data['CC_EST']

    # Check if training data is not empty
    if not X_train.empty and not y_train.empty:
        model = XGBRegressor(objective='reg:squarederror', random_state=0, ↴
            alpha=0.1, early_stopping_rounds=None)
        model.fit(X_train, y_train)

    # Evaluation
    train_r2 = model.score(X_train, y_train)
    cv_scores = cross_val_score(model, X_train, y_train, cv=5) # Example ↴
        5-fold CV

    print(f"Train R^2: {train_r2}, CV Scores: {cv_scores}")

    # Create a DataFrame with the feature importances

```

```

        feature_importances = pd.DataFrame({'feature': features, 'importance': model.feature_importances_})

        # Sort the DataFrame by importance in descending order
        feature_importances = feature_importances.sort_values(by='importance', ascending=False)

    print(feature_importances)

    # Backcasting Phase
    for country in df_filtered['iso2c'].unique(): # Iterate over unique countries
        country_data = df_filtered[df_filtered['iso2c'] == country]
        for backcast_year in range(2011, 1959, -1):
            data_for_backcast = country_data[country_data['year'] <= backcast_year]

            # Feature Selection (Important: Align with the training set's features)
            X_backcast = data_for_backcast[features]

            # Only predict for rows where all feature values are present
            X_backcast = X_backcast.dropna(subset=features)

            if not X_backcast.empty: # Check if there's any data after handling missing values
                y_pred = model.predict(X_backcast)

                predictions.append({
                    'iso2c': country,
                    'year': backcast_year,
                    'predicted_CC_EST': y_pred[0]
                })
            else:
                print(f"Skipping year {backcast_year} for {country} due to insufficient data after handling missing values")
        else:
            print("Insufficient training data")

    return pd.DataFrame(predictions)

# Example Usage (assuming your DataFrame is already 'df_dr')
df_filtered = df[df['year'] >= 2012] # Filter for the iterative process
predictions_df = backcast_cc_est_iteratively(df.copy(), df_filtered.copy())
print(predictions_df)
'''
```

```
[137]: '\n\ndef backcast_cc_est_iteratively(df, df_filtered, target_col='CC_EST',\n    max_lead_years=2):\n    """Backcasts a target column for each country and year,\n        using a model trained on a fixed recent time period.\n        """\n\n    predictions = []\n\n        # Training Phase\n        train_start_year = 2022\n\n    train_end_year = 2015\n        training_data = df_filtered[(df_filtered['year'] <=\n        train_start_year) & (df_filtered['year'] >= train_end_year)]\n\n        # Select\n    Features (Customize!)\n        features = [col for col in training_data.columns\n        if col not in [target_col, 'iso2c', 'country', 'year']] \n\n        # Handle\n    Missing Values in 'CC_EST'\n        training_data =\n    training_data.dropna(subset=['CC_EST'])\n\n        X_train =\n    training_data[features]\n        y_train = training_data['CC_EST']\n\n        # Check\n    if training data is not empty\n        if not X_train.empty and not y_train.empty:\n            model = XGBRegressor(objective='reg:squarederror', random_state=0, alpha=0.1,\n            early_stopping_rounds=None)\n            model.fit(X_train, y_train)\n\n            #\n        Evaluation\n            train_r2 = model.score(X_train, y_train)\n            cv_scores =\n        cross_val_score(model, X_train, y_train, cv=5) # Example 5-fold CV\n\n        print(f"Train R^2: {train_r2}, CV Scores: {cv_scores}")\n\n        # Create a\n    DataFrame with the feature importances\n        feature_importances =\n    pd.DataFrame({\n        'feature': features,\n        'importance':\n        model.feature_importances_})\n\n        # Sort the DataFrame by importance in\n    descending order\n        feature_importances =\n    feature_importances.sort_values(by='importance', ascending=False)\n\n        print(feature_importances)\n\n        # Backcasting Phase\n        for\n    country in df_filtered['iso2c'].unique(): # Iterate over unique countries\n        country_data = df_filtered[df_filtered['iso2c'] == country]\n\n        for\n    backcast_year in range(2011, 1959, -1):\n            data_for_backcast =\n        country_data[country_data['year'] <= backcast_year]\n\n            #\n        Feature Selection (Important: Align with the training set's features)\n            X_backcast = data_for_backcast[features]\n\n            # Only predict for\n        rows where all feature values are present\n            X_backcast =\n\n            X_backcast.dropna(subset=features)\n\n            if not X_backcast.empty:\n\n                # Check if there's any data after handling missing values\n                y_pred = model.predict(X_backcast)\n\n                predictions.append({\n                    'iso2c': country,\n                    'year': backcast_year,\n                    'predicted_CC_EST': y_pred[0]\n                })\n\n            else:\n                print(f"Skipping year {backcast_year} for {country}\n        due to insufficient data after handling missing values")\n            else:\n                print("Insufficient training data")\n\n            return pd.DataFrame(predictions)\n\n# Example Usage (assuming your DataFrame is already 'df_dr')\n\n    df_filtered =\n    df[df['year'] >= 2012] # Filter for the iterative process\n\n    predictions_df =\n    backcast_cc_est_iteratively(df.copy(), df_filtered.copy())\n\n    print(predictions_df)\n'
```

```
[138]: def backcast_cc_est_iteratively(df, target_col='CC_EST', max_lead_years=2):\n    """Backcasts a target column for each country and year,\n        using an ARIMA model trained on a fixed recent time period."""
```

```

predictions = []
for country in df['iso2c'].unique():
    country_data = df[df['iso2c'] == country].sort_values(by='year')

    # Training Phase
    train_start_year = 2022
    train_end_year = 2015
    training_data = country_data[(country_data['year'] >= train_start_year) &
                                (country_data['year'] <= train_end_year)]
    y_train = training_data[target_col] # Assuming 'CC_EST' is the target

    # ARIMA Model
    model = ARIMA(y_train, order=(2, 1, 1)) # Adjust the order as needed
    model_fit = model.fit()

    # Backcasting Phase
    for backcast_year in range(2011, 1959, -1):
        # Use past values and the fitted model to predict for 'backcast_year'
        new_obs = model_fit.forecast()
        y_pred = new_obs[0]

        predictions.append({
            'iso2c': country,
            'year': backcast_year,
            'predicted_CC_EST': y_pred
        })

return pd.DataFrame(predictions)

```

```

[139]: '''
import pandas as pd
from prophet import Prophet

def backcast_cc_est_iteratively(df, target_col='CC_EST', max_lead_years=2):
    # ... (Same initial part as above) ...

    # Prophet Model
    country_data.rename(columns={ target_col: 'y', 'year': 'ds' }, inplace=True)
    model = Prophet()
    model.fit(country_data)

    # Backcasting Phase
    for backcast_year in range(2011, 1959, -1):
        future_df = model.make_future_dataframe(periods=1, freq='Y') # 1 year
        prediction
        future_df['ds'] = backcast_year
        forecast = model.predict(future_df)

```

```

y_pred = forecast['yhat'].iloc[0]

predictions.append({
    'iso2c': country,
    'year': backcast_year,
    'predicted_CC_EST': y_pred
})
return pd.DataFrame(predictions)

"""

```

```
[139]: """\nimport pandas as pd\nfrom prophet import Prophet\n\ndef backcast_cc_est_iteratively(df, target_col='CC_EST', max_lead_years=2):\n    # ... (Same initial part as above) ...\n    # Prophet Model\n    country_data.rename(columns={target_col: 'y', 'year': 'ds'}, inplace=True)\n    model = Prophet()\n    model.fit(country_data)\n    # Backcasting Phase\n    for backcast_year in range(2011, 1959, -1):\n        future_df =\n            model.make_future_dataframe(periods=1, freq='Y') # 1 year prediction\n        future_df['ds'] = backcast_year\n        forecast = model.predict(future_df)\n        y_pred = forecast['yhat'].iloc[0]\n        predictions.append({\n            'iso2c': country,\n            'year': backcast_year,\n            'predicted_CC_EST': y_pred\n        })\n    return pd.DataFrame(predictions)\n\n"""


```

```
[140]: def prepare_lead_data(df, target_col, lead_cols, time_col, country_col):\n    """Shifts features to create lead versions for backcasting,\n    accounting for country and year structure.\n    """\n\n    df_lead = df.copy()\n\n    for col in lead_cols:\n        df_lead[f'{col}_lead1'] = df_lead.sort_values([time_col]).\n        ↪groupby(country_col)[col].shift(-1) # Shift down by 1\n\n    df_lead.fillna(0, inplace=True)\n    return df_lead\n\n\ndef prepare_data_for_backcast(df, target_col, lead_cols, time_col, country_col):\n    """Shifts target feature to the next year for backcasting,\n    accounting for country and year structure.\n    """\n\n    df_lead = df.copy()\n\n    # Assuming target variable is 'CC_EST'\n    df_lead['CC_EST_lead1'] = df_lead.sort_values([time_col]).\n    ↪groupby(country_col)[target_col].shift(-1) # Shift target down by 1 (next\n    ↪year)
```

```

df_lead.fillna(0, inplace=True) # Handle missing values (e.g., at the end
                                ↵of histories)
return df_lead

def remove_correlated_features(df, threshold):
    """Removes highly correlated features based on a threshold."""
    corr_matrix = df.dropna().corr().abs()
    upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).
                                ↵astype(bool))
    cols_to_drop = [col for col in upper.columns if any(upper[col] > threshold)]
    df = df.drop(cols_to_drop, axis=1)
    return df

def custom_cross_val_score(model, X, y, cv):
    """Calculates cross-validation scores for a given model."""
    scores = []
    kf = KFold(n_splits=cv)

    for train_index, val_index in kf.split(X):
        x_train, x_val = X.iloc[train_index], X.iloc[val_index]
        y_train, y_val = y.iloc[train_index], y.iloc[val_index]

        model.fit(x_train, y_train, eval_set=[(x_val, y_val)], verbose=False)

        y_val_pred = model.predict(x_val)
        r2_val = round(r2_score(y_val, y_val_pred), 5)
        scores.append(r2_val)

    return scores

```

```

[141]: def build_and_evaluate_model(df, target_col_name, var_list, n_leads=2):
    """
    Builds, evaluates and returns results for an XGBoost model.
    Incorporates lead variables of the target column.
    """

    # Create a copy of the DataFrame to avoid modifying the original
    df = df.copy()

    # Create lead variables
    for i in range(1, n_leads + 1):
        lead_col = df.groupby(['country'])[[str(target_col_name)]].shift(-i)
        df[f'{target_col_name}_lead{i}'] = lead_col[str(target_col_name)]

    # Drop rows with NaN values in the target column and lead variables

```

```

df = df.dropna(subset=['country'] + [target_col_name] + [
    f'{target_col_name}_lead{i}' for i in range(1, n_leads + 1)])

# No need to add lead variables to var_list, they're already created
X = df[['country'] + var_list + [f'{target_col_name}_lead{i}' for i in
    range(1, n_leads + 1)]]

x_train_val, x_test, y_train_val, y_test = train_test_split(X,
    df[target_col_name], test_size=0.2, random_state=0)

# Exclude 'country' column when removing correlated features
x_lead = remove_correlated_features(x_train_val.drop(columns='country').
    copy(), threshold=0.8)

model = XGBRegressor(objective='reg:squarederror', random_state=0, alpha=0.
    1, early_stopping_rounds=10)

model.fit(x_lead, y_train_val, eval_set=[(x_test.drop(columns='country'),
    y_test)], verbose=False)

y_train_val_pred = model.predict(x_train_val.drop(columns='country'))
y_test_pred = model.predict(x_test.drop(columns='country'))
r2_train_val = round(r2_score(y_train_val, y_train_val_pred), 5)
r2_test = round(r2_score(y_test, y_test_pred), 5)

rmse_train_val = round(np.sqrt(mean_squared_error(y_train_val,
    y_train_val_pred)), 5)
rmse_test = round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 5)

cv_scores = custom_cross_val_score(model, x_train_val.
    drop(columns='country'), y_train_val, cv=5)
mean_cv_score = round(np.mean(cv_scores), 5)

feature_importances = pd.DataFrame({
    'Feature': x_lead.columns.tolist(),
    'Importance': model.feature_importances_
}).sort_values(by='Importance', ascending=False)

# Update the lead variables with the predicted values
df.loc[x_train_val.index, f'{target_col_name}_lead1'] = y_train_val_pred
df.loc[x_test.index, f'{target_col_name}_lead1'] = y_test_pred

if n_leads > 1:
    df[f'{target_col_name}_lead2'] = df.
        groupby(['country'])[[f'{target_col_name}_lead1']].shift(-1)

```

```

        df[f'{target_col_name}_lead2'] = df[f'{target_col_name}_lead2'].
        ↪fillna(method='ffill')

    return r2_train_val, r2_test, rmse_train_val, rmse_test, mean_cv_score, □
    ↪feature_importances, df, model

```

```
[142]: # ****
# Data Preparation (Replace with your data loading and filtering)
# *****

# Create DataFrames containing highly correlated variables
df_0 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_0].copy()
df_5 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_5].copy()
df_6 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_6].copy()
df_7 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_7].copy()
df_8 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_8].copy()
df_9 = df_dr[['iso2c', 'country', 'year', 'CC_EST'] + vars_9].copy()

y = df_dr['CC_EST'].values

# Datasets creation
datasets = [df_5, df_6, df_7, df_8]
vars_list = [vars_5, vars_6, vars_7, vars_8]

# ****
# Main Execution
# *****

r2_train_vals = []
r2_tests = []
rmse_train_vals = []
rmse_tests = []
mean_cv_scores = []
feature_importances_dfs = {}

for i, (df, var_list) in enumerate(zip(datasets, vars_list), start=5):
    print(f"Iteration {i}")
    results = build_and_evaluate_model(df, 'CC_EST', var_list)

    r2_train_vals.append(results[0])
    r2_tests.append(results[1])
    rmse_train_vals.append(results[2])
    rmse_tests.append(results[3])
    mean_cv_scores.append(results[4])
    feature_importances_dfs[f'feature_importances_{i}'] = results[5]
    df_new = results[6]
```

```

    print(f"Model with correlation >= 0.{i}:")
    print(f"Training+Validation R^2: {r2_train_vals[-1]}, RMSE: {rmse_train_vals[-1]}")
    print(f"Testing R^2: {r2_tests[-1]}, RMSE: {rmse_tests[-1]}")
    print(f"Mean cross-validation score: {mean_cv_scores[-1]}\n")
    print(feature_importances_dfs[f'feature_importances_{i}'])
    print("\n")

```

Iteration 5

```

Model with correlation >= 0.5:
Training+Validation R^2: 0.99788, RMSE: 0.92129
Testing R^2: 0.983, RMSE: 2.59014
Mean cross-validation score: 0.98171

```

	Feature	Importance
253	CC_EST_lead1	0.746632
254	CC_EST_lead2	0.021110
62	SH_DYN_MORT_MA	0.009215
133	SP_DYN_T065_FE_ZS	0.008897
117	SP_POP_7074_FE_5Y	0.006084
..
46	SP_DYN_IMRT_IN	0.000000
131	SE_TER CUAT_BA_FE_ZS	0.000000
49	SH_STA_AIRP_MA_P5	0.000000
51	SP_DYN_IMRT_FE_IN	0.000000
194	SL_EMP_WORK_ZS	0.000000

[255 rows x 2 columns]

Iteration 6

```

Model with correlation >= 0.6:
Training+Validation R^2: 0.99708, RMSE: 1.08032
Testing R^2: 0.98416, RMSE: 2.50043
Mean cross-validation score: 0.98098

```

	Feature	Importance
128	CC_EST_lead1	0.833415
129	CC_EST_lead2	0.015279
52	SP_DYN_LE00_FE_IN	0.005336
46	SP_DYN_T065_MA_ZS	0.005084
82	NY_GDP_PCAP_PP_KD	0.003807
..
111	FX_OWN_TOTL_40_ZS	0.000000
85	NY_GDP_PCAP_KD	0.000000
103	FX_OWN_TOTL_OL_ZS	0.000000
97	HD_HCI_OVRL_MA	0.000000
0	DTNFL_UNWT_CD	0.000000

[130 rows x 2 columns]

Iteration 7
Model with correlation >= 0.7:
Training+Validation R^2: 0.99669, RMSE: 1.15012
Testing R^2: 0.98547, RMSE: 2.3953
Mean cross-validation score: 0.98136

	Feature	Importance
54	CC_EST_lead1	0.915265
55	CC_EST_lead2	0.013398
40	NY_ADJ_NNTY_PC_CD	0.003573
48	NE_CON_PRVT_PC_KD	0.003427
20	NY_GNP_PCAP_CD	0.003018
53	IQ_CPA_TRAN_XQ	0.002818
36	FX_OWN_TOTL_FE_ZS	0.002789
8	NY_GDP_PCAP_PP_KD	0.002681
12	IT_MLT_MAIN_P2	0.002570
31	SH_XPD_GHED_PP_CD	0.002403
47	NY_GNP_PCAP_KD	0.002153
50	NY_ADJ_NNTY_PC_KD	0.002138
9	FX_OWN_TOTL_SO_ZS	0.002018
37	FX_OWN_TOTL_40_ZS	0.001891
28	PA_NUS_PPPC_RF	0.001867
45	LP_LPI_OVRL_XQ	0.001850
22	SH_XPD_CHEX_PC_CD	0.001834
32	IQ_CPA_PUBS_XQ	0.001793
38	IQ_CPA_PROP_XQ	0.001730
13	FS_AST_DOMO_GD_ZS	0.001714
6	SE_LP_V_PRIM_LD_MA	0.001657
39	FX_OWN_TOTL_PL_ZS	0.001619
19	SH_XPD_GHED_PC_CD	0.001596
46	LP_LPI_INF_R_XQ	0.001545
24	HD_HCI_OVRL_UB	0.001481
11	NY_GDP_PCAP_KD	0.001394
44	NY_GNP_PCAP_PP_KD	0.001354
26	HD_HCI_OVRL	0.001319
33	SH_XPD_CHEX_PP_CD	0.001249
51	SI_SPR_PCAP	0.001244
4	SE_LP_V_PRIM_LD	0.001240
27	HD_HCI_OVRL_LB	0.001238
49	LP_LPI_CUST_XQ	0.001147
42	NV_SRV_EMPL_KD	0.001112
21	HD_HCI_OVRL_UB_MA	0.001030
43	LP_LPI_LOGS_XQ	0.000969
15	HD_HCI_OVRL_UB_FE	0.000934

```

7      IC_BUS_DFRN_XQ    0.000836
41     SP_POP_SCIE_RD_P6  0.000787
52      SI_SPR_PC40    0.000776
10     LP_LPI_TIME_XQ    0.000776
25     HD_HCI_OVRL_LB_MA  0.000566
35     LP_LPI_TRAC_XQ    0.000479
16     HD_HCI_OVRL_LB_FE  0.000467
5      SI_POV_MDIM_17    0.000465
30     FX_OWN_TOTL_YG_ZS  0.000394
3      SE_LPV_PRIM_FE    0.000367
1      SE_LPV_PRIM        0.000349
23     HD_HCI_OVRL_MA    0.000342
14     FX_OWN_TOTL_60_ZS  0.000229
2      SE_LPV_PRIM_MA    0.000107
34     FX_OWN_TOTL_ZS    0.000000
29     FX_OWN_TOTL_OL_ZS  0.000000
18     FX_OWN_TOTL_MA_ZS  0.000000
17     HD_HCI_OVRL_FE    0.000000
0      DT_NFL_UNWT_CD    0.000000

```

Iteration 8

Model with correlation >= 0.8:

Training+Validation R^2: 0.99425, RMSE: 1.51691

Testing R^2: 0.98527, RMSE: 2.41136

Mean cross-validation score: 0.98119

	Feature	Importance
7	CC_EST_lead1	0.975617
8	CC_EST_lead2	0.011002
6	IQ_CPA_TRAN_XQ	0.004073
1	NE_CON_PRVT_PC_KD	0.002831
3	NY_ADJ_NNTY_PC_KD	0.002073
5	SI_SPR_PC40	0.001758
2	LP_LPI_CUST_XQ	0.001504
4	SI_SPR_PCAP	0.001142
0	DT_NFL_UNWT_CD	0.000000

```
[143]: pd.set_option('display.max_rows', 100)
```

```
# Now print the DataFrames
print('\n Feature Importances for Correlation >= 0.5')
print(feature_importances_dfs['feature_importances_5'])
print('\n Feature Importances for Correlation >= 0.6')
print(feature_importances_dfs['feature_importances_6'])
```

```

print('\n Feature Importances for Correlation >= 0.7')
print(feature_importances_dfs['feature_importances_7'])

```

Feature Importances for Correlation >= 0.5

	Feature	Importance
253	CC_EST_lead1	0.746632
254	CC_EST_lead2	0.021110
62	SH_DYN_MORT_MA	0.009215
133	SP_DYN_T065_FE_ZS	0.008897
117	SP_POP_7074_FE_5Y	0.006084
..
46	SP_DYN_IMRT_IN	0.000000
131	SE_TER_CUAT_BA_FE_ZS	0.000000
49	SH_STA_AIRP_MA_P5	0.000000
51	SP_DYN_IMRT_FE_IN	0.000000
194	SL_EMP_WORK_ZS	0.000000

[255 rows x 2 columns]

Feature Importances for Correlation >= 0.6

	Feature	Importance
128	CC_EST_lead1	0.833415
129	CC_EST_lead2	0.015279
52	SP_DYN_LE00_FE_IN	0.005336
46	SP_DYN_T065_MA_ZS	0.005084
82	NY_GDP_PCAP_PP_KD	0.003807
..
111	FX_OWN_TOTL_40_ZS	0.000000
85	NY_GDP_PCAP_KD	0.000000
103	FX_OWN_TOTL_OL_ZS	0.000000
97	HD_HCI_OVRL_MA	0.000000
0	DT_NFL_UNWT_CD	0.000000

[130 rows x 2 columns]

Feature Importances for Correlation >= 0.7

	Feature	Importance
54	CC_EST_lead1	0.915265
55	CC_EST_lead2	0.013398
40	NY_ADJ_NNTY_PC_CD	0.003573
48	NE_CON_PRVT_PC_KD	0.003427
20	NY_GNP_PCAP_CD	0.003018
53	IQ_CPA_TRAN_XQ	0.002818
36	FX_OWN_TOTL_FE_ZS	0.002789
8	NY_GDP_PCAP_PP_KD	0.002681
12	IT_MLT_MAIN_P2	0.002570
31	SH_XPD_GHED_PP_CD	0.002403

47	NY_GNP_PCAP_KD	0.002153
50	NY_ADJ_NNTY_PC_KD	0.002138
9	FX_OWN_TOTL_SO_ZS	0.002018
37	FX_OWN_TOTL_40_ZS	0.001891
28	PA_NUS_PPPC_RF	0.001867
45	LP_LPI_OVRL_XQ	0.001850
22	SH_XPD_CHEX_PC_CD	0.001834
32	IQ_CPA_PUBS_XQ	0.001793
38	IQ_CPA_PROP_XQ	0.001730
13	FS_AST_DOMO_GD_ZS	0.001714
6	SE_LP_V_PRIM_LD_MA	0.001657
39	FX_OWN_TOTL_PL_ZS	0.001619
19	SH_XPD_GHED_PC_CD	0.001596
46	LP_LPI_INFR_XQ	0.001545
24	HD_HCI_OVRL_UB	0.001481
11	NY_GDP_PCAP_KD	0.001394
44	NY_GNP_PCAP_PP_KD	0.001354
26	HD_HCI_OVRL	0.001319
33	SH_XPD_CHEX_PP_CD	0.001249
51	SI_SPR_PCAP	0.001244
4	SE_LP_V_PRIM_LD	0.001240
27	HD_HCI_OVRL_LB	0.001238
49	LP_LPI_CUST_XQ	0.001147
42	NV_SRV_EMPL_KD	0.001112
21	HD_HCI_OVRL_UB_MA	0.001030
43	LP_LPI_LOGS_XQ	0.000969
15	HD_HCI_OVRL_UB_FE	0.000934
7	IC_BUS_DFRN_XQ	0.000836
41	SP_POP_SCIE_RD_P6	0.000787
52	SI_SPR_PC40	0.000776
10	LP_LPI_TIME_XQ	0.000776
25	HD_HCI_OVRL_LB_MA	0.000566
35	LP_LPI_TRAC_XQ	0.000479
16	HD_HCI_OVRL_LB_FE	0.000467
5	SI_POV_MDIM_17	0.000465
30	FX_OWN_TOTL_YG_ZS	0.000394
3	SE_LP_V_PRIM_FE	0.000367
1	SE_LP_V_PRIM	0.000349
23	HD_HCI_OVRL_MA	0.000342
14	FX_OWN_TOTL_60_ZS	0.000229
2	SE_LP_V_PRIM_MA	0.000107
34	FX_OWN_TOTL_ZS	0.000000
29	FX_OWN_TOTL_DL_ZS	0.000000
18	FX_OWN_TOTL_MA_ZS	0.000000
17	HD_HCI_OVRL_FE	0.000000
0	DTNFL_UNWT_CD	0.000000

```
[144]: r2_train_vals = []
r2_tests = []
rmse_train_vals = []
rmse_tests = []
mean_cv_scores = []
feature_importances_dfs = {}

# Remove duplicate columns in df_0
df_0 = df_0.loc[:,~df_0.columns.duplicated()]

results = build_and_evaluate_model(df_0, 'CC_EST', vars_0)

r2_train_vals.append(results[0])
r2_tests.append(results[1])
rmse_train_vals.append(results[2])
rmse_tests.append(results[3])
mean_cv_scores.append(results[4])
feature_importances_dfs['feature_importances'] = results[5]
df_new = results[6]

print(f"Full Model:")
print(f"Training+Validation R^2: {r2_train_vals[-1]}, RMSE:{rmse_train_vals[-1]}")
print(f"Testing R^2: {r2_tests[-1]}, RMSE: {rmse_tests[-1]}")
print(f"Mean cross-validation score: {mean_cv_scores[-1]}\n")
print(feature_importances_dfs['feature_importances'])
print("\n")
```

Full Model:
 Training+Validation R²: 0.99926, RMSE: 0.54537
 Testing R²: 0.98441, RMSE: 2.4806
 Mean cross-validation score: 0.98112

	Feature	Importance
1449	CC_EST_lead1	0.592026
1450	CC_EST_lead2	0.027190
318	SE_PRM_UNER_MA	0.017460
484	GC_REV_XGRT_CN	0.012413
244	IC_ELC_TIME	0.011027
...
786	per_allsp_ben_q1_tot	0.000000
787	SL_TLF_ACTI_1524_MA_NE_ZS	0.000000
788	NY_ADJ_NNAT_CD	0.000000
789	ST_INT_XPND_MP_ZS	0.000000
0	DT_NFL_UNWT_CD	0.000000

[1451 rows x 2 columns]

```
[145]: """
# Replace 'vars_full' with your actual list of variables, excluding 'iso2c', ↵
# 'country', and 'CC_EST'
vars_full = list(vars_0)

print("Running model on full dataset")
for i in range(0, 9):
    # Calculate the correlation of each variable with 'CC_EST'
    correl = df_0[vars_full].corrwith(df['CC_EST']).abs()

    # Get the variables with a correlation greater than or equal to the ↵
    # threshold
    vars_correl = correl[correl >= 0.1 * i].index.tolist()

    results = build_and_evaluate_model(df_0, 'CC_EST', vars_correl, n_leads=2)

    print(f"\n Model with correlation >= 0.{i}:")
    print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
    print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
    print(f"Mean cross-validation score: {results[4]}\n")
    print(results[5]) # Feature importances
    print('\n')
    print(results[6][['country', 'year', 'CC_EST', 'CC_EST_lead1', ↵
        'CC_EST_lead2']])
"""


```

```
[145]: '\n# Replace \'vars_full\' with your actual list of variables, excluding
\'iso2c\', \'country\', and \'CC_EST\'\nvars_full =
list(vars_0)\n\nprint("Running model on full dataset")\nfor i in range(0,
9):\n    # Calculate the correlation of each variable with \'CC_EST\'\n    correl = df_0[vars_full].corrwith(df['CC_EST']).abs()\n    # Get the
variables with a correlation greater than or equal to the threshold\n    vars_correl = correl[correl >= 0.1 * i].index.tolist()\n    results =
build_and_evaluate_model(df_0, 'CC_EST', vars_correl, n_leads=2)\n\n    print(f"\n Model with correlation >= 0.{i}:")\n    print(f"Training+Validation
R^2: {results[0]}, RMSE: {results[2]}")\n    print(f"Testing R^2: {results[1]},
RMSE: {results[3]}")\n    print(f"Mean cross-validation score:
{results[4]}\n")\n    print(results[5]) # Feature importances\n
print('\n')\n    print(results[6][['country', 'year', 'CC_EST',
'CC_EST_lead1', 'CC_EST_lead2']])\n'
```

```
[146]: """
correl = df_0[vars_full].corrwith(df['CC_EST']).abs()
```

```

results = build_and_evaluate_model(df_0, 'CC_EST', vars_correl, n_leads=2)

print(f"\n Full Model")
print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
print(f"Mean cross-validation score: {results[4]}\n")
print(results[5]) # Feature importances
print('\n')
print(results[6][['country', 'year', 'CC_EST', 'CC_EST_lead1', 'CC_EST_lead2']])
"""

```

```

[146]: \ncorrel = df_0[vars_full].corrwith(df[['CC_EST']]).abs()\n\nresults =
build_and_evaluate_model(df_0, 'CC_EST', vars_correl, n_leads=2)\n\nprint(f"\n Full Model")\nprint(f"Training+Validation R^2: {results[0]}, RMSE:
{results[2]}")\nprint(f"Testing R^2: {results[1]}, RMSE:
{results[3]}")\nprint(f"Mean cross-validation score:
{results[4]}\n")\nprint(results[5]) # Feature
importances\nprint('')\nprint(results[6][['country', 'year', 'CC_EST',
'CC_EST_lead1', 'CC_EST_lead2']])\n'

```

```

[147]: df = pd.read_csv('WDI_data.csv')
df = df.drop(columns=['CPI_EST'])
# Make corruption more intuitive with higher values indicating more corruption
df['CC_EST'] = (-1)*df['CC_EST']
df['CC_EST'] = (df['CC_EST'] + 2.5) * 20
# Drop years 1970-2011 if column 'year' is less than 2012
df_dr = df[df['year'] >= 2012]
# Make years integers
df['year'] = df['year'].astype(int)

```

```

[148]: features = df.select_dtypes(include=[np.number]).columns.tolist()
features.remove('year')
features.remove('CC_EST')
vars_0.remove('year')

r2_train_val, r2_test, rmse_train_val, rmse_test, mean_cv_score, ▾
↪feature_importances, df, model = build_and_evaluate_model(df, 'CC_EST', ▾
↪vars_0, n_leads=2)

print(f"\n Full Model")
print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
print(f"Mean cross-validation score: {results[4]}\n")
print(results[5]) # Feature importances
print('')
print(results[6][['country', 'year', 'CC_EST', 'CC_EST_lead1', 'CC_EST_lead2']])

```

```

Full Model
Training+Validation R^2: 0.99926, RMSE: 0.54537
Testing R^2: 0.98441, RMSE: 2.4806
Mean cross-validation score: 0.98112

```

	Feature	Importance
1449	CC_EST_lead1	0.592026
1450	CC_EST_lead2	0.027190
318	SE_PRM_UNER_MA	0.017460
484	GC_REV_XGRT_CN	0.012413
244	IC_ELC_TIME	0.011027
...
786	per_allsp_ben_q1_tot	0.000000
787	SL_TLF_ACTI_1524_MA_NE_ZS	0.000000
788	NY_ADJ_NNAT_CD	0.000000
789	ST_INT_XPND_MP_ZS	0.000000
0	DTNFL_UNWT_CD	0.000000

[1451 rows x 2 columns]

	country	year	CC_EST	CC_EST_lead1	CC_EST_lead2
52	Andorra	2012	24.789383	25.087683	24.872046
53	Andorra	2013	24.929030	24.872046	25.697569
54	Andorra	2014	25.585828	25.697569	26.203590
55	Andorra	2015	26.963785	26.203590	25.972879
56	Andorra	2016	26.808882	25.972879	26.224508
...
13727	Zimbabwe	2016	75.768814	75.931007	75.689507
13728	Zimbabwe	2017	75.969706	75.689507	75.437538
13729	Zimbabwe	2018	74.920013	75.437538	75.374451
13730	Zimbabwe	2019	75.423806	75.374451	75.653221
13731	Zimbabwe	2020	75.759847	75.653221	75.653221

[1839 rows x 5 columns]

```

[149]: df = pd.read_csv('WDI_data.csv')
df = df.drop(columns=['CPI_EST'])
# Make corruption more intuitive with higher values indicating more corruption
df['CC_EST'] = (-1)*df['CC_EST']
df['CC_EST'] = (df['CC_EST'] + 2.5) * 20
# Drop years 1970-2011 if column 'year' is less than 2012
df_dr = df[df['year'] >= 2012]
# Make years integers
df['year'] = df['year'].astype(int)

```

```
[150]: # Get the list of unique countries
countries = df['iso2c'].unique()

# Create a new dataframe to store the results
df_new = pd.DataFrame()

# Get the feature names the model was trained on
model.get_booster().feature_names = features
model_features = model.get_booster().feature_names

# Iterate over each country
for country in countries:
    # Get the data for the current country and create a copy of it
    df_country = df[df['iso2c'] == country].copy()

    # Sort the data in descending order of the year
    df_country = df_country.sort_values('year', ascending=False)

    # Create lead variables
    # Create lead variables if they don't already exist
    if 'CC_EST_lead1' not in df_country.columns:
        df_country['CC_EST_lead1'] = df_country['CC_EST'].shift(+1)
    if 'CC_EST_lead2' not in df_country.columns:
        df_country['CC_EST_lead2'] = df_country['CC_EST'].shift(+2)

    # Get the latest year in the data
    latest_year = df_country['year'].max()

    # Skip the current country if all its 'year' values are NaN
    if np.isnan(latest_year):
        continue

    latest_year = int(latest_year)

    # Iterate from the latest year to 1960
    for year in range(latest_year, 1959, -1):
        # Check if CC_EST for the current year is NaN
        if df_country.loc[df_country['year'] == year, 'CC_EST'].isna().any():
            # Select only numeric columns for X, excluding the specified columns
            X = df_country.loc[df_country['year'] == year, model_features]

            X = X.ffill()

            y_pred = model.predict(X)

            # Update the data with the predicted value
            df_country.loc[df_country['year'] == year, 'CC_EST'] = y_pred
```

```

# Update the lead variables
if year - 1 >= 1960:
    df_country.loc[df_country['year'] == year - 1, 'CC_EST_lead1'] ↴
↪= y_pred
    if year - 2 >= 1960:
        df_country.loc[df_country['year'] == year - 2, 'CC_EST_lead2'] ↴
↪= y_pred

# Append the data for the current country to the new dataframe
df_new = pd.concat([df_new, df_country])

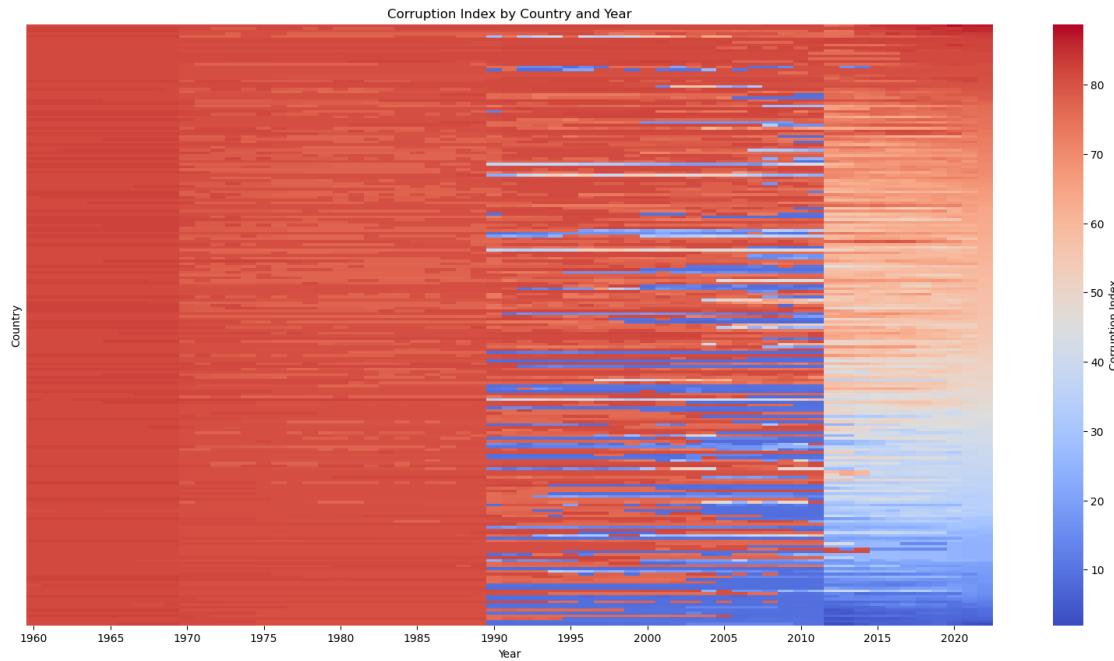
```

[153]: # Give a relative rank of CC_EST for each country in each year
`df_new['CC_EST_rank'] = df_new.groupby('year')['CC_EST'].rank(pct=True)`

Give a rank of CC_EST for each country in each year as integers
`df_new['CC_EST_rank_int'] = df_new.groupby('year')['CC_EST'].rank(method='dense', ascending=True)`

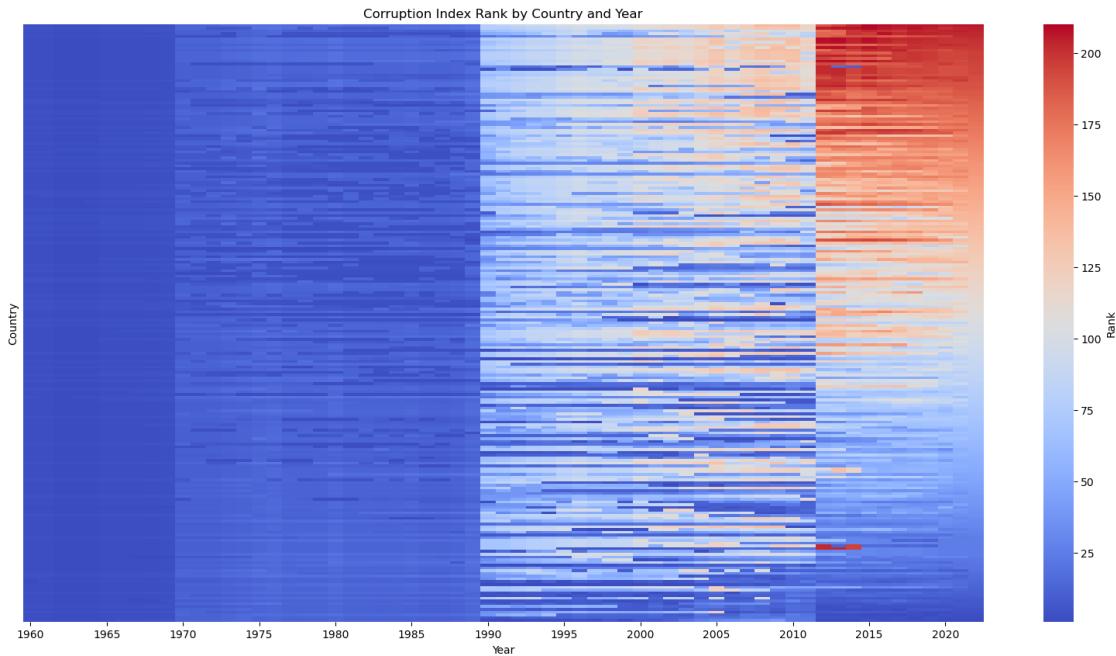
[154]: # Graphically illustrate the generale moves of country and corruption levels over time
`# Create a pivot table of the data for the heatmap`
`df_heatmap = df_new.pivot(index='country', columns='year', values='CC_EST')`
`df_heatmap = df_heatmap.sort_values(by=2022, ascending=False)`

Create a heatmap of the data, dont show country labels, and 1970 to 2022
`plt.figure(figsize=(20, 10))`
`sns.heatmap(df_heatmap, cmap='coolwarm', cbar_kws={'label': 'Corruption Index'})`
`plt.xlim(0, 63)`
`plt.title('Corruption Index by Country and Year')`
`plt.xlabel('Year')`
`plt.ylabel('Country')`
`plt.show()`



```
[155]: # Plot changes in country ranks over time
# Create a pivot table of the data for the heatmap
df_heatmap = df_new.pivot(index='country', columns='year',
                           values='CC_EST_rank_int')
df_heatmap = df_heatmap.sort_values(by=2022, ascending=False)

# Create a heatmap of the data, don't show country labels, and 1970 to 2022,
# ranking from lowest to highest in 2022
plt.figure(figsize=(20, 10))
sns.heatmap(df_heatmap, cmap='coolwarm', cbar_kws={'label': 'Rank'},
            xticklabels=5, yticklabels=False)
plt.xlim(0, 63)
plt.title('Corruption Index Rank by Country and Year')
plt.xlabel('Year')
plt.ylabel('Country')
plt.show()
```



```
[156]: # print mean and median values of CC_EST for the whole dataset
mean_cc_est = df['CC_EST'].mean()
median_cc_est = df['CC_EST'].median()

print(f"The mean CC_EST value across all years is {mean_cc_est:.2f}")
print(f"The median CC_EST value across all years is {median_cc_est:.2f}")
```

The mean CC_EST value across all years is 50.56
The median CC_EST value across all years is 54.82

```
[157]: # Plot a histogram of the forecasted corruption levels in 1960
plt.figure(figsize=(20, 10))

# Use seaborn's histplot function to plot the histogram
# Set the color, edgecolor and linewidth for the bars
sns.histplot(df_new[df_new['year'] == 1960]['CC_EST'], bins=100, kde=True, color="#00688B",
             edgecolor="#00688B", linewidth=1)

# Calculate the mean and median of the forecasted corruption levels
mean_est_1960 = df_new[df_new['year'] == 1960]['CC_EST'].mean()
median_est_1960 = df_new[df_new['year'] == 1960]['CC_EST'].median()

# Add a vertical line at the mean
plt.axvline(mean_est_1960, color="#00688B", linestyle='--', linewidth=2)
```

```

# Add a vertical line at the median
plt.axvline(median_est_1960, color='#00688B', linestyle='--', linewidth=2)

# Set the labels and title with larger fonts
plt.xlabel('Forecasted Corruption Index', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Forecasted Corruption Index in 1960', fontsize=16)

# Set the grid style
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Add a legend for mean and median
plt.legend([f'Mean: {mean_est_1960:.2f}', f'Median: {median_est_1960:.2f}'],  

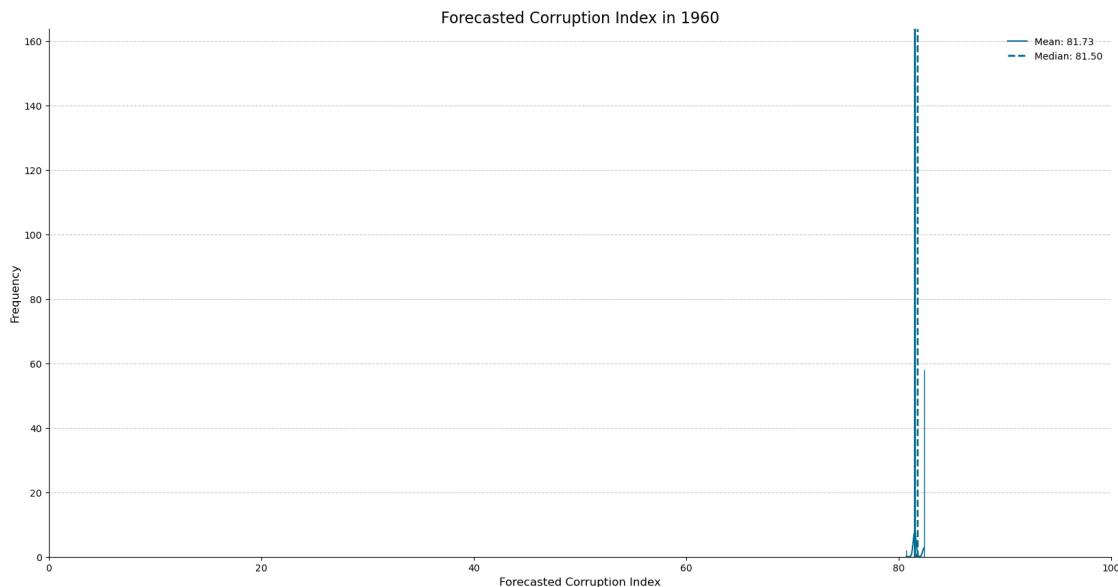
    frameon=False)

# Remove top and right spines
sns.despine()

# Set the x-axis limits to 0 and 100
plt.xlim(0, 100)

# Show the plot
plt.show()

```



```
[158]: # Convert the list of tuples to a dictionary
importances_dict = results[5].set_index('Feature')['Importance'].to_dict()
```

```

# Extract feature names
features = list(importances_dict.keys())

# Since all values are already floats, there's no need to calculate the mean
importance_scores = list(importances_dict.values())

# Create a 2D line plot
fig, ax = plt.subplots(figsize=(20, 10))

# Create a sequence of numbers for the x-axis
x = np.arange(len(features))

# Create the 2D line plot
ax.plot(x, importance_scores, color='#00688B')

# Set the ticks on the x-axis to be the new labels and remove the lines (ticks)
ax.set_xticks(x, minor=False)
ax.set_xticklabels([])
ax.tick_params(axis='x', which='both', length=0)

# Set labels
ax.set_xlabel('Features')
ax.set_ylabel('Importance Scores')

sns.despine()

# Show the plot
plt.show()

```



4 Ignore

```
[159]: df = pd.read_csv('WDI_data.csv')
df = df.drop(columns=['CPI_EST'])
# Make corruption more intuitive with higher values indicating more corruption
df['CC_EST'] = (-1)*df['CC_EST']
df['CC_EST'] = (df['CC_EST'] + 2.5) * 20
# Drop years 1960-2011 if column 'year' is less than 2012
df_dr = df[df['year'] >= 2012]
```

```
[160]: print("Running model on variables with highest correlation")
for i in range(5, 9):
    # Select only numeric columns
    df_numeric = df.select_dtypes(include=[np.number])
    df_numeric = df_numeric.drop(columns=['year'])

    # Calculate the correlation of each column with 'CC_EST'
    correl = df_numeric.corrwith(df_numeric['CC_EST']).abs()

    # Get the variables with a correlation greater than or equal to the
    # threshold
    vars_correl = correl[correl >= 0.1 * i].index.tolist()
    vars_correl.remove('CC_EST')

    # Get the count of non-null values in each of the selected columns
    non_null_counts = df[vars_correl].count()

    # Sort the columns in descending order of non-null count
    sorted_columns = non_null_counts.sort_values(ascending=False)

    # Get the top 200 columns
    top_columns = sorted_columns.index[:200]

    # Remove correlated features from the top columns
    df_top_columns = df[top_columns]
    df_top_columns = remove_correlated_features(df_top_columns, threshold=0.8)

    # Update the top_columns list to include only the columns present in the
    # DataFrame
    top_columns = df_top_columns.columns.tolist()

    results = build_and_evaluate_model(df, 'CC_EST', top_columns, n_leads=2)

    print(f"\n Model with correlation >= 0.{i}:")
    print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
    print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
    print(f"Mean cross-validation score: {results[4]}\n")
```

```

    print(results[5]) # Feature importances
    print('\n')
    print(results[6][['country', 'year', 'CC_EST', 'CC_EST_lead1', ▾
    ↵'CC_EST_lead2']])

```

Running model on variables with highest correlation

Model with correlation >= 0.5:
Training+Validation R^2: 0.99724, RMSE: 1.05141
Testing R^2: 0.98363, RMSE: 2.54194
Mean cross-validation score: 0.98093

	Feature	Importance
200	CC_EST_lead1	0.764917
22	SP_POP_65UP_FE_ZS	0.019109
201	CC_EST_lead2	0.010726
49	SH_DYN_MORT_MA	0.009604
68	SH_DYN_2024	0.008769
..
89	SL_EMP_SELF_FE_ZS	0.000000
85	SL_EMP_WORK_MA_ZS	0.000000
77	SL_EMP_VULN_ZS	0.000000
172	DC_ODA_TOTL_GN_ZS	0.000000
78	SL_EMP_SELF_MA_ZS	0.000000

[202 rows x 2 columns]

	country	year	CC_EST	CC_EST_lead1	CC_EST_lead2
52	Andorra	2012	24.789383	25.437155	25.838907
53	Andorra	2013	24.929030	25.838907	25.815485
54	Andorra	2014	25.585828	25.815485	26.518415
55	Andorra	2015	26.963785	26.518415	25.815485
56	Andorra	2016	26.808882	25.815485	26.365913
..
13727	Zimbabwe	2016	75.768814	75.965584	76.028229
13728	Zimbabwe	2017	75.969706	76.028229	75.654556
13729	Zimbabwe	2018	74.920013	75.654556	75.259323
13730	Zimbabwe	2019	75.423806	75.259323	75.202583
13731	Zimbabwe	2020	75.759847	75.202583	75.202583

[1839 rows x 5 columns]

Model with correlation >= 0.6:
Training+Validation R^2: 0.99667, RMSE: 1.15464
Testing R^2: 0.98397, RMSE: 2.51512
Mean cross-validation score: 0.98103

	Feature	Importance
140	CC_EST_lead1	0.843027
141	CC_EST_lead2	0.012561
19	SP_POP_DPND_YG	0.005727
36	NY_GDP_PCAP_PP_CD	0.003892
3	SP_POP_1014_MA_5Y	0.003728
..
12	SP_POP_0014_MA_ZS	0.000000
124	DT_ODA_OATL_KD	0.000000
123	DT_ODA_OATL_CD	0.000000
114	HD_HCI_OVRL_LB_FE	0.000000
108	FX_OWN_TOTL_FE_ZS	0.000000

[142 rows x 2 columns]

	country	year	CC_EST	CC_EST_lead1	CC_EST_lead2
52	Andorra	2012	24.789383	26.062393	26.157280
53	Andorra	2013	24.929030	26.157280	26.157280
54	Andorra	2014	25.585828	26.157280	26.157280
55	Andorra	2015	26.963785	26.157280	25.883503
56	Andorra	2016	26.808882	25.883503	26.157280
..
13727	Zimbabwe	2016	75.768814	75.609756	75.609756
13728	Zimbabwe	2017	75.969706	75.609756	75.270378
13729	Zimbabwe	2018	74.920013	75.270378	75.354019
13730	Zimbabwe	2019	75.423806	75.354019	75.583359
13731	Zimbabwe	2020	75.759847	75.583359	75.583359

[1839 rows x 5 columns]

```
Model with correlation >= 0.7:
Training+Validation R^2: 0.99681, RMSE: 1.12965
Testing R^2: 0.98481, RMSE: 2.44897
Mean cross-validation score: 0.98072
```

	Feature	Importance
60	CC_EST_lead1	0.910143
61	CC_EST_lead2	0.013431
4	NY_GDP_PCAP_PP_KD	0.005459
54	SE_LPV_PRIM_MA	0.003643
20	IQ_CPA_TRAN_XQ	0.003316
34	FX_OWN_TOTL_FE_ZS	0.003275
31	HD_HCI_OVRL	0.002915
15	SH_XPD_GHED_PP_CD	0.002555
14	SH_XPD_GHED_PC_CD	0.002452
19	IQ_CPA_PUBS_XQ	0.002321
3	PA_NUS_PPPC_RF	0.002202

11	IT_NET_BBND_P2	0.002105
22	LP_LPI_TIME_XQ	0.002005
46	HD_HCI_OVRL_UB_MA	0.001903
13	SH_XPD_CHEX_PP_CD	0.001862
2	NE_CON_PRVT_PC_KD	0.001812
26	LP_LPI_OVRL_XQ	0.001779
6	SL_GDP_PCAP_EM_KD	0.001746
28	FS_AST_DOMO_GD_ZS	0.001706
39	FX_OWN_TOTL_40_ZS	0.001579
51	SE_LPV_PRIM_LD	0.001547
1	NY_ADJ_NNTY_PC_CD	0.001505
56	SE_LPV_PRIM_LD_FE	0.001478
58	SI_POV_MDIM_17_XQ	0.001447
16	SH_MED_NUMW_P3	0.001445
8	NY_ADJ_NNTY_PC_KD	0.001420
7	NY_GNP_PCAP_KD	0.001394
29	HD_HCI_OVRL_LB	0.001350
10	NY_GNP_PCAP_PP_KD	0.001338
5	NY_GNP_PCAP_PP_CD	0.001301
17	SP_POP_SCIE_RD_P6	0.001232
53	SI_SPR_PCAP	0.001202
9	NV_SRV_EMPL_KD	0.001136
47	SI_POV_MDIM_17	0.001132
27	IC_BUS_DFRN_XQ	0.001104
32	FX_OWN_TOTL_OL_ZS	0.001078
24	LP_LPI_INFR_XQ	0.000844
0	NY_GNP_PCAP_CD	0.000807
50	SE_LPV_PRIM	0.000800
41	HD_HCI_OVRL_FE	0.000787
21	LP_LPI_TRAC_XQ	0.000786
18	IQ_CPA_PROP_XQ	0.000777
23	LP_LPI_LOGS_XQ	0.000733
25	LP_LPI_CUST_XQ	0.000713
36	FX_OWN_TOTL_MA_ZS	0.000667
30	HD_HCI_OVRL_UB	0.000609
33	FX_OWN_TOTL_YG_ZS	0.000604
12	SH_XPD_CHEX_PC_CD	0.000592
52	SI_SPR_PC40	0.000564
55	SE_LPV_PRIM_LD_MA	0.000534
38	FX_OWN_TOTL_PL_ZS	0.000449
37	FX_OWN_TOTL_SO_ZS	0.000418
49	DT_ODA_OATL_KD	0.000000
44	HD_HCI_OVRL_LB_MA	0.000000
48	DT_ODA_OATL_CD	0.000000
35	FX_OWN_TOTL_ZS	0.000000
40	FX_OWN_TOTL_60_ZS	0.000000
57	SE_LPV_PRIM_FE	0.000000
42	HD_HCI_OVRL_LB_FE	0.000000

```

59      DT_NFL_UNWT_CD    0.000000
45      HD_HCI_OVRL_MA    0.000000
43  HD_HCI_OVRL_UB_FE    0.000000

```

	country	year	CC_EST	CC_EST_lead1	CC_EST_lead2
52	Andorra	2012	24.789383	26.087563	29.862108
53	Andorra	2013	24.929030	29.862108	26.625412
54	Andorra	2014	25.585828	26.625412	26.295208
55	Andorra	2015	26.963785	26.295208	26.625412
56	Andorra	2016	26.808882	26.625412	26.625412
...
13727	Zimbabwe	2016	75.768814	75.803139	75.851654
13728	Zimbabwe	2017	75.969706	75.851654	75.047043
13729	Zimbabwe	2018	74.920013	75.047043	75.470879
13730	Zimbabwe	2019	75.423806	75.470879	75.409477
13731	Zimbabwe	2020	75.759847	75.409477	75.409477

[1839 rows x 5 columns]

```

Model with correlation >= 0.8:
Training+Validation R^2: 0.99285, RMSE: 1.69117
Testing R^2: 0.98457, RMSE: 2.46818
Mean cross-validation score: 0.98116

```

	Feature	Importance
9	CC_EST_lead1	0.972498
10	CC_EST_lead2	0.014830
2	IQ_CPA_TRAN_XQ	0.004247
0	NY_ADJ_NNTY_PC_KD	0.002593
1	SP_POP_SCIE_RD_P6	0.001587
3	LP_LPI_CUST_XQ	0.001579
7	SI_SPR_PCAP	0.001365
6	SI_SPR_PC40	0.001301
4	DT_ODA_OATL_CD	0.000000
5	DT_ODA_OATL_KD	0.000000
8	DT_NFL_UNWT_CD	0.000000

	country	year	CC_EST	CC_EST_lead1	CC_EST_lead2
52	Andorra	2012	24.789383	26.043514	29.111116
53	Andorra	2013	24.929030	29.111116	26.543472
54	Andorra	2014	25.585828	26.543472	26.543472
55	Andorra	2015	26.963785	26.543472	26.543472
56	Andorra	2016	26.808882	26.543472	26.543472
...
13727	Zimbabwe	2016	75.768814	75.817085	75.626099
13728	Zimbabwe	2017	75.969706	75.626099	75.226929

```

13729 Zimbabwe 2018 74.920013    75.226929    75.290924
13730 Zimbabwe 2019 75.423806    75.290924    75.290924
13731 Zimbabwe 2020 75.759847    75.290924    75.290924

```

[1839 rows x 5 columns]

5 Model 3

```

[161]: df = pd.read_csv('WDI_data.csv')
# Remove column CPI_EST
df = df.drop(columns=['CPI_EST'])
# Make corruption more intuitive with higher values indicating more corruption
df['CC_EST'] = (-1)*df['CC_EST']
# Make years integers
df['year'] = df['year'].astype(int)
# Convert all CC_EST values to positive between 0 and 100
df['CC_EST'] = (df['CC_EST'] + 2.5) * 20

```

```

[162]: # Create a column that has the average CC_EST for each country between the year
       ↴2012 and 2022 only but prints for all years given the country
df['WB_CC_EST_avg'] = df.groupby('iso2c')['CC_EST'].transform(lambda x: x[(df['year'] >= 2012) & (df['year'] <= 2022)].mean())
df['WB_CC_EST_avg'] = np.log1p(df['WB_CC_EST_avg'])

```

```

[163]: # count the number of columns in df
len(df.columns)

```

[163]: 1455

```

[164]: # Filter df to only include years from 1970 to 2011
df_pre_2012 = df[(df['year'] >= 1960) & (df['year'] < 2012)]

# Calculate the percentage of missing values in each column for years prior to
# ↴2012
missing_percentages = df_pre_2012.isnull().mean()

# Select the columns that have less than 50% missing values, or are 'CC_EST',
# ↴'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', and 'WB_CC_EST_avg'
cols_to_keep = [col for col in df.columns if col in ['CC_EST', 'NY_GDP_PCAP_KD',
           ↴'NY_GDP_PCAP_CD', 'WB_CC_EST_avg', 'SL_UEM_TOTL_NE_ZS', 'SE_SEC_DURS',
           ↴'SI_POV_GINI'] or (col in missing_percentages and missing_percentages[col] <
           ↴0.5)]

# Keep only the selected columns in df
df = df[cols_to_keep]

```

```
[165]: # View
df[['iso2c', 'country', 'year', 'CC_EST', 'WB_CC_EST_avg', 'SL_UEM_TOTL_NE_ZS', ↴
    'SE_SEC_DURS', 'SI_POV_GINI']]
```

	iso2c	country	year	CC_EST	WB_CC_EST_avg	SL_UEM_TOTL_NE_ZS	\
0	AD	Andorra	1960	NaN	3.282977	NaN	NaN
1	AD	Andorra	1961	NaN	3.282977	NaN	NaN
2	AD	Andorra	1962	NaN	3.282977	NaN	NaN
3	AD	Andorra	1963	NaN	3.282977	NaN	NaN
4	AD	Andorra	1964	NaN	3.282977	NaN	NaN
...
13729	ZW	Zimbabwe	2018	74.920013	4.347206	NaN	NaN
13730	ZW	Zimbabwe	2019	75.423806	4.347206	7.373	7.373
13731	ZW	Zimbabwe	2020	75.759847	4.347206	NaN	NaN
13732	ZW	Zimbabwe	2021	75.071001	4.347206	9.540	9.540
13733	ZW	Zimbabwe	2022	75.102789	4.347206	NaN	NaN
	SE_SEC_DURS	SI_POV_GINI					
0				NaN	NaN		
1				NaN	NaN		
2				NaN	NaN		
3				NaN	NaN		
4				NaN	NaN		
...		
13729				6.0	NaN		
13730				6.0	50.3		
13731				6.0	NaN		
13732				6.0	NaN		
13733				6.0	NaN		

[13734 rows x 8 columns]

```
[166]: import pycountry

# Create a dictionary that maps ISO 2-letter country codes to a dictionary containing the ISO 3-letter and numeric country codes
country_codes = {country.alpha_2: {'alpha-3': country.alpha_3, 'numeric': country.numeric} for country in pycountry.countries}

# Now you can use this dictionary in your code
df['iso3c'] = df['iso2c'].map(lambda x: country_codes.get(x, {}).get('alpha-3'))
df['iso3n'] = df['iso2c'].map(lambda x: country_codes.get(x, {}).get('numeric'))
```

```
[167]: """
def backfill_based_on_avg_change(df, column):
    """
```

```

Backfill missing values in the specified column based on the averageu
→percentage change of the next 5 years for each country.

"""
# Iterate over each country
for country in df['iso2c'].unique():
    # Get the data for the current country
    df_country = df[df['iso2c'] == country].copy()

    # Sort the data by year in ascending order
    df_country.sort_values('year', inplace=True)

    # Identify the indices of the missing values in the column
    missing_indices = df_country[df_country[column].isnull() &
        →(df_country['year'] < 2012)].index.tolist()

    # For each missing value, calculate the average percentage change of
    →the next 5 years and use it to fill the missing value
    for idx in missing_indices:
        if df_country.loc[idx, 'year'] < 1950:
            continue
        next_10_years_avg_pct_change = df_country.loc[idx+1:idx+11, column].
        →pct_change().mean()

        # Apply the average percentage change to each year as it
        →extrapolates backwards
        for i in reversed(range(missing_indices[0], idx+1)):
            if i+1 in df_country.index:
                df_country.loc[i, column] = df_country.loc[i+1, column] *u
        →(1 - next_10_years_avg_pct_change)

    # Update the column in the original DataFrame
    df.loc[df['iso2c'] == country, column] = df_country[column]

return df

# Use the function to backfill the missing values in 'NY_GDP_PCAP_KD' and
→'NY_GDP_PCAP_CD'
df = backfill_based_on_avg_change(df, 'NY_GDP_PCAP_KD')
df = backfill_based_on_avg_change(df, 'NY_GDP_PCAP_CD')
df['NY_GDP_PCAP_KD_rel'] = df.groupby('year')['NY_GDP_PCAP_KD'].
    →transform(lambda x: x / x.mean())
df['NY_GDP_PCAP_CD_rel'] = df.groupby('year')['NY_GDP_PCAP_CD'].
    →transform(lambda x: x / x.mean())
"""

```

Cell In[167], line 4

Backfill missing values in the specified column based on the average percentage change of the next 5 years for each country.

IndentationError: unexpected indent

```
[168]: def backfill_based_on_avg_change(df, column):
    """
    Backfill missing values in the specified column based on the average percentage change of the next 10 years for each country.
    """
    # Iterate over each country
    for country in df['iso2c'].unique():
        # Get the data for the current country
        df_country = df[df['iso2c'] == country].copy()

        # Sort the data by year in ascending order
        df_country.sort_values('year', inplace=True)

        # Identify the indices of the missing values in the column
        missing_indices = df_country[df_country[column].isnull() & (df_country['year'] < 2012)].index.tolist()

        # For each missing value, calculate the average percentage change of the next 10 years and use it to fill the missing value
        for idx in missing_indices:
            if df_country.loc[idx, 'year'] < 1950:
                continue
            next_10_years_avg_pct_change = df_country.loc[idx+1:idx+11, column].pct_change().mean()

            # Apply a resistance factor to the average percentage change
            resistance_factor = 1 / (abs(next_10_years_avg_pct_change) + 1)
            if abs(next_10_years_avg_pct_change) > 0.05:
                resistance_factor *= 1 / (10 * abs(next_10_years_avg_pct_change) + 1)
            next_10_years_avg_pct_change *= resistance_factor

            # Apply the average percentage change to each year as it extrapolates backwards
            for i in reversed(range(missing_indices[0], idx+1)):
                if i+1 in df_country.index:
                    df_country.loc[i, column] = df_country.loc[i+1, column] * (1 - next_10_years_avg_pct_change)

        # Update the column in the original DataFrame
```

```

df.loc[df['iso2c'] == country, column] = df_country[column]

return df

# Use the function to backfill the missing values in 'NY_GDP_PCAP_KD' and
↪ 'NY_GDP_PCAP_CD'
df = backfill_based_on_avg_change(df, 'NY_GDP_PCAP_KD')
df = backfill_based_on_avg_change(df, 'NY_GDP_PCAP_CD')
df = backfill_based_on_avg_change(df, 'NY_GDP_PCAP_CN')
df['NY_GDP_PCAP_KD_rel'] = df.groupby('year')['NY_GDP_PCAP_KD'].
    ↪transform(lambda x: x / x.mean())
df['NY_GDP_PCAP_CD_rel'] = df.groupby('year')['NY_GDP_PCAP_CD'].
    ↪transform(lambda x: x / x.mean())
df['NY_GDP_PCAP_CD_rel'] = df.groupby('year')['NY_GDP_PCAP_CN'].
    ↪transform(lambda x: x / x.mean())

```

```

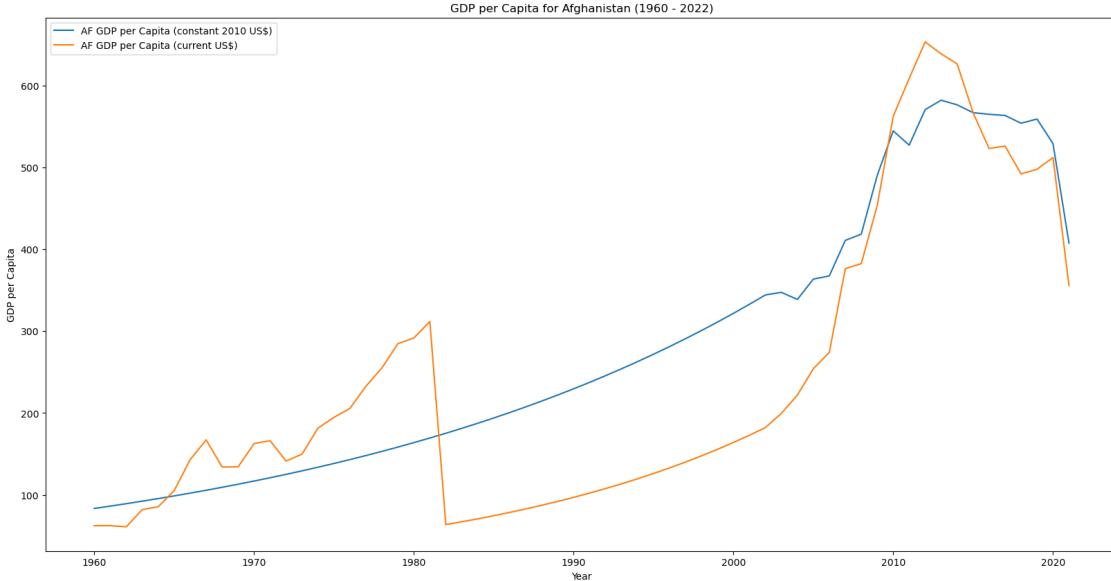
[180]: # Plot GDP per capita in constant 2010 US$ and current US$ from 1960 to 2022
      ↪for Afghanistan
# Create a line plot of the data
plt.figure(figsize=(20, 10))

countries = ['AF']
for country in countries:
    # Plot the GDP per capita in constant 2010 US$ for the current country
    sns.lineplot(data=df[df['iso2c'] == country], x='year', y='NY_GDP_PCAP_KD',
    ↪label=f'{country} GDP per Capita (constant 2010 US$)')

    # Plot the GDP per capita in current US$ for the current country
    sns.lineplot(data=df[df['iso2c'] == country], x='year', y='NY_GDP_PCAP_CD',
    ↪label=f'{country} GDP per Capita (current US$)')

plt.title('GDP per Capita for Afghanistan (1960 - 2022)')
plt.xlabel('Year')
plt.ylabel('GDP per Capita')
plt.legend()
plt.show()

```



```
[181]: df_dr = df[df['year'] >= 2012]
df_dr = df_dr.dropna(subset=['CC_EST'])

# Create a new dataframe with only iso2c, country, and CC_EST
df_corr = df_dr[['iso2c', 'country', 'year', 'CC_EST']]

# Filter the data for the years 2012 and 2022
df_filtered = df_corr[df_corr['year'].isin([2012, 2022])]
```



```
[182]: # Calculate the linear differences over time for each given country
change_by_country = df_filtered.pivot_table(index='iso2c', columns='year',
                                             values='CC_EST', aggfunc='first')

# Calculate the change for each country
change_by_country['avg_change'] = (change_by_country[2022] -
                                   change_by_country[2012]) / 10

# Calculate the mean and median CC_EST values across all years for each country
mean_cc_est_by_country = df_filtered.groupby('iso2c')['CC_EST'].mean()
median_cc_est_by_country = df_filtered.groupby('iso2c')['CC_EST'].median()

# Merge with the original DataFrame to get the country names
df_trend = pd.merge(change_by_country['avg_change'], mean_cc_est_by_country,
                     left_index=True, right_index=True, how='left')
df_trend = pd.merge(df_trend, median_cc_est_by_country, left_index=True,
                     right_index=True, how='left')
```

```

# Reset index and rename columns
df_trend.reset_index(inplace=True)
df_trend.rename(columns={'CC_EST_x': 'mean_CC_EST', 'CC_EST_y': 'median_CC_EST'}, inplace=True)

# Include country names
df_trend = pd.merge(df_trend, df_filtered[['iso2c', 'country']].drop_duplicates(), on='iso2c', how='left')

print(df_trend)

```

	iso2c	avg_change	mean_CC_EST	median_CC_EST	country
0	AD	-0.019346	24.692656	24.692656	Andorra
1	AE	0.000799	26.889273	26.889273	United Arab Emirates
2	AF	-0.493193	76.141493	76.141493	Afghanistan
3	AG	1.899853	34.288647	34.288647	Antigua and Barbuda
4	AL	-0.741708	61.866050	61.866050	Albania
..
199	XK	-0.779124	59.171521	59.171521	Kosovo
200	YE	0.841619	79.383066	79.383066	Yemen, Rep.
201	ZA	0.271189	55.039358	55.039358	South Africa
202	ZM	0.475517	58.206411	58.206411	Zambia
203	ZW	-0.253328	76.369429	76.369429	Zimbabwe

[204 rows x 5 columns]

```

[183]: # Calculate the mean 'mean_CC_EST' value for each country
mean_values = df_trend.groupby('country')['mean_CC_EST'].mean()

# Find the index of the maximum and minimum mean 'mean_CC_EST' values
most_corrupt_country = mean_values.idxmax()
least_corrupt_country = mean_values.idxmin()

print("The most corrupt country (on average between 2012 and 2022) is", most_corrupt_country)
print("The least corrupt country (on average between 2012 and 2022) is", least_corrupt_country)

# Calculate the mean 'avg_change' value for each country
mean_changes = df_trend.groupby('country')['avg_change'].mean()

print("The country that became more corrupt (rose the most) on average between 2012 and 2022 is", mean_changes.idxmax())
print("The country that became less corrupt (fell the most) on average between 2012 and 2022 is", mean_changes.idxmin())

```

The most corrupt country (on average between 2012 and 2022) is Somalia
The least corrupt country (on average between 2012 and 2022) is Denmark

The country that became more corrupt (rose the most) on average between 2012 and 2022 is Antigua and Barbuda

The country that became less corrupt (fell the most) on average between 2012 and 2022 is Seychelles

```
[184]: df_correl = pd.read_csv('correlations.csv')
# Rank in order of correlation with CC_EST
df_correl = df_correl.sort_values('correlation', ascending=False)

[185]: # Remove rows in df_dr that have missing values in CC_EST column
df_dr = df[df['year'] >= 2012]
df_dr = df_dr.dropna(subset=['CC_EST'])

# Filter to only show correlations greater than or equal to +- 0.5
df_correl_5 = df_correl[abs(df_correl['correlation'])] >= 0.5]
# Filter to only show correlations greater than or equal to +- 0.6
df_correl_6 = df_correl[abs(df_correl['correlation'])] >= 0.6]
# Filter to only show correlations greater than or equal to +- 0.7
df_correl_7 = df_correl[abs(df_correl['correlation'])] >= 0.7]
# Filter to only show correlations greater than or equal to +- 0.8
df_correl_8 = df_correl[abs(df_correl['correlation'])] >= 0.8]
# Filter to only show correlations greater than or equal to +- 0.9
df_correl_9 = df_correl[abs(df_correl['correlation'])] >= 0.9]

print(f"The number of correlations with an absolute value greater than or equal to 0.5 is", len(df_correl_5))
print(f"The number of correlations with an absolute value greater than or equal to 0.6 is", len(df_correl_6))
print(f"The number of correlations with an absolute value greater than or equal to 0.7 is", len(df_correl_7))
print(f"The number of correlations with an absolute value greater than or equal to 0.8 is", len(df_correl_8))
print(f"The number of correlations with an absolute value greater than or equal to 0.9 is", len(df_correl_9))
```

The number of correlations with an absolute value greater than or equal to 0.5 is 253

The number of correlations with an absolute value greater than or equal to 0.6 is 128

The number of correlations with an absolute value greater than or equal to 0.7 is 54

The number of correlations with an absolute value greater than or equal to 0.8 is 7

The number of correlations with an absolute value greater than or equal to 0.9 is 1

```
[186]: # Merge together corruption values and the largest correlators
# Create variables that are a list of the variables that are highly correlated with CC_EST
vars_5 = df_correl_5.columns.tolist()
vars_5 = [var for var in vars_5 if var in df_dr.columns]
vars_6 = df_correl_6.columns.tolist()
vars_6 = [var for var in vars_6 if var in df_dr.columns]
vars_7 = df_correl_7.columns.tolist()
vars_7 = [var for var in vars_7 if var in df_dr.columns]
vars_8 = df_correl_8.columns.tolist()
vars_8 = [var for var in vars_8 if var in df_dr.columns]
vars_9 = df_correl_9.columns.tolist()
vars_9 = [var for var in vars_9 if var in df_dr.columns]

# Merge the corruption values with the variables that are highly correlated with CC_EST
df_5 = df_dr[['iso2c', 'country', 'year', 'CC_EST', 'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel', 'SL_UEM_TOTL_NE_ZS', 'SE_SEC_DURS', 'SI_POV_GINI']] + vars_5
df_6 = df_dr[['iso2c', 'country', 'year', 'CC_EST', 'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel', 'SL_UEM_TOTL_NE_ZS', 'SE_SEC_DURS', 'SI_POV_GINI']] + vars_6
df_7 = df_dr[['iso2c', 'country', 'year', 'CC_EST', 'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel', 'SL_UEM_TOTL_NE_ZS', 'SE_SEC_DURS', 'SI_POV_GINI']] + vars_7
df_8 = df_dr[['iso2c', 'country', 'year', 'CC_EST', 'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel', 'SL_UEM_TOTL_NE_ZS', 'SE_SEC_DURS', 'SI_POV_GINI']] + vars_8
df_9 = df_dr[['iso2c', 'country', 'year', 'CC_EST', 'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel', 'SL_UEM_TOTL_NE_ZS', 'SE_SEC_DURS', 'SI_POV_GINI']] + vars_9
# Build random forest model to predict corruption from highly correlated variables
# Simplify notation:
x5 = df_5.iloc[:,2:].values
x6 = df_6.iloc[:,2:].values
x7 = df_7.iloc[:,2:].values
x8 = df_8.iloc[:,2:].values
x9 = df_9.iloc[:,2:].values

y = df_dr['CC_EST'].values
```

```
[187]: def prepare_lead_data(df, target_col, lead_cols, time_col, country_col):
    """Shifts features to create lead versions for backcasting,
    accounting for country and year structure.
    """

```

```

df_lead = df.copy()

for col in lead_cols:
    df_lead[f'{col}_lead1'] = df_lead.sort_values([time_col]).groupby(country_col)[col].shift(-1)

df_lead.fillna(0, inplace=True)
return df_lead


def remove_correlated_features(df, threshold):
    """Removes highly correlated features based on a threshold."""
    corr_matrix = df.dropna().corr().abs()
    upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
    cols_to_drop = [col for col in upper.columns if any(upper[col] > threshold)]
    df = df.drop(cols_to_drop, axis=1)
    return df


def custom_cross_val_score(model, X, y, cv):
    """Calculates cross-validation scores for a given model."""
    scores = []
    kf = KFold(n_splits=cv)

    for train_index, val_index in kf.split(X):
        x_train, x_val = X.iloc[train_index], X.iloc[val_index]
        y_train, y_val = y.iloc[train_index], y.iloc[val_index]

        model.fit(x_train, y_train, eval_set=[(x_val, y_val)], verbose=False)

        y_val_pred = model.predict(x_val)
        r2_val = round(r2_score(y_val, y_val_pred), 5)
        scores.append(r2_val)

    return scores

```

```

[188]: def custom_objective(y_true, y_pred, indices):
    """
    Custom objective function that penalizes predictions close to or beyond the
    bounds of 0 and 100,
    penalizes large changes from the previous year's prediction, and penalizes
    an overall movement greater than 20 from the 2012 value.
    """

    # Convert y_true and y_pred to pandas Series
    y_true = pd.Series(y_true)
    y_pred = pd.Series(y_pred)

```

```

# Select the corresponding value from y_2012 for each y_true and y_pred
y_2012 = df['WB_CC_EST_avg'].loc[indices]

# Calculate the squared error
squared_error = (y_true - y_pred) ** 2

# Calculate the penalty term
penalty = np.abs(y_pred - y_2012)

# Add the penalty term to the squared error
penalized_squared_error = squared_error + penalty

# Calculate the first derivative (gradient) of the penalized squared error
grad = -2 * (y_true - y_pred) + np.sign(y_pred - y_2012)

# Calculate the second derivative (Hessian) of the penalized squared error
hess = np.ones_like(grad) * 2

return grad, hess

class TqdmCallback(xgb.callback.TrainingCallback):
    def __init__(self, bar):
        self._bar = bar

    def after_iteration(self, model, epoch, evals_log):
        self._bar.update(1)
        return False

def build_and_evaluate_model(df, target_col_name, var_list, n_leads=2):
    """
    Builds, evaluates, and returns results for an XGBoost model.
    """
    # Create a copy of the DataFrame to avoid modifying the original
    df = df.copy()

    # Exclude 'iso2c', 'country', target_col_name, and lead variables from
    # var_list
    var_list = [var for var in var_list if var not in ['iso2c', 'country', target_col_name] + [f"{target_col_name}_lead{i}" for i in range(1, n_leads + 1)]]

    # Remove rows with invalid values in the target column
    df = df[np.isfinite(df[target_col_name]) & (abs(df[target_col_name]) <= 1e30)]

    # Add a new feature for the previous year's target value
    df[target_col_name + '_prev'] = df.groupby('iso2c')[target_col_name].shift()

```

```

var_list.append(target_col_name + '_prev')

X = df[var_list]

df_train_val, df_test = train_test_split(df, test_size=0.2, random_state=0)

x_train_val, x_test, y_train_val, y_test = train_test_split(X, □
↳ df[target_col_name], test_size=0.2, random_state=0)

# Remove correlated features
x_lead = x_train_val
x_test = x_test[x_lead.columns]

# Align datasets before model training
x_lead, x_test = x_lead.align(x_test, join='inner', axis=1)

model = XGBRegressor(
    objective='reg:squarederror',
    random_state=0,
    alpha=1.0,
    reg_lambda=10.0,
    early_stopping_rounds=10
)

with tqdm(total=model.get_params()['n_estimators']) as pbar:
    model.fit(
        x_lead,
        y_train_val,
        eval_set=[(x_test, y_test)],
        verbose=False,
        callbacks=[TqdmCallback(pbar)]
    )

y_train_val_pred = model.predict(x_lead)
y_test_pred = model.predict(x_test)

# Clip the predictions to be within the desired range
y_train_val_pred = np.clip(y_train_val_pred, 0, 100)
y_test_pred = np.clip(y_test_pred, 0, 100)

r2_train_val = round(r2_score(y_train_val, y_train_val_pred), 5)
r2_test = round(r2_score(y_test, y_test_pred), 5)

rmse_train_val = round(np.sqrt(mean_squared_error(y_train_val, □
↳ y_train_val_pred)), 5)
rmse_test = round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 5)

```

```

cv_scores = custom_cross_val_score(model, x_train_val, y_train_val, cv=5)
mean_cv_score = round(np.mean(cv_scores), 5)

# Get the feature importances
feature_importances = model.feature_importances_

# Get the feature names from x_lead
feature_names = x_lead.columns.tolist()

# Align feature importances with feature names
feature_importances_aligned = pd.Series(feature_importances, index=feature_names)

# Create a DataFrame with the aligned feature importances
feature_importances_df = pd.DataFrame({
    'Feature': feature_importances_aligned.index,
    'Importance': np.round(feature_importances_aligned.values, 4)
}).sort_values(by='Importance', ascending=False)

return r2_train_val, r2_test, rmse_train_val, rmse_test, mean_cv_score,
feature_importances_df, x_test.index, y_test_pred, y_test, df, model,
feature_importances

```

[189]: *## TESTS ACROSS DIFFERENT CORRELATORY LEVELS*

```

# Set the style to 'default' to make the background white
style.use('default')

# Replace 'vars_full' with your actual list of variables, excluding 'iso2c',
'country', and 'CC_EST'
vars_full = [var for var in df.columns if var not in ['iso2c', 'country', 'CC_EST']]

print("Running model on full dataset")
for i in range(0, 9):
    # Calculate the correlation of each variable with 'CC_EST'
    correl = df[vars_full].select_dtypes(include=['number']).corrwith(df['CC_EST']).abs()

    # Get the variables with a correlation greater than or equal to the
threshold
    vars_correl = correl[correl >= 0.1 * i].index.tolist()

    results = build_and_evaluate_model(df, 'CC_EST', vars_correl, n_leads=2)

```

```

print(f"\nModel with correlation >= 0.{i}:")
print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
print(f"Mean cross-validation score: {results[4]}\n")
print(results[5]) # Feature importances
print('\n')
print(results[9][['country', 'year', 'CC_EST']])

# Generate the graphs
y_test = results[8]
y_test_pred = results[7]

# Create a scatter plot for the actual vs predicted values
abs_diffs = np.abs(y_test - y_test_pred)

# Create a DataFrame with the actual values and absolute differences
df_plot = pd.DataFrame({'Actual': y_test, 'AbsDifference': abs_diffs})

# Define the bins for the actual values
bins = np.linspace(0, 100, 50)

# Calculate the mid-points of the bins
bin_midpoints = bins[:-1] + np.diff(bins) / 2

# Create a new column for the binned actual values
df_plot['ActualBin'] = pd.cut(df_plot['Actual'], bins, labels=bin_midpoints)

# Group by the binned actual values and calculate the variance of the
# absolute differences for each group
var_abs_diffs = df_plot.groupby('ActualBin')['AbsDifference'].var()

# Create a scatter plot for the actual vs predicted values
fig = plt.figure(figsize=(10, 10))
gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1])
ax0 = plt.subplot(gs[0])
ax0.scatter(y_test, y_test_pred, alpha=0.7, color='#00688B', s=20)
ax0.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], lw=3)
ax0.set_xlim([0, 100])
ax0.set_ylabel('Predicted', fontsize=14)
ax0.set_title(f'Actual vs Predicted Values and Variance of Absolute
#Differences\n Correlation > 0.{i}', fontsize=16)
ax0.grid(True, color='grey', linestyle='-', linewidth=0.25, alpha=0.5)

# Create a line plot for the binned actual values vs variance of the
# absolute differences
ax1 = plt.subplot(gs[1])

```

```

# Apply LOESS to smooth the variance curve
smoothed = lowess(var_abs_diffs, var_abs_diffs.index, frac=0.5)
index, data = zip(*smoothed)
ax1.plot(index, data, color='#00688B')

ax1.set_ylim([0, 16])
ax1.set_xlim([0, 100])
ax1.set_ylabel('Variance of Abs. Diff.', fontsize=14)
ax1.grid(axis='y', color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

ax1.set_xticklabels([])
ax1.set_xticks([])
ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)
ax0.spines['right'].set_visible(False)
ax0.spines['top'].set_visible(False)

plt.tight_layout()
plt.show()

```

Running model on full dataset

```

Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
13it [00:00, 28.98it/s]29it [00:01, 23.54it/s]

```

Model with correlation >= 0.0:
Training+Validation R^2: 0.997, RMSE: 1.09271
Testing R^2: 0.98088, RMSE: 2.79089
Mean cross-validation score: 0.98084

	Feature	Importance
298	WB_CC_EST_avg	0.8569
301	CC_EST_prev	0.0157
140	NY_GDP_PCAP_CN	0.0111
201	SP_POP_0014_MA_IN	0.0031
153	NY_GNP_PCAP_CN	0.0029
..
213	SP_POP_1564_MA_IN	0.0000
258	SP_POP_TOTL_MA_ZS	0.0000
259	SP_RUR_TOTL	0.0000
130	NY_GDP_FCST_CN	0.0000

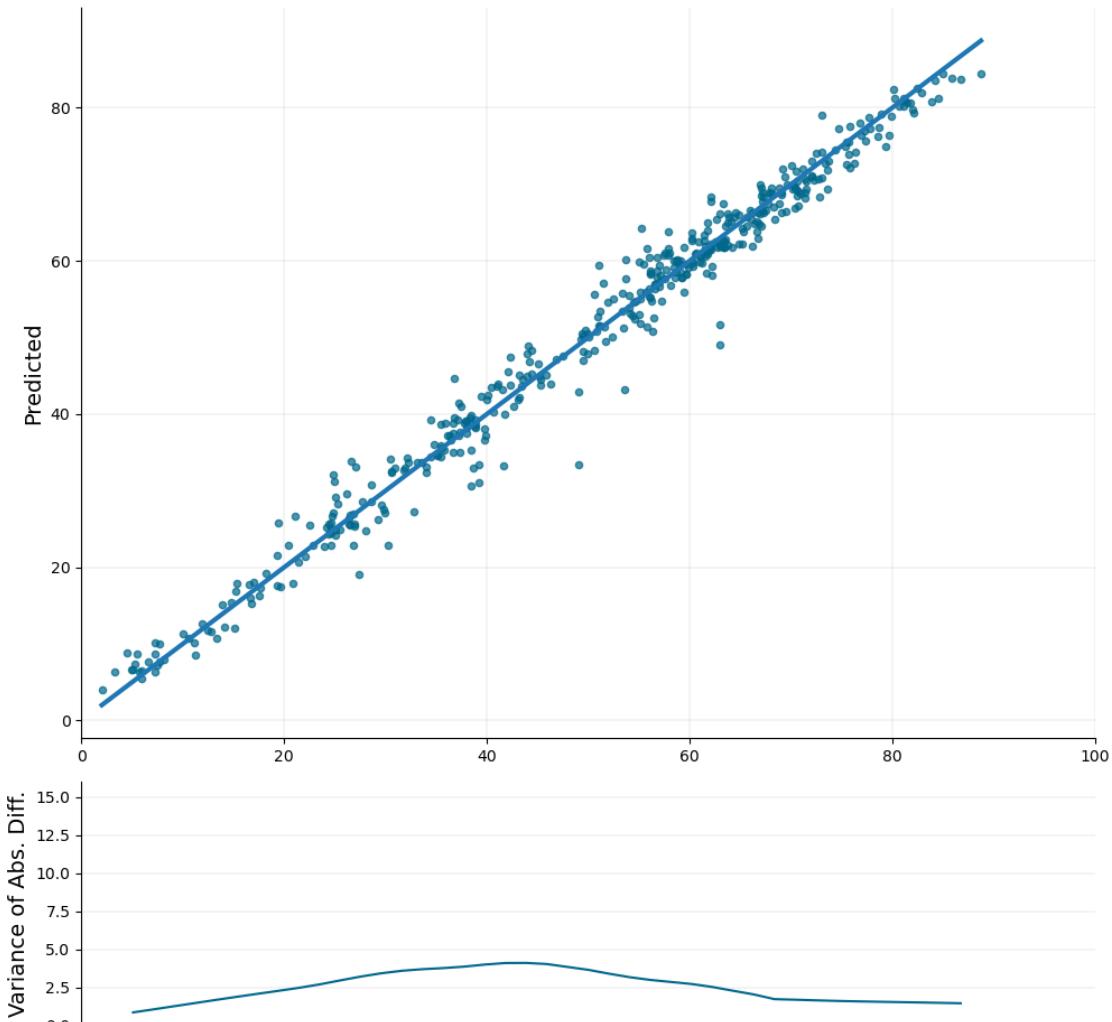
```
257 SP_POP_TOTL_MA_IN      0.0000
```

```
[302 rows x 2 columns]
```

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
...
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

```
[2249 rows x 3 columns]
```

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.0



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
35it [00:00, 49.44it/s]
```

Model with correlation >= 0.1:
 Training+Validation R^2: 0.99761, RMSE: 0.97555
 Testing R^2: 0.97951, RMSE: 2.88927
 Mean cross-validation score: 0.98049

Feature Importance

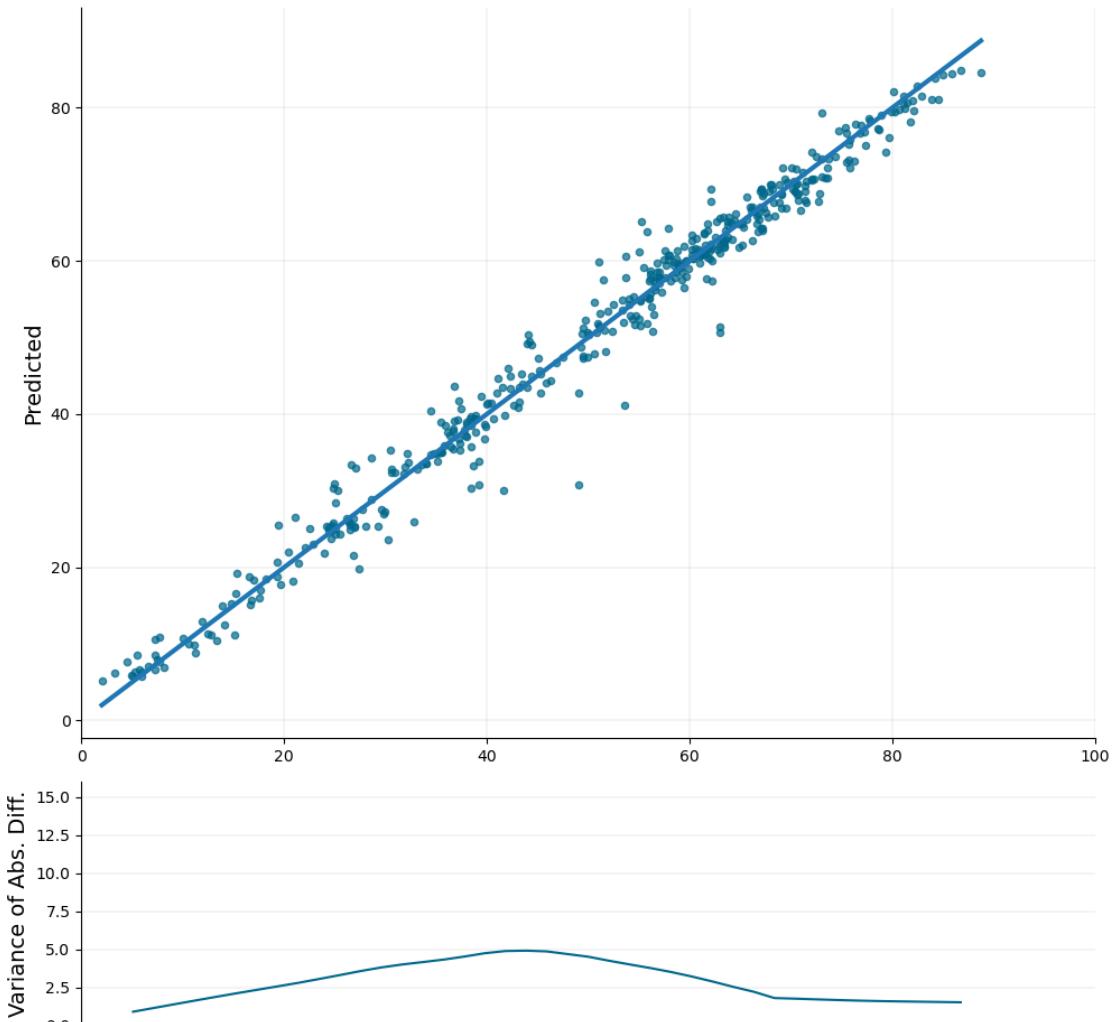
193	WB_CC_EST_avg	0.8901
195	CC_EST_prev	0.0138
113	SP_POP_0004_FE_5Y	0.0060
116	SP_POP_0014_FE_ZS	0.0057
12	DT_ODA_ALLD_KD	0.0033
..
165	SP_URB_TOTL_IN_ZS	0.0000
105	SP_DYN_IMRT_IN	0.0000
149	SP_POP_65UP_MA_ZS	0.0000
150	SP_POP_65UP_TO_ZS	0.0000
147	SP_POP_6569_MA_5Y	0.0000

[196 rows x 2 columns]

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
..
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

[2249 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.1



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
43it [00:00, 81.71it/s]
```

Model with correlation >= 0.2:
 Training+Validation R^2: 0.99793, RMSE: 0.90663
 Testing R^2: 0.97913, RMSE: 2.91614
 Mean cross-validation score: 0.98014

Feature Importance

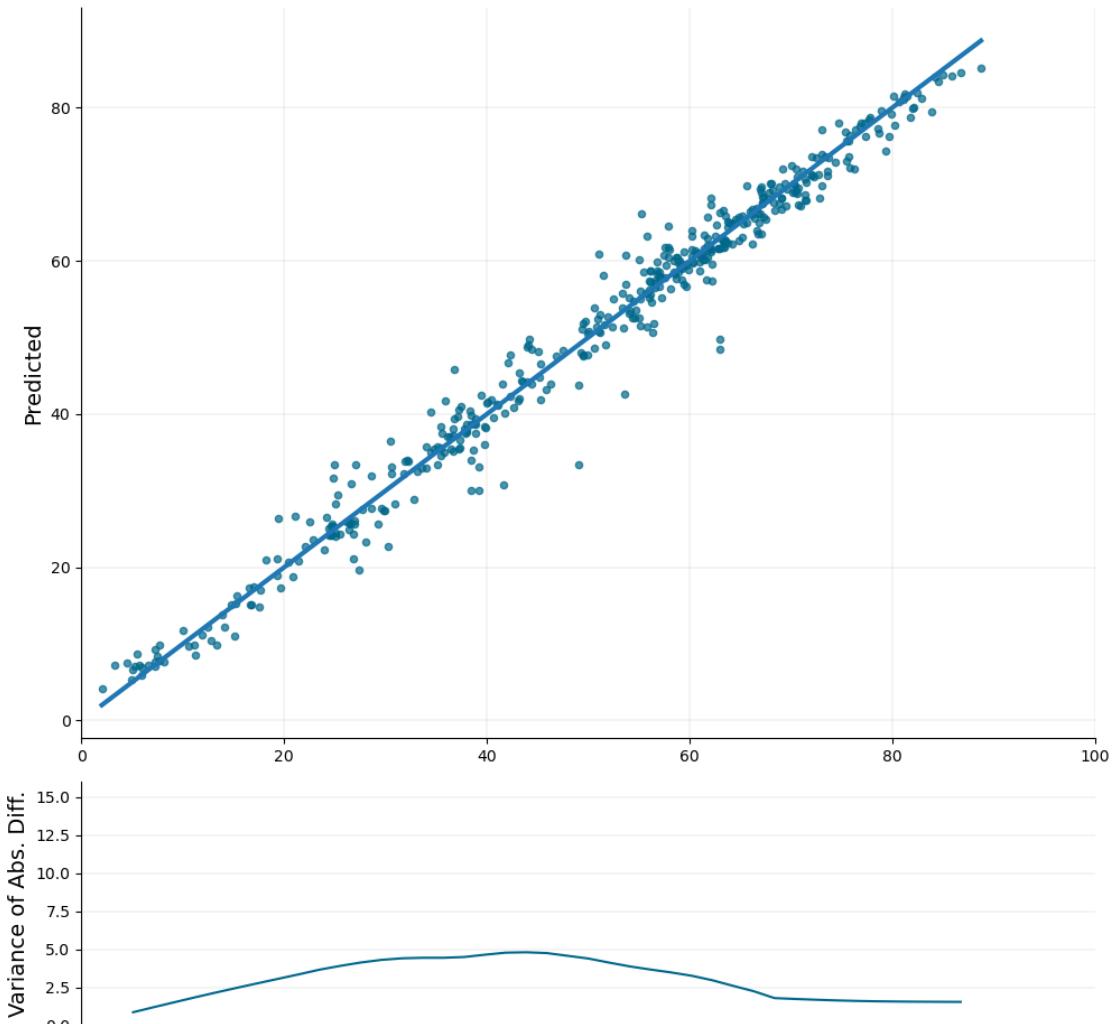
130	WB_CC_EST_avg	0.9400
132	CC_EST_prev	0.0129
69	SP_POP_0004_FE_5Y	0.0038
71	SP_POP_0014_FE_ZS	0.0018
9	DT_ODA_ALLD_KD	0.0015
..
11	DT_ODA_ODAT_KD	0.0000
116	SP_URB_TOTL_IN_ZS	0.0000
110	SP_POP_DPND_OL	0.0000
53	SH_DYN_MORT_MA	0.0000
72	SP_POP_0014_MA_ZS	0.0000

[133 rows x 2 columns]

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
..
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

[2249 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.2



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
84it [00:00, 97.44it/s]
```

Model with correlation >= 0.3:
 Training+Validation R^2: 0.99944, RMSE: 0.47309
 Testing R^2: 0.98085, RMSE: 2.7933
 Mean cross-validation score: 0.98077

Feature Importance

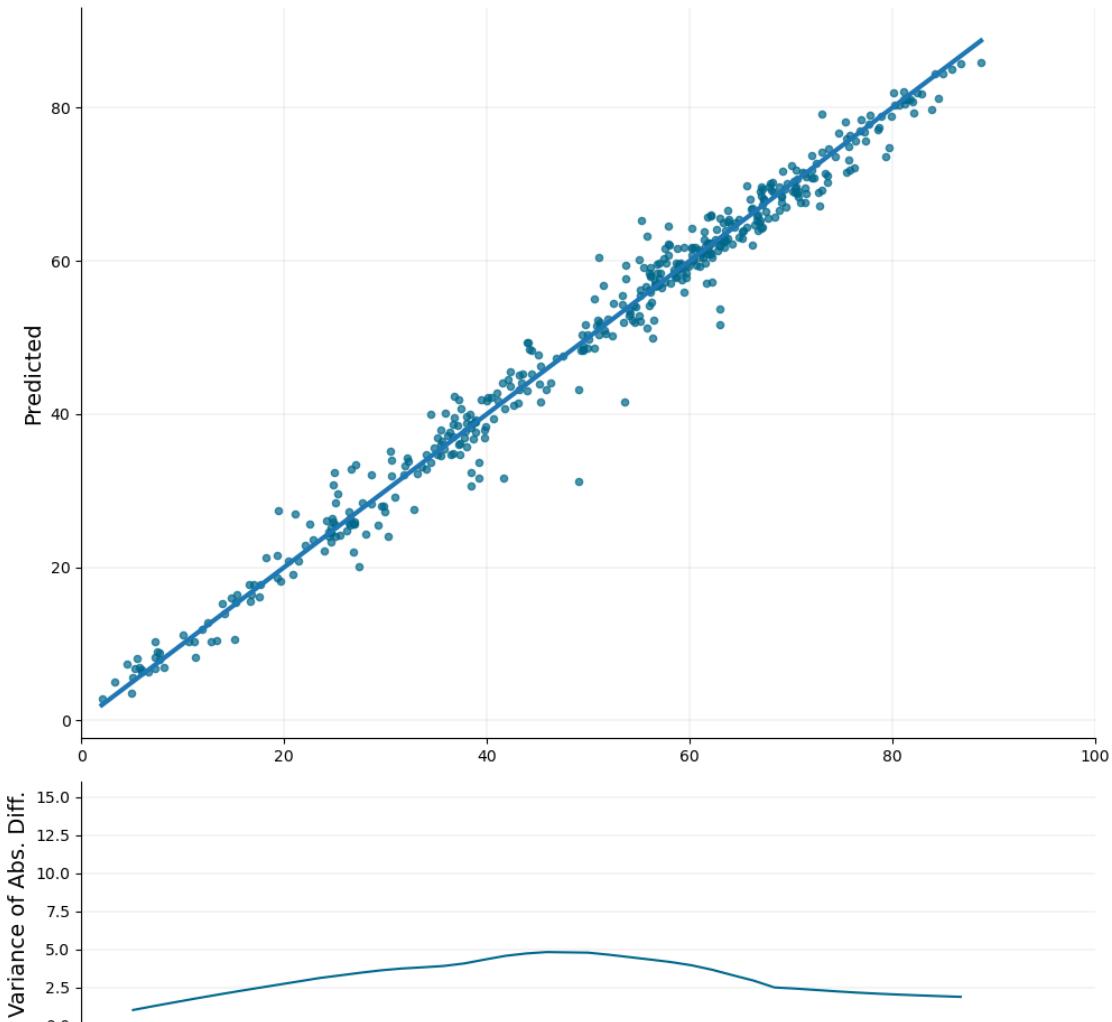
99	WB_CC_EST_avg	0.9453
101	CC_EST_prev	0.0133
52	SP_POP_0004_FE_5Y	0.0027
54	SP_POP_0014_FE_ZS	0.0020
51	SP_DYN_T065_MA_ZS	0.0018
..
82	SP_POP_65UP_TO_ZS	0.0001
34	SH_DYN_MORT	0.0001
0	AG_CON_FERT_ZS	0.0001
95	SP_URB_TOTL_IN_ZS	0.0000
81	SP_POP_65UP_MA_ZS	0.0000

[102 rows x 2 columns]

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
..
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

[2249 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.3



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
100it [00:00, 156.44it/s]
```

Model with correlation ≥ 0.4 :
 Training+Validation R²: 0.99967, RMSE: 0.35981
 Testing R²: 0.9823, RMSE: 2.68567
 Mean cross-validation score: 0.98179

Feature Importance

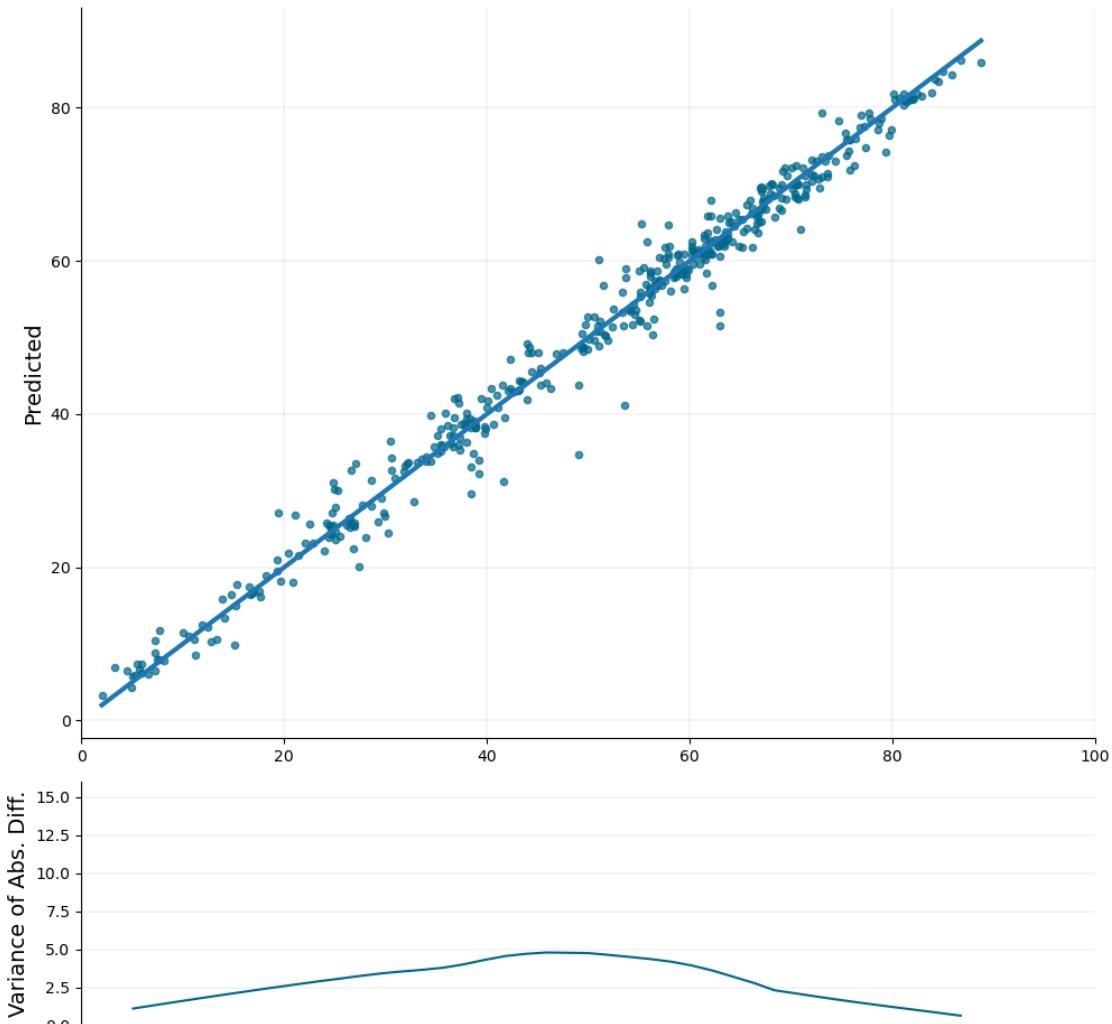
72	WB_CC_EST_avg	0.9452
74	CC_EST_prev	0.0129
29	SP_POP_0004_FE_5Y	0.0033
31	SP_POP_0014_FE_ZS	0.0027
28	SP_DYN_TO65_MA_ZS	0.0020
33	SP_POP_0014_TO_ZS	0.0012
22	SP_DYN_IMRT_MA_IN	0.0012
38	SP_POP_1519_FE_5Y	0.0011
59	SP_POP_65UP_TO_ZS	0.0010
47	SP_POP_4549_FE_5Y	0.0009
6	NV_AGR_TOTL_ZS	0.0009
62	SP_POP_7579_FE_5Y	0.0009
18	SP_DYN_AMRT_MA	0.0008
26	SP_DYN_TFRT_IN	0.0008
66	SP_POP_DPNP	0.0007
21	SP_DYN_IMRT_IN	0.0007
35	SP_POP_0509_MA_5Y	0.0007
60	SP_POP_7074_FE_5Y	0.0007
17	SP_DYN_AMRT_FE	0.0007
40	SP_POP_1564_FE_ZS	0.0007
11	SG_LAW_INDX	0.0007
45	SP_POP_4044_FE_5Y	0.0007
73	NY_GDP_PCAP_KD_rel	0.0006
20	SP_DYN_IMRT_FE_IN	0.0006
43	SP_POP_2024_FE_5Y	0.0006
10	NY_GNP_PCAP_CD	0.0006
64	SP_POP_80UP_FE_5Y	0.0005
44	SP_POP_2024_MA_5Y	0.0005
46	SP_POP_4044_MA_5Y	0.0005
50	SP_POP_5054_MA_5Y	0.0005
53	SP_POP_6064_FE_5Y	0.0005
54	SP_POP_6064_MA_5Y	0.0005
24	SP_DYN_LEOO_IN	0.0005
23	SP_DYN_LEOO_FE_IN	0.0005
5	NE_EXP_GNFS_ZS	0.0005
9	NY_GDP_PCAP_KD	0.0004
55	SP_POP_6569_FE_5Y	0.0004
51	SP_POP_5559_FE_5Y	0.0004
3	IT_MLT_MAIN_P2	0.0004
49	SP_POP_5054_FE_5Y	0.0004
48	SP_POP_4549_MA_5Y	0.0004
7	NV_SRV_TOTL_ZS	0.0004
71	TM_VAL_MRCH_HI_ZS	0.0004
42	SP_POP_1564_TO_ZS	0.0004
70	SP_URB_TOTL_IN_ZS	0.0004
32	SP_POP_0014_MA_ZS	0.0004
56	SP_POP_6569_MA_5Y	0.0004
13	SH_DYN_MORT_FE	0.0004

39	SP_POP_1519_MA_5Y	0.0004
69	SP_RUR_TOTL_ZS	0.0004
68	SP_POP_DPND_YG	0.0004
19	SP_DYN_CBRT_IN	0.0004
65	SP_POP_80UP_MA_5Y	0.0004
67	SP_POP_DPND_DL	0.0003
0	FD_AST_PRVT_GD_ZS	0.0003
52	SP_POP_5559_MA_5Y	0.0003
30	SP_POP_0004_MA_5Y	0.0003
2	IT_CEL_SETS_P2	0.0003
4	NE_CON_PRVT_ZS	0.0003
12	SH_DYN_MORT	0.0003
14	SH_DYN_MORT_MA	0.0003
27	SP_DYN_T065_FE_ZS	0.0003
37	SP_POP_1014_MA_5Y	0.0003
34	SP_POP_0509_FE_5Y	0.0002
63	SP_POP_7579_MA_5Y	0.0002
61	SP_POP_7074_MA_5Y	0.0002
41	SP_POP_1564_MA_ZS	0.0002
25	SP_DYN_LE00_MA_IN	0.0002
16	SP_ADO_TFR	0.0002
15	SH_DYN_NMRT	0.0002
36	SP_POP_1014_FE_5Y	0.0002
1	FM_AST_PRVT_GD_ZS	0.0002
8	NY_GDP_PCAP_CD	0.0002
58	SP_POP_65UP_MA_ZS	0.0002
57	SP_POP_65UP_FE_ZS	0.0002

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
...
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

[2249 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.4



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
57it [00:00, 185.36it/s]
```

Model with correlation ≥ 0.5 :
 Training+Validation $R^2: 0.99803$, RMSE: 0.88484
 Testing $R^2: 0.98124$, RMSE: 2.76455
 Mean cross-validation score: 0.98201

Feature Importance

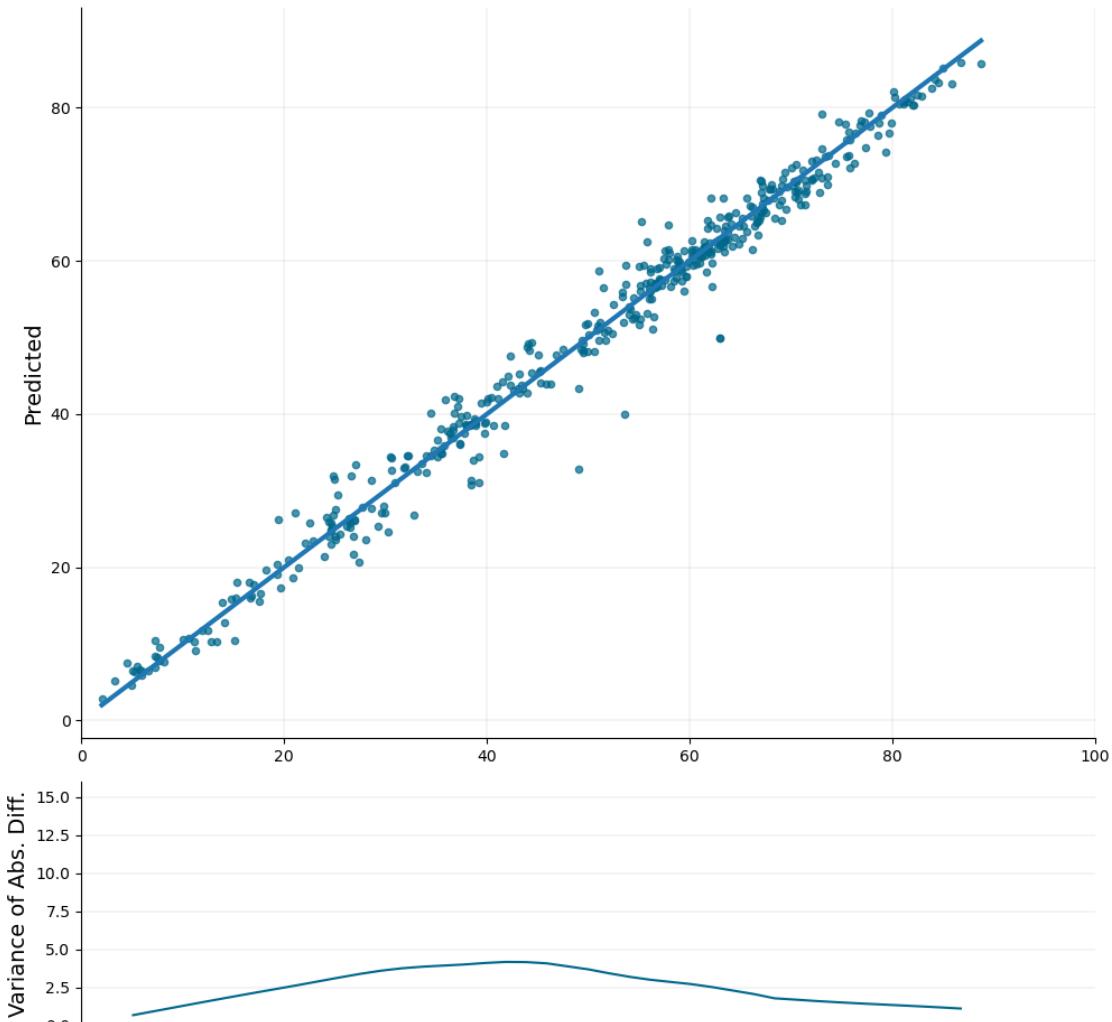
59	WB_CC_EST_avg	0.9524
61	CC_EST_prev	0.0138
25	SP_POP_0004_FE_5Y	0.0022
34	SP_POP_1519_FE_5Y	0.0013
19	SP_DYN_LEOO_FE_IN	0.0010
18	SP_DYN_IMRT_MA_IN	0.0010
14	SP_DYN_AMRT_MA	0.0009
37	SP_POP_4549_FE_5Y	0.0009
17	SP_DYN_IMRT_IN	0.0009
3	NV_AGR_TOTL_ZS	0.0008
54	SP_POP_80UP_FE_5Y	0.0008
22	SP_DYN_TFRT_IN	0.0008
24	SP_DYN_T065_MA_ZS	0.0007
32	SP_POP_1014_FE_5Y	0.0007
44	SP_POP_6064_MA_5Y	0.0007
43	SP_POP_6064_FE_5Y	0.0007
23	SP_DYN_T065_FE_ZS	0.0007
45	SP_POP_6569_FE_5Y	0.0006
36	SP_POP_2024_FE_5Y	0.0006
39	SP_POP_5054_FE_5Y	0.0006
40	SP_POP_5054_MA_5Y	0.0006
46	SP_POP_6569_MA_5Y	0.0006
41	SP_POP_5559_FE_5Y	0.0006
49	SP_POP_65UP_TO_ZS	0.0006
56	SP_POP_DPND_DL	0.0006
33	SP_POP_1014_MA_5Y	0.0005
60	NY_GDP_PCAP_KD_rel	0.0005
4	NV_SRV_TOTL_ZS	0.0005
38	SP_POP_4549_MA_5Y	0.0005
58	TM_VAL_MRCH_HI_ZS	0.0005
35	SP_POP_1519_MA_5Y	0.0005
42	SP_POP_5559_MA_5Y	0.0005
1	FM_AST_PRVT_GD_ZS	0.0005
30	SP_POP_0509_FE_5Y	0.0005
29	SP_POP_0014_TO_ZS	0.0005
11	SH_DYN_NMRT	0.0005
13	SP_DYN_AMRT_FE	0.0005
52	SP_POP_7579_FE_5Y	0.0005
48	SP_POP_65UP_MA_ZS	0.0004
53	SP_POP_7579_MA_5Y	0.0004
51	SP_POP_7074_MA_5Y	0.0004
50	SP_POP_7074_FE_5Y	0.0004
0	FD_AST_PRVT_GD_ZS	0.0004
31	SP_POP_0509_MA_5Y	0.0004
7	NY_GNP_PCAP_CD	0.0004
21	SP_DYN_LEOO_MA_IN	0.0004
6	NY_GDP_PCAP_KD	0.0004
16	SP_DYN_IMRT_FE_IN	0.0004

15	SP_DYN_CBRT_IN	0.0004
12	SP_ADO_TFRT	0.0004
27	SP_POP_0014_FE_ZS	0.0004
9	SH_DYN_MORT_FE	0.0004
2	IT_MLT_MAIN_P2	0.0003
5	NY_GDP_PCAP_CD	0.0003
57	SP_POP_DPND_YG	0.0003
28	SP_POP_0014_MA_ZS	0.0003
55	SP_POP_80UP_MA_5Y	0.0003
8	SH_DYN_MORT	0.0003
20	SP_DYN_LEOO_IN	0.0003
47	SP_POP_65UP_FE_ZS	0.0003
26	SP_POP_0004_MA_5Y	0.0003
10	SH_DYN_MORT_MA	0.0002

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
...
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

[2249 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.5



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
37it [00:00, 257.91it/s]
```

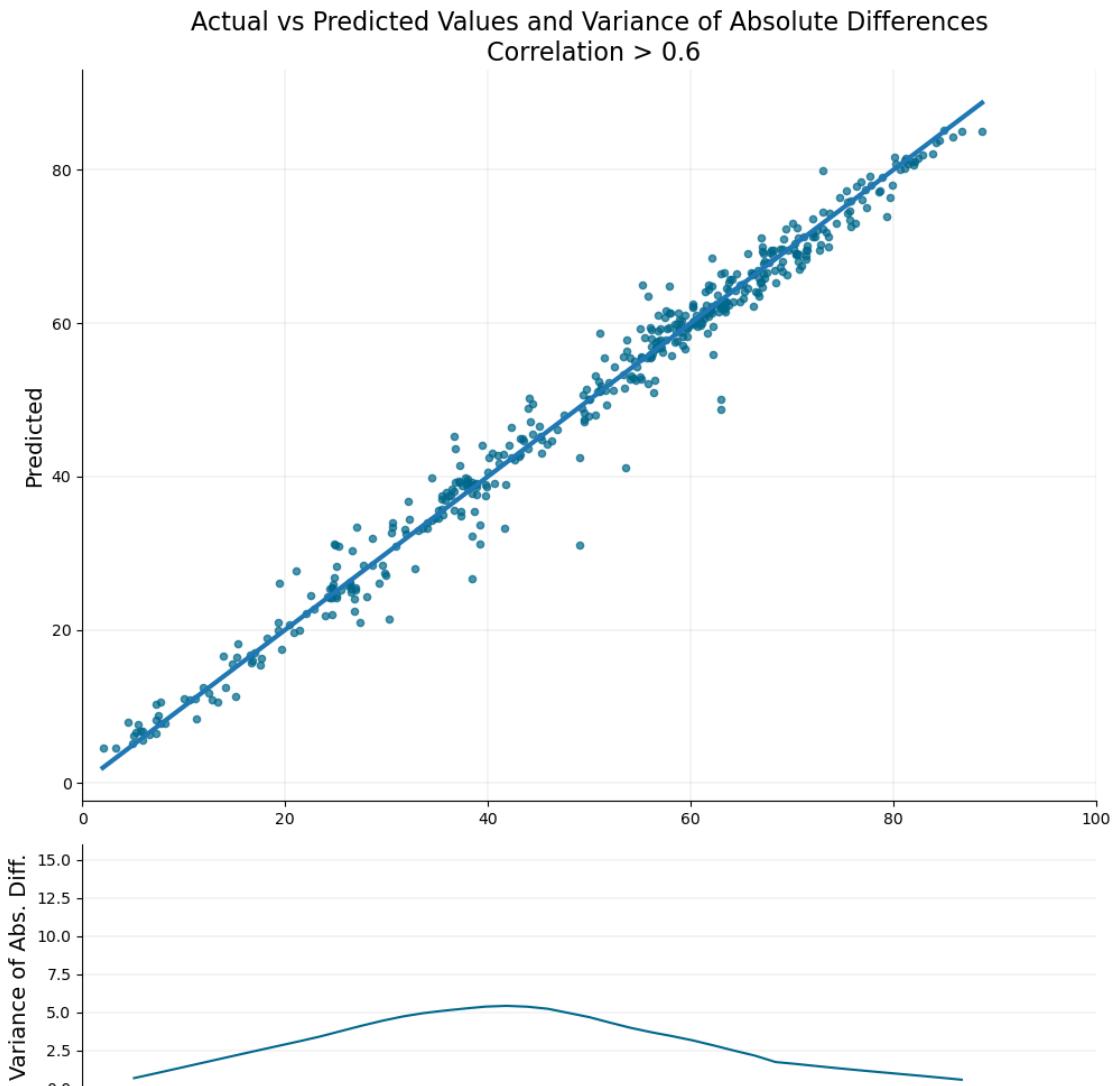
Model with correlation ≥ 0.6 :
 Training+Validation R²: 0.99576, RMSE: 1.29878
 Testing R²: 0.98007, RMSE: 2.84968
 Mean cross-validation score: 0.98152

Feature Importance

30	WB_CC_EST_avg	0.9610
32	CC_EST_prev	0.0176
10	SP_POP_0004_FE_5Y	0.0019
19	SP_POP_1519_FE_5Y	0.0013
9	SP_DYN_T065_MA_ZS	0.0011
28	SP_POP_7579_MA_5Y	0.0009
20	SP_POP_1519_MA_5Y	0.0009
5	SP_DYN_CBRT_IN	0.0009
31	NY_GDP_PCAP_KD_rel	0.0008
26	SP_POP_65UP_TO_ZS	0.0008
27	SP_POP_7074_MA_5Y	0.0007
11	SP_POP_0004_MA_5Y	0.0007
23	SP_POP_6064_MA_5Y	0.0007
29	SP_POP_DPND_YG	0.0007
6	SP_DYN_LE00_FE_IN	0.0007
15	SP_POP_0509_FE_5Y	0.0006
25	SP_POP_65UP_MA_ZS	0.0006
17	SP_POP_1014_FE_5Y	0.0006
24	SP_POP_6569_MA_5Y	0.0006
4	SH_DYN_NMRT	0.0006
2	NY_GDP_PCAP_KD	0.0006
21	SP_POP_5054_MA_5Y	0.0006
22	SP_POP_5559_MA_5Y	0.0006
0	IT_MLT_MAIN_P2	0.0005
12	SP_POP_0014_FE_ZS	0.0005
8	SP_DYN_LE00_MA_IN	0.0005
7	SP_DYN_LE00_IN	0.0005
3	NY_GNP_PCAP_CD	0.0005
16	SP_POP_0509_MA_5Y	0.0005
18	SP_POP_1014_MA_5Y	0.0004
14	SP_POP_0014_TO_ZS	0.0004
13	SP_POP_0014_MA_ZS	0.0004
1	NY_GDP_PCAP_CD	0.0003

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
...
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

[2249 rows x 3 columns]



```
Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is deprecated for better compatibility with scikit-learn, use `callbacks` in constructor or `set_params` instead.
```

```
    warnings.warn(  
28it [00:00, 407.54it/s]
```

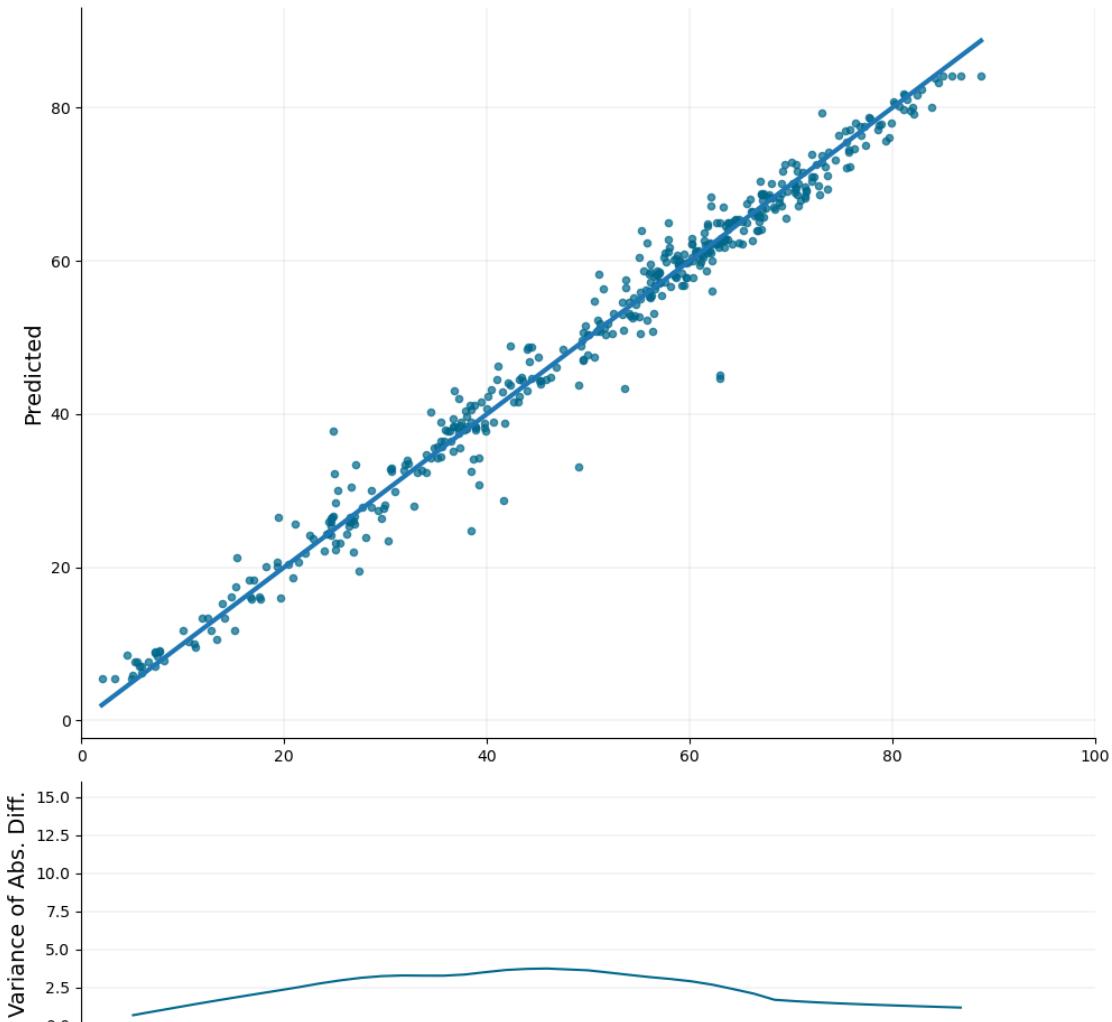
```
Model with correlation >= 0.7:  
Training+Validation R^2: 0.98944, RMSE: 2.04887  
Testing R^2: 0.97783, RMSE: 3.0056  
Mean cross-validation score: 0.98092
```

	Feature	Importance
1	WB_CC_EST_avg	0.9738
2	CC_EST_prev	0.0243
0	NY_GNP_PCAP_CD	0.0019

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
...
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

[2249 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.7



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
38it [00:00, 156.68it/s]
```

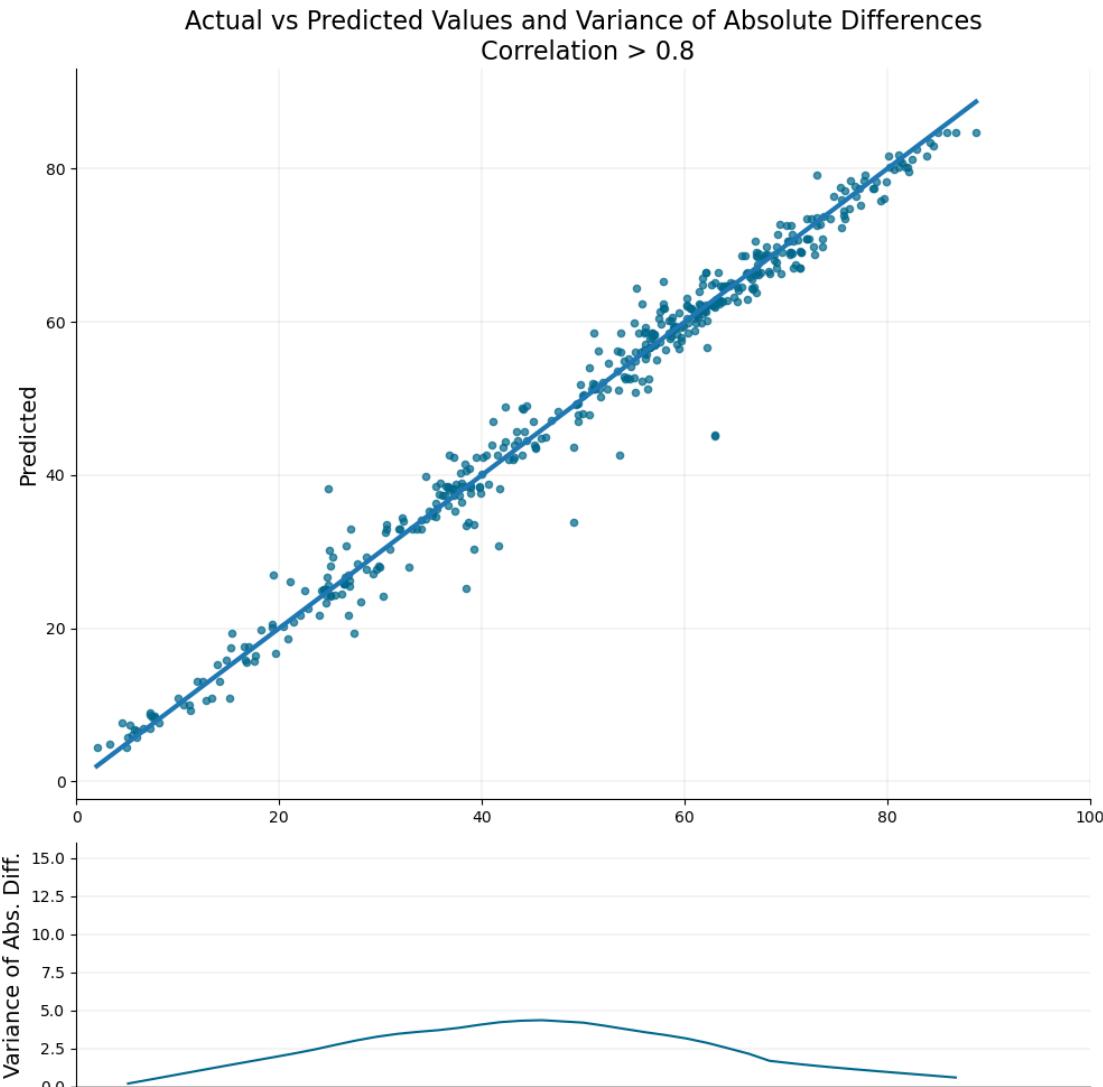
Model with correlation ≥ 0.8 :
 Training+Validation R²: 0.98902, RMSE: 2.08932
 Testing R²: 0.97892, RMSE: 2.93073
 Mean cross-validation score: 0.98086

Feature	Importance

```
0   WB_CC_EST_avg      0.9679
1   CC_EST_prev        0.0321
```

```
      country  year    CC_EST
52   Andorra  2012  24.789383
53   Andorra  2013  24.929030
54   Andorra  2014  25.585828
55   Andorra  2015  26.963785
56   Andorra  2016  26.808882
...
13729  Zimbabwe  2018  74.920013
13730  Zimbabwe  2019  75.423806
13731  Zimbabwe  2020  75.759847
13732  Zimbabwe  2021  75.071001
13733  Zimbabwe  2022  75.102789
```

[2249 rows x 3 columns]



```
[190]: ### TESTS FOR SELECTED VARIABLES -- The core variables that were used in the study

# Set the style to 'default' to make the background white
style.use('default')

# Manually selected variables
vars_selected = ['NY_GDP_PCAP_KD', 'SL_UEM_TOTL_NE_ZS', 'NE_CON_GOVT_CD', 'FP_CPI_TOTL_ZG', 'SE_SEC_DURS', 'SI_POV_GINI', 'BX_KLT_DINV_CD_WD']
```

```

# vars_selected = ['CC_EST_prev', 'WB_CC_EST_avg', 'NY_GDP_PCAP_KD', □
↳ 'NY_GDP_PCAP_CD', 'SL_UEM_TOTL_NE_ZE', 'NE_CON_GOVT_CD', 'FP_CPI_TOTL_ZG', □
↳ 'HD_HCI_LAYS', 'SI_POV_GINI', 'BX_KLT_DINV_CD_WD']

if 'CC_EST_prev' in vars_selected:
    vars_selected.remove('CC_EST_prev')

results = build_and_evaluate_model(df, 'CC_EST', vars_selected, n_leads=2)

print(f"\n Model with selected variables:")
print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
print(f"Mean cross-validation score: {results[4]}\n")
print(results[5]) # Feature importances
print('\n')
print(results[9][['country', 'year', 'CC_EST']])

# Generate the graphs
y_test = results[8]
y_test_pred = results[7]

# Create a scatter plot for the actual vs predicted values
abs_diffs = np.abs(y_test - y_test_pred)

# Create a DataFrame with the actual values and absolute differences
df_plot = pd.DataFrame({'Actual': y_test, 'AbsDifference': abs_diffs})

# Define the bins for the actual values
bins = np.linspace(0, 100, 50)

# Calculate the mid-points of the bins
bin_midpoints = bins[:-1] + np.diff(bins) / 2

# Create a new column for the binned actual values
df_plot['ActualBin'] = pd.cut(df_plot['Actual'], bins, labels=bin_midpoints)

# Group by the binned actual values and calculate the variance of the absolute
# differences for each group
var_abs_diffs = df_plot.groupby('ActualBin')['AbsDifference'].var()

# Create a scatter plot for the actual vs predicted values
fig = plt.figure(figsize=(10, 10))
gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1])
ax0 = plt.subplot(gs[0])
ax0.scatter(y_test, y_test_pred, alpha=0.7, color='#00688B', s=20)
ax0.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], lw=3)
ax0.set_xlim([0, 100]) # Set the x-axis range

```

```

ax0.set_ylabel('Predicted', fontsize=14)
ax0.set_title('Actual vs Predicted Values and Variance of Absolute  

    ↵Differences', fontsize=16)
ax0.grid(True, color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

# Create a line plot for the binned actual values vs variance of the absolute  

    ↵differences
ax1 = plt.subplot(gs[1])

# Apply LOESS to smooth the variance curve
smoothed = lowess(var_abs_diffs, var_abs_diffs.index, frac=0.5)
index, data = zip(*smoothed)
ax1.plot(index, data, color='#00688B') # DeepSkyBlue4 color

ax1.set_ylim([0, 15])
ax1.set_xlim([0, 100])
ax1.set_ylabel('Variance of Abs. Diff.', fontsize=14)
ax1.grid(axis='y', color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

ax1.set_xticklabels([])
ax1.set_xticks([])
ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)
ax0.spines['right'].set_visible(False)
ax0.spines['top'].set_visible(False)

plt.tight_layout()
plt.show()

```

0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is deprecated for better compatibility with scikit-learn, use `callbacks` in constructor or `set_params` instead.

```

warnings.warn(
37it [00:00, 440.75it/s]

```

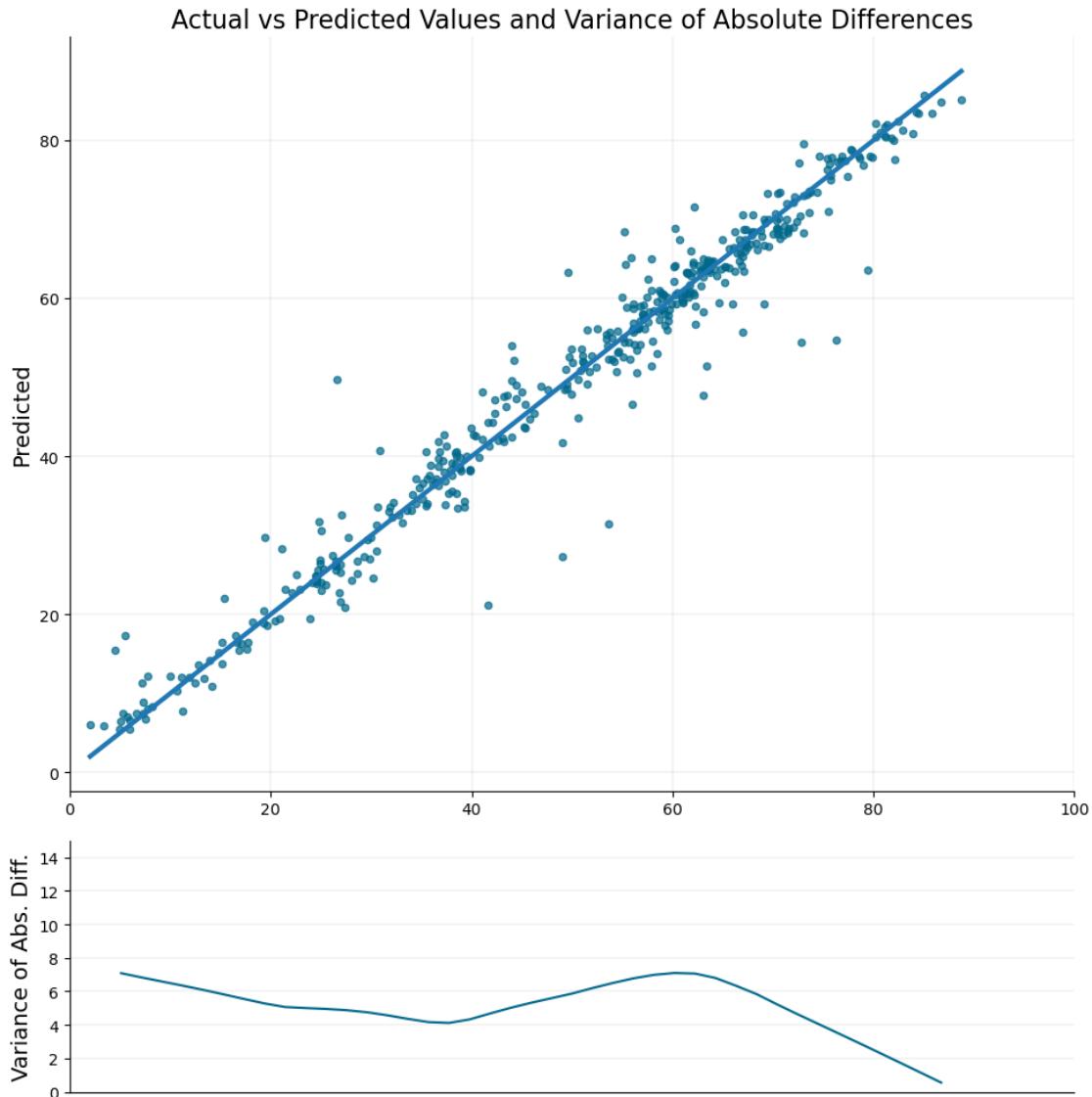
Model with selected variables:
Training+Validation R^2: 0.9857, RMSE: 2.38414
Testing R^2: 0.95934, RMSE: 4.06987
Mean cross-validation score: 0.95222

	Feature	Importance
7	CC_EST_prev	0.8528
0	NY_GDP_PCAP_KD	0.0998
4	SE_SEC_DURS	0.0094
6	BX_KLT_DINV_CD_WD	0.0094

```
3     FP_CPI_TOTL_ZG      0.0086
2     NE_CON_GOV_CD       0.0077
5     SI_POV_GINI        0.0066
1     SL_UEM_TOTL_NE_ZS    0.0057
```

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
..
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

[2249 rows x 3 columns]



[191]: *### TESTS FOR SELECTED VARIABLES*

```
# Set the style to 'default' to make the background white
style.use('default')

# Manually selected variables
# vars_selected = ['NY_GDP_PCAP_KD', 'SL_UEM_TOTL_NE_ZS', 'NE_CON_GOVT_CD', 'FP_CPI_TOTL_ZG', 'SE_SEC_DURS', 'SI_POV_GINI', 'BX_KLT_DINV_CD_WD']
```

```

vars_selected = ['CC_EST_prev', 'WB_CC_EST_avg', 'NY_GDP_PCAP_KD', □
↳ 'SL_UEM_TOTL_NE_ZS', 'NE_CON_GOVT_CD', 'FP_CPI_TOTL_ZG', 'SE_SEC_DURS', □
↳ 'SI_POV_GINI', 'BX_KLT_DINV_CD_WD']

if 'CC_EST_prev' in vars_selected:
    vars_selected.remove('CC_EST_prev')

results = build_and_evaluate_model(df, 'CC_EST', vars_selected, n_leads=2)

print(f"\n Model with selected variables:")
print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
print(f"Mean cross-validation score: {results[4]}\n")
print(results[5]) # Feature importances
print('\n')
print(results[9][['country', 'year', 'CC_EST']])

# Generate the graphs
y_test = results[8]
y_test_pred = results[7]

# Create a scatter plot for the actual vs predicted values
abs_diffs = np.abs(y_test - y_test_pred)

# Create a DataFrame with the actual values and absolute differences
df_plot = pd.DataFrame({'Actual': y_test, 'AbsDifference': abs_diffs})

# Define the bins for the actual values
bins = np.linspace(0, 100, 50)

# Calculate the mid-points of the bins
bin_midpoints = bins[:-1] + np.diff(bins) / 2

# Create a new column for the binned actual values
df_plot['ActualBin'] = pd.cut(df_plot['Actual'], bins, labels=bin_midpoints)

# Group by the binned actual values and calculate the variance of the absolute
# differences for each group
var_abs_diffs = df_plot.groupby('ActualBin')['AbsDifference'].var()

# Create a scatter plot for the actual vs predicted values
fig = plt.figure(figsize=(10, 10))
gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1])
ax0 = plt.subplot(gs[0])
ax0.scatter(y_test, y_test_pred, alpha=0.7, color='#00688B', s=20)
ax0.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], lw=3)
ax0.set_xlim([0, 100]) # Set the x-axis range

```

```

ax0.set_ylabel('Predicted', fontsize=14)
ax0.set_title('Actual vs Predicted Values and Variance of Absolute  

    ↵Differences', fontsize=16)
ax0.grid(True, color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

# Create a line plot for the binned actual values vs variance of the absolute  

    ↵differences
ax1 = plt.subplot(gs[1])

# Apply LOESS to smooth the variance curve
smoothed = lowess(var_abs_diffs, var_abs_diffs.index, frac=0.5)
index, data = zip(*smoothed)
ax1.plot(index, data, color='#00688B') # DeepSkyBlue4 color

ax1.set_ylim([0, 15])
ax1.set_xlim([0, 100])
ax1.set_ylabel('Variance of Abs. Diff.', fontsize=14)
ax1.grid(axis='y', color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

ax1.set_xticklabels([])
ax1.set_xticks([])
ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)
ax0.spines['right'].set_visible(False)
ax0.spines['top'].set_visible(False)

plt.tight_layout()
plt.show()

```

0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is deprecated for better compatibility with scikit-learn, use `callbacks` in constructor or `set_params` instead.

 warnings.warn(

34it [00:00, 422.24it/s]

Model with selected variables:

Training+Validation R^2: 0.99312, RMSE: 1.65399

Testing R^2: 0.97812, RMSE: 2.98524

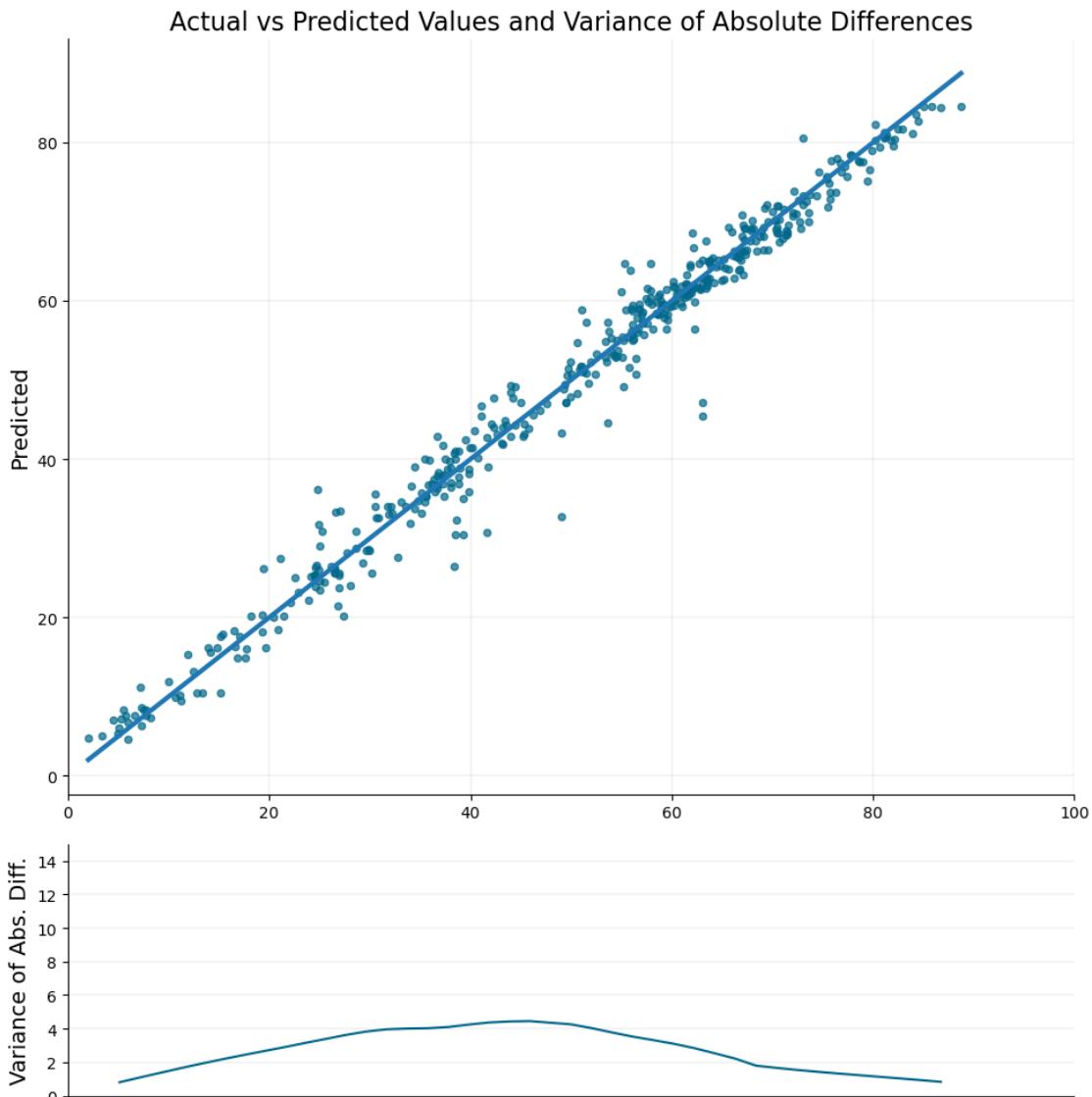
Mean cross-validation score: 0.98046

	Feature	Importance
0	WB_CC_EST_avg	0.9619
8	CC_EST_prev	0.0265
5	SE_SEC_DURS	0.0025
1	NY_GDP_PCAP_KD	0.0019

```
3      NE_CON_GOV_CD      0.0017
6      SI_POV_GINI      0.0015
7  BX_KLT_DINV_CD_WD      0.0015
4      FP_CPI_TOTL_ZG      0.0013
2  SL_UEM_TOTL_NE_ZS      0.0011
```

	country	year	CC_EST
52	Andorra	2012	24.789383
53	Andorra	2013	24.929030
54	Andorra	2014	25.585828
55	Andorra	2015	26.963785
56	Andorra	2016	26.808882
...
13729	Zimbabwe	2018	74.920013
13730	Zimbabwe	2019	75.423806
13731	Zimbabwe	2020	75.759847
13732	Zimbabwe	2021	75.071001
13733	Zimbabwe	2022	75.102789

[2249 rows x 3 columns]



```
[192]: ## TAKES c.4 MINUTES TO RUN
### TESTS FOR ALL VARIABLES
```

```
# Get the unique country codes
countries = df['iso2c'].unique()

# Create a new dataframe to store the results
df_new = pd.DataFrame()

# Get the variables for the model
```

```

vars_full = df.columns.tolist()
vars_full.remove('iso3c')
vars_full.remove('iso3n')
vars_full.remove('year')

# Remove 'CC_EST_prev' from vars_full if it's already included
if 'CC_EST_prev' in vars_full:
    vars_full.remove('CC_EST_prev')

# Train the model before getting the feature names
results = build_and_evaluate_model(df, 'CC_EST', vars_full)

# Get the trained model from the results
model = results[10]

# Get the feature names the model was trained on
model_features = model.get_booster().feature_names

# Iterate over each country
for country in tqdm(countries):
    # Get the data for the current country and create a copy of it
    df_country = df[df['iso2c'] == country].copy()

    # Add a new feature for the previous year's CC_EST value
    df_country['CC_EST_prev'] = df_country.groupby('iso2c')['CC_EST'].shift()

    # Sort the data in descending order of the year
    df_country = df_country.sort_values('year', ascending=False)

    # Get the latest year in the data
    latest_year = df_country['year'].max()

    # Skip the current country if all its 'year' values are NaN
    if np.isnan(latest_year):
        continue

    latest_year = int(latest_year)

    # Iterate from the latest year to 1950
    for year in range(latest_year, 1949, -1):
        # Check if CC_EST for the current year is NaN
        if df_country.loc[df_country['year'] == year, 'CC_EST'].isna().any():
            # If it is NaN, use the model to predict CC_EST for that year based
            # on other values
            X = df_country.loc[df_country['year'] == year, model_features]

            # Forward fill missing values in X

```

```

X = X.ffill()

y_pred = model.predict(X)

# Update the data with the predicted value
df_country.loc[df_country['year'] == year, 'CC_EST'] = y_pred

# Append the data for the current country to the new dataframe
df_new = pd.concat([df_new, df_country])

```

Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is deprecated for better compatibility with scikit-learn, use `callbacks` in constructor or `set_params` instead.

```

warnings.warn(
60it [00:01, 38.04it/s]
100%| 218/218 [02:12<00:00, 1.65it/s]

```

[193]: # print the results of the build and evaluate model:

```

print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
print(f"Mean cross-validation score: {results[4]}\n")
print(results[5]) # Feature importances

```

```

Training+Validation R^2: 0.99941, RMSE: 0.4837
Testing R^2: 0.98036, RMSE: 2.82851
Mean cross-validation score: 0.98041

```

	Feature	Importance
303	WB_CC_EST_avg	0.8595
306	CC_EST_prev	0.0151
145	NY_GDP_PCAP_CN	0.0093
205	SP_POP_0014_FE_ZS	0.0075
206	SP_POP_0014_MA_IN	0.0060
..
140	NY_GDP_MKTP_KD	0.0000
142	NY_GDP_MKTP_KN	0.0000
144	NY_GDP_PCAP_CD	0.0000
50	ER_FSH_PROD_MT	0.0000
0	AG_AGR_TRAC_NO	0.0000

[307 rows x 2 columns]

[196]: # Give a relative rank of CC_EST for each country in each year

```

df_new['CC_EST_rank'] = df_new.groupby('year')['CC_EST'].rank(pct=True)

# Give a rank of CC_EST for each country in each year as integers

```

```
df_new['CC_EST_rank_int'] = df_new.groupby('year')['CC_EST'].
    rank(method='dense', ascending=True)
```

	country	year	CC_EST	CC_EST_rank	CC_EST_rank_int
3149	Denmark	2022	1.945114	0.004608	1.0
3148	Denmark	2021	3.324947	0.004608	1.0
3147	Denmark	2020	5.270624	0.004608	1.0
3146	Denmark	2019	7.567654	0.009217	2.0
3145	Denmark	2018	6.870418	0.009217	2.0
3144	Denmark	2017	6.197357	0.013825	3.0
3143	Denmark	2016	5.942297	0.009217	2.0
3142	Denmark	2015	6.278753	0.018433	4.0
3141	Denmark	2014	4.953194	0.004608	1.0
3140	Denmark	2013	2.014718	0.004608	1.0
3139	Denmark	2012	2.350755	0.004608	1.0
3138	Denmark	2011	5.485703	0.013825	3.0
3137	Denmark	2010	5.430544	0.004608	1.0
3136	Denmark	2009	5.835021	0.013825	3.0
3135	Denmark	2008	5.498519	0.004608	1.0
3134	Denmark	2007	6.149212	0.009217	2.0
3133	Denmark	2006	6.926727	0.023041	5.0
3132	Denmark	2005	7.087546	0.013825	3.0
3131	Denmark	2004	7.756819	0.018433	4.0
3130	Denmark	2003	7.781825	0.023041	5.0
3129	Denmark	2002	7.914946	0.018433	4.0
3128	Denmark	2001	8.319429	0.023041	5.0
3127	Denmark	2000	7.880605	0.018433	4.0
3126	Denmark	1999	8.289908	0.023041	5.0
3125	Denmark	1998	7.178010	0.018433	4.0
3124	Denmark	1997	6.699611	0.013825	3.0
3123	Denmark	1996	6.931729	0.013825	3.0
3122	Denmark	1995	7.312321	0.023041	5.0
3121	Denmark	1994	6.874060	0.018433	4.0
3120	Denmark	1993	6.779128	0.009217	2.0
3119	Denmark	1992	6.920821	0.013825	3.0
3118	Denmark	1991	7.300401	0.023041	5.0
3117	Denmark	1990	7.093511	0.013825	3.0
3116	Denmark	1989	7.194421	0.013825	3.0
3115	Denmark	1988	7.495773	0.018433	4.0
3114	Denmark	1987	7.901844	0.018433	4.0
3113	Denmark	1986	7.387969	0.013825	3.0
3112	Denmark	1985	8.012049	0.013825	3.0
3111	Denmark	1984	9.776796	0.023041	5.0
3110	Denmark	1983	9.703144	0.013825	3.0
3109	Denmark	1982	9.305244	0.009217	2.0
3108	Denmark	1981	9.600747	0.013825	3.0
3107	Denmark	1980	9.514429	0.018433	4.0

3106	Denmark	1979	8.956038	0.009217	2.0
3105	Denmark	1978	9.096681	0.004608	1.0
3104	Denmark	1977	9.162324	0.004608	1.0
3103	Denmark	1976	8.471530	0.004608	1.0
3102	Denmark	1975	8.718865	0.009217	2.0
3101	Denmark	1974	6.277755	0.004608	1.0
3100	Denmark	1973	6.275676	0.009217	2.0
3099	Denmark	1972	6.205935	0.009217	2.0
3098	Denmark	1971	6.337743	0.004608	1.0
3097	Denmark	1970	8.983508	0.009217	2.0
3096	Denmark	1969	7.157137	0.009217	2.0
3095	Denmark	1968	7.059511	0.009217	2.0
3094	Denmark	1967	7.306514	0.013825	3.0
3093	Denmark	1966	7.042351	0.009217	2.0
3092	Denmark	1965	10.192755	0.013825	3.0
3091	Denmark	1964	8.139769	0.018433	4.0
3090	Denmark	1963	7.920730	0.013825	3.0
3089	Denmark	1962	8.104650	0.013825	3.0
3088	Denmark	1961	8.104650	0.013825	3.0
3087	Denmark	1960	10.912830	0.018433	4.0

```
[197]: # Append the new data of iso2c, year, CC_EST, CC_EST_rank, and CC_EST_rank_int
      ↪to a new csv file
df_new[['iso2c', 'year', 'CC_EST', 'CC_EST_rank', 'CC_EST_rank_int']].
      ↪to_csv('backcasted_corruption_data.csv', index=False)
```

```
[198]: # Print the mean, median, and range of predicted values for CC_EST in 1960,
      ↪with the countries that have the highest and lowest values
# Get the mean, median, and range of predicted values for CC_EST in 1960
mean_1960 = round(df_new[df_new['year'] == 1960]['CC_EST'].mean(), 3)
median_1960 = round(df_new[df_new['year'] == 1960]['CC_EST'].median(), 3)
range_1960 = round(df_new[df_new['year'] == 1960]['CC_EST'].max() -
      ↪df_new[df_new['year'] == 1960]['CC_EST'].min(), 3)

# Get the countries with the highest and lowest predicted values for CC_EST in
      ↪1960
highest_1960 = df_new[df_new['year'] == 1960].sort_values('CC_EST',
      ↪ascending=False).iloc[0]['country']
lowest_1960 = df_new[df_new['year'] == 1960].sort_values('CC_EST').
      ↪iloc[0]['country']

print(f"The mean predicted value for CC_EST in 1960 is {mean_1960}")
print(f"The median predicted value for CC_EST in 1960 is {median_1960}")
print(f"The range of predicted values for CC_EST in 1960 is {range_1960}")
print(f"The country with the highest predicted value for CC_EST in 1960 is
      ↪{highest_1960} with a value of {df_new[df_new['year'] == 1960].
      ↪sort_values('CC_EST', ascending=False).iloc[0]['CC_EST']}")
```

```

print(f"The country with the lowest predicted value for CC_EST in 1960 is "
      f"{lowest_1960} with a value of {df_new[df_new['year'] == 1960].
      sort_values('CC_EST').iloc[0]['CC_EST']}")
```

The mean predicted value for CC_EST in 1960 is 51.858
The median predicted value for CC_EST in 1960 is 55.928
The range of predicted values for CC_EST in 1960 is 63.989
The country with the highest predicted value for CC_EST in 1960 is South Sudan with a value of 73.01355743408203
The country with the lowest predicted value for CC_EST in 1960 is Sweden with a value of 9.024733543395996

```
[199]: # Print the country that has gotten worse by the most and the country that has improved by the most
# Calculate the change in CC_EST for each country between 1960 and 2022
df_change = df_new.pivot(index='iso2c', columns='year', values='CC_EST')
df_change['change'] = df_change[2012] - df_change[1970]

# Get the country that has gotten worse by the most and the country that has improved by the most
most_improved = df_change['change'].idxmax()
most_worsened = df_change['change'].idxmin()

print(f"The country that has improved the most between 1970 and 2012 is "
      f"{df_new[df_new['iso2c'] == most_improved].iloc[0]['country']} with a change"
      f"of {df_change.loc[most_improved, 'change']}")

print(f"The country that has worsened the most between 1970 and 2012 is "
      f"{df_new[df_new['iso2c'] == most_worsened].iloc[0]['country']} with a change"
      f"of {df_change.loc[most_worsened, 'change']}")

# List worst to best
# Create a mapping of country codes to country names
country_mapping = df_new.set_index('iso2c')['country'].drop_duplicates()

# Map the country codes in df_change to country names
df_change['country'] = df_change.index.map(country_mapping)
# Sort df_change by 'change' in descending order and display the top 10
df_change.sort_values('change', ascending=True)[['country', 'change']].head(10)
```

The country that has improved the most between 1970 and 2012 is Somalia with a change of 15.701358786367194
The country that has worsened the most between 1970 and 2012 is Barbados with a change of -30.04481506083008

```
[199]: year          country      change
iso2c
BB           Barbados   -30.044815
CL           Chile     -27.740088
```

BS	Bahamas, The	-25.729905
GL	Greenland	-21.831344
UY	Uruguay	-19.933293
QA	Qatar	-19.172745
KN	St. Kitts and Nevis	-18.669986
KY	Cayman Islands	-16.346680
FR	France	-15.648187
AG	Antigua and Barbuda	-15.518497

```
[200]: # Filter the DataFrame to include only the years between 1960 and 2012
df_filtered = df_new[(df_new['year'] >= 1960) & (df['year'] <= 2012)]

plt.figure(figsize=(20, 10))
sns.histplot(df_filtered['CC_EST'], bins=30, color='#00688B', edgecolor='#00688B', kde=True)

# calculate the means of the corruption estimates between 2012 and 2022
mean_cc_est_2012_2022 = df_new[(df_new['year'] >= 2012) & (df_new['year'] <= 2022)].groupby('iso2c')['CC_EST'].mean()
median_cc_est_2012_2022 = df_new[(df_new['year'] >= 2012) & (df_new['year'] <= 2022)].groupby('iso2c')['CC_EST'].median()

mean_2012_2022 = mean_cc_est_2012_2022.mean()
median_2012_2022 = mean_cc_est_2012_2022.median()

# Calculate the mean and median
mean = df_filtered['CC_EST'].mean()
median = df_filtered['CC_EST'].median()

# Add lines for the mean and median
plt.axvline(mean, color='#00688B', linestyle='--', label=f'Mean: {mean:.3f}')
plt.axvline(median, color='#00688B', linestyle='-', label=f'Median: {median:.3f}')
plt.axvline(mean_2012_2022, color='orange', linestyle='--', label=f'Mean (2012-2022): {mean_2012_2022:.3f}')
plt.axvline(median_2012_2022, color='orange', linestyle='-', label=f'Median (2012-2022): {median_2012_2022:.3f}')

# Customize the title and labels
plt.title('Mean Corruption Estimates (1960-2012)', fontsize=16)
plt.xlabel('Mean Corruption Estimate', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.xlim(0,100)

# Customize the grid
plt.grid(axis='y', linestyle='--', alpha=0.7)
```

```

plt.legend(frameon=False)

# Remove the top and right spines
sns.despine()

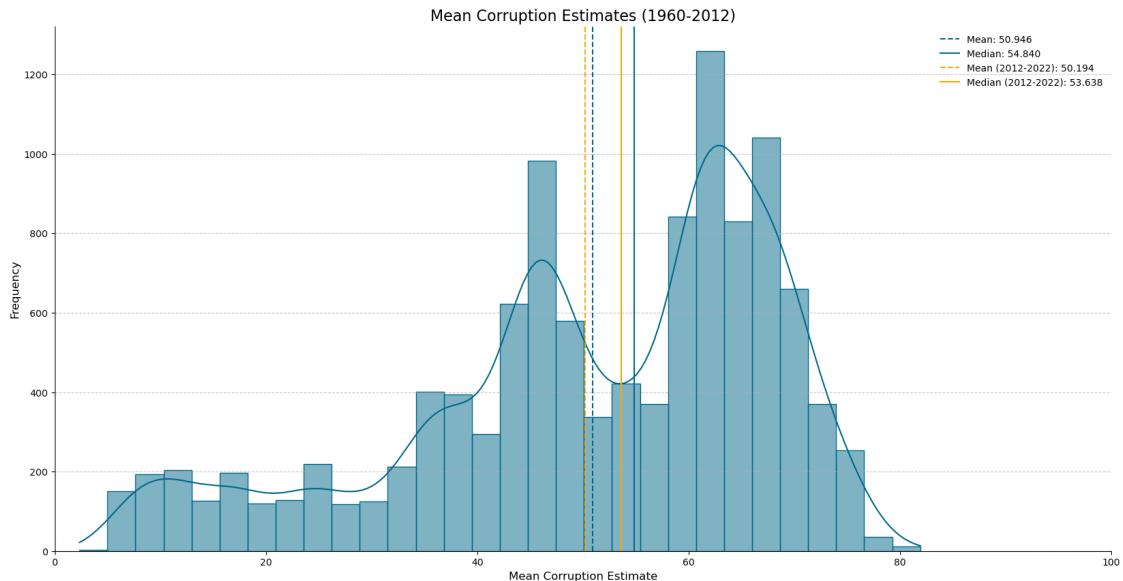
plt.show()

```

```

/var/folders/1z/rmhh3bk123qg9411_qfj8858000gn/T/ipykernel_37400/1212402169.py:2
: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
df_filtered = df_new[(df_new['year'] >= 1960) & (df['year'] <= 2012)]

```

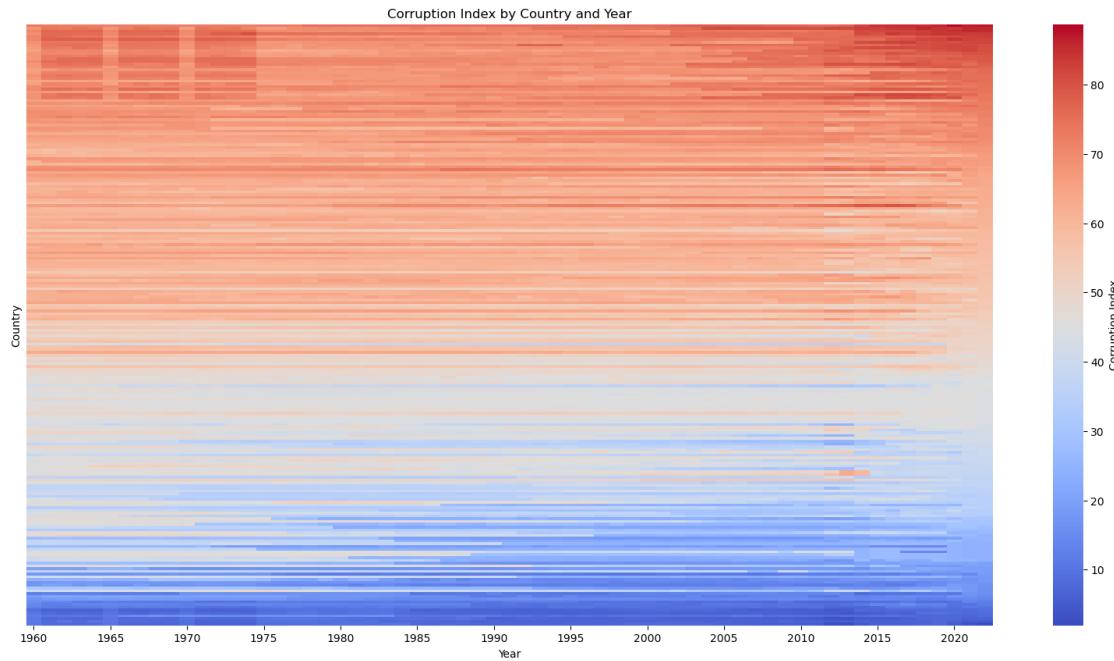


```

[201]: # Graphically illustrate the generale moves of country and corruption levels over time
# Create a pivot table of the data for the heatmap
df_heatmap = df_new.pivot(index='country', columns='year', values='CC_EST')
df_heatmap = df_heatmap.sort_values(by=2022, ascending=False)

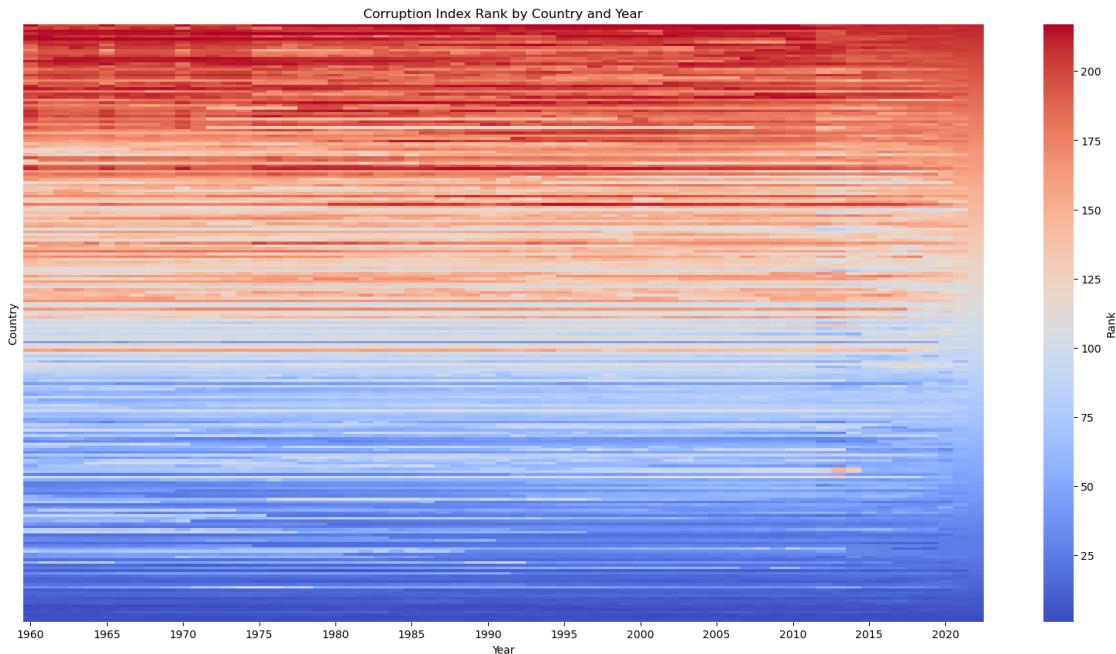
# Create a heatmap of the data, dont show country labels, and 1970 to 2022
plt.figure(figsize=(20, 10))
sns.heatmap(df_heatmap, cmap='coolwarm', cbar_kws={'label': 'Corruption Index'}, xticklabels=5, yticklabels=False)
plt.xlim(0, 63)
plt.title('Corruption Index by Country and Year')
plt.xlabel('Year')
plt.ylabel('Country')
plt.show()

```



```
[202]: # Plot changes in country ranks over time
# Create a pivot table of the data for the heatmap
df_heatmap = df_new.pivot(index='country', columns='year',
                           values='CC_EST_rank_int')
df_heatmap = df_heatmap.sort_values(by=2022, ascending=False)

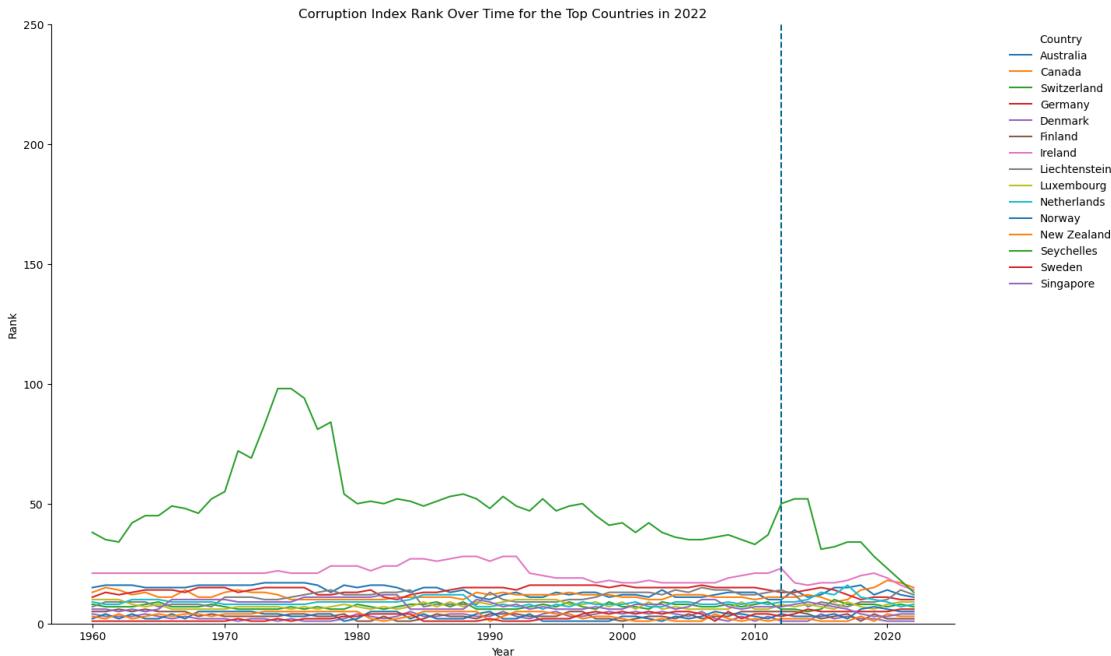
# Create a heatmap of the data, don't show country labels, and 1970 to 2022,
# ranking from lowest to highest in 2022
plt.figure(figsize=(20, 10))
sns.heatmap(df_heatmap, cmap='coolwarm', cbar_kws={'label': 'Rank'},
            xticklabels=5, yticklabels=False)
plt.xlim(0, 63)
plt.title('Corruption Index Rank by Country and Year')
plt.xlabel('Year')
plt.ylabel('Country')
plt.show()
```



```
[203]: # Plot graph of CC_EST_rank_int for the top 10 countries with the highest rank
      ↪in 2022
top_countries = df_heatmap[2022].sort_values(ascending=True).head(15).index

# Filter the data for the top 10 countries
df_top = df_new[df_new['country'].isin(top_countries)]

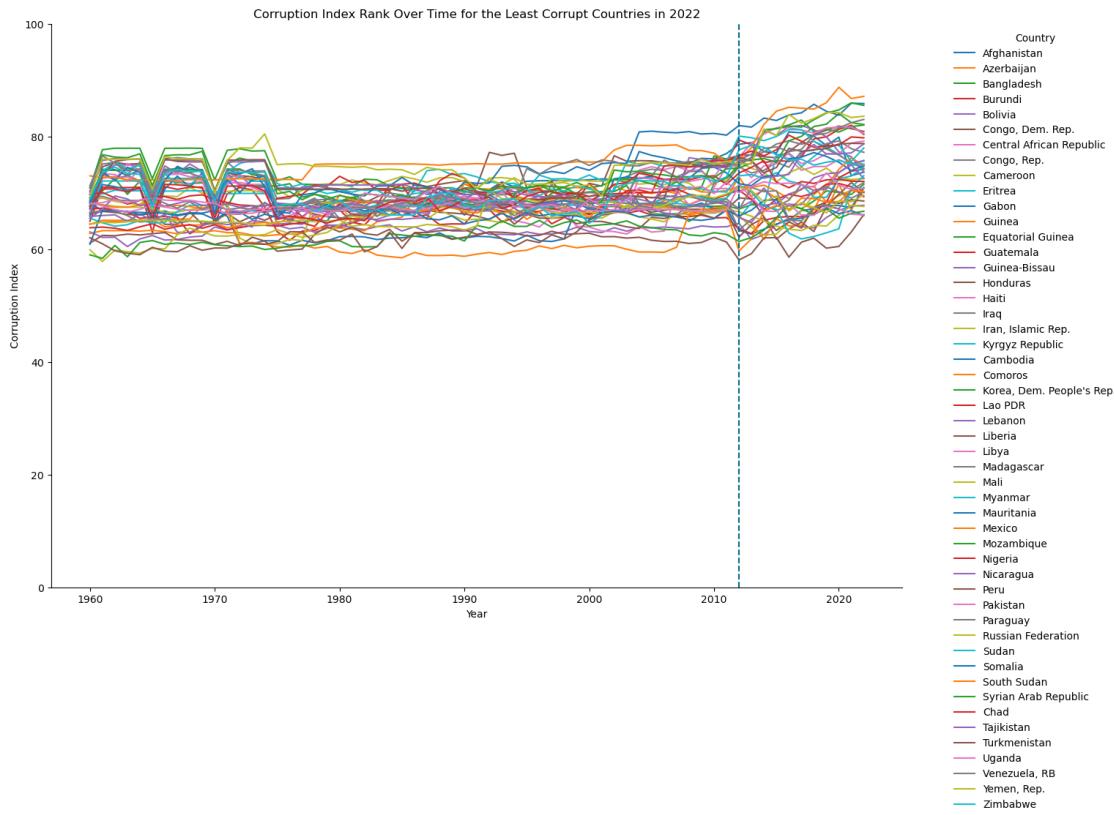
# Create a line plot of the rank over time for the top 10 countries
plt.figure(figsize=(15, 10))
sns.lineplot(data=df_top, x='year', y='CC_EST_rank_int', hue='country',
             ↪palette='tab10')
plt.axvline(x=2012, color="#00688B", linestyle='--')
plt.title('Corruption Index Rank Over Time for the Top Countries in 2022')
plt.ylim(0, 250)
plt.xlabel('Year')
plt.ylabel('Rank')
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left',
           ↪frameon=False)
sns.despine()
plt.show()
```



```
[204]: # Plot graph of CC_EST_rank_int for the top 10 countries with the highest rank in 2022
# Get the top 10 countries with the highest rank in 2022
top_countries = df_heatmap[2022].sort_values(ascending=False).head(50).index

# Filter the data for the top 10 countries
df_top = df_new[df_new['country'].isin(top_countries)]

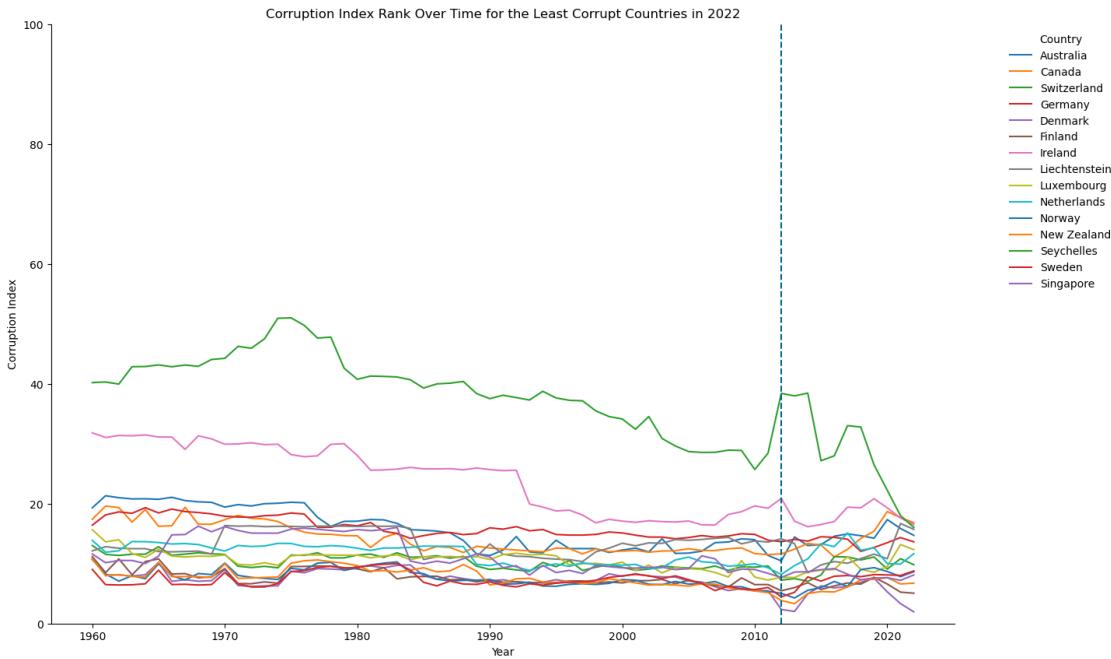
# Create a line plot of the rank over time for the top 10 countries
plt.figure(figsize=(15, 10))
sns.lineplot(data=df_top, x='year', y='CC_EST', hue='country', palette='tab10')
plt.axvline(x=2012, color='#00688B', linestyle='--')
plt.ylim(0, 100)
plt.title('Corruption Index Rank Over Time for the Least Corrupt Countries in 2022')
plt.xlabel('Year')
plt.ylabel('Corruption Index')
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left',
frameon=False)
sns.despine()
plt.show()
```



```
[205]: # Plot graph of CC_EST_rank_int for the top 10 countries with the highest rank in 2022
# Get the top 10 countries with the highest rank in 2022
top_countries = df_heatmap[2022].sort_values(ascending=True).head(15).index

# Filter the data for the top 10 countries
df_top = df_new[df_new['country'].isin(top_countries)]

# Create a line plot of the rank over time for the top 10 countries
plt.figure(figsize=(15, 10))
sns.lineplot(data=df_top, x='year', y='CC_EST', hue='country', palette='tab10')
plt.axvline(x=2012, color="#00688B", linestyle='--')
plt.ylim(0, 100)
plt.title('Corruption Index Rank Over Time for the Least Corrupt Countries in 2022')
plt.xlabel('Year')
plt.ylabel('Corruption Index')
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left', frameon=False)
sns.despine()
plt.show()
```



```
[206]: # create df_new_morphed that is df with cc_est replaced with CC_EST from df_new
df_new_morphed = df.copy()
df_new_morphed['CC_EST'] = df_new['CC_EST']

# Add columns for CC_EST_rank and CC_EST_rank_int
df_new_morphed['CC_EST_rank'] = df_new['CC_EST_rank']
df_new_morphed['CC_EST_rank_int'] = df_new['CC_EST_rank_int']

# Save the morphed data to a new CSV file
df_new_morphed.to_csv('df_model3.csv', index=False)
```

```
[207]: results[5]
```

	Feature	Importance
303	WB_CC_EST_avg	0.8595
306	CC_EST_prev	0.0151
145	NY_GDP_PCAP_CN	0.0093
205	SP_POP_0014_FE_ZS	0.0075
206	SP_POP_0014_MA_IN	0.0060
214	SP_POP_1519_FE_5Y	0.0024
121	NY_ADJ_DFOR_CD	0.0022
287	TM_VAL_MRCH_WL_CD	0.0021
158	NY_GNP_PCAP_CN	0.0020
246	SP_POP_65UP_TO	0.0019
156	NY_GNP_MKTP_CN	0.0018

134	NY_GDP_FCST_CD	0.0018
16	AG_SRF_TOTL_K2	0.0016
193	SP_DYN_IMRT_FE_IN	0.0015
196	SP_DYN_LEOO_FE_IN	0.0015
12	AG_PRD_CREL_MT	0.0015
253	SP_POP_80UP_MA_5Y	0.0015
154	NY_GNP_ATLS_CD	0.0014
73	IT_MLT_MAIN	0.0013
46	EN_URB_MCTY	0.0013
34	EN_ATM_CO2E_LF_KT	0.0013
157	NY_GNP_PCAP_CD	0.0013
268	SP_URB_TOTL	0.0011
81	NE_CON_PRVT_CD	0.0011
199	SP_DYN_TFRT_IN	0.0011
261	SP_POP_TOTL_FE_ZS	0.0011
276	TM_VAL_MRCH_AL_ZS	0.0010
304	NY_GDP_PCAP_KD_rel	0.0010
234	SP_POP_5054_FE_5Y	0.0009
198	SP_DYN_LEOO_MA_IN	0.0009
252	SP_POP_80UP_FE_5Y	0.0009
37	EN_ATM_CO2E_SF_ZS	0.0008
71	IT_CEL_SETS	0.0008
109	NV_AGR_TOTL_CN	0.0008
119	NY_ADJ_AEDU_CD	0.0008
54	FI_RES_TOTL_CD	0.0008
111	NV_AGR_TOTL_KN	0.0008
173	SE_SEC_DURS	0.0007
136	NY_GDP_FRST_RT_ZS	0.0007
113	NV_IND_TOTL_CD	0.0007
190	SP_DYN_AMRT_MA	0.0007
176	SE_SEC_ENRL_GC_FE_ZS	0.0007
107	NE_TRD_GNFS_ZS	0.0007
298	TX_VAL_MRCH_R4_ZS	0.0007
291	TX_VAL_MMTR_ZS_UN	0.0007
87	NE_DAB_TOTL_CD	0.0007
180	SH_DTH_NMRT	0.0006
184	SH_DYN_NMRT	0.0006
31	DT_ODA_ODAT_PC_ZS	0.0006
215	SP_POP_1519_MA_5Y	0.0006
240	SP_POP_6569_FE_5Y	0.0006
53	FD_AST_PRVT_GD_ZS	0.0006
128	NY_ADJ_DNGY_GN_ZS	0.0006
120	NY_ADJ_AEDU_GN_ZS	0.0005
92	NE_EXP_GNFS_ZS	0.0005
76	MS_MIL_XPND_CN	0.0005
105	NE_RSB_GNFS_CN	0.0005
187	SM_POP_NETM	0.0005

150	NY_GDP_TOTL_RT_ZS	0.0005
167	SE_PRM_ENRL	0.0005
170	SE_PRM_ENRR_FE	0.0005
185	SI_POV_GINI	0.0005
188	SP_ADO_TFRT	0.0005
197	SP_DYN_LE00_IN	0.0005
209	SP_POP_0014_TO_ZS	0.0005
232	SP_POP_4549_FE_5Y	0.0005
251	SP_POP_7579_MA_5Y	0.0005
271	TM_VAL_AGRI_ZS_UN	0.0005
290	TX_VAL_MANF_ZS_UN	0.0005
72	IT_CEL_SETS_P2	0.0005
153	NY_GDS_TOTL_ZS	0.0005
6	AG_LND_ARBL_ZS	0.0005
3	AG_LND_AGRI_ZS	0.0005
52	ER_H2O_INTR_PC	0.0005
32	EN_ATM_CO2E_GF_KT	0.0005
256	SP_POP_DPND_OL	0.0004
279	TM_VAL_MRCH_OR_ZS	0.0004
29	DT_ODA_ODAT_CD	0.0004
100	NE_GDI_TOTL_ZS	0.0004
285	TM_VAL_MRCH_R6_ZS	0.0004
283	TM_VAL_MRCH_R4_ZS	0.0004
178	SH_DTH_IMRT	0.0004
30	DT_ODA_ODAT_KD	0.0004
278	TM_VAL_MRCH_HI_ZS	0.0004
201	SP_DYN_TO65_MA_ZS	0.0004
192	SP_DYN_CDRT_IN	0.0004
126	NY_ADJ_DMIN_GN_ZS	0.0004
38	EN_ATM_GHGO_KT_CE	0.0004
274	TM_VAL_MANF_ZS_UN	0.0004
273	TM_VAL_FUEL_ZS_UN	0.0004
58	FM_AST_NFRG_CN	0.0004
36	EN_ATM_CO2E_SF_KT	0.0004
26	DTNFL_UNDP_CD	0.0004
35	EN_ATM_CO2E_LF_ZS	0.0004
296	TX_VAL_MRCH_R1_ZS	0.0004
244	SP_POP_65UP_MA_IN	0.0004
90	NE_EXP_GNFS_CD	0.0004
74	IT_MLT_MAIN_P2	0.0004
75	MS_MIL_XPND_CD	0.0004
48	ER_FSH_AQUA_MT	0.0004
302	TX_VAL_MRCH_WL_CD	0.0004
65	FP_CPI_TOTL	0.0004
203	SP_POP_0004_MA_5Y	0.0004
207	SP_POP_0014_MA_ZS	0.0004
164	SE_ENR_PRIM_FM_ZS	0.0004

91	NE_EXP_GNFS_CN	0.0004
49	ER_FSH_CAPT_MT	0.0003
227	SP_POP_3034_MA_5Y	0.0003
181	SH_DYN_MORT	0.0003
228	SP_POP_3539_FE_5Y	0.0003
171	SE_PRM_ENRR_MA	0.0003
249	SP_POP_7074_MA_5Y	0.0003
229	SP_POP_3539_MA_5Y	0.0003
148	NY_GDP_PCAP_KN	0.0003
132	NY_GDP_DEFL_ZS	0.0003
147	NY_GDP_PCAP_KD_ZG	0.0003
236	SP_POP_5559_FE_5Y	0.0003
141	NY_GDP_MKTP_KD_ZG	0.0003
230	SP_POP_4044_FE_5Y	0.0003
20	BX_KLT_DINV_CD_WD	0.0003
270	TG_VAL_TOTL_GD_ZS	0.0003
23	DC_DAC_GBRL_CD	0.0003
114	NV_IND_TOTL_CN	0.0003
64	FM_LBL_BMNY_ZG	0.0003
62	FM_LBL_BMNY_GD_ZS	0.0003
293	TX_VAL_MRCH_CD_WT	0.0003
295	TX_VAL_MRCH_OR_ZS	0.0003
66	FP_CPI_TOTL_ZG	0.0003
288	TX_VAL_AGRI_ZS_UN	0.0003
223	SP_POP_2024_MA_5Y	0.0003
281	TM_VAL_MRCH_R2_ZS	0.0003
282	TM_VAL_MRCH_R3_ZS	0.0003
8	AG_LND_CROP_ZS	0.0003
77	MS_MIL_XPND_GD_ZS	0.0003
45	EN_URB_LCTY_UR_ZS	0.0003
115	NV_IND_TOTL_ZS	0.0003
163	PA_NUS_FCRF	0.0002
238	SP_POP_6064_FE_5Y	0.0002
213	SP_POP_1014_MA_5Y	0.0002
18	BX_GRT_EXTA_CD_WD	0.0002
22	DC_DAC_DEUL_CD	0.0002
21	BX_KLT_DINV_WD_GD_ZS	0.0002
212	SP_POP_1014_FE_5Y	0.0002
239	SP_POP_6064_MA_5Y	0.0002
186	SL_UEM_TOTL_NE_ZS	0.0002
237	SP_POP_5559_MA_5Y	0.0002
169	SE_PRM_ENRR	0.0002
216	SP_POP_1564_FE_IN	0.0002
226	SP_POP_3034_FE_5Y	0.0002
27	DT_ODA_ALLD_CD	0.0002
19	BX_GRT_TECH_CD_WD	0.0002
174	SE_SEC_ENRL	0.0002

224	SP_POP_2529_FE_5Y	0.0002
225	SP_POP_2529_MA_5Y	0.0002
182	SH_DYN_MORT_FE	0.0002
69	IS_AIR_GOOD_MT_K1	0.0002
44	EN_URB_LCTY	0.0002
70	IS_AIR_PSGR	0.0002
125	NY_ADJ_DMIN_CD	0.0002
124	NY_ADJ_DKAP_GN_ZS	0.0002
133	NY_GDP_DISC_CN	0.0002
265	SP_RUR_TOTL_ZG	0.0002
129	NY_ADJ_DRES_GN_ZS	0.0002
55	FI_RES_XGLD_CD	0.0002
267	SP_URB_GROW	0.0002
17	AG_YLD_CREL_KG	0.0002
272	TM_VAL_FOOD_ZS_UN	0.0002
258	SP_POP_GROW	0.0002
56	FM_AST_CGOV_ZG_M3	0.0002
57	FM_AST_DOMS_CN	0.0002
275	TM_VAL_MMTL_ZS_UN	0.0002
112	NV_AGR_TOTL_ZS	0.0002
60	FM_AST_PRVT_ZG_M3	0.0002
280	TM_VAL_MRCH_R1_ZS	0.0002
97	NE_GDI_STKB_CN	0.0002
260	SP_POP_TOTL_FE_IN	0.0002
146	NY_GDP_PCAP_KD	0.0002
68	IS_AIR_DPRT	0.0002
257	SP_POP_DPND_YG	0.0002
89	NE_DAB_TOTL_ZS	0.0002
211	SP_POP_0509_MA_5Y	0.0002
13	AG_PRD_CROP_XD	0.0002
14	AG_PRD_FOOD_XD	0.0002
15	AG_PRD_LVSK_XD	0.0002
86	NE_CON_TOTL_ZS	0.0002
137	NY_GDP_MINR_RT_ZS	0.0002
151	NY_GDS_TOTL_CD	0.0002
294	TX_VAL_MRCH_HI_ZS	0.0002
297	TX_VAL_MRCH_R3_ZS	0.0002
299	TX_VAL_MRCH_R5_ZS	0.0002
301	TX_VAL_MRCH_RS_ZS	0.0002
289	TX_VAL_FOOD_ZS_UN	0.0001
292	TX_VAL_MRCH_AL_ZS	0.0001
5	AG_LND_ARBL_HA_PC	0.0001
217	SP_POP_1564_FE_ZS	0.0001
219	SP_POP_1564_MA_ZS	0.0001
245	SP_POP_65UP_MA_ZS	0.0001
7	AG_LND_CREL_HA	0.0001
286	TM_VAL_MRCH_RS_ZS	0.0001

277	TM_VAL_MRCH_CD_WT	0.0001
266	SP_RUR_TOTL_ZS	0.0001
231	SP_POP_4044_MA_5Y	0.0001
235	SP_POP_5054_MA_5Y	0.0001
250	SP_POP_7579_FE_5Y	0.0001
248	SP_POP_7074_FE_5Y	0.0001
247	SP_POP_65UP_TO_ZS	0.0001
241	SP_POP_6569_MA_5Y	0.0001
242	SP_POP_65UP_FE_IN	0.0001
243	SP_POP_65UP_FE_ZS	0.0001
254	SP_POP_BRTH_MF	0.0001
189	SP_DYN_AMRT_FE	0.0001
210	SP_POP_0509_FE_5Y	0.0001
84	NE_CON_TOTL_CD	0.0001
168	SE_PRM_ENRL_FE_ZS	0.0001
43	EN_POP_DNST	0.0001
162	PA_NUS_ATLS	0.0001
96	NE_GDI_STKB_CD	0.0001
160	NY_GSR_NFCY_CN	0.0001
159	NY_GSR_NFCY_CD	0.0001
1	AG_CON_FERT_ZS	0.0001
152	NY_GDS_TOTL_CN	0.0001
149	NY_GDP_PETR_RT_ZS	0.0001
143	NY_GDP_NGAS_RT_ZS	0.0001
103	NE_IMP_GNFS_ZS	0.0001
104	NE_RSB_GNFS_CD	0.0001
63	FM_LBL_BMNY_IR_ZS	0.0001
106	NE_RSB_GNFS_ZS	0.0001
61	FM_LBL_BMNY_CN	0.0001
110	NV_AGR_TOTL_KD	0.0001
59	FM_AST_PRVT_GD_ZS	0.0001
118	NV_SRV_TOTL_ZS	0.0001
131	NY_GDP_DEF_CD_ZG	0.0001
85	NE_CON_TOTL_CN	0.0001
95	NE_GDI_FTOT_ZS	0.0001
175	SE_SEC_ENRL_GC	0.0001
191	SP_DYN_CBRT_IN	0.0001
24	DC_DAC_JPNL_CD	0.0001
25	DC_DAC_TOTL_CD	0.0001
202	SP_POP_0004_FE_5Y	0.0001
200	SP_DYN_TO65_FE_ZS	0.0001
28	DT_ODA_ALLD_KD	0.0001
67	FS_AST_CGOV_GD_ZS	0.0001
78	NE_CON_GOVT_CD	0.0001
195	SP_DYN_IMRT_MA_IN	0.0001
123	NY_ADJ_DKAP_CD	0.0001
33	EN_ATM_CO2E_GF_ZS	0.0001

80	NE_CON_GOVT_ZS	0.0001
82	NE_CON_PRVT_CN	0.0001
177	SG_LAW_INDX	0.0001
98	NE_GDI_TOTL_CD	0.0000
94	NE_GDI_FTOT_CN	0.0000
9	AG_LND_PRCP_MM	0.0000
122	NY_ADJ_DFOR_GN_ZS	0.0000
93	NE_GDI_FTOT_CD	0.0000
83	NE_CON_PRVT_ZS	0.0000
117	NV_SRV_TOTL_CN	0.0000
116	NV_SRV_TOTL_CD	0.0000
305	NY_GDP_PCAP_CD_rel	0.0000
2	AG_LND_AGRI_K2	0.0000
108	NV_AGR_TOTL_CD	0.0000
99	NE_GDI_TOTL_CN	0.0000
269	SP_URB_TOTL_IN_ZS	0.0000
4	AG_LND_ARBL_HA	0.0000
284	TM_VAL_MRCH_R5_ZS	0.0000
79	NE_CON_GOVT_CN	0.0000
88	NE_DAB_TOTL_CN	0.0000
101	NE_IMP_GNFS_CD	0.0000
300	TX_VAL_MRCH_R6_ZS	0.0000
102	NE_IMP_GNFS_CN	0.0000
259	SP_POP_TOTL	0.0000
10	AG_LND_TOTL_K2	0.0000
161	NY_TAX_NIND_CN	0.0000
166	SE_PRM_DURS	0.0000
42	EN_ATM_NOXE_EG_ZS	0.0000
172	SE_SECAGES	0.0000
41	EN_ATM_NOXE_AG_ZS	0.0000
233	SP_POP_4549_MA_5Y	0.0000
40	EN_ATM_METH_EG_ZS	0.0000
179	SH_DTH_MORT	0.0000
39	EN_ATM_METH_AG_ZS	0.0000
183	SH_DYN_MORT_MA	0.0000
194	SP_DYN_IMRT_IN	0.0000
222	SP_POP_2024_FE_5Y	0.0000
221	SP_POP_1564_TO_ZS	0.0000
220	SP_POP_1564_TO	0.0000
218	SP_POP_1564_MA_IN	0.0000
204	SP_POP_0014_FE_IN	0.0000
165	SE_PRMAGES	0.0000
47	EN_URB_MCTY_TL_ZS	0.0000
127	NY_ADJ_DNGY_CD	0.0000
155	NY_GNP_MKTP_CD	0.0000
130	NY_GDP_COAL_RT_ZS	0.0000
264	SP_RUR_TOTL	0.0000

```

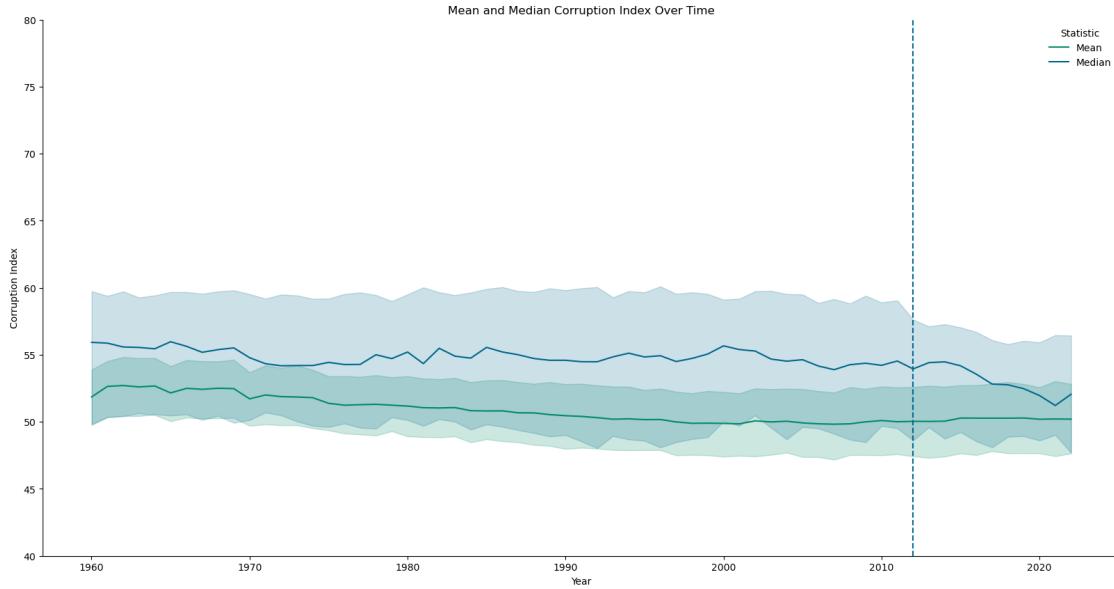
263      SP_POP_TOTL_MA_ZS      0.0000
262      SP_POP_TOTL_MA_IN      0.0000
11       AG_LND_TRAC_ZS      0.0000
135      NY_GDP_FCST_CN      0.0000
208      SP_POP_0014_TO      0.0000
51       ER_H2O_INTR_K3      0.0000
138      NY_GDP_MKTP_CD      0.0000
139      NY_GDP_MKTP_CN      0.0000
255      SP_POP_DPND      0.0000
140      NY_GDP_MKTP_KD      0.0000
142      NY_GDP_MKTP_KN      0.0000
144      NY_GDP_PCAP_CD      0.0000
50       ER_FSH_PROD_MT      0.0000
0        AG_AGR_TRAC_NO      0.0000

```

```

[208]: # Plot the aggregate mean and median CC_EST values (all countries together) over time in a line graph
# Create a line plot of the mean and median CC_EST values over time
plt.figure(figsize=(20, 10))
sns.lineplot(data=df_new, x='year', y='CC_EST', estimator='mean', label='Mean', color = '#008B68')
sns.lineplot(data=df_new, x='year', y='CC_EST', estimator='median', label='Median', color = '#00688B')
plt.axvline(x=2012, color='#00688B', linestyle='--')
plt.title('Mean and Median Corruption Index Over Time')
plt.xlabel('Year')
plt.ylabel('Corruption Index')
plt.ylim(40,80)
plt.legend(title='Statistic', frameon=False)
sns.despine()
plt.show()

```



```
[209]: # Filter the DataFrame to include only the years 1960 and 2022
df_filtered = df_new[df_new['year'].isin([1960, 2022])]

# Pivot the DataFrame to have one row per country and one column per year
df_pivot = df_filtered.pivot(index='country', columns='year', values='CC_EST')

# Calculate the change from 1960 to 2022 for each country
df_pivot['change'] = df_pivot[2022] - df_pivot[1960]

# Plot a histogram of the change in CC_EST
plt.figure(figsize=(20, 10))
sns.histplot(df_pivot['change'], bins=20, color='#00688B', edgecolor='#00688B', kde=True)

# Calculate the mean and median
mean = df_pivot['change'].mean()
median = df_pivot['change'].median()

# Add lines for the mean and median
plt.axvline(mean, color='#00688B', linestyle='--', label=f'Mean: {mean:.3f}')
plt.axvline(median, color='#00688B', linestyle='-', label=f'Median: {median:.3f}')

# Customize the title and labels
plt.title('Change in Corruption Estimates by Country (1960-2022)', fontsize=16)
plt.xlabel('Change', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
```

```

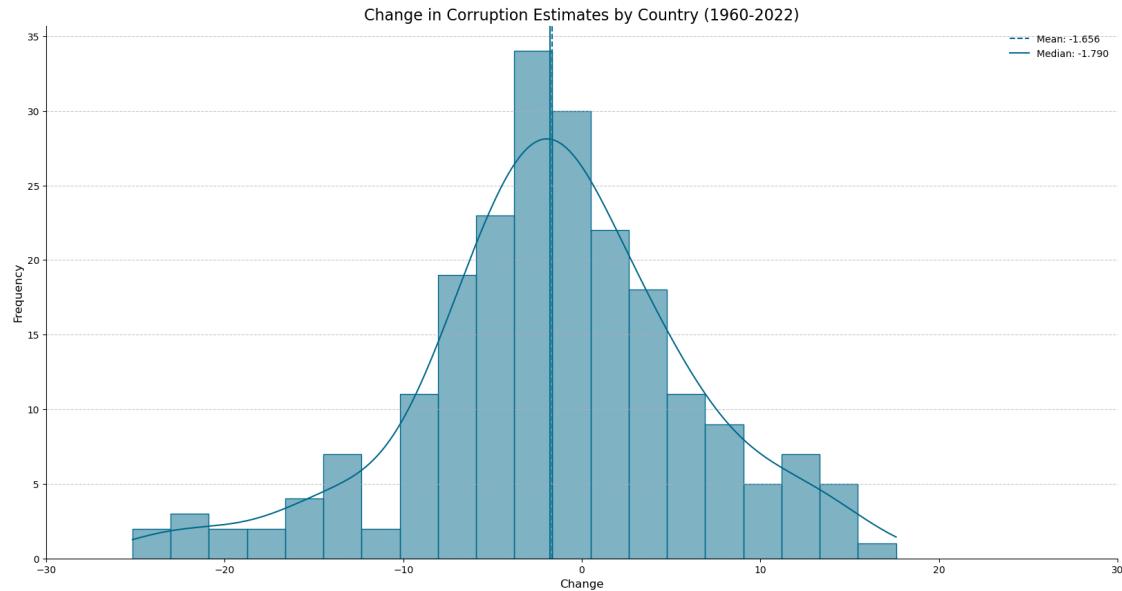
plt.xlim(-30,30)

# Customize the grid
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend(frameon=False)

# Remove the top and right spines
sns.despine()

plt.show()

```



```

[210]: # Plot a histogram of the average yearly change in CC_EST
plt.figure(figsize=(20, 10))
sns.histplot(df_change['change'], bins=20,color="#00688B", edgecolor="#00688B",  

↳kde=True)

# Calculate the mean and median
mean = df_change['change'].mean()
median = df_change['change'].median()

# Add lines for the mean and median
plt.axvline(mean, color="#00688B", linestyle='--', label=f'Mean: {mean:.3f}')
plt.axvline(median, color="#00688B", linestyle='-', label=f'Median: {median:.  

↳3f}')

# Customize the title and labels
plt.title('Change in Corruption Estimates (1960-2022)', fontsize=16)

```

```

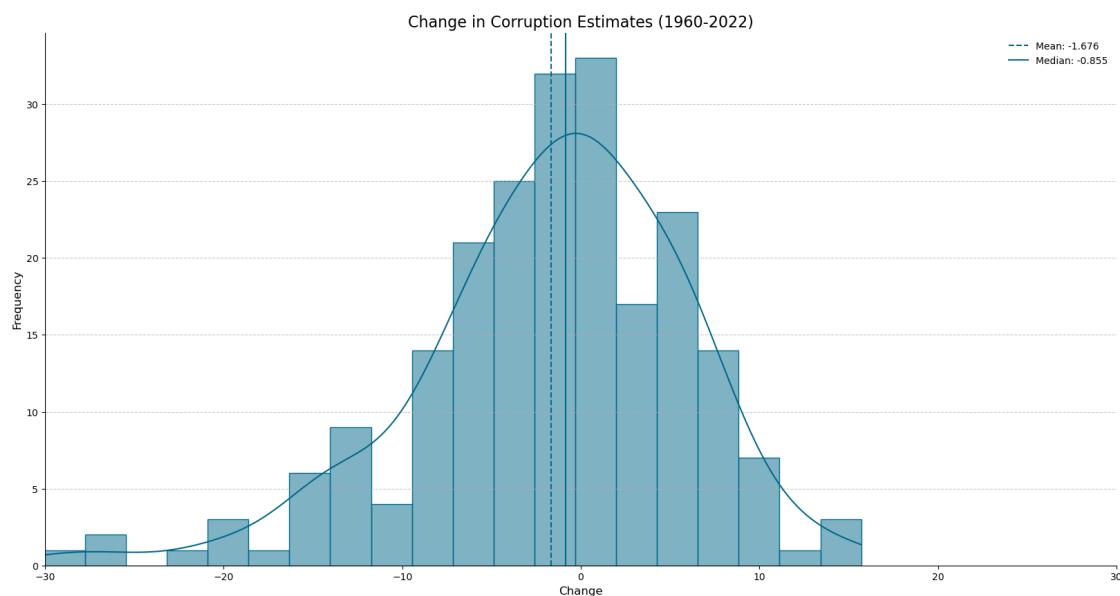
plt.xlabel('Change', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.xlim(-30,30)

# Customize the grid
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend(frameon=False)

# Remove the top and right spines
sns.despine()

plt.show()

```



```

[211]: # Calculate the average change in CC_EST between years
df_change = df_new.pivot(index='iso2c', columns='year', values='CC_EST')
df_change['change'] = (df_change[2022] - df_change[1960]) / (2022 - 1960)

# Sort the DataFrame in ascending order by 'change'
df_change = df_change.sort_values(by='change')

fig, ax = plt.subplots(figsize=(20, 10))

sns.barplot(x=df_change.index, y='change', data=df_change, ax=ax, palette = 'crest')

ax.set_title('Average Yearly Change in Corruption Estimates (1960-2022)', fontsize=20)

```

```

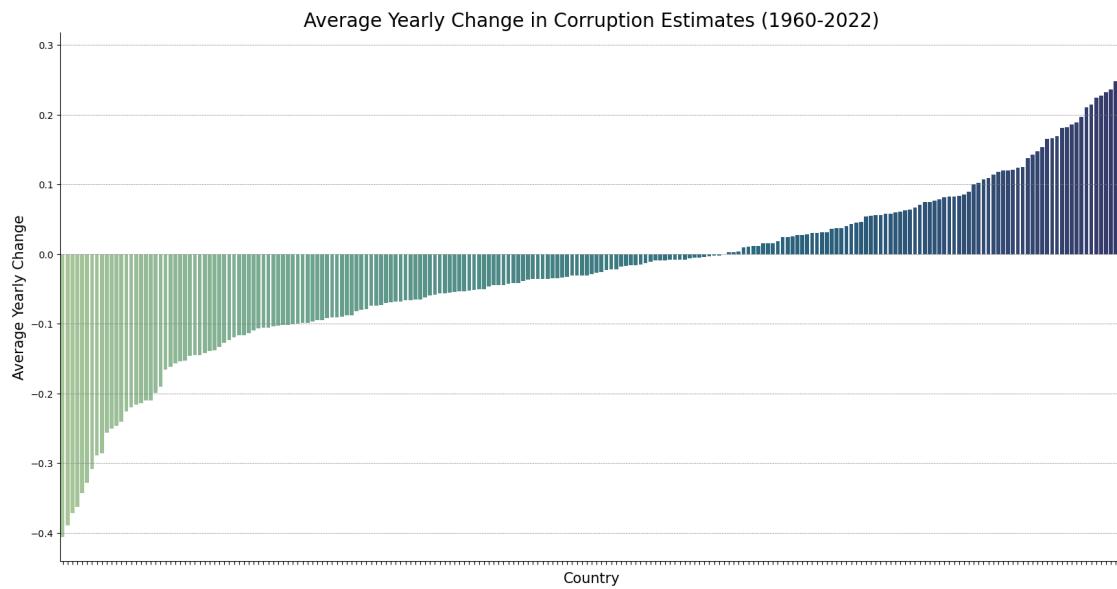
ax.set_xlabel('Country', fontsize=15)
ax.set_ylabel('Average Yearly Change', fontsize=15)

ax.set_xticklabels([])
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

ax.grid(color='gray', linestyle='--', linewidth=0.5, axis='y')

plt.show()

```



```

[212]: # Convert the list of tuples to a dictionary
importances_dict = results[5].set_index('Feature')['Importance'].to_dict()

# Extract feature names
features = list(importances_dict.keys())

# Since all values are already floats, there's no need to calculate the mean
importance_scores = list(importances_dict.values())

# Create a 2D line plot
fig, ax = plt.subplots(figsize=(20, 10))

# Create a sequence of numbers for the x-axis
x = np.arange(len(features))

# Create the 2D line plot

```

```

ax.plot(x, importance_scores, color="#00688B")

# Set the ticks on the x-axis to be the new labels and remove the lines (ticks)
ax.set_xticks(x, minor=False)
ax.set_xticklabels([])
ax.tick_params(axis='x', which='both', length=0)

# Set labels
ax.set_xlabel('Features')
ax.set_ylabel('Importance Scores')

sns.despine()

# Show the plot
plt.show()

```



```

[213]: countries = df['iso2c'].unique()

vars_full = df.columns.tolist()
vars_full.remove('iso3c')
vars_full.remove('iso3n')
vars_full.remove('year')

### TAKES AROUND 3 MINUTES
# Calculate the absolute correlation of each variable with the target variable
# Ensure that all columns in df are numeric
df_numeric = df[vars_full].apply(pd.to_numeric, errors='coerce')

```

```

# Calculate the absolute correlation of each variable with the target variable
correl = df_numeric.corrwith(df['CC_EST']).abs()

# Sort the variables by their correlation with the target variable
vars_sorted = correl.sort_values(ascending=False).index.tolist()

results = []
tree_depths = []
r2_scores = []

# Set the maximum number of variables to 10 or the total number of variables, whichever is smaller
max_vars = min(100, len(vars_sorted))

class TqdmCallback(xgb.callback.TrainingCallback):
    def __init__(self, bar):
        self._bar = bar

    def after_iteration(self, model, epoch, evals_log):
        self._bar.update(1)
        return False

# Create a progress bar
with tqdm(total=max_vars) as pbar:
    model = XGBRegressor(
        objective='reg:squarederror',
        random_state=0,
        alpha=1.0,
        reg_lambda=10.0,
        early_stopping_rounds=10,
        callbacks=[TqdmCallback(pbar)])
)

# Create a progress bar
with tqdm(total=max_vars) as pbar:
    # Iterate over the number of variables to include in the model
    for i in range(1, max_vars + 1):
        # Select the top i variables
        vars_selected = vars_sorted[:i]

        # Build and evaluate the model with the selected variables
        try:
            result = build_and_evaluate_model(df, 'CC_EST', vars_selected, n_leads=2)
        except Exception as e:

```

```

        print(f"Error while building model with variables {vars_selected}: {e}")
    continue

    # Store the result
    results.append(result)

    # Access the R^2 score for the test set
    r2_test = result[1]

    # Store the R^2 score
    r2_scores.append(r2_test)

    # Access the model object, which is assumed to be the last element in
    # the tuple
    model = result[10]

    # Retrieve the 'max_depth' parameter from the model's parameters
    tree_depth = model.get_params().get('max_depth', 'Not set')

    # Store the tree depth
    tree_depths.append(tree_depth)

    # Update the progress bar description
    pbar.set_description(f"Number of variables: {i}, R^2: {r2_test:.2f},"
    #Tree Depth: {tree_depth}")

    # Update the progress bar
    pbar.update()

```

```

0%|          | 0/100 [00:00<?, ?it/s]
0%|          | 0/100 [00:00<?,
?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
49it [00:00, 1203.05it/s]
Number of variables: 1, R^2: 0.88, Tree Depth: None:  1%|          | 1/100
[00:00<00:21,  4.51it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
38it [00:00, 1056.81it/s]
Number of variables: 2, R^2: 0.98, Tree Depth: None:  2%|          | 2/100
[00:00<00:20,  4.85it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-

```

```
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
29it [00:00, 981.69it/s]
Number of variables: 3, R^2: 0.98, Tree Depth: None:  3%|           | 3/100
[0:00<00:19,  5.00it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
26it [00:00, 774.87it/s]
Number of variables: 4, R^2: 0.98, Tree Depth: None:  4%|           | 4/100
[0:00<00:20,  4.66it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
28it [00:00, 784.80it/s]
Number of variables: 5, R^2: 0.98, Tree Depth: None:  5%|           | 5/100
[0:01<00:22,  4.15it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
33it [00:00, 826.30it/s]
Number of variables: 6, R^2: 0.98, Tree Depth: None:  6%|           | 6/100
[0:01<00:24,  3.86it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
30it [00:00, 753.79it/s]
Number of variables: 7, R^2: 0.98, Tree Depth: None:  7%|           | 7/100
[0:01<00:24,  3.74it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
38it [00:00, 695.22it/s]
Number of variables: 8, R^2: 0.98, Tree Depth: None:  8%|           | 8/100
[0:02<00:26,  3.54it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
28it [00:00, 499.68it/s]
Number of variables: 9, R^2: 0.98, Tree Depth: None:  9%|           | 9/100
```

```

[00:02<00:30,  2.95it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
37it [00:00, 629.05it/s]
Number of variables: 10, R^2: 0.98, Tree Depth: None: 10%|           | 10/100
[00:02<00:31,  2.90it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
28it [00:00, 127.47it/s]
Number of variables: 11, R^2: 0.98, Tree Depth: None: 11%|           | 11/100
[00:03<00:37,  2.38it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
39it [00:00, 548.04it/s]
Number of variables: 12, R^2: 0.98, Tree Depth: None: 12%|           | 12/100
[00:03<00:38,  2.28it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
35it [00:00, 565.12it/s]
Number of variables: 13, R^2: 0.98, Tree Depth: None: 13%|           | 13/100
[00:04<00:37,  2.33it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
70it [00:00, 473.44it/s]
Number of variables: 14, R^2: 0.98, Tree Depth: None: 14%|           | 14/100
[00:04<00:39,  2.15it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
86it [00:00, 584.39it/s]
Number of variables: 15, R^2: 0.98, Tree Depth: None: 15%|           | 15/100
[00:05<00:44,  1.93it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
28it [00:00, 560.44it/s]

```

```
Number of variables: 16, R^2: 0.98, Tree Depth: None: 16% | 16/100
[00:06<00:42, 1.96it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
57it [00:00, 450.13it/s]
Number of variables: 17, R^2: 0.98, Tree Depth: None: 17% | 17/100
[00:06<00:43, 1.93it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
45it [00:00, 461.19it/s]
Number of variables: 18, R^2: 0.98, Tree Depth: None: 18% | 18/100
[00:07<00:44, 1.84it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
56it [00:00, 504.27it/s]
Number of variables: 19, R^2: 0.98, Tree Depth: None: 19% | 19/100
[00:07<00:45, 1.80it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
31it [00:00, 437.73it/s]
Number of variables: 20, R^2: 0.98, Tree Depth: None: 20% | 20/100
[00:08<00:45, 1.76it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
53it [00:00, 407.00it/s]
Number of variables: 21, R^2: 0.98, Tree Depth: None: 21% | 21/100
[00:09<00:48, 1.64it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
29it [00:00, 330.39it/s]
Number of variables: 22, R^2: 0.98, Tree Depth: None: 22% | 22/100
[00:09<00:49, 1.59it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
```

```
65it [00:00, 183.16it/s]
Number of variables: 23, R^2: 0.98, Tree Depth: None: 23%| 23/100
[00:10<00:52, 1.46it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
60it [00:00, 441.61it/s]
Number of variables: 24, R^2: 0.98, Tree Depth: None: 24%| 24/100
[00:11<00:51, 1.47it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
41it [00:00, 346.61it/s]
Number of variables: 25, R^2: 0.98, Tree Depth: None: 25%| 25/100
[00:12<00:54, 1.39it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
52it [00:00, 410.60it/s]
Number of variables: 26, R^2: 0.98, Tree Depth: None: 26%| 26/100
[00:12<00:56, 1.31it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
44it [00:00, 346.75it/s]
Number of variables: 27, R^2: 0.98, Tree Depth: None: 27%| 27/100
[00:13<00:56, 1.29it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
64it [00:00, 328.78it/s]
Number of variables: 28, R^2: 0.98, Tree Depth: None: 28%| 28/100
[00:14<00:56, 1.28it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
67it [00:00, 395.03it/s]
Number of variables: 29, R^2: 0.98, Tree Depth: None: 29%| 29/100
[00:15<00:53, 1.32it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
```

```

    warnings.warn(
74it [00:00, 384.76it/s]
Number of variables: 30, R^2: 0.98, Tree Depth: None: 30%|           | 30/100
[00:15<00:53,  1.30it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
99it [00:00, 236.30it/s]
Number of variables: 31, R^2: 0.98, Tree Depth: None: 31%|           | 31/100
[00:17<00:58,  1.18it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
48it [00:00, 347.39it/s]
Number of variables: 32, R^2: 0.98, Tree Depth: None: 32%|           | 32/100
[00:17<00:57,  1.18it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
39it [00:00, 329.95it/s]
Number of variables: 33, R^2: 0.98, Tree Depth: None: 33%|           | 33/100
[00:18<00:55,  1.20it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
73it [00:00, 268.88it/s]
Number of variables: 34, R^2: 0.98, Tree Depth: None: 34%|           | 34/100
[00:19<00:56,  1.16it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
33it [00:00, 315.70it/s]
Number of variables: 35, R^2: 0.98, Tree Depth: None: 35%|           | 35/100
[00:20<00:55,  1.16it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
64it [00:00, 183.60it/s]
Number of variables: 36, R^2: 0.98, Tree Depth: None: 36%|           | 36/100
[00:21<01:03,  1.00it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in

```

```
constructor or `set_params` instead.

    warnings.warn(
86it [00:00, 320.26it/s]
Number of variables: 37, R^2: 0.98, Tree Depth: None: 37%|           | 37/100
[00:22<01:02,  1.01it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
53it [00:00, 322.43it/s]
Number of variables: 38, R^2: 0.98, Tree Depth: None: 38%|           | 38/100
[00:23<01:03,  1.02s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
74it [00:00, 294.66it/s]
Number of variables: 39, R^2: 0.98, Tree Depth: None: 39%|           | 39/100
[00:24<01:03,  1.04s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
55it [00:00, 170.42it/s]
Number of variables: 40, R^2: 0.98, Tree Depth: None: 40%|           | 40/100
[00:26<01:03,  1.06s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
72it [00:00, 317.89it/s]
Number of variables: 41, R^2: 0.98, Tree Depth: None: 41%|           | 41/100
[00:27<01:03,  1.08s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
74it [00:00, 242.84it/s]
Number of variables: 42, R^2: 0.98, Tree Depth: None: 42%|           | 42/100
[00:28<01:07,  1.17s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
46it [00:00, 229.25it/s]
Number of variables: 43, R^2: 0.98, Tree Depth: None: 43%|           | 43/100
[00:29<01:08,  1.20s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
```

```
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
56it [00:00, 100.61it/s]
Number of variables: 44, R^2: 0.98, Tree Depth: None: 44%|        | 44/100
[00:31<01:12,  1.29s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
48it [00:00, 262.73it/s]
Number of variables: 45, R^2: 0.98, Tree Depth: None: 45%|        | 45/100
[00:32<01:08,  1.24s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
50it [00:00, 254.20it/s]
Number of variables: 46, R^2: 0.98, Tree Depth: None: 46%|        | 46/100
[00:33<01:05,  1.20s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
77it [00:00, 248.67it/s]
Number of variables: 47, R^2: 0.98, Tree Depth: None: 47%|        | 47/100
[00:34<01:07,  1.27s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
51it [00:00, 132.41it/s]
Number of variables: 48, R^2: 0.98, Tree Depth: None: 48%|        | 48/100
[00:36<01:09,  1.34s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
46it [00:00, 110.73it/s]
Number of variables: 49, R^2: 0.98, Tree Depth: None: 49%|        | 49/100
[00:37<01:10,  1.38s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
40it [00:00, 210.09it/s]
Number of variables: 50, R^2: 0.98, Tree Depth: None: 50%|        | 50/100
[00:39<01:08,  1.36s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
```

```
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
51it [00:00, 125.60it/s]
Number of variables: 51, R^2: 0.98, Tree Depth: None: 51%|      | 51/100
[00:40<01:11,  1.46s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
48it [00:00, 113.70it/s]
Number of variables: 52, R^2: 0.98, Tree Depth: None: 52%|      | 52/100
[00:42<01:11,  1.49s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
64it [00:00, 72.65it/s]
Number of variables: 53, R^2: 0.98, Tree Depth: None: 53%|      | 53/100
[00:44<01:17,  1.64s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
54it [00:00, 107.21it/s]
Number of variables: 54, R^2: 0.98, Tree Depth: None: 54%|      | 54/100
[00:46<01:15,  1.64s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
65it [00:00, 198.20it/s]
Number of variables: 55, R^2: 0.98, Tree Depth: None: 55%|      | 55/100
[00:48<01:19,  1.77s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
66it [00:00, 129.28it/s]
Number of variables: 56, R^2: 0.98, Tree Depth: None: 56%|      | 56/100
[00:50<01:19,  1.81s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
53it [00:00, 110.83it/s]
Number of variables: 57, R^2: 0.98, Tree Depth: None: 57%|      | 57/100
```

```

[00:52<01:22,  1.91s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
32it [00:00, 174.73it/s]
Number of variables: 58, R^2: 0.98, Tree Depth: None: 58%|           | 58/100
[00:54<01:23,  1.98s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
64it [00:00, 186.08it/s]
Number of variables: 59, R^2: 0.98, Tree Depth: None: 59%|           | 59/100
[00:56<01:27,  2.12s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
74it [00:00, 207.03it/s]
Number of variables: 60, R^2: 0.98, Tree Depth: None: 60%|           | 60/100
[00:59<01:27,  2.19s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
58it [00:00, 204.51it/s]
Number of variables: 61, R^2: 0.98, Tree Depth: None: 61%|           | 61/100
[01:00<01:17,  1.99s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
57it [00:00, 114.64it/s]
Number of variables: 62, R^2: 0.98, Tree Depth: None: 62%|           | 62/100
[01:02<01:13,  1.95s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
37it [00:00, 134.52it/s]
Number of variables: 63, R^2: 0.98, Tree Depth: None: 63%|           | 63/100
[01:04<01:16,  2.08s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
52it [00:00, 212.35it/s]

```

```
Number of variables: 64, R^2: 0.98, Tree Depth: None: 64%| 64/100
[01:07<01:15, 2.10s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
61it [00:00, 170.67it/s]
Number of variables: 65, R^2: 0.98, Tree Depth: None: 65%| 65/100
[01:10<01:21, 2.34s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
42it [00:00, 163.94it/s]
Number of variables: 66, R^2: 0.98, Tree Depth: None: 66%| 66/100
[01:12<01:17, 2.28s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:00, 182.79it/s]
Number of variables: 67, R^2: 0.98, Tree Depth: None: 67%| 67/100
[01:14<01:16, 2.31s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
86it [00:00, 171.94it/s]
Number of variables: 68, R^2: 0.98, Tree Depth: None: 68%| 68/100
[01:16<01:12, 2.28s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
73it [00:00, 184.95it/s]
Number of variables: 69, R^2: 0.98, Tree Depth: None: 69%| 69/100
[01:18<01:10, 2.27s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
50it [00:00, 154.06it/s]
Number of variables: 70, R^2: 0.98, Tree Depth: None: 70%| 70/100
[01:21<01:11, 2.39s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
```

```
60it [00:00, 174.67it/s]
Number of variables: 71, R^2: 0.98, Tree Depth: None: 71%| 71/100
[01:23<01:07, 2.32s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
67it [00:00, 174.03it/s]
Number of variables: 72, R^2: 0.98, Tree Depth: None: 72%| 72/100
[01:25<01:02, 2.25s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
55it [00:00, 164.84it/s]
Number of variables: 73, R^2: 0.98, Tree Depth: None: 73%| 73/100
[01:28<01:03, 2.34s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
65it [00:00, 139.50it/s]
Number of variables: 74, R^2: 0.98, Tree Depth: None: 74%| 74/100
[01:31<01:02, 2.41s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
100it [00:00, 150.53it/s]
Number of variables: 75, R^2: 0.98, Tree Depth: None: 75%| 75/100
[01:33<01:03, 2.55s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
67it [00:00, 155.32it/s]
Number of variables: 76, R^2: 0.98, Tree Depth: None: 76%| 76/100
[01:35<00:56, 2.34s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
43it [00:00, 94.75it/s]
Number of variables: 77, R^2: 0.98, Tree Depth: None: 77%| 77/100
[01:38<00:54, 2.39s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
```

```

    warnings.warn(
65it [00:00, 150.82it/s]
Number of variables: 78, R^2: 0.98, Tree Depth: None: 78%|      | 78/100
[01:40<00:52,  2.36s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
44it [00:00, 102.07it/s]
Number of variables: 79, R^2: 0.98, Tree Depth: None: 79%|      | 79/100
[01:43<00:51,  2.48s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
51it [00:00, 86.50it/s]
Number of variables: 80, R^2: 0.98, Tree Depth: None: 80%|      | 80/100
[01:46<00:53,  2.68s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
65it [00:00, 114.54it/s]
Number of variables: 81, R^2: 0.98, Tree Depth: None: 81%|      | 81/100
[01:49<00:50,  2.66s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
34it [00:00, 125.40it/s]
Number of variables: 82, R^2: 0.98, Tree Depth: None: 82%|      | 82/100
[01:51<00:45,  2.53s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
85it [00:00, 147.73it/s]
Number of variables: 83, R^2: 0.98, Tree Depth: None: 83%|      | 83/100
[01:54<00:44,  2.61s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
48it [00:00, 74.61it/s]
Number of variables: 84, R^2: 0.98, Tree Depth: None: 84%|      | 84/100
[01:56<00:41,  2.61s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in

```

```
constructor or `set_params` instead.

    warnings.warn(
87it [00:00, 129.79it/s]
Number of variables: 85, R^2: 0.98, Tree Depth: None: 85%|      | 85/100
[01:59<00:41,  2.76s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
42it [00:00, 125.44it/s]
Number of variables: 86, R^2: 0.98, Tree Depth: None: 86%|      | 86/100
[02:02<00:37,  2.68s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
73it [00:00, 133.62it/s]
Number of variables: 87, R^2: 0.98, Tree Depth: None: 87%|      | 87/100
[02:04<00:33,  2.61s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
46it [00:00, 115.43it/s]
Number of variables: 88, R^2: 0.98, Tree Depth: None: 88%|      | 88/100
[02:07<00:30,  2.57s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
100it [00:00, 134.30it/s]
Number of variables: 89, R^2: 0.98, Tree Depth: None: 89%|      | 89/100
[02:10<00:29,  2.67s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
91it [00:00, 131.58it/s]
Number of variables: 90, R^2: 0.98, Tree Depth: None: 90%|      | 90/100
[02:12<00:26,  2.69s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
100it [00:00, 104.33it/s]
Number of variables: 91, R^2: 0.98, Tree Depth: None: 91%|      | 91/100
[02:16<00:25,  2.88s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
```

```
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
36it [00:00, 118.39it/s]
Number of variables: 92, R^2: 0.98, Tree Depth: None: 92%|      | 92/100
[02:19<00:23,  2.90s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:00, 129.74it/s]
Number of variables: 93, R^2: 0.98, Tree Depth: None: 93%|      | 93/100
[02:22<00:20,  2.96s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
59it [00:00, 92.60it/s]
Number of variables: 94, R^2: 0.98, Tree Depth: None: 94%|      | 94/100
[02:25<00:18,  3.09s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
81it [00:00, 123.10it/s]
Number of variables: 95, R^2: 0.98, Tree Depth: None: 95%|      | 95/100
[02:28<00:15,  3.07s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
61it [00:00, 107.96it/s]
Number of variables: 96, R^2: 0.98, Tree Depth: None: 96%|      | 96/100
[02:31<00:12,  3.15s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
43it [00:00, 93.67it/s]
Number of variables: 97, R^2: 0.98, Tree Depth: None: 97%|      | 97/100
[02:34<00:09,  3.10s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:00, 112.91it/s]
Number of variables: 98, R^2: 0.98, Tree Depth: None: 98%|      | 98/100
[02:37<00:06,  3.02s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
```

```

packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
67it [00:00, 117.21it/s]
Number of variables: 99, R^2: 0.98, Tree Depth: None: 99%|      | 99/100
[02:40<00:03,  3.04s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
97it [00:00, 107.20it/s]
Number of variables: 100, R^2: 0.98, Tree Depth: None: 100%|      | 100/100
[02:44<00:00,  1.64s/it]

```

[214]: # Plot

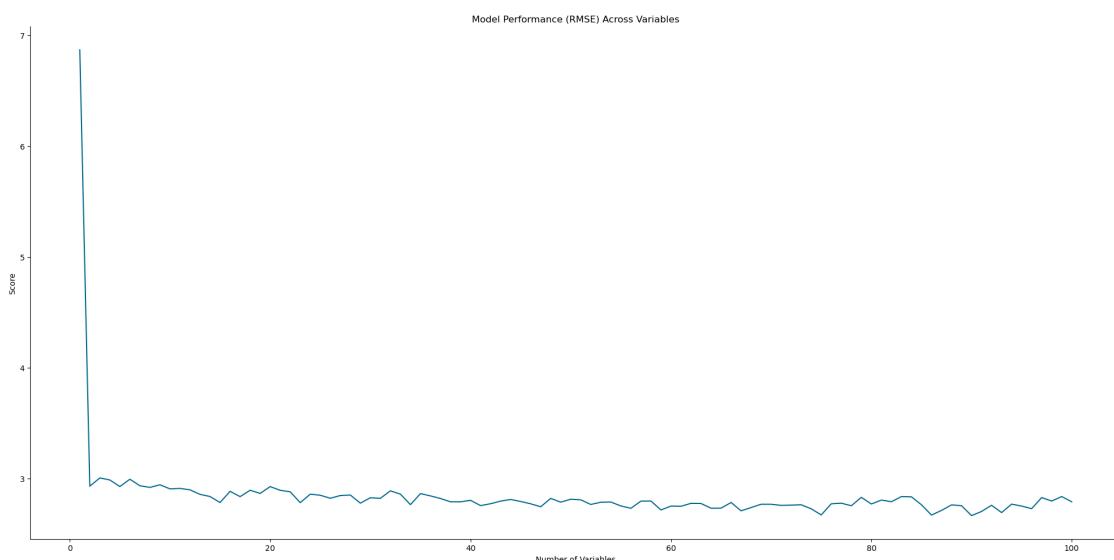
```

fig, ax = plt.subplots(figsize=(20, 10))

# Line plot of R^2 and RMSE across different thresholds
r2_scores = [res[1] for res in results]
rmse_scores = [res[3] for res in results]
num_vars = range(1, max_vars + 1)
ax.plot(num_vars, rmse_scores, color = '#00688B')
ax.set_xlabel('Number of Variables')
ax.set_ylabel('Score')
ax.set_title('Model Performance (RMSE) Across Variables')
sns.despine()

plt.tight_layout()
plt.show()

```



```
[215]: # Calculate the absolute differences for each model
abs_diffs = [abs(res[7] - res[8]) for res in results]

# Create a new subplot for the variance plot
fig, ax = plt.subplots(figsize=(20, 10))

# Get a color map
colors = sns.color_palette("mako", len(abs_diffs))

# KDE plot of the absolute differences for each model
for i, abs_diff in enumerate(abs_diffs):
    sns.kdeplot(abs_diff, color=colors[i], ax=ax)

# Calculate the aggregate mean and median of the absolute differences
mean_abs_diff = np.mean([np.mean(abs_diff) for abs_diff in abs_diffs])
median_abs_diff = np.median([np.median(abs_diff) for abs_diff in abs_diffs])

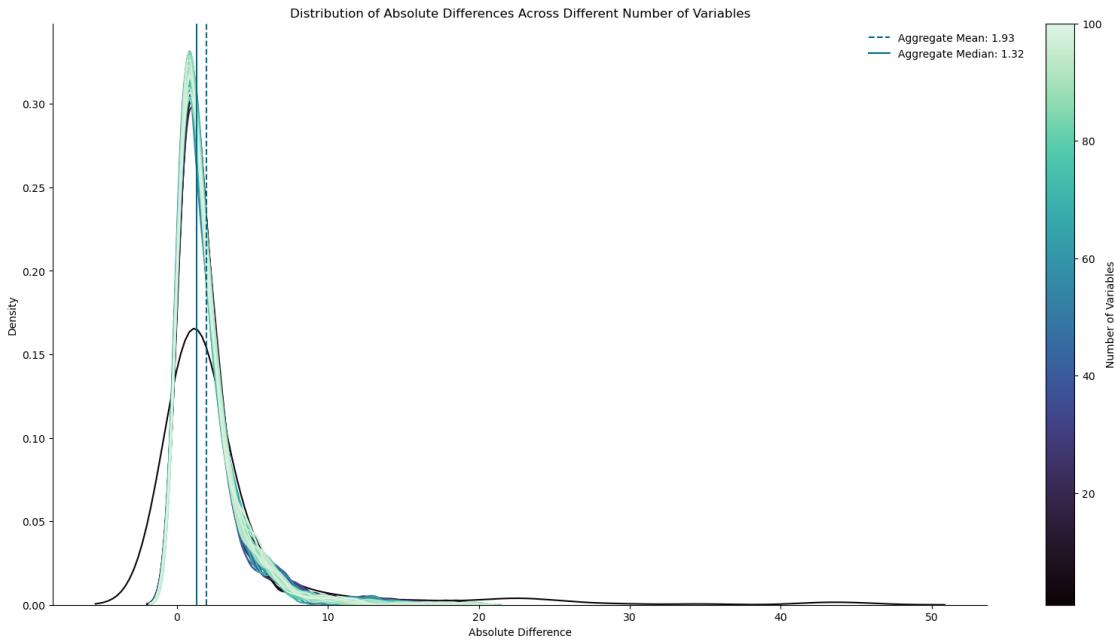
# Add a vertical line at the aggregate mean and median
ax.axvline(mean_abs_diff, color='#00688B', linestyle='--', label=f'Aggregate Mean: {mean_abs_diff:.2f}')
ax.axvline(median_abs_diff, color='#00688B', linestyle='-', label=f'Aggregate Median: {median_abs_diff:.2f}')

ax.set_xlabel('Absolute Difference')
ax.set_title('Distribution of Absolute Differences Across Different Number of Variables')

# Create a colorbar
norm = Normalize(vmin=1, vmax=len(abs_diffs))
cbar = plt.colorbar(cm.ScalarMappable(norm=norm, cmap=sns.color_palette("mako", len(abs_diffs))), ax=ax)
cbar.set_label('Number of Variables')

ax.legend(bbox_to_anchor=(1.05, 1), frameon=False)
sns.despine()

plt.show()
```



```
[216]: import seaborn as sns

# Calculate the absolute differences for each model
abs_diffs = [abs(res[7] - res[8]) for res in results]

# Create a new subplot for the variance plot
fig, ax = plt.subplots(figsize=(20, 10))

# Get a color map
colors = sns.color_palette("mako", len(abs_diffs))

# KDE plot of the absolute differences for each model
for i, abs_diff in enumerate(abs_diffs):
    sns.kdeplot(abs_diff, color=colors[i], ax=ax)

# Calculate the aggregate mean and median of the absolute differences
mean_abs_diff = np.mean([np.mean(abs_diff) for abs_diff in abs_diffs])
median_abs_diff = np.median([np.median(abs_diff) for abs_diff in abs_diffs])

# Add a vertical line at the aggregate mean and median
ax.axvline(mean_abs_diff, color="#00688B", linestyle='--', label=f'Aggregate\u20d7Mean: {mean_abs_diff:.2f}')
ax.axvline(median_abs_diff, color="#00688B", linestyle='-', label=f'Aggregate\u20d7Median: {median_abs_diff:.2f}')

ax.set_xlabel('Absolute Difference')
```

```

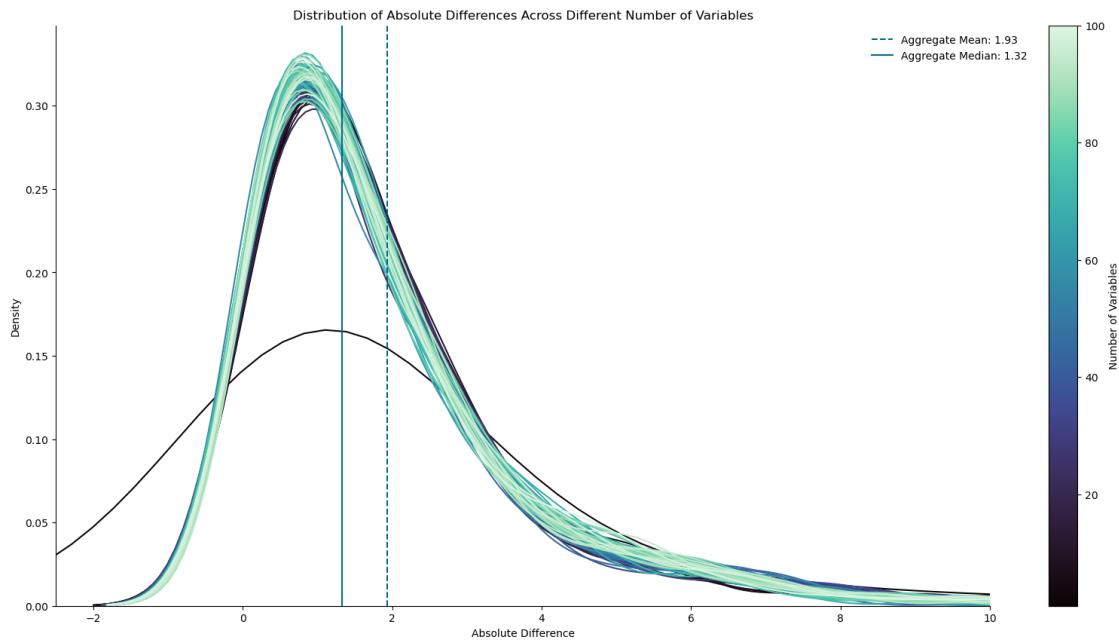
ax.set_title('Distribution of Absolute Differences Across Different Number of Variables')
ax.set_xlim(-2.5,10)

# Create a colorbar
norm = Normalize(vmin=1, vmax=len(abs_diffs))
cbar = plt.colorbar(cm.ScalarMappable(norm=norm, cmap=sns.color_palette("mako", as_cmap=True)), ax=ax)
cbar.set_label('Number of Variables')

ax.legend(bbox_to_anchor=(1.05, 1), frameon=False)
sns.despine()

plt.show()

```



```

[217]: # Create a new subplot for the scatter plot
fig, ax = plt.subplots(figsize=(20, 10))

# Create a colormap
cmap = sns.color_palette("mako", as_cmap=True)

# Calculate the number of lines to plot
num_lines = len(results)

# Scatter plot of the model accuracy over epochs for all models
for i, res in enumerate(results):

```

```

# Calculate the color for the current line
color = cmap(i / num_lines)

ax.plot(range(len(res[10].evals_result()['validation_0']['rmse'])),
        res[10].evals_result()['validation_0']['rmse'],
        label=f'{len(res[5])} variables',
        color=color)

ax.set_xlabel('Epoch')
ax.set_ylabel('RMSE')
ax.set_title('Model Accuracy Over Epochs')
ax.set_ylim(2.6, 3.2)

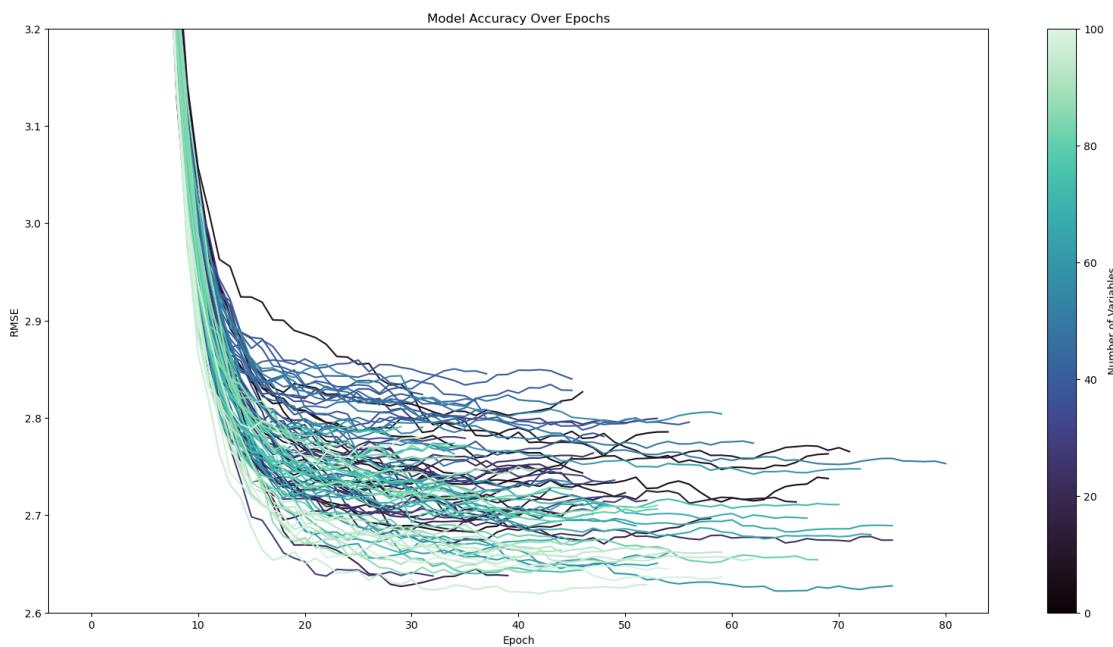
# Create a colorbar
sm = plt.cm.ScalarMappable(cmap=cmap, norm=plt.Normalize(vmin=0,
    vmax=num_lines))
plt.colorbar(sm, label='Number of Variables')

plt.show()

```

/var/folders/1z/rmh3bk123qg9411_qfj8858000gn/T/ipykernel_37400/1067117465.py:2
7: MatplotlibDeprecationWarning: Unable to determine Axes to steal space for
Colorbar. Using gca(), but will raise in the future. Either provide the *cax*
argument to use as the Axes for the Colorbar, provide the *ax* argument to steal
space from it, or add *mappable* to an Axes.

```
plt.colorbar(sm, label='Number of Variables')
```



```
[218]: def custom_objective(y_true, y_pred):
    """
        Custom objective function that penalizes predictions close to or beyond the
        bounds of 0 and 100,
        penalizes large changes from the previous year's prediction, and penalizes
        an overall movement greater than 20 from the 2012 value.
        Also penalizes the overreliance on the 'CPI_EST_avg' feature.
    """
    # Convert y_true and y_pred to pandas Series
    y_true = pd.Series(y_true)
    y_pred = pd.Series(y_pred)

    # Calculate the squared error
    squared_error = (y_true - y_pred) ** 2

    # Calculate the penalty term for the overreliance on the 'CPI_EST_avg' feature
    penalty_CPI_EST_avg = np.abs(y_pred)

    # Increase the penalty for the overreliance on the 'CPI_EST_avg' feature
    penalty_CPI_EST_avg *= 100

    # Add the penalty terms to the squared error
    penalized_squared_error = squared_error + penalty_CPI_EST_avg

    # Calculate the first derivative (gradient) of the penalized squared error
    grad = -2 * (y_true - y_pred) + np.sign(y_pred)

    # Calculate the second derivative (Hessian) of the penalized squared error
    hess = np.ones_like(grad) * 2

    return grad, hess

class TqdmCallback(xgb.callback.TrainingCallback):
    def __init__(self, bar):
        self._bar = bar

    def after_iteration(self, model, epoch, evals_log):
        self._bar.update(1)
        return False

    def build_and_evaluate_model(self, df, target_col_name, var_list, n_leads=2, max_depth=None):
        """
            Builds, evaluates, and returns results for an XGBoost model.
        """
        # Create a copy of the DataFrame to avoid modifying the original

```

```

df = df.copy()

# Exclude 'iso2c', 'country', target_col_name, and lead variables from
var_list
var_list = [var for var in var_list if var not in ['iso2c', 'country',  

target_col_name, 'year', 'iso3n'] + [f"{target_col_name}_lead{i}" for i in  

range(1, n_leads + 1)]]
if 'WB_CC_EST_avg' not in var_list:
    var_list.append('WB_CC_EST_avg')

# Remove rows with invalid values in the target column
df = df[np.isfinite(df[target_col_name]) & (abs(df[target_col_name]) <=  

1e30)]

# Add a new feature for the previous year's target value
df[target_col_name + '_prev'] = df.groupby('iso2c')[target_col_name].shift()
var_list.append(target_col_name + '_prev')

X = df[var_list]

df_train_val, df_test = train_test_split(df, test_size=0.1, random_state=0)

x_train_val, x_test, y_train_val, y_test = train_test_split(X,  

df[target_col_name], test_size=0.1, random_state=0)

# Use the custom objective function in the XGBoost model
model = XGBRegressor(
    objective=custom_objective,
    random_state=0,
    alpha=1.0,
    reg_lambda=10.0,
    early_stopping_rounds=10,
    max_depth=max_depth,
    gamma=2.0
)

with tqdm(total=model.get_params()['n_estimators']) as pbar:
    model.fit(
        x_train_val,
        y_train_val,
        eval_set=[(x_test, y_test)],
        verbose=False,
        callbacks=[TqdmCallback(pbar)]
    )

y_train_val_pred = model.predict(x_train_val)
y_test_pred = model.predict(x_test)

```

```

# Clip the predictions to be within the desired range
y_train_val_pred = np.clip(y_train_val_pred, 0, 100)
y_test_pred = np.clip(y_test_pred, 0, 100)

r2_train_val = round(r2_score(y_train_val, y_train_val_pred), 5)
r2_test = round(r2_score(y_test, y_test_pred), 5)

rmse_train_val = round(np.sqrt(mean_squared_error(y_train_val, y_train_val_pred)), 5)
rmse_test = round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 5)

cv_scores = custom_cross_val_score(model, x_train_val, y_train_val, cv=5)
mean_cv_score = round(np.mean(cv_scores), 5)

# Get the feature importances
feature_importances = model.feature_importances_

# Get the feature names from x_lead
feature_names = x_train_val.columns.tolist()

# Align feature importances with feature names
feature_importances_aligned = pd.Series(feature_importances, index=feature_names)

# Create a DataFrame with the aligned feature importances
feature_importances_df = pd.DataFrame({
    'Feature': feature_importances_aligned.index,
    'Importance': np.round(feature_importances_aligned.values, 4)
}).sort_values(by='Importance', ascending=False)

return r2_train_val, r2_test, rmse_train_val, rmse_test, mean_cv_score, feature_importances_df, x_test.index, y_test_pred, y_test, df, model

```

[219]: *### TAKES c.12 minutes!*

```

# Assuming df is your DataFrame and build_and_evaluate_model is a defined function
# Assuming vars_full is a list of column names in df

# Select the full set of variables
vars_selected = df[vars_full].columns.tolist()

# Define max depth values to test
max_depth_values = list(range(1, 21))

```

```

# Initialize lists to store R^2 scores for training and test sets
train_r2_scores = []
test_r2_scores = []

# Create a progress bar
with tqdm(total=len(max_depth_values)) as pbar:
    # Iterate over each max_depth value
    for max_depth in max_depth_values:
        # Build and evaluate the model with the full set of variables and
        # current max_depth
        try:
            result = build_and_evaluate_model(df, 'CC_EST', vars_selected,
                                              n_leads=2, max_depth=max_depth)
        except Exception as e:
            print(f"Error while building model at max_depth {max_depth}: {e}")
            continue

        # Access the R^2 scores for the training and test sets
        r2_train = result[0]
        r2_test = result[1]

        # Store the R^2 scores
        train_r2_scores.append(r2_train)
        test_r2_scores.append(r2_test)

        # Update the progress bar description
        pbar.set_description(f"Max Depth: {max_depth}, Train R^2: {r2_train:.2f}, Test R^2: {r2_test:.2f}")

        # Update the progress bar
        pbar.update()

```

```

0% | 0/20 [00:00<?, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
100it [00:00, 181.07it/s]
Max Depth: 1, Train R^2: 0.98, Test R^2: 0.96: 5% | 1/20
[00:02<00:55, 2.94s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
100it [00:00, 125.68it/s]
Max Depth: 2, Train R^2: 0.99, Test R^2: 0.97: 10% | 2/20

```

```

[00:08<01:21,  4.51s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
77it [00:00, 102.54it/s]
Max Depth: 3, Train R^2: 0.99, Test R^2: 0.98:  15%|           | 3/20
[00:12<01:15,  4.43s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
50it [00:01, 43.86it/s]
Max Depth: 4, Train R^2: 0.99, Test R^2: 0.97:  20%|           | 4/20
[00:19<01:25,  5.32s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
44it [00:01, 43.94it/s]
Max Depth: 5, Train R^2: 1.00, Test R^2: 0.97:  25%|           | 5/20
[00:28<01:39,  6.62s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
40it [00:01, 33.84it/s]
Max Depth: 6, Train R^2: 1.00, Test R^2: 0.97:  30%|           | 6/20
[00:35<01:36,  6.90s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
45it [00:01, 29.00it/s]
Max Depth: 7, Train R^2: 1.00, Test R^2: 0.97:  35%|           | 7/20
[00:45<01:41,  7.83s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
47it [00:01, 28.87it/s]
Max Depth: 8, Train R^2: 1.00, Test R^2: 0.97:  40%|           | 8/20
[00:57<01:48,  9.02s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
67it [00:01, 39.27it/s]

```

```

Max Depth: 9, Train R^2: 1.00, Test R^2: 0.97: 45%|           | 9/20
[01:07<01:43,  9.42s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
37it [00:02, 17.72it/s]
Max Depth: 10, Train R^2: 1.00, Test R^2: 0.97: 50%|           | 10/20
[01:20<01:46, 10.64s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
43it [00:02, 14.36it/s]
Max Depth: 11, Train R^2: 1.00, Test R^2: 0.97: 55%|           | 11/20
[01:32<01:39, 11.05s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
41it [00:01, 22.12it/s]
Max Depth: 12, Train R^2: 1.00, Test R^2: 0.97: 60%|           | 12/20
[01:42<01:25, 10.63s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
91it [00:01, 47.97it/s]
Max Depth: 13, Train R^2: 1.00, Test R^2: 0.97: 65%|           | 13/20
[01:53<01:14, 10.58s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
67it [00:01, 34.57it/s]
Max Depth: 14, Train R^2: 1.00, Test R^2: 0.97: 70%|           | 14/20
[02:03<01:03, 10.63s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
94it [00:01, 50.11it/s]
Max Depth: 15, Train R^2: 1.00, Test R^2: 0.97: 75%|           | 15/20
[02:14<00:52, 10.59s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(

```

```

40it [00:03, 12.46it/s]
Max Depth: 16, Train R^2: 1.00, Test R^2: 0.97: 80%|       | 16/20
[02:26<00:44, 11.04s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
95it [00:02, 44.26it/s]
Max Depth: 17, Train R^2: 1.00, Test R^2: 0.97: 85%|       | 17/20
[02:37<00:33, 11.17s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
92it [00:02, 44.34it/s]
Max Depth: 18, Train R^2: 1.00, Test R^2: 0.97: 90%|       | 18/20
[02:48<00:22, 11.15s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
36it [00:01, 21.48it/s]
Max Depth: 19, Train R^2: 1.00, Test R^2: 0.97: 95%|       | 19/20
[03:00<00:11, 11.29s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
90it [00:02, 40.45it/s]
Max Depth: 20, Train R^2: 1.00, Test R^2: 0.97: 100%|       | 20/20
[03:11<00:00, 9.60s/it]

```

```

[220]: for result in results:
    print(f"Model with {len(result[5])} variables and max depth {result[10]..
        ↪get_params()['max_depth']}:")
    print(f"Training+Validation R^2: {result[0]}, RMSE: {result[2]}")
    print(f"Testing R^2: {result[1]}, RMSE: {result[3]}")
    print(f"Mean cross-validation score: {result[4]}\n")
    print(result[5]) # Feature importances
    print('\n')

```

```

Model with 1 variables and max depth None:
Training+Validation R^2: 0.89712, RMSE: 6.39414
Testing R^2: 0.88423, RMSE: 6.86756
Mean cross-validation score: 0.89481

```

	Feature	Importance
0	CC_EST_prev	1.0

Model with 2 variables and max depth None:
Training+Validation R^2: 0.98902, RMSE: 2.08932
Testing R^2: 0.97892, RMSE: 2.93073
Mean cross-validation score: 0.98086

	Feature	Importance
0	WB_CC_EST_avg	0.9679
1	CC_EST_prev	0.0321

Model with 3 variables and max depth None:
Training+Validation R^2: 0.98966, RMSE: 2.02743
Testing R^2: 0.97783, RMSE: 3.00534
Mean cross-validation score: 0.98091

	Feature	Importance
0	WB_CC_EST_avg	0.9715
2	CC_EST_prev	0.0264
1	NY_GNP_PCAP_CD	0.0022

Model with 4 variables and max depth None:
Training+Validation R^2: 0.98992, RMSE: 2.0017
Testing R^2: 0.9781, RMSE: 2.98701
Mean cross-validation score: 0.98049

	Feature	Importance
0	WB_CC_EST_avg	0.9699
3	CC_EST_prev	0.0257
2	NY_GDP_PCAP_KD_rel	0.0025
1	NY_GNP_PCAP_CD	0.0019

Model with 5 variables and max depth None:
Training+Validation R^2: 0.99064, RMSE: 1.92914
Testing R^2: 0.97896, RMSE: 2.92737
Mean cross-validation score: 0.98082

	Feature	Importance
0	WB_CC_EST_avg	0.9704
4	CC_EST_prev	0.0229
3	NY_GDP_PCAP_KD	0.0028
2	NY_GDP_PCAP_KD_rel	0.0020
1	NY_GNP_PCAP_CD	0.0018

Model with 6 variables and max depth None:
 Training+Validation R^2: 0.99162, RMSE: 1.82442
 Testing R^2: 0.978, RMSE: 2.99384
 Mean cross-validation score: 0.98082

	Feature	Importance
0	WB_CC_EST_avg	0.9652
5	CC_EST_prev	0.0265
3	NY_GDP_PCAP_KD	0.0028
2	NY_GDP_PCAP_KD_rel	0.0020
4	NY_GDP_PCAP_CD	0.0019
1	NY_GNP_PCAP_CD	0.0017

Model with 7 variables and max depth None:
 Training+Validation R^2: 0.99151, RMSE: 1.83715
 Testing R^2: 0.97886, RMSE: 2.93463
 Mean cross-validation score: 0.98052

	Feature	Importance
0	WB_CC_EST_avg	0.9675
6	CC_EST_prev	0.0235
5	SP_DYN_LE00_MA_IN	0.0020
2	NY_GDP_PCAP_KD_rel	0.0019
3	NY_GDP_PCAP_KD	0.0019
4	NY_GDP_PCAP_CD	0.0018
1	NY_GNP_PCAP_CD	0.0014

Model with 8 variables and max depth None:
 Training+Validation R^2: 0.99292, RMSE: 1.67714
 Testing R^2: 0.97907, RMSE: 2.91996
 Mean cross-validation score: 0.98034

	Feature	Importance
0	WB_CC_EST_avg	0.9709
7	CC_EST_prev	0.0189
3	NY_GDP_PCAP_KD	0.0021
5	SP_DYN_LE00_MA_IN	0.0019
4	NY_GDP_PCAP_CD	0.0018
2	NY_GDP_PCAP_KD_rel	0.0016
6	SP_DYN_LE00_IN	0.0015
1	NY_GNP_PCAP_CD	0.0013

Model with 9 variables and max depth None:
 Training+Validation R^2: 0.99126, RMSE: 1.86399
 Testing R^2: 0.97873, RMSE: 2.94361

Mean cross-validation score: 0.98066

	Feature	Importance
0	WB_CC_EST_avg	0.9716
8	CC_EST_prev	0.0178
3	NY_GDP_PCAP_KD	0.0019
7	SP_DYN_LE00_FE_IN	0.0017
4	NY_GDP_PCAP_CD	0.0016
5	SP_DYN_LE00_MA_IN	0.0016
6	SP_DYN_LE00_IN	0.0015
2	NY_GDP_PCAP_KD_rel	0.0013
1	NY_GNP_PCAP_CD	0.0010

Model with 10 variables and max depth None:

Training+Validation R^2: 0.99387, RMSE: 1.56134

Testing R^2: 0.97926, RMSE: 2.90647

Mean cross-validation score: 0.98122

	Feature	Importance
0	WB_CC_EST_avg	0.9680
9	CC_EST_prev	0.0198
8	SP_POP_0014_MA_ZS	0.0021
4	NY_GDP_PCAP_CD	0.0017
7	SP_DYN_LE00_FE_IN	0.0016
2	NY_GDP_PCAP_KD_rel	0.0015
6	SP_DYN_LE00_IN	0.0015
3	NY_GDP_PCAP_KD	0.0014
5	SP_DYN_LE00_MA_IN	0.0014
1	NY_GNP_PCAP_CD	0.0010

Model with 11 variables and max depth None:

Training+Validation R^2: 0.99246, RMSE: 1.73146

Testing R^2: 0.9792, RMSE: 2.91078

Mean cross-validation score: 0.98092

	Feature	Importance
0	WB_CC_EST_avg	0.9642
10	CC_EST_prev	0.0224
8	SP_POP_0014_MA_ZS	0.0021
7	SP_DYN_LE00_FE_IN	0.0018
9	SP_POP_0509_MA_5Y	0.0018
2	NY_GDP_PCAP_KD_rel	0.0015
5	SP_DYN_LE00_MA_IN	0.0015
6	SP_DYN_LE00_IN	0.0014
4	NY_GDP_PCAP_CD	0.0013
3	NY_GDP_PCAP_KD	0.0011

1 NY_GNP_PCAP_CD 0.0010

Model with 12 variables and max depth None:
Training+Validation R^2: 0.99431, RMSE: 1.50432
Testing R^2: 0.97938, RMSE: 2.89803
Mean cross-validation score: 0.98103

	Feature	Importance
0	WB_CC_EST_avg	0.9672
11	CC_EST_prev	0.0194
8	SP_POP_0014_MA_ZS	0.0019
7	SP_DYN_LEOO_FE_IN	0.0016
10	SP_POP_0014_TO_ZS	0.0015
3	NY_GDP_PCAP_KD	0.0014
5	SP_DYN_LEOO_MA_IN	0.0013
9	SP_POP_0509_MA_5Y	0.0013
2	NY_GDP_PCAP_KD_rel	0.0012
4	NY_GDP_PCAP_CD	0.0012
6	SP_DYN_LEOO_IN	0.0011
1	NY_GNP_PCAP_CD	0.0010

Model with 13 variables and max depth None:
Training+Validation R^2: 0.9935, RMSE: 1.60749
Testing R^2: 0.97996, RMSE: 2.85718
Mean cross-validation score: 0.98103

	Feature	Importance
0	WB_CC_EST_avg	0.9647
12	CC_EST_prev	0.0201
8	SP_POP_0014_MA_ZS	0.0023
10	SP_POP_0014_TO_ZS	0.0020
7	SP_DYN_LEOO_FE_IN	0.0015
11	SP_POP_1014_MA_5Y	0.0015
3	NY_GDP_PCAP_KD	0.0012
5	SP_DYN_LEOO_MA_IN	0.0012
6	SP_DYN_LEOO_IN	0.0012
9	SP_POP_0509_MA_5Y	0.0012
2	NY_GDP_PCAP_KD_rel	0.0011
4	NY_GDP_PCAP_CD	0.0010
1	NY_GNP_PCAP_CD	0.0008

Model with 14 variables and max depth None:
Training+Validation R^2: 0.99699, RMSE: 1.09304
Testing R^2: 0.98024, RMSE: 2.8375
Mean cross-validation score: 0.98122

	Feature	Importance
0	WB_CC_EST_avg	0.9558
13	CC_EST_prev	0.0257
8	SP_POP_0014_MA_ZS	0.0030
10	SP_POP_0014_TO_ZS	0.0024
3	NY_GDP_PCAP_KD	0.0016
5	SP_DYN_LE00_MA_IN	0.0015
7	SP_DYN_LE00_FE_IN	0.0015
9	SP_POP_0509_MA_5Y	0.0015
2	NY_GDP_PCAP_KD_rel	0.0014
12	SP_POP_65UP_MA_ZS	0.0013
6	SP_DYN_LE00_IN	0.0012
11	SP_POP_1014_MA_5Y	0.0012
4	NY_GDP_PCAP_CD	0.0009
1	NY_GNP_PCAP_CD	0.0008

Model with 15 variables and max depth None:

Training+Validation R^2: 0.99798, RMSE: 0.89503

Testing R^2: 0.98098, RMSE: 2.78332

Mean cross-validation score: 0.98109

	Feature	Importance
0	WB_CC_EST_avg	0.9681
14	CC_EST_prev	0.0168
8	SP_POP_0014_MA_ZS	0.0021
10	SP_POP_0014_TO_ZS	0.0016
13	SP_POP_0004_MA_5Y	0.0014
5	SP_DYN_LE00_MA_IN	0.0012
3	NY_GDP_PCAP_KD	0.0011
7	SP_DYN_LE00_FE_IN	0.0011
11	SP_POP_1014_MA_5Y	0.0011
2	NY_GDP_PCAP_KD_rel	0.0010
6	SP_DYN_LE00_IN	0.0010
12	SP_POP_65UP_MA_ZS	0.0010
1	NY_GNP_PCAP_CD	0.0008
4	NY_GDP_PCAP_CD	0.0008
9	SP_POP_0509_MA_5Y	0.0008

Model with 16 variables and max depth None:

Training+Validation R^2: 0.99297, RMSE: 1.67183

Testing R^2: 0.97957, RMSE: 2.88478

Mean cross-validation score: 0.98109

	Feature	Importance
0	WB_CC_EST_avg	0.9628

15	CC_EST_prev	0.0190
8	SP_POP_0014_MA_ZS	0.0026
10	SP_POP_0014_TO_ZS	0.0018
13	SP_POP_0004_MA_5Y	0.0017
3	NY_GDP_PCAP_KD	0.0014
5	SP_DYN_LE00_MA_IN	0.0014
12	SP_POP_65UP_MA_ZS	0.0014
11	SP_POP_1014_MA_5Y	0.0013
7	SP_DYN_LE00_FE_IN	0.0011
14	SP_POP_6569_MA_5Y	0.0011
6	SP_DYN_LE00_IN	0.0010
9	SP_POP_0509_MA_5Y	0.0010
4	NY_GDP_PCAP_CD	0.0009
1	NY_GNP_PCAP_CD	0.0008
2	NY_GDP_PCAP_KD_rel	0.0008

Model with 17 variables and max depth None:

Training+Validation R^2: 0.99676, RMSE: 1.13429

Testing R^2: 0.98025, RMSE: 2.83648

Mean cross-validation score: 0.98128

	Feature	Importance
0	WB_CC_EST_avg	0.9555
16	CC_EST_prev	0.0226
8	SP_POP_0014_MA_ZS	0.0031
10	SP_POP_0014_TO_ZS	0.0026
11	SP_POP_1014_MA_5Y	0.0016
13	SP_POP_0004_MA_5Y	0.0016
12	SP_POP_65UP_MA_ZS	0.0015
3	NY_GDP_PCAP_KD	0.0014
7	SP_DYN_LE00_FE_IN	0.0013
14	SP_POP_6569_MA_5Y	0.0013
5	SP_DYN_LE00_MA_IN	0.0012
15	SP_POP_5054_MA_5Y	0.0012
2	NY_GDP_PCAP_KD_rel	0.0011
6	SP_DYN_LE00_IN	0.0011
9	SP_POP_0509_MA_5Y	0.0011
4	NY_GDP_PCAP_CD	0.0010
1	NY_GNP_PCAP_CD	0.0010

Model with 18 variables and max depth None:

Training+Validation R^2: 0.99578, RMSE: 1.29549

Testing R^2: 0.97944, RMSE: 2.89432

Mean cross-validation score: 0.9807

Feature Importance

0	WB_CC_EST_avg	0.9548
17	CC_EST_prev	0.0234
8	SP_POP_0014_MA_ZS	0.0036
10	SP_POP_0014_TO_ZS	0.0023
13	SP_POP_0004_MA_5Y	0.0016
11	SP_POP_1014_MA_5Y	0.0015
3	NY_GDP_PCAP_KD	0.0014
12	SP_POP_65UP_MA_ZS	0.0012
14	SP_POP_6569_MA_5Y	0.0012
16	SP_POP_7074_MA_5Y	0.0012
5	SP_DYN_LE00_MA_IN	0.0011
7	SP_DYN_LE00_FE_IN	0.0011
15	SP_POP_5054_MA_5Y	0.0011
2	NY_GDP_PCAP_KD_rel	0.0010
4	NY_GDP_PCAP_CD	0.0010
6	SP_DYN_LE00_IN	0.0010
1	NY_GNP_PCAP_CD	0.0008
9	SP_POP_0509_MA_5Y	0.0008

Model with 19 variables and max depth None:
 Training+Validation R^2: 0.99688, RMSE: 1.11265
 Testing R^2: 0.97984, RMSE: 2.86566
 Mean cross-validation score: 0.98055

	Feature	Importance
0	WB_CC_EST_avg	0.9518
18	CC_EST_prev	0.0242
8	SP_POP_0014_MA_ZS	0.0035
10	SP_POP_0014_TO_ZS	0.0029
13	SP_POP_0004_MA_5Y	0.0016
11	SP_POP_1014_MA_5Y	0.0015
16	SP_POP_7074_MA_5Y	0.0014
17	SP_POP_5559_MA_5Y	0.0013
2	NY_GDP_PCAP_KD_rel	0.0013
12	SP_POP_65UP_MA_ZS	0.0012
6	SP_DYN_LE00_IN	0.0012
15	SP_POP_5054_MA_5Y	0.0012
5	SP_DYN_LE00_MA_IN	0.0012
3	NY_GDP_PCAP_KD	0.0011
7	SP_DYN_LE00_FE_IN	0.0011
14	SP_POP_6569_MA_5Y	0.0010
9	SP_POP_0509_MA_5Y	0.0010
1	NY_GNP_PCAP_CD	0.0008
4	NY_GDP_PCAP_CD	0.0008

Model with 20 variables and max depth None:

Training+Validation R^2: 0.99385, RMSE: 1.56379
 Testing R^2: 0.97897, RMSE: 2.92714
 Mean cross-validation score: 0.98074

	Feature	Importance
0	WB_CC_EST_avg	0.9482
19	CC_EST_prev	0.0235
8	SP_POP_0014_MA_ZS	0.0035
18	SP_POP_0014_FE_ZS	0.0031
13	SP_POP_0004_MA_5Y	0.0019
9	SP_POP_0509_MA_5Y	0.0019
3	NY_GDP_PCAP_KD	0.0017
10	SP_POP_0014_TO_ZS	0.0016
11	SP_POP_1014_MA_5Y	0.0015
5	SP_DYN_LE00_MA_IN	0.0015
17	SP_POP_5559_MA_5Y	0.0014
7	SP_DYN_LE00_FE_IN	0.0012
12	SP_POP_65UP_MA_ZS	0.0012
15	SP_POP_5054_MA_5Y	0.0012
16	SP_POP_7074_MA_5Y	0.0012
4	NY_GDP_PCAP_CD	0.0012
2	NY_GDP_PCAP_KD_rel	0.0012
14	SP_POP_6569_MA_5Y	0.0011
1	NY_GNP_PCAP_CD	0.0009
6	SP_DYN_LE00_IN	0.0009

Model with 21 variables and max depth None:
 Training+Validation R^2: 0.99673, RMSE: 1.14077
 Testing R^2: 0.97945, RMSE: 2.89366
 Mean cross-validation score: 0.98088

	Feature	Importance
0	WB_CC_EST_avg	0.9569
20	CC_EST_prev	0.0200
18	SP_POP_0014_FE_ZS	0.0034
8	SP_POP_0014_MA_ZS	0.0028
12	SP_POP_65UP_MA_ZS	0.0014
17	SP_POP_5559_MA_5Y	0.0013
11	SP_POP_1014_MA_5Y	0.0012
19	IT_MLT_MAIN_P2	0.0012
3	NY_GDP_PCAP_KD	0.0012
13	SP_POP_0004_MA_5Y	0.0012
5	SP_DYN_LE00_MA_IN	0.0011
6	SP_DYN_LE00_IN	0.0010
15	SP_POP_5054_MA_5Y	0.0009
16	SP_POP_7074_MA_5Y	0.0009
2	NY_GDP_PCAP_KD_rel	0.0009

14	SP_POP_6569_MA_5Y	0.0009
10	SP_POP_0014_TO_ZS	0.0009
9	SP_POP_0509_MA_5Y	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
4	NY_GDP_PCAP_CD	0.0007
1	NY_GNP_PCAP_CD	0.0006

Model with 22 variables and max depth None:

Training+Validation R^2: 0.99372, RMSE: 1.57953

Testing R^2: 0.97964, RMSE: 2.88025

Mean cross-validation score: 0.98102

	Feature	Importance
0	WB_CC_EST_avg	0.9533
21	CC_EST_prev	0.0220
8	SP_POP_0014_MA_ZS	0.0030
18	SP_POP_0014_FE_ZS	0.0022
11	SP_POP_1014_MA_5Y	0.0018
13	SP_POP_0004_MA_5Y	0.0014
20	SP_DYN_T065_MA_ZS	0.0014
17	SP_POP_5559_MA_5Y	0.0013
19	IT_MLT_MAIN_P2	0.0012
12	SP_POP_65UP_MA_ZS	0.0012
10	SP_POP_0014_TO_ZS	0.0011
3	NY_GDP_PCAP_KD	0.0011
15	SP_POP_5054_MA_5Y	0.0011
16	SP_POP_7074_MA_5Y	0.0011
14	SP_POP_6569_MA_5Y	0.0010
7	SP_DYN_LE00_FE_IN	0.0010
6	SP_DYN_LE00_IN	0.0010
9	SP_POP_0509_MA_5Y	0.0008
5	SP_DYN_LE00_MA_IN	0.0008
4	NY_GDP_PCAP_CD	0.0008
2	NY_GDP_PCAP_KD_rel	0.0008
1	NY_GNP_PCAP_CD	0.0006

Model with 23 variables and max depth None:

Training+Validation R^2: 0.99757, RMSE: 0.98201

Testing R^2: 0.981, RMSE: 2.78247

Mean cross-validation score: 0.98053

	Feature	Importance
0	WB_CC_EST_avg	0.9537
22	CC_EST_prev	0.0215
18	SP_POP_0014_FE_ZS	0.0027
8	SP_POP_0014_MA_ZS	0.0026

11	SP_POP_1014_MA_5Y	0.0014
13	SP_POP_0004_MA_5Y	0.0014
20	SP_DYN_T065_MA_ZS	0.0014
10	SP_POP_0014_TO_ZS	0.0014
21	SP_POP_0509_FE_5Y	0.0012
19	IT_MLT_MAIN_P2	0.0012
17	SP_POP_5559_MA_5Y	0.0012
16	SP_POP_7074_MA_5Y	0.0012
12	SP_POP_65UP_MA_ZS	0.0012
3	NY_GDP_PCAP_KD	0.0010
15	SP_POP_5054_MA_5Y	0.0010
6	SP_DYN_LE00_IN	0.0010
5	SP_DYN_LE00_MA_IN	0.0009
2	NY_GDP_PCAP_KD_rel	0.0009
14	SP_POP_6569_MA_5Y	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
4	NY_GDP_PCAP_CD	0.0007
1	NY_GNP_PCAP_CD	0.0006
9	SP_POP_0509_MA_5Y	0.0006

Model with 24 variables and max depth None:

Training+Validation R^2: 0.99705, RMSE: 1.08352

Testing R^2: 0.97994, RMSE: 2.85872

Mean cross-validation score: 0.98049

	Feature	Importance
0	WB_CC_EST_avg	0.9571
23	CC_EST_prev	0.0184
8	SP_POP_0014_MA_ZS	0.0028
18	SP_POP_0014_FE_ZS	0.0020
20	SP_DYN_T065_MA_ZS	0.0014
16	SP_POP_7074_MA_5Y	0.0014
10	SP_POP_0014_TO_ZS	0.0014
13	SP_POP_0004_MA_5Y	0.0013
11	SP_POP_1014_MA_5Y	0.0013
17	SP_POP_5559_MA_5Y	0.0012
21	SP_POP_0509_FE_5Y	0.0011
12	SP_POP_65UP_MA_ZS	0.0011
19	IT_MLT_MAIN_P2	0.0010
7	SP_DYN_LE00_FE_IN	0.0009
6	SP_DYN_LE00_IN	0.0009
22	SP_POP_1014_FE_5Y	0.0009
3	NY_GDP_PCAP_KD	0.0008
14	SP_POP_6569_MA_5Y	0.0008
15	SP_POP_5054_MA_5Y	0.0008
5	SP_DYN_LE00_MA_IN	0.0008
2	NY_GDP_PCAP_KD_rel	0.0008

9	SP_POP_0509_MA_5Y	0.0007
1	NY_GNP_PCAP_CD	0.0006
4	NY_GDP_PCAP_CD	0.0006

Model with 25 variables and max depth None:
 Training+Validation R^2: 0.99584, RMSE: 1.28575
 Testing R^2: 0.98007, RMSE: 2.84913
 Mean cross-validation score: 0.98118

	Feature	Importance
0	WB_CC_EST_avg	0.9614
24	CC_EST_prev	0.0184
8	SP_POP_0014_MA_ZS	0.0019
23	SP_DYN_CBRT_IN	0.0013
18	SP_POP_0014_FE_ZS	0.0013
20	SP_DYN_T065_MA_ZS	0.0012
13	SP_POP_0004_MA_5Y	0.0010
3	NY_GDP_PCAP_KD	0.0010
17	SP_POP_5559_MA_5Y	0.0010
21	SP_POP_0509_FE_5Y	0.0009
19	IT_MLT_MAIN_P2	0.0009
11	SP_POP_1014_MA_5Y	0.0009
16	SP_POP_7074_MA_5Y	0.0008
22	SP_POP_1014_FE_5Y	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
15	SP_POP_5054_MA_5Y	0.0008
2	NY_GDP_PCAP_KD_rel	0.0007
10	SP_POP_0014_TO_ZS	0.0007
9	SP_POP_0509_MA_5Y	0.0007
6	SP_DYN_LE00_IN	0.0007
14	SP_POP_6569_MA_5Y	0.0006
5	SP_DYN_LE00_MA_IN	0.0006
4	NY_GDP_PCAP_CD	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
1	NY_GNP_PCAP_CD	0.0005

Model with 26 variables and max depth None:
 Training+Validation R^2: 0.99692, RMSE: 1.10658
 Testing R^2: 0.98045, RMSE: 2.82192
 Mean cross-validation score: 0.98038

	Feature	Importance
0	WB_CC_EST_avg	0.9589
25	CC_EST_prev	0.0167
8	SP_POP_0014_MA_ZS	0.0029
18	SP_POP_0014_FE_ZS	0.0022

23	SP_DYN_CBRT_IN	0.0014
20	SP_DYN_T065_MA_ZS	0.0013
11	SP_POP_1014_MA_5Y	0.0012
19	IT_MLT_MAIN_P2	0.0010
13	SP_POP_0004_MA_5Y	0.0010
24	SP_POP_6064_MA_5Y	0.0010
10	SP_POP_0014_TO_ZS	0.0010
22	SP_POP_1014_FE_5Y	0.0009
16	SP_POP_7074_MA_5Y	0.0009
17	SP_POP_5559_MA_5Y	0.0009
7	SP_DYN_LE00_FE_IN	0.0008
12	SP_POP_65UP_MA_ZS	0.0008
3	NY_GDP_PCAP_KD	0.0008
15	SP_POP_5054_MA_5Y	0.0008
21	SP_POP_0509_FE_5Y	0.0007
2	NY_GDP_PCAP_KD_rel	0.0007
9	SP_POP_0509_MA_5Y	0.0007
6	SP_DYN_LE00_IN	0.0007
5	SP_DYN_LE00_MA_IN	0.0007
14	SP_POP_6569_MA_5Y	0.0007
1	NY_GNP_PCAP_CD	0.0006
4	NY_GDP_PCAP_CD	0.0006

Model with 27 variables and max depth None:

Training+Validation R^2: 0.99577, RMSE: 1.29706

Testing R^2: 0.98011, RMSE: 2.84639

Mean cross-validation score: 0.98027

	Feature	Importance
0	WB_CC_EST_avg	0.9470
26	CC_EST_prev	0.0233
18	SP_POP_0014_FE_ZS	0.0029
8	SP_POP_0014_MA_ZS	0.0028
11	SP_POP_1014_MA_5Y	0.0017
23	SP_DYN_CBRT_IN	0.0016
20	SP_DYN_T065_MA_ZS	0.0016
13	SP_POP_0004_MA_5Y	0.0014
25	SP_POP_0004_FE_5Y	0.0014
19	IT_MLT_MAIN_P2	0.0012
3	NY_GDP_PCAP_KD	0.0011
24	SP_POP_6064_MA_5Y	0.0010
17	SP_POP_5559_MA_5Y	0.0010
16	SP_POP_7074_MA_5Y	0.0010
2	NY_GDP_PCAP_KD_rel	0.0010
10	SP_POP_0014_TO_ZS	0.0010
9	SP_POP_0509_MA_5Y	0.0010
14	SP_POP_6569_MA_5Y	0.0010

7	SP_DYN_LE00_FE_IN	0.0009
22	SP_POP_1014_FE_5Y	0.0009
15	SP_POP_5054_MA_5Y	0.0008
21	SP_POP_0509_FE_5Y	0.0008
5	SP_DYN_LE00_MA_IN	0.0008
1	NY_GNP_PCAP_CD	0.0007
12	SP_POP_65UP_MA_ZS	0.0007
6	SP_DYN_LE00_IN	0.0007
4	NY_GDP_PCAP_CD	0.0007

Model with 28 variables and max depth None:

Training+Validation R^2: 0.99767, RMSE: 0.96324
 Testing R^2: 0.98005, RMSE: 2.85122
 Mean cross-validation score: 0.98054

	Feature	Importance
0	WB_CC_EST_avg	0.9448
27	CC_EST_prev	0.0219
8	SP_POP_0014_MA_ZS	0.0039
18	SP_POP_0014_FE_ZS	0.0031
20	SP_DYN_T065_MA_ZS	0.0019
11	SP_POP_1014_MA_5Y	0.0017
25	SP_POP_0004_FE_5Y	0.0014
23	SP_DYN_CBRT_IN	0.0014
13	SP_POP_0004_MA_5Y	0.0014
9	SP_POP_0509_MA_5Y	0.0013
19	IT_MLT_MAIN_P2	0.0012
3	NY_GDP_PCAP_KD	0.0011
17	SP_POP_5559_MA_5Y	0.0011
16	SP_POP_7074_MA_5Y	0.0011
15	SP_POP_5054_MA_5Y	0.0011
10	SP_POP_0014_TO_ZS	0.0011
26	SP_POP_65UP_TO_ZS	0.0011
24	SP_POP_6064_MA_5Y	0.0010
7	SP_DYN_LE00_FE_IN	0.0010
6	SP_DYN_LE00_IN	0.0009
5	SP_DYN_LE00_MA_IN	0.0009
21	SP_POP_0509_FE_5Y	0.0009
2	NY_GDP_PCAP_KD_rel	0.0008
22	SP_POP_1014_FE_5Y	0.0008
12	SP_POP_65UP_MA_ZS	0.0008
14	SP_POP_6569_MA_5Y	0.0008
1	NY_GNP_PCAP_CD	0.0007
4	NY_GDP_PCAP_CD	0.0007

Model with 29 variables and max depth None:

Training+Validation R^2: 0.99808, RMSE: 0.87319
 Testing R^2: 0.98106, RMSE: 2.77775
 Mean cross-validation score: 0.98105

	Feature	Importance
0	WB_CC_EST_avg	0.9550
28	CC_EST_prev	0.0190
8	SP_POP_0014_MA_ZS	0.0031
21	SP_POP_0509_FE_5Y	0.0022
20	SP_DYN_T065_MA_ZS	0.0012
23	SP_DYN_CBRT_IN	0.0012
27	SP_POP_7579_MA_5Y	0.0011
26	SP_POP_65UP_TO_ZS	0.0011
19	IT_MLT_MAIN_P2	0.0011
13	SP_POP_0004_MA_5Y	0.0010
22	SP_POP_1014_FE_5Y	0.0010
11	SP_POP_1014_MA_5Y	0.0010
3	NY_GDP_PCAP_KD	0.0009
24	SP_POP_6064_MA_5Y	0.0009
25	SP_POP_0004_FE_5Y	0.0009
9	SP_POP_0509_MA_5Y	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
17	SP_POP_5559_MA_5Y	0.0008
2	NY_GDP_PCAP_KD_rel	0.0008
15	SP_POP_5054_MA_5Y	0.0007
1	NY_GNP_PCAP_CD	0.0007
10	SP_POP_0014_TO_ZS	0.0007
14	SP_POP_6569_MA_5Y	0.0007
16	SP_POP_7074_MA_5Y	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
6	SP_DYN_LE00_IN	0.0006
5	SP_DYN_LE00_MA_IN	0.0006
4	NY_GDP_PCAP_CD	0.0006
18	SP_POP_0014_FE_ZS	0.0003

Model with 30 variables and max depth None:
 Training+Validation R^2: 0.99839, RMSE: 0.79872
 Testing R^2: 0.98039, RMSE: 2.82648
 Mean cross-validation score: 0.98111

	Feature	Importance
0	WB_CC_EST_avg	0.9472
29	CC_EST_prev	0.0201
8	SP_POP_0014_MA_ZS	0.0037
21	SP_POP_0509_FE_5Y	0.0023
23	SP_DYN_CBRT_IN	0.0016
20	SP_DYN_T065_MA_ZS	0.0016

25	SP_POP_0004_FE_5Y	0.0014
13	SP_POP_0004_MA_5Y	0.0014
19	IT_MLT_MAIN_P2	0.0014
22	SP_POP_1014_FE_5Y	0.0013
11	SP_POP_1014_MA_5Y	0.0013
16	SP_POP_7074_MA_5Y	0.0012
17	SP_POP_5559_MA_5Y	0.0012
27	SP_POP_7579_MA_5Y	0.0012
26	SP_POP_65UP_TO_ZS	0.0012
24	SP_POP_6064_MA_5Y	0.0011
15	SP_POP_5054_MA_5Y	0.0010
3	NY_GDP_PCAP_KD	0.0010
2	NY_GDP_PCAP_KD_rel	0.0009
14	SP_POP_6569_MA_5Y	0.0008
10	SP_POP_0014_TO_ZS	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
18	SP_POP_0014_FE_ZS	0.0007
1	NY_GNP_PCAP_CD	0.0007
9	SP_POP_0509_MA_5Y	0.0007
6	SP_DYN_LE00_IN	0.0007
5	SP_DYN_LE00_MA_IN	0.0007
28	SH_DYN_NMRT	0.0007
12	SP_POP_65UP_MA_ZS	0.0006
4	NY_GDP_PCAP_CD	0.0005

Model with 31 variables and max depth None:

Training+Validation R^2: 0.99907, RMSE: 0.60946

Testing R^2: 0.98046, RMSE: 2.82144

Mean cross-validation score: 0.98096

	Feature	Importance
0	WB_CC_EST_avg	0.9547
30	CC_EST_prev	0.0197
8	SP_POP_0014_MA_ZS	0.0020
29	SP_POP_1519_MA_5Y	0.0018
20	SP_DYN_T065_MA_ZS	0.0016
7	SP_DYN_LE00_FE_IN	0.0014
21	SP_POP_0509_FE_5Y	0.0014
26	SP_POP_65UP_TO_ZS	0.0011
22	SP_POP_1014_FE_5Y	0.0011
17	SP_POP_5559_MA_5Y	0.0010
27	SP_POP_7579_MA_5Y	0.0010
23	SP_DYN_CBRT_IN	0.0010
19	IT_MLT_MAIN_P2	0.0009
16	SP_POP_7074_MA_5Y	0.0008
28	SH_DYN_NMRT	0.0008
25	SP_POP_0004_FE_5Y	0.0008

15	SP_POP_5054_MA_5Y	0.0008
14	SP_POP_6569_MA_5Y	0.0008
3	NY_GDP_PCAP_KD	0.0007
24	SP_POP_6064_MA_5Y	0.0007
9	SP_POP_0509_MA_5Y	0.0007
6	SP_DYN_LE00_IN	0.0007
5	SP_DYN_LE00_MA_IN	0.0007
13	SP_POP_0004_MA_5Y	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
2	NY_GDP_PCAP_KD_rel	0.0006
1	NY_GNP_PCAP_CD	0.0005
11	SP_POP_1014_MA_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0005
18	SP_POP_0014_FE_ZS	0.0003
4	NY_GDP_PCAP_CD	0.0003

Model with 32 variables and max depth None:

Training+Validation R^2: 0.99672, RMSE: 1.14164

Testing R^2: 0.97951, RMSE: 2.8892

Mean cross-validation score: 0.98099

	Feature	Importance
0	WB_CC_EST_avg	0.9481
31	CC_EST_prev	0.0211
8	SP_POP_0014_MA_ZS	0.0026
29	SP_POP_1519_MA_5Y	0.0021
20	SP_DYN_T065_MA_ZS	0.0018
7	SP_DYN_LE00_FE_IN	0.0016
23	SP_DYN_CBRT_IN	0.0014
27	SP_POP_7579_MA_5Y	0.0013
25	SP_POP_0004_FE_5Y	0.0013
22	SP_POP_1014_FE_5Y	0.0013
26	SP_POP_65UP_TO_ZS	0.0012
21	SP_POP_0509_FE_5Y	0.0012
17	SP_POP_5559_MA_5Y	0.0011
19	IT_MLT_MAIN_P2	0.0010
24	SP_POP_6064_MA_5Y	0.0010
30	SP_POP_DPND_YG	0.0009
16	SP_POP_7074_MA_5Y	0.0009
15	SP_POP_5054_MA_5Y	0.0008
14	SP_POP_6569_MA_5Y	0.0008
9	SP_POP_0509_MA_5Y	0.0008
28	SH_DYN_NMRT	0.0008
5	SP_DYN_LE00_MA_IN	0.0008
3	NY_GDP_PCAP_KD	0.0007
13	SP_POP_0004_MA_5Y	0.0007
12	SP_POP_65UP_MA_ZS	0.0007

6	SP_DYN_LE00_IN	0.0007
2	NY_GDP_PCAP_KD_rel	0.0007
1	NY_GNP_PCAP_CD	0.0006
11	SP_POP_1014_MA_5Y	0.0006
4	NY_GDP_PCAP_CD	0.0006
18	SP_POP_0014_FE_ZS	0.0005
10	SP_POP_0014_TO_ZS	0.0004

Model with 33 variables and max depth None:
 Training+Validation R^2: 0.99601, RMSE: 1.25884
 Testing R^2: 0.97994, RMSE: 2.85907
 Mean cross-validation score: 0.98138

	Feature	Importance
0	WB_CC_EST_avg	0.9426
32	CC_EST_prev	0.0254
8	SP_POP_0014_MA_ZS	0.0030
29	SP_POP_1519_MA_5Y	0.0021
23	SP_DYN_CBRT_IN	0.0019
20	SP_DYN_T065_MA_ZS	0.0016
25	SP_POP_0004_FE_5Y	0.0016
27	SP_POP_7579_MA_5Y	0.0013
26	SP_POP_65UP_TO_ZS	0.0012
7	SP_DYN_LE00_FE_IN	0.0012
24	SP_POP_6064_MA_5Y	0.0011
30	SP_POP_DPND_YG	0.0011
22	SP_POP_1014_FE_5Y	0.0010
19	IT_MLT_MAIN_P2	0.0010
28	SH_DYN_NMRT	0.0010
31	SP_POP_1519_FE_5Y	0.0010
14	SP_POP_6569_MA_5Y	0.0009
15	SP_POP_5054_MA_5Y	0.0009
21	SP_POP_0509_FE_5Y	0.0009
16	SP_POP_7074_MA_5Y	0.0009
17	SP_POP_5559_MA_5Y	0.0008
13	SP_POP_0004_MA_5Y	0.0008
18	SP_POP_0014_FE_ZS	0.0007
3	NY_GDP_PCAP_KD	0.0007
12	SP_POP_65UP_MA_ZS	0.0007
5	SP_DYN_LE00_MA_IN	0.0007
2	NY_GDP_PCAP_KD_rel	0.0007
9	SP_POP_0509_MA_5Y	0.0006
6	SP_DYN_LE00_IN	0.0006
4	NY_GDP_PCAP_CD	0.0006
1	NY_GNP_PCAP_CD	0.0005
11	SP_POP_1014_MA_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0004

Model with 34 variables and max depth None:
 Training+Validation R^2: 0.99829, RMSE: 0.8244
 Testing R^2: 0.98124, RMSE: 2.7642
 Mean cross-validation score: 0.98128

	Feature	Importance
0	WB_CC_EST_avg	0.9431
33	CC_EST_prev	0.0203
8	SP_POP_0014_MA_ZS	0.0047
29	SP_POP_1519_MA_5Y	0.0030
32	SP_POP_5054_FE_5Y	0.0015
23	SP_DYN_CBRT_IN	0.0015
7	SP_DYN_LE00_FE_IN	0.0014
20	SP_DYN_T065_MA_ZS	0.0013
25	SP_POP_0004_FE_5Y	0.0012
24	SP_POP_6064_MA_5Y	0.0012
19	IT_MLT_MAIN_P2	0.0012
21	SP_POP_0509_FE_5Y	0.0012
27	SP_POP_7579_MA_5Y	0.0011
16	SP_POP_7074_MA_5Y	0.0011
17	SP_POP_5559_MA_5Y	0.0011
22	SP_POP_1014_FE_5Y	0.0010
5	SP_DYN_LE00_MA_IN	0.0010
28	SH_DYN_NMRT	0.0010
10	SP_POP_0014_TO_ZS	0.0010
26	SP_POP_65UP_TO_ZS	0.0009
3	NY_GDP_PCAP_KD	0.0009
31	SP_POP_1519_FE_5Y	0.0009
15	SP_POP_5054_MA_5Y	0.0008
30	SP_POP_DPND_YG	0.0008
11	SP_POP_1014_MA_5Y	0.0008
6	SP_DYN_LE00_IN	0.0008
14	SP_POP_6569_MA_5Y	0.0008
2	NY_GDP_PCAP_KD_rel	0.0007
13	SP_POP_0004_MA_5Y	0.0007
12	SP_POP_65UP_MA_ZS	0.0007
9	SP_POP_0509_MA_5Y	0.0007
18	SP_POP_0014_FE_ZS	0.0007
1	NY_GNP_PCAP_CD	0.0005
4	NY_GDP_PCAP_CD	0.0004

Model with 35 variables and max depth None:
 Training+Validation R^2: 0.99455, RMSE: 1.47232
 Testing R^2: 0.97986, RMSE: 2.86417
 Mean cross-validation score: 0.98112

	Feature	Importance
0	WB_CC_EST_avg	0.9392
34	CC_EST_prev	0.0226
8	SP_POP_0014_MA_ZS	0.0033
29	SP_POP_1519_MA_5Y	0.0029
7	SP_DYN_LE00_FE_IN	0.0017
21	SP_POP_0509_FE_5Y	0.0015
20	SP_DYN_T065_MA_ZS	0.0015
26	SP_POP_65UP_TO_ZS	0.0015
23	SP_DYN_CBRT_IN	0.0015
27	SP_POP_7579_MA_5Y	0.0012
28	SH_DYN_NMRT	0.0012
22	SP_POP_1014_FE_5Y	0.0012
32	SP_POP_5054_FE_5Y	0.0012
33	SP_POP_80UP_FE_5Y	0.0012
31	SP_POP_1519_FE_5Y	0.0011
14	SP_POP_6569_MA_5Y	0.0011
25	SP_POP_0004_FE_5Y	0.0011
24	SP_POP_6064_MA_5Y	0.0011
19	IT_MLT_MAIN_P2	0.0011
9	SP_POP_0509_MA_5Y	0.0010
16	SP_POP_7074_MA_5Y	0.0010
3	NY_GDP_PCAP_KD	0.0010
17	SP_POP_5559_MA_5Y	0.0010
2	NY_GDP_PCAP_KD_rel	0.0009
15	SP_POP_5054_MA_5Y	0.0009
13	SP_POP_0004_MA_5Y	0.0008
12	SP_POP_65UP_MA_ZS	0.0008
6	SP_DYN_LE00_IN	0.0008
5	SP_DYN_LE00_MA_IN	0.0008
4	NY_GDP_PCAP_CD	0.0008
11	SP_POP_1014_MA_5Y	0.0007
30	SP_POP_DPND_YG	0.0007
1	NY_GNP_PCAP_CD	0.0006
10	SP_POP_0014_TO_ZS	0.0005
18	SP_POP_0014_FE_ZS	0.0004

Model with 36 variables and max depth None:

Training+Validation R^2: 0.99796, RMSE: 0.89979

Testing R^2: 0.98016, RMSE: 2.84311

Mean cross-validation score: 0.98137

	Feature	Importance
0	WB_CC_EST_avg	0.9534
35	CC_EST_prev	0.0173
8	SP_POP_0014_MA_ZS	0.0023

29	SP_POP_1519_MA_5Y	0.0018
23	SP_DYN_CBRT_IN	0.0015
34	SP_DYN_IMRT_MA_IN	0.0012
7	SP_DYN_LEOO_FE_IN	0.0012
20	SP_DYN_T065_MA_ZS	0.0012
27	SP_POP_7579_MA_5Y	0.0011
26	SP_POP_65UP_TO_ZS	0.0011
19	IT_MLT_MAIN_P2	0.0009
24	SP_POP_6064_MA_5Y	0.0009
16	SP_POP_7074_MA_5Y	0.0009
31	SP_POP_1519_FE_5Y	0.0009
13	SP_POP_0004_MA_5Y	0.0009
32	SP_POP_5054_FE_5Y	0.0009
33	SP_POP_80UP_FE_5Y	0.0009
14	SP_POP_6569_MA_5Y	0.0008
17	SP_POP_5559_MA_5Y	0.0008
25	SP_POP_0004_FE_5Y	0.0008
21	SP_POP_0509_FE_5Y	0.0008
22	SP_POP_1014_FE_5Y	0.0007
15	SP_POP_5054_MA_5Y	0.0007
3	NY_GDP_PCAP_KD	0.0007
30	SP_POP_DPND_YG	0.0007
5	SP_DYN_LEOO_MA_IN	0.0007
28	SH_DYN_NMRT	0.0007
11	SP_POP_1014_MA_5Y	0.0006
4	NY_GDP_PCAP_CD	0.0006
2	NY_GDP_PCAP_KD_rel	0.0005
1	NY_GNP_PCAP_CD	0.0005
6	SP_DYN_LEOO_IN	0.0005
18	SP_POP_0014_FE_ZS	0.0005
12	SP_POP_65UP_MA_ZS	0.0004
9	SP_POP_0509_MA_5Y	0.0004
10	SP_POP_0014_TO_ZS	0.0003

Model with 37 variables and max depth None:

Training+Validation R^2: 0.99889, RMSE: 0.66528

Testing R^2: 0.98048, RMSE: 2.81984

Mean cross-validation score: 0.98108

	Feature	Importance
0	WB_CC_EST_avg	0.9474
36	CC_EST_prev	0.0201
8	SP_POP_0014_MA_ZS	0.0026
29	SP_POP_1519_MA_5Y	0.0021
27	SP_POP_7579_MA_5Y	0.0013
23	SP_DYN_CBRT_IN	0.0013
7	SP_DYN_LEOO_FE_IN	0.0013

16	SP_POP_7074_MA_5Y	0.0012
34	SP_DYN_IMRT_MA_IN	0.0012
20	SP_DYN_T065_MA_ZS	0.0012
31	SP_POP_1519_FE_5Y	0.0010
22	SP_POP_1014_FE_5Y	0.0010
21	SP_POP_0509_FE_5Y	0.0010
19	IT_MLT_MAIN_P2	0.0010
33	SP_POP_80UP_FE_5Y	0.0009
13	SP_POP_0004_MA_5Y	0.0009
35	SP_POP_65UP_FE_ZS	0.0009
32	SP_POP_5054_FE_5Y	0.0009
24	SP_POP_6064_MA_5Y	0.0009
6	SP_DYN_LE00_IN	0.0008
5	SP_DYN_LE00_MA_IN	0.0008
26	SP_POP_65UP_TO_ZS	0.0008
14	SP_POP_6569_MA_5Y	0.0008
3	NY_GDP_PCAP_KD	0.0008
17	SP_POP_5559_MA_5Y	0.0008
30	SP_POP_DPND_YG	0.0008
11	SP_POP_1014_MA_5Y	0.0007
12	SP_POP_65UP_MA_ZS	0.0007
28	SH_DYN_NMRT	0.0007
15	SP_POP_5054_MA_5Y	0.0006
25	SP_POP_0004_FE_5Y	0.0006
2	NY_GDP_PCAP_KD_rel	0.0005
1	NY_GNP_PCAP_CD	0.0005
4	NY_GDP_PCAP_CD	0.0005
10	SP_POP_0014_TO_ZS	0.0004
9	SP_POP_0509_MA_5Y	0.0004
18	SP_POP_0014_FE_ZS	0.0004

Model with 38 variables and max depth None:

Training+Validation R^2: 0.9971, RMSE: 1.07302

Testing R^2: 0.98089, RMSE: 2.78991

Mean cross-validation score: 0.98097

	Feature	Importance
0	WB_CC_EST_avg	0.9417
37	CC_EST_prev	0.0209
8	SP_POP_0014_MA_ZS	0.0031
18	SP_POP_0014_FE_ZS	0.0029
29	SP_POP_1519_MA_5Y	0.0023
23	SP_DYN_CBRT_IN	0.0018
26	SP_POP_65UP_TO_ZS	0.0018
20	SP_DYN_T065_MA_ZS	0.0014
34	SP_DYN_IMRT_MA_IN	0.0013
31	SP_POP_1519_FE_5Y	0.0012

28	SH_DYN_NMRT	0.0011
16	SP_POP_7074_MA_5Y	0.0011
21	SP_POP_0509_FE_5Y	0.0010
17	SP_POP_5559_MA_5Y	0.0010
7	SP_DYN_LE00_FE_IN	0.0009
32	SP_POP_5054_FE_5Y	0.0009
30	SP_POP_DPND_YG	0.0009
35	SP_POP_65UP_FE_ZS	0.0009
36	NV_SRV_TOTL_ZS	0.0009
24	SP_POP_6064_MA_5Y	0.0009
33	SP_POP_80UP_FE_5Y	0.0008
19	IT_MLT_MAIN_P2	0.0008
3	NY_GDP_PCAP_KD	0.0007
15	SP_POP_5054_MA_5Y	0.0007
27	SP_POP_7579_MA_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
13	SP_POP_0004_MA_5Y	0.0007
6	SP_DYN_LE00_IN	0.0007
5	SP_DYN_LE00_MA_IN	0.0007
4	NY_GDP_PCAP_CD	0.0007
2	NY_GDP_PCAP_KD_rel	0.0007
25	SP_POP_0004_FE_5Y	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
11	SP_POP_1014_MA_5Y	0.0006
9	SP_POP_0509_MA_5Y	0.0006
22	SP_POP_1014_FE_5Y	0.0005
1	NY_GNP_PCAP_CD	0.0005
10	SP_POP_0014_TO_ZS	0.0005

Model with 39 variables and max depth None:

Training+Validation R^2: 0.99863, RMSE: 0.73675

Testing R^2: 0.9809, RMSE: 2.7898

Mean cross-validation score: 0.98081

	Feature	Importance
0	WB_CC_EST_avg	0.9495
38	CC_EST_prev	0.0183
18	SP_POP_0014_FE_ZS	0.0030
8	SP_POP_0014_MA_ZS	0.0029
29	SP_POP_1519_MA_5Y	0.0018
23	SP_DYN_CBRT_IN	0.0013
26	SP_POP_65UP_TO_ZS	0.0013
20	SP_DYN_T065_MA_ZS	0.0011
37	SP_DYN_IMRT_IN	0.0011
31	SP_POP_1519_FE_5Y	0.0010
30	SP_POP_DPND_YG	0.0009
32	SP_POP_5054_FE_5Y	0.0009

34	SP_DYN_IMRT_MA_IN	0.0009
21	SP_POP_0509_FE_5Y	0.0009
27	SP_POP_7579_MA_5Y	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
24	SP_POP_6064_MA_5Y	0.0008
3	NY_GDP_PCAP_KD	0.0007
28	SH_DYN_NMRT	0.0007
36	NV_SRV_TOTL_ZS	0.0007
19	IT_MLT_MAIN_P2	0.0007
17	SP_POP_5559_MA_5Y	0.0007
16	SP_POP_7074_MA_5Y	0.0007
13	SP_POP_0004_MA_5Y	0.0007
2	NY_GDP_PCAP_KD_rel	0.0006
25	SP_POP_0004_FE_5Y	0.0006
15	SP_POP_5054_MA_5Y	0.0006
22	SP_POP_1014_FE_5Y	0.0006
5	SP_DYN_LE00_MA_IN	0.0006
33	SP_POP_80UP_FE_5Y	0.0006
35	SP_POP_65UP_FE_ZS	0.0005
14	SP_POP_6569_MA_5Y	0.0005
9	SP_POP_0509_MA_5Y	0.0005
6	SP_DYN_LE00_IN	0.0005
4	NY_GDP_PCAP_CD	0.0004
1	NY_GNP_PCAP_CD	0.0004
10	SP_POP_0014_TO_ZS	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
11	SP_POP_1014_MA_5Y	0.0004

Model with 40 variables and max depth None:

Training+Validation R^2: 0.99747, RMSE: 1.00263

Testing R^2: 0.98071, RMSE: 2.80354

Mean cross-validation score: 0.98075

	Feature	Importance
0	WB_CC_EST_avg	0.9450
39	CC_EST_prev	0.0207
18	SP_POP_0014_FE_ZS	0.0038
8	SP_POP_0014_MA_ZS	0.0024
29	SP_POP_1519_MA_5Y	0.0019
23	SP_DYN_CBRT_IN	0.0015
20	SP_DYN_TO65_MA_ZS	0.0012
31	SP_POP_1519_FE_5Y	0.0011
26	SP_POP_65UP_TO_ZS	0.0011
38	SP_POP_DPND_DL	0.0010
16	SP_POP_7074_MA_5Y	0.0009
17	SP_POP_5559_MA_5Y	0.0009
7	SP_DYN_LE00_FE_IN	0.0009

34	SP_DYN_IMRT_MA_IN	0.0009
36	NV_SRV_TOTL_ZS	0.0008
32	SP_POP_5054_FE_5Y	0.0008
19	IT_MLT_MAIN_P2	0.0008
28	SH_DYN_NMRT	0.0008
27	SP_POP_7579_MA_5Y	0.0008
37	SP_DYN_IMRT_IN	0.0008
24	SP_POP_6064_MA_5Y	0.0007
30	SP_POP_DPND_YG	0.0007
33	SP_POP_80UP_FE_5Y	0.0007
35	SP_POP_65UP_FE_ZS	0.0007
21	SP_POP_0509_FE_5Y	0.0007
3	NY_GDP_PCAP_KD	0.0007
14	SP_POP_6569_MA_5Y	0.0007
13	SP_POP_0004_MA_5Y	0.0007
25	SP_POP_0004_FE_5Y	0.0006
15	SP_POP_5054_MA_5Y	0.0006
10	SP_POP_0014_TO_ZS	0.0006
6	SP_DYN_LE00_IN	0.0006
5	SP_DYN_LE00_MA_IN	0.0006
22	SP_POP_1014_FE_5Y	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
11	SP_POP_1014_MA_5Y	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
1	NY_GNP_PCAP_CD	0.0004
9	SP_POP_0509_MA_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0004

Model with 41 variables and max depth None:

Training+Validation R^2: 0.99835, RMSE: 0.80965

Testing R^2: 0.98136, RMSE: 2.75559

Mean cross-validation score: 0.98038

	Feature	Importance
0	WB_CC_EST_avg	0.9504
40	CC_EST_prev	0.0172
8	SP_POP_0014_MA_ZS	0.0020
29	SP_POP_1519_MA_5Y	0.0018
20	SP_DYN_TO65_MA_ZS	0.0014
23	SP_DYN_CBRT_IN	0.0014
7	SP_DYN_LE00_FE_IN	0.0012
32	SP_POP_5054_FE_5Y	0.0011
37	SP_DYN_IMRT_IN	0.0010
35	SP_POP_65UP_FE_ZS	0.0010
34	SP_DYN_IMRT_MA_IN	0.0010
39	SP_ADO_TFRT	0.0010
26	SP_POP_65UP_TO_ZS	0.0010

30	SP_POP_DPND_YG	0.0010
16	SP_POP_7074_MA_5Y	0.0009
24	SP_POP_6064_MA_5Y	0.0009
17	SP_POP_5559_MA_5Y	0.0009
31	SP_POP_1519_FE_5Y	0.0009
36	NV_SRV_TOTL_ZS	0.0008
18	SP_POP_0014_FE_ZS	0.0008
19	IT_MLT_MAIN_P2	0.0008
13	SP_POP_0004_MA_5Y	0.0008
11	SP_POP_1014_MA_5Y	0.0008
38	SP_POP_DPND_OL	0.0007
28	SH_DYN_NMRT	0.0007
27	SP_POP_7579_MA_5Y	0.0007
21	SP_POP_0509_FE_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
6	SP_DYN_LE00_IN	0.0007
3	NY_GDP_PCAP_KD	0.0006
10	SP_POP_0014_TO_ZS	0.0006
25	SP_POP_0004_FE_5Y	0.0005
22	SP_POP_1014_FE_5Y	0.0005
15	SP_POP_5054_MA_5Y	0.0005
33	SP_POP_80UP_FE_5Y	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
4	NY_GDP_PCAP_CD	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
5	SP_DYN_LE00_MA_IN	0.0004
1	NY_GNP_PCAP_CD	0.0003
9	SP_POP_0509_MA_5Y	0.0003

Model with 42 variables and max depth None:

Training+Validation R^2: 0.99861, RMSE: 0.74217

Testing R^2: 0.98113, RMSE: 2.77279

Mean cross-validation score: 0.98062

	Feature	Importance
0	WB_CC_EST_avg	0.9470
41	CC_EST_prev	0.0170
8	SP_POP_0014_MA_ZS	0.0028
26	SP_POP_65UP_TO_ZS	0.0024
29	SP_POP_1519_MA_5Y	0.0022
20	SP_DYN_TO65_MA_ZS	0.0013
13	SP_POP_0004_MA_5Y	0.0013
38	SP_POP_DPND_OL	0.0011
33	SP_POP_80UP_FE_5Y	0.0011
34	SP_DYN_IMRT_MA_IN	0.0010
40	SP_POP_5559_FE_5Y	0.0010
23	SP_DYN_CBRT_IN	0.0010

31	SP_POP_1519_FE_5Y	0.0010
32	SP_POP_5054_FE_5Y	0.0010
22	SP_POP_1014_FE_5Y	0.0009
35	SP_POP_65UP_FE_ZS	0.0009
36	NV_SRV_TOTL_ZS	0.0009
15	SP_POP_5054_MA_5Y	0.0008
17	SP_POP_5559_MA_5Y	0.0008
10	SP_POP_0014_TO_ZS	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
39	SP_ADO_TFRT	0.0008
24	SP_POP_6064_MA_5Y	0.0008
25	SP_POP_0004_FE_5Y	0.0008
14	SP_POP_6569_MA_5Y	0.0008
27	SP_POP_7579_MA_5Y	0.0007
37	SP_DYN_IMRT_IN	0.0007
21	SP_POP_0509_FE_5Y	0.0007
19	IT_MLT_MAIN_P2	0.0007
18	SP_POP_0014_FE_ZS	0.0007
16	SP_POP_7074_MA_5Y	0.0007
28	SH_DYN_NMRT	0.0006
3	NY_GDP_PCAP_KD	0.0006
11	SP_POP_1014_MA_5Y	0.0006
6	SP_DYN_LE00_IN	0.0006
5	SP_DYN_LE00_MA_IN	0.0006
30	SP_POP_DPND_YG	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
1	NY_GNP_PCAP_CD	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
9	SP_POP_0509_MA_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0004

Model with 43 variables and max depth None:

Training+Validation R^2: 0.99667, RMSE: 1.14972

Testing R^2: 0.98081, RMSE: 2.79624

Mean cross-validation score: 0.98085

	Feature	Importance
0	WB_CC_EST_avg	0.9380
42	CC_EST_prev	0.0207
8	SP_POP_0014_MA_ZS	0.0034
29	SP_POP_1519_MA_5Y	0.0023
13	SP_POP_0004_MA_5Y	0.0016
26	SP_POP_65UP_TO_ZS	0.0014
40	SP_POP_5559_FE_5Y	0.0013
32	SP_POP_5054_FE_5Y	0.0013
33	SP_POP_80UP_FE_5Y	0.0012
20	SP_DYN_TO65_MA_ZS	0.0012

31	SP_POP_1519_FE_5Y	0.0012
18	SP_POP_0014_FE_ZS	0.0011
22	SP_POP_1014_FE_5Y	0.0011
34	SP_DYN_IMRT_MA_IN	0.0011
36	NV_SRV_TOTL_ZS	0.0011
23	SP_DYN_CBRT_IN	0.0011
14	SP_POP_6569_MA_5Y	0.0010
35	SP_POP_65UP_FE_ZS	0.0010
17	SP_POP_5559_MA_5Y	0.0010
38	SP_POP_DPND_DL	0.0010
25	SP_POP_0004_FE_5Y	0.0010
24	SP_POP_6064_MA_5Y	0.0009
39	SP_ADO_TFRT	0.0009
41	SP_DYN_IMRT_FE_IN	0.0009
21	SP_POP_0509_FE_5Y	0.0009
10	SP_POP_0014_TO_ZS	0.0009
16	SP_POP_7074_MA_5Y	0.0009
7	SP_DYN_LEOO_FE_IN	0.0009
27	SP_POP_7579_MA_5Y	0.0008
19	IT_MLT_MAIN_P2	0.0008
15	SP_POP_5054_MA_5Y	0.0008
11	SP_POP_1014_MA_5Y	0.0007
6	SP_DYN_LEOO_IN	0.0007
37	SP_DYN_IMRT_IN	0.0007
5	SP_DYN_LEOO_MA_IN	0.0006
3	NY_GDP_PCAP_KD	0.0006
30	SP_POP_DPND_YG	0.0006
28	SH_DYN_NMRT	0.0006
2	NY_GDP_PCAP_KD_rel	0.0006
9	SP_POP_0509_MA_5Y	0.0005
1	NY_GNP_PCAP_CD	0.0005
4	NY_GDP_PCAP_CD	0.0005
12	SP_POP_65UP_MA_ZS	0.0004

Model with 44 variables and max depth None:

Training+Validation R^2: 0.99751, RMSE: 0.99516

Testing R^2: 0.9806, RMSE: 2.81094

Mean cross-validation score: 0.98083

	Feature	Importance
0	WB_CC_EST_avg	0.9432
43	CC_EST_prev	0.0178
8	SP_POP_0014_MA_ZS	0.0032
29	SP_POP_1519_MA_5Y	0.0025
26	SP_POP_65UP_TO_ZS	0.0013
13	SP_POP_0004_MA_5Y	0.0013
33	SP_POP_80UP_FE_5Y	0.0013

20	SP_DYN_T065_MA_ZS	0.0012
31	SP_POP_1519_FE_5Y	0.0012
32	SP_POP_5054_FE_5Y	0.0012
23	SP_DYN_CBRT_IN	0.0012
40	SP_POP_5559_FE_5Y	0.0012
36	NV_SRV_TOTL_ZS	0.0010
41	SP_DYN_IMRT_FE_IN	0.0010
10	SP_POP_0014_TO_ZS	0.0010
34	SP_DYN_IMRT_MA_IN	0.0009
35	SP_POP_65UP_FE_ZS	0.0009
39	SP_ADO_TFRT	0.0009
30	SP_POP_DPND_YG	0.0009
22	SP_POP_1014_FE_5Y	0.0009
7	SP_DYN_LE00_FE_IN	0.0009
14	SP_POP_6569_MA_5Y	0.0008
16	SP_POP_7074_MA_5Y	0.0008
38	SP_POP_DPND_DL	0.0008
21	SP_POP_0509_FE_5Y	0.0008
27	SP_POP_7579_MA_5Y	0.0008
17	SP_POP_5559_MA_5Y	0.0008
42	SP_POP_80UP_MA_5Y	0.0007
11	SP_POP_1014_MA_5Y	0.0007
15	SP_POP_5054_MA_5Y	0.0007
19	IT_MLT_MAIN_P2	0.0007
25	SP_POP_0004_FE_5Y	0.0007
24	SP_POP_6064_MA_5Y	0.0007
2	NY_GDP_PCAP_KD_rel	0.0007
28	SH_DYN_NMRT	0.0006
9	SP_POP_0509_MA_5Y	0.0006
37	SP_DYN_IMRT_IN	0.0006
6	SP_DYN_LE00_IN	0.0006
5	SP_DYN_LE00_MA_IN	0.0006
4	NY_GDP_PCAP_CD	0.0006
18	SP_POP_0014_FE_ZS	0.0005
3	NY_GDP_PCAP_KD	0.0005
1	NY_GNP_PCAP_CD	0.0005
12	SP_POP_65UP_MA_ZS	0.0003

Model with 45 variables and max depth None:

Training+Validation R^2: 0.99716, RMSE: 1.06242

Testing R^2: 0.98085, RMSE: 2.7929

Mean cross-validation score: 0.98071

	Feature	Importance
0	WB_CC_EST_avg	0.9416
44	CC_EST_prev	0.0181
8	SP_POP_0014_MA_ZS	0.0027

29	SP_POP_1519_MA_5Y	0.0021
33	SP_POP_80UP_FE_5Y	0.0016
26	SP_POP_65UP_TO_ZS	0.0014
23	SP_DYN_CBRT_IN	0.0013
13	SP_POP_0004_MA_5Y	0.0013
32	SP_POP_5054_FE_5Y	0.0013
25	SP_POP_0004_FE_5Y	0.0013
40	SP_POP_5559_FE_5Y	0.0012
22	SP_POP_1014_FE_5Y	0.0012
38	SP_POP_DPND_DL	0.0011
35	SP_POP_65UP_FE_ZS	0.0011
31	SP_POP_1519_FE_5Y	0.0011
20	SP_DYN_TO65_MA_ZS	0.0011
43	SP_DYN_TO65_FE_ZS	0.0011
36	NV_SRV_TOTL_ZS	0.0010
17	SP_POP_5559_MA_5Y	0.0010
18	SP_POP_0014_FE_ZS	0.0010
39	SP_ADO_TFRT	0.0009
34	SP_DYN_IMRT_MA_IN	0.0009
7	SP_DYN_LE00_FE_IN	0.0008
27	SP_POP_7579_MA_5Y	0.0008
41	SP_DYN_IMRT_FE_IN	0.0008
10	SP_POP_0014_TO_ZS	0.0008
24	SP_POP_6064_MA_5Y	0.0008
42	SP_POP_80UP_MA_5Y	0.0008
21	SP_POP_0509_FE_5Y	0.0008
19	IT_MLT_MAIN_P2	0.0008
16	SP_POP_7074_MA_5Y	0.0008
5	SP_DYN_LE00_MA_IN	0.0007
15	SP_POP_5054_MA_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
11	SP_POP_1014_MA_5Y	0.0006
30	SP_POP_DPND_YG	0.0006
6	SP_DYN_LE00_IN	0.0006
2	NY_GDP_PCAP_KD_rel	0.0006
28	SH_DYN_NMRT	0.0005
9	SP_POP_0509_MA_5Y	0.0005
3	NY_GDP_PCAP_KD	0.0005
37	SP_DYN_IMRT_IN	0.0004
4	NY_GDP_PCAP_CD	0.0004
1	NY_GNP_PCAP_CD	0.0004
12	SP_POP_65UP_MA_ZS	0.0002

Model with 46 variables and max depth None:

Training+Validation R^2: 0.99729, RMSE: 1.03843

Testing R^2: 0.98114, RMSE: 2.77197

Mean cross-validation score: 0.98103

	Feature	Importance
0	WB_CC_EST_avg	0.9290
45	CC_EST_prev	0.0240
18	SP_POP_0014_FE_ZS	0.0038
8	SP_POP_0014_MA_ZS	0.0031
29	SP_POP_1519_MA_5Y	0.0028
40	SP_POP_5559_FE_5Y	0.0017
20	SP_DYN_T065_MA_ZS	0.0015
34	SP_DYN_IMRT_MA_IN	0.0014
26	SP_POP_65UP_TO_ZS	0.0014
38	SP_POP_DPND_DL	0.0012
35	SP_POP_65UP_FE_ZS	0.0012
17	SP_POP_5559_MA_5Y	0.0011
22	SP_POP_1014_FE_5Y	0.0011
42	SP_POP_80UP_MA_5Y	0.0011
21	SP_POP_0509_FE_5Y	0.0011
39	SP_ADO_TFRT	0.0011
19	IT_MLT_MAIN_P2	0.0011
23	SP_DYN_CBRT_IN	0.0010
33	SP_POP_80UP_FE_5Y	0.0010
36	NV_SRV_TOTL_ZS	0.0010
41	SP_DYN_IMRT_FE_IN	0.0010
44	SP_POP_4549_MA_5Y	0.0010
25	SP_POP_0004_FE_5Y	0.0010
32	SP_POP_5054_FE_5Y	0.0009
43	SP_DYN_T065_FE_ZS	0.0009
31	SP_POP_1519_FE_5Y	0.0009
24	SP_POP_6064_MA_5Y	0.0009
11	SP_POP_1014_MA_5Y	0.0009
7	SP_DYN_LEOO_FE_IN	0.0009
3	NY_GDP_PCAP_KD	0.0008
16	SP_POP_7074_MA_5Y	0.0008
14	SP_POP_6569_MA_5Y	0.0008
28	SH_DYN_NMRT	0.0008
37	SP_DYN_IMRT_IN	0.0008
2	NY_GDP_PCAP_KD_rel	0.0007
9	SP_POP_0509_MA_5Y	0.0007
13	SP_POP_0004_MA_5Y	0.0007
4	NY_GDP_PCAP_CD	0.0006
5	SP_DYN_LEOO_MA_IN	0.0006
6	SP_DYN_LEOO_IN	0.0006
27	SP_POP_7579_MA_5Y	0.0006
1	NY_GNP_PCAP_CD	0.0006
15	SP_POP_5054_MA_5Y	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
10	SP_POP_0014_TO_ZS	0.0004
30	SP_POP_DPND_YG	0.0003

Model with 47 variables and max depth None:
 Training+Validation R^2: 0.99889, RMSE: 0.66338
 Testing R^2: 0.9815, RMSE: 2.74523
 Mean cross-validation score: 0.98109

	Feature	Importance
0	WB_CC_EST_avg	0.9372
46	CC_EST_prev	0.0180
18	SP_POP_0014_FE_ZS	0.0034
8	SP_POP_0014_MA_ZS	0.0025
29	SP_POP_1519_MA_5Y	0.0023
40	SP_POP_5559_FE_5Y	0.0018
23	SP_DYN_CBRT_IN	0.0017
37	SP_DYN_IMRT_IN	0.0013
35	SP_POP_65UP_FE_ZS	0.0012
25	SP_POP_0004_FE_5Y	0.0012
7	SP_DYN_LE00_FE_IN	0.0012
39	SP_ADO_TFRT	0.0012
20	SP_DYN_T065_MA_ZS	0.0012
26	SP_POP_65UP_TO_ZS	0.0010
31	SP_POP_1519_FE_5Y	0.0010
34	SP_DYN_IMRT_MA_IN	0.0010
30	SP_POP_DPND_YG	0.0010
36	NV_SRV_TOTL_ZS	0.0010
42	SP_POP_80UP_MA_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0010
16	SP_POP_7074_MA_5Y	0.0009
19	IT_MLT_MAIN_P2	0.0009
38	SP_POP_DPND_DL	0.0009
21	SP_POP_0509_FE_5Y	0.0009
43	SP_DYN_T065_FE_ZS	0.0009
15	SP_POP_5054_MA_5Y	0.0009
33	SP_POP_80UP_FE_5Y	0.0008
44	SP_POP_4549_MA_5Y	0.0008
24	SP_POP_6064_MA_5Y	0.0008
28	SH_DYN_NMRT	0.0008
11	SP_POP_1014_MA_5Y	0.0008
17	SP_POP_5559_MA_5Y	0.0007
6	SP_DYN_LE00_IN	0.0007
27	SP_POP_7579_MA_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
2	NY_GDP_PCAP_KD_rel	0.0007
22	SP_POP_1014_FE_5Y	0.0007
32	SP_POP_5054_FE_5Y	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
3	NY_GDP_PCAP_KD	0.0006

5	SP_DYN_LE00_MA_IN	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
13	SP_POP_0004_MA_5Y	0.0005
1	NY_GNP_PCAP_CD	0.0005
9	SP_POP_0509_MA_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0004
4	NY_GDP_PCAP_CD	0.0004

Model with 48 variables and max depth None:

Training+Validation R^2: 0.99738, RMSE: 1.01959

Testing R^2: 0.98047, RMSE: 2.82092

Mean cross-validation score: 0.98094

	Feature	Importance
0	WB_CC_EST_avg	0.9357
47	CC_EST_prev	0.0176
18	SP_POP_0014_FE_ZS	0.0050
8	SP_POP_0014_MA_ZS	0.0026
29	SP_POP_1519_MA_5Y	0.0023
23	SP_DYN_CBRT_IN	0.0016
20	SP_DYN_T065_MA_ZS	0.0016
40	SP_POP_5559_FE_5Y	0.0014
33	SP_POP_80UP_FE_5Y	0.0013
26	SP_POP_65UP_TO_ZS	0.0013
39	SP_ADO_TFRT	0.0012
34	SP_DYN_IMRT_MA_IN	0.0012
19	IT_MLT_MAIN_P2	0.0011
45	SP_POP_6569_FE_5Y	0.0011
41	SP_DYN_IMRT_FE_IN	0.0011
32	SP_POP_5054_FE_5Y	0.0010
30	SP_POP_DPND_YG	0.0010
21	SP_POP_0509_FE_5Y	0.0010
46	SP_POP_7074_FE_5Y	0.0010
24	SP_POP_6064_MA_5Y	0.0009
16	SP_POP_7074_MA_5Y	0.0009
28	SH_DYN_NMRT	0.0009
11	SP_POP_1014_MA_5Y	0.0009
7	SP_DYN_LE00_FE_IN	0.0009
42	SP_POP_80UP_MA_5Y	0.0008
43	SP_DYN_T065_FE_ZS	0.0008
36	NV_SRV_TOTL_ZS	0.0008
25	SP_POP_0004_FE_5Y	0.0008
15	SP_POP_5054_MA_5Y	0.0008
38	SP_POP_DPND_DL	0.0007
10	SP_POP_0014_TO_ZS	0.0007
4	NY_GDP_PCAP_CD	0.0007
31	SP_POP_1519_FE_5Y	0.0007

22	SP_POP_1014_FE_5Y	0.0007
17	SP_POP_5559_MA_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
37	SP_DYN_IMRT_IN	0.0007
44	SP_POP_4549_MA_5Y	0.0007
13	SP_POP_0004_MA_5Y	0.0007
5	SP_DYN_LE00_MA_IN	0.0006
6	SP_DYN_LE00_IN	0.0006
27	SP_POP_7579_MA_5Y	0.0006
3	NY_GDP_PCAP_KD	0.0006
9	SP_POP_0509_MA_5Y	0.0005
1	NY_GNP_PCAP_CD	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
12	SP_POP_65UP_MA_ZS	0.0004
35	SP_POP_65UP_FE_ZS	0.0004

Model with 49 variables and max depth None:

Training+Validation R^2: 0.99699, RMSE: 1.09281

Testing R^2: 0.98094, RMSE: 2.78672

Mean cross-validation score: 0.98101

	Feature	Importance
0	WB_CC_EST_avg	0.9374
48	CC_EST_prev	0.0165
18	SP_POP_0014_FE_ZS	0.0037
8	SP_POP_0014_MA_ZS	0.0031
29	SP_POP_1519_MA_5Y	0.0025
20	SP_DYN_T065_MA_ZS	0.0020
40	SP_POP_5559_FE_5Y	0.0018
41	SP_DYN_IMRT_FE_IN	0.0014
34	SP_DYN_IMRT_MA_IN	0.0014
7	SP_DYN_LE00_FE_IN	0.0013
33	SP_POP_80UP_FE_5Y	0.0012
23	SP_DYN_CBRT_IN	0.0012
25	SP_POP_0004_FE_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0010
43	SP_DYN_T065_FE_ZS	0.0010
39	SP_ADO_TFRT	0.0010
24	SP_POP_6064_MA_5Y	0.0010
46	SP_POP_7074_FE_5Y	0.0009
21	SP_POP_0509_FE_5Y	0.0009
38	SP_POP_DPND_DL	0.0009
36	NV_SRV_TOTL_ZS	0.0009
19	IT_MLT_MAIN_P2	0.0009
44	SP_POP_4549_MA_5Y	0.0008
37	SP_DYN_IMRT_IN	0.0008
31	SP_POP_1519_FE_5Y	0.0008

32	SP_POP_5054_FE_5Y	0.0008
11	SP_POP_1014_MA_5Y	0.0008
15	SP_POP_5054_MA_5Y	0.0007
22	SP_POP_1014_FE_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
47	SP_POP_6064_FE_5Y	0.0007
26	SP_POP_65UP_TO_ZS	0.0007
28	SH_DYN_NMRT	0.0007
10	SP_POP_0014_TO_ZS	0.0007
30	SP_POP_DPND_YG	0.0007
35	SP_POP_65UP_FE_ZS	0.0007
17	SP_POP_5559_MA_5Y	0.0006
13	SP_POP_0004_MA_5Y	0.0006
16	SP_POP_7074_MA_5Y	0.0006
42	SP_POP_80UP_MA_5Y	0.0006
3	NY_GDP_PCAP_KD	0.0006
5	SP_DYN_LE00_MA_IN	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
6	SP_DYN_LE00_IN	0.0005
9	SP_POP_0509_MA_5Y	0.0005
4	NY_GDP_PCAP_CD	0.0004
1	NY_GNP_PCAP_CD	0.0004
12	SP_POP_65UP_MA_ZS	0.0003
27	SP_POP_7579_MA_5Y	0.0002

Model with 50 variables and max depth None:

Training+Validation R^2: 0.99633, RMSE: 1.20836

Testing R^2: 0.98057, RMSE: 2.81339

Mean cross-validation score: 0.9807

	Feature	Importance
0	WB_CC_EST_avg	0.9419
49	CC_EST_prev	0.0170
18	SP_POP_0014_FE_ZS	0.0024
8	SP_POP_0014_MA_ZS	0.0024
29	SP_POP_1519_MA_5Y	0.0020
40	SP_POP_5559_FE_5Y	0.0014
20	SP_DYN_TO65_MA_ZS	0.0013
23	SP_DYN_CBRT_IN	0.0012
48	SH_DYN_MORT_MA	0.0011
21	SP_POP_0509_FE_5Y	0.0011
38	SP_POP_DPND_DL	0.0011
34	SP_DYN_IMRT_MA_IN	0.0011
45	SP_POP_6569_FE_5Y	0.0010
41	SP_DYN_IMRT_FE_IN	0.0010
37	SP_DYN_IMRT_IN	0.0010
32	SP_POP_5054_FE_5Y	0.0010

39	SP_ADO_TFRT	0.0009
31	SP_POP_1519_FE_5Y	0.0009
24	SP_POP_6064_MA_5Y	0.0009
7	SP_DYN_LE00_FE_IN	0.0009
46	SP_POP_7074_FE_5Y	0.0009
33	SP_POP_80UP_FE_5Y	0.0008
43	SP_DYN_T065_FE_ZS	0.0008
47	SP_POP_6064_FE_5Y	0.0008
26	SP_POP_65UP_TO_ZS	0.0008
25	SP_POP_0004_FE_5Y	0.0008
19	IT_MLT_MAIN_P2	0.0008
3	NY_GDP_PCAP_KD	0.0007
11	SP_POP_1014_MA_5Y	0.0007
16	SP_POP_7074_MA_5Y	0.0007
17	SP_POP_5559_MA_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
28	SH_DYN_NMRT	0.0007
30	SP_POP_DPND_YG	0.0007
4	NY_GDP_PCAP_CD	0.0006
6	SP_DYN_LE00_IN	0.0006
36	NV_SRV_TOTL_ZS	0.0006
15	SP_POP_5054_MA_5Y	0.0006
44	SP_POP_4549_MA_5Y	0.0006
13	SP_POP_0004_MA_5Y	0.0006
42	SP_POP_80UP_MA_5Y	0.0006
22	SP_POP_1014_FE_5Y	0.0005
35	SP_POP_65UP_FE_ZS	0.0005
1	NY_GNP_PCAP_CD	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
5	SP_DYN_LE00_MA_IN	0.0004
27	SP_POP_7579_MA_5Y	0.0003
9	SP_POP_0509_MA_5Y	0.0003
12	SP_POP_65UP_MA_ZS	0.0002
10	SP_POP_0014_TO_ZS	0.0002

Model with 51 variables and max depth None:

Training+Validation R^2: 0.99723, RMSE: 1.04932

Testing R^2: 0.98065, RMSE: 2.80766

Mean cross-validation score: 0.98072

	Feature	Importance
0	WB_CC_EST_avg	0.9496
50	CC_EST_prev	0.0145
18	SP_POP_0014_FE_ZS	0.0024
8	SP_POP_0014_MA_ZS	0.0021
29	SP_POP_1519_MA_5Y	0.0020
20	SP_DYN_T065_MA_ZS	0.0013

40	SP_POP_5559_FE_5Y	0.0012
32	SP_POP_5054_FE_5Y	0.0010
23	SP_DYN_CBRT_IN	0.0010
34	SP_DYN_IMRT_MA_IN	0.0010
49	SH_DYN_MORT	0.0009
21	SP_POP_0509_FE_5Y	0.0008
31	SP_POP_1519_FE_5Y	0.0008
24	SP_POP_6064_MA_5Y	0.0008
46	SP_POP_7074_FE_5Y	0.0008
39	SP_ADO_TFRT	0.0008
48	SH_DYN_MORT_MA	0.0008
41	SP_DYN_IMRT_FE_IN	0.0007
38	SP_POP_DPND_DL	0.0007
33	SP_POP_80UP_FE_5Y	0.0007
37	SP_DYN_IMRT_IN	0.0007
43	SP_DYN_TO65_FE_ZS	0.0007
11	SP_POP_1014_MA_5Y	0.0007
45	SP_POP_6569_FE_5Y	0.0007
47	SP_POP_6064_FE_5Y	0.0007
30	SP_POP_DPND_YG	0.0006
28	SH_DYN_NMRT	0.0006
25	SP_POP_0004_FE_5Y	0.0006
3	NY_GDP_PCAP_KD	0.0006
22	SP_POP_1014_FE_5Y	0.0006
16	SP_POP_7074_MA_5Y	0.0006
15	SP_POP_5054_MA_5Y	0.0006
14	SP_POP_6569_MA_5Y	0.0006
44	SP_POP_4549_MA_5Y	0.0006
7	SP_DYN_LE00_FE_IN	0.0006
19	IT_MLT_MAIN_P2	0.0006
42	SP_POP_80UP_MA_5Y	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
36	NV_SRV_TOTL_ZS	0.0005
17	SP_POP_5559_MA_5Y	0.0005
13	SP_POP_0004_MA_5Y	0.0005
4	NY_GDP_PCAP_CD	0.0005
1	NY_GNP_PCAP_CD	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
6	SP_DYN_LE00_IN	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
35	SP_POP_65UP_FE_ZS	0.0003
27	SP_POP_7579_MA_5Y	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
10	SP_POP_0014_TO_ZS	0.0002
9	SP_POP_0509_MA_5Y	0.0002

Model with 52 variables and max depth None:

Training+Validation R^2: 0.9968, RMSE: 1.12759
 Testing R^2: 0.98122, RMSE: 2.76605
 Mean cross-validation score: 0.9811

	Feature	Importance
0	WB_CC_EST_avg	0.9335
51	CC_EST_prev	0.0173
8	SP_POP_0014_MA_ZS	0.0040
49	SH_DYN_MORT	0.0027
29	SP_POP_1519_MA_5Y	0.0025
18	SP_POP_0014_FE_ZS	0.0025
40	SP_POP_5559_FE_5Y	0.0015
45	SP_POP_6569_FE_5Y	0.0015
20	SP_DYN_T065_MA_ZS	0.0014
21	SP_POP_0509_FE_5Y	0.0014
50	SP_DYN_TFRT_IN	0.0013
32	SP_POP_5054_FE_5Y	0.0012
34	SP_DYN_IMRT_MA_IN	0.0012
46	SP_POP_7074_FE_5Y	0.0011
7	SP_DYN_LE00_FE_IN	0.0011
39	SP_ADO_TFRT	0.0011
35	SP_POP_65UP_FE_ZS	0.0011
48	SH_DYN_MORT_MA	0.0010
11	SP_POP_1014_MA_5Y	0.0010
31	SP_POP_1519_FE_5Y	0.0010
24	SP_POP_6064_MA_5Y	0.0009
16	SP_POP_7074_MA_5Y	0.0009
44	SP_POP_4549_MA_5Y	0.0009
19	IT_MLT_MAIN_P2	0.0009
43	SP_DYN_T065_FE_ZS	0.0009
38	SP_POP_DPNP_OL	0.0009
47	SP_POP_6064_FE_5Y	0.0009
23	SP_DYN_CBRT_IN	0.0008
42	SP_POP_80UP_MA_5Y	0.0008
41	SP_DYN_IMRT_FE_IN	0.0008
36	NV_SRV_TOTL_ZS	0.0007
26	SP_POP_65UP_TO_ZS	0.0007
28	SH_DYN_NMRT	0.0007
3	NY_GDP_PCAP_KD	0.0007
25	SP_POP_0004_FE_5Y	0.0007
15	SP_POP_5054_MA_5Y	0.0007
33	SP_POP_80UP_FE_5Y	0.0006
1	NY_GNP_PCAP_CD	0.0006
14	SP_POP_6569_MA_5Y	0.0006
6	SP_DYN_LE00_IN	0.0006
4	NY_GDP_PCAP_CD	0.0006
22	SP_POP_1014_FE_5Y	0.0006
2	NY_GDP_PCAP_KD_rel	0.0005

17	SP_POP_5559_MA_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0005
27	SP_POP_7579_MA_5Y	0.0005
30	SP_POP_DPND_YG	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
5	SP_DYN_LE00_MA_IN	0.0004
13	SP_POP_0004_MA_5Y	0.0003
9	SP_POP_0509_MA_5Y	0.0003
37	SP_DYN_IMRT_IN	0.0002

Model with 53 variables and max depth None:

Training+Validation R^2: 0.99818, RMSE: 0.85103

Testing R^2: 0.98094, RMSE: 2.78629

Mean cross-validation score: 0.98083

	Feature	Importance
0	WB_CC_EST_avg	0.9242
52	CC_EST_prev	0.0221
8	SP_POP_0014_MA_ZS	0.0044
29	SP_POP_1519_MA_5Y	0.0025
20	SP_DYN_T065_MA_ZS	0.0022
18	SP_POP_0014_FE_ZS	0.0022
51	SP_POP_4549_FE_5Y	0.0018
38	SP_POP_DPND_OL	0.0016
45	SP_POP_6569_FE_5Y	0.0014
21	SP_POP_0509_FE_5Y	0.0014
33	SP_POP_80UP_FE_5Y	0.0013
25	SP_POP_0004_FE_5Y	0.0013
43	SP_DYN_T065_FE_ZS	0.0013
47	SP_POP_6064_FE_5Y	0.0012
46	SP_POP_7074_FE_5Y	0.0012
32	SP_POP_5054_FE_5Y	0.0011
40	SP_POP_5559_FE_5Y	0.0011
39	SP_ADO_TFRT	0.0011
34	SP_DYN_IMRT_MA_IN	0.0010
44	SP_POP_4549_MA_5Y	0.0010
19	IT_MLT_MAIN_P2	0.0010
7	SP_DYN_LE00_FE_IN	0.0010
31	SP_POP_1519_FE_5Y	0.0010
50	SP_DYN_TFRT_IN	0.0010
27	SP_POP_7579_MA_5Y	0.0009
16	SP_POP_7074_MA_5Y	0.0009
15	SP_POP_5054_MA_5Y	0.0009
24	SP_POP_6064_MA_5Y	0.0009
42	SP_POP_80UP_MA_5Y	0.0009
41	SP_DYN_IMRT_FE_IN	0.0009
30	SP_POP_DPND_YG	0.0009

49	SH_DYN_MORT	0.0009
36	NV_SRV_TOTL_ZS	0.0008
35	SP_POP_65UP_FE_ZS	0.0008
17	SP_POP_5559_MA_5Y	0.0008
23	SP_DYN_CBRT_IN	0.0008
14	SP_POP_6569_MA_5Y	0.0008
12	SP_POP_65UP_MA_ZS	0.0008
2	NY_GDP_PCAP_KD_rel	0.0007
28	SH_DYN_NMRT	0.0007
3	NY_GDP_PCAP_KD	0.0007
11	SP_POP_1014_MA_5Y	0.0007
9	SP_POP_0509_MA_5Y	0.0007
26	SP_POP_65UP_TO_ZS	0.0007
10	SP_POP_0014_TO_ZS	0.0006
48	SH_DYN_MORT_MA	0.0006
6	SP_DYN_LEOO_IN	0.0006
1	NY_GNP_PCAP_CD	0.0005
22	SP_POP_1014_FE_5Y	0.0005
13	SP_POP_0004_MA_5Y	0.0005
4	NY_GDP_PCAP_CD	0.0005
37	SP_DYN_IMRT_IN	0.0004
5	SP_DYN_LEOO_MA_IN	0.0004

Model with 54 variables and max depth None:

Training+Validation R^2: 0.99745, RMSE: 1.00665

Testing R^2: 0.98092, RMSE: 2.78813

Mean cross-validation score: 0.9808

	Feature	Importance
0	WB_CC_EST_avg	0.9234
53	CC_EST_prev	0.0240
8	SP_POP_0014_MA_ZS	0.0036
29	SP_POP_1519_MA_5Y	0.0030
18	SP_POP_0014_FE_ZS	0.0019
51	SP_POP_4549_FE_5Y	0.0018
20	SP_DYN_T065_MA_ZS	0.0016
47	SP_POP_6064_FE_5Y	0.0013
33	SP_POP_80UP_FE_5Y	0.0013
34	SP_DYN_IMRT_MA_IN	0.0013
43	SP_DYN_T065_FE_ZS	0.0013
30	SP_POP_DPND_YG	0.0012
52	SH_DYN_MORT_FE	0.0012
26	SP_POP_65UP_TO_ZS	0.0012
40	SP_POP_5559_FE_5Y	0.0012
50	SP_DYN_TFRT_IN	0.0011
24	SP_POP_6064_MA_5Y	0.0011
21	SP_POP_0509_FE_5Y	0.0010

27	SP_POP_7579_MA_5Y	0.0010
17	SP_POP_5559_MA_5Y	0.0010
19	IT_MLT_MAIN_P2	0.0010
32	SP_POP_5054_FE_5Y	0.0010
16	SP_POP_7074_MA_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0010
7	SP_DYN_LEOO_FE_IN	0.0010
31	SP_POP_1519_FE_5Y	0.0010
39	SP_ADO_TFRT	0.0009
41	SP_DYN_IMRT_FE_IN	0.0009
38	SP_POP_DPND_DL	0.0009
42	SP_POP_80UP_MA_5Y	0.0009
44	SP_POP_4549_MA_5Y	0.0009
46	SP_POP_7074_FE_5Y	0.0009
28	SH_DYN_NMRT	0.0009
25	SP_POP_0004_FE_5Y	0.0009
15	SP_POP_5054_MA_5Y	0.0009
10	SP_POP_0014_TO_ZS	0.0009
35	SP_POP_65UP_FE_ZS	0.0008
36	NV_SRV_TOTL_ZS	0.0008
14	SP_POP_6569_MA_5Y	0.0008
49	SH_DYN_MORT	0.0007
23	SP_DYN_CBRT_IN	0.0007
22	SP_POP_1014_FE_5Y	0.0007
48	SH_DYN_MORT_MA	0.0007
2	NY_GDP_PCAP_KD_rel	0.0006
37	SP_DYN_IMRT_IN	0.0006
11	SP_POP_1014_MA_5Y	0.0006
6	SP_DYN_LEOO_IN	0.0006
4	NY_GDP_PCAP_CD	0.0005
5	SP_DYN_LEOO_MA_IN	0.0005
13	SP_POP_0004_MA_5Y	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
1	NY_GNP_PCAP_CD	0.0005
9	SP_POP_0509_MA_5Y	0.0004
3	NY_GDP_PCAP_KD	0.0004

Model with 55 variables and max depth None:

Training+Validation R^2: 0.99816, RMSE: 0.85589

Testing R^2: 0.9814, RMSE: 2.75259

Mean cross-validation score: 0.98059

	Feature	Importance
0	WB_CC_EST_avg	0.9243
54	CC_EST_prev	0.0241
8	SP_POP_0014_MA_ZS	0.0040
29	SP_POP_1519_MA_5Y	0.0034

51	SP_POP_4549_FE_5Y	0.0023
20	SP_DYN_T065_MA_ZS	0.0017
26	SP_POP_65UP_TO_ZS	0.0017
50	SP_DYN_TFRT_IN	0.0016
53	SP_POP_7579_FE_5Y	0.0013
47	SP_POP_6064_FE_5Y	0.0012
24	SP_POP_6064_MA_5Y	0.0012
41	SP_DYN_IMRT_FE_IN	0.0012
18	SP_POP_0014_FE_ZS	0.0011
44	SP_POP_4549_MA_5Y	0.0011
40	SP_POP_5559_FE_5Y	0.0011
32	SP_POP_5054_FE_5Y	0.0010
37	SP_DYN_IMRT_IN	0.0010
39	SP_ADO_TFRT	0.0010
25	SP_POP_0004_FE_5Y	0.0010
43	SP_DYN_T065_FE_ZS	0.0010
7	SP_DYN_LE00_FE_IN	0.0010
33	SP_POP_80UP_FE_5Y	0.0010
19	IT_MLT_MAIN_P2	0.0009
34	SP_DYN_IMRT_MA_IN	0.0009
21	SP_POP_0509_FE_5Y	0.0009
49	SH_DYN_MORT	0.0009
45	SP_POP_6569_FE_5Y	0.0009
31	SP_POP_1519_FE_5Y	0.0009
46	SP_POP_7074_FE_5Y	0.0009
17	SP_POP_5559_MA_5Y	0.0008
14	SP_POP_6569_MA_5Y	0.0008
42	SP_POP_80UP_MA_5Y	0.0008
36	NV_SRV_TOTL_ZS	0.0008
35	SP_POP_65UP_FE_ZS	0.0008
2	NY_GDP_PCAP_KD_rel	0.0007
23	SP_DYN_CBRT_IN	0.0007
12	SP_POP_65UP_MA_ZS	0.0007
52	SH_DYN_MORT_FE	0.0007
5	SP_DYN_LE00_MA_IN	0.0007
3	NY_GDP_PCAP_KD	0.0006
16	SP_POP_7074_MA_5Y	0.0006
15	SP_POP_5054_MA_5Y	0.0006
13	SP_POP_0004_MA_5Y	0.0006
11	SP_POP_1014_MA_5Y	0.0006
1	NY_GNP_PCAP_CD	0.0005
38	SP_POP_DPNP_DL	0.0005
22	SP_POP_1014_FE_5Y	0.0005
9	SP_POP_0509_MA_5Y	0.0005
48	SH_DYN_MORT_MA	0.0005
28	SH_DYN_NMRT	0.0004
10	SP_POP_0014_TO_ZS	0.0004
6	SP_DYN_LE00_IN	0.0004

27	SP_POP_7579_MA_5Y	0.0004
30	SP_POP_DPND_YG	0.0003
4	NY_GDP_PCAP_CD	0.0003

Model with 56 variables and max depth None:
 Training+Validation R^2: 0.99827, RMSE: 0.82931
 Testing R^2: 0.98169, RMSE: 2.73141
 Mean cross-validation score: 0.98133

	Feature	Importance
0	WB_CC_EST_avg	0.9294
55	CC_EST_prev	0.0237
8	SP_POP_0014_MA_ZS	0.0041
29	SP_POP_1519_MA_5Y	0.0024
20	SP_DYN_T065_MA_ZS	0.0017
51	SP_POP_4549_FE_5Y	0.0014
54	SP_DYN_AMRT_MA	0.0014
40	SP_POP_5559_FE_5Y	0.0013
18	SP_POP_0014_FE_ZS	0.0013
50	SP_DYN_TFRT_IN	0.0013
24	SP_POP_6064_MA_5Y	0.0012
45	SP_POP_6569_FE_5Y	0.0011
41	SP_DYN_IMRT_FE_IN	0.0011
33	SP_POP_80UP_FE_5Y	0.0011
44	SP_POP_4549_MA_5Y	0.0010
42	SP_POP_80UP_MA_5Y	0.0010
53	SP_POP_7579_FE_5Y	0.0010
39	SP_ADO_TFRT	0.0010
7	SP_DYN_LE00_FE_IN	0.0010
34	SP_DYN_IMRT_MA_IN	0.0010
32	SP_POP_5054_FE_5Y	0.0010
25	SP_POP_0004_FE_5Y	0.0009
19	IT_MLT_MAIN_P2	0.0009
31	SP_POP_1519_FE_5Y	0.0009
35	SP_POP_65UP_FE_ZS	0.0008
52	SH_DYN_MORT_FE	0.0008
17	SP_POP_5559_MA_5Y	0.0008
49	SH_DYN_MORT	0.0007
22	SP_POP_1014_FE_5Y	0.0007
43	SP_DYN_T065_FE_ZS	0.0007
14	SP_POP_6569_MA_5Y	0.0007
37	SP_DYN_IMRT_IN	0.0007
36	NV_SRV_TOTL_ZS	0.0007
23	SP_DYN_CBRT_IN	0.0007
46	SP_POP_7074_FE_5Y	0.0007
30	SP_POP_DPND_YG	0.0007
21	SP_POP_0509_FE_5Y	0.0006

38	SP_POP_DPND_DL	0.0006
47	SP_POP_6064_FE_5Y	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
13	SP_POP_0004_MA_5Y	0.0006
6	SP_DYN_LE00_IN	0.0005
16	SP_POP_7074_MA_5Y	0.0005
15	SP_POP_5054_MA_5Y	0.0005
1	NY_GNP_PCAP_CD	0.0005
48	SH_DYN_MORT_MA	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
27	SP_POP_7579_MA_5Y	0.0004
11	SP_POP_1014_MA_5Y	0.0004
9	SP_POP_0509_MA_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0004
28	SH_DYN_NMRT	0.0004
3	NY_GDP_PCAP_KD	0.0003
26	SP_POP_65UP_TO_ZS	0.0003
10	SP_POP_0014_TO_ZS	0.0003
5	SP_DYN_LE00_MA_IN	0.0003

Model with 57 variables and max depth None:

Training+Validation R^2: 0.99765, RMSE: 0.96738

Testing R^2: 0.98081, RMSE: 2.7961

Mean cross-validation score: 0.98103

	Feature	Importance
0	WB_CC_EST_avg	0.9388
56	CC_EST_prev	0.0216
8	SP_POP_0014_MA_ZS	0.0025
29	SP_POP_1519_MA_5Y	0.0020
51	SP_POP_4549_FE_5Y	0.0017
54	SP_DYN_AMRT_MA	0.0011
45	SP_POP_6569_FE_5Y	0.0011
24	SP_POP_6064_MA_5Y	0.0010
25	SP_POP_0004_FE_5Y	0.0010
48	SH_DYN_MORT_MA	0.0010
50	SP_DYN_TFRT_IN	0.0010
40	SP_POP_5559_FE_5Y	0.0010
20	SP_DYN_TO65_MA_ZS	0.0010
55	NV_AGR_TOTL_ZS	0.0009
31	SP_POP_1519_FE_5Y	0.0009
34	SP_DYN_IMRT_MA_IN	0.0008
35	SP_POP_65UP_FE_ZS	0.0008
38	SP_POP_DPND_DL	0.0008
23	SP_DYN_CBRT_IN	0.0008
21	SP_POP_0509_FE_5Y	0.0008
47	SP_POP_6064_FE_5Y	0.0008

10	SP_POP_0014_TO_ZS	0.0008
46	SP_POP_7074_FE_5Y	0.0007
15	SP_POP_5054_MA_5Y	0.0007
36	NV_SRV_TOTL_ZS	0.0007
7	SP_DYN_LE00_FE_IN	0.0007
44	SP_POP_4549_MA_5Y	0.0007
43	SP_DYN_T065_FE_ZS	0.0007
19	IT_MLT_MAIN_P2	0.0006
33	SP_POP_80UP_FE_5Y	0.0006
37	SP_DYN_IMRT_IN	0.0006
52	SH_DYN_MORT_FE	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
32	SP_POP_5054_FE_5Y	0.0006
18	SP_POP_0014_FE_ZS	0.0006
26	SP_POP_65UP_TO_ZS	0.0006
16	SP_POP_7074_MA_5Y	0.0006
39	SP_ADO_TFRT	0.0006
49	SH_DYN_MORT	0.0005
53	SP_POP_7579_FE_5Y	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
17	SP_POP_5559_MA_5Y	0.0005
14	SP_POP_6569_MA_5Y	0.0005
11	SP_POP_1014_MA_5Y	0.0005
42	SP_POP_80UP_MA_5Y	0.0004
3	NY_GDP_PCAP_KD	0.0004
27	SP_POP_7579_MA_5Y	0.0004
22	SP_POP_1014_FE_5Y	0.0004
9	SP_POP_0509_MA_5Y	0.0004
30	SP_POP_DPNP_YG	0.0003
1	NY_GNP_PCAP_CD	0.0003
13	SP_POP_0004_MA_5Y	0.0003
12	SP_POP_65UP_MA_ZS	0.0003
6	SP_DYN_LE00_IN	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
4	NY_GDP_PCAP_CD	0.0003
28	SH_DYN_NMRT	0.0003

Model with 58 variables and max depth None:

Training+Validation R^2: 0.99545, RMSE: 1.34516

Testing R^2: 0.98079, RMSE: 2.79715

Mean cross-validation score: 0.98154

	Feature	Importance
0	WB_CC_EST_avg	0.9201
57	CC_EST_prev	0.0243
29	SP_POP_1519_MA_5Y	0.0034
8	SP_POP_0014_MA_ZS	0.0029

20	SP_DYN_T065_MA_ZS	0.0024
51	SP_POP_4549_FE_5Y	0.0020
21	SP_POP_0509_FE_5Y	0.0017
54	SP_DYN_AMRT_MA	0.0015
7	SP_DYN_LE00_FE_IN	0.0015
46	SP_POP_7074_FE_5Y	0.0015
50	SP_DYN_TFRT_IN	0.0014
47	SP_POP_6064_FE_5Y	0.0013
43	SP_DYN_T065_FE_ZS	0.0013
30	SP_POP_DPND_YG	0.0013
45	SP_POP_6569_FE_5Y	0.0012
55	NV_AGR_TOTL_ZS	0.0012
53	SP_POP_7579_FE_5Y	0.0012
40	SP_POP_5559_FE_5Y	0.0012
24	SP_POP_6064_MA_5Y	0.0012
38	SP_POP_DPND_DL	0.0012
31	SP_POP_1519_FE_5Y	0.0012
18	SP_POP_0014_FE_ZS	0.0011
25	SP_POP_0004_FE_5Y	0.0010
39	SP_ADO_TFRT	0.0010
34	SP_DYN_IMRT_MA_IN	0.0009
52	SH_DYN_MORT_FE	0.0009
44	SP_POP_4549_MA_5Y	0.0009
10	SP_POP_0014_TO_ZS	0.0009
33	SP_POP_80UP_FE_5Y	0.0008
36	NV_SRV_TOTL_ZS	0.0008
27	SP_POP_7579_MA_5Y	0.0008
48	SH_DYN_MORT_MA	0.0008
19	IT_MLT_MAIN_P2	0.0008
56	TM_VAL_MRCH_HI_ZS	0.0008
15	SP_POP_5054_MA_5Y	0.0008
49	SH_DYN_MORT	0.0007
5	SP_DYN_LE00_MA_IN	0.0007
14	SP_POP_6569_MA_5Y	0.0007
37	SP_DYN_IMRT_IN	0.0007
32	SP_POP_5054_FE_5Y	0.0007
41	SP_DYN_IMRT_FE_IN	0.0006
16	SP_POP_7074_MA_5Y	0.0006
17	SP_POP_5559_MA_5Y	0.0006
13	SP_POP_0004_MA_5Y	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
22	SP_POP_1014_FE_5Y	0.0006
9	SP_POP_0509_MA_5Y	0.0006
35	SP_POP_65UP_FE_ZS	0.0006
23	SP_DYN_CBRT_IN	0.0006
6	SP_DYN_LE00_IN	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
28	SH_DYN_NMRT	0.0005

11	SP_POP_1014_MA_5Y	0.0005
42	SP_POP_80UP_MA_5Y	0.0005
1	NY_GNP_PCAP_CD	0.0004
4	NY_GDP_PCAP_CD	0.0004
3	NY_GDP_PCAP_KD	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003

Model with 59 variables and max depth None:
 Training+Validation R^2: 0.99843, RMSE: 0.79011
 Testing R^2: 0.98188, RMSE: 2.7166
 Mean cross-validation score: 0.98173

	Feature	Importance
0	WB_CC_EST_avg	0.9240
58	CC_EST_prev	0.0206
8	SP_POP_0014_MA_ZS	0.0037
29	SP_POP_1519_MA_5Y	0.0030
25	SP_POP_0004_FE_5Y	0.0017
51	SP_POP_4549_FE_5Y	0.0017
18	SP_POP_0014_FE_ZS	0.0015
50	SP_DYN_TFRT_IN	0.0015
20	SP_DYN_T065_MA_ZS	0.0015
24	SP_POP_6064_MA_5Y	0.0014
47	SP_POP_6064_FE_5Y	0.0014
31	SP_POP_1519_FE_5Y	0.0014
43	SP_DYN_T065_FE_ZS	0.0014
54	SP_DYN_AMRT_MA	0.0013
46	SP_POP_7074_FE_5Y	0.0013
40	SP_POP_5559_FE_5Y	0.0013
53	SP_POP_7579_FE_5Y	0.0012
26	SP_POP_65UP_TO_ZS	0.0012
45	SP_POP_6569_FE_5Y	0.0011
22	SP_POP_1014_FE_5Y	0.0010
23	SP_DYN_CBRT_IN	0.0010
33	SP_POP_80UP_FE_5Y	0.0010
38	SP_POP_DPND_OL	0.0010
55	NV_AGR_TOTL_ZS	0.0010
41	SP_DYN_IMRT_FE_IN	0.0009
52	SH_DYN_MORT_FE	0.0009
57	FM_AST_PRVT_GD_ZS	0.0009
34	SP_DYN_IMRT_MA_IN	0.0009
44	SP_POP_4549_MA_5Y	0.0009
30	SP_POP_DPND_YG	0.0009
7	SP_DYN_LE00_FE_IN	0.0009
19	IT_MLT_MAIN_P2	0.0009
39	SP_ADO_TFRT	0.0008
10	SP_POP_0014_TO_ZS	0.0008

56	TM_VAL_MRCH_HI_ZS	0.0008
35	SP_POP_65UP_FE_ZS	0.0008
21	SP_POP_0509_FE_5Y	0.0008
49	SH_DYN_MORT	0.0008
17	SP_POP_5559_MA_5Y	0.0007
16	SP_POP_7074_MA_5Y	0.0007
6	SP_DYN_LE00_IN	0.0007
15	SP_POP_5054_MA_5Y	0.0007
37	SP_DYN_IMRT_IN	0.0007
32	SP_POP_5054_FE_5Y	0.0007
36	NV_SRV_TOTL_ZS	0.0006
27	SP_POP_7579_MA_5Y	0.0006
14	SP_POP_6569_MA_5Y	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
11	SP_POP_1014_MA_5Y	0.0005
3	NY_GDP_PCAP_KD	0.0005
13	SP_POP_0004_MA_5Y	0.0005
48	SH_DYN_MORT_MA	0.0004
42	SP_POP_80UP_MA_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
9	SP_POP_0509_MA_5Y	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
28	SH_DYN_NMRT	0.0003
1	NY_GNP_PCAP_CD	0.0002

Model with 60 variables and max depth None:

Training+Validation R^2: 0.99888, RMSE: 0.66699

Testing R^2: 0.98142, RMSE: 2.75145

Mean cross-validation score: 0.98223

	Feature	Importance
0	WB_CC_EST_avg	0.9399
59	CC_EST_prev	0.0197
29	SP_POP_1519_MA_5Y	0.0024
8	SP_POP_0014_MA_ZS	0.0019
54	SP_DYN_AMRT_MA	0.0012
58	FD_AST_PRVT_GD_ZS	0.0012
20	SP_DYN_T065_MA_ZS	0.0012
51	SP_POP_4549_FE_5Y	0.0011
45	SP_POP_6569_FE_5Y	0.0011
50	SP_DYN_TFRT_IN	0.0011
33	SP_POP_80UP_FE_5Y	0.0010
18	SP_POP_0014_FE_ZS	0.0010
43	SP_DYN_T065_FE_ZS	0.0009
46	SP_POP_7074_FE_5Y	0.0009
37	SP_DYN_IMRT_IN	0.0009

35	SP_POP_65UP_FE_ZS	0.0009
44	SP_POP_4549_MA_5Y	0.0008
40	SP_POP_5559_FE_5Y	0.0008
53	SP_POP_7579_FE_5Y	0.0008
47	SP_POP_6064_FE_5Y	0.0008
31	SP_POP_1519_FE_5Y	0.0008
56	TM_VAL_MRCH_HI_ZS	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
55	NV_AGR_TOTL_ZS	0.0008
14	SP_POP_6569_MA_5Y	0.0007
24	SP_POP_6064_MA_5Y	0.0007
25	SP_POP_0004_FE_5Y	0.0007
41	SP_DYN_IMRT_FE_IN	0.0007
39	SP_ADO_TFRT	0.0007
38	SP_POP_DPND_OL	0.0007
22	SP_POP_1014_FE_5Y	0.0007
32	SP_POP_5054_FE_5Y	0.0007
15	SP_POP_5054_MA_5Y	0.0007
34	SP_DYN_IMRT_MA_IN	0.0006
21	SP_POP_0509_FE_5Y	0.0006
36	NV_SRV_TOTL_ZS	0.0005
13	SP_POP_0004_MA_5Y	0.0005
16	SP_POP_7074_MA_5Y	0.0005
17	SP_POP_5559_MA_5Y	0.0005
42	SP_POP_80UP_MA_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0005
27	SP_POP_7579_MA_5Y	0.0005
19	IT_MLT_MAIN_P2	0.0005
23	SP_DYN_CBRT_IN	0.0005
57	FM_AST_PRVT_GD_ZS	0.0005
52	SH_DYN_MORT_FE	0.0005
3	NY_GDP_PCAP_KD	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
28	SH_DYN_NMRT	0.0004
26	SP_POP_65UP_TO_ZS	0.0004
2	NY_GDP_PCAP_KD_rel	0.0003
49	SH_DYN_MORT	0.0003
11	SP_POP_1014_MA_5Y	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
4	NY_GDP_PCAP_CD	0.0003
30	SP_POP_DPND_YG	0.0003
48	SH_DYN_MORT_MA	0.0002
1	NY_GNP_PCAP_CD	0.0002
9	SP_POP_0509_MA_5Y	0.0002
6	SP_DYN_LE00_IN	0.0002

Model with 61 variables and max depth None:

Training+Validation R^2: 0.99821, RMSE: 0.84428
 Testing R^2: 0.98144, RMSE: 2.74984
 Mean cross-validation score: 0.98157

	Feature	Importance
0	WB_CC_EST_avg	0.9338
60	CC_EST_prev	0.0228
8	SP_POP_0014_MA_ZS	0.0022
29	SP_POP_1519_MA_5Y	0.0020
50	SP_DYN_TFRT_IN	0.0015
20	SP_DYN_T065_MA_ZS	0.0014
18	SP_POP_0014_FE_ZS	0.0014
51	SP_POP_4549_FE_5Y	0.0013
58	FD_AST_PRVT_GD_ZS	0.0012
45	SP_POP_6569_FE_5Y	0.0012
54	SP_DYN_AMRT_MA	0.0012
25	SP_POP_0004_FE_5Y	0.0011
21	SP_POP_0509_FE_5Y	0.0011
43	SP_DYN_T065_FE_ZS	0.0010
40	SP_POP_5559_FE_5Y	0.0010
31	SP_POP_1519_FE_5Y	0.0010
55	NV_AGR_TOTL_ZS	0.0009
47	SP_POP_6064_FE_5Y	0.0009
52	SH_DYN_MORT_FE	0.0008
24	SP_POP_6064_MA_5Y	0.0008
42	SP_POP_80UP_MA_5Y	0.0008
26	SP_POP_65UP_TO_ZS	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
38	SP_POP_DPND_OL	0.0007
44	SP_POP_4549_MA_5Y	0.0007
53	SP_POP_7579_FE_5Y	0.0007
41	SP_DYN_IMRT_FE_IN	0.0007
46	SP_POP_7074_FE_5Y	0.0007
37	SP_DYN_IMRT_IN	0.0006
35	SP_POP_65UP_FE_ZS	0.0006
59	SP_DYN_AMRT_FE	0.0006
15	SP_POP_5054_MA_5Y	0.0006
6	SP_DYN_LE00_IN	0.0006
17	SP_POP_5559_MA_5Y	0.0006
57	FM_AST_PRVT_GD_ZS	0.0006
56	TM_VAL_MRCH_HI_ZS	0.0006
22	SP_POP_1014_FE_5Y	0.0006
19	IT_MLT_MAIN_P2	0.0006
49	SH_DYN_MORT	0.0005
30	SP_POP_DPND_YG	0.0005
39	SP_ADO_TFRT	0.0005
36	NV_SRV_TOTL_ZS	0.0005
13	SP_POP_0004_MA_5Y	0.0005

14	SP_POP_6569_MA_5Y	0.0005
16	SP_POP_7074_MA_5Y	0.0005
23	SP_DYN_CBRT_IN	0.0005
3	NY_GDP_PCAP_KD	0.0005
33	SP_POP_80UP_FE_5Y	0.0005
34	SP_DYN_IMRT_MA_IN	0.0005
28	SH_DYN_NMRT	0.0004
48	SH_DYN_MORT_MA	0.0004
27	SP_POP_7579_MA_5Y	0.0004
32	SP_POP_5054_FE_5Y	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
11	SP_POP_1014_MA_5Y	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
1	NY_GNP_PCAP_CD	0.0003
9	SP_POP_0509_MA_5Y	0.0003
5	SP_DYN_LEOO_MA_IN	0.0003
4	NY_GDP_PCAP_CD	0.0003
10	SP_POP_0014_TO_ZS	0.0001

Model with 62 variables and max depth None:
 Training+Validation R^2: 0.99803, RMSE: 0.88484
 Testing R^2: 0.98108, RMSE: 2.77611
 Mean cross-validation score: 0.98177

	Feature	Importance
0	WB_CC_EST_avg	0.9335
61	CC_EST_prev	0.0192
8	SP_POP_0014_MA_ZS	0.0025
29	SP_POP_1519_MA_5Y	0.0021
58	FD_AST_PRVT_GD_ZS	0.0020
51	SP_POP_4549_FE_5Y	0.0014
22	SP_POP_1014_FE_5Y	0.0012
7	SP_DYN_LEOO_FE_IN	0.0012
50	SP_DYN_TFRT_IN	0.0012
20	SP_DYN_TO65_MA_ZS	0.0012
24	SP_POP_6064_MA_5Y	0.0011
34	SP_DYN_IMRT_MA_IN	0.0011
33	SP_POP_80UP_FE_5Y	0.0009
25	SP_POP_0004_FE_5Y	0.0009
32	SP_POP_5054_FE_5Y	0.0009
43	SP_DYN_TO65_FE_ZS	0.0009
45	SP_POP_6569_FE_5Y	0.0009
55	NV_AGR_TOTL_ZS	0.0009
47	SP_POP_6064_FE_5Y	0.0009
31	SP_POP_1519_FE_5Y	0.0009
54	SP_DYN_AMRT_MA	0.0008
38	SP_POP_DPND_DL	0.0008

10	SP_POP_0014_TO_ZS	0.0008
36	NV_SRV_TOTL_ZS	0.0008
59	SP_DYN_AMRT_FE	0.0008
21	SP_POP_0509_FE_5Y	0.0008
9	SP_POP_0509_MA_5Y	0.0008
14	SP_POP_6569_MA_5Y	0.0008
40	SP_POP_5559_FE_5Y	0.0008
44	SP_POP_4549_MA_5Y	0.0008
52	SH_DYN_MORT_FE	0.0007
49	SH_DYN_MORT	0.0007
41	SP_DYN_IMRT_FE_IN	0.0007
15	SP_POP_5054_MA_5Y	0.0007
56	TM_VAL_MRCH_HI_ZS	0.0007
60	SP_POP_2024_FE_5Y	0.0007
17	SP_POP_5559_MA_5Y	0.0007
23	SP_DYN_CBRT_IN	0.0007
18	SP_POP_0014_FE_ZS	0.0007
46	SP_POP_7074_FE_5Y	0.0007
53	SP_POP_7579_FE_5Y	0.0006
57	FM_AST_PRVT_GD_ZS	0.0006
11	SP_POP_1014_MA_5Y	0.0006
19	IT_MLT_MAIN_P2	0.0006
28	SH_DYN_NMRT	0.0006
39	SP_ADO_TFRT	0.0006
26	SP_POP_65UP_TO_ZS	0.0006
35	SP_POP_65UP_FE_ZS	0.0006
2	NY_GDP_PCAP_KD_rel	0.0005
13	SP_POP_0004_MA_5Y	0.0005
3	NY_GDP_PCAP_KD	0.0005
16	SP_POP_7074_MA_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0004
27	SP_POP_7579_MA_5Y	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
30	SP_POP_DPND_YG	0.0004
42	SP_POP_80UP_MA_5Y	0.0004
37	SP_DYN_IMRT_IN	0.0004
5	SP_DYN_LEOO_MA_IN	0.0003
6	SP_DYN_LEOO_IN	0.0003
48	SH_DYN_MORT_MA	0.0003
1	NY_GNP_PCAP_CD	0.0003

Model with 63 variables and max depth None:
 Training+Validation R^2: 0.99611, RMSE: 1.24272
 Testing R^2: 0.9811, RMSE: 2.77471
 Mean cross-validation score: 0.98174

Feature Importance

0	WB_CC_EST_avg	0.9408
62	CC_EST_prev	0.0192
8	SP_POP_0014_MA_ZS	0.0022
29	SP_POP_1519_MA_5Y	0.0019
58	FD_AST_PRVT_GD_ZS	0.0017
51	SP_POP_4549_FE_5Y	0.0013
45	SP_POP_6569_FE_5Y	0.0010
7	SP_DYN_LE00_FE_IN	0.0010
50	SP_DYN_TFRT_IN	0.0010
18	SP_POP_0014_FE_ZS	0.0009
25	SP_POP_0004_FE_5Y	0.0009
47	SP_POP_6064_FE_5Y	0.0009
9	SP_POP_0509_MA_5Y	0.0009
46	SP_POP_7074_FE_5Y	0.0008
20	SP_DYN_T065_MA_ZS	0.0008
38	SP_POP_DPND_OL	0.0008
53	SP_POP_7579_FE_5Y	0.0007
54	SP_DYN_AMRT_MA	0.0007
43	SP_DYN_T065_FE_ZS	0.0007
41	SP_DYN_IMRT_FE_IN	0.0007
55	NV_AGR_TOTL_ZS	0.0007
40	SP_POP_5559_FE_5Y	0.0007
39	SP_ADO_TFRT	0.0007
35	SP_POP_65UP_FE_ZS	0.0007
32	SP_POP_5054_FE_5Y	0.0007
31	SP_POP_1519_FE_5Y	0.0007
24	SP_POP_6064_MA_5Y	0.0007
44	SP_POP_4549_MA_5Y	0.0006
15	SP_POP_5054_MA_5Y	0.0006
60	SP_POP_2024_FE_5Y	0.0006
33	SP_POP_80UP_FE_5Y	0.0006
34	SP_DYN_IMRT_MA_IN	0.0006
56	TM_VAL_MRCH_HI_ZS	0.0006
36	NV_SRV_TOTL_ZS	0.0006
10	SP_POP_0014_TO_ZS	0.0006
13	SP_POP_0004_MA_5Y	0.0006
61	SP_POP_2024_MA_5Y	0.0005
59	SP_DYN_AMRT_FE	0.0005
57	FM_AST_PRVT_GD_ZS	0.0005
52	SH_DYN_MORT_FE	0.0005
49	SH_DYN_MORT	0.0005
48	SH_DYN_MORT_MA	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
3	NY_GDP_PCAP_KD	0.0005
17	SP_POP_5559_MA_5Y	0.0005
19	IT_MLT_MAIN_P2	0.0005
21	SP_POP_0509_FE_5Y	0.0005
22	SP_POP_1014_FE_5Y	0.0005

2	NY_GDP_PCAP_KD_rel	0.0004
14	SP_POP_6569_MA_5Y	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
11	SP_POP_1014_MA_5Y	0.0004
30	SP_POP_DPNP_YG	0.0004
4	NY_GDP_PCAP_CD	0.0004
23	SP_DYN_CBRT_IN	0.0004
28	SH_DYN_NMRT	0.0003
1	NY_GNP_PCAP_CD	0.0003
16	SP_POP_7074_MA_5Y	0.0003
42	SP_POP_80UP_MA_5Y	0.0003
37	SP_DYN_IMRT_IN	0.0003
5	SP_DYN_LE00_MA_IN	0.0002
6	SP_DYN_LE00_IN	0.0002
27	SP_POP_7579_MA_5Y	0.0002

Model with 64 variables and max depth None:

Training+Validation R^2: 0.9978, RMSE: 0.93602

Testing R^2: 0.98168, RMSE: 2.73225

Mean cross-validation score: 0.98125

	Feature	Importance
0	WB_CC_EST_avg	0.9267
63	CC_EST_prev	0.0199
8	SP_POP_0014_MA_ZS	0.0041
29	SP_POP_1519_MA_5Y	0.0028
58	FD_AST_PRVT_GD_ZS	0.0025
51	SP_POP_4549_FE_5Y	0.0017
50	SP_DYN_TFRT_IN	0.0014
20	SP_DYN_T065_MA_ZS	0.0013
9	SP_POP_0509_MA_5Y	0.0013
31	SP_POP_1519_FE_5Y	0.0011
52	SH_DYN_MORT_FE	0.0011
24	SP_POP_6064_MA_5Y	0.0011
47	SP_POP_6064_FE_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0010
32	SP_POP_5054_FE_5Y	0.0010
54	SP_DYN_AMRT_MA	0.0010
60	SP_POP_2024_FE_5Y	0.0010
25	SP_POP_0004_FE_5Y	0.0010
46	SP_POP_7074_FE_5Y	0.0009
53	SP_POP_7579_FE_5Y	0.0009
42	SP_POP_80UP_MA_5Y	0.0009
40	SP_POP_5559_FE_5Y	0.0009
55	NV_AGR_TOTL_ZS	0.0009
2	NY_GDP_PCAP_KD_rel	0.0009
33	SP_POP_80UP_FE_5Y	0.0009

7	SP_DYN_LE00_FE_IN	0.0009
22	SP_POP_1014_FE_5Y	0.0009
34	SP_DYN_IMRT_MA_IN	0.0008
62	SP_URB_TOTL_IN_ZS	0.0008
56	TM_VAL_MRCH_HI_ZS	0.0008
37	SP_DYN_IMRT_IN	0.0008
23	SP_DYN_CBRT_IN	0.0008
30	SP_POP_DPND_YG	0.0008
21	SP_POP_0509_FE_5Y	0.0008
44	SP_POP_4549_MA_5Y	0.0007
39	SP_ADO_TFRT	0.0007
57	FM_AST_PRVT_GD_ZS	0.0007
49	SH_DYN_MORT	0.0007
14	SP_POP_6569_MA_5Y	0.0007
61	SP_POP_2024_MA_5Y	0.0007
36	NV_SRV_TOTL_ZS	0.0007
41	SP_DYN_IMRT_FE_IN	0.0006
17	SP_POP_5559_MA_5Y	0.0006
38	SP_POP_DPND_DL	0.0006
16	SP_POP_7074_MA_5Y	0.0006
26	SP_POP_65UP_TO_ZS	0.0006
13	SP_POP_0004_MA_5Y	0.0006
19	IT_MLT_MAIN_P2	0.0006
18	SP_POP_0014_FE_ZS	0.0005
15	SP_POP_5054_MA_5Y	0.0005
3	NY_GDP_PCAP_KD	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
59	SP_DYN_AMRT_FE	0.0005
6	SP_DYN_LE00_IN	0.0004
27	SP_POP_7579_MA_5Y	0.0004
1	NY_GNP_PCAP_CD	0.0004
43	SP_DYN_TO65_FE_ZS	0.0004
35	SP_POP_65UP_FE_ZS	0.0004
28	SH_DYN_NMRT	0.0003
10	SP_POP_0014_TO_ZS	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
11	SP_POP_1014_MA_5Y	0.0003
4	NY_GDP_PCAP_CD	0.0003
48	SH_DYN_MORT_MA	0.0003

Model with 65 variables and max depth None:
 Training+Validation R^2: 0.99841, RMSE: 0.79413
 Testing R^2: 0.98167, RMSE: 2.73287
 Mean cross-validation score: 0.98132

	Feature	Importance
0	WB_CC_EST_avg	0.9359

64	CC_EST_prev	0.0184
8	SP_POP_0014_MA_ZS	0.0029
29	SP_POP_1519_MA_5Y	0.0022
58	FD_AST_PRVT_GD_ZS	0.0017
51	SP_POP_4549_FE_5Y	0.0014
9	SP_POP_0509_MA_5Y	0.0012
45	SP_POP_6569_FE_5Y	0.0012
52	SH_DYN_MORT_FE	0.0011
20	SP_DYN_T065_MA_ZS	0.0011
60	SP_POP_2024_FE_5Y	0.0010
24	SP_POP_6064_MA_5Y	0.0010
50	SP_DYN_TFRT_IN	0.0010
49	SH_DYN_MORT	0.0009
46	SP_POP_7074_FE_5Y	0.0008
40	SP_POP_5559_FE_5Y	0.0008
42	SP_POP_80UP_MA_5Y	0.0008
21	SP_POP_0509_FE_5Y	0.0008
33	SP_POP_80UP_FE_5Y	0.0008
47	SP_POP_6064_FE_5Y	0.0008
31	SP_POP_1519_FE_5Y	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
54	SP_DYN_AMRT_MA	0.0007
44	SP_POP_4549_MA_5Y	0.0007
55	NV_AGR_TOTL_ZS	0.0007
56	TM_VAL_MRCH_HI_ZS	0.0007
39	SP_ADO_TFRT	0.0007
37	SP_DYN_IMRT_IN	0.0007
36	NV_SRV_TOTL_ZS	0.0007
57	FM_AST_PRVT_GD_ZS	0.0007
34	SP_DYN_IMRT_MA_IN	0.0007
2	NY_GDP_PCAP_KD_rel	0.0007
25	SP_POP_0004_FE_5Y	0.0007
13	SP_POP_0004_MA_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
23	SP_DYN_CBRT_IN	0.0007
32	SP_POP_5054_FE_5Y	0.0006
53	SP_POP_7579_FE_5Y	0.0006
17	SP_POP_5559_MA_5Y	0.0006
22	SP_POP_1014_FE_5Y	0.0006
19	IT_MLT_MAIN_P2	0.0006
30	SP_POP_DPND_YG	0.0006
35	SP_POP_65UP_FE_ZS	0.0006
63	SP_RUR_TOTL_ZS	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
41	SP_DYN_IMRT_FE_IN	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
61	SP_POP_2024_MA_5Y	0.0005
16	SP_POP_7074_MA_5Y	0.0005

18	SP_POP_0014_FE_ZS	0.0005
38	SP_POP_DPND_DL	0.0005
3	NY_GDP_PCAP_KD	0.0004
15	SP_POP_5054_MA_5Y	0.0004
59	SP_DYN_AMRT_FE	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
43	SP_DYN_T065_FE_ZS	0.0004
6	SP_DYN_LE00_IN	0.0004
4	NY_GDP_PCAP_CD	0.0003
1	NY_GNP_PCAP_CD	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
27	SP_POP_7579_MA_5Y	0.0003
28	SH_DYN_NMRT	0.0003
48	SH_DYN_MORT_MA	0.0003
10	SP_POP_0014_TO_ZS	0.0002
11	SP_POP_1014_MA_5Y	0.0002

Model with 66 variables and max depth None:

Training+Validation R^2: 0.99703, RMSE: 1.08664

Testing R^2: 0.98098, RMSE: 2.7836

Mean cross-validation score: 0.9813

	Feature	Importance
0	WB_CC_EST_avg	0.9272
65	CC_EST_prev	0.0231
18	SP_POP_0014_FE_ZS	0.0034
8	SP_POP_0014_MA_ZS	0.0024
29	SP_POP_1519_MA_5Y	0.0020
58	FD_AST_PRVT_GD_ZS	0.0018
64	SG_LAW_INDX	0.0016
50	SP_DYN_TFRT_IN	0.0015
25	SP_POP_0004_FE_5Y	0.0014
20	SP_DYN_T065_MA_ZS	0.0014
53	SP_POP_7579_FE_5Y	0.0012
55	NV_AGR_TOTL_ZS	0.0011
45	SP_POP_6569_FE_5Y	0.0010
21	SP_POP_0509_FE_5Y	0.0010
51	SP_POP_4549_FE_5Y	0.0010
54	SP_DYN_AMRT_MA	0.0010
7	SP_DYN_LE00_FE_IN	0.0009
43	SP_DYN_T065_FE_ZS	0.0009
60	SP_POP_2024_FE_5Y	0.0009
49	SH_DYN_MORT	0.0009
42	SP_POP_80UP_MA_5Y	0.0008
40	SP_POP_5559_FE_5Y	0.0008
34	SP_DYN_IMRT_MA_IN	0.0008
44	SP_POP_4549_MA_5Y	0.0008

52	SH_DYN_MORT_FE	0.0008
38	SP_POP_DPND_OL	0.0007
36	NV_SRV_TOTL_ZS	0.0007
32	SP_POP_5054_FE_5Y	0.0007
33	SP_POP_80UP_FE_5Y	0.0007
28	SH_DYN_NMRT	0.0007
24	SP_POP_6064_MA_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
56	TM_VAL_MRCH_HI_ZS	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
15	SP_POP_5054_MA_5Y	0.0006
46	SP_POP_7074_FE_5Y	0.0006
16	SP_POP_7074_MA_5Y	0.0006
47	SP_POP_6064_FE_5Y	0.0006
35	SP_POP_65UP_FE_ZS	0.0006
31	SP_POP_1519_FE_5Y	0.0006
30	SP_POP_DPND_YG	0.0005
6	SP_DYN_LEOO_IN	0.0005
39	SP_ADO_TFRT	0.0005
13	SP_POP_0004_MA_5Y	0.0005
59	SP_DYN_AMRT_FE	0.0005
3	NY_GDP_PCAP_KD	0.0005
61	SP_POP_2024_MA_5Y	0.0005
37	SP_DYN_IMRT_IN	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
17	SP_POP_5559_MA_5Y	0.0005
19	IT_MLT_MAIN_P2	0.0005
62	SP_URB_TOTL_IN_ZS	0.0005
41	SP_DYN_IMRT_FE_IN	0.0005
23	SP_DYN_CBRT_IN	0.0005
57	FM_AST_PRVT_GD_ZS	0.0005
27	SP_POP_7579_MA_5Y	0.0004
9	SP_POP_0509_MA_5Y	0.0004
10	SP_POP_0014_TO_ZS	0.0004
11	SP_POP_1014_MA_5Y	0.0004
22	SP_POP_1014_FE_5Y	0.0004
1	NY_GNP_PCAP_CD	0.0003
63	SP_RUR_TOTL_ZS	0.0003
4	NY_GDP_PCAP_CD	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
5	SP_DYN_LEOO_MA_IN	0.0002
48	SH_DYN_MORT_MA	0.0002

Model with 67 variables and max depth None:

Training+Validation R^2: 0.99952, RMSE: 0.43649

Testing R^2: 0.98199, RMSE: 2.70866

Mean cross-validation score: 0.98151

	Feature	Importance
0	WB_CC_EST_avg	0.9461
66	CC_EST_prev	0.0163
8	SP_POP_0014_MA_ZS	0.0022
29	SP_POP_1519_MA_5Y	0.0018
18	SP_POP_0014_FE_ZS	0.0016
58	FD_AST_PRVT_GD_ZS	0.0015
64	SG_LAW_INDX	0.0012
50	SP_DYN_TFRT_IN	0.0009
20	SP_DYN_T065_MA_ZS	0.0009
54	SP_DYN_AMRT_MA	0.0008
52	SH_DYN_MORT_FE	0.0008
46	SP_POP_7074_FE_5Y	0.0007
53	SP_POP_7579_FE_5Y	0.0007
41	SP_DYN_IMRT_FE_IN	0.0007
49	SH_DYN_MORT	0.0007
45	SP_POP_6569_FE_5Y	0.0007
25	SP_POP_0004_FE_5Y	0.0007
51	SP_POP_4549_FE_5Y	0.0007
31	SP_POP_1519_FE_5Y	0.0006
32	SP_POP_5054_FE_5Y	0.0006
44	SP_POP_4549_MA_5Y	0.0006
60	SP_POP_2024_FE_5Y	0.0006
10	SP_POP_0014_TO_ZS	0.0006
7	SP_DYN_LE00_FE_IN	0.0006
36	NV_SRV_TOTL_ZS	0.0006
40	SP_POP_5559_FE_5Y	0.0006
47	SP_POP_6064_FE_5Y	0.0005
3	NY_GDP_PCAP_KD	0.0005
37	SP_DYN_IMRT_IN	0.0005
38	SP_POP_DPND_OL	0.0005
34	SP_DYN_IMRT_MA_IN	0.0005
27	SP_POP_7579_MA_5Y	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
23	SP_DYN_CBRT_IN	0.0005
19	IT_MLT_MAIN_P2	0.0005
55	NV_AGR_TOTL_ZS	0.0005
14	SP_POP_6569_MA_5Y	0.0005
63	SP_RUR_TOTL_ZS	0.0005
65	NE_CON_PRVT_ZS	0.0005
42	SP_POP_80UP_MA_5Y	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004
61	SP_POP_2024_MA_5Y	0.0004
33	SP_POP_80UP_FE_5Y	0.0004
43	SP_DYN_T065_FE_ZS	0.0004
21	SP_POP_0509_FE_5Y	0.0004

6	SP_DYN_LE00_IN	0.0004
9	SP_POP_0509_MA_5Y	0.0004
11	SP_POP_1014_MA_5Y	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
15	SP_POP_5054_MA_5Y	0.0004
39	SP_ADO_TFRT	0.0004
17	SP_POP_5559_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
24	SP_POP_6064_MA_5Y	0.0004
35	SP_POP_65UP_FE_ZS	0.0004
28	SH_DYN_NMRT	0.0003
13	SP_POP_0004_MA_5Y	0.0003
48	SH_DYN_MORT_MA	0.0003
1	NY_GNP_PCAP_CD	0.0003
62	SP_URB_TOTL_IN_ZS	0.0003
4	NY_GDP_PCAP_CD	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
22	SP_POP_1014_FE_5Y	0.0002
30	SP_POP_DPND_YG	0.0002
59	SP_DYN_AMRT_FE	0.0002
5	SP_DYN_LE00_MA_IN	0.0002

Model with 68 variables and max depth None:

Training+Validation R^2: 0.99929, RMSE: 0.53088

Testing R^2: 0.9816, RMSE: 2.73818

Mean cross-validation score: 0.98134

	Feature	Importance
0	WB_CC_EST_avg	0.9426
67	CC_EST_prev	0.0192
29	SP_POP_1519_MA_5Y	0.0019
18	SP_POP_0014_FE_ZS	0.0019
8	SP_POP_0014_MA_ZS	0.0018
58	FD_AST_PRVT_GD_ZS	0.0014
20	SP_DYN_T065_MA_ZS	0.0012
64	SG_LAW_INDX	0.0011
50	SP_DYN_TFRT_IN	0.0010
54	SP_DYN_AMRT_MA	0.0008
25	SP_POP_0004_FE_5Y	0.0008
55	NV_AGR_TOTL_ZS	0.0008
31	SP_POP_1519_FE_5Y	0.0007
53	SP_POP_7579_FE_5Y	0.0007
60	SP_POP_2024_FE_5Y	0.0007
7	SP_DYN_LE00_FE_IN	0.0007
45	SP_POP_6569_FE_5Y	0.0006
42	SP_POP_80UP_MA_5Y	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006

40	SP_POP_5559_FE_5Y	0.0006
39	SP_ADO_TFRT	0.0006
46	SP_POP_7074_FE_5Y	0.0006
38	SP_POP_DPND_OL	0.0006
32	SP_POP_5054_FE_5Y	0.0006
49	SH_DYN_MORT	0.0006
51	SP_POP_4549_FE_5Y	0.0006
44	SP_POP_4549_MA_5Y	0.0006
34	SP_DYN_IMRT_MA_IN	0.0006
66	SP_POP_4044_FE_5Y	0.0006
57	FM_AST_PRVT_GD_ZS	0.0005
37	SP_DYN_IMRT_IN	0.0005
47	SP_POP_6064_FE_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0005
24	SP_POP_6064_MA_5Y	0.0005
17	SP_POP_5559_MA_5Y	0.0005
62	SP_URB_TOTL_IN_ZS	0.0005
21	SP_POP_0509_FE_5Y	0.0005
36	NV_SRV_TOTL_ZS	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0005
33	SP_POP_80UP_FE_5Y	0.0005
23	SP_DYN_CBRT_IN	0.0005
52	SH_DYN_MORT_FE	0.0005
13	SP_POP_0004_MA_5Y	0.0004
65	NE_CON_PRVT_ZS	0.0004
61	SP_POP_2024_MA_5Y	0.0004
27	SP_POP_7579_MA_5Y	0.0004
14	SP_POP_6569_MA_5Y	0.0004
43	SP_DYN_T065_FE_ZS	0.0004
15	SP_POP_5054_MA_5Y	0.0004
19	IT_MLT_MAIN_P2	0.0004
3	NY_GDP_PCAP_KD	0.0004
59	SP_DYN_AMRT_FE	0.0004
30	SP_POP_DPND_YG	0.0004
16	SP_POP_7074_MA_5Y	0.0004
2	NY_GDP_PCAP_KD_rel	0.0003
1	NY_GNP_PCAP_CD	0.0003
26	SP_POP_65UP_TO_ZS	0.0003
22	SP_POP_1014_FE_5Y	0.0003
12	SP_POP_65UP_MA_ZS	0.0003
11	SP_POP_1014_MA_5Y	0.0003
6	SP_DYN_LE00_IN	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
35	SP_POP_65UP_FE_ZS	0.0003
28	SH_DYN_NMRT	0.0002
4	NY_GDP_PCAP_CD	0.0002
48	SH_DYN_MORT_MA	0.0001
9	SP_POP_0509_MA_5Y	0.0001

63 SP_RUR_TOTL_ZS 0.0001

Model with 69 variables and max depth None:
Training+Validation R^2: 0.99892, RMSE: 0.65517
Testing R^2: 0.98119, RMSE: 2.76807
Mean cross-validation score: 0.98125

	Feature	Importance
0	WB_CC_EST_avg	0.9362
68	CC_EST_prev	0.0194
8	SP_POP_0014_MA_ZS	0.0020
29	SP_POP_1519_MA_5Y	0.0020
18	SP_POP_0014_FE_ZS	0.0018
58	FD_AST_PRVT_GD_ZS	0.0018
64	SG_LAW_INDX	0.0013
20	SP_DYN_T065_MA_ZS	0.0013
50	SP_DYN_TFRT_IN	0.0011
25	SP_POP_0004_FE_5Y	0.0011
51	SP_POP_4549_FE_5Y	0.0010
31	SP_POP_1519_FE_5Y	0.0009
34	SP_DYN_IMRT_MA_IN	0.0008
41	SP_DYN_IMRT_FE_IN	0.0008
40	SP_POP_5559_FE_5Y	0.0008
38	SP_POP_DPND_DL	0.0008
54	SP_DYN_AMRT_MA	0.0008
53	SP_POP_7579_FE_5Y	0.0008
60	SP_POP_2024_FE_5Y	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
21	SP_POP_0509_FE_5Y	0.0007
66	SP_POP_4044_FE_5Y	0.0007
45	SP_POP_6569_FE_5Y	0.0007
44	SP_POP_4549_MA_5Y	0.0007
42	SP_POP_80UP_MA_5Y	0.0007
37	SP_DYN_IMRT_IN	0.0007
32	SP_POP_5054_FE_5Y	0.0007
10	SP_POP_0014_TO_ZS	0.0006
52	SH_DYN_MORT_FE	0.0006
55	NV_AGR_TOTL_ZS	0.0006
19	IT_MLT_MAIN_P2	0.0006
36	NV_SRV_TOTL_ZS	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
61	SP_POP_2024_MA_5Y	0.0006
27	SP_POP_7579_MA_5Y	0.0006
24	SP_POP_6064_MA_5Y	0.0006
33	SP_POP_80UP_FE_5Y	0.0005
22	SP_POP_1014_FE_5Y	0.0005
49	SH_DYN_MORT	0.0005

23	SP_DYN_CBRT_IN	0.0005
47	SP_POP_6064_FE_5Y	0.0005
46	SP_POP_7074_FE_5Y	0.0005
15	SP_POP_5054_MA_5Y	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0005
6	SP_DYN_LE00_IN	0.0005
39	SP_ADO_TFRT	0.0005
65	NE_CON_PRVT_ZS	0.0005
14	SP_POP_6569_MA_5Y	0.0005
13	SP_POP_0004_MA_5Y	0.0005
3	NY_GDP_PCAP_KD	0.0005
57	FM_AST_PRVT_GD_ZS	0.0005
28	SH_DYN_NMRT	0.0005
59	SP_DYN_AMRT_FE	0.0004
67	SP_POP_1564_MA_ZS	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
43	SP_DYN_T065_FE_ZS	0.0004
30	SP_POP_DPND_YG	0.0004
26	SP_POP_65UP_TO_ZS	0.0004
17	SP_POP_5559_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
35	SP_POP_65UP_FE_ZS	0.0004
1	NY_GNP_PCAP_CD	0.0003
12	SP_POP_65UP_MA_ZS	0.0003
11	SP_POP_1014_MA_5Y	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
63	SP_RUR_TOTL_ZS	0.0002
9	SP_POP_0509_MA_5Y	0.0002
4	NY_GDP_PCAP_CD	0.0002
48	SH_DYN_MORT_MA	0.0001

Model with 70 variables and max depth None:

Training+Validation R^2: 0.99794, RMSE: 0.9042

Testing R^2: 0.98119, RMSE: 2.7679

Mean cross-validation score: 0.9816

	Feature	Importance
0	WB_CC_EST_avg	0.9245
69	CC_EST_prev	0.0224
64	SG_LAW_INDX	0.0024
29	SP_POP_1519_MA_5Y	0.0024
8	SP_POP_0014_MA_ZS	0.0024
18	SP_POP_0014_FE_ZS	0.0019
58	FD_AST_PRVT_GD_ZS	0.0017
25	SP_POP_0004_FE_5Y	0.0016
20	SP_DYN_T065_MA_ZS	0.0015
34	SP_DYN_IMRT_MA_IN	0.0014

50	SP_DYN_TFRT_IN	0.0014
40	SP_POP_5559_FE_5Y	0.0011
7	SP_DYN_LE00_FE_IN	0.0010
41	SP_DYN_IMRT_FE_IN	0.0010
51	SP_POP_4549_FE_5Y	0.0010
35	SP_POP_65UP_FE_ZS	0.0009
42	SP_POP_80UP_MA_5Y	0.0009
31	SP_POP_1519_FE_5Y	0.0009
53	SP_POP_7579_FE_5Y	0.0009
54	SP_DYN_AMRT_MA	0.0009
55	NV_AGR_TOTL_ZS	0.0009
60	SP_POP_2024_FE_5Y	0.0009
10	SP_POP_0014_TO_ZS	0.0008
37	SP_DYN_IMRT_IN	0.0008
57	FM_AST_PRVT_GD_ZS	0.0008
45	SP_POP_6569_FE_5Y	0.0008
32	SP_POP_5054_FE_5Y	0.0007
44	SP_POP_4549_MA_5Y	0.0007
46	SP_POP_7074_FE_5Y	0.0007
39	SP_ADO_TFRT	0.0007
38	SP_POP_DPND_OL	0.0007
33	SP_POP_80UP_FE_5Y	0.0007
36	NV_SRV_TOTL_ZS	0.0007
27	SP_POP_7579_MA_5Y	0.0007
16	SP_POP_7074_MA_5Y	0.0007
24	SP_POP_6064_MA_5Y	0.0007
19	IT_MLT_MAIN_P2	0.0007
56	TM_VAL_MRCH_HI_ZS	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
65	NE_CON_PRVT_ZS	0.0006
66	SP_POP_4044_FE_5Y	0.0006
52	SH_DYN_MORT_FE	0.0006
61	SP_POP_2024_MA_5Y	0.0006
21	SP_POP_0509_FE_5Y	0.0006
15	SP_POP_5054_MA_5Y	0.0006
23	SP_DYN_CBRT_IN	0.0006
3	NY_GDP_PCAP_KD	0.0006
68	SP_POP_DPND	0.0006
11	SP_POP_1014_MA_5Y	0.0005
28	SH_DYN_NMRT	0.0005
14	SP_POP_6569_MA_5Y	0.0005
43	SP_DYN_TO65_FE_ZS	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
17	SP_POP_5559_MA_5Y	0.0005
63	SP_RUR_TOTL_ZS	0.0004
59	SP_DYN_AMRT_FE	0.0004
67	SP_POP_1564_MA_ZS	0.0004
30	SP_POP_DPND_YG	0.0004

12	SP_POP_65UP_MA_ZS	0.0004
13	SP_POP_0004_MA_5Y	0.0004
49	SH_DYN_MORT	0.0004
47	SP_POP_6064_FE_5Y	0.0004
22	SP_POP_1014_FE_5Y	0.0004
6	SP_DYN_LE00_IN	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
48	SH_DYN_MORT_MA	0.0003
1	NY_GNP_PCAP_CD	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
4	NY_GDP_PCAP_CD	0.0002
9	SP_POP_0509_MA_5Y	0.0001

Model with 71 variables and max depth None:
 Training+Validation R^2: 0.99844, RMSE: 0.78795
 Testing R^2: 0.98133, RMSE: 2.75778
 Mean cross-validation score: 0.98115

	Feature	Importance
0	WB_CC_EST_avg	0.9086
70	CC_EST_prev	0.0235
8	SP_POP_0014_MA_ZS	0.0036
18	SP_POP_0014_FE_ZS	0.0035
29	SP_POP_1519_MA_5Y	0.0031
58	FD_AST_PRVT_GD_ZS	0.0026
64	SG_LAW_INDX	0.0017
20	SP_DYN_T065_MA_ZS	0.0017
53	SP_POP_7579_FE_5Y	0.0017
46	SP_POP_7074_FE_5Y	0.0016
50	SP_DYN_TFRT_IN	0.0015
51	SP_POP_4549_FE_5Y	0.0014
52	SH_DYN_MORT_FE	0.0014
25	SP_POP_0004_FE_5Y	0.0013
69	SP_POP_4044_MA_5Y	0.0012
7	SP_DYN_LE00_FE_IN	0.0012
60	SP_POP_2024_FE_5Y	0.0012
37	SP_DYN_IMRT_IN	0.0012
54	SP_DYN_AMRT_MA	0.0011
44	SP_POP_4549_MA_5Y	0.0011
55	NV_AGR_TOTL_ZS	0.0011
31	SP_POP_1519_FE_5Y	0.0011
66	SP_POP_4044_FE_5Y	0.0011
34	SP_DYN_IMRT_MA_IN	0.0011
45	SP_POP_6569_FE_5Y	0.0011
40	SP_POP_5559_FE_5Y	0.0010
24	SP_POP_6064_MA_5Y	0.0010
42	SP_POP_80UP_MA_5Y	0.0009

36	NV_SRV_TOTL_ZS	0.0009
23	SP_DYN_CBRT_IN	0.0009
49	SH_DYN_MORT	0.0009
56	TM_VAL_MRCH_HI_ZS	0.0008
41	SP_DYN_IMRT_FE_IN	0.0008
62	SP_URB_TOTL_IN_ZS	0.0008
68	SP_POP_DPND	0.0008
38	SP_POP_DPND_OL	0.0008
35	SP_POP_65UP_FE_ZS	0.0008
21	SP_POP_0509_FE_5Y	0.0008
6	SP_DYN_LE00_IN	0.0008
33	SP_POP_80UP_FE_5Y	0.0008
32	SP_POP_5054_FE_5Y	0.0008
27	SP_POP_7579_MA_5Y	0.0008
3	NY_GDP_PCAP_KD	0.0007
65	NE_CON_PRVT_ZS	0.0007
10	SP_POP_0014_TO_ZS	0.0007
57	FM_AST_PRVT_GD_ZS	0.0007
16	SP_POP_7074_MA_5Y	0.0007
19	IT_MLT_MAIN_P2	0.0007
15	SP_POP_5054_MA_5Y	0.0007
39	SP_ADO_TFRT	0.0007
48	SH_DYN_MORT_MA	0.0006
17	SP_POP_5559_MA_5Y	0.0006
14	SP_POP_6569_MA_5Y	0.0006
43	SP_DYN_T065_FE_ZS	0.0006
61	SP_POP_2024_MA_5Y	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
59	SP_DYN_AMRT_FE	0.0006
22	SP_POP_1014_FE_5Y	0.0005
30	SP_POP_DPND_YG	0.0005
63	SP_RUR_TOTL_ZS	0.0005
11	SP_POP_1014_MA_5Y	0.0005
28	SH_DYN_NMRT	0.0005
13	SP_POP_0004_MA_5Y	0.0005
47	SP_POP_6064_FE_5Y	0.0005
1	NY_GNP_PCAP_CD	0.0004
26	SP_POP_65UP_TO_ZS	0.0004
4	NY_GDP_PCAP_CD	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
9	SP_POP_0509_MA_5Y	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
67	SP_POP_1564_MA_ZS	0.0002

Model with 72 variables and max depth None:
 Training+Validation R^2: 0.99868, RMSE: 0.72363
 Testing R^2: 0.9813, RMSE: 2.76002

Mean cross-validation score: 0.9811

	Feature	Importance
0	WB_CC_EST_avg	0.9099
71	CC_EST_prev	0.0231
18	SP_POP_0014_FE_ZS	0.0046
8	SP_POP_0014_MA_ZS	0.0034
29	SP_POP_1519_MA_5Y	0.0030
58	FD_AST_PRVT_GD_ZS	0.0023
64	SG_LAW_INDX	0.0017
20	SP_DYN_T065_MA_ZS	0.0016
53	SP_POP_7579_FE_5Y	0.0016
50	SP_DYN_TFRT_IN	0.0015
52	SH_DYN_MORT_FE	0.0014
60	SP_POP_2024_FE_5Y	0.0013
51	SP_POP_4549_FE_5Y	0.0013
46	SP_POP_7074_FE_5Y	0.0012
25	SP_POP_0004_FE_5Y	0.0012
37	SP_DYN_IMRT_IN	0.0012
7	SP_DYN_LE00_FE_IN	0.0012
54	SP_DYN_AMRT_MA	0.0011
55	NV_AGR_TOTL_ZS	0.0011
31	SP_POP_1519_FE_5Y	0.0011
69	SP_POP_4044_MA_5Y	0.0011
41	SP_DYN_IMRT_FE_IN	0.0010
34	SP_DYN_IMRT_MA_IN	0.0010
44	SP_POP_4549_MA_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0010
66	SP_POP_4044_FE_5Y	0.0010
40	SP_POP_5559_FE_5Y	0.0010
24	SP_POP_6064_MA_5Y	0.0010
49	SH_DYN_MORT	0.0009
6	SP_DYN_LE00_IN	0.0009
23	SP_DYN_CBRT_IN	0.0009
21	SP_POP_0509_FE_5Y	0.0009
32	SP_POP_5054_FE_5Y	0.0009
42	SP_POP_80UP_MA_5Y	0.0009
27	SP_POP_7579_MA_5Y	0.0009
38	SP_POP_DPND_OL	0.0008
57	FM_AST_PRVT_GD_ZS	0.0008
56	TM_VAL_MRCH_HI_ZS	0.0008
36	NV_SRV_TOTL_ZS	0.0008
16	SP_POP_7074_MA_5Y	0.0008
39	SP_ADO_TFRT	0.0007
68	SP_POP_DPND	0.0007
35	SP_POP_65UP_FE_ZS	0.0007
33	SP_POP_80UP_FE_5Y	0.0007
62	SP_URB_TOTL_IN_ZS	0.0007

12	SP_POP_65UP_MA_ZS	0.0007
3	NY_GDP_PCAP_KD	0.0007
15	SP_POP_5054_MA_5Y	0.0007
19	IT_MLT_MAIN_P2	0.0007
65	NE_CON_PRVT_ZS	0.0006
61	SP_POP_2024_MA_5Y	0.0006
59	SP_DYN_AMRT_FE	0.0006
14	SP_POP_6569_MA_5Y	0.0006
17	SP_POP_5559_MA_5Y	0.0006
48	SH_DYN_MORT_MA	0.0006
47	SP_POP_6064_FE_5Y	0.0005
43	SP_DYN_T065_FE_ZS	0.0005
22	SP_POP_1014_FE_5Y	0.0005
28	SH_DYN_NMRT	0.0005
30	SP_POP_DPND_YG	0.0005
11	SP_POP_1014_MA_5Y	0.0005
13	SP_POP_0004_MA_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0004
63	SP_RUR_TOTL_ZS	0.0004
26	SP_POP_65UP_TO_ZS	0.0004
1	NY_GNP_PCAP_CD	0.0004
4	NY_GDP_PCAP_CD	0.0004
70	SP_POP_1564_TO_ZS	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
5	SP_DYN_LE00_MA_IN	0.0003
9	SP_POP_0509_MA_5Y	0.0002
67	SP_POP_1564_MA_ZS	0.0002

Model with 73 variables and max depth None:

Training+Validation R^2: 0.99813, RMSE: 0.86116

Testing R^2: 0.98126, RMSE: 2.76312

Mean cross-validation score: 0.98129

	Feature	Importance
0	WB_CC_EST_avg	0.9146
72	CC_EST_prev	0.0237
18	SP_POP_0014_FE_ZS	0.0034
29	SP_POP_1519_MA_5Y	0.0027
8	SP_POP_0014_MA_ZS	0.0022
58	FD_AST_PRVT_GD_ZS	0.0018
64	SG_LAW_INDX	0.0016
20	SP_DYN_T065_MA_ZS	0.0016
51	SP_POP_4549_FE_5Y	0.0014
53	SP_POP_7579_FE_5Y	0.0014
50	SP_DYN_TFRT_IN	0.0013
54	SP_DYN_AMRT_MA	0.0013
71	NE_EXP_GNFS_ZS	0.0012

31	SP_POP_1519_FE_5Y	0.0011
60	SP_POP_2024_FE_5Y	0.0011
16	SP_POP_7074_MA_5Y	0.0011
25	SP_POP_0004_FE_5Y	0.0011
32	SP_POP_5054_FE_5Y	0.0010
24	SP_POP_6064_MA_5Y	0.0010
49	SH_DYN_MORT	0.0010
40	SP_POP_5559_FE_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0010
55	NV_AGR_TOTL_ZS	0.0009
13	SP_POP_0004_MA_5Y	0.0009
42	SP_POP_80UP_MA_5Y	0.0009
67	SP_POP_1564_MA_ZS	0.0009
44	SP_POP_4549_MA_5Y	0.0009
30	SP_POP_DPND_YG	0.0009
37	SP_DYN_IMRT_IN	0.0009
52	SH_DYN_MORT_FE	0.0009
27	SP_POP_7579_MA_5Y	0.0009
56	TM_VAL_MRCH_HI_ZS	0.0009
65	NE_CON_PRVT_ZS	0.0008
59	SP_DYN_AMRT_FE	0.0008
34	SP_DYN_IMRT_MA_IN	0.0008
66	SP_POP_4044_FE_5Y	0.0008
48	SH_DYN_MORT_MA	0.0008
41	SP_DYN_IMRT_FE_IN	0.0008
33	SP_POP_80UP_FE_5Y	0.0008
61	SP_POP_2024_MA_5Y	0.0007
57	FM_AST_PRVT_GD_ZS	0.0007
69	SP_POP_4044_MA_5Y	0.0007
46	SP_POP_7074_FE_5Y	0.0007
70	SP_POP_1564_TO_ZS	0.0007
36	NV_SRV_TOTL_ZS	0.0007
43	SP_DYN_T065_FE_ZS	0.0007
21	SP_POP_0509_FE_5Y	0.0007
38	SP_POP_DPND_OL	0.0007
14	SP_POP_6569_MA_5Y	0.0007
35	SP_POP_65UP_FE_ZS	0.0007
39	SP_ADO_TFRT	0.0007
19	IT_MLT_MAIN_P2	0.0007
7	SP_DYN_LE00_FE_IN	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
17	SP_POP_5559_MA_5Y	0.0006
15	SP_POP_5054_MA_5Y	0.0006
23	SP_DYN_CBRT_IN	0.0006
28	SH_DYN_NMRT	0.0006
6	SP_DYN_LE00_IN	0.0005
1	NY_GNP_PCAP_CD	0.0005
47	SP_POP_6064_FE_5Y	0.0005

11	SP_POP_1014_MA_5Y	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
3	NY_GDP_PCAP_KD	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
63	SP_RUR_TOTL_ZS	0.0003
10	SP_POP_0014_TO_ZS	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
4	NY_GDP_PCAP_CD	0.0003
9	SP_POP_0509_MA_5Y	0.0002
68	SP_POP_DPND	0.0002
22	SP_POP_1014_FE_5Y	0.0002
26	SP_POP_65UP_TO_ZS	0.0001

Model with 74 variables and max depth None:

Training+Validation R^2: 0.99861, RMSE: 0.74312

Testing R^2: 0.98175, RMSE: 2.72687

Mean cross-validation score: 0.98093

	Feature	Importance
0	WB_CC_EST_avg	0.9305
73	CC_EST_prev	0.0180
18	SP_POP_0014_FE_ZS	0.0053
8	SP_POP_0014_MA_ZS	0.0025
29	SP_POP_1519_MA_5Y	0.0022
20	SP_DYN_T065_MA_ZS	0.0016
58	FD_AST_PRVT_GD_ZS	0.0015
64	SG_LAW_INDX	0.0015
54	SP_DYN_AMRT_MA	0.0012
72	SP_POP_1564_FE_ZS	0.0012
50	SP_DYN_TFRT_IN	0.0012
41	SP_DYN_IMRT_FE_IN	0.0009
7	SP_DYN_LE00_FE_IN	0.0009
53	SP_POP_7579_FE_5Y	0.0008
55	NV_AGR_TOTL_ZS	0.0008
40	SP_POP_5559_FE_5Y	0.0008
32	SP_POP_5054_FE_5Y	0.0007
31	SP_POP_1519_FE_5Y	0.0007
34	SP_DYN_IMRT_MA_IN	0.0007
26	SP_POP_65UP_TO_ZS	0.0007
51	SP_POP_4549_FE_5Y	0.0007
46	SP_POP_7074_FE_5Y	0.0007
35	SP_POP_65UP_FE_ZS	0.0007
45	SP_POP_6569_FE_5Y	0.0007
65	NE_CON_PRVT_ZS	0.0007
44	SP_POP_4549_MA_5Y	0.0007
71	NE_EXP_GNFS_ZS	0.0007
69	SP_POP_4044_MA_5Y	0.0007

68	SP_POP_DPND	0.0007
62	SP_URB_TOTL_IN_ZS	0.0007
66	SP_POP_4044_FE_5Y	0.0007
24	SP_POP_6064_MA_5Y	0.0006
3	NY_GDP_PCAP_KD	0.0006
10	SP_POP_0014_TO_ZS	0.0006
37	SP_DYN_IMRT_IN	0.0006
25	SP_POP_0004_FE_5Y	0.0006
60	SP_POP_2024_FE_5Y	0.0006
19	IT_MLT_MAIN_P2	0.0006
27	SP_POP_7579_MA_5Y	0.0006
52	SH_DYN_MORT_FE	0.0005
61	SP_POP_2024_MA_5Y	0.0005
49	SH_DYN_MORT	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0005
63	SP_RUR_TOTL_ZS	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
5	SP_DYN_LE00_MA_IN	0.0005
36	NV_SRV_TOTL_ZS	0.0005
33	SP_POP_80UP_FE_5Y	0.0005
16	SP_POP_7074_MA_5Y	0.0005
42	SP_POP_80UP_MA_5Y	0.0005
23	SP_DYN_CBRT_IN	0.0005
43	SP_DYN_T065_FE_ZS	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004
4	NY_GDP_PCAP_CD	0.0004
9	SP_POP_0509_MA_5Y	0.0004
67	SP_POP_1564_MA_ZS	0.0004
21	SP_POP_0509_FE_5Y	0.0004
59	SP_DYN_AMRT_FE	0.0004
22	SP_POP_1014_FE_5Y	0.0004
30	SP_POP_DPND_YG	0.0004
1	NY_GNP_PCAP_CD	0.0004
39	SP_ADO_TFRT	0.0004
48	SH_DYN_MORT_MA	0.0004
38	SP_POP_DPND_DL	0.0004
17	SP_POP_5559_MA_5Y	0.0003
15	SP_POP_5054_MA_5Y	0.0003
14	SP_POP_6569_MA_5Y	0.0003
13	SP_POP_0004_MA_5Y	0.0003
12	SP_POP_65UP_MA_ZS	0.0003
11	SP_POP_1014_MA_5Y	0.0003
47	SP_POP_6064_FE_5Y	0.0003
28	SH_DYN_NMRT	0.0002
6	SP_DYN_LE00_IN	0.0002
70	SP_POP_1564_TO_ZS	0.0001

Model with 75 variables and max depth None:
 Training+Validation R^2: 0.99967, RMSE: 0.35981
 Testing R^2: 0.98248, RMSE: 2.67159
 Mean cross-validation score: 0.98158

	Feature	Importance
0	WB_CC_EST_avg	0.9243
74	CC_EST_prev	0.0187
18	SP_POP_0014_FE_ZS	0.0045
8	SP_POP_0014_MA_ZS	0.0026
29	SP_POP_1519_MA_5Y	0.0024
20	SP_DYN_TO65_MA_ZS	0.0019
58	FD_AST_PRVT_GD_ZS	0.0017
53	SP_POP_7579_FE_5Y	0.0014
54	SP_DYN_AMRT_MA	0.0013
50	SP_DYN_TFRT_IN	0.0012
64	SG_LAW_INDX	0.0012
66	SP_POP_4044_FE_5Y	0.0011
10	SP_POP_0014_TO_ZS	0.0011
51	SP_POP_4549_FE_5Y	0.0011
46	SP_POP_7074_FE_5Y	0.0011
34	SP_DYN_IMRT_MA_IN	0.0011
72	SP_POP_1564_FE_ZS	0.0010
26	SP_POP_65UP_TO_ZS	0.0010
25	SP_POP_0004_FE_5Y	0.0010
31	SP_POP_1519_FE_5Y	0.0010
44	SP_POP_4549_MA_5Y	0.0009
55	NV_AGR_TOTL_ZS	0.0009
71	NE_EXP_GNFS_ZS	0.0009
7	SP_DYN_LE00_FE_IN	0.0009
41	SP_DYN_IMRT_FE_IN	0.0009
68	SP_POP_DPNd	0.0008
60	SP_POP_2024_FE_5Y	0.0008
36	NV_SRV_TOTL_ZS	0.0007
9	SP_POP_0509_MA_5Y	0.0007
24	SP_POP_6064_MA_5Y	0.0007
61	SP_POP_2024_MA_5Y	0.0007
19	IT_MLT_MAIN_P2	0.0007
69	SP_POP_4044_MA_5Y	0.0007
45	SP_POP_6569_FE_5Y	0.0006
47	SP_POP_6064_FE_5Y	0.0006
43	SP_DYN_TO65_FE_ZS	0.0006
56	TM_VAL_MRCH_HI_ZS	0.0006
52	SH_DYN_MORT_FE	0.0006
40	SP_POP_5559_FE_5Y	0.0006
37	SP_DYN_IMRT_IN	0.0006
3	NY_GDP_PCAP_KD	0.0006
57	FM_AST_PRVT_GD_ZS	0.0006

23	SP_DYN_CBRT_IN	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
65	NE_CON_PRVT_ZS	0.0006
15	SP_POP_5054_MA_5Y	0.0006
73	IT_CEL_SETS_P2	0.0006
42	SP_POP_80UP_MA_5Y	0.0005
33	SP_POP_80UP_FE_5Y	0.0005
32	SP_POP_5054_FE_5Y	0.0005
30	SP_POP_DPND_YG	0.0005
48	SH_DYN_MORT_MA	0.0005
49	SH_DYN_MORT	0.0005
14	SP_POP_6569_MA_5Y	0.0005
59	SP_DYN_AMRT_FE	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
39	SP_ADO_TFRT	0.0004
17	SP_POP_5559_MA_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0004
5	SP_DYN_LEOO_MA_IN	0.0004
6	SP_DYN_LEOO_IN	0.0004
11	SP_POP_1014_MA_5Y	0.0004
1	NY_GNP_PCAP_CD	0.0004
38	SP_POP_DPND_DL	0.0004
21	SP_POP_0509_FE_5Y	0.0004
22	SP_POP_1014_FE_5Y	0.0004
28	SH_DYN_NMRT	0.0004
63	SP_RUR_TOTL_ZS	0.0003
13	SP_POP_0004_MA_5Y	0.0003
67	SP_POP_1564_MA_ZS	0.0003
35	SP_POP_65UP_FE_ZS	0.0003
16	SP_POP_7074_MA_5Y	0.0003
12	SP_POP_65UP_MA_ZS	0.0002
27	SP_POP_7579_MA_5Y	0.0002
70	SP_POP_1564_TO_ZS	0.0001

Model with 76 variables and max depth None:
 Training+Validation R^2: 0.99881, RMSE: 0.68768
 Testing R^2: 0.98114, RMSE: 2.77205
 Mean cross-validation score: 0.98122

	Feature	Importance
0	WB_CC_EST_avg	0.9063
75	CC_EST_prev	0.0217
8	SP_POP_0014_MA_ZS	0.0062
18	SP_POP_0014_FE_ZS	0.0032
29	SP_POP_1519_MA_5Y	0.0030
50	SP_DYN_TFRT_IN	0.0019
66	SP_POP_4044_FE_5Y	0.0018

64	SG_LAW_INDX	0.0017
72	SP_POP_1564_FE_ZS	0.0016
20	SP_DYN_T065_MA_ZS	0.0016
51	SP_POP_4549_FE_5Y	0.0015
58	FD_AST_PRVT_GD_ZS	0.0015
49	SH_DYN_MORT	0.0013
53	SP_POP_7579_FE_5Y	0.0013
47	SP_POP_6064_FE_5Y	0.0013
23	SP_DYN_CBRT_IN	0.0013
52	SH_DYN_MORT_FE	0.0012
54	SP_DYN_AMRT_MA	0.0012
45	SP_POP_6569_FE_5Y	0.0011
44	SP_POP_4549_MA_5Y	0.0011
61	SP_POP_2024_MA_5Y	0.0011
74	AG_YLD_CREL_KG	0.0011
32	SP_POP_5054_FE_5Y	0.0010
31	SP_POP_1519_FE_5Y	0.0010
41	SP_DYN_IMRT_FE_IN	0.0010
24	SP_POP_6064_MA_5Y	0.0010
71	NE_EXP_GNFS_ZS	0.0010
25	SP_POP_0004_FE_5Y	0.0009
26	SP_POP_65UP_TO_ZS	0.0009
42	SP_POP_80UP_MA_5Y	0.0009
12	SP_POP_65UP_MA_ZS	0.0009
60	SP_POP_2024_FE_5Y	0.0009
21	SP_POP_0509_FE_5Y	0.0009
34	SP_DYN_IMRT_MA_IN	0.0009
69	SP_POP_4044_MA_5Y	0.0009
73	IT_CEL_SETS_P2	0.0008
46	SP_POP_7074_FE_5Y	0.0008
48	SH_DYN_MORT_MA	0.0008
39	SP_ADO_TFR	0.0008
36	NV_SRV_TOTL_ZS	0.0008
13	SP_POP_0004_MA_5Y	0.0008
56	TM_VAL_MRCH_HI_ZS	0.0008
65	NE_CON_PRVT_ZS	0.0007
57	FM_AST_PRVT_GD_ZS	0.0007
59	SP_DYN_AMRT_FE	0.0007
2	NY_GDP_PCAP_KD_rel	0.0007
22	SP_POP_1014_FE_5Y	0.0007
19	IT_MLT_MAIN_P2	0.0007
17	SP_POP_5559_MA_5Y	0.0007
38	SP_POP_DPNP_OL	0.0006
3	NY_GDP_PCAP_KD	0.0006
37	SP_DYN_IMRT_IN	0.0006
7	SP_DYN_LE00_FE_IN	0.0006
70	SP_POP_1564_TO_ZS	0.0006
55	NV_AGR_TOTL_ZS	0.0006

27	SP_POP_7579_MA_5Y	0.0006
10	SP_POP_0014_TO_ZS	0.0006
40	SP_POP_5559_FE_5Y	0.0006
11	SP_POP_1014_MA_5Y	0.0005
4	NY_GDP_PCAP_CD	0.0005
5	SP_DYN_LE00_MA_IN	0.0005
67	SP_POP_1564_MA_ZS	0.0005
43	SP_DYN_TO65_FE_ZS	0.0005
62	SP_URB_TOTL_IN_ZS	0.0005
33	SP_POP_80UP_FE_5Y	0.0005
15	SP_POP_5054_MA_5Y	0.0005
63	SP_RUR_TOTL_ZS	0.0004
14	SP_POP_6569_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
28	SH_DYN_NMRT	0.0004
68	SP_POP_DPND	0.0004
9	SP_POP_0509_MA_5Y	0.0004
30	SP_POP_DPND_YG	0.0004
6	SP_DYN_LE00_IN	0.0004
1	NY_GNP_PCAP_CD	0.0004
35	SP_POP_65UP_FE_ZS	0.0002

Model with 77 variables and max depth None:

Training+Validation R^2: 0.99743, RMSE: 1.01044

Testing R^2: 0.98106, RMSE: 2.77762

Mean cross-validation score: 0.98121

	Feature	Importance
0	WB_CC_EST_avg	0.8963
76	CC_EST_prev	0.0259
18	SP_POP_0014_FE_ZS	0.0068
8	SP_POP_0014_MA_ZS	0.0037
29	SP_POP_1519_MA_5Y	0.0032
72	SP_POP_1564_FE_ZS	0.0026
64	SG_LAW_INDX	0.0021
58	FD_AST_PRVT_GD_ZS	0.0021
50	SP_DYN_TFRT_IN	0.0019
54	SP_DYN_AMRT_MA	0.0018
20	SP_DYN_TO65_MA_ZS	0.0016
34	SP_DYN_IMRT_MA_IN	0.0015
51	SP_POP_4549_FE_5Y	0.0015
66	SP_POP_4044_FE_5Y	0.0015
53	SP_POP_7579_FE_5Y	0.0015
45	SP_POP_6569_FE_5Y	0.0014
38	SP_POP_DPND_DL	0.0014
10	SP_POP_0014_TO_ZS	0.0014
71	NE_EXP_GNFS_ZS	0.0014

49	SH_DYN_MORT	0.0013
74	AG_YLD_CREL_KG	0.0012
37	SP_DYN_IMRT_IN	0.0012
73	IT_CEL_SETS_P2	0.0011
52	SH_DYN_MORT_FE	0.0011
40	SP_POP_5559_FE_5Y	0.0011
25	SP_POP_0004_FE_5Y	0.0011
75	SP_URB_GROW	0.0010
33	SP_POP_80UP_FE_5Y	0.0010
14	SP_POP_6569_MA_5Y	0.0010
27	SP_POP_7579_MA_5Y	0.0010
26	SP_POP_65UP_TO_ZS	0.0010
43	SP_DYN_T065_FE_ZS	0.0009
32	SP_POP_5054_FE_5Y	0.0009
42	SP_POP_80UP_MA_5Y	0.0009
36	NV_SRV_TOTL_ZS	0.0009
60	SP_POP_2024_FE_5Y	0.0008
44	SP_POP_4549_MA_5Y	0.0008
46	SP_POP_7074_FE_5Y	0.0008
41	SP_DYN_IMRT_FE_IN	0.0008
39	SP_ADO_TFRT	0.0008
31	SP_POP_1519_FE_5Y	0.0008
21	SP_POP_0509_FE_5Y	0.0008
19	IT_MLT_MAIN_P2	0.0008
12	SP_POP_65UP_MA_ZS	0.0007
57	FM_AST_PRVT_GD_ZS	0.0007
56	TM_VAL_MRCH_HI_ZS	0.0007
28	SH_DYN_NMRT	0.0007
69	SP_POP_4044_MA_5Y	0.0007
47	SP_POP_6064_FE_5Y	0.0007
55	NV_AGR_TOTL_ZS	0.0007
65	NE_CON_PRVT_ZS	0.0007
63	SP_RUR_TOTL_ZS	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
15	SP_POP_5054_MA_5Y	0.0006
11	SP_POP_1014_MA_5Y	0.0006
7	SP_DYN_LE00_FE_IN	0.0006
4	NY_GDP_PCAP_CD	0.0006
59	SP_DYN_AMRT_FE	0.0006
13	SP_POP_0004_MA_5Y	0.0005
61	SP_POP_2024_MA_5Y	0.0005
3	NY_GDP_PCAP_KD	0.0005
1	NY_GNP_PCAP_CD	0.0005
22	SP_POP_1014_FE_5Y	0.0005
48	SH_DYN_MORT_MA	0.0005
24	SP_POP_6064_MA_5Y	0.0005
35	SP_POP_65UP_FE_ZS	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005

16	SP_POP_7074_MA_5Y	0.0004
17	SP_POP_5559_MA_5Y	0.0004
70	SP_POP_1564_TO_ZS	0.0004
30	SP_POP_DPND_YG	0.0004
5	SP_DYN_LE00_MA_IN	0.0004
23	SP_DYN_CBRT_IN	0.0003
67	SP_POP_1564_MA_ZS	0.0003
6	SP_DYN_LE00_IN	0.0003
9	SP_POP_0509_MA_5Y	0.0002
68	SP_POP_DPND	0.0000

Model with 78 variables and max depth None:

Training+Validation R^2: 0.99887, RMSE: 0.66998

Testing R^2: 0.98138, RMSE: 2.7545

Mean cross-validation score: 0.98144

	Feature	Importance
0	WB_CC_EST_avg	0.9102
77	CC_EST_prev	0.0227
58	FD_AST_PRVT_GD_ZS	0.0043
8	SP_POP_0014_MA_ZS	0.0036
29	SP_POP_1519_MA_5Y	0.0034
18	SP_POP_0014_FE_ZS	0.0031
50	SP_DYN_TFRT_IN	0.0019
30	SP_POP_DPND_YG	0.0018
64	SG_LAW_INDX	0.0017
20	SP_DYN_T065_MA_ZS	0.0014
53	SP_POP_7579_FE_5Y	0.0014
72	SP_POP_1564_FE_ZS	0.0014
51	SP_POP_4549_FE_5Y	0.0013
46	SP_POP_7074_FE_5Y	0.0012
43	SP_DYN_T065_FE_ZS	0.0012
66	SP_POP_4044_FE_5Y	0.0011
40	SP_POP_5559_FE_5Y	0.0010
71	NE_EXP_GNFS_ZS	0.0010
41	SP_DYN_IMRT_FE_IN	0.0010
7	SP_DYN_LE00_FE_IN	0.0010
45	SP_POP_6569_FE_5Y	0.0010
76	FM_LBL_BMNY_GD_ZS	0.0009
34	SP_DYN_IMRT_MA_IN	0.0009
69	SP_POP_4044_MA_5Y	0.0009
52	SH_DYN_MORT_FE	0.0009
33	SP_POP_80UP_FE_5Y	0.0009
47	SP_POP_6064_FE_5Y	0.0008
74	AG_YLD_CREL_KG	0.0008
38	SP_POP_DPND_OL	0.0008
62	SP_URB_TOTL_IN_ZS	0.0008

75	SP_URB_GROW	0.0008
39	SP_ADO_TFRT	0.0008
11	SP_POP_1014_MA_5Y	0.0008
14	SP_POP_6569_MA_5Y	0.0008
26	SP_POP_65UP_TO_ZS	0.0008
32	SP_POP_5054_FE_5Y	0.0007
44	SP_POP_4549_MA_5Y	0.0007
73	IT_CEL_SETS_P2	0.0007
60	SP_POP_2024_FE_5Y	0.0007
57	FM_AST_PRVT_GD_ZS	0.0007
54	SP_DYN_AMRT_MA	0.0007
17	SP_POP_5559_MA_5Y	0.0007
19	IT_MLT_MAIN_P2	0.0007
21	SP_POP_0509_FE_5Y	0.0007
55	NV_AGR_TOTL_ZS	0.0007
36	NV_SRV_TOTL_ZS	0.0007
24	SP_POP_6064_MA_5Y	0.0006
37	SP_DYN_IMRT_IN	0.0006
6	SP_DYN_LEOO_IN	0.0006
25	SP_POP_0004_FE_5Y	0.0006
28	SH_DYN_NMRT	0.0006
65	NE_CON_PRVT_ZS	0.0006
56	TM_VAL_MRCH_HI_ZS	0.0006
63	SP_RUR_TOTL_ZS	0.0006
4	NY_GDP_PCAP_CD	0.0005
5	SP_DYN_LEOO_MA_IN	0.0005
35	SP_POP_65UP_FE_ZS	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
13	SP_POP_0004_MA_5Y	0.0005
61	SP_POP_2024_MA_5Y	0.0005
15	SP_POP_5054_MA_5Y	0.0005
22	SP_POP_1014_FE_5Y	0.0005
16	SP_POP_7074_MA_5Y	0.0005
42	SP_POP_80UP_MA_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0004
31	SP_POP_1519_FE_5Y	0.0004
67	SP_POP_1564_MA_ZS	0.0004
59	SP_DYN_AMRT_FE	0.0003
48	SH_DYN_MORT_MA	0.0003
49	SH_DYN_MORT	0.0003
9	SP_POP_0509_MA_5Y	0.0003
3	NY_GDP_PCAP_KD	0.0003
23	SP_DYN_CBRT_IN	0.0003
68	SP_POP_DPNP	0.0003
1	NY_GNP_PCAP_CD	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
27	SP_POP_7579_MA_5Y	0.0002
70	SP_POP_1564_TO_ZS	0.0000

Model with 79 variables and max depth None:
 Training+Validation R^2: 0.99753, RMSE: 0.9911
 Testing R^2: 0.98034, RMSE: 2.83014
 Mean cross-validation score: 0.98124

	Feature	Importance
0	WB_CC_EST_avg	0.9164
78	CC_EST_prev	0.0218
18	SP_POP_0014_FE_ZS	0.0031
58	FD_AST_PRVT_GD_ZS	0.0026
29	SP_POP_1519_MA_5Y	0.0022
8	SP_POP_0014_MA_ZS	0.0020
30	SP_POP_DPNP_YG	0.0017
64	SG_LAW_INDX	0.0016
45	SP_POP_6569_FE_5Y	0.0016
20	SP_DYN_T065_MA_ZS	0.0015
49	SH_DYN_MORT	0.0014
53	SP_POP_7579_FE_5Y	0.0014
54	SP_DYN_AMRT_MA	0.0013
25	SP_POP_0004_FE_5Y	0.0013
50	SP_DYN_TFRT_IN	0.0013
72	SP_POP_1564_FE_ZS	0.0012
26	SP_POP_65UP_TO_ZS	0.0012
31	SP_POP_1519_FE_5Y	0.0012
66	SP_POP_4044_FE_5Y	0.0012
75	SP_URB_GROW	0.0010
51	SP_POP_4549_FE_5Y	0.0010
19	IT_MLT_MAIN_P2	0.0010
38	SP_POP_DPNP_DL	0.0009
74	AG_YLD_CREL_KG	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
46	SP_POP_7074_FE_5Y	0.0008
60	SP_POP_2024_FE_5Y	0.0008
52	SH_DYN_MORT_FE	0.0008
71	NE_EXP_GNFS_ZS	0.0008
34	SP_DYN_IMRT_MA_IN	0.0008
76	FM_LBL_BMNY_GD_ZS	0.0007
63	SP_RUR_TOTL_ZS	0.0007
44	SP_POP_4549_MA_5Y	0.0007
43	SP_DYN_T065_FE_ZS	0.0007
55	NV_AGR_TOTL_ZS	0.0007
21	SP_POP_0509_FE_5Y	0.0007
33	SP_POP_80UP_FE_5Y	0.0007
32	SP_POP_5054_FE_5Y	0.0007
61	SP_POP_2024_MA_5Y	0.0007
73	IT_CEL_SETS_P2	0.0007

69	SP_POP_4044_MA_5Y	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
14	SP_POP_6569_MA_5Y	0.0006
47	SP_POP_6064_FE_5Y	0.0006
17	SP_POP_5559_MA_5Y	0.0006
23	SP_DYN_CBRT_IN	0.0006
65	NE_CON_PRVT_ZS	0.0006
57	FM_AST_PRVT_GD_ZS	0.0006
56	TM_VAL_MRCH_HI_ZS	0.0005
62	SP_URB_TOTL_IN_ZS	0.0005
59	SP_DYN_AMRT_FE	0.0005
39	SP_ADO_TFRT	0.0005
48	SH_DYN_MORT_MA	0.0005
36	NV_SRV_TOTL_ZS	0.0005
13	SP_POP_0004_MA_5Y	0.0005
11	SP_POP_1014_MA_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0005
6	SP_DYN_LE00_IN	0.0005
77	SI_POV_GINI	0.0005
67	SP_POP_1564_MA_ZS	0.0005
42	SP_POP_80UP_MA_5Y	0.0004
37	SP_DYN_IMRT_IN	0.0004
24	SP_POP_6064_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
15	SP_POP_5054_MA_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0004
70	SP_POP_1564_TO_ZS	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
68	SP_POP_DPNP	0.0003
3	NY_GDP_PCAP_KD	0.0003
1	NY_GNP_PCAP_CD	0.0003
28	SH_DYN_NMRT	0.0003
27	SP_POP_7579_MA_5Y	0.0003
22	SP_POP_1014_FE_5Y	0.0003
9	SP_POP_0509_MA_5Y	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
40	SP_POP_5559_FE_5Y	0.0003
35	SP_POP_65UP_FE_ZS	0.0002

Model with 80 variables and max depth None:
 Training+Validation R^2: 0.99804, RMSE: 0.88151
 Testing R^2: 0.98116, RMSE: 2.7703
 Mean cross-validation score: 0.98142

	Feature	Importance
0	WB_CC_EST_avg	0.9036

79	CC_EST_prev	0.0242
18	SP_POP_0014_FE_ZS	0.0064
58	FD_AST_PRVT_GD_ZS	0.0038
29	SP_POP_1519_MA_5Y	0.0033
8	SP_POP_0014_MA_ZS	0.0026
30	SP_POP_DPNP_YG	0.0025
50	SP_DYN_TFRT_IN	0.0016
25	SP_POP_0004_FE_5Y	0.0016
64	SG_LAW_INDX	0.0016
72	SP_POP_1564_FE_ZS	0.0015
51	SP_POP_4549_FE_5Y	0.0014
20	SP_DYN_T065_MA_ZS	0.0012
66	SP_POP_4044_FE_5Y	0.0012
31	SP_POP_1519_FE_5Y	0.0012
71	NE_EXP_GNFS_ZS	0.0012
45	SP_POP_6569_FE_5Y	0.0011
34	SP_DYN_IMRT_MA_IN	0.0011
60	SP_POP_2024_FE_5Y	0.0010
62	SP_URB_TOTL_IN_ZS	0.0010
73	IT_CEL_SETS_P2	0.0010
53	SP_POP_7579_FE_5Y	0.0010
47	SP_POP_6064_FE_5Y	0.0010
46	SP_POP_7074_FE_5Y	0.0009
76	FM_LBL_BMNY_GD_ZS	0.0009
44	SP_POP_4549_MA_5Y	0.0009
19	IT_MLT_MAIN_P2	0.0008
74	AG_YLD_CREL_KG	0.0008
43	SP_DYN_T065_FE_ZS	0.0008
75	SP_URB_GROW	0.0008
78	NE_TRD_GNFS_ZS	0.0008
32	SP_POP_5054_FE_5Y	0.0008
54	SP_DYN_AMRT_MA	0.0008
55	NV_AGR_TOTL_ZS	0.0008
57	FM_AST_PRVT_GD_ZS	0.0008
21	SP_POP_0509_FE_5Y	0.0008
38	SP_POP_DPNP_OL	0.0007
48	SH_DYN_MORT_MA	0.0007
40	SP_POP_5559_FE_5Y	0.0007
36	NV_SRV_TOTL_ZS	0.0007
59	SP_DYN_AMRT_FE	0.0007
33	SP_POP_80UP_FE_5Y	0.0007
24	SP_POP_6064_MA_5Y	0.0007
63	SP_RUR_TOTL_ZS	0.0007
17	SP_POP_5559_MA_5Y	0.0007
65	NE_CON_PRVT_ZS	0.0007
14	SP_POP_6569_MA_5Y	0.0007
7	SP_DYN_LE00_FE_IN	0.0007
56	TM_VAL_MRCH_HI_ZS	0.0006

52	SH_DYN_MORT_FE	0.0006
77	SI_POV_GINI	0.0006
61	SP_POP_2024_MA_5Y	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
49	SH_DYN_MORT	0.0006
37	SP_DYN_IMRT_IN	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
35	SP_POP_65UP_FE_ZS	0.0006
15	SP_POP_5054_MA_5Y	0.0006
39	SP_ADO_TFRT	0.0006
22	SP_POP_1014_FE_5Y	0.0005
5	SP_DYN_LE00_MA_IN	0.0005
3	NY_GDP_PCAP_KD	0.0005
13	SP_POP_0004_MA_5Y	0.0005
42	SP_POP_80UP_MA_5Y	0.0005
69	SP_POP_4044_MA_5Y	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
23	SP_DYN_CBRT_IN	0.0004
70	SP_POP_1564_TO_ZS	0.0004
27	SP_POP_7579_MA_5Y	0.0004
11	SP_POP_1014_MA_5Y	0.0004
9	SP_POP_0509_MA_5Y	0.0004
6	SP_DYN_LE00_IN	0.0004
28	SH_DYN_NMRT	0.0004
4	NY_GDP_PCAP_CD	0.0004
16	SP_POP_7074_MA_5Y	0.0003
67	SP_POP_1564_MA_ZS	0.0003
68	SP_POP_DPND	0.0003
1	NY_GNP_PCAP_CD	0.0003
26	SP_POP_65UP_TO_ZS	0.0002
10	SP_POP_0014_TO_ZS	0.0002

Model with 81 variables and max depth None:

Training+Validation R^2: 0.99873, RMSE: 0.71074

Testing R^2: 0.98068, RMSE: 2.80555

Mean cross-validation score: 0.98103

	Feature	Importance
0	WB_CC_EST_avg	0.9261
80	CC_EST_prev	0.0196
18	SP_POP_0014_FE_ZS	0.0028
58	FD_AST_PRVT_GD_ZS	0.0023
8	SP_POP_0014_MA_ZS	0.0019
30	SP_POP_DPND_YG	0.0019
29	SP_POP_1519_MA_5Y	0.0018
64	SG_LAW_INDX	0.0015
50	SP_DYN_TFRT_IN	0.0013

49	SH_DYN_MORT	0.0013
72	SP_POP_1564_FE_ZS	0.0011
71	NE_EXP_GNFS_ZS	0.0011
25	SP_POP_0004_FE_5Y	0.0010
53	SP_POP_7579_FE_5Y	0.0010
66	SP_POP_4044_FE_5Y	0.0010
70	SP_POP_1564_TO_ZS	0.0010
48	SH_DYN_MORT_MA	0.0010
21	SP_POP_0509_FE_5Y	0.0010
20	SP_DYN_T065_MA_ZS	0.0010
34	SP_DYN_IMRT_MA_IN	0.0009
51	SP_POP_4549_FE_5Y	0.0009
45	SP_POP_6569_FE_5Y	0.0008
74	AG_YLD_CREL_KG	0.0008
54	SP_DYN_AMRT_MA	0.0008
76	FM_LBL_BMNY_GD_ZS	0.0008
46	SP_POP_7074_FE_5Y	0.0007
60	SP_POP_2024_FE_5Y	0.0007
32	SP_POP_5054_FE_5Y	0.0007
75	SP_URB_GROW	0.0007
73	IT_CEL_SETS_P2	0.0007
78	NE_TRD_GNFS_ZS	0.0007
36	NV_SRV_TOTL_ZS	0.0006
52	SH_DYN_MORT_FE	0.0006
55	NV_AGR_TOTL_ZS	0.0006
19	IT_MLT_MAIN_P2	0.0006
38	SP_POP_DPND_DL	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
31	SP_POP_1519_FE_5Y	0.0006
33	SP_POP_80UP_FE_5Y	0.0006
57	FM_AST_PRVT_GD_ZS	0.0005
17	SP_POP_5559_MA_5Y	0.0005
59	SP_DYN_AMRT_FE	0.0005
10	SP_POP_0014_TO_ZS	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
14	SP_POP_6569_MA_5Y	0.0005
15	SP_POP_5054_MA_5Y	0.0005
7	SP_DYN_LE00_FE_IN	0.0005
69	SP_POP_4044_MA_5Y	0.0005
44	SP_POP_4549_MA_5Y	0.0005
62	SP_URB_TOTL_IN_ZS	0.0005
77	SI_POV_GINI	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
39	SP_ADO_TFR	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0005
67	SP_POP_1564_MA_ZS	0.0004
68	SP_POP_DPND	0.0004
63	SP_RUR_TOTL_ZS	0.0004

65	NE_CON_PRVT_ZS	0.0004
61	SP_POP_2024_MA_5Y	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
47	SP_POP_6064_FE_5Y	0.0004
24	SP_POP_6064_MA_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0004
5	SP_DYN_LE00_MA_IN	0.0004
13	SP_POP_0004_MA_5Y	0.0004
43	SP_DYN_T065_FE_ZS	0.0004
79	NY_GDP_TOTL_RT_ZS	0.0003
11	SP_POP_1014_MA_5Y	0.0003
22	SP_POP_1014_FE_5Y	0.0003
23	SP_DYN_CBRT_IN	0.0003
27	SP_POP_7579_MA_5Y	0.0003
28	SH_DYN_NMRT	0.0003
35	SP_POP_65UP_FE_ZS	0.0003
3	NY_GDP_PCAP_KD	0.0003
42	SP_POP_80UP_MA_5Y	0.0003
40	SP_POP_5559_FE_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0002
9	SP_POP_0509_MA_5Y	0.0002
6	SP_DYN_LE00_IN	0.0002
1	NY_GNP_PCAP_CD	0.0002
37	SP_DYN_IMRT_IN	0.0001

Model with 82 variables and max depth None:

Training+Validation R^2: 0.99638, RMSE: 1.20004

Testing R^2: 0.98088, RMSE: 2.79083

Mean cross-validation score: 0.9809

	Feature	Importance
0	WB_CC_EST_avg	0.9378
81	CC_EST_prev	0.0174
18	SP_POP_0014_FE_ZS	0.0025
29	SP_POP_1519_MA_5Y	0.0019
58	FD_AST_PRVT_GD_ZS	0.0018
8	SP_POP_0014_MA_ZS	0.0018
64	SG_LAW_INDX	0.0013
34	SP_DYN_IMRT_MA_IN	0.0013
45	SP_POP_6569_FE_5Y	0.0012
30	SP_POP_DPNP_YG	0.0010
20	SP_DYN_T065_MA_ZS	0.0010
50	SP_DYN_TFRT_IN	0.0010
72	SP_POP_1564_FE_ZS	0.0009
66	SP_POP_4044_FE_5Y	0.0008
37	SP_DYN_IMRT_IN	0.0008
53	SP_POP_7579_FE_5Y	0.0008

71	NE_EXP_GNFS_ZS	0.0007
49	SH_DYN_MORT	0.0007
38	SP_POP_DPND_DL	0.0007
76	FM_LBL_BMNY_GD_ZS	0.0007
78	NE_TRD_GNFS_ZS	0.0007
51	SP_POP_4549_FE_5Y	0.0007
54	SP_DYN_AMRT_MA	0.0007
32	SP_POP_5054_FE_5Y	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
31	SP_POP_1519_FE_5Y	0.0006
60	SP_POP_2024_FE_5Y	0.0006
73	IT_CEL_SETS_P2	0.0006
74	AG_YLD_CREL_KG	0.0006
19	IT_MLT_MAIN_P2	0.0006
36	NV_SRV_TOTL_ZS	0.0005
77	SI_POV_GINI	0.0005
44	SP_POP_4549_MA_5Y	0.0005
25	SP_POP_0004_FE_5Y	0.0005
75	SP_URB_GROW	0.0005
48	SH_DYN_MORT_MA	0.0005
55	NV_AGR_TOTL_ZS	0.0005
70	SP_POP_1564_TO_ZS	0.0005
47	SP_POP_6064_FE_5Y	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004
5	SP_DYN_LEOO_MA_IN	0.0004
79	NY_GDP_TOTL_RT_ZS	0.0004
62	SP_URB_TOTL_IN_ZS	0.0004
7	SP_DYN_LEOO_FE_IN	0.0004
80	DTNFL_UNDP_CD	0.0004
59	SP_DYN_AMRT_FE	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
26	SP_POP_65UP_TO_ZS	0.0004
46	SP_POP_7074_FE_5Y	0.0004
13	SP_POP_0004_MA_5Y	0.0004
14	SP_POP_6569_MA_5Y	0.0004
40	SP_POP_5559_FE_5Y	0.0004
15	SP_POP_5054_MA_5Y	0.0004
17	SP_POP_5559_MA_5Y	0.0004
68	SP_POP_DPND	0.0004
33	SP_POP_80UP_FE_5Y	0.0004
21	SP_POP_0509_FE_5Y	0.0004
65	NE_CON_PRVT_ZS	0.0003
61	SP_POP_2024_MA_5Y	0.0003
63	SP_RUR_TOTL_ZS	0.0003
42	SP_POP_80UP_MA_5Y	0.0003
24	SP_POP_6064_MA_5Y	0.0003
4	NY_GDP_PCAP_CD	0.0003

39	SP_ADO_TFRT	0.0003
10	SP_POP_0014_TO_ZS	0.0003
11	SP_POP_1014_MA_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0003
43	SP_DYN_T065_FE_ZS	0.0002
9	SP_POP_0509_MA_5Y	0.0002
22	SP_POP_1014_FE_5Y	0.0002
23	SP_DYN_CBRT_IN	0.0002
2	NY_GDP_PCAP_KD_rel	0.0002
27	SP_POP_7579_MA_5Y	0.0002
28	SH_DYN_NMRT	0.0002
69	SP_POP_4044_MA_5Y	0.0002
35	SP_POP_65UP_FE_ZS	0.0002
1	NY_GNP_PCAP_CD	0.0002
3	NY_GDP_PCAP_KD	0.0002
67	SP_POP_1564_MA_ZS	0.0001
6	SP_DYN_LE00_IN	0.0001
52	SH_DYN_MORT_FE	0.0001

Model with 83 variables and max depth None:

Training+Validation R^2: 0.99941, RMSE: 0.48531

Testing R^2: 0.98024, RMSE: 2.83698

Mean cross-validation score: 0.98073

	Feature	Importance
0	WB_CC_EST_avg	0.9142
82	CC_EST_prev	0.0224
29	SP_POP_1519_MA_5Y	0.0044
18	SP_POP_0014_FE_ZS	0.0029
8	SP_POP_0014_MA_ZS	0.0024
58	FD_AST_PRVT_GD_ZS	0.0018
64	SG_LAW_INDX	0.0016
72	SP_POP_1564_FE_ZS	0.0016
34	SP_DYN_IMRT_MA_IN	0.0015
20	SP_DYN_T065_MA_ZS	0.0013
50	SP_DYN_TFRT_IN	0.0013
48	SH_DYN_MORT_MA	0.0013
30	SP_POP_DPND_YG	0.0012
54	SP_DYN_AMRT_MA	0.0012
66	SP_POP_4044_FE_5Y	0.0011
59	SP_DYN_AMRT_FE	0.0011
26	SP_POP_65UP_TO_ZS	0.0010
7	SP_DYN_LE00_FE_IN	0.0010
33	SP_POP_80UP_FE_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0010
71	NE_EXP_GNFS_ZS	0.0009
21	SP_POP_0509_FE_5Y	0.0009

74	AG_YLD_CREL_KG	0.0009
25	SP_POP_0004_FE_5Y	0.0009
51	SP_POP_4549_FE_5Y	0.0009
53	SP_POP_7579_FE_5Y	0.0008
73	IT_CEL_SETS_P2	0.0008
75	SP_URB_GROW	0.0008
10	SP_POP_0014_TO_ZS	0.0008
31	SP_POP_1519_FE_5Y	0.0008
60	SP_POP_2024_FE_5Y	0.0007
39	SP_ADO_TFRT	0.0007
42	SP_POP_80UP_MA_5Y	0.0007
32	SP_POP_5054_FE_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
80	DT_NFL_UNDP_CD	0.0007
78	NE_TRD_GNFS_ZS	0.0007
68	SP_POP_DPND	0.0007
69	SP_POP_4044_MA_5Y	0.0007
79	NY_GDP_TOTL_RT_ZS	0.0006
49	SH_DYN_MORT	0.0006
77	SI_POV_GINI	0.0006
76	FM_LBL_BMNY_GD_ZS	0.0006
46	SP_POP_7074_FE_5Y	0.0006
44	SP_POP_4549_MA_5Y	0.0006
56	TM_VAL_MRCH_HI_ZS	0.0006
63	SP_RUR_TOTL_ZS	0.0006
24	SP_POP_6064_MA_5Y	0.0006
38	SP_POP_DPND_DL	0.0006
28	SH_DYN_NMRT	0.0006
36	NV_SRV_TOTL_ZS	0.0006
67	SP_POP_1564_MA_ZS	0.0006
55	NV_AGR_TOTL_ZS	0.0006
70	SP_POP_1564_TO_ZS	0.0005
57	FM_AST_PRVT_GD_ZS	0.0005
81	NY_ADJ_DRES_GN_ZS	0.0005
41	SP_DYN_IMRT_FE_IN	0.0005
40	SP_POP_5559_FE_5Y	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
35	SP_POP_65UP_FE_ZS	0.0005
23	SP_DYN_CBRT_IN	0.0005
47	SP_POP_6064_FE_5Y	0.0005
19	IT_MLT_MAIN_P2	0.0005
15	SP_POP_5054_MA_5Y	0.0005
52	SH_DYN_MORT_FE	0.0004
4	NY_GDP_PCAP_CD	0.0004
5	SP_DYN_LE00_MA_IN	0.0004
61	SP_POP_2024_MA_5Y	0.0004
62	SP_URB_TOTL_IN_ZS	0.0004
11	SP_POP_1014_MA_5Y	0.0004

65	NE_CON_PRVT_ZS	0.0004
13	SP_POP_0004_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
17	SP_POP_5559_MA_5Y	0.0003
1	NY_GNP_PCAP_CD	0.0003
3	NY_GDP_PCAP_KD	0.0003
27	SP_POP_7579_MA_5Y	0.0002
37	SP_DYN_IMRT_IN	0.0002
6	SP_DYN_LEOO_IN	0.0002
43	SP_DYN_T065_FE_ZS	0.0002
22	SP_POP_1014_FE_5Y	0.0002
9	SP_POP_0509_MA_5Y	0.0001

Model with 84 variables and max depth None:

Training+Validation R^2: 0.99811, RMSE: 0.86655

Testing R^2: 0.98027, RMSE: 2.83485

Mean cross-validation score: 0.98056

	Feature	Importance
0	WB_CC_EST_avg	0.9090
83	CC_EST_prev	0.0223
18	SP_POP_0014_FE_ZS	0.0041
8	SP_POP_0014_MA_ZS	0.0032
29	SP_POP_1519_MA_5Y	0.0032
64	SG_LAW_INDX	0.0020
50	SP_DYN_TFRT_IN	0.0019
58	FD_AST_PRVT_GD_ZS	0.0018
51	SP_POP_4549_FE_5Y	0.0017
20	SP_DYN_T065_MA_ZS	0.0017
25	SP_POP_0004_FE_5Y	0.0015
72	SP_POP_1564_FE_ZS	0.0014
45	SP_POP_6569_FE_5Y	0.0014
53	SP_POP_7579_FE_5Y	0.0012
71	NE_EXP_GNFS_ZS	0.0011
47	SP_POP_6064_FE_5Y	0.0011
49	SH_DYN_MORT	0.0011
26	SP_POP_65UP_TO_ZS	0.0011
40	SP_POP_5559_FE_5Y	0.0009
33	SP_POP_80UP_FE_5Y	0.0009
78	NE_TRD_GNFS_ZS	0.0009
34	SP_DYN_IMRT_MA_IN	0.0009
80	DTNFL_UNDP_CD	0.0009
35	SP_POP_65UP_FE_ZS	0.0009
82	EN.URB.MCTY.TL.ZS	0.0009
66	SP_POP_4044_FE_5Y	0.0009
60	SP_POP_2024_FE_5Y	0.0008

19	IT_MLT_MAIN_P2	0.0008
21	SP_POP_0509_FE_5Y	0.0008
75	SP_URB_GROW	0.0008
76	FM_LBL_BMNY_GD_ZS	0.0008
77	SI_POV_GINI	0.0008
52	SH_DYN_MORT_FE	0.0008
79	NY_GDP_TOTL_RT_ZS	0.0008
46	SP_POP_7074_FE_5Y	0.0007
54	SP_DYN_AMRT_MA	0.0007
42	SP_POP_80UP_MA_5Y	0.0007
59	SP_DYN_AMRT_FE	0.0007
74	AG_YLD_CREL_KG	0.0007
68	SP_POP_DPND	0.0007
36	NV_SRV_TOTL_ZS	0.0007
81	NY_ADJ_DRES_GN_ZS	0.0007
6	SP_DYN_LE00_IN	0.0007
73	IT_CEL_SETS_P2	0.0007
14	SP_POP_6569_MA_5Y	0.0006
57	FM_AST_PRVT_GD_ZS	0.0006
61	SP_POP_2024_MA_5Y	0.0006
55	NV_AGR_TOTL_ZS	0.0006
10	SP_POP_0014_TO_ZS	0.0006
12	SP_POP_65UP_MA_ZS	0.0006
38	SP_POP_DPND_OL	0.0006
17	SP_POP_5559_MA_5Y	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
24	SP_POP_6064_MA_5Y	0.0006
69	SP_POP_4044_MA_5Y	0.0006
32	SP_POP_5054_FE_5Y	0.0006
65	NE_CON_PRVT_ZS	0.0006
31	SP_POP_1519_FE_5Y	0.0006
67	SP_POP_1564_MA_ZS	0.0005
63	SP_RUR_TOTL_ZS	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0005
39	SP_ADO_TFRT	0.0005
7	SP_DYN_LE00_FE_IN	0.0005
28	SH_DYN_NMRT	0.0005
48	SH_DYN_MORT_MA	0.0005
22	SP_POP_1014_FE_5Y	0.0005
11	SP_POP_1014_MA_5Y	0.0005
13	SP_POP_0004_MA_5Y	0.0004
44	SP_POP_4549_MA_5Y	0.0004
1	NY_GNP_PCAP_CD	0.0004
30	SP_POP_DPND_YG	0.0004
23	SP_DYN_CBRT_IN	0.0004
16	SP_POP_7074_MA_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0003

5	SP_DYN_LE00_MA_IN	0.0003
9	SP_POP_0509_MA_5Y	0.0003
27	SP_POP_7579_MA_5Y	0.0003
15	SP_POP_5054_MA_5Y	0.0003
37	SP_DYN_IMRT_IN	0.0003
41	SP_DYN_IMRT_FE_IN	0.0003
3	NY_GDP_PCAP_KD	0.0003
43	SP_DYN_T065_FE_ZS	0.0002
70	SP_POP_1564_TO_ZS	0.0001

Model with 85 variables and max depth None:

Training+Validation R^2: 0.99938, RMSE: 0.49602

Testing R^2: 0.98126, RMSE: 2.76332

Mean cross-validation score: 0.98086

	Feature	Importance
0	WB_CC_EST_avg	0.8975
84	CC_EST_prev	0.0263
29	SP_POP_1519_MA_5Y	0.0049
8	SP_POP_0014_MA_ZS	0.0042
18	SP_POP_0014_FE_ZS	0.0039
25	SP_POP_0004_FE_5Y	0.0027
50	SP_DYN_TFRT_IN	0.0021
51	SP_POP_4549_FE_5Y	0.0020
72	SP_POP_1564_FE_ZS	0.0020
58	FD_AST_PRVT_GD_ZS	0.0019
20	SP_DYN_T065_MA_ZS	0.0019
64	SG_LAW_INDX	0.0013
33	SP_POP_80UP_FE_5Y	0.0013
71	NE_EXP_GNFS_ZS	0.0013
45	SP_POP_6569_FE_5Y	0.0013
53	SP_POP_7579_FE_5Y	0.0013
54	SP_DYN_AMRT_MA	0.0013
78	NE_TRD_GNFS_ZS	0.0012
74	AG_YLD_CREL_KG	0.0011
75	SP.URB.GROW	0.0011
32	SP_POP_5054_FE_5Y	0.0011
73	IT_CEL_SETS_P2	0.0010
16	SP_POP_7074_MA_5Y	0.0010
31	SP_POP_1519_FE_5Y	0.0010
66	SP_POP_4044_FE_5Y	0.0010
46	SP_POP_7074_FE_5Y	0.0009
65	NE_CON_PRVT_ZS	0.0009
27	SP_POP_7579_MA_5Y	0.0009
34	SP_DYN_IMRT_MA_IN	0.0008
82	EN.URB.MCTY.TL.ZS	0.0008
12	SP_POP_65UP_MA_ZS	0.0008

39	SP_ADO_TFRT	0.0008
36	NV_SRV_TOTL_ZS	0.0008
21	SP_POP_0509_FE_5Y	0.0008
6	SP_DYN_LE00_IN	0.0008
76	FM_LBL_BMNY_GD_ZS	0.0008
77	SI_POV_GINI	0.0008
83	TX_VAL_MRCH_HI_ZS	0.0008
44	SP_POP_4549_MA_5Y	0.0007
56	TM_VAL_MRCH_HI_ZS	0.0007
7	SP_DYN_LE00_FE_IN	0.0007
60	SP_POP_2024_FE_5Y	0.0007
49	SH_DYN_MORT	0.0007
55	NV_AGR_TOTL_ZS	0.0007
43	SP_DYN_T065_FE_ZS	0.0007
19	IT_MLT_MAIN_P2	0.0007
17	SP_POP_5559_MA_5Y	0.0007
63	SP_RUR_TOTL_ZS	0.0007
5	SP_DYN_LE00_MA_IN	0.0006
67	SP_POP_1564_MA_ZS	0.0006
61	SP_POP_2024_MA_5Y	0.0006
23	SP_DYN_CBRT_IN	0.0006
57	FM_AST_PRVT_GD_ZS	0.0006
79	NY_GDP_TOTL_RT_ZS	0.0006
69	SP_POP_4044_MA_5Y	0.0006
14	SP_POP_6569_MA_5Y	0.0006
13	SP_POP_0004_MA_5Y	0.0006
38	SP_POP_DPND_DL	0.0006
80	DT_NFL_UNDP_CD	0.0006
22	SP_POP_1014_FE_5Y	0.0006
48	SH_DYN_MORT_MA	0.0006
47	SP_POP_6064_FE_5Y	0.0006
81	NY_ADJ_DRES_GN_ZS	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
62	SP_URB_TOTL_IN_ZS	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
11	SP_POP_1014_MA_5Y	0.0005
15	SP_POP_5054_MA_5Y	0.0005
24	SP_POP_6064_MA_5Y	0.0005
35	SP_POP_65UP_FE_ZS	0.0005
52	SH_DYN_MORT_FE	0.0005
40	SP_POP_5559_FE_5Y	0.0004
37	SP_DYN_IMRT_IN	0.0004
3	NY_GDP_PCAP_KD	0.0004
41	SP_DYN_IMRT_FE_IN	0.0003
59	SP_DYN_AMRT_FE	0.0003
28	SH_DYN_NMRT	0.0003
68	SP_POP_DPND	0.0003
10	SP_POP_0014_TO_ZS	0.0003

4	NY_GDP_PCAP_CD	0.0003
1	NY_GNP_PCAP_CD	0.0002
9	SP_POP_0509_MA_5Y	0.0002
42	SP_POP_80UP_MA_5Y	0.0002
70	SP_POP_1564_TO_ZS	0.0000
30	SP_POP_DPND_YG	0.0000

Model with 86 variables and max depth None:

Training+Validation R^2: 0.99771, RMSE: 0.95333

Testing R^2: 0.9825, RMSE: 2.67023

Mean cross-validation score: 0.98119

	Feature	Importance
0	WB_CC_EST_avg	0.9100
85	CC_EST_prev	0.0213
18	SP_POP_0014_FE_ZS	0.0110
29	SP_POP_1519_MA_5Y	0.0028
8	SP_POP_0014_MA_ZS	0.0025
58	FD_AST_PRVT_GD_ZS	0.0016
9	SP_POP_0509_MA_5Y	0.0015
70	SP_POP_1564_TO_ZS	0.0014
50	SP_DYN_TFRT_IN	0.0013
54	SP_DYN_AMRT_MA	0.0013
51	SP_POP_4549_FE_5Y	0.0012
24	SP_POP_6064_MA_5Y	0.0012
10	SP_POP_0014_TO_ZS	0.0012
20	SP_DYN_T065_MA_ZS	0.0010
64	SG_LAW_INDX	0.0010
45	SP_POP_6569_FE_5Y	0.0009
84	NY_GDP_FRST_RT_ZS	0.0009
53	SP_POP_7579_FE_5Y	0.0009
59	SP_DYN_AMRT_FE	0.0009
60	SP_POP_2024_FE_5Y	0.0009
34	SP_DYN_IMRT_MA_IN	0.0009
33	SP_POP_80UP_FE_5Y	0.0009
23	SP_DYN_CBRT_IN	0.0009
80	DT_NFL_UNDP_CD	0.0008
55	NV_AGR_TOTL_ZS	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
77	SI_POV_GINI	0.0008
71	NE_EXP_GNFS_ZS	0.0008
19	IT_MLT_MAIN_P2	0.0008
69	SP_POP_4044_MA_5Y	0.0007
67	SP_POP_1564_MA_ZS	0.0007
74	AG_YLD_CREL_KG	0.0007
17	SP_POP_5559_MA_5Y	0.0007
82	EN_URB_MCTY_TL_ZS	0.0007

39	SP_ADO_TFRT	0.0007
62	SP_URB_TOTL_IN_ZS	0.0007
72	SP_POP_1564_FE_ZS	0.0007
31	SP_POP_1519_FE_5Y	0.0007
44	SP_POP_4549_MA_5Y	0.0007
63	SP_RUR_TOTL_ZS	0.0006
81	NY_ADJ_DRES_GN_ZS	0.0006
48	SH_DYN_MORT_MA	0.0006
65	NE_CON_PRVT_ZS	0.0006
83	TX_VAL_MRCH_HI_ZS	0.0006
73	IT_CEL_SETS_P2	0.0006
78	NE_TRD_GNFS_ZS	0.0006
66	SP_POP_4044_FE_5Y	0.0006
43	SP_DYN_T065_FE_ZS	0.0006
47	SP_POP_6064_FE_5Y	0.0006
13	SP_POP_0004_MA_5Y	0.0006
36	NV_SRV_TOTL_ZS	0.0006
38	SP_POP_DPND_OL	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
14	SP_POP_6569_MA_5Y	0.0006
46	SP_POP_7074_FE_5Y	0.0006
25	SP_POP_0004_FE_5Y	0.0006
52	SH_DYN_MORT_FE	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0005
68	SP_POP_DPND	0.0005
40	SP_POP_5559_FE_5Y	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
37	SP_DYN_IMRT_IN	0.0005
61	SP_POP_2024_MA_5Y	0.0005
79	NY_GDP_TOTL_RT_ZS	0.0004
76	FM_LBL_BMNY_GD_ZS	0.0004
75	SP_URB_GROW	0.0004
15	SP_POP_5054_MA_5Y	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004
3	NY_GDP_PCAP_KD	0.0004
32	SP_POP_5054_FE_5Y	0.0004
42	SP_POP_80UP_MA_5Y	0.0004
22	SP_POP_1014_FE_5Y	0.0003
21	SP_POP_0509_FE_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0003
6	SP_DYN_LE00_IN	0.0003
11	SP_POP_1014_MA_5Y	0.0003
26	SP_POP_65UP_TO_ZS	0.0003
5	SP_DYN_LE00_MA_IN	0.0002
49	SH_DYN_MORT	0.0002
1	NY_GNP_PCAP_CD	0.0002
35	SP_POP_65UP_FE_ZS	0.0002

30	SP_POP_DPND_YG	0.0002
28	SH_DYN_NMRT	0.0002
27	SP_POP_7579_MA_5Y	0.0002
4	NY_GDP_PCAP_CD	0.0001

Model with 87 variables and max depth None:
 Training+Validation R^2: 0.99921, RMSE: 0.55952
 Testing R^2: 0.98194, RMSE: 2.71244
 Mean cross-validation score: 0.98112

	Feature	Importance
0	WB_CC_EST_avg	0.9035
86	CC_EST_prev	0.0236
18	SP_POP_0014_FE_ZS	0.0044
8	SP_POP_0014_MA_ZS	0.0033
29	SP_POP_1519_MA_5Y	0.0027
50	SP_DYN_TFRT_IN	0.0025
20	SP_DYN_T065_MA_ZS	0.0020
53	SP_POP_7579_FE_5Y	0.0016
23	SP_DYN_CBRT_IN	0.0015
51	SP_POP_4549_FE_5Y	0.0014
45	SP_POP_6569_FE_5Y	0.0014
25	SP_POP_0004_FE_5Y	0.0014
63	SP_RUR_TOTL_ZS	0.0014
7	SP_DYN_LE00_FE_IN	0.0014
84	NY_GDP_FRST_RT_ZS	0.0012
72	SP_POP_1564_FE_ZS	0.0012
64	SG_LAW_INDX	0.0011
55	NV_AGR_TOTL_ZS	0.0011
34	SP_DYN_IMRT_MA_IN	0.0011
54	SP_DYN_AMRT_MA	0.0011
71	NE_EXP_GNFS_ZS	0.0011
82	EN_URB_MCTY_TL_ZS	0.0011
58	FD_AST_PRVT_GD_ZS	0.0010
33	SP_POP_80UP_FE_5Y	0.0010
78	NE_TRD_GNFS_ZS	0.0010
60	SP_POP_2024_FE_5Y	0.0010
10	SP_POP_0014_TO_ZS	0.0010
31	SP_POP_1519_FE_5Y	0.0010
41	SP_DYN_IMRT_FE_IN	0.0009
46	SP_POP_7074_FE_5Y	0.0009
32	SP_POP_5054_FE_5Y	0.0009
80	DTNFL_UNDP_CD	0.0009
26	SP_POP_65UP_TO_ZS	0.0009
74	AG_YLD_CREL_KG	0.0008
75	SP_URB_GROW	0.0008
68	SP_POP_DPND	0.0008

85	SE_SEC_ENRL_GC_FE_ZS	0.0008
73	IT_CEL_SETS_P2	0.0008
37	SP_DYN_IMRT_IN	0.0008
19	IT_MLT_MAIN_P2	0.0007
59	SP_DYN_AMRT_FE	0.0007
77	SI_POV_GINI	0.0007
69	SP_POP_4044_MA_5Y	0.0007
81	NY_ADJ_DRES_GN_ZS	0.0007
56	TM_VAL_MRCH_HI_ZS	0.0007
47	SP_POP_6064_FE_5Y	0.0007
70	SP_POP_1564_TO_ZS	0.0007
67	SP_POP_1564_MA_ZS	0.0007
42	SP_POP_80UP_MA_5Y	0.0007
65	NE_CON_PRVT_ZS	0.0006
66	SP_POP_4044_FE_5Y	0.0006
44	SP_POP_4549_MA_5Y	0.0006
3	NY_GDP_PCAP_KD	0.0006
40	SP_POP_5559_FE_5Y	0.0006
39	SP_ADO_TFRT	0.0006
38	SP_POP_DPND_DL	0.0006
36	NV_SRV_TOTL_ZS	0.0006
24	SP_POP_6064_MA_5Y	0.0006
76	FM_LBL_BMNY_GD_ZS	0.0006
16	SP_POP_7074_MA_5Y	0.0006
83	TX_VAL_MRCH_HI_ZS	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
79	NY_GDP_TOTL_RT_ZS	0.0005
2	NY_GDP_PCAP_KD_rel	0.0005
43	SP_DYN_T065_FE_ZS	0.0005
13	SP_POP_0004_MA_5Y	0.0005
14	SP_POP_6569_MA_5Y	0.0005
17	SP_POP_5559_MA_5Y	0.0005
57	FM_AST_PRVT_GD_ZS	0.0005
22	SP_POP_1014_FE_5Y	0.0004
5	SP_DYN_LE00_MA_IN	0.0004
11	SP_POP_1014_MA_5Y	0.0004
15	SP_POP_5054_MA_5Y	0.0004
21	SP_POP_0509_FE_5Y	0.0004
35	SP_POP_65UP_FE_ZS	0.0004
48	SH_DYN_MORT_MA	0.0004
27	SP_POP_7579_MA_5Y	0.0003
30	SP_POP_DPND_YG	0.0003
1	NY_GNP_PCAP_CD	0.0003
12	SP_POP_65UP_MA_ZS	0.0003
61	SP_POP_2024_MA_5Y	0.0003
28	SH_DYN_NMRT	0.0002
9	SP_POP_0509_MA_5Y	0.0002
6	SP_DYN_LE00_IN	0.0002

52	SH_DYN_MORT_FE	0.0002
4	NY_GDP_PCAP_CD	0.0002
49	SH_DYN_MORT	0.0000

Model with 88 variables and max depth None:
 Training+Validation R^2: 0.99813, RMSE: 0.86092
 Testing R^2: 0.98127, RMSE: 2.76208
 Mean cross-validation score: 0.98057

	Feature	Importance
0	WB_CC_EST_avg	0.9041
87	CC_EST_prev	0.0214
50	SP_DYN_TFRRT_IN	0.0036
8	SP_POP_0014_MA_ZS	0.0035
58	FD_AST_PRVT_GD_ZS	0.0030
29	SP_POP_1519_MA_5Y	0.0030
31	SP_POP_1519_FE_5Y	0.0019
20	SP_DYN_T065_MA_ZS	0.0018
75	SP_URB_GROW	0.0018
45	SP_POP_6569_FE_5Y	0.0017
53	SP_POP_7579_FE_5Y	0.0016
54	SP_DYN_AMRT_MA	0.0015
7	SP_DYN_LE00_FE_IN	0.0015
51	SP_POP_4549_FE_5Y	0.0013
72	SP_POP_1564_FE_ZS	0.0013
74	AG_YLD_CREL_KG	0.0012
40	SP_POP_5559_FE_5Y	0.0012
25	SP_POP_0004_FE_5Y	0.0012
3	NY_GDP_PCAP_KD	0.0012
34	SP_DYN_IMRT_MA_IN	0.0011
41	SP_DYN_IMRT_FE_IN	0.0011
68	SP_POP_DPNP	0.0011
67	SP_POP_1564_MA_ZS	0.0010
86	SP_RUR_TOTL_ZG	0.0010
60	SP_POP_2024_FE_5Y	0.0010
69	SP_POP_4044_MA_5Y	0.0010
71	NE_EXP_GNFS_ZS	0.0010
83	TX_VAL_MRCH_HI_ZS	0.0010
80	DTNFL_UNDP_CD	0.0009
47	SP_POP_6064_FE_5Y	0.0009
37	SP_DYN_IMRT_IN	0.0009
85	SE_SEC_ENRL_GC_FE_ZS	0.0009
78	NE_TRD_GNFS_ZS	0.0009
55	NV_AGR_TOTL_ZS	0.0009
56	TM_VAL_MRCH_HI_ZS	0.0009
64	SG_LAW_INDX	0.0009
35	SP_POP_65UP_FE_ZS	0.0008

49	SH_DYN_MORT	0.0008
81	NY_ADJ_DRES_GN_ZS	0.0007
77	SI_POV_GINI	0.0007
76	FM_LBL_BMNY_GD_ZS	0.0007
61	SP_POP_2024_MA_5Y	0.0007
44	SP_POP_4549_MA_5Y	0.0007
14	SP_POP_6569_MA_5Y	0.0007
19	IT_MLT_MAIN_P2	0.0007
36	NV_SRV_TOTL_ZS	0.0007
32	SP_POP_5054_FE_5Y	0.0007
23	SP_DYN_CBRT_IN	0.0007
65	NE_CON_PRVT_ZS	0.0006
73	IT_CEL_SETS_P2	0.0006
15	SP_POP_5054_MA_5Y	0.0006
21	SP_POP_0509_FE_5Y	0.0006
66	SP_POP_4044_FE_5Y	0.0006
22	SP_POP_1014_FE_5Y	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
57	FM_AST_PRVT_GD_ZS	0.0006
10	SP_POP_0014_TO_ZS	0.0006
6	SP_DYN_LEOO_IN	0.0006
39	SP_ADO_TFRT	0.0006
84	NY_GDP_FRST_RT_ZS	0.0006
17	SP_POP_5559_MA_5Y	0.0006
26	SP_POP_65UP_TO_ZS	0.0005
24	SP_POP_6064_MA_5Y	0.0005
9	SP_POP_0509_MA_5Y	0.0005
82	EN_URB_MCTY_TL_ZS	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
33	SP_POP_80UP_FE_5Y	0.0005
48	SH_DYN_MORT_MA	0.0005
42	SP_POP_80UP_MA_5Y	0.0005
79	NY_GDP_TOTL_RT_ZS	0.0004
11	SP_POP_1014_MA_5Y	0.0004
13	SP_POP_0004_MA_5Y	0.0004
43	SP_DYN_T065_FE_ZS	0.0004
16	SP_POP_7074_MA_5Y	0.0004
27	SP_POP_7579_MA_5Y	0.0004
30	SP_POP_DPND_YG	0.0004
38	SP_POP_DPND_OL	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
63	SP_RUR_TOTL_ZS	0.0003
28	SH_DYN_NMRT	0.0003
46	SP_POP_7074_FE_5Y	0.0003
5	SP_DYN_LEOO_MA_IN	0.0003
4	NY_GDP_PCAP_CD	0.0003
18	SP_POP_0014_FE_ZS	0.0002
1	NY_GNP_PCAP_CD	0.0002

59	SP_DYN_AMRT_FE	0.0002
70	SP_POP_1564_TO_ZS	0.0001
52	SH_DYN_MORT_FE	0.0000

Model with 89 variables and max depth None:
 Training+Validation R^2: 0.99972, RMSE: 0.33648
 Testing R^2: 0.98137, RMSE: 2.75502
 Mean cross-validation score: 0.98098

	Feature	Importance
0	WB_CC_EST_avg	0.9228
88	CC_EST_prev	0.0183
8	SP_POP_0014_MA_ZS	0.0032
29	SP_POP_1519_MA_5Y	0.0029
58	FD_AST_PRVT_GD_ZS	0.0023
50	SP_DYN_TFRT_IN	0.0019
40	SP_POP_5559_FE_5Y	0.0014
7	SP_DYN_LE00_FE_IN	0.0012
20	SP_DYN_T065_MA_ZS	0.0012
45	SP_POP_6569_FE_5Y	0.0011
51	SP_POP_4549_FE_5Y	0.0011
53	SP_POP_7579_FE_5Y	0.0011
54	SP_DYN_AMRT_MA	0.0011
34	SP_DYN_IMRT_MA_IN	0.0011
72	SP_POP_1564_FE_ZS	0.0011
87	TX_VAL_MANF_ZS_UN	0.0010
25	SP_POP_0004_FE_5Y	0.0010
74	AG_YLD_CREL_KG	0.0009
60	SP_POP_2024_FE_5Y	0.0008
26	SP_POP_65UP_TO_ZS	0.0008
31	SP_POP_1519_FE_5Y	0.0008
64	SG_LAW_INDX	0.0008
83	TX_VAL_MRCH_HI_ZS	0.0008
24	SP_POP_6064_MA_5Y	0.0008
86	SP_RUR_TOTL_ZG	0.0007
49	SH_DYN_MORT	0.0007
65	NE_CON_PRVT_ZS	0.0007
77	SI_POV_GINI	0.0007
55	NV_AGR_TOTL_ZS	0.0007
71	NE_EXP_GNFS_ZS	0.0007
78	NE_TRD_GNFS_ZS	0.0007
44	SP_POP_4549_MA_5Y	0.0007
23	SP_DYN_CBRT_IN	0.0007
32	SP_POP_5054_FE_5Y	0.0006
73	IT_CEL_SETS_P2	0.0006
52	SH_DYN_MORT_FE	0.0006
76	FM_LBL_BMNY_GD_ZS	0.0006

56	TM_VAL_MRCH_HI_ZS	0.0006
14	SP_POP_6569_MA_5Y	0.0006
47	SP_POP_6064_FE_5Y	0.0006
80	DT_NFL_UNDP_CD	0.0006
82	EN_URB_MCTY_TL_ZS	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
13	SP_POP_0004_MA_5Y	0.0006
85	SE_SEC_ENRL_GC_FE_ZS	0.0006
38	SP_POP_DPND_DL	0.0006
69	SP_POP_4044_MA_5Y	0.0006
67	SP_POP_1564_MA_ZS	0.0006
30	SP_POP_DPND_YG	0.0006
75	SP_URB_GROW	0.0006
12	SP_POP_65UP_MA_ZS	0.0005
19	IT_MLT_MAIN_P2	0.0005
57	FM_AST_PRVT_GD_ZS	0.0005
61	SP_POP_2024_MA_5Y	0.0005
66	SP_POP_4044_FE_5Y	0.0005
21	SP_POP_0509_FE_5Y	0.0005
46	SP_POP_7074_FE_5Y	0.0005
81	NY_ADJ_DRES_GN_ZS	0.0005
6	SP_DYN_LEOO_IN	0.0005
84	NY_GDP_FRST_RT_ZS	0.0005
37	SP_DYN_IMRT_IN	0.0005
10	SP_POP_0014_TO_ZS	0.0004
5	SP_DYN_LEOO_MA_IN	0.0004
4	NY_GDP_PCAP_CD	0.0004
79	NY_GDP_TOTL_RT_ZS	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
15	SP_POP_5054_MA_5Y	0.0004
36	NV_SRV_TOTL_ZS	0.0004
39	SP_ADO_TFRT	0.0004
28	SH_DYN_NMRT	0.0004
59	SP_DYN_AMRT_FE	0.0004
17	SP_POP_5559_MA_5Y	0.0004
42	SP_POP_80UP_MA_5Y	0.0004
3	NY_GDP_PCAP_KD	0.0003
35	SP_POP_65UP_FE_ZS	0.0003
43	SP_DYN_T065_FE_ZS	0.0003
16	SP_POP_7074_MA_5Y	0.0003
18	SP_POP_0014_FE_ZS	0.0003
11	SP_POP_1014_MA_5Y	0.0003
33	SP_POP_80UP_FE_5Y	0.0003
62	SP_URB_TOTL_IN_ZS	0.0003
68	SP_POP_DPND	0.0003
9	SP_POP_0509_MA_5Y	0.0002
22	SP_POP_1014_FE_5Y	0.0002
1	NY_GNP_PCAP_CD	0.0002

27	SP_POP_7579_MA_5Y	0.0002
70	SP_POP_1564_TO_ZS	0.0001
63	SP_RUR_TOTL_ZS	0.0000
48	SH_DYN_MORT_MA	0.0000

Model with 90 variables and max depth None:
 Training+Validation R^2: 0.99955, RMSE: 0.42368
 Testing R^2: 0.98256, RMSE: 2.66556
 Mean cross-validation score: 0.98106

	Feature	Importance
0	WB_CC_EST_avg	0.9205
89	CC_EST_prev	0.0223
8	SP_POP_0014_MA_ZS	0.0026
29	SP_POP_1519_MA_5Y	0.0026
18	SP_POP_0014_FE_ZS	0.0019
58	FD_AST_PRVT_GD_ZS	0.0016
31	SP_POP_1519_FE_5Y	0.0015
20	SP_DYN_T065_MA_ZS	0.0015
50	SP_DYN_TFRT_IN	0.0013
53	SP_POP_7579_FE_5Y	0.0012
35	SP_POP_65UP_FE_ZS	0.0012
51	SP_POP_4549_FE_5Y	0.0012
45	SP_POP_6569_FE_5Y	0.0011
32	SP_POP_5054_FE_5Y	0.0011
38	SP_POP_DPNP_DL	0.0010
87	TX_VAL_MANF_ZS_UN	0.0009
86	SP_RUR_TOTL_ZG	0.0009
56	TM_VAL_MRCH_HI_ZS	0.0009
78	NE_TRD_GNFS_ZS	0.0008
7	SP_DYN_LE00_FE_IN	0.0008
69	SP_POP_4044_MA_5Y	0.0008
64	SG_LAW_INDX	0.0008
54	SP_DYN_AMRT_MA	0.0008
47	SP_POP_6064_FE_5Y	0.0008
77	SI_POV_GINI	0.0008
76	FM_LBL_BMNY_GD_ZS	0.0007
19	IT_MLT_MAIN_P2	0.0007
52	SH_DYN_MORT_FE	0.0007
23	SP_DYN_CBRT_IN	0.0007
74	AG_YLD_CREL_KG	0.0007
66	SP_POP_4044_FE_5Y	0.0007
85	SE_SEC_ENRL_GC_FE_ZS	0.0007
55	NV_AGR_TOTL_ZS	0.0006
61	SP_POP_2024_MA_5Y	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
73	IT_CEL_SETS_P2	0.0006

39	SP_ADO_TFRT	0.0006
71	NE_EXP_GNFS_ZS	0.0006
72	SP_POP_1564_FE_ZS	0.0006
33	SP_POP_80UP_FE_5Y	0.0006
88	NY_ADJ_DKAP_GN_ZS	0.0006
83	TX_VAL_MRCH_HI_ZS	0.0006
24	SP_POP_6064_MA_5Y	0.0006
27	SP_POP_7579_MA_5Y	0.0006
81	NY_ADJ_DRES_GN_ZS	0.0006
34	SP_DYN_IMRT_MA_IN	0.0006
16	SP_POP_7074_MA_5Y	0.0005
36	NV_SRV_TOTL_ZS	0.0005
68	SP_POP_DPND	0.0005
65	NE_CON_PRVT_ZS	0.0005
14	SP_POP_6569_MA_5Y	0.0005
30	SP_POP_DPND_YG	0.0005
57	FM_AST_PRVT_GD_ZS	0.0005
17	SP_POP_5559_MA_5Y	0.0005
37	SP_DYN_IMRT_IN	0.0005
80	DT_NFL_UNDP_CD	0.0005
49	SH_DYN_MORT	0.0005
25	SP_POP_0004_FE_5Y	0.0005
48	SH_DYN_MORT_MA	0.0005
43	SP_DYN_T065_FE_ZS	0.0005
4	NY_GDP_PCAP_CD	0.0004
79	NY_GDP_TOTL_RT_ZS	0.0004
67	SP_POP_1564_MA_ZS	0.0004
84	NY_GDP_FRST_RT_ZS	0.0004
41	SP_DYN_IMRT_FE_IN	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
42	SP_POP_80UP_MA_5Y	0.0004
82	EN.URB_MCTY_TL_ZS	0.0004
60	SP_POP_2024_FE_5Y	0.0004
26	SP_POP_65UP_TO_ZS	0.0004
44	SP_POP_4549_MA_5Y	0.0004
75	SP_URB_GROW	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
21	SP_POP_0509_FE_5Y	0.0003
28	SH_DYN_NMRT	0.0003
11	SP_POP_1014_MA_5Y	0.0003
13	SP_POP_0004_MA_5Y	0.0003
15	SP_POP_5054_MA_5Y	0.0003
46	SP_POP_7074_FE_5Y	0.0003
40	SP_POP_5559_FE_5Y	0.0003
3	NY_GDP_PCAP_KD	0.0003
59	SP_DYN_AMRT_FE	0.0003
22	SP_POP_1014_FE_5Y	0.0002
1	NY_GNP_PCAP_CD	0.0002

6	SP_DYN_LE00_IN	0.0002
5	SP_DYN_LE00_MA_IN	0.0002
70	SP_POP_1564_TO_ZS	0.0002
10	SP_POP_0014_TO_ZS	0.0001
9	SP_POP_0509_MA_5Y	0.0001
63	SP_RUR_TOTL_ZS	0.0000

Model with 91 variables and max depth None:

Training+Validation R^2: 0.99974, RMSE: 0.31933

Testing R^2: 0.98207, RMSE: 2.70299

Mean cross-validation score: 0.9809

	Feature	Importance
0	WB_CC_EST_avg	0.9125
90	CC_EST_prev	0.0234
8	SP_POP_0014_MA_ZS	0.0032
29	SP_POP_1519_MA_5Y	0.0028
18	SP_POP_0014_FE_ZS	0.0022
20	SP_DYN_T065_MA_ZS	0.0018
51	SP_POP_4549_FE_5Y	0.0016
50	SP_DYN_TFRT_IN	0.0015
7	SP_DYN_LE00_FE_IN	0.0013
58	FD_AST_PRVT_GD_ZS	0.0013
53	SP_POP_7579_FE_5Y	0.0013
15	SP_POP_5054_MA_5Y	0.0011
25	SP_POP_0004_FE_5Y	0.0011
23	SP_DYN_CBRT_IN	0.0011
45	SP_POP_6569_FE_5Y	0.0011
32	SP_POP_5054_FE_5Y	0.0010
64	SG_LAW_INDX	0.0010
71	NE_EXP_GNFS_ZS	0.0010
27	SP_POP_7579_MA_5Y	0.0009
66	SP_POP_4044_FE_5Y	0.0009
33	SP_POP_80UP_FE_5Y	0.0009
31	SP_POP_1519_FE_5Y	0.0009
70	SP_POP_1564_TO_ZS	0.0009
87	TX_VAL_MANF_ZS_UN	0.0009
72	SP_POP_1564_FE_ZS	0.0009
16	SP_POP_7074_MA_5Y	0.0009
54	SP_DYN_AMRT_MA	0.0008
47	SP_POP_6064_FE_5Y	0.0008
86	SP_RUR_TOTL_ZG	0.0008
10	SP_POP_0014_TO_ZS	0.0008
39	SP_ADO_TFRT	0.0008
56	TM_VAL_MRCH_HI_ZS	0.0008
55	NV_AGR_TOTL_ZS	0.0008
76	FM_LBL_BMNY_GD_ZS	0.0008

62	SP_URB_TOTL_IN_ZS	0.0007
73	IT_CEL_SETS_P2	0.0007
74	AG_YLD_CREL_KG	0.0007
34	SP_DYN_IMRT_MA_IN	0.0007
85	SE_SEC_ENRL_GC_FE_ZS	0.0007
84	NY_GDP_FRST_RT_ZS	0.0007
82	EN_URB_MCTY_TL_ZS	0.0007
26	SP_POP_65UP_TO_ZS	0.0007
24	SP_POP_6064_MA_5Y	0.0007
78	NE_TRD_GNFS_ZS	0.0007
38	SP_POP_DPND_DL	0.0006
21	SP_POP_0509_FE_5Y	0.0006
61	SP_POP_2024_MA_5Y	0.0006
69	SP_POP_4044_MA_5Y	0.0006
57	FM_AST_PRVT_GD_ZS	0.0006
83	TX_VAL_MRCH_HI_ZS	0.0006
65	NE_CON_PRVT_ZS	0.0006
80	DTNFL_UNDP_CD	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
77	SI_POV_GINI	0.0006
19	IT_MLT_MAIN_P2	0.0005
79	NY_GDP_TOTL_RT_ZS	0.0005
13	SP_POP_0004_MA_5Y	0.0005
35	SP_POP_65UP_FE_ZS	0.0005
36	NV_SRV_TOTL_ZS	0.0005
60	SP_POP_2024_FE_5Y	0.0005
88	NY_ADJ_DKAP_GN_ZS	0.0005
68	SP_POP_DPND	0.0005
89	NE_CON_TOTL_ZS	0.0005
52	SH_DYN_MORT_FE	0.0005
81	NY_ADJ_DRES_GN_ZS	0.0004
75	SP_URB_GROW	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
40	SP_POP_5559_FE_5Y	0.0004
59	SP_DYN_AMRT_FE	0.0004
48	SH_DYN_MORT_MA	0.0004
3	NY_GDP_PCAP_KD	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
44	SP_POP_4549_MA_5Y	0.0004
43	SP_DYN_T065_FE_ZS	0.0004
42	SP_POP_80UP_MA_5Y	0.0004
22	SP_POP_1014_FE_5Y	0.0004
28	SH_DYN_NMRT	0.0003
11	SP_POP_1014_MA_5Y	0.0003
14	SP_POP_6569_MA_5Y	0.0003
46	SP_POP_7074_FE_5Y	0.0003
37	SP_DYN_IMRT_IN	0.0003
5	SP_DYN_LE00_MA_IN	0.0003

49	SH_DYN_MORT	0.0003
6	SP_DYN_LE00_IN	0.0003
63	SP_RUR_TOTL_ZS	0.0002
17	SP_POP_5559_MA_5Y	0.0002
1	NY_GNP_PCAP_CD	0.0002
4	NY_GDP_PCAP_CD	0.0002
67	SP_POP_1564_MA_ZS	0.0001
9	SP_POP_0509_MA_5Y	0.0001
30	SP_POP_DPND_YG	0.0001

Model with 92 variables and max depth None:

Training+Validation R^2: 0.99717, RMSE: 1.06067

Testing R^2: 0.98132, RMSE: 2.75886

Mean cross-validation score: 0.98081

	Feature	Importance
0	WB_CC_EST_avg	0.9028
91	CC_EST_prev	0.0236
18	SP_POP_0014_FE_ZS	0.0057
29	SP_POP_1519_MA_5Y	0.0027
8	SP_POP_0014_MA_ZS	0.0027
51	SP_POP_4549_FE_5Y	0.0021
20	SP_DYN_T065_MA_ZS	0.0019
58	FD_AST_PRVT_GD_ZS	0.0018
86	SP_RUR_TOTL_ZG	0.0014
50	SP_DYN_TFRT_IN	0.0013
45	SP_POP_6569_FE_5Y	0.0013
10	SP_POP_0014_TO_ZS	0.0013
23	SP_DYN_CBRT_IN	0.0012
71	NE_EXP_GNFS_ZS	0.0011
53	SP_POP_7579_FE_5Y	0.0011
66	SP_POP_4044_FE_5Y	0.0011
7	SP_DYN_LE00_FE_IN	0.0011
32	SP_POP_5054_FE_5Y	0.0010
52	SH_DYN_MORT_FE	0.0010
25	SP_POP_0004_FE_5Y	0.0010
24	SP_POP_6064_MA_5Y	0.0010
47	SP_POP_6064_FE_5Y	0.0010
15	SP_POP_5054_MA_5Y	0.0010
87	TX_VAL_MANF_ZS_UN	0.0010
64	SG_LAW_INDX	0.0010
77	SI_POV_GINI	0.0009
31	SP_POP_1519_FE_5Y	0.0009
21	SP_POP_0509_FE_5Y	0.0009
55	NV_AGR_TOTL_ZS	0.0009
26	SP_POP_65UP_TO_ZS	0.0009
61	SP_POP_2024_MA_5Y	0.0009

41	SP_DYN_IMRT_FE_IN	0.0009
69	SP_POP_4044_MA_5Y	0.0009
76	FM_LBL_BMNY_GD_ZS	0.0008
56	TM_VAL_MRCH_HI_ZS	0.0008
74	AG_YLD_CREL_KG	0.0008
65	NE_CON_PRVT_ZS	0.0008
78	NE_TRD_GNFS_ZS	0.0008
82	EN_URB_MCTY_TL_ZS	0.0008
72	SP_POP_1564_FE_ZS	0.0008
34	SP_DYN_IMRT_MA_IN	0.0008
16	SP_POP_7074_MA_5Y	0.0008
84	NY_GDP_FRST_RT_ZS	0.0008
88	NY_ADJ_DKAP_GN_ZS	0.0007
62	SP_URB_TOTL_IN_ZS	0.0007
85	SE_SEC_ENRL_GC_FE_ZS	0.0007
19	IT_MLT_MAIN_P2	0.0007
27	SP_POP_7579_MA_5Y	0.0007
54	SP_DYN_AMRT_MA	0.0007
83	TX_VAL_MRCH_HI_ZS	0.0007
39	SP_ADO_TFRT	0.0007
80	DT_NFL_UNDP_CD	0.0007
33	SP_POP_80UP_FE_5Y	0.0006
60	SP_POP_2024_FE_5Y	0.0006
89	NE_CON_TOTL_ZS	0.0006
68	SP_POP_DPNd	0.0006
35	SP_POP_65UP_FE_ZS	0.0006
44	SP_POP_4549_MA_5Y	0.0006
43	SP_DYN_TO65_FE_ZS	0.0006
73	IT_CEL_SETS_P2	0.0006
57	FM_AST_PRVT_GD_ZS	0.0006
40	SP_POP_5559_FE_5Y	0.0006
22	SP_POP_1014_FE_5Y	0.0006
59	SP_DYN_AMRT_FE	0.0005
79	NY_GDP_TOTL_RT_ZS	0.0005
14	SP_POP_6569_MA_5Y	0.0005
13	SP_POP_0004_MA_5Y	0.0005
12	SP_POP_65UP_MA_ZS	0.0005
75	SP_URB_GROW	0.0005
3	NY_GDP_PCAP_KD	0.0005
81	NY_ADJ_DRES_GN_ZS	0.0005
36	NV_SRV_TOTL_ZS	0.0005
90	NY_GDS_TOTL_ZS	0.0005
2	NY_GDP_PCAP_KD_rel	0.0004
46	SP_POP_7074_FE_5Y	0.0004
49	SH_DYN_MORT	0.0004
48	SH_DYN_MORT_MA	0.0004
38	SP_POP_DPNd_OL	0.0004
37	SP_DYN_IMRT_IN	0.0004

11	SP_POP_1014_MA_5Y	0.0004
67	SP_POP_1564_MA_ZS	0.0003
28	SH_DYN_NMRT	0.0003
5	SP_DYN_LE00_MA_IN	0.0003
4	NY_GDP_PCAP_CD	0.0002
63	SP_RUR_TOTL_ZS	0.0002
1	NY_GNP_PCAP_CD	0.0002
42	SP_POP_80UP_MA_5Y	0.0002
17	SP_POP_5559_MA_5Y	0.0002
9	SP_POP_0509_MA_5Y	0.0002
6	SP_DYN_LE00_IN	0.0002
30	SP_POP_DPND_YG	0.0001
70	SP_POP_1564_TO_ZS	0.0000

Model with 93 variables and max depth None:

Training+Validation R^2: 0.99968, RMSE: 0.35501

Testing R^2: 0.9822, RMSE: 2.69299

Mean cross-validation score: 0.98119

	Feature	Importance
0	WB_CC_EST_avg	0.9315
92	CC_EST_prev	0.0174
18	SP_POP_0014_FE_ZS	0.0037
8	SP_POP_0014_MA_ZS	0.0020
29	SP_POP_1519_MA_5Y	0.0020
72	SP_POP_1564_FE_ZS	0.0019
20	SP_DYN_TO65_MA_ZS	0.0016
64	SG_LAW_INDX	0.0014
34	SP_DYN_IMRT_MA_IN	0.0010
58	FD_AST_PRVT_GD_ZS	0.0010
53	SP_POP_7579_FE_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0009
31	SP_POP_1519_FE_5Y	0.0009
50	SP_DYN_TFRT_IN	0.0009
51	SP_POP_4549_FE_5Y	0.0009
23	SP_DYN_CBRT_IN	0.0008
91	NY_ADJ_AEDU_GN_ZS	0.0008
71	NE_EXP_GNFS_ZS	0.0008
82	EN_URB_MCTY_TL_ZS	0.0008
37	SP_DYN_IMRT_IN	0.0007
25	SP_POP_0004_FE_5Y	0.0006
74	AG_YLD_CREL_KG	0.0006
32	SP_POP_5054_FE_5Y	0.0006
66	SP_POP_4044_FE_5Y	0.0006
7	SP_DYN_LE00_FE_IN	0.0006
24	SP_POP_6064_MA_5Y	0.0006
54	SP_DYN_AMRT_MA	0.0006

85	SE_SEC_ENRL_GC_FE_ZS	0.0006
86	SP_RUR_TOTL_ZG	0.0006
87	TX_VAL_MANF_ZS_UN	0.0006
60	SP_POP_2024_FE_5Y	0.0005
69	SP_POP_4044_MA_5Y	0.0005
49	SH_DYN_MORT	0.0005
39	SP_ADO_TFRT	0.0005
67	SP_POP_1564_MA_ZS	0.0005
55	NV_AGR_TOTL_ZS	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0005
68	SP_POP_DPND	0.0005
40	SP_POP_5559_FE_5Y	0.0005
77	SI_POV_GINI	0.0005
81	NY_ADJ_DRES_GN_ZS	0.0005
90	NY_GDS_TOTL_ZS	0.0005
89	NE_CON_TOTL_ZS	0.0005
35	SP_POP_65UP_FE_ZS	0.0005
10	SP_POP_0014_TO_ZS	0.0005
78	NE_TRD_GNFS_ZS	0.0005
14	SP_POP_6569_MA_5Y	0.0005
80	DT_NFL_UNDP_CD	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
15	SP_POP_5054_MA_5Y	0.0005
84	NY_GDP_FRST_RT_ZS	0.0004
63	SP_RUR_TOTL_ZS	0.0004
88	NY_ADJ_DKAP_GN_ZS	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004
76	FM_LBL_BMNY_GD_ZS	0.0004
83	TX_VAL_MRCH_HI_ZS	0.0004
19	IT_MLT_MAIN_P2	0.0004
41	SP_DYN_IMRT_FE_IN	0.0004
65	NE_CON_PRVT_ZS	0.0004
73	IT_CEL_SETS_P2	0.0004
44	SP_POP_4549_MA_5Y	0.0004
36	NV_SRV_TOTL_ZS	0.0004
47	SP_POP_6064_FE_5Y	0.0004
79	NY_GDP_TOTL_RT_ZS	0.0003
75	SP_URB_GROW	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
62	SP_URB_TOTL_IN_ZS	0.0003
33	SP_POP_80UP_FE_5Y	0.0003
9	SP_POP_0509_MA_5Y	0.0003
13	SP_POP_0004_MA_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0003
30	SP_POP_DPND_YG	0.0003
28	SH_DYN_NMRT	0.0003
46	SP_POP_7074_FE_5Y	0.0003
3	NY_GDP_PCAP_KD	0.0003

52	SH_DYN_MORT_FE	0.0003
21	SP_POP_0509_FE_5Y	0.0002
5	SP_DYN_LE00_MA_IN	0.0002
6	SP_DYN_LE00_IN	0.0002
11	SP_POP_1014_MA_5Y	0.0002
12	SP_POP_65UP_MA_ZS	0.0002
43	SP_DYN_T065_FE_ZS	0.0002
1	NY_GNP_PCAP_CD	0.0002
61	SP_POP_2024_MA_5Y	0.0002
48	SH_DYN_MORT_MA	0.0002
38	SP_POP_DPND_DL	0.0002
42	SP_POP_80UP_MA_5Y	0.0002
17	SP_POP_5559_MA_5Y	0.0002
22	SP_POP_1014_FE_5Y	0.0001
70	SP_POP_1564_TO_ZS	0.0001
27	SP_POP_7579_MA_5Y	0.0001
59	SP_DYN_AMRT_FE	0.0001
4	NY_GDP_PCAP_CD	0.0001

Model with 94 variables and max depth None:

Training+Validation R^2: 0.99892, RMSE: 0.65616

Testing R^2: 0.98118, RMSE: 2.7693

Mean cross-validation score: 0.98092

	Feature	Importance
0	WB_CC_EST_avg	0.9339
93	CC_EST_prev	0.0192
8	SP_POP_0014_MA_ZS	0.0028
29	SP_POP_1519_MA_5Y	0.0021
18	SP_POP_0014_FE_ZS	0.0016
58	FD_AST_PRVT_GD_ZS	0.0013
20	SP_DYN_T065_MA_ZS	0.0012
50	SP_DYN_TFRT_IN	0.0012
72	SP_POP_1564_FE_ZS	0.0011
31	SP_POP_1519_FE_5Y	0.0010
64	SG_LAW_INDX	0.0009
53	SP_POP_7579_FE_5Y	0.0009
51	SP_POP_4549_FE_5Y	0.0009
91	NY_ADJ_AEDU_GN_ZS	0.0008
35	SP_POP_65UP_FE_ZS	0.0007
37	SP_DYN_IMRT_IN	0.0007
45	SP_POP_6569_FE_5Y	0.0007
78	NE_TRD_GNFS_ZS	0.0007
55	NV_AGR_TOTL_ZS	0.0007
54	SP_DYN_AMRT_MA	0.0007
66	SP_POP_4044_FE_5Y	0.0007
87	TX_VAL_MANF_ZS_UN	0.0007

52	SH_DYN_MORT_FE	0.0007
71	NE_EXP_GNFS_ZS	0.0006
7	SP_DYN_LE00_FE_IN	0.0006
17	SP_POP_5559_MA_5Y	0.0006
67	SP_POP_1564_MA_ZS	0.0006
23	SP_DYN_CBRT_IN	0.0006
49	SH_DYN_MORT	0.0006
85	SE_SEC_ENRL_GC_FE_ZS	0.0006
74	AG_YLD_CREL_KG	0.0005
76	FM_LBL_BMNY_GD_ZS	0.0005
60	SP_POP_2024_FE_5Y	0.0005
65	NE_CON_PRVT_ZS	0.0005
32	SP_POP_5054_FE_5Y	0.0005
62	SP_URB_TOTL_IN_ZS	0.0005
84	NY_GDP_FRST_RT_ZS	0.0005
89	NE_CON_TOTL_ZS	0.0005
86	SP_RUR_TOTL_ZG	0.0005
82	EN_URB_MCTY_TL_ZS	0.0005
16	SP_POP_7074_MA_5Y	0.0005
33	SP_POP_80UP_FE_5Y	0.0004
11	SP_POP_1014_MA_5Y	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004
56	TM_VAL_MRCH_HI_ZS	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
14	SP_POP_6569_MA_5Y	0.0004
15	SP_POP_5054_MA_5Y	0.0004
19	IT_MLT_MAIN_P2	0.0004
34	SP_DYN_IMRT_MA_IN	0.0004
24	SP_POP_6064_MA_5Y	0.0004
83	TX_VAL_MRCH_HI_ZS	0.0004
81	NY_ADJ_DRES_GN_ZS	0.0004
73	IT_CEL_SETS_P2	0.0004
42	SP_POP_80UP_MA_5Y	0.0004
80	DT_NFL_UNDP_CD	0.0004
77	SI_POV_GINI	0.0004
47	SP_POP_6064_FE_5Y	0.0004
69	SP_POP_4044_MA_5Y	0.0003
75	SP_URB_GROW	0.0003
79	NY_GDP_TOTL_RT_ZS	0.0003
88	NY_ADJ_DKAP_GN_ZS	0.0003
92	NE_IMP_GNFS_ZS	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
44	SP_POP_4549_MA_5Y	0.0003
26	SP_POP_65UP_TO_ZS	0.0003
61	SP_POP_2024_MA_5Y	0.0003
36	NV_SRV_TOTL_ZS	0.0003
25	SP_POP_0004_FE_5Y	0.0003
39	SP_ADO_TFRT	0.0003

40	SP_POP_5559_FE_5Y	0.0003
22	SP_POP_1014_FE_5Y	0.0003
28	SH_DYN_NMRT	0.0003
3	NY_GDP_PCAP_KD	0.0003
13	SP_POP_0004_MA_5Y	0.0002
10	SP_POP_0014_TO_ZS	0.0002
6	SP_DYN_LE00_IN	0.0002
90	NY_GDS_TOTL_ZS	0.0002
5	SP_DYN_LE00_MA_IN	0.0002
4	NY_GDP_PCAP_CD	0.0002
63	SP_RUR_TOTL_ZS	0.0002
30	SP_POP_DPND_YG	0.0002
27	SP_POP_7579_MA_5Y	0.0002
43	SP_DYN_T065_FE_ZS	0.0002
46	SP_POP_7074_FE_5Y	0.0002
70	SP_POP_1564_TO_ZS	0.0002
68	SP_POP_DPND	0.0002
48	SH_DYN_MORT_MA	0.0002
38	SP_POP_DPND_DL	0.0001
41	SP_DYN_IMRT_FE_IN	0.0001
21	SP_POP_0509_FE_5Y	0.0001
9	SP_POP_0509_MA_5Y	0.0001
1	NY_GNP_PCAP_CD	0.0001
59	SP_DYN_AMRT_FE	0.0001

Model with 95 variables and max depth None:

Training+Validation R^2: 0.99944, RMSE: 0.47081

Testing R^2: 0.98142, RMSE: 2.75138

Mean cross-validation score: 0.9809

	Feature	Importance
0	WB_CC_EST_avg	0.9189
94	CC_EST_prev	0.0199
18	SP_POP_0014_FE_ZS	0.0030
29	SP_POP_1519_MA_5Y	0.0026
8	SP_POP_0014_MA_ZS	0.0023
53	SP_POP_7579_FE_5Y	0.0018
58	FD_AST_PRVT_GD_ZS	0.0015
20	SP_DYN_T065_MA_ZS	0.0015
50	SP_DYN_TFRT_IN	0.0014
72	SP_POP_1564_FE_ZS	0.0013
31	SP_POP_1519_FE_5Y	0.0013
45	SP_POP_6569_FE_5Y	0.0012
91	NY_ADJ_AEDU_GN_ZS	0.0012
23	SP_DYN_CBRT_IN	0.0011
51	SP_POP_4549_FE_5Y	0.0011
32	SP_POP_5054_FE_5Y	0.0010

37	SP_DYN_IMRT_IN	0.0010
71	NE_EXP_GNFS_ZS	0.0010
64	SG_LAW_INDX	0.0009
24	SP_POP_6064_MA_5Y	0.0009
62	SP_URB_TOTL_IN_ZS	0.0009
87	TX_VAL_MANF_ZS_UN	0.0008
54	SP_DYN_AMRT_MA	0.0008
78	NE_TRD_GNFS_ZS	0.0008
89	NE_CON_TOTL_ZS	0.0008
60	SP_POP_2024_FE_5Y	0.0007
84	NY_GDP_FRST_RT_ZS	0.0007
66	SP_POP_4044_FE_5Y	0.0007
15	SP_POP_5054_MA_5Y	0.0007
11	SP_POP_1014_MA_5Y	0.0007
74	AG_YLD_CREL_KG	0.0007
7	SP_DYN_LE00_FE_IN	0.0007
77	SI_POV_GINI	0.0007
65	NE_CON_PRVT_ZS	0.0006
82	EN_URB_MCTY_TL_ZS	0.0006
70	SP_POP_1564_TO_ZS	0.0006
73	IT_CEL_SETS_P2	0.0006
55	NV_AGR_TOTL_ZS	0.0006
76	FM_LBL_BMNY_GD_ZS	0.0006
47	SP_POP_6064_FE_5Y	0.0006
41	SP_DYN_IMRT_FE_IN	0.0006
14	SP_POP_6569_MA_5Y	0.0006
86	SP_RUR_TOTL_ZG	0.0006
85	SE_SEC_ENRL_GC_FE_ZS	0.0006
81	NY_ADJ_DRES_GN_ZS	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0005
93	NE_EXP_GNFS_CD	0.0005
92	NE_IMP_GNFS_ZS	0.0005
16	SP_POP_7074_MA_5Y	0.0005
19	IT_MLT_MAIN_P2	0.0005
25	SP_POP_0004_FE_5Y	0.0005
42	SP_POP_80UP_MA_5Y	0.0005
88	NY_ADJ_DKAP_GN_ZS	0.0005
57	FM_AST_PRVT_GD_ZS	0.0005
80	DT_NFL_UNDP_CD	0.0005
46	SP_POP_7074_FE_5Y	0.0005
52	SH_DYN_MORT_FE	0.0005
43	SP_DYN_T065_FE_ZS	0.0005
38	SP_POP_DPND_OL	0.0005
83	TX_VAL_MRCH_HI_ZS	0.0005
3	NY_GDP_PCAP_KD	0.0005
44	SP_POP_4549_MA_5Y	0.0005
39	SP_ADO_TFRT	0.0005
34	SP_DYN_IMRT_MA_IN	0.0005

33	SP_POP_80UP_FE_5Y	0.0004
79	NY_GDP_TOTL_RT_ZS	0.0004
75	SP_URB_GROW	0.0004
90	NY_GDS_TOTL_ZS	0.0004
13	SP_POP_0004_MA_5Y	0.0004
40	SP_POP_5559_FE_5Y	0.0004
61	SP_POP_2024_MA_5Y	0.0004
67	SP_POP_1564_MA_ZS	0.0004
17	SP_POP_5559_MA_5Y	0.0004
63	SP_RUR_TOTL_ZS	0.0004
36	NV_SRV_TOTL_ZS	0.0004
69	SP_POP_4044_MA_5Y	0.0003
12	SP_POP_65UP_MA_ZS	0.0003
22	SP_POP_1014_FE_5Y	0.0003
6	SP_DYN_LE00_IN	0.0003
35	SP_POP_65UP_FE_ZS	0.0003
68	SP_POP_DPND	0.0003
2	NY_GDP_PCAP_KD_rel	0.0002
59	SP_DYN_AMRT_FE	0.0002
49	SH_DYN_MORT	0.0002
1	NY_GNP_PCAP_CD	0.0002
30	SP_POP_DPND_YG	0.0002
27	SP_POP_7579_MA_5Y	0.0002
26	SP_POP_65UP_TO_ZS	0.0002
5	SP_DYN_LE00_MA_IN	0.0002
4	NY_GDP_PCAP_CD	0.0002
48	SH_DYN_MORT_MA	0.0002
28	SH_DYN_NMRT	0.0001
10	SP_POP_0014_TO_ZS	0.0001
9	SP_POP_0509_MA_5Y	0.0001
21	SP_POP_0509_FE_5Y	0.0000

Model with 96 variables and max depth None:

Training+Validation R^2: 0.99888, RMSE: 0.66585

Testing R^2: 0.98173, RMSE: 2.72818

Mean cross-validation score: 0.98124

	Feature	Importance
0	WB_CC_EST_avg	0.9273
95	CC_EST_prev	0.0203
18	SP_POP_0014_FE_ZS	0.0043
29	SP_POP_1519_MA_5Y	0.0025
8	SP_POP_0014_MA_ZS	0.0025
20	SP_DYN_TO65_MA_ZS	0.0016
50	SP_DYN_TFRT_IN	0.0013
58	FD_AST_PRVT_GD_ZS	0.0013
53	SP_POP_7579_FE_5Y	0.0012

94	NY_ADJ_DFOR_GN_ZS	0.0012
72	SP_POP_1564_FE_ZS	0.0009
91	NY_ADJ_AEDU_GN_ZS	0.0009
64	SG_LAW_INDX	0.0009
7	SP_DYN_LE00_FE_IN	0.0008
47	SP_POP_6064_FE_5Y	0.0008
51	SP_POP_4549_FE_5Y	0.0008
45	SP_POP_6569_FE_5Y	0.0007
78	NE_TRD_GNFS_ZS	0.0007
38	SP_POP_DPND_OL	0.0007
37	SP_DYN_IMRT_IN	0.0007
31	SP_POP_1519_FE_5Y	0.0007
16	SP_POP_7074_MA_5Y	0.0007
71	NE_EXP_GNFS_ZS	0.0006
55	NV_AGR_TOTL_ZS	0.0006
33	SP_POP_80UP_FE_5Y	0.0006
30	SP_POP_DPND_YG	0.0006
25	SP_POP_0004_FE_5Y	0.0006
23	SP_DYN_CBRT_IN	0.0006
93	NE_EXP_GNFS_CD	0.0006
49	SH_DYN_MORT	0.0005
39	SP_ADO_TFRT	0.0005
13	SP_POP_0004_MA_5Y	0.0005
65	NE_CON_PRVT_ZS	0.0005
86	SP_RUR_TOTL_ZG	0.0005
69	SP_POP_4044_MA_5Y	0.0005
15	SP_POP_5054_MA_5Y	0.0005
10	SP_POP_0014_TO_ZS	0.0005
54	SP_DYN_AMRT_MA	0.0005
84	NY_GDP_FRST_RT_ZS	0.0005
87	TX_VAL_MANF_ZS_UN	0.0005
61	SP_POP_2024_MA_5Y	0.0005
35	SP_POP_65UP_FE_ZS	0.0005
77	SI_POV_GINI	0.0005
82	EN_URB_MCTY_TL_ZS	0.0005
32	SP_POP_5054_FE_5Y	0.0005
60	SP_POP_2024_FE_5Y	0.0005
24	SP_POP_6064_MA_5Y	0.0005
3	NY_GDP_PCAP_KD	0.0005
89	NE_CON_TOTL_ZS	0.0004
88	NY_ADJ_DKAP_GN_ZS	0.0004
80	DT_NFL_UNDP_CD	0.0004
66	SP_POP_4044_FE_5Y	0.0004
68	SP_POP_DPND	0.0004
79	NY_GDP_TOTL_RT_ZS	0.0004
73	IT_CEL_SETS_P2	0.0004
74	AG_YLD_CREL_KG	0.0004
83	TX_VAL_MRCH_HI_ZS	0.0004

76	FM_LBL_BMNY_GD_ZS	0.0004
85	SE_SEC_ENRL_GC_FE_ZS	0.0004
48	SH_DYN_MORT_MA	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004
56	TM_VAL_MRCH_HI_ZS	0.0004
36	NV_SRV_TOTL_ZS	0.0004
19	IT_MLT_MAIN_P2	0.0004
40	SP_POP_5559_FE_5Y	0.0004
44	SP_POP_4549_MA_5Y	0.0004
81	NY_ADJ_DRES_GN_ZS	0.0003
34	SP_DYN_IMRT_MA_IN	0.0003
41	SP_DYN_IMRT_FE_IN	0.0003
28	SH_DYN_NMRT	0.0003
46	SP_POP_7074_FE_5Y	0.0003
67	SP_POP_1564_MA_ZS	0.0003
11	SP_POP_1014_MA_5Y	0.0003
92	NE_IMP_GNFS_ZS	0.0003
27	SP_POP_7579_MA_5Y	0.0003
14	SP_POP_6569_MA_5Y	0.0002
17	SP_POP_5559_MA_5Y	0.0002
9	SP_POP_0509_MA_5Y	0.0002
5	SP_DYN_LE00_MA_IN	0.0002
59	SP_DYN_AMRT_FE	0.0002
26	SP_POP_65UP_TO_ZS	0.0002
75	SP_URB_GROW	0.0002
42	SP_POP_80UP_MA_5Y	0.0002
43	SP_DYN_TO65_FE_ZS	0.0002
63	SP_RUR_TOTL_ZS	0.0002
62	SP_URB_TOTL_IN_ZS	0.0002
2	NY_GDP_PCAP_KD_rel	0.0002
22	SP_POP_1014_FE_5Y	0.0001
21	SP_POP_0509_FE_5Y	0.0001
1	NY_GNP_PCAP_CD	0.0001
52	SH_DYN_MORT_FE	0.0001
90	NY_GDS_TOTL_ZS	0.0001
6	SP_DYN_LE00_IN	0.0001
4	NY_GDP_PCAP_CD	0.0001
70	SP_POP_1564_TO_ZS	0.0000

Model with 97 variables and max depth None:
 Training+Validation R^2: 0.99772, RMSE: 0.95198
 Testing R^2: 0.98036, RMSE: 2.82839
 Mean cross-validation score: 0.98114

	Feature	Importance
0	WB_CC_EST_avg	0.9151

96	CC_EST_prev	0.0199
18	SP_POP_0014_FE_ZS	0.0061
58	FD_AST_PRVT_GD_ZS	0.0029
8	SP_POP_0014_MA_ZS	0.0024
29	SP_POP_1519_MA_5Y	0.0024
20	SP_DYN_T065_MA_ZS	0.0020
72	SP_POP_1564_FE_ZS	0.0015
50	SP_DYN_TFRT_IN	0.0014
25	SP_POP_0004_FE_5Y	0.0014
94	NY_ADJ_DFOR_GN_ZS	0.0012
31	SP_POP_1519_FE_5Y	0.0012
52	SH_DYN_MORT_FE	0.0012
38	SP_POP_DPND_DL	0.0011
91	NY_ADJ_AEDU_GN_ZS	0.0011
51	SP_POP_4549_FE_5Y	0.0009
64	SG_LAW_INDX	0.0009
71	NE_EXP_GNFS_ZS	0.0009
35	SP_POP_65UP_FE_ZS	0.0008
86	SP_RUR_TOTL_ZG	0.0008
53	SP_POP_7579_FE_5Y	0.0008
48	SH_DYN_MORT_MA	0.0008
87	TX_VAL_MANF_ZS_UN	0.0007
66	SP_POP_4044_FE_5Y	0.0007
16	SP_POP_7074_MA_5Y	0.0007
55	NV_AGR_TOTL_ZS	0.0007
33	SP_POP_80UP_FE_5Y	0.0007
23	SP_DYN_CBRT_IN	0.0007
45	SP_POP_6569_FE_5Y	0.0007
81	NY_ADJ_DRES_GN_ZS	0.0007
82	EN_URB_MCTY_TL_ZS	0.0006
67	SP_POP_1564_MA_ZS	0.0006
65	NE_CON_PRVT_ZS	0.0006
7	SP_DYN_LE00_FE_IN	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
60	SP_POP_2024_FE_5Y	0.0006
56	TM_VAL_MRCH_HI_ZS	0.0006
89	NE_CON_TOTL_ZS	0.0006
78	NE_TRD_GNFS_ZS	0.0006
73	IT_CEL_SETS_P2	0.0006
36	NV_SRV_TOTL_ZS	0.0006
27	SP_POP_7579_MA_5Y	0.0006
34	SP_DYN_IMRT_MA_IN	0.0006
74	AG_YLD_CREL_KG	0.0006
32	SP_POP_5054_FE_5Y	0.0005
95	AG_CON_FERT_ZS	0.0005
93	NE_EXP_GNFS_CD	0.0005
69	SP_POP_4044_MA_5Y	0.0005
24	SP_POP_6064_MA_5Y	0.0005

19	IT_MLT_MAIN_P2	0.0005
26	SP_POP_65UP_TO_ZS	0.0005
76	FM_LBL_BMNY_GD_ZS	0.0005
85	SE_SEC_ENRL_GC_FE_ZS	0.0005
61	SP_POP_2024_MA_5Y	0.0005
77	SI_POV_GINI	0.0005
39	SP_ADO_TFRT	0.0005
54	SP_DYN_AMRT_MA	0.0005
83	TX_VAL_MRCH_HI_ZS	0.0005
80	DT_NFL_UNDP_CD	0.0005
13	SP_POP_0004_MA_5Y	0.0005
14	SP_POP_6569_MA_5Y	0.0005
3	NY_GDP_PCAP_KD	0.0005
44	SP_POP_4549_MA_5Y	0.0005
46	SP_POP_7074_FE_5Y	0.0004
79	NY_GDP_TOTL_RT_ZS	0.0004
68	SP_POP_DPND	0.0004
47	SP_POP_6064_FE_5Y	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004
15	SP_POP_5054_MA_5Y	0.0004
88	NY_ADJ_DKAP_GN_ZS	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
84	NY_GDP_FRST_RT_ZS	0.0004
75	SP_URB_GROW	0.0003
92	NE_IMP_GNFS_ZS	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
22	SP_POP_1014_FE_5Y	0.0003
17	SP_POP_5559_MA_5Y	0.0003
6	SP_DYN_LE00_IN	0.0003
43	SP_DYN_T065_FE_ZS	0.0003
42	SP_POP_80UP_MA_5Y	0.0003
41	SP_DYN_IMRT_FE_IN	0.0003
40	SP_POP_5559_FE_5Y	0.0003
10	SP_POP_0014_TO_ZS	0.0003
30	SP_POP_DPND_YG	0.0003
28	SH_DYN_NMRT	0.0003
11	SP_POP_1014_MA_5Y	0.0003
9	SP_POP_0509_MA_5Y	0.0002
70	SP_POP_1564_TO_ZS	0.0002
21	SP_POP_0509_FE_5Y	0.0002
37	SP_DYN_IMRT_IN	0.0002
1	NY_GNP_PCAP_CD	0.0002
59	SP_DYN_AMRT_FE	0.0002
49	SH_DYN_MORT	0.0002
63	SP_RUR_TOTL_ZS	0.0001
5	SP_DYN_LE00_MA_IN	0.0001
4	NY_GDP_PCAP_CD	0.0001
90	NY_GDS_TOTL_ZS	0.0000

Model with 98 variables and max depth None:
 Training+Validation R^2: 0.99977, RMSE: 0.2997
 Testing R^2: 0.9808, RMSE: 2.79678
 Mean cross-validation score: 0.98072

	Feature	Importance
0	WB_CC_EST_avg	0.8957
97	CC_EST_prev	0.0230
18	SP_POP_0014_FE_ZS	0.0055
8	SP_POP_0014_MA_ZS	0.0035
20	SP_DYN_T065_MA_ZS	0.0031
29	SP_POP_1519_MA_5Y	0.0030
72	SP_POP_1564_FE_ZS	0.0021
50	SP_DYN_TFRT_IN	0.0018
91	NY_ADJ_AEDU_GN_ZS	0.0018
53	SP_POP_7579_FE_5Y	0.0017
23	SP_DYN_CBRT_IN	0.0015
58	FD_AST_PRVT_GD_ZS	0.0015
96	DT_ODA_ALLD_CD	0.0014
51	SP_POP_4549_FE_5Y	0.0014
64	SG_LAW_INDX	0.0013
45	SP_POP_6569_FE_5Y	0.0013
31	SP_POP_1519_FE_5Y	0.0012
93	NE_EXP_GNFS_CD	0.0012
12	SP_POP_65UP_MA_ZS	0.0011
30	SP_POP_DPND_YG	0.0011
87	TX_VAL_MANF_ZS_UN	0.0011
61	SP_POP_2024_MA_5Y	0.0010
60	SP_POP_2024_FE_5Y	0.0010
94	NY_ADJ_DFOR_GN_ZS	0.0010
43	SP_DYN_T065_FE_ZS	0.0010
7	SP_DYN_LEO0_FE_IN	0.0010
86	SP_RUR_TOTL_ZG	0.0009
44	SP_POP_4549_MA_5Y	0.0009
33	SP_POP_80UP_FE_5Y	0.0009
19	IT_MLT_MAIN_P2	0.0009
80	DT_NFL_UNDP_CD	0.0009
76	FM_LBL_BMNY_GD_ZS	0.0009
84	NY_GDP_FRST_RT_ZS	0.0009
82	EN.URB_MCTY_TL_ZS	0.0008
47	SP_POP_6064_FE_5Y	0.0008
78	NE_TRD_GNFS_ZS	0.0008
56	TM_VAL_MRCH_HI_ZS	0.0008
57	FM_AST_PRVT_GD_ZS	0.0008
41	SP_DYN_IMRT_FE_IN	0.0008
81	NY_ADJ_DRES_GN_ZS	0.0008

42	SP_POP_80UP_MA_5Y	0.0008
39	SP_ADO_TFRT	0.0008
40	SP_POP_5559_FE_5Y	0.0008
3	NY_GDP_PCAP_KD	0.0007
68	SP_POP_DPNd	0.0007
92	NE_IMP_GNFS_ZS	0.0007
85	SE_SEC_ENRL_GC_FE_ZS	0.0007
24	SP_POP_6064_MA_5Y	0.0007
89	NE_CON_TOTL_ZS	0.0007
88	NY_ADJ_DKAP_GN_ZS	0.0007
77	SI_POV_GINI	0.0007
83	TX_VAL_MRCH_HI_ZS	0.0007
32	SP_POP_5054_FE_5Y	0.0007
62	SP_URB_TOTL_IN_ZS	0.0006
74	AG_YLD_CREL_KG	0.0006
73	IT_CEL_SETS_P2	0.0006
71	NE_EXP_GNFS_ZS	0.0006
79	NY_GDP_TOTL_RT_ZS	0.0006
66	SP_POP_4044_FE_5Y	0.0006
65	NE_CON_PRVT_ZS	0.0006
35	SP_POP_65UP_FE_ZS	0.0006
48	SH_DYN_MORT_MA	0.0006
14	SP_POP_6569_MA_5Y	0.0006
34	SP_DYN_IMRT_MA_IN	0.0006
55	NV_AGR_TOTL_ZS	0.0006
54	SP_DYN_AMRT_MA	0.0006
16	SP_POP_7074_MA_5Y	0.0005
17	SP_POP_5559_MA_5Y	0.0005
21	SP_POP_0509_FE_5Y	0.0005
11	SP_POP_1014_MA_5Y	0.0005
36	NV_SRV_TOTL_ZS	0.0005
22	SP_POP_1014_FE_5Y	0.0005
95	AG_CON_FERT_ZS	0.0005
5	SP_DYN_LEO0_MA_IN	0.0005
25	SP_POP_0004_FE_5Y	0.0005
59	SP_DYN_AMRT_FE	0.0005
2	NY_GDP_PCAP_KD_rel	0.0004
75	SP_URB_GROW	0.0004
46	SP_POP_7074_FE_5Y	0.0004
4	NY_GDP_PCAP_CD	0.0004
15	SP_POP_5054_MA_5Y	0.0004
27	SP_POP_7579_MA_5Y	0.0004
28	SH_DYN_NMRT	0.0004
67	SP_POP_1564_MA_ZS	0.0004
69	SP_POP_4044_MA_5Y	0.0004
1	NY_GNP_PCAP_CD	0.0003
52	SH_DYN_MORT_FE	0.0003
38	SP_POP_DPNd_DL	0.0003

90	NY_GDS_TOTL_ZS	0.0003
10	SP_POP_0014_TO_ZS	0.0003
13	SP_POP_0004_MA_5Y	0.0002
9	SP_POP_0509_MA_5Y	0.0002
6	SP_DYN_LEOO_IN	0.0002
37	SP_DYN_IMRT_IN	0.0001
49	SH_DYN_MORT	0.0001
70	SP_POP_1564_TO_ZS	0.0000
63	SP_RUR_TOTL_ZS	0.0000
26	SP_POP_65UP_TO_ZS	0.0000

Model with 99 variables and max depth None:

Training+Validation R^2: 0.99911, RMSE: 0.59498

Testing R^2: 0.98024, RMSE: 2.83739

Mean cross-validation score: 0.98101

	Feature	Importance
0	WB_CC_EST_avg	0.9146
98	CC_EST_prev	0.0217
18	SP_POP_0014_FE_ZS	0.0064
8	SP_POP_0014_MA_ZS	0.0035
29	SP_POP_1519_MA_5Y	0.0026
58	FD_AST_PRVT_GD_ZS	0.0022
50	SP_DYN_TFRT_IN	0.0018
20	SP_DYN_T065_MA_ZS	0.0016
97	DT_ODA_ALLD_KD	0.0015
72	SP_POP_1564_FE_ZS	0.0013
38	SP_POP_DPND_OL	0.0013
91	NY_ADJ_AEDU_GN_ZS	0.0009
31	SP_POP_1519_FE_5Y	0.0009
63	SP_RUR_TOTL_ZS	0.0009
52	SH_DYN_MORT_FE	0.0008
53	SP_POP_7579_FE_5Y	0.0008
94	NY_ADJ_DFOR_GN_ZS	0.0008
23	SP_DYN_CBRT_IN	0.0007
51	SP_POP_4549_FE_5Y	0.0007
86	SP_RUR_TOTL_ZG	0.0007
47	SP_POP_6064_FE_5Y	0.0007
32	SP_POP_5054_FE_5Y	0.0007
45	SP_POP_6569_FE_5Y	0.0007
7	SP_DYN_LEOO_FE_IN	0.0007
93	NE_EXP_GNFS_CD	0.0007
87	TX_VAL_MANF_ZS_UN	0.0006
64	SG_LAW_INDX	0.0006
96	DT_ODA_ALLD_CD	0.0006
76	FM_LBL_BMNY_GD_ZS	0.0006
74	AG_YLD_CREL_KG	0.0006

3	NY_GDP_PCAP_KD	0.0006
40	SP_POP_5559_FE_5Y	0.0006
55	NV_AGR_TOTL_ZS	0.0006
14	SP_POP_6569_MA_5Y	0.0006
84	NY_GDP_FRST_RT_ZS	0.0006
16	SP_POP_7074_MA_5Y	0.0006
62	SP_URB_TOTL_IN_ZS	0.0006
89	NE_CON_TOTL_ZS	0.0006
71	NE_EXP_GNFS_ZS	0.0006
60	SP_POP_2024_FE_5Y	0.0006
56	TM_VAL_MRCH_HI_ZS	0.0005
77	SI_POV_GINI	0.0005
54	SP_DYN_AMRT_MA	0.0005
65	NE_CON_PRVT_ZS	0.0005
66	SP_POP_4044_FE_5Y	0.0005
73	IT_CEL_SETS_P2	0.0005
35	SP_POP_65UP_FE_ZS	0.0005
44	SP_POP_4549_MA_5Y	0.0005
78	NE_TRD_GNFS_ZS	0.0005
95	AG_CON_FERT_ZS	0.0005
10	SP_POP_0014_TO_ZS	0.0005
92	NE_IMP_GNFS_ZS	0.0005
15	SP_POP_5054_MA_5Y	0.0005
90	NY_GDS_TOTL_ZS	0.0005
24	SP_POP_6064_MA_5Y	0.0005
25	SP_POP_0004_FE_5Y	0.0005
27	SP_POP_7579_MA_5Y	0.0005
85	SE_SEC_ENRL_GC_FE_ZS	0.0005
33	SP_POP_80UP_FE_5Y	0.0005
68	SP_POP_DPNP	0.0005
82	EN_URB_MCTY_TL_ZS	0.0005
81	NY_ADJ_DRES_GN_ZS	0.0005
80	DT_NFL_UNDP_CD	0.0005
41	SP_DYN_IMRT_FE_IN	0.0005
69	SP_POP_4044_MA_5Y	0.0005
83	TX_VAL_MRCH_HI_ZS	0.0004
88	NY_ADJ_DKAP_GN_ZS	0.0004
79	NY_GDP_TOTL_RT_ZS	0.0004
2	NY_GDP_PCAP_KD_rel	0.0004
19	IT_MLT_MAIN_P2	0.0004
11	SP_POP_1014_MA_5Y	0.0004
61	SP_POP_2024_MA_5Y	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004
34	SP_DYN_IMRT_MA_IN	0.0004
36	NV_SRV_TOTL_ZS	0.0004
39	SP_ADO_TFRT	0.0004
13	SP_POP_0004_MA_5Y	0.0003
22	SP_POP_1014_FE_5Y	0.0003

26	SP_POP_65UP_TO_ZS	0.0003
28	SH_DYN_NMRT	0.0003
12	SP_POP_65UP_MA_ZS	0.0003
42	SP_POP_80UP_MA_5Y	0.0003
17	SP_POP_5559_MA_5Y	0.0003
43	SP_DYN_T065_FE_ZS	0.0003
75	SP.URB.GROW	0.0003
59	SP_DYN_AMRT_FE	0.0003
9	SP_POP_0509_MA_5Y	0.0002
5	SP_DYN_LE00_MA_IN	0.0002
4	NY_GDP_PCAP_CD	0.0002
67	SP_POP_1564_MA_ZS	0.0002
30	SP_POP_DPND_YG	0.0002
70	SP_POP_1564_TO_ZS	0.0002
46	SP_POP_7074_FE_5Y	0.0002
1	NY_GNP_PCAP_CD	0.0002
21	SP_POP_0509_FE_5Y	0.0001
37	SP_DYN_IMRT_IN	0.0001
6	SP_DYN_LE00_IN	0.0001
48	SH_DYN_MORT_MA	0.0001
49	SH_DYN_MORT	0.0000

Model with 100 variables and max depth None:

Training+Validation R^2: 0.99964, RMSE: 0.38039

Testing R^2: 0.9809, RMSE: 2.78979

Mean cross-validation score: 0.98105

	Feature	Importance
0	WB_CC_EST_avg	0.9168
99	CC_EST_prev	0.0209
8	SP_POP_0014_MA_ZS	0.0028
29	SP_POP_1519_MA_5Y	0.0027
72	SP_POP_1564_FE_ZS	0.0018
18	SP_POP_0014_FE_ZS	0.0017
20	SP_DYN_T065_MA_ZS	0.0017
97	DT_ODA_ALLD_KD	0.0015
50	SP_DYN_TFRT_IN	0.0014
64	SG_LAW_INDX	0.0014
38	SP_POP_DPND_OL	0.0014
94	NY_ADJ_DFOR_GN_ZS	0.0013
91	NY_ADJ_AEDU_GN_ZS	0.0013
58	FD_AST_PRVT_GD_ZS	0.0013
51	SP_POP_4549_FE_5Y	0.0013
53	SP_POP_7579_FE_5Y	0.0012
31	SP_POP_1519_FE_5Y	0.0011
7	SP_DYN_LE00_FE_IN	0.0010
27	SP_POP_7579_MA_5Y	0.0010

42	SP_POP_80UP_MA_5Y	0.0008
45	SP_POP_6569_FE_5Y	0.0008
25	SP_POP_0004_FE_5Y	0.0008
21	SP_POP_0509_FE_5Y	0.0007
47	SP_POP_6064_FE_5Y	0.0007
49	SH_DYN_MORT	0.0007
66	SP_POP_4044_FE_5Y	0.0007
85	SE_SEC_ENRL_GC_FE_ZS	0.0007
52	SH_DYN_MORT_FE	0.0007
54	SP_DYN_AMRT_MA	0.0006
86	SP_RUR_TOTL_ZG	0.0006
80	DT_NFL_UNDP_CD	0.0006
82	EN_URB_MCTY_TL_ZS	0.0006
78	NE_TRD_GNFS_ZS	0.0006
84	NY_GDP_FRST_RT_ZS	0.0006
77	SI_POV_GINI	0.0006
40	SP_POP_5559_FE_5Y	0.0006
35	SP_POP_65UP_FE_ZS	0.0006
33	SP_POP_80UP_FE_5Y	0.0006
17	SP_POP_5559_MA_5Y	0.0006
96	DT_ODA_ALLD_CD	0.0006
23	SP_DYN_CBRT_IN	0.0006
93	NE_EXP_GNFS_CD	0.0006
34	SP_DYN_IMRT_MA_IN	0.0006
74	AG_YLD_CREL_KG	0.0006
87	TX_VAL_MANF_ZS_UN	0.0006
32	SP_POP_5054_FE_5Y	0.0006
55	NV_AGR_TOTL_ZS	0.0005
61	SP_POP_2024_MA_5Y	0.0005
56	TM_VAL_MRCH_HI_ZS	0.0005
71	NE_EXP_GNFS_ZS	0.0005
69	SP_POP_4044_MA_5Y	0.0005
76	FM_LBL_BMNY_GD_ZS	0.0005
73	IT_CEL_SETS_P2	0.0005
15	SP_POP_5054_MA_5Y	0.0005
39	SP_ADO_TFR	0.0005
11	SP_POP_1014_MA_5Y	0.0005
89	NE_CON_TOTL_ZS	0.0005
24	SP_POP_6064_MA_5Y	0.0005
4	NY_GDP_PCAP_CD	0.0005
81	NY_ADJ_DRES_GN_ZS	0.0004
83	TX_VAL_MRCH_HI_ZS	0.0004
75	SP_URB_GROW	0.0004
88	NY_ADJ_DKAP_GN_ZS	0.0004
95	AG_CON_FERT_ZS	0.0004
98	IS_AIR_GOOD_MT_K1	0.0004
68	SP_POP_DPN	0.0004
57	FM_AST_PRVT_GD_ZS	0.0004

65	NE_CON_PRVT_ZS	0.0004
12	SP_POP_65UP_MA_ZS	0.0004
14	SP_POP_6569_MA_5Y	0.0004
19	IT_MLT_MAIN_P2	0.0004
44	SP_POP_4549_MA_5Y	0.0004
62	SP_URB_TOTL_IN_ZS	0.0004
60	SP_POP_2024_FE_5Y	0.0004
36	NV_SRV_TOTL_ZS	0.0004
41	SP_DYN_IMRT_FE_IN	0.0004
48	SH_DYN_MORT_MA	0.0004
3	NY_GDP_PCAP_KD	0.0004
67	SP_POP_1564_MA_ZS	0.0003
28	SH_DYN_NMRT	0.0003
92	NE_IMP_GNFS_ZS	0.0003
22	SP_POP_1014_FE_5Y	0.0003
13	SP_POP_0004_MA_5Y	0.0003
26	SP_POP_65UP_TO_ZS	0.0003
43	SP_DYN_T065_FE_ZS	0.0003
46	SP_POP_7074_FE_5Y	0.0003
79	NY_GDP_TOTL_RT_ZS	0.0003
2	NY_GDP_PCAP_KD_rel	0.0003
37	SP_DYN_IMRT_IN	0.0002
1	NY_GNP_PCAP_CD	0.0002
90	NY_GDS_TOTL_ZS	0.0002
16	SP_POP_7074_MA_5Y	0.0002
10	SP_POP_0014_TO_ZS	0.0002
5	SP_DYN_LE00_MA_IN	0.0002
59	SP_DYN_AMRT_FE	0.0002
63	SP_RUR_TOTL_ZS	0.0002
9	SP_POP_0509_MA_5Y	0.0002
30	SP_POP_DPND_YG	0.0001
6	SP_DYN_LE00_IN	0.0001
70	SP_POP_1564_TO_ZS	0.0000

```
[221]: # Plotting
fig, ax = plt.subplots(figsize=(20, 15))

# Plot the R^2 scores for the training set
ax.plot(max_depth_values, train_r2_scores, marker='o', color='#00688B', □
         label='Train R^2')

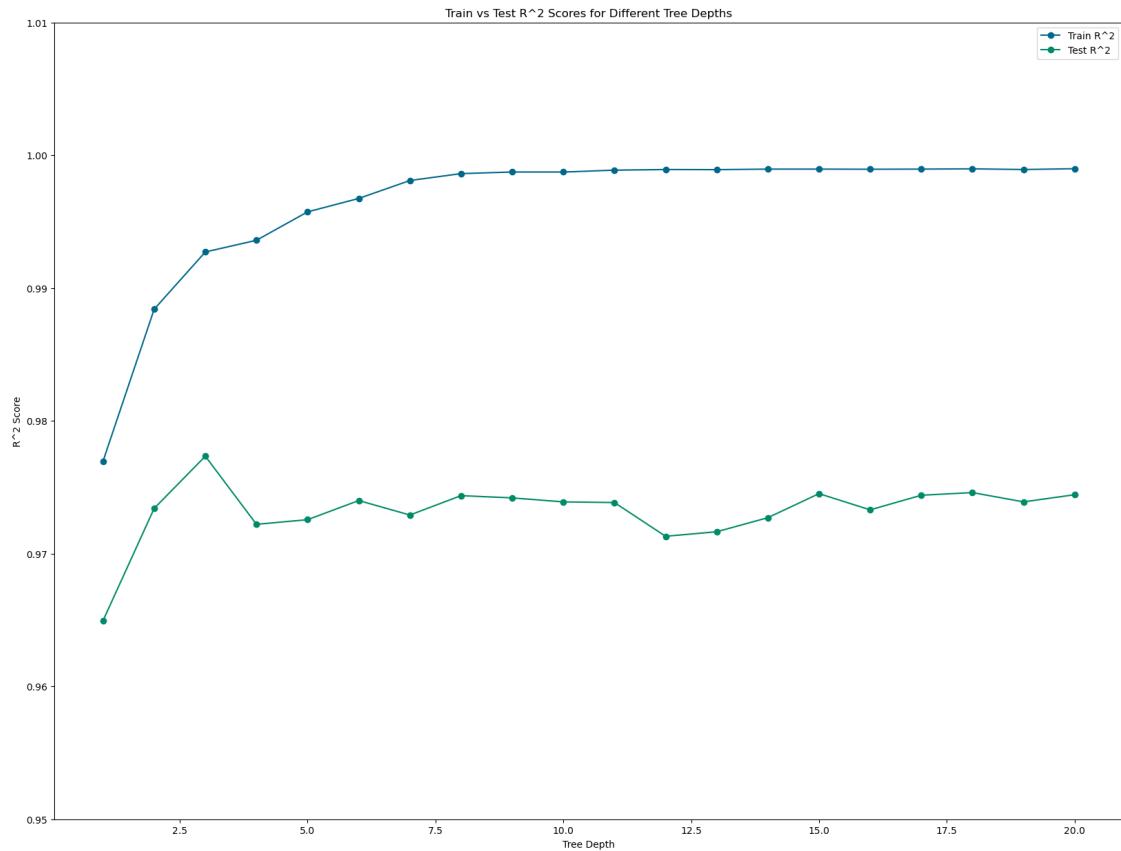
# Plot the R^2 scores for the test set
ax.plot(max_depth_values, test_r2_scores, marker='o', color='#008B68', □
         label='Test R^2')
```

```

ax.set_xlabel('Tree Depth')
ax.set_ylabel('R^2 Score')
ax.set_title('Train vs Test R^2 Scores for Different Tree Depths')
ax.set_ylim(0.95,1.01)
ax.legend()

plt.show()

```



```

[222]: # Plot the difference between the training and test R^2 scores
fig, ax = plt.subplots(figsize=(20, 10))

# Calculate the difference between the training and test R^2 scores
r2_diff = np.array(train_r2_scores) - np.array(test_r2_scores)

# Plot the difference between the training and test R^2 scores
ax.plot(max_depth_values, r2_diff, marker='o', color="#00688B")

ax.set_xlabel('Tree Depth')
ax.set_ylabel('R^2 Difference')

```

```

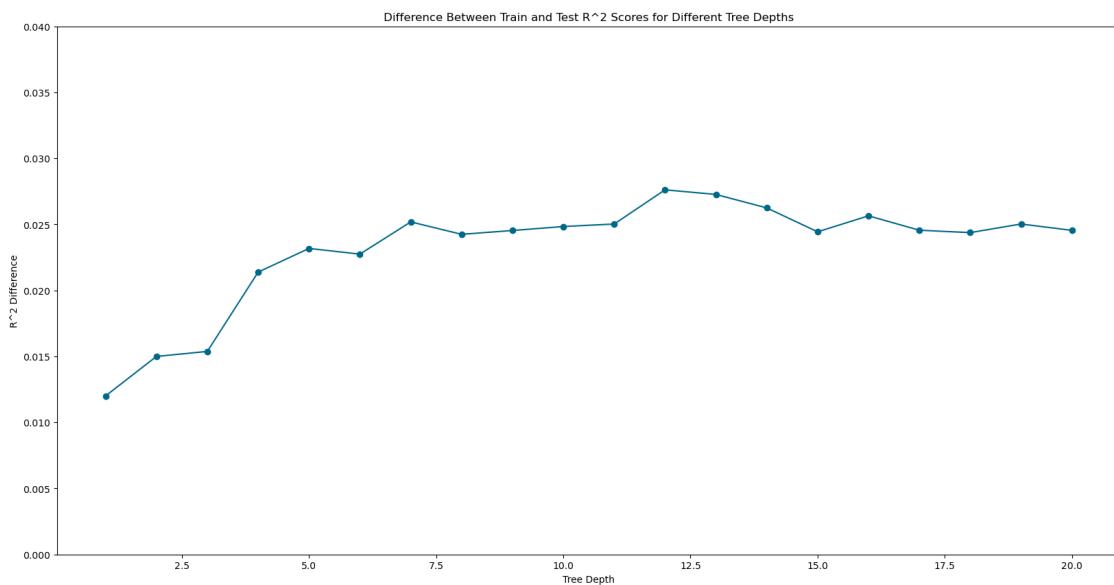
ax.set_title('Difference Between Train and Test R^2 Scores for Different Tree Depths')
ax.set_ylim(0,0.04)

plt.show()

# Print the maximum R^2 score for the test set and the corresponding tree depth
max_r2_test = max(test_r2_scores)
max_r2_test_depth = max_depth_values[test_r2_scores.index(max_r2_test)]

print(f"The maximum R^2 score for the test set is {max_r2_test:.5f} at a tree depth of {max_r2_test_depth}")

```



The maximum R^2 score for the test set is 0.97734 at a tree depth of 3

6 Model 4

```

[223]: df = pd.read_csv('WDI_data.csv')
df['CC_EST'] = ((df['CC_EST'] * (-1)) + 2.5) * 20
# Calculate the correlation between column CC_EST and column CPI_EST
correlation = df['CC_EST'].corr(df['CPI_EST'])
correlation

```

[223]: 0.9888007719282952

```

[224]: df = pd.read_csv('WDI_data.csv')

```

```
# Remove CC_EST
df = df.drop(columns=['CC_EST'])
# Make years integers
df['year'] = df['year'].astype(int)
```

[225]: # Create a column that has the average CPI_EST for each country between the ↵ year 2012 and 2022 only but prints for all years given the country
df['CPI_EST_avg'] = df.groupby('iso2c')['CPI_EST'].transform(lambda x: ↵ x[(df['year'] >= 2012) & (df['year'] <= 2022)].mean())
df['CPI_EST_avg'] = np.log1p(df['CPI_EST_avg'])
Drop all rows with missing values in 'CPI_EST_avg'
df = df.dropna(subset=['CPI_EST_avg'])

[226]: # Filter df to only include years from 1970 to 2011
df_pre_2012 = df[(df['year'] >= 1970) & (df['year'] < 2012)]

Calculate the percentage of missing values in each column for years prior to ↵ 2012
missing_percentages = df_pre_2012.isnull().mean()

Select the columns that have less than 50% missing values, or are 'CC_EST', ↵ 'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', and 'WB_CC_EST_avg'
cols_to_keep = [col for col in df.columns if col in ↵ ['CPI_EST', 'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', 'CPI_EST_avg'] or (col in ↵ missing_percentages and missing_percentages[col] < 0.5)]

Keep only the selected columns in df
df = df[cols_to_keep]

[227]: # View
df[['iso2c', 'country', 'year', 'CPI_EST', 'CPI_EST_avg']]

	iso2c	country	year	CPI_EST	CPI_EST_avg
63	AE	United Arab Emirates	1960	NaN	3.457177
64	AE	United Arab Emirates	1961	NaN	3.457177
65	AE	United Arab Emirates	1962	NaN	3.457177
66	AE	United Arab Emirates	1963	NaN	3.457177
67	AE	United Arab Emirates	1964	NaN	3.457177
...
13729	ZW	Zimbabwe	2018	78.0	4.368296
13730	ZW	Zimbabwe	2019	76.0	4.368296
13731	ZW	Zimbabwe	2020	76.0	4.368296
13732	ZW	Zimbabwe	2021	77.0	4.368296
13733	ZW	Zimbabwe	2022	77.0	4.368296

[11277 rows x 5 columns]

```
[228]: # Create a dictionary that maps ISO 2-letter country codes to a dictionary containing the ISO 3-letter and numeric country codes
country_codes = {country.alpha_2: {'alpha-3': country.alpha_3, 'numeric': country.numeric} for country in pycountry.countries}

# Now you can use this dictionary in your code
df['iso3c'] = df['iso2c'].map(lambda x: country_codes.get(x, {}).get('alpha-3'))
df['iso3n'] = df['iso2c'].map(lambda x: country_codes.get(x, {}).get('numeric'))
```

```
[229]: def backfill_based_on_avg_change(df, column):
    """
    Backfill missing values in the specified column based on the average percentage change of the next 10 years for each country.
    """

    # Iterate over each country
    for country in df['iso2c'].unique():
        # Get the data for the current country
        df_country = df[df['iso2c'] == country].copy()

        # Sort the data by year in ascending order
        df_country.sort_values('year', inplace=True)

        # Identify the indices of the missing values in the column
        missing_indices = df_country[df_country[column].isnull() & (df_country['year'] < 2012)].index.tolist()

        # For each missing value, calculate the average percentage change of the next 10 years and use it to fill the missing value
        for idx in missing_indices:
            if df_country.loc[idx, 'year'] < 1950:
                continue
            next_10_years_avg_pct_change = df_country.loc[idx+1:idx+11, column].pct_change().mean()

            # Apply a resistance factor to the average percentage change
            resistance_factor = 1 / (abs(next_10_years_avg_pct_change) + 1)
            if abs(next_10_years_avg_pct_change) > 0.05:
                resistance_factor *= 1 / (10 * abs(next_10_years_avg_pct_change) + 1)
            next_10_years_avg_pct_change *= resistance_factor

            # Apply the average percentage change to each year as it extrapolates backwards
            for i in reversed(range(missing_indices[0], idx+1)):
                if i+1 in df_country.index:
```

```

        df_country.loc[i, column] = df_country.loc[i+1, column] * ↵
        ↵(1 - next_10_years_avg_pct_change)

    # Update the column in the original DataFrame
    df.loc[df['iso2c'] == country, column] = df_country[column]

    return df

# Use the function to backfill the missing values in 'NY_GDP_PCAP_KD' and ↵
    ↵'NY_GDP_PCAP_CD'
df = backfill_based_on_avg_change(df, 'NY_GDP_PCAP_KD')
df = backfill_based_on_avg_change(df, 'NY_GDP_PCAP_CD')
df = backfill_based_on_avg_change(df, 'NY_GDP_PCAP_CN')
df['NY_GDP_PCAP_KD_rel'] = df.groupby('year')['NY_GDP_PCAP_KD']. ↵
    ↵transform(lambda x: x / x.mean())
df['NY_GDP_PCAP_CD_rel'] = df.groupby('year')['NY_GDP_PCAP_CD']. ↵
    ↵transform(lambda x: x / x.mean())
df['NY_GDP_PCAP_CN_rel'] = df.groupby('year')['NY_GDP_PCAP_CN']. ↵
    ↵transform(lambda x: x / x.mean())

```

[230]: # View
df[['iso2c', 'country', 'year', 'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', ↵
 ↵'NY_GDP_PCAP_CD_rel']]

	iso2c	country	year	NY_GDP_PCAP_KD	NY_GDP_PCAP_CD	NY_GDP_PCAP_CD_rel
63	AE	United Arab Emirates	1960	147090.258253	20783.760079	12.692847
64	AE	United Arab Emirates	1961	143884.292488	21154.209911	12.010375
65	AE	United Arab Emirates	1962	140748.203658	21531.262642	11.495933
66	AE	United Arab Emirates	1963	137680.468732	21915.035962	10.994984
67	AE	United Arab Emirates	1964	134679.597875	22305.649660	10.029993
...
13729	ZW	Zimbabwe	2018	1462.590279	2269.177012	0.001235
13730	ZW	Zimbabwe	2019	1342.989586	1421.868596	0.003255
13731	ZW	Zimbabwe	2020	1213.117057	1372.696674	0.017137
13732	ZW	Zimbabwe	2021	1289.199067	1773.920411	
13733	ZW	Zimbabwe	2022	1345.769082	1676.821489	

```
13732          0.027691
13733          0.076770
```

[11277 rows x 6 columns]

```
[231]: df_dr = df[df['year'] >= 2012]
df_dr = df_dr.dropna(subset=['CPI_EST'])

# Create a new dataframe with only iso2c, country, and CC_EST
df_corr = df_dr[['iso2c', 'country', 'year', 'CPI_EST']]

# Filter the data for the years 2012 and 2022
df_filtered = df_corr[df_corr['year'].isin([2012, 2022])]
```



```
[232]: # Calculate the linear differences over time for each given country
change_by_country = df_filtered.pivot_table(index='iso2c', columns='year',  

    ↪values='CPI_EST', aggfunc='first')

# Calculate the change for each country
change_by_country['avg_change'] = (change_by_country[2022] -  

    ↪change_by_country[2012]) / 10

# Calculate the mean and median CC_EST values across all years for each country
mean_cc_est_by_country = df_filtered.groupby('iso2c')['CPI_EST'].mean()
median_cc_est_by_country = df_filtered.groupby('iso2c')['CPI_EST'].median()

# Merge with the original DataFrame to get the country names
df_trend = pd.merge(change_by_country['avg_change'], mean_cc_est_by_country,  

    ↪left_index=True, right_index=True, how='left')
df_trend = pd.merge(df_trend, median_cc_est_by_country, left_index=True,  

    ↪right_index=True, how='left')

# Reset index and rename columns
df_trend.reset_index(inplace=True)
df_trend.rename(columns={'CPI_EST_x': 'mean_CPI_EST', 'PIC_EST_y':  

    ↪'median_CPI_EST'}, inplace=True)

# Include country names
df_trend = pd.merge(df_trend, df_filtered[['iso2c', 'country']].  

    ↪drop_duplicates(), on='iso2c', how='left')

print(df_trend)
```

	iso2c	avg_change	mean_CPI_EST	CPI_EST_y	country
0	AE	0.1	32.5	32.5	United Arab Emirates
1	AF	-1.6	84.0	84.0	Afghanistan
2	AL	-0.3	65.5	65.5	Albania

3	AM	-1.2	60.0	60.0	Armenia
4	AO	-1.1	72.5	72.5	Angola
5	AR	-0.3	63.5	63.5	Argentina
6	AT	-0.2	30.0	30.0	Austria
7	AU	1.0	20.0	20.0	Australia
8	AZ	0.4	75.0	75.0	Azerbaijan
9	BA	0.8	62.0	62.0	Bosnia and Herzegovina
10	BB	1.1	29.5	29.5	Barbados
11	BD	0.1	74.5	74.5	Bangladesh
12	BE	0.2	26.0	26.0	Belgium
13	BF	-0.4	60.0	60.0	Burkina Faso
14	BG	-0.2	58.0	58.0	Bulgaria
15	BH	0.7	52.5	52.5	Bahrain
16	BI	0.2	82.0	82.0	Burundi
17	BJ	-0.7	60.5	60.5	Benin
18	BN	NaN	45.0	45.0	Brunei Darussalam
19	BO	0.3	67.5	67.5	Bolivia
20	BR	0.5	59.5	59.5	Brazil
21	BS	0.7	32.5	32.5	Bahamas, The
22	BT	-0.5	34.5	34.5	Bhutan
23	BW	0.5	37.5	37.5	Botswana
24	BY	-0.8	65.0	65.0	Belarus
25	CA	1.0	21.0	21.0	Canada
26	CD	0.1	79.5	79.5	Congo, Dem. Rep.
27	CF	0.2	75.0	75.0	Central African Republic
28	CG	0.5	76.5	76.5	Congo, Rep.
29	CH	0.4	16.0	16.0	Switzerland
30	CI	-0.8	67.0	67.0	Cote d'Ivoire
31	CL	0.5	30.5	30.5	Chile
32	CM	0.0	74.0	74.0	Cameroon
33	CN	-0.6	58.0	58.0	China
34	CO	-0.3	62.5	62.5	Colombia
35	CR	0.0	46.0	46.0	Costa Rica
36	CU	0.3	53.5	53.5	Cuba
37	CV	0.0	40.0	40.0	Cabo Verde
38	CY	1.4	41.0	41.0	Cyprus
39	CZ	-0.7	47.5	47.5	Czechia
40	DE	0.0	21.0	21.0	Germany
41	DJ	0.6	67.0	67.0	Djibouti
42	DK	0.0	10.0	10.0	Denmark
43	DM	0.3	43.5	43.5	Dominica
44	DO	0.0	68.0	68.0	Dominican Republic
45	DZ	0.1	66.5	66.5	Algeria
46	EC	-0.4	66.0	66.0	Ecuador
47	EE	-1.0	31.0	31.0	Estonia
48	EG	0.2	69.0	69.0	Egypt, Arab Rep.
49	ER	0.3	76.5	76.5	Eritrea
50	ES	0.5	37.5	37.5	Spain

51	ET	-0.5	64.5	64.5	Ethiopia
52	FI	0.3	11.5	11.5	Finland
53	FJ	NaN	47.0	47.0	Fiji
54	FR	-0.1	28.5	28.5	France
55	GA	0.6	68.0	68.0	Gabon
56	GB	0.1	26.5	26.5	United Kingdom
57	GD	NaN	48.0	48.0	Grenada
58	GE	-0.4	46.0	46.0	Georgia
59	GH	0.2	56.0	56.0	Ghana
60	GM	0.0	66.0	66.0	Gambia, The
61	GN	-0.1	75.5	75.5	Guinea
62	GQ	0.3	81.5	81.5	Equatorial Guinea
63	GR	-1.6	56.0	56.0	Greece
64	GT	0.9	71.5	71.5	Guatemala
65	GW	0.4	77.0	77.0	Guinea-Bissau
66	GY	-1.2	66.0	66.0	Guyana
67	HK	0.1	23.5	23.5	Hong Kong SAR, China
68	HN	0.5	74.5	74.5	Honduras
69	HR	-0.4	52.0	52.0	Croatia
70	HT	0.2	82.0	82.0	Haiti
71	HU	1.3	51.5	51.5	Hungary
72	ID	-0.2	67.0	67.0	Indonesia
73	IE	-0.8	27.0	27.0	Ireland
74	IL	-0.3	38.5	38.5	Israel
75	IN	-0.4	62.0	62.0	India
76	IQ	-0.5	79.5	79.5	Iraq
77	IR	0.3	73.5	73.5	Iran, Islamic Rep.
78	IS	0.8	22.0	22.0	Iceland
79	IT	-1.4	51.0	51.0	Italy
80	JM	-0.6	59.0	59.0	Jamaica
81	JO	0.1	52.5	52.5	Jordan
82	JP	0.1	26.5	26.5	Japan
83	KE	-0.5	70.5	70.5	Kenya
84	KG	-0.3	74.5	74.5	Kyrgyz Republic
85	KH	-0.2	77.0	77.0	Cambodia
86	KM	0.9	76.5	76.5	Comoros
87	KP	-0.9	87.5	87.5	Korea, Dem. People's Rep.
88	KR	-0.7	40.5	40.5	Korea, Rep.
89	KW	0.2	57.0	57.0	Kuwait
90	KZ	-0.8	68.0	68.0	Kazakhstan
91	LA	-1.0	74.0	74.0	Lao PDR
92	LB	0.6	73.0	73.0	Lebanon
93	LC	1.6	37.0	37.0	St. Lucia
94	LK	0.4	62.0	62.0	Sri Lanka
95	LR	1.5	66.5	66.5	Liberia
96	LS	0.8	59.0	59.0	Lesotho
97	LT	-0.8	42.0	42.0	Lithuania
98	LU	0.3	21.5	21.5	Luxembourg

99	LV	-1.0	46.0	46.0	Latvia
100	LY	0.4	81.0	81.0	Libya
101	MA	-0.1	62.5	62.5	Morocco
102	MD	-0.3	62.5	62.5	Moldova
103	ME	-0.4	57.0	57.0	Montenegro
104	MG	0.6	71.0	71.0	Madagascar
105	MK	0.3	58.5	58.5	North Macedonia
106	ML	0.6	69.0	69.0	Mali
107	MM	-0.8	81.0	81.0	Myanmar
108	MN	0.3	65.5	65.5	Mongolia
109	MR	0.1	69.5	69.5	Mauritania
110	MT	0.6	46.0	46.0	Malta
111	MU	0.7	46.5	46.5	Mauritius
112	MV	NaN	60.0	60.0	Maldives
113	MW	0.3	64.5	64.5	Malawi
114	MX	0.3	67.5	67.5	Mexico
115	MY	0.2	52.0	52.0	Malaysia
116	MZ	0.5	71.5	71.5	Mozambique
117	NE	0.1	67.5	67.5	Niger
118	NG	0.3	74.5	74.5	Nigeria
119	NI	1.0	76.0	76.0	Nicaragua
120	NL	0.4	18.0	18.0	Netherlands
121	NO	0.1	15.5	15.5	Norway
122	NP	-0.7	69.5	69.5	Nepal
123	NZ	0.3	11.5	11.5	New Zealand
124	OM	0.3	54.5	54.5	Oman
125	PA	0.2	63.0	63.0	Panama
126	PE	0.2	63.0	63.0	Peru
127	PG	-0.5	72.5	72.5	Papua New Guinea
128	PH	0.1	66.5	66.5	Philippines
129	PK	0.0	73.0	73.0	Pakistan
130	PL	0.3	43.5	43.5	Poland
131	PR	NaN	37.0	37.0	Puerto Rico
132	PT	0.1	37.5	37.5	Portugal
133	PY	-0.3	73.5	73.5	Paraguay
134	QA	1.0	37.0	37.0	Qatar
135	RO	-0.2	55.0	55.0	Romania
136	RS	0.3	62.5	62.5	Serbia
137	RU	0.0	72.0	72.0	Russian Federation
138	RW	0.2	48.0	48.0	Rwanda
139	SA	-0.7	52.5	52.5	Saudi Arabia
140	SB	NaN	58.0	58.0	Solomon Islands
141	SC	-1.8	39.0	39.0	Seychelles
142	SD	-0.9	82.5	82.5	Sudan
143	SE	0.5	14.5	14.5	Sweden
144	SG	0.4	15.0	15.0	Singapore
145	SI	0.5	41.5	41.5	Slovenia
146	SK	-0.7	50.5	50.5	Slovak Republic

147	SL	-0.3	67.5	67.5	Sierra Leone
148	SN	-0.7	60.5	60.5	Senegal
149	SO	-0.4	90.0	90.0	Somalia
150	SR	-0.3	61.5	61.5	Suriname
151	SS	NaN	87.0	87.0	South Sudan
152	ST	-0.3	56.5	56.5	Sao Tome and Principe
153	SV	0.5	64.5	64.5	El Salvador
154	SY	1.3	80.5	80.5	Syrian Arab Republic
155	SZ	0.7	66.5	66.5	Eswatini
156	TD	0.0	81.0	81.0	Chad
157	TG	0.0	70.0	70.0	Togo
158	TH	0.1	63.5	63.5	Thailand
159	TJ	-0.2	77.0	77.0	Tajikistan
160	TL	-0.9	62.5	62.5	Timor-Leste
161	TM	-0.2	82.0	82.0	Turkmenistan
162	TN	0.1	59.5	59.5	Tunisia
163	TR	1.3	57.5	57.5	Turkiye
164	TT	-0.3	59.5	59.5	Trinidad and Tobago
165	TZ	-0.3	63.5	63.5	Tanzania
166	UA	-0.7	70.5	70.5	Ukraine
167	UG	0.3	72.5	72.5	Uganda
168	US	0.4	29.0	29.0	United States
169	UY	-0.2	27.0	27.0	Uruguay
170	UZ	-1.4	76.0	76.0	Uzbekistan
171	VC	0.2	39.0	39.0	St. Vincent and the Grenadines
172	VE	0.5	83.5	83.5	Venezuela, RB
173	VN	-1.1	63.5	63.5	Viet Nam
174	VU	NaN	52.0	52.0	Vanuatu
175	YE	0.7	80.5	80.5	Yemen, Rep.
176	ZA	0.0	57.0	57.0	South Africa
177	ZM	0.4	65.0	65.0	Zambia
178	ZW	-0.3	78.5	78.5	Zimbabwe

```
[233]: # Max and min values
# Find the index of the maximum and minimum mean_CC_EST values
most_corrupt_idx = df_trend['mean_CPI_EST'].idxmax()
least_corrupt_idx = df_trend['mean_CPI_EST'].idxmin()

most_corrupt_country = df_trend.loc[most_corrupt_idx, 'country']
least_corrupt_country = df_trend.loc[least_corrupt_idx, 'country']

print("The most corrupt country (on average between 2012 and 2022) is", ↪
      most_corrupt_country)
print("The least corrupt country (on average between 2012 and 2022) is", ↪
      least_corrupt_country)
print("The country that became more corrupt (rose the most) on average between ↪
      2012 and 2022 is", df_trend.loc[df_trend['avg_change'].idxmax(), 'country'])
```

```
print("The country that became less corrupt (fell the most) on average between\u
↪2012 and 2022 is", df_trend.loc[df_trend['avg_change'].idxmin(), 'country'])
```

The most corrupt country (on average between 2012 and 2022) is Somalia
The least corrupt country (on average between 2012 and 2022) is Denmark
The country that became more corrupt (rose the most) on average between 2012 and 2022 is St. Lucia
The country that became less corrupt (fell the most) on average between 2012 and 2022 is Seychelles

```
[234]: # Print values of the averages across 2012 and 2022 and their respective\u
↪countries
print("The average CPI_EST value for the most corrupt country is", df_trend.
↪loc[most_corrupt_idx, 'mean_CPI_EST'])
print("The average CPI_EST value for the least corrupt country is", df_trend.
↪loc[least_corrupt_idx, 'mean_CPI_EST'])
print("The average CPI_EST value for the country that became more corrupt is", df_trend.
↪loc[df_trend['avg_change'].idxmax(), 'mean_CPI_EST'])
print("The average CPI_EST value for the country that became less corrupt is", df_trend.
↪loc[df_trend['avg_change'].idxmin(), 'mean_CPI_EST'])
```

The average CPI_EST value for the most corrupt country is 90.0
The average CPI_EST value for the least corrupt country is 10.0
The average CPI_EST value for the country that became more corrupt is 37.0
The average CPI_EST value for the country that became less corrupt is 39.0

```
[235]: # Print how much less corrupt the Seychelles became per year between 2012 and\u
↪2022
seychelles_idx = df_trend[df_trend['country'] == 'Seychelles'].index[0]
seychelles_change = df_trend.loc[seychelles_idx, 'avg_change']
print("Seychelles became less corrupt by an average of", seychelles_change, "points per year between 2012 and 2022")

# Print how much more corrupt St. Lucia became over the 10 years
st_lucia_idx = df_trend[df_trend['country'] == 'St. Lucia'].index[0]
st_lucia_change = df_trend.loc[st_lucia_idx, 'avg_change']
print("St. Lucia became more corrupt by", st_lucia_change, "points on average between 2012 and 2022")
```

Seychelles became less corrupt by an average of -1.8 points per year between 2012 and 2022

St. Lucia became more corrupt by 1.6 points on average between 2012 and 2022

```
[236]: df_correl = pd.read_csv('correlations.csv')
# Rank in order of correlation with CC_EST
df_correl = df_correl.sort_values('correlation', ascending=False)
```

```
[237]: # Remove rows in df_dr that have missing values in CC_EST column
df_dr = df[df['year'] >= 2012]
df_dr = df_dr.dropna(subset=['CPI_EST'])

# Filter to only show correlations greater than or equal to +- 0.5
df_correl_5 = df_correl[abs(df_correl['correlation'])] >= 0.5]
# Filter to only show correlations greater than or equal to +- 0.6
df_correl_6 = df_correl[abs(df_correl['correlation'])] >= 0.6]
# Filter to only show correlations greater than or equal to +- 0.7
df_correl_7 = df_correl[abs(df_correl['correlation'])] >= 0.7]
# Filter to only show correlations greater than or equal to +- 0.8
df_correl_8 = df_correl[abs(df_correl['correlation'])] >= 0.8]
# Filter to only show correlations greater than or equal to +- 0.9
df_correl_9 = df_correl[abs(df_correl['correlation'])] >= 0.9]

print(f"The number of correlations with an absolute value greater than or equal to 0.5 is", len(df_correl_5))
print(f"The number of correlations with an absolute value greater than or equal to 0.6 is", len(df_correl_6))
print(f"The number of correlations with an absolute value greater than or equal to 0.7 is", len(df_correl_7))
print(f"The number of correlations with an absolute value greater than or equal to 0.8 is", len(df_correl_8))
print(f"The number of correlations with an absolute value greater than or equal to 0.9 is", len(df_correl_9))
```

The number of correlations with an absolute value greater than or equal to 0.5 is 253
The number of correlations with an absolute value greater than or equal to 0.6 is 128
The number of correlations with an absolute value greater than or equal to 0.7 is 54
The number of correlations with an absolute value greater than or equal to 0.8 is 7
The number of correlations with an absolute value greater than or equal to 0.9 is 1

```
[238]: # Merge together corruption values and the largest correlators
# Create variables that are a list of the variables that are highly correlated with CC_EST
vars_5 = df_correl_5.columns.tolist()
vars_5 = [var for var in vars_5 if var in df_dr.columns]
vars_6 = df_correl_6.columns.tolist()
vars_6 = [var for var in vars_6 if var in df_dr.columns]
vars_7 = df_correl_7.columns.tolist()
vars_7 = [var for var in vars_7 if var in df_dr.columns]
vars_8 = df_correl_8.columns.tolist()
```

```

vars_8 = [var for var in vars_8 if var in df_dr.columns]
vars_9 = df_correl_9.columns.tolist()
vars_9 = [var for var in vars_9 if var in df_dr.columns]

# Merge the corruption values with the variables that are highly correlated with CC_EST
df_5 = df_dr[['iso2c', 'country', 'year', 'CPI_EST', 'NY_GDP_PCAP_KD',  

    ↵'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel']] + vars_5]
df_6 = df_dr[['iso2c', 'country', 'year', 'CPI_EST', 'NY_GDP_PCAP_KD',  

    ↵'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel']] + vars_6]
df_7 = df_dr[['iso2c', 'country', 'year', 'CPI_EST', 'NY_GDP_PCAP_KD',  

    ↵'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel']] + vars_7]
df_8 = df_dr[['iso2c', 'country', 'year', 'CPI_EST', 'NY_GDP_PCAP_KD',  

    ↵'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel']] + vars_8]
df_9 = df_dr[['iso2c', 'country', 'year', 'CPI_EST', 'NY_GDP_PCAP_KD',  

    ↵'NY_GDP_PCAP_CD', 'NY_GDP_PCAP_KD_rel', 'NY_GDP_PCAP_CD_rel']] + vars_9]
# Build random forest model to predict corruption from highly correlated variables
# Simplify notation:
x5 = df_5.iloc[:,2:].values
x6 = df_6.iloc[:,2:].values
x7 = df_7.iloc[:,2:].values
x8 = df_8.iloc[:,2:].values
x9 = df_9.iloc[:,2:].values

y = df_dr['CPI_EST'].values

```

```

[239]: def prepare_lead_data(df, target_col, lead_cols, time_col, country_col):
    """Shifts features to create lead versions for backcasting,
    accounting for country and year structure.
    """
    df_lead = df.copy()

    for col in lead_cols:
        df_lead[f'{col}_lead1'] = df_lead.sort_values([time_col]).  

            ↵groupby(country_col)[col].shift(-1)

    df_lead.fillna(0, inplace=True)
    return df_lead

def remove_correlated_features(df, threshold):
    """Removes highly correlated features based on a threshold."""
    corr_matrix = df.dropna().corr().abs()
    upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).  

        ↵astype(bool))

```

```

cols_to_drop = [col for col in upper.columns if any(upper[col] > threshold)]
df = df.drop(cols_to_drop, axis=1)
return df

def custom_cross_val_score(model, X, y, cv):
    """Calculates cross-validation scores for a given model."""
    scores = []
    kf = KFold(n_splits=cv)

    for train_index, val_index in kf.split(X):
        x_train, x_val = X.iloc[train_index], X.iloc[val_index]
        y_train, y_val = y.iloc[train_index], y.iloc[val_index]

        model.fit(x_train, y_train, eval_set=[(x_val, y_val)], verbose=False)

        y_val_pred = model.predict(x_val)
        r2_val = round(r2_score(y_val, y_val_pred), 5)
        scores.append(r2_val)

    return scores

```

```

[240]: def custom_objective(y_true, y_pred, indices):
    """
    Custom objective function that penalizes predictions close to or beyond the
    bounds of 0 and 100,
    penalizes large changes from the previous year's prediction, and penalizes
    an overall movement greater than 20 from the 2012 value.
    """

    # Convert y_true and y_pred to pandas Series
    y_true = pd.Series(y_true)
    y_pred = pd.Series(y_pred)

    # Select the corresponding value from y_2012 for each y_true and y_pred
    y_2012 = df['CPI_EST_avg'].loc[indices]

    # Calculate the squared error
    squared_error = (y_true - y_pred) ** 2

    # Calculate the penalty term
    penalty = np.abs(y_pred - y_2012)

    # Add the penalty term to the squared error
    penalized_squared_error = squared_error + penalty

    # Calculate the first derivative (gradient) of the penalized squared error
    grad = -2 * (y_true - y_pred) + np.sign(y_pred - y_2012)

```

```

# Calculate the second derivative (Hessian) of the penalized squared error
hess = np.ones_like(grad) * 2

return grad, hess

class TqdmCallback(xgb.callback.TrainingCallback):
    def __init__(self, bar):
        self._bar = bar

    def after_iteration(self, model, epoch, evals_log):
        self._bar.update(1)
        return False

def build_and_evaluate_model(df, target_col_name, var_list, n_leads=2):
    """
    Builds, evaluates, and returns results for an XGBoost model.
    """

    # Create a copy of the DataFrame to avoid modifying the original
    df = df.copy()

    # Exclude 'iso2c', 'country', target_col_name, and lead variables from
    # var_list
    var_list = [var for var in var_list if var not in ['iso2c', 'country', target_col_name] + [f"{target_col_name}_lead{i}" for i in range(1, n_leads + 1)]]

    # Remove rows with invalid values in the target column
    df = df[np.isfinite(df[target_col_name]) & (abs(df[target_col_name]) <= 1e30)]

    # Add a new feature for the previous year's target value
    df[target_col_name + '_prev'] = df.groupby('iso2c')[target_col_name].shift()
    var_list.append(target_col_name + '_prev')

    X = df[var_list]

    df_train_val, df_test = train_test_split(df, test_size=0.2, random_state=0)

    x_train_val, x_test, y_train_val, y_test = train_test_split(X, df[target_col_name], test_size=0.2, random_state=0)

    # Remove correlated features
    x_lead = x_train_val
    x_test = x_test[x_lead.columns]

    # Align datasets before model training
    x_lead, x_test = x_lead.align(x_test, join='inner', axis=1)

```

```

model = XGBRegressor(
    objective='reg:squarederror',
    random_state=0,
    alpha=1.0,
    reg_lambda=10.0,
    early_stopping_rounds=10
)

with tqdm(total=model.get_params()['n_estimators']) as pbar:
    model.fit(
        x_lead,
        y_train_val,
        eval_set=[(x_test, y_test)],
        verbose=False,
        callbacks=[TqdmCallback(pbar)]
)

```

y_train_val_pred = model.predict(x_lead)
y_test_pred = model.predict(x_test)

Clip the predictions to be within the desired range
y_train_val_pred = np.clip(y_train_val_pred, 0, 100)
y_test_pred = np.clip(y_test_pred, 0, 100)

r2_train_val = round(r2_score(y_train_val, y_train_val_pred), 5)
r2_test = round(r2_score(y_test, y_test_pred), 5)

rmse_train_val = round(np.sqrt(mean_squared_error(y_train_val, y_train_val_pred)), 5)
rmse_test = round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 5)

cv_scores = custom_cross_val_score(model, x_train_val, y_train_val, cv=5)
mean_cv_score = round(np.mean(cv_scores), 5)

Get the feature importances
feature_importances = model.feature_importances_

Get the feature names from x_lead
feature_names = x_lead.columns.tolist()

Align feature importances with feature names
feature_importances_aligned = pd.Series(feature_importances, index=feature_names)

Create a DataFrame with the aligned feature importances

```

feature_importances_df = pd.DataFrame({
    'Feature': feature_importances_aligned.index,
    'Importance': np.round(feature_importances_aligned.values, 4)
}).sort_values(by='Importance', ascending=False)

return r2_train_val, r2_test, rmse_train_val, rmse_test, mean_cv_score, □
feature_importances_df, x_test.index, y_test_pred, y_test, df, model, □
feature_importances

```

```

[241]: ## TESTS ACROSS DIFFERENT CORRELATORY LEVELS
# Set the style to 'default' to make the background white
style.use('default')

# Replace 'vars_full' with your actual list of variables, excluding 'iso2c', □
# 'country', and 'CC_EST'
vars_full = [var for var in df.columns if var not in ['iso2c', 'country', □
# 'CPI_EST']]]

print("Running model on full dataset")
for i in range(0, 9):
    # Calculate the correlation of each variable with 'CC_EST'
    correl = df[vars_full].select_dtypes(include=['number']).corrwith(df['CPI_EST']).abs()

    # Get the variables with a correlation greater than or equal to the
    # threshold
    vars_correl = correl[correl >= 0.1 * i].index.tolist()

    results = build_and_evaluate_model(df, 'CPI_EST', vars_correl, n_leads=2)

    print(f"\nModel with correlation >= 0.{i}:")
    print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
    print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
    print(f"Mean cross-validation score: {results[4]}\n")
    print(results[5]) # Feature importances
    print('\n')
    print(results[9][['country', 'year', 'CPI_EST']])

    # Generate the graphs
    y_test = results[8]
    y_test_pred = results[7]

    # Create a scatter plot for the actual vs predicted values
    abs_diffs = np.abs(y_test - y_test_pred)

    # Create a DataFrame with the actual values and absolute differences

```

```

df_plot = pd.DataFrame({'Actual': y_test, 'AbsDifference': abs_diffs})

# Define the bins for the actual values
bins = np.linspace(0, 100, 50)

# Calculate the mid-points of the bins
bin_midpoints = bins[:-1] + np.diff(bins) / 2

# Create a new column for the binned actual values
df_plot['ActualBin'] = pd.cut(df_plot['Actual'], bins, labels=bin_midpoints)

# Group by the binned actual values and calculate the variance of the
absolute differences for each group
var_abs_diffs = df_plot.groupby('ActualBin')['AbsDifference'].var()

fig = plt.figure(figsize=(10, 10))
gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1])
ax0 = plt.subplot(gs[0])
ax0.scatter(y_test, y_test_pred, alpha=0.7, color='#00688B', s=20)
ax0.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], lw=3)
ax0.set_xlim([0, 100])
ax0.set_ylabel('Predicted', fontsize=14)
ax0.set_title(f'Actual vs Predicted Values and Variance of Absolute'
Differences\n Correlation > 0.{i}', fontsize=16)

# Hide the right and top spines
ax0.spines['right'].set_visible(False)
ax0.spines['top'].set_visible(False)

# Only show ticks on the left and bottom spines
ax0.yaxis.set_ticks_position('left')
ax0.xaxis.set_ticks_position('bottom')

ax0.grid(True, color='grey', linestyle='-', linewidth=0.25, alpha=0.5)

# Create a line plot for the binned actual values vs variance of the
absolute differences
ax1 = plt.subplot(gs[1])

# Apply LOESS to smooth the variance curve
smoothed = lowess(var_abs_diffs, var_abs_diffs.index, frac=0.5)
index, data = zip(*smoothed)
ax1.plot(index, data, color='#00688B')

ax1.set_ylim([0, 15])
ax1.set_xlim([0, 100])
ax1.set_ylabel('Variance of Abs. Diff.', fontsize=14)

```

```

# Hide the right and top spines
ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)
ax1.set_xticklabels([])
ax1.set_xticks([])

# Only show ticks on the left and bottom spines
ax1.yaxis.set_ticks_position('left')
ax1.xaxis.set_ticks_position('bottom')

ax1.grid(axis='y', color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

plt.tight_layout()
plt.show()

```

Running model on full dataset

```

0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

warnings.warn(
51it [00:02, 22.67it/s]

```

Model with correlation >= 0.0:
Training+Validation R^2: 0.99942, RMSE: 0.45797
Testing R^2: 0.98508, RMSE: 2.51085
Mean cross-validation score: 0.98438

	Feature	Importance
512	CPI_EST_avg	0.8700
515	CPI_EST_prev	0.0289
387	SP_DYN_LE00_FE_IN	0.0030
117	EN_ATM_CO2E_GF_KT	0.0026
513	NY_GDP_PCAP_KD_rel	0.0025
..
265	NY_ADJ_AEDU_CD	0.0000
263	NV_SRV_TOTL_KN	0.0000
260	NV_SRV_TOTL_CN	0.0000
259	NV_SRV_TOTL_CD	0.0000
258	NV_MNF_TXTL_ZS_UN	0.0000

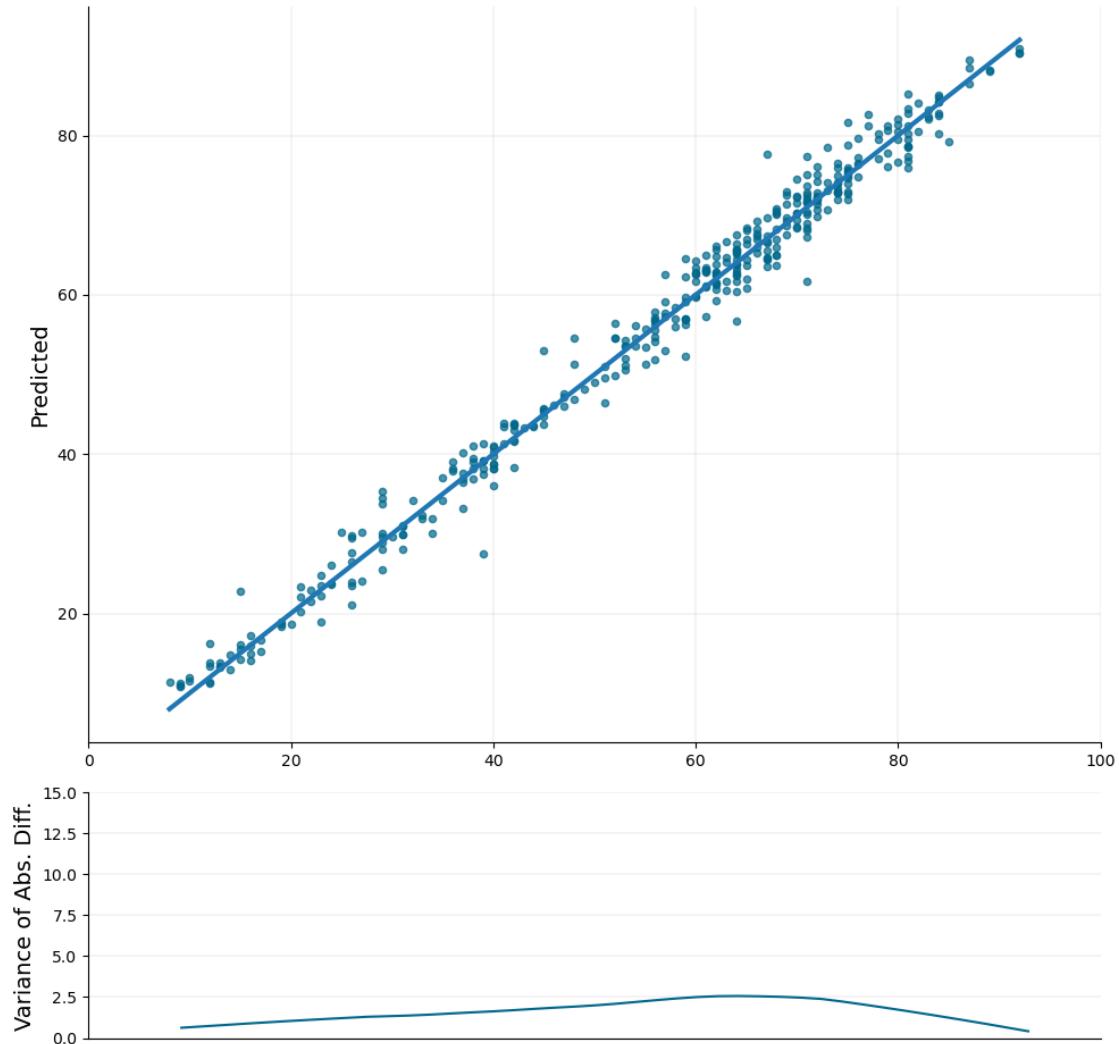
[516 rows x 2 columns]

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0

116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0
13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.0



Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-

```
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
```

```
    warnings.warn(
50it [00:01, 32.87it/s]
```

```
Model with correlation >= 0.1:
Training+Validation R^2: 0.99924, RMSE: 0.52322
Testing R^2: 0.98542, RMSE: 2.48173
Mean cross-validation score: 0.98455
```

	Feature	Importance
310	CPI_EST_avg	0.8656
312	CPI_EST_prev	0.0283
141	NY_ADJ_DFOR_GN_ZS	0.0035
311	NY_GDP_PCAP_KD_rel	0.0024
238	SP_POP_1564_TO_ZS	0.0023
259	SP_POP_65UP_MA_ZS	0.0023
68	EN_ATM_CO2E_GF_KT	0.0021
219	SP_DYN_LEOO_FE_IN	0.0019
195	SH_DTH_1519	0.0018
162	NY_GDS_TOTL_ZS	0.0017
292	TM_VAL_SERV_CD_WT	0.0017
23	BX_GSR_GNFS_CD	0.0015
220	SP_DYN_LEOO_IN	0.0013
62	EG_IMP_CONS_ZS	0.0012
128	NV_IND_MANF_CD	0.0012
107	NE_CON_PRVT_PC_KD	0.0012
270	SP_POP_DPND_YG	0.0012
277	TM_VAL_FOOD_ZS_UN	0.0011
51	DT_ODA_ODAT_PC_ZS	0.0010
207	SH_DYN_NMRT	0.0010
285	TM_VAL_MRCH_R2_ZS	0.0010
73	EN_ATM_CO2E_SF_ZS	0.0010
282	TM_VAL_MRCH_CD_WT	0.0009
170	NY_TRF_NCTR_CD	0.0009
109	NE_CON_TOTL_CD	0.0008
234	SP_POP_1519_FE_5Y	0.0008
296	TX_VAL_INSF_ZS_WT	0.0008
97	IT_CEL_SETS_P2	0.0008
225	SP_POP_0004_FE_5Y	0.0008
256	SP_POP_6569_FE_5Y	0.0008
148	NY_ADJ_NNTY_CD	0.0008
201	SH_DYN_1014	0.0008
171	PA_NUS_ATLS	0.0008
210	SM_POP_NETM	0.0007
165	NY_GNP_PCAP_CD	0.0007

36	DC_DAC_GBRL_CD	0.0007
211	SM_POP_REFG_OR	0.0007
100	IT_NET_USER_ZS	0.0007
28	BX_GSR_TRAV_ZS	0.0007
232	SP_POP_1014_FE_5Y	0.0007
240	SP_POP_2024_MA_5Y	0.0007
262	SP_POP_7074_MA_5Y	0.0007
127	NV_AGR_TOTL_ZS	0.0006
290	TM_VAL_MRCH_WL_CD	0.0006
38	DC_DAC_JPNL_CD	0.0006
271	SP_POP_GROW	0.0006
59	EG_ELC_RNWX_KH	0.0006
190	SE_SEC_ENRR_MA	0.0006
191	SE_TER_ENRR	0.0006
230	SP_POP_0509_FE_5Y	0.0006
224	SP_DYN_T065_MA_ZS	0.0006
82	FD_AST_PRVT_GD_ZS	0.0005
306	TX_VAL_MRCH_WL_CD	0.0005
139	NY_ADJ_AEDU_GN_ZS	0.0005
140	NY_ADJ_DFOR_CD	0.0005
9	BM_GSR_MRCH_CD	0.0005
76	EN_CO2_OTHX_ZS	0.0005
75	EN_CO2_ETOT_ZS	0.0005
74	EN_ATM_NOXE_EG_KT_CE	0.0005
291	TM_VAL_OTHZ_ZS_WT	0.0005
72	EN_ATM_CO2E_PC	0.0005
115	NE_EXP_GNFS_CD	0.0005
13	BM_KLT_DINV_CD_WD	0.0005
154	NY_GDP_MINR_RT_ZS	0.0005
283	TM_VAL_MRCH_HI_ZS	0.0005
44	DT_NFL_UNDP_CD	0.0005
278	TM_VAL_FUEL_ZS_UN	0.0005
52	DT_TDS_MLAT_CD	0.0005
209	SH_IMM_MEAS	0.0005
308	TX_VAL_SERV_CD_WT	0.0005
45	DT_NFL_UNFP_CD	0.0005
183	SE_PRM_GINT_ZS	0.0005
175	SE_ENR_SECO_FM_ZS	0.0005
1	AG_LND_AGRI_ZS	0.0005
89	FM_LBL_BMNY_GD_ZS	0.0005
87	FM_AST_PRVT_GD_ZS	0.0005
17	BN_CAB_XOKA_GD_ZS	0.0005
125	NE_RSB_GNFS_ZS	0.0004
124	NE_IMP_GNFS_ZS	0.0004
108	NE_CON_PRVT_ZS	0.0004
103	NE_CON_GOVTD_KD	0.0004
307	TX_VAL_OTHZ_ZS_WT	0.0004
203	SH_DYN_2024	0.0004

242	SP_POP_2529_MA_5Y	0.0004
221	SP_DYN_LE00_MA_IN	0.0004
150	NY_GDP_DEFLL_KD_ZG	0.0004
236	SP_POP_1564_FE_ZS	0.0004
273	SP_RUR_TOTL_ZS	0.0004
216	SP_DYN_IMRT_FE_IN	0.0004
173	SE_ENR_PRIM_FM_ZS	0.0004
239	SP_POP_2024_FE_5Y	0.0004
267	SP_POP_BRTH_MF	0.0004
143	NY_ADJ_DKAP_GN_ZS	0.0004
156	NY_GDP_MKTP_KD	0.0004
80	ER_GDP_FWTL_M3_KD	0.0004
53	EG_ELC_COAL_ZS	0.0004
14	BM_KLT_DINV_WD_GD_ZS	0.0004
24	BX_GSR_INSF_ZS	0.0004
12	BM_GSR_TRAN_ZS	0.0004
55	EG_ELC_HYRO_ZS	0.0004
21	BX_GRT_TECH_CD_WD	0.0004
39	DC_DAC_NLDL_CD	0.0004
101	MS_MIL_XPND_CD	0.0003
174	SE_ENR_PRSC_FM_ZS	0.0003
205	SH_DYN_MORT_FE	0.0003
260	SP_POP_65UP_TO_ZS	0.0003
135	NV_SRV_TOTL_CD	0.0003
136	NV_SRV_TOTL_KD	0.0003
137	NV_SRV_TOTL_ZS	0.0003
261	SP_POP_7074_FE_5Y	0.0003
79	EN_URB_MCTY_TL_ZS	0.0003
78	EN_POP_DNST	0.0003
77	EN_CO2_TRAN_ZS	0.0003
288	TM_VAL_MRCH_R6_ZS	0.0003
58	EG_ELC_PETR_ZS	0.0003
286	TM_VAL_MRCH_R4_ZS	0.0003
129	NV_IND_MANF_KD	0.0003
60	EG_ELC_RNWX_ZS	0.0003
93	FS_AST_PRVT_GD_ZS	0.0003
15	BM_TRF_PRVT_CD	0.0003
153	NY_GDP_FRST_RT_ZS	0.0003
281	TM_VAL_MRCH_AL_ZS	0.0003
264	SP_POP_7579_MA_5Y	0.0003
279	TM_VAL_MANF_ZS_UN	0.0003
235	SP_POP_1519_MA_5Y	0.0003
265	SP_POP_80UP_FE_5Y	0.0003
18	BN_GSR_FCTY_CD	0.0003
276	TG_VAL_TOTL_GD_ZS	0.0003
92	FP_CPI_TOTL_ZG	0.0003
212	SP_ADO_TFRT	0.0003
147	NY_ADJICTR_GN_ZS	0.0003

99	IT_MLT_MAIN_P2	0.0003
189	SE_SEC_ENRR_FE	0.0003
86	FM_AST_CGOV_ZG_M3	0.0003
246	SP_POP_4044_FE_5Y	0.0003
245	SP_POP_3539_MA_5Y	0.0003
305	TX_VAL_MRCH_R6_ZS	0.0003
105	NE_CON_PRVT_CD	0.0003
184	SE_PRM_REPT_ZS	0.0003
91	FP_CPI_TOTL	0.0003
121	NE_GDI_TOTL_KD	0.0003
122	NE_IMP_GNFS_CD	0.0003
182	SE_PRM_ENRR_FE	0.0003
84	FI_RES_TOTL_MO	0.0003
252	SP_POP_5559_FE_5Y	0.0003
297	TX_VAL_MANF_ZS_UN	0.0003
81	ER_H2O_INTR_PC	0.0003
247	SP_POP_4044_MA_5Y	0.0003
196	SH_DTH_2024	0.0002
244	SP_POP_3539_FE_5Y	0.0002
22	BX_GSR_FCTY_CD	0.0002
180	SE_PRM_ENRL_FE_ZS	0.0002
241	SP_POP_2529_FE_5Y	0.0002
202	SH_DYN_1519	0.0002
61	EG_FEC_RNEW_ZS	0.0002
172	PA_NUS_FCRF	0.0002
31	BX_TRF_CURR_CD	0.0002
187	SE_SEC_ENRL_GC_FE_ZS	0.0002
186	SE_SEC_ENRL_FE_ZS	0.0002
185	SE_PRM_TCHR_FE_ZS	0.0002
250	SP_POP_5054_FE_5Y	0.0002
49	DT_ODA_ODAT GI_ZS	0.0002
30	BX_PEF_TOTL_CD_WD	0.0002
157	NY_GDP_PCAP_CD	0.0002
272	SP_RUR_TOTL_ZG	0.0002
111	NE_CON_TOTL_ZS	0.0002
302	TX_VAL_MRCH_R1_ZS	0.0002
214	SP_DYN_AMRT_MA	0.0002
85	FI_RES_XGLD_CD	0.0002
114	NE_DAB_TOTL_ZS	0.0002
43	DTNFL_UNCF_CD	0.0002
112	NE_DAB_TOTL_CD	0.0002
37	DC_DAC_ITAL_CD	0.0002
19	BN_TRF_CURR_CD	0.0002
4	AG_YLD_CREL_KG	0.0002
90	FM_LBL_BMNY_ZG	0.0002
309	TX_VAL_TRVL_ZS_WT	0.0002
3	AG_PRD_CROP_XD	0.0002
102	NE_CON_GOVT_CD	0.0002

96	IS_AIR_PSGR	0.0002
301	TX_VAL_MRCH_HI_ZS	0.0002
299	TX_VAL_MRCH_AL_ZS	0.0002
298	TX_VAL_MMTL_ZS_UN	0.0002
144	NY_ADJ_DMIN_GN_ZS	0.0002
274	SP_URB_GROW	0.0002
16	BM_TRF_PWKR_CD_DT	0.0002
284	TM_VAL_MRCH_R1_ZS	0.0002
70	EN_ATM_CO2E_LF_KT	0.0002
29	BX_KLT_DINV_CD_WD	0.0002
146	NY_ADJ_DRES_GN_ZS	0.0002
47	DT_ODA_ALLD_KD	0.0002
289	TM_VAL_MRCH_RS_ZS	0.0002
133	NV_MNF_FBT0_ZS_UN	0.0002
295	TX_VAL_FUEL_ZS_UN	0.0002
199	SH_DTH_NMRT	0.0002
34	DC_DAC_CHEL_CD	0.0001
233	SP_POP_1014_MA_5Y	0.0001
35	DC_DAC_DEUL_CD	0.0001
33	DC_DAC_CANL_CD	0.0001
32	BX_TRF_PWKR_DT_GD_ZS	0.0001
94	IS_AIR_DPRT	0.0001
243	SP_POP_3034_MA_5Y	0.0001
293	TM_VAL_TRAN_ZS_WT	0.0001
2	AG_LND_CROP_ZS	0.0001
5	BG_GSR_NFSV_GD_ZS	0.0001
6	BM_GSR_CMCP_ZS	0.0001
304	TX_VAL_MRCH_R5_ZS	0.0001
303	TX_VAL_MRCH_R4_ZS	0.0001
7	BM_GSR_FCTY_CD	0.0001
294	TX_VAL_FOOD_ZS_UN	0.0001
287	TM_VAL_MRCH_R5_ZS	0.0001
248	SP_POP_4549_FE_5Y	0.0001
280	TM_VAL_MMTL_ZS_UN	0.0001
41	DC_DAC_USAL_CD	0.0001
20	BX_GRT_EXTA_CD_WD	0.0001
266	SP_POP_80UP_MA_5Y	0.0001
258	SP_POP_65UP_FE_ZS	0.0001
254	SP_POP_6064_FE_5Y	0.0001
249	SP_POP_4549_MA_5Y	0.0001
40	DC_DAC_TOTL_CD	0.0001
0	AG_CON_FERT_ZS	0.0001
42	DT_DOD_DECT_GN_ZS	0.0001
56	EG_ELC_LOSS_ZS	0.0001
181	SE_PRM_ENRL_TC_ZS	0.0001
179	SE_PRM_ENRL	0.0001
176	SE_PRE_ENRR	0.0001
169	NY_TAX_NIND_CD	0.0001

168	NY_GSR_NFCY_CD	0.0001
167	NY_GNSICTR_ZS	0.0001
65	EG_USE_CRNW_ZS	0.0001
160	NY_GDP_TOTL_RT_ZS	0.0001
66	EG_USE_ELEC_KH_PC	0.0001
152	NY_GDP_FCST_KD	0.0001
69	EN_ATM_CO2E_GF_ZS	0.0001
145	NY_ADJ_DNGY_GN_ZS	0.0001
71	EN_ATM_CO2E_LF_ZS	0.0001
138	NY_ADJ_AEDU_CD	0.0001
134	NV_MNF_TXTL_ZS_UN	0.0001
132	NV_IND_TOTL_ZS	0.0001
130	NV_IND_TOTL_CD	0.0001
83	FI_RES_TOTL_CD	0.0001
123	NE_IMP_GNFS_KD	0.0001
120	NE_GDI_TOTL_CD	0.0001
116	NE_EXP_GNFS_KD	0.0001
88	FM_AST_PRVT_ZG_M3	0.0001
104	NE_CON_GOVT_ZS	0.0001
188	SE_SEC_ENRR	0.0001
166	NY_GNSICTR_CD	0.0001
46	DT_ODA_ALLD_CD	0.0001
193	SH_DTH_0509	0.0001
208	SH_IMM_IDPT	0.0001
54	EG_ELC_FOSL_ZS	0.0001
204	SH_DYN_MORT	0.0001
213	SP_DYN_AMRT_FE	0.0001
11	BM_GSR_TOTL_CD	0.0000
215	SP_DYN_CBRT_IN	0.0000
231	SP_POP_0509_MA_5Y	0.0000
8	BM_GSR_GNFS_CD	0.0000
10	BM_GSR_NFSV_CD	0.0000
192	SG_LAW_INDX	0.0000
142	NY_ADJ_DKAP_CD	0.0000
229	SP_POP_0014_TO_ZS	0.0000
48	DT_ODA_ODAT_CD	0.0000
149	NY_ADJ_NNTY_PC_CD	0.0000
131	NV_IND_TOTL_KD	0.0000
126	NE_TRD_GNFS_ZS	0.0000
206	SH_DYN_MORT_MA	0.0000
228	SP_POP_0014_MA_ZS	0.0000
227	SP_POP_0014_FE_ZS	0.0000
300	TX_VAL_MRCH_CD_WT	0.0000
226	SP_POP_0004_MA_5Y	0.0000
119	NE_GDI_FTOT_KD	0.0000
118	NE_GDI_FTOT_CD	0.0000
117	NE_EXP_GNFS_ZS	0.0000
217	SP_DYN_IMRT_IN	0.0000

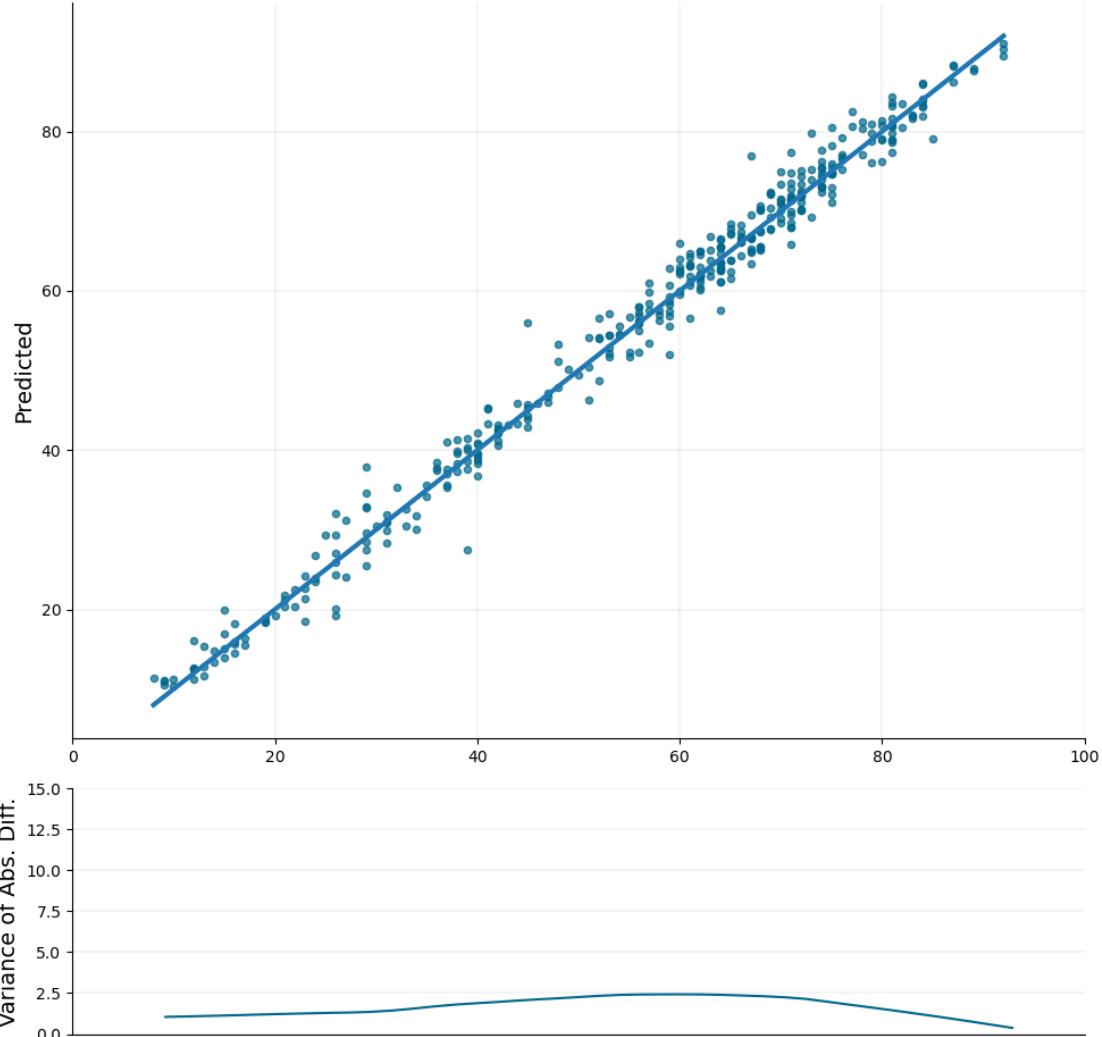
113	NE_DAB_TOTL_KD	0.0000
110	NE_CON_TOTL_KD	0.0000
218	SP_DYN_IMRT_MA_IN	0.0000
106	NE_CON_PRVT_KD	0.0000
223	SP_DYN_TO65_FE_ZS	0.0000
98	IT_MLT_MAIN	0.0000
151	NY_GDP_FCST_CD	0.0000
155	NY_GDP_MKTP_CD	0.0000
67	EG_USE_PCAP_KG_OE	0.0000
237	SP_POP_1564_MA_ZS	0.0000
57	EG_ELC_NUCL_ZS	0.0000
251	SP_POP_5054_MA_5Y	0.0000
194	SH_DTH_1014	0.0000
253	SP_POP_5559_MA_5Y	0.0000
255	SP_POP_6064_MA_5Y	0.0000
26	BX_GSR_NFSV_CD	0.0000
257	SP_POP_6569_MA_5Y	0.0000
178	SE_PRM_DURS	0.0000
25	BX_GSR_MRCH_CD	0.0000
177	SE_PRM_AGES	0.0000
197	SH_DTH_IMRT	0.0000
63	EG_USE_COMM_CL_ZS	0.0000
263	SP_POP_7579_FE_5Y	0.0000
27	BX_GSR_TOTL_CD	0.0000
198	SH_DTH_MORT	0.0000
200	SH_DYN_0509	0.0000
222	SP_DYN_TFRT_IN	0.0000
268	SP_POP_DPND	0.0000
269	SP_POP_DPND_OL	0.0000
64	EG_USE_COMM_FO_ZS	0.0000
164	NY_GNP_MKTP_CD	0.0000
163	NY_GNP_ATLS_CD	0.0000
95	IS_AIR_GOOD_MT_K1	0.0000
161	NY_GDS_TOTL_CD	0.0000
275	SP_URB_TOTL_IN_ZS	0.0000
159	NY_GDP_PETR_RT_ZS	0.0000
158	NY_GDP_PCAP_KD	0.0000
50	DT_ODA_ODAT_KD	0.0000

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0
116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0

13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.1



```
Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
```

```
    warnings.warn(
51it [00:00, 58.26it/s]
```

Model with correlation >= 0.2:
 Training+Validation R^2: 0.99901, RMSE: 0.59837
 Testing R^2: 0.98503, RMSE: 2.51451
 Mean cross-validation score: 0.98413

	Feature	Importance
212	CPI_EST_avg	0.8881
214	CPI_EST_prev	0.0244
180	SP_POP_65UP_MA_ZS	0.0027
41	EG_ELC_RNWX_KH	0.0020
97	NY_GDP_FRST_RT_ZS	0.0016
147	SP_DYN_T065_MA_ZS	0.0015
77	NE_IMP_GNFS_KD	0.0014
90	NY_ADJ_DFOR_GN_ZS	0.0014
45	EG_USE_CRNW_ZS	0.0013
142	SP_DYN_LE00_FE_IN	0.0013
49	EN_POP_DNST	0.0012
188	SP_POP_DPND	0.0012
131	SH_DYN_NMRT	0.0012
63	NE_CON_GOVT_KD	0.0011
213	NY_GDP_PCAP_KD_rel	0.0011
205	TM_VAL_SERV_CD_WT	0.0011
163	SP_POP_2024_MA_5Y	0.0010
191	SP_POP_GROW	0.0010
177	SP_POP_6569_FE_5Y	0.0010
197	TM_VAL_FOOD_ZS_UN	0.0009
68	NE_CON_TOTL_CD	0.0009
82	NV_IND_MANF_KD	0.0009
124	SH_DYN_0509	0.0009
125	SH_DYN_1014	0.0009
37	DT_ODA_ODAT_PC_ZS	0.0009
172	SP_POP_5054_MA_5Y	0.0009
17	BX_GSR_GNFS_CD	0.0009
162	SP_POP_2024_FE_5Y	0.0008
166	SP_POP_3539_MA_5Y	0.0008
185	SP_POP_7579_MA_5Y	0.0008
65	NE_CON_PRVT_KD	0.0008
173	SP_POP_5559_FE_5Y	0.0008
59	IT_CEL_SETS_P2	0.0007
87	NV_SRV_TOTL_ZS	0.0007
210	TX_VAL_MRCH_WL_CD	0.0007
1	AG_LND_AGRI_ZS	0.0007
198	TM_VAL_MRCH_AL_ZS	0.0007
121	SE_TER_ENRR	0.0007
110	SE_ENR_SECO_FM_ZS	0.0007
174	SP_POP_5559_MA_5Y	0.0007
193	SP_RUR_TOTL_ZS	0.0007

148	SP_POP_0004_FE_5Y	0.0007
196	TG_VAL_TOTL_GD_ZS	0.0007
138	SP_DYN_CBRT_IN	0.0007
79	NE_RSB_GNFS_ZS	0.0006
54	FM_LBL_BMNY_GD_ZS	0.0006
91	NY_ADJ_DKAP_GN_ZS	0.0006
106	NY_TAX_NIND_CD	0.0006
62	NE_CON_GOVTC_CD	0.0006
159	SP_POP_1564_FE_ZS	0.0006
114	SE_PRM_REPT_ZS	0.0006
126	SH_DYN_1519	0.0006
66	NE_CON_PRVT_PC_KD	0.0005
158	SP_POP_1519_MA_5Y	0.0005
151	SP_POP_0014_MA_ZS	0.0005
78	NE_IMP_GNFS_ZS	0.0005
164	SP_POP_2529_FE_5Y	0.0005
167	SP_POP_4044_FE_5Y	0.0005
84	NV_MNF_TXTL_ZS_UN	0.0005
123	SH_DTH_0509	0.0005
183	SP_POP_7074_MA_5Y	0.0005
182	SP_POP_7074_FE_5Y	0.0005
133	SH_IMM_MEAS	0.0005
143	SP_DYN_LE00_IN	0.0005
107	NY_TRF_NCTR_CD	0.0005
52	FD_AST_PRVT_GD_ZS	0.0005
61	IT_NET_USER_ZS	0.0005
20	BX_GSR_TOTL_CD	0.0005
7	BM_GSR_MRCH_CD	0.0005
51	ER_GDP_FWTL_M3_KD	0.0005
40	EG_ELC_PETR_ZS	0.0005
137	SP_DYN_AMRT_MA	0.0004
111	SE_PRE_ENRR	0.0004
11	BM_KLT_DINV_CD_WD	0.0004
50	EN_URB_MCTY_TL_ZS	0.0004
92	NY_ADJ_DRES_GN_ZS	0.0004
10	BM_GSR_TRAN_ZS	0.0004
134	SM_POP_REFG_OR	0.0004
32	DTNFL_UNFP_CD	0.0004
101	NY_GDP_TOTL_RT_ZS	0.0004
122	SG_LAW_INDX	0.0004
117	SE_SEC_ENRL_GC_FE_ZS	0.0004
116	SE_SEC_ENRL_FE_ZS	0.0004
43	EG_FEC_RNEW_ZS	0.0004
187	SP_POP_80UP_MA_5Y	0.0004
120	SE_SEC_ENRR_MA	0.0004
53	FM_AST_PRVT_GD_ZS	0.0004
26	DC_DAC_GBRL_CD	0.0004
60	IT_MLT_MAIN_P2	0.0004

168	SP_POP_4044_MA_5Y	0.0004
165	SP_POP_3539_FE_5Y	0.0004
109	SE_ENR_PRSC_FM_ZS	0.0004
13	BM_TRF_PWKR_CD_DT	0.0004
157	SP_POP_1519_FE_5Y	0.0003
12	BM_TRF_PRVT_CD	0.0003
19	BX_GSR_NFSV_CD	0.0003
186	SP_POP_80UP_FE_5Y	0.0003
171	SP_POP_5054_FE_5Y	0.0003
118	SE_SEC_ENRR	0.0003
119	SE_SEC_ENRR_FE	0.0003
170	SP_POP_4549_MA_5Y	0.0003
22	BX_TRF_CURR_CD	0.0003
33	DT_ODA_ALLD_CD	0.0003
127	SH_DYN_2024	0.0003
31	DTNFL_UNDP_CD	0.0003
16	BX_GSR_FCTY_CD	0.0003
132	SH_IMM_IDPT	0.0003
154	SP_POP_0509_MA_5Y	0.0003
27	DC_DAC_NLDL_CD	0.0003
192	SP_RUR_TOTL_ZG	0.0003
64	NE_CON_GOV_T_ZS	0.0003
44	EG_USE_COMM_CL_ZS	0.0003
202	TM_VAL_MRCH_R4_ZS	0.0003
70	NE_CON_TOTL_ZS	0.0003
74	NE_EXP_GNFS_ZS	0.0003
76	NE_IMP_GNFS_CD	0.0003
58	IS_AIR_PSGR	0.0003
57	IS_AIR_GOOD_MT_K1	0.0003
81	NV_AGR_TOTL_ZS	0.0003
86	NV_SRV_TOTL_KD	0.0003
208	TX_VAL_MRCH_CD_WT	0.0003
206	TM_VAL_TRAN_ZS_WT	0.0003
204	TM_VAL_OTHR_ZS_WT	0.0003
175	SP_POP_6064_FE_5Y	0.0003
108	SE_ENR_PRIM_FM_ZS	0.0003
100	NY_GDP_PCAP_KD	0.0003
201	TM_VAL_MRCH_R1_ZS	0.0003
200	TM_VAL_MRCH_HI_ZS	0.0003
23	BX_TRF_PWKR_DT_GD_ZS	0.0002
14	BN_TRF_CURR_CD	0.0002
56	IS_AIR_DPRT	0.0002
155	SP_POP_1014_FE_5Y	0.0002
181	SP_POP_65UP_TO_ZS	0.0002
46	EG_USE_ELEC_KH_PC	0.0002
115	SE_PRM_TCHR_FE_ZS	0.0002
15	BX_GRT_EXTA_CD_WD	0.0002
211	TX_VAL_SERV_CD_WT	0.0002

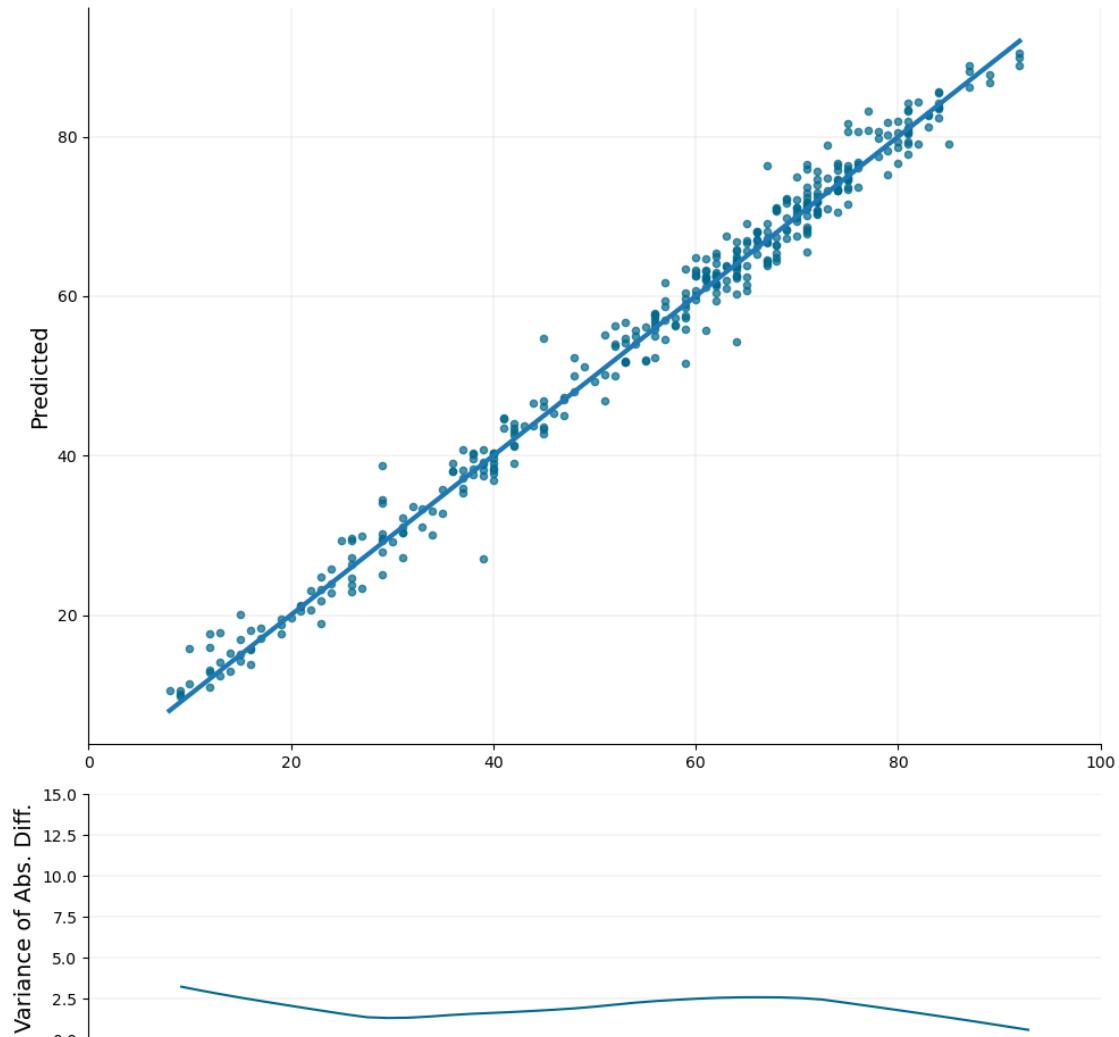
24	DC_DAC_CANL_CD	0.0002
8	BM_GSR_NFSV_CD	0.0002
195	SP_URB_TOTL_IN_ZS	0.0002
73	NE_EXP_GNFS_KD	0.0002
21	BX_KLT_DINV_CD_WD	0.0002
72	NE_EXP_GNFS_CD	0.0002
169	SP_POP_4549_FE_5Y	0.0002
71	NE_DAB_TOTL_ZS	0.0002
113	SE_PRM_ENRL_TC_ZS	0.0002
3	BG_GSR_NFSV_GD_ZS	0.0002
18	BX_GSR_MRCH_CD	0.0002
4	BM_GSR_CMCP_ZS	0.0002
150	SP_POP_0014_FE_ZS	0.0002
149	SP_POP_0004_MA_5Y	0.0002
209	TX_VAL_MRCH_HI_ZS	0.0002
194	SP_URB_GROW	0.0002
203	TM_VAL_MRCH_WL_CD	0.0002
48	EN_ATM_CO2E_PC	0.0002
130	SH_DYN_MORT_MA	0.0002
190	SP_POP_DPND_YG	0.0002
34	DT_ODA_ALLD_KD	0.0002
5	BM_GSR_FCTY_CD	0.0002
135	SP_ADO_TFRT	0.0002
176	SP_POP_6064_MA_5Y	0.0002
104	NY_GNP_MKTP_CD	0.0002
85	NV_SRV_TOTL_CD	0.0002
112	SE_PRM_ENRL_FE_ZS	0.0002
144	SP_DYN_LEOO_MA_IN	0.0002
55	FS_AST_PRVT_GD_ZS	0.0002
30	DT_NFL_UNCF_CD	0.0002
29	DC_DAC_USAL_CD	0.0002
83	NV_MNF_FBTO_ZS_UN	0.0002
179	SP_POP_65UP_FE_ZS	0.0001
184	SP_POP_7579_FE_5Y	0.0001
207	TX_VAL_MANF_ZS_UN	0.0001
189	SP_POP_DPND_DL	0.0001
6	BM_GSR_GNFS_CD	0.0001
199	TM_VAL_MRCH_CD_WT	0.0001
2	AG_YLD_CREL_KG	0.0001
140	SP_DYN_IMRT_IN	0.0001
25	DC_DAC_CHEL_CD	0.0001
35	DT_ODA_ODAT_CD	0.0001
67	NE_CON_PRVT_ZS	0.0001
75	NE_GDI_FTOT_KD	0.0001
80	NE_TRD_GNFS_ZS	0.0001
89	NY_ADJ_AEDU_GN_ZS	0.0001
93	NY_ADJ_NNTY_CD	0.0001
94	NY_ADJ_NNTY_PC_CD	0.0001

99	NY_GDP_PCAP_CD	0.0001
47	EG_USE_PCAP_KG_OE	0.0001
42	EG_ELC_RNWX_ZS	0.0001
38	EG_ELC_LOSS_ZS	0.0001
36	DT_ODA_ODAT_KD	0.0001
0	AG_CON_FERT_ZS	0.0001
136	SP_DYN_AMRT_FE	0.0001
156	SP_POP_1014_MA_5Y	0.0001
28	DC_DAC_TOTL_CD	0.0001
146	SP_DYN_TO65_FE_ZS	0.0001
129	SH_DYN_MORT_FE	0.0000
96	NY_GDP_FCST_KD	0.0000
161	SP_POP_1564_TO_ZS	0.0000
69	NE_CON_TOTL_KD	0.0000
160	SP_POP_1564_MA_ZS	0.0000
88	NY_ADJ_AEDU_CD	0.0000
153	SP_POP_0509_FE_5Y	0.0000
152	SP_POP_0014_TO_ZS	0.0000
95	NY_GDP_FCST_CD	0.0000
98	NY_GDP_MKTP_CD	0.0000
128	SH_DYN_MORT	0.0000
102	NY_GDS_TOTL_ZS	0.0000
103	NY_GNP_ATLS_CD	0.0000
105	NY_GNP_PCAP_CD	0.0000
145	SP_DYN_TFRT_IN	0.0000
141	SP_DYN_IMRT_MA_IN	0.0000
178	SP_POP_6569_MA_5Y	0.0000
139	SP_DYN_IMRT_FE_IN	0.0000
9	BM_GSR_TOTL_CD	0.0000
39	EG_ELC_NUCL_ZS	0.0000

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0
116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0
13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.2



```
Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
66it [00:00, 71.24it/s]
```

```
Model with correlation >= 0.3:
Training+Validation R^2: 0.99944, RMSE: 0.44949
Testing R^2: 0.98442, RMSE: 2.56512
Mean cross-validation score: 0.98476
```

Feature	Importance

158	CPI_EST_avg	0.9176
160	CPI_EST_prev	0.0244
159	NY_GDP_PCAP_KD_rel	0.0047
87	SH_DYN_NMRT	0.0015
132	SP_POP_65UP_MA_ZS	0.0013
131	SP_POP_65UP_FE_ZS	0.0012
26	EG_USE_CRNW_ZS	0.0011
77	SE_SEC_ENRR_MA	0.0011
104	SP_POP_0004_MA_5Y	0.0010
102	SP_DYN_TO65_MA_ZS	0.0010
76	SE_SEC_ENRR_FE	0.0010
57	NY_ADJ_DFOR_GN_ZS	0.0010
86	SH_DYN_MORT_MA	0.0009
80	SH_DYN_0509	0.0009
105	SP_POP_0014_FE_ZS	0.0007
103	SP_POP_0004_FE_5Y	0.0007
139	SP_POP_80UP_MA_5Y	0.0007
117	SP_POP_2024_FE_5Y	0.0007
124	SP_POP_5054_MA_5Y	0.0006
146	SP_URB_TOTL_IN_ZS	0.0006
97	SP_DYN_LE00_FE_IN	0.0006
116	SP_POP_1564_TO_ZS	0.0006
14	BX_GSR_GNFS_CD	0.0006
125	SP_POP_5559_FE_5Y	0.0006
23	EG_ELC_RNWX_ZS	0.0005
114	SP_POP_1564_FE_ZS	0.0005
83	SH_DYN_2024	0.0005
72	SE_PRM_REPT_ZS	0.0005
81	SH_DYN_1014	0.0005
39	IT_NET_USER_ZS	0.0005
37	IT_CEL_SETS_P2	0.0005
147	TM_VAL_MRCH_CD_WT	0.0005
98	SP_DYN_LE00_IN	0.0005
126	SP_POP_5559_MA_5Y	0.0005
17	BX_GSR_TOTL_CD	0.0005
142	SP_POP_DPND_YG	0.0005
144	SP_RUR_TOTL_ZS	0.0005
78	SE_TER_ENRR	0.0005
56	NY_ADJ_AEDU_GN_ZS	0.0005
129	SP_POP_6569_FE_5Y	0.0005
63	NY_GDP_PCAP_KD	0.0004
61	NY_GDP_FRST_RT_ZS	0.0004
136	SP_POP_7579_FE_5Y	0.0004
92	SP_DYN_AMRT_MA	0.0004
123	SP_POP_5054_FE_5Y	0.0004
137	SP_POP_7579_MA_5Y	0.0004
45	NE_EXP_GNFS_CD	0.0004
58	NY_ADJ_DKAP_GN_ZS	0.0004

66	NY_GNP_PCAP_CD	0.0004
112	SP_POP_1519_FE_5Y	0.0004
34	FM_LBL_BMNY_GD_ZS	0.0004
30	EN_URB_MCTY_TL_ZS	0.0004
118	SP_POP_2024_MA_5Y	0.0004
120	SP_POP_4044_MA_5Y	0.0004
79	SG_LAW_INDX	0.0004
25	EG_USE_COMM_CL_ZS	0.0004
89	SH_IMM_MEAS	0.0004
145	SP_URB_GROW	0.0003
101	SP_DYN_T065_FE_ZS	0.0003
90	SP_ADO_TFRT	0.0003
143	SP_RUR_TOTL_ZG	0.0003
99	SP_DYN_LE00_MA_IN	0.0003
100	SP_DYN_TFRT_IN	0.0003
7	BM_GSR_NFSV_CD	0.0003
6	BM_GSR_MRCH_CD	0.0003
127	SP_POP_6064_FE_5Y	0.0003
111	SP_POP_1014_MA_5Y	0.0003
113	SP_POP_1519_MA_5Y	0.0003
121	SP_POP_4549_FE_5Y	0.0003
122	SP_POP_4549_MA_5Y	0.0003
11	BM_TRF_PWKR_CD_DT	0.0003
134	SP_POP_7074_FE_5Y	0.0003
154	TX_VAL_MRCH_CD_WT	0.0003
49	NE_IMP_GNFS_KD	0.0003
33	FM_AST_PRVT_GD_ZS	0.0003
55	NV_SRV_TOTL_ZS	0.0003
36	IS_AIR_GOOD_MT_K1	0.0003
150	TM_VAL_OTHR_ZS_WT	0.0003
38	IT_MLT_MAIN_P2	0.0003
31	ER_GDP_FWTL_M3_KD	0.0003
50	NE_RSB_GNFS_ZS	0.0003
62	NY_GDP_PCAP_CD	0.0003
156	TX_VAL_MRCH_WL_CD	0.0003
47	NE_EXP_GNFS_ZS	0.0003
68	NY_TRF_NCTR_CD	0.0003
20	DTNFL_UNFP_CD	0.0003
19	DTNFL_UNDP_CD	0.0003
70	SE_PRE_ENRR	0.0003
41	NE_CON_PRVT_PC_KD	0.0003
53	NV_MNF_FBT0_ZS_UN	0.0003
35	FS_AST_PRVT_GD_ZS	0.0002
155	TX_VAL_MRCH_HI_ZS	0.0002
110	SP_POP_1014_FE_5Y	0.0002
138	SP_POP_80UP_FE_5Y	0.0002
32	FD_AST_PRVT_GD_ZS	0.0002
115	SP_POP_1564_MA_ZS	0.0002

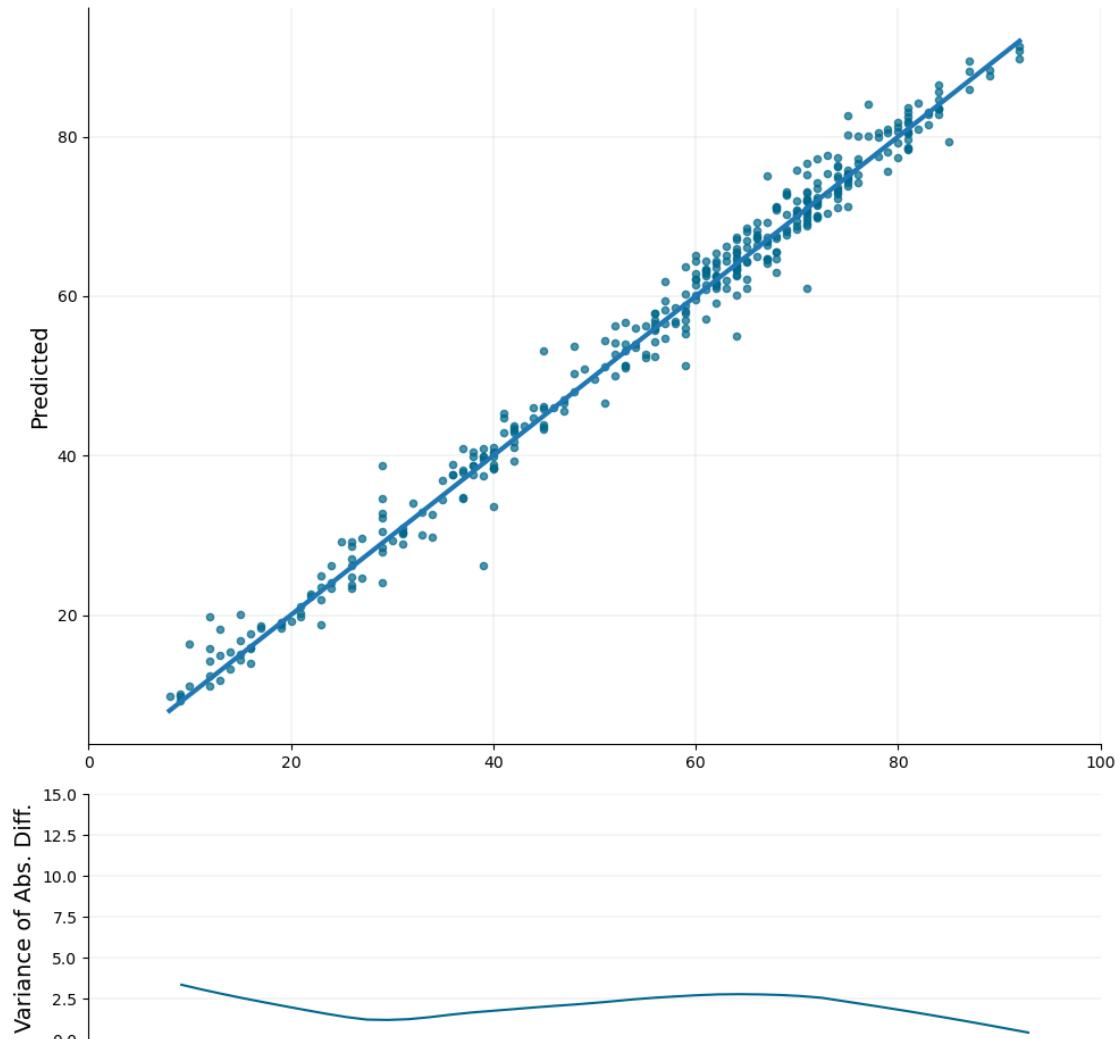
42	NE_CON_PRVT_ZS	0.0002
29	EN_ATM_CO2E_PC	0.0002
119	SP_POP_4044_FE_5Y	0.0002
27	EG_USE_ELEC_KH_PC	0.0002
13	BX_GSR_FCTY_CD	0.0002
24	EG_FEC_RNEW_ZS	0.0002
128	SP_POP_6064_MA_5Y	0.0002
18	BX_TRF_PWKR_DT_GD_ZS	0.0002
141	SP_POP_DPND_DL	0.0002
73	SE_PRM_TCHR_FE_ZS	0.0002
43	NE_CON_TOTL_ZS	0.0002
9	BM_GSR_TRAN_ZS	0.0002
74	SE_SEC_ENRL_GC_FE_ZS	0.0002
75	SE_SEC_ENRR	0.0002
10	BM_TRF_PRVT_CD	0.0002
69	SE_ENR_PRSC_FM_ZS	0.0002
67	NY_TAX_NIND_CD	0.0002
65	NY_GDS_TOTL_ZS	0.0002
82	SH_DYN_1519	0.0002
149	TM_VAL_MRCH_WL_CD	0.0002
44	NE_DAB_TOTL_ZS	0.0002
85	SH_DYN_MORT_FE	0.0002
60	NY_ADJ_NNTY_PC_CD	0.0002
88	SH_IMM_IDPT	0.0002
54	NV_MNF_TXTL_ZS_UN	0.0002
94	SP_DYN_IMRT_FE_IN	0.0002
52	NV_AGR_TOTL_ZS	0.0002
96	SP_DYN_IMRT_MA_IN	0.0002
152	TM_VAL_TRAN_ZS_WT	0.0002
153	TX_VAL_MANF_ZS_UN	0.0002
151	TM_VAL_SERV_CD_WT	0.0001
157	TX_VAL_SERV_CD_WT	0.0001
3	BM_GSR_CMCP_ZS	0.0001
133	SP_POP_65UP_TO_ZS	0.0001
2	BG_GSR_NFSV_GD_ZS	0.0001
135	SP_POP_7074_MA_5Y	0.0001
148	TM_VAL_MRCH_HI_ZS	0.0001
0	AG_CON_FERT_ZS	0.0001
1	AG_YLD_CREL_KG	0.0001
91	SP_DYN_AMRT_FE	0.0001
12	BN_TRF_CURR_CD	0.0001
59	NY_ADJ_DRES_GN_ZS	0.0001
64	NY_GDP_TOTL_RT_ZS	0.0001
71	SE_PRM_ENRL_TC_ZS	0.0001
5	BM_GSR_GNFS_CD	0.0001
84	SH_DYN_MORT	0.0001
15	BX_GSR_MRCH_CD	0.0001
40	NE_CON_GOVT_ZS	0.0001

46	NE_EXP_GNFS_KD	0.0001
28	EG_USE_PCAP_KG_OE	0.0001
22	EG_ELC_NUCL_ZS	0.0001
21	EG_ELC_LOSS_ZS	0.0001
16	BX_GSR_NFSV_CD	0.0001
108	SP_POP_0509_FE_5Y	0.0001
109	SP_POP_0509_MA_5Y	0.0001
130	SP_POP_6569_MA_5Y	0.0001
51	NE_TRD_GNFS_ZS	0.0001
4	BM_GSR_FCTY_CD	0.0000
48	NE_IMP_GNFS_CD	0.0000
93	SP_DYN_CBRT_IN	0.0000
95	SP_DYN_IMRT_IN	0.0000
106	SP_POP_0014_MA_ZS	0.0000
140	SP_POP_DPND	0.0000
107	SP_POP_0014_TO_ZS	0.0000
8	BM_GSR_TOTL_CD	0.0000

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0
116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0
13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.3



```
Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
```

```
warnings.warn(
63it [00:00, 104.50it/s]
```

```
Model with correlation >= 0.4:
Training+Validation R^2: 0.99912, RMSE: 0.56283
Testing R^2: 0.98578, RMSE: 2.45111
Mean cross-validation score: 0.98457
```

Feature	Importance

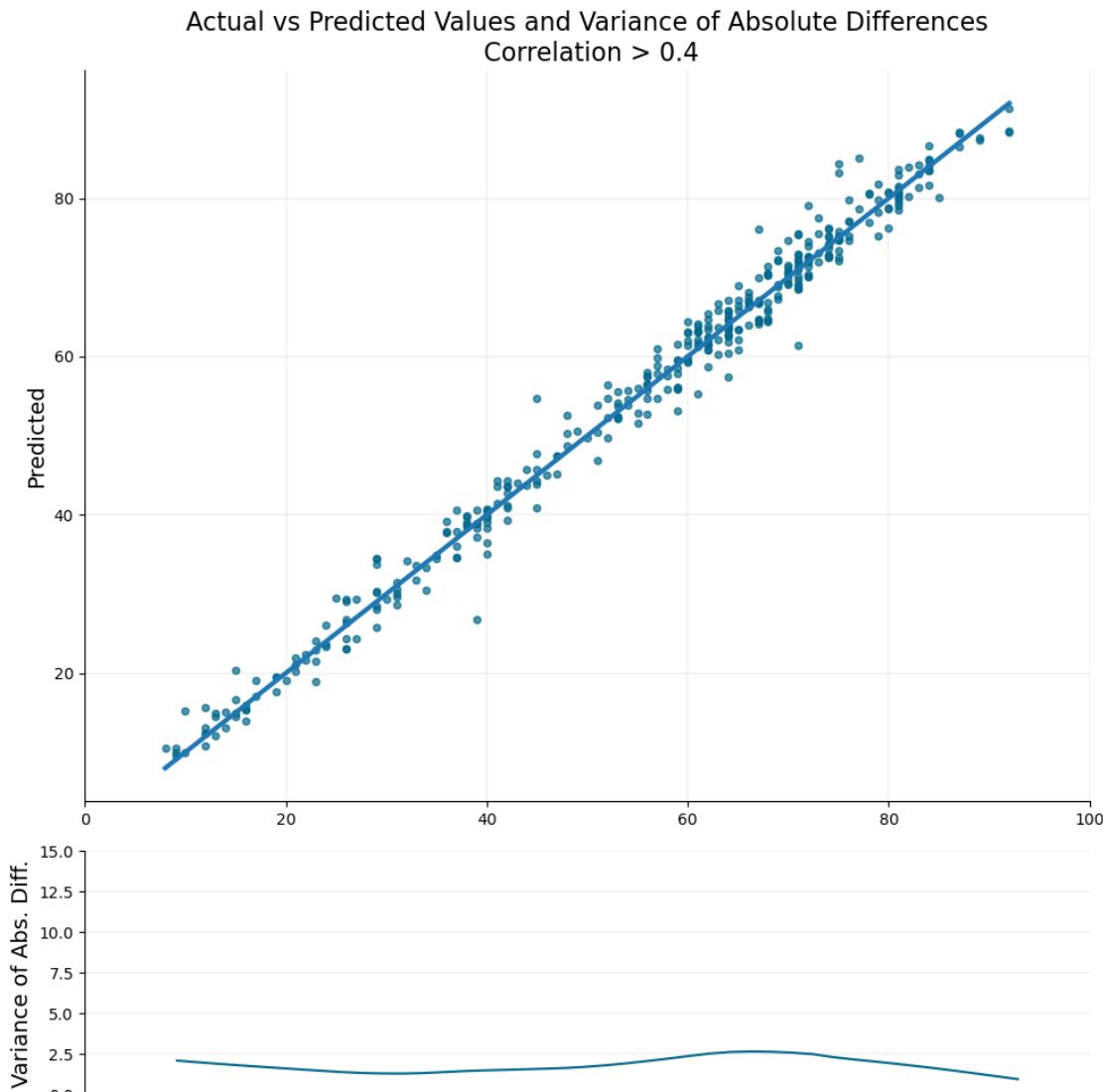
106	CPI_EST_avg	0.9543
108	CPI_EST_prev	0.0177
107	NY_GDP_PCAP_KD_rel	0.0016
87	SP_POP_65UP_FE_ZS	0.0010
88	SP_POP_65UP_MA_ZS	0.0008
98	SP_POP_DPND_YG	0.0007
43	SH_DYN_NMRT	0.0007
58	SP_DYN_T065_MA_ZS	0.0007
95	SP_POP_80UP_MA_5Y	0.0006
7	EG_USE_ELEC_KH_PC	0.0006
89	SP_POP_65UP_TO_ZS	0.0006
73	SP_POP_2024_FE_5Y	0.0005
32	SE_SEC_ENRR_FE	0.0005
81	SP_POP_5559_FE_5Y	0.0005
39	SH_DYN_2024	0.0005
13	IT_CEL_SETS_P2	0.0004
33	SE_SEC_ENRR_MA	0.0004
105	TX_VAL_SERV_CD_WT	0.0004
92	SP_POP_7579_FE_5Y	0.0004
53	SP_DYN_LE00_FE_IN	0.0004
84	SP_POP_6064_MA_5Y	0.0004
85	SP_POP_6569_FE_5Y	0.0004
77	SP_POP_4549_FE_5Y	0.0004
78	SP_POP_4549_MA_5Y	0.0004
80	SP_POP_5054_MA_5Y	0.0004
82	SP_POP_5559_MA_5Y	0.0004
76	SP_POP_4044_MA_5Y	0.0003
70	SP_POP_1564_FE_ZS	0.0003
57	SP_DYN_T065_FE_ZS	0.0003
50	SP_DYN_IMRT_FE_IN	0.0003
49	SP_DYN_CBRT_IN	0.0003
45	SH_IMM_MEAS	0.0003
41	SH_DYN_MORT_FE	0.0003
35	SG_LAW_INDX	0.0003
86	SP_POP_6569_MA_5Y	0.0003
54	SP_DYN_LE00_IN	0.0003
23	NY_ADJ_AEDU_GN_ZS	0.0003
34	SE_TER_ENRR	0.0003
22	NV_SRV_TOTL_ZS	0.0003
6	EG_ELC_RNWX_ZS	0.0003
12	FS_AST_PRVT_GD_ZS	0.0003
15	IT_NET_USER_ZS	0.0003
36	SH_DYN_0509	0.0002
83	SP_POP_6064_FE_5Y	0.0002
71	SP_POP_1564_MA_ZS	0.0002
37	SH_DYN_1014	0.0002
74	SP_POP_2024_MA_5Y	0.0002
75	SP_POP_4044_FE_5Y	0.0002

20	NE_IMP_GNFS_KD	0.0002
19	NE_EXP_GNFS_ZS	0.0002
79	SP_POP_5054_FE_5Y	0.0002
16	NE_CON_PRVT_PC_KD	0.0002
14	IT_MLT_MAIN_P2	0.0002
67	SP_POP_1014_MA_5Y	0.0002
9	EN_ATM_CO2E_PC	0.0002
90	SP_POP_7074_FE_5Y	0.0002
93	SP_POP_7579_MA_5Y	0.0002
96	SP_POP_DPND	0.0002
97	SP_POP_DPND_OL	0.0002
5	EG_ELC_LOSS_ZS	0.0002
99	SP_RUR_TOTL_ZS	0.0002
102	TM_VAL_OTHR_ZS_WT	0.0002
2	BM_GSR_NFSV_CD	0.0002
69	SP_POP_1519_MA_5Y	0.0002
72	SP_POP_1564_TO_ZS	0.0002
56	SP_DYN_TFRT_IN	0.0002
40	SH_DYN_MORT	0.0002
28	SE_PRE_ENRR	0.0002
46	SP_ADO_TFRT	0.0002
26	NY_GDP_PCAP_KD	0.0002
31	SE_SEC_ENRR	0.0002
55	SP_DYN_LE00_MA_IN	0.0002
25	NY_GDP_PCAP_CD	0.0002
48	SP_DYN_AMRT_MA	0.0002
59	SP_POP_0004_FE_5Y	0.0002
38	SH_DYN_1519	0.0002
64	SP_POP_0509_FE_5Y	0.0002
103	TM_VAL_SERV_CD_WT	0.0001
101	TM_VAL_MRCH_HI_ZS	0.0001
47	SP_DYN_AMRT_FE	0.0001
104	TM_VAL_TRAN_ZS_WT	0.0001
100	SP_URB_TOTL_IN_ZS	0.0001
4	BX_GSR_NFSV_CD	0.0001
30	SE_PRM_TCHR_FE_ZS	0.0001
3	BM_GSR_TRAN_ZS	0.0001
44	SH_IMM_IDPT	0.0001
29	SE_PRM_ENRL_TC_ZS	0.0001
8	EG_USE_PCAP_KG_OE	0.0001
91	SP_POP_7074_MA_5Y	0.0001
65	SP_POP_0509_MA_5Y	0.0001
11	FM_AST_PRVT_GD_ZS	0.0001
10	FD_AST_PRVT_GD_ZS	0.0001
66	SP_POP_1014_FE_5Y	0.0001
68	SP_POP_1519_FE_5Y	0.0001
63	SP_POP_0014_TO_ZS	0.0001
62	SP_POP_0014_MA_ZS	0.0001

61	SP_POP_0014_FE_ZS	0.0001
60	SP_POP_0004_MA_5Y	0.0001
24	NY_ADJ_NNTY_PC_CD	0.0001
21	NV_AGR_TOTL_ZS	0.0001
1	BM_GSR_FCTY_CD	0.0001
18	NE_EXP_GNFS_KD	0.0001
17	NE_CON_PRVT_ZS	0.0001
51	SP_DYN_IMRT_IN	0.0001
27	NY_GNP_PCAP_CD	0.0001
0	AG_YLD_CREL_KG	0.0001
42	SH_DYN_MORT_MA	0.0000
52	SP_DYN_IMRT_MA_IN	0.0000
94	SP_POP_80UP_FE_5Y	0.0000

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0
116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0
13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
```

```
    warnings.warn(
81it [00:00, 171.25it/s]
```

```
Model with correlation >= 0.5:
Training+Validation R^2: 0.99934, RMSE: 0.49024
Testing R^2: 0.98656, RMSE: 2.38294
Mean cross-validation score: 0.98482
```

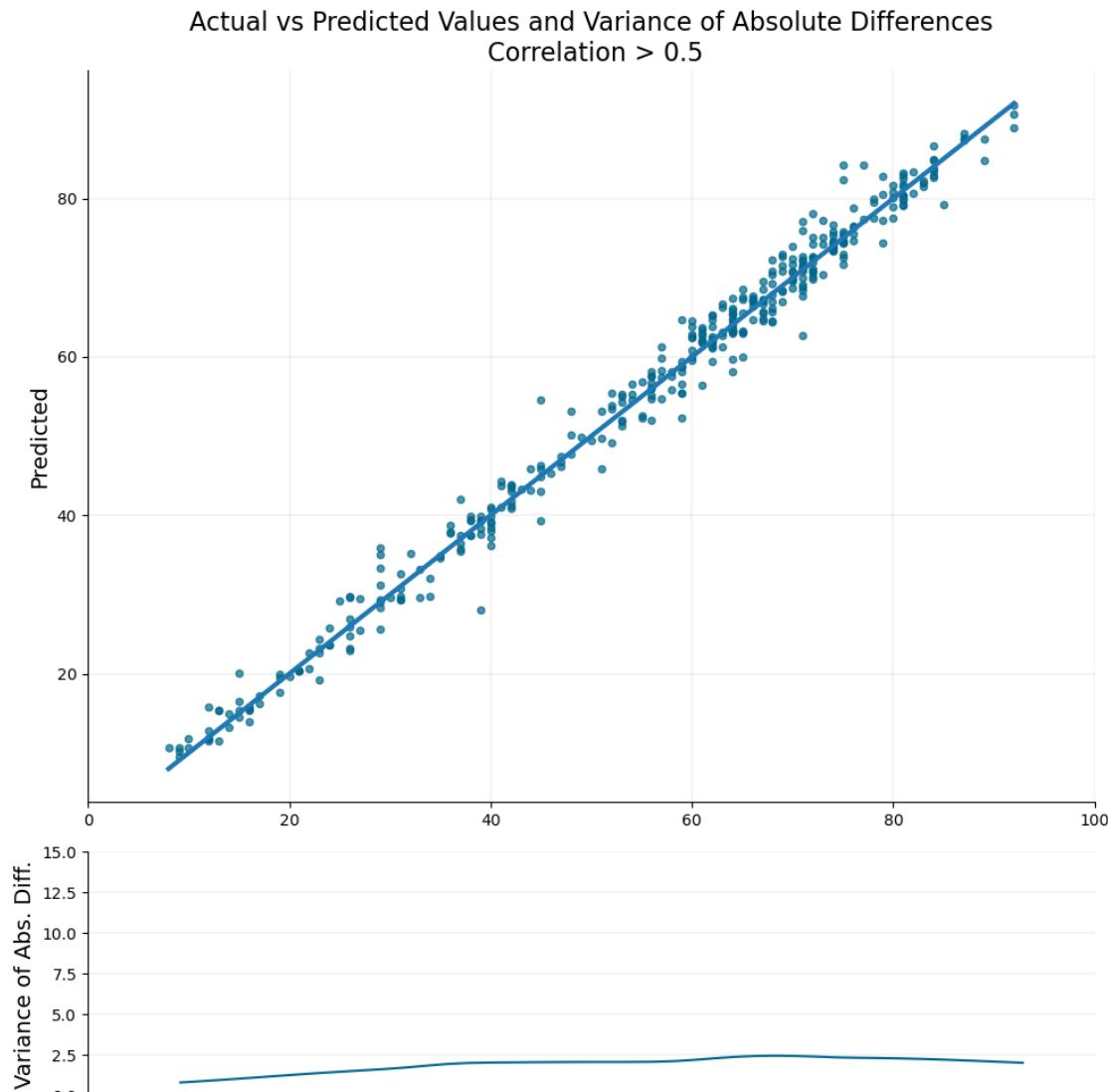
Feature	Importance

78	CPI_EST_avg	0.9244
80	CPI_EST_prev	0.0281
79	NY_GDP_PCAP_KD_rel	0.0048
64	SP_POP_65UP_FE_ZS	0.0014
23	SH_DYN_2024	0.0014
17	SE_SEC_ENRR_FE	0.0013
65	SP_POP_65UP_MA_ZS	0.0012
68	SP_POP_7074_MA_5Y	0.0012
41	SP_POP_0004_FE_5Y	0.0010
36	SP_DYN_LE00_IN	0.0010
24	SH_DYN_MORT	0.0010
69	SP_POP_7579_FE_5Y	0.0009
35	SP_DYN_LE00_FE_IN	0.0009
25	SH_DYN_MORT_FE	0.0009
58	SP_POP_5559_FE_5Y	0.0008
20	SG_LAW_INDX	0.0008
75	SP_RUR_TOTL_ZS	0.0008
14	SE_PRE_ENRR	0.0007
9	NV_SRV_TOTL_ZS	0.0007
52	SP_POP_2024_FE_5Y	0.0007
4	FS_AST_PRVT_GD_ZS	0.0007
31	SP_DYN_CBRT_IN	0.0007
27	SH_DYN_NMRT	0.0007
62	SP_POP_6569_FE_5Y	0.0007
56	SP_POP_5054_FE_5Y	0.0006
12	NY_GDP_PCAP_KD	0.0006
70	SP_POP_7579_MA_5Y	0.0006
22	SH_DYN_1519	0.0006
67	SP_POP_7074_FE_5Y	0.0006
39	SP_DYN_T065_FE_ZS	0.0006
48	SP_POP_1014_FE_5Y	0.0006
47	SP_POP_0509_MA_5Y	0.0006
57	SP_POP_5054_MA_5Y	0.0006
18	SE_SEC_ENRR_MA	0.0006
43	SP_POP_0014_FE_ZS	0.0005
71	SP_POP_80UP_FE_5Y	0.0005
54	SP_POP_4549_FE_5Y	0.0005
16	SE_SEC_ENRR	0.0005
3	FM_AST_PRVT_GD_ZS	0.0005
49	SP_POP_1014_MA_5Y	0.0005
46	SP_POP_0509_FE_5Y	0.0005
37	SP_DYN_LE00_MA_IN	0.0005
1	EG_USE_PCAP_KG_OE	0.0005
51	SP_POP_1519_MA_5Y	0.0005
6	IT_NET_USER_ZS	0.0005
60	SP_POP_6064_FE_5Y	0.0005
13	NY_GNP_PCAP_CD	0.0005
19	SE_TER_ENRR	0.0005

21	SH_DYN_1014	0.0005
50	SP_POP_1519_FE_5Y	0.0005
72	SP_POP_80UP_MA_5Y	0.0004
0	EG_USE_ELEC_KH_PC	0.0004
45	SP_POP_0014_TO_ZS	0.0004
30	SP_DYN_AMRT_MA	0.0004
2	FD_AST_PRVT_GD_ZS	0.0004
5	IT_MLT_MAIN_P2	0.0004
7	NE_CON_PRVT_PC_KD	0.0004
11	NY_GDP_PCAP_CD	0.0004
28	SP_ADO_TFRT	0.0004
29	SP_DYN_AMRT_FE	0.0004
40	SP_DYN_TO65_MA_ZS	0.0004
38	SP_DYN_TFRT_IN	0.0004
61	SP_POP_6064_MA_5Y	0.0003
55	SP_POP_4549_MA_5Y	0.0003
77	TM_VAL_MRCH_HI_ZS	0.0003
42	SP_POP_0004_MA_5Y	0.0003
73	SP_POP_DPND_OL	0.0003
8	NV_AGR_TOTL_ZS	0.0003
32	SP_DYN_IMRT_FE_IN	0.0003
66	SP_POP_65UP_TO_ZS	0.0003
53	SP_POP_2024_MA_5Y	0.0003
59	SP_POP_5559_MA_5Y	0.0003
63	SP_POP_6569_MA_5Y	0.0003
15	SE_PRM_ENRL_TC_ZS	0.0002
74	SP_POP_DPND_YG	0.0002
33	SP_DYN_IMRT_IN	0.0002
44	SP_POP_0014_MA_ZS	0.0002
10	NY_ADJ_NNTY_PC_CD	0.0002
26	SH_DYN_MORT_MA	0.0000
34	SP_DYN_IMRT_MA_IN	0.0000
76	SP_URB_TOTL_IN_ZS	0.0000

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0
116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0
13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]



```
Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is deprecated for better compatibility with scikit-learn, use `callbacks` in constructor or `set_params` instead.
```

```
    warnings.warn(  
97it [00:00, 215.43it/s]
```

```
Model with correlation >= 0.6:  
Training+Validation R^2: 0.99935, RMSE: 0.4867  
Testing R^2: 0.98628, RMSE: 2.40711  
Mean cross-validation score: 0.98431
```

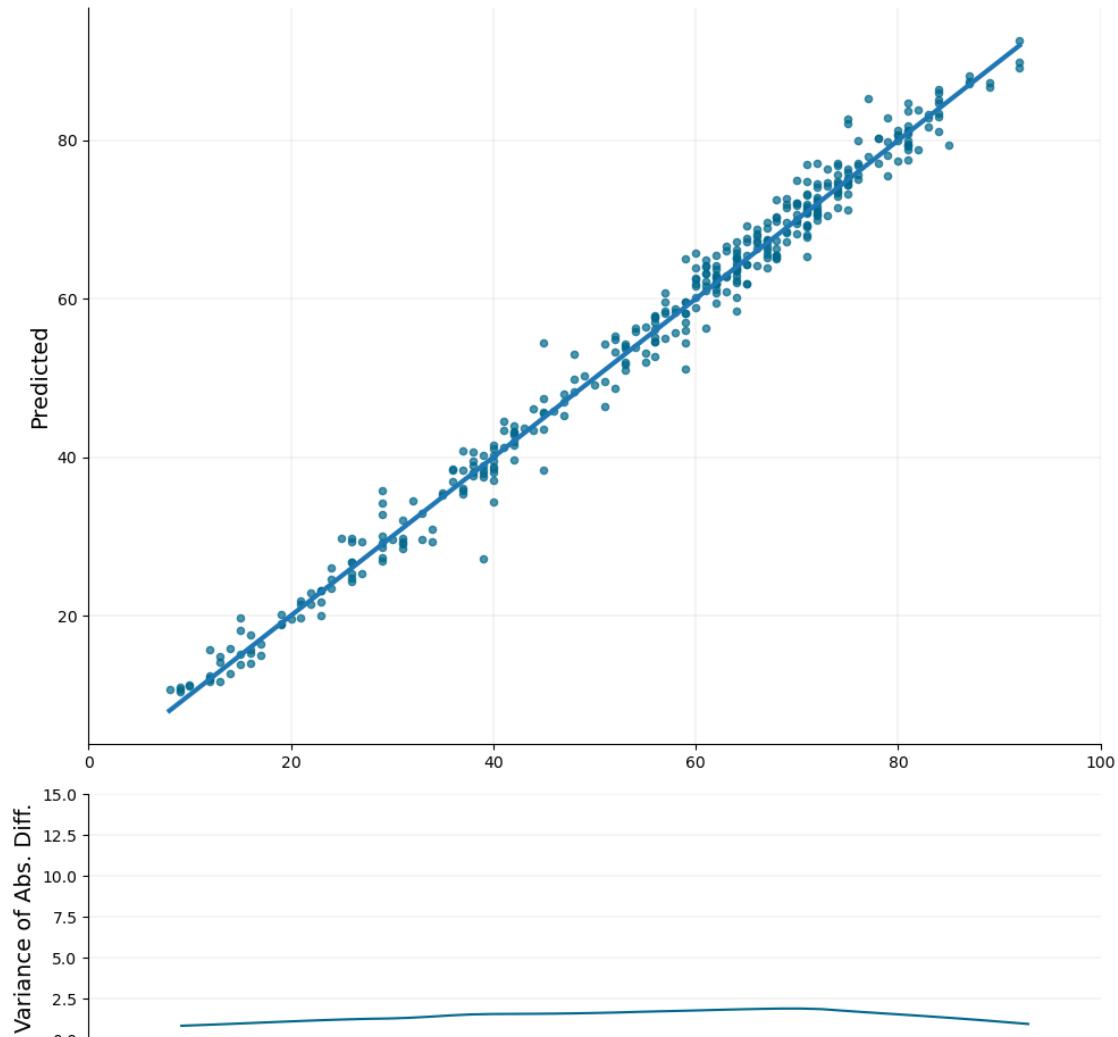
	Feature	Importance
44	CPI_EST_avg	0.9374
46	CPI_EST_prev	0.0273
45	NY_GDP_PCAP_KD_rel	0.0024
32	SP_POP_6569_FE_5Y	0.0015
9	SE_SEC_ENRR_FE	0.0014
35	SP_POP_65UP_MA_ZS	0.0014
41	SP_POP_80UP_MA_5Y	0.0013
14	SP_DYN_LE00_FE_IN	0.0013
16	SP_DYN_LE00_MA_IN	0.0012
7	NY_GNP_PCAP_CD	0.0011
40	SP_POP_80UP_FE_5Y	0.0011
10	SE_SEC_ENRR_MA	0.0011
43	SP_POP_DPNP_YG	0.0010
15	SP_DYN_LE00_IN	0.0009
34	SP_POP_65UP_FE_ZS	0.0009
38	SP_POP_7074_MA_5Y	0.0009
17	SP_DYN_T065_MA_ZS	0.0008
26	SP_POP_1014_MA_5Y	0.0008
4	NY_ADJ_NNTY_PC_CD	0.0008
28	SP_POP_1519_MA_5Y	0.0008
37	SP_POP_7074_FE_5Y	0.0007
27	SP_POP_1519_FE_5Y	0.0007
0	EG_USE_ELEC_KH_PC	0.0007
13	SP_DYN_CBRT_IN	0.0007
22	SP_POP_0014_TO_ZS	0.0007
11	SH_DYN_NMRT	0.0007
2	IT_NET_USER_ZS	0.0007
18	SP_POP_0004_FE_5Y	0.0007
25	SP_POP_1014_FE_5Y	0.0006
39	SP_POP_7579_MA_5Y	0.0006
29	SP_POP_5054_MA_5Y	0.0006
30	SP_POP_5559_MA_5Y	0.0006
31	SP_POP_6064_MA_5Y	0.0006
6	NY_GDP_PCAP_KD	0.0006
12	SP_DYN_AMRT_MA	0.0006
5	NY_GDP_PCAP_CD	0.0005
19	SP_POP_0004_MA_5Y	0.0005
3	NE_CON_PRVT_PC_KD	0.0005
20	SP_POP_0014_FE_ZS	0.0005
21	SP_POP_0014_MA_ZS	0.0005
8	SE_SEC_ENRR	0.0005
1	IT_MLT_MAIN_P2	0.0004
24	SP_POP_0509_MA_5Y	0.0004
33	SP_POP_6569_MA_5Y	0.0004
23	SP_POP_0509_FE_5Y	0.0004
42	SP_POP_DPNP_DL	0.0003

36 SP_POP_65UP_TO_ZS 0.0003

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0
116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0
13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.6



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
37it [00:00, 357.92it/s]
```

```
Model with correlation >= 0.7:
Training+Validation R^2: 0.99402, RMSE: 1.47055
Testing R^2: 0.98509, RMSE: 2.50966
Mean cross-validation score: 0.9836
```

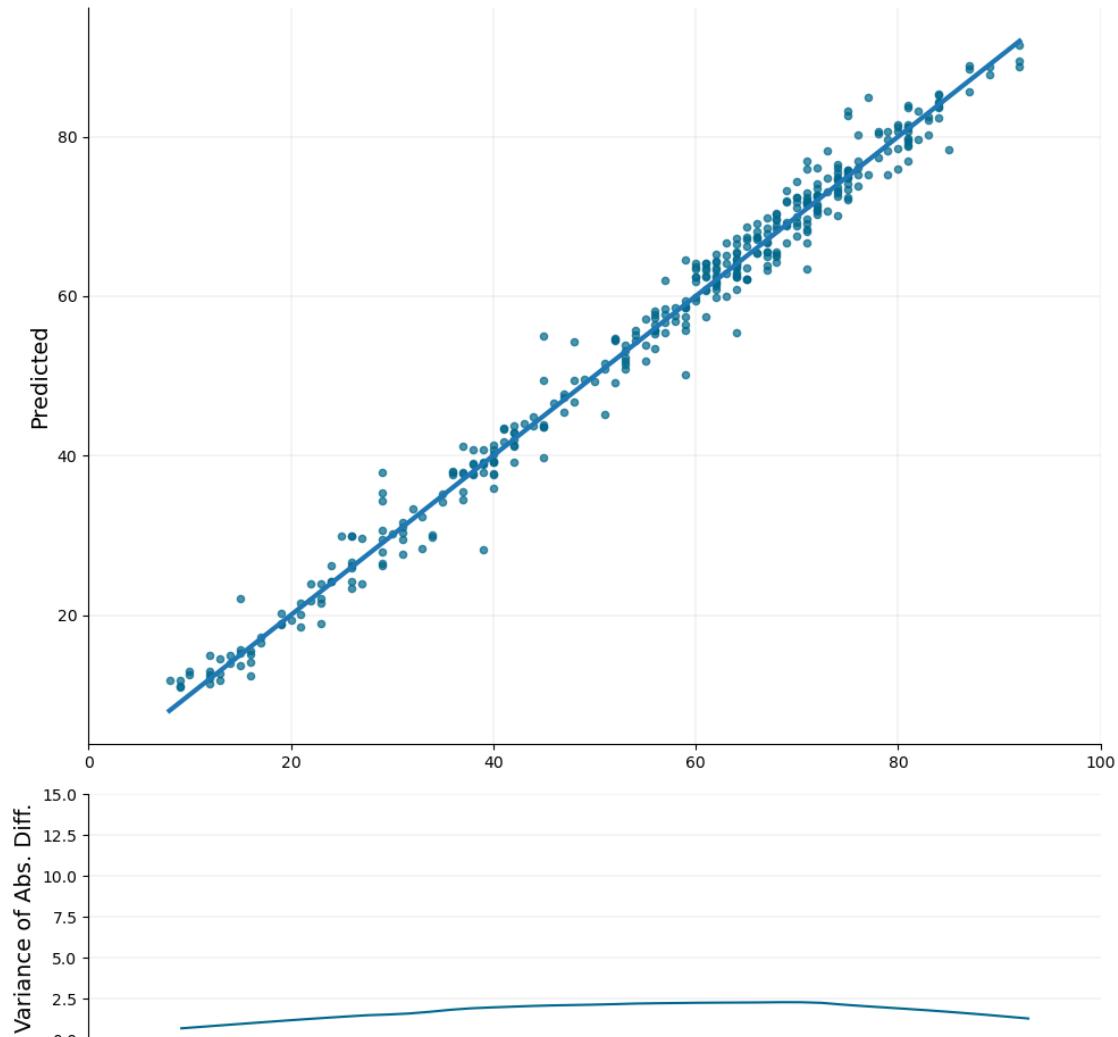
Feature	Importance

7	CPI_EST_avg	0.9648
9	CPI_EST_prev	0.0267
5	NY_GNP_PCAP_CD	0.0015
8	NY_GDP_PCAP_KD_rel	0.0014
4	NY_GDP_PCAP_KD	0.0013
6	SP_DYN_LEOO_MA_IN	0.0010
0	IT_NET_USER_ZS	0.0009
1	NE_CON_PRVT_PC_KD	0.0008
2	NY_ADJ_NNTY_PC_CD	0.0008
3	NY_GDP_PCAP_CD	0.0008

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0
116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0
13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]

Actual vs Predicted Values and Variance of Absolute Differences
Correlation > 0.7



```
0it [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
    warnings.warn(
28it [00:00, 143.35it/s]
```

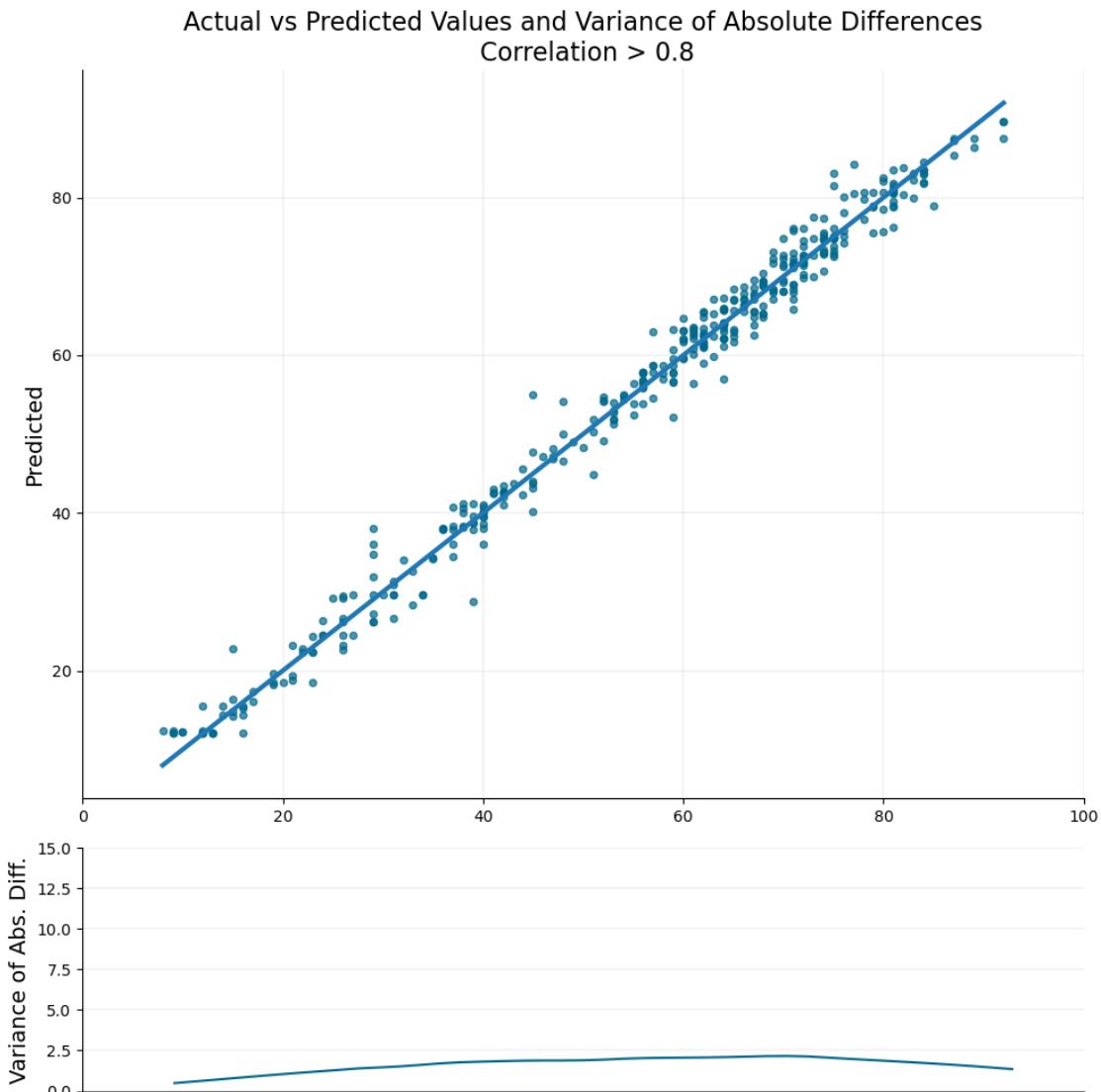
```
Model with correlation >= 0.8:
Training+Validation R^2: 0.99142, RMSE: 1.76257
Testing R^2: 0.98562, RMSE: 2.46421
Mean cross-validation score: 0.98382
```

Feature	Importance

```
3      CPI_EST_avg      0.9690
4      CPI_EST_prev      0.0270
2      NY_GNP_PCAP_CD    0.0019
1  NY_ADJ_NNTY_PC_CD   0.0011
0  NE_CON_PRVT_PC_KD   0.0010
```

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0
116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0
13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]



```
[242]: ### TESTS FOR SELECTED VARIABLES
```

```
# Set the style to 'default' to make the background white
style.use('default')

# Manually selected variables
vars_selected = ['CPI_EST_avg', 'NY_GDP_PCAP_KD', 'NY_GDP_PCAP_CD', ↴
                  'CPI_EST_prev']

if 'CPI_EST_prev' in vars_selected:
    vars_selected.remove('CPI_EST_prev')
```

```

results = build_and_evaluate_model(df, 'CPI_EST', vars_selected, n_leads=2)

print(f"\n Model with selected variables:")
print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
print(f"Mean cross-validation score: {results[4]}\n")
print(results[5]) # Feature importances
print('\n')
print(results[9][['country', 'year', 'CPI_EST']])

# Generate the graphs
y_test = results[8]
y_test_pred = results[7]

# Create a scatter plot for the actual vs predicted values
abs_diffs = np.abs(y_test - y_test_pred)

# Create a DataFrame with the actual values and absolute differences
df_plot = pd.DataFrame({'Actual': y_test, 'AbsDifference': abs_diffs})

# Define the bins for the actual values
bins = np.linspace(0, 100, 50)

# Calculate the mid-points of the bins
bin_midpoints = bins[:-1] + np.diff(bins) / 2

# Create a new column for the binned actual values
df_plot['ActualBin'] = pd.cut(df_plot['Actual'], bins, labels=bin_midpoints)

# Group by the binned actual values and calculate the variance of the absolute differences for each group
var_abs_diffs = df_plot.groupby('ActualBin')['AbsDifference'].var()

# Create a scatter plot for the actual vs predicted values
fig = plt.figure(figsize=(10, 10))
gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1])
ax0 = plt.subplot(gs[0])
ax0.scatter(y_test, y_test_pred, alpha=0.7, color='#00688B', s=20)
ax0.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], lw=3)
ax0.set_xlim([0, 100])
ax0.set_ylabel('Predicted', fontsize=14)
ax0.set_title('Actual vs Predicted Values and Variance of Absolute Differences', fontsize=16)

# Hide the right and top spines
ax0.spines['right'].set_visible(False)
ax0.spines['top'].set_visible(False)

```

```

# Only show ticks on the left and bottom spines
ax0.yaxis.set_ticks_position('left')
ax0.xaxis.set_ticks_position('bottom')

ax0.grid(True, color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

# Create a line plot for the binned actual values vs variance of the absolute differences
ax1 = plt.subplot(gs[1])

# Apply LOESS to smooth the variance curve
smoothed = lowess(var_abs_diffs, var_abs_diffs.index, frac=0.5)
index, data = zip(*smoothed)
ax1.plot(index, data, color='#00688B')

ax1.set_xlim([0, 15])
ax1.set_ylim([0, 100])
ax1.set_ylabel('Variance of Abs. Diff.', fontsize=14)

# Hide the right and top spines
ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)
ax1.set_xticklabels([])
ax1.set_xticks([])

# Only show ticks on the left and bottom spines
ax1.yaxis.set_ticks_position('left')
ax1.xaxis.set_ticks_position('bottom')

ax1.grid(axis='y', color='grey', linestyle='--', linewidth=0.25, alpha=0.5)

plt.tight_layout()
plt.show()

```

Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is deprecated for better compatibility with scikit-learn, use `callbacks` in constructor or `set_params` instead.

```

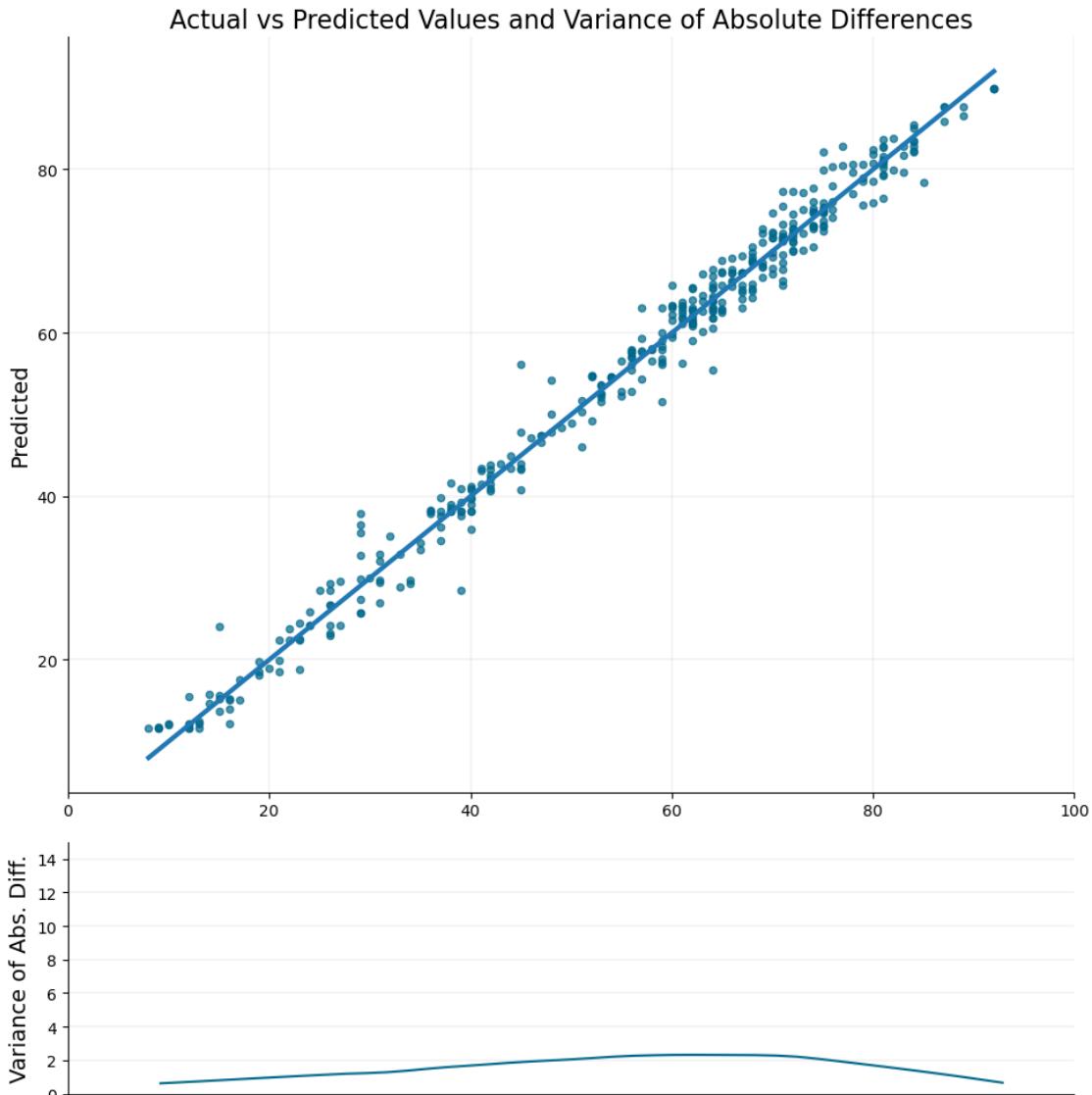
    warnings.warn(
31it [00:00, 642.98it/s]
```

Model with selected variables:
Training+Validation R^2: 0.99134, RMSE: 1.77002
Testing R^2: 0.98519, RMSE: 2.50144
Mean cross-validation score: 0.98423

	Feature	Importance
0	CPI_EST_avg	0.9655
3	CPI_EST_prev	0.0311
1	NY_GDP_PCAP_KD	0.0017
2	NY_GDP_PCAP_CD	0.0016

	country	year	CPI_EST
115	United Arab Emirates	2012	32.0
116	United Arab Emirates	2013	31.0
117	United Arab Emirates	2014	30.0
118	United Arab Emirates	2015	30.0
119	United Arab Emirates	2016	34.0
...
13729	Zimbabwe	2018	78.0
13730	Zimbabwe	2019	76.0
13731	Zimbabwe	2020	76.0
13732	Zimbabwe	2021	77.0
13733	Zimbabwe	2022	77.0

[1919 rows x 3 columns]



```
[243]: country_codes = {country.alpha_2: {'alpha-3': country.alpha_3, 'numeric': country.numeric} for country in pycountry.countries}

# Now you can use this dictionary in your code
df['iso3c'] = df['iso2c'].map(lambda x: country_codes.get(x, {}).get('alpha-3'))
df['iso3n'] = df['iso2c'].map(lambda x: country_codes.get(x, {}).get('numeric'))

[244]: ## TAKES c.4 MINUTES TO RUN
### TESTS FOR ALL VARIABLES
# Get the unique country codes
countries = df['iso2c'].unique()
```

```

# Create a new dataframe to store the results
df_new = pd.DataFrame()

# Get the variables for the model
vars_full = df.columns.tolist()
vars_full.remove('iso3c')
vars_full.remove('iso3n')
vars_full.remove('year')

# Remove 'CC_EST_prev' from vars_full if it's already included
if 'CPI_EST_prev' in vars_full:
    vars_full.remove('CPI_EST_prev')

# Train the model before getting the feature names
results = build_and_evaluate_model(df, 'CPI_EST', vars_full)

# Get the trained model from the results
model = results[10]

# Get the feature names the model was trained on
model_features = model.get_booster().feature_names

# Iterate over each country
for country in tqdm(countries):
    # Get the data for the current country and create a copy of it
    df_country = df[df['iso2c'] == country].copy()

    # Add a new feature for the previous year's CC_EST value
    df_country['CPI_EST_prev'] = df_country.groupby('iso2c')['CPI_EST'].shift()

    # Sort the data in descending order of the year
    df_country = df_country.sort_values('year', ascending=False)

    # Get the latest year in the data
    latest_year = df_country['year'].max()

    # Skip the current country if all its 'year' values are NaN
    if np.isnan(latest_year):
        continue

    latest_year = int(latest_year)

    # Iterate from the latest year to 1970
    for year in range(latest_year, 1959, -1):
        # Check if CC_EST for the current year is NaN
        if df_country.loc[df_country['year'] == year, 'CPI_EST'].isna().any():

```

```

# If it is NaN, use the model to predict CC_EST for that year based on other values
X = df_country.loc[df_country['year'] == year, model_features]

# Forward fill missing values in X
X = X.fillna()

y_pred = model.predict(X)

# Update the data with the predicted value
df_country.loc[df_country['year'] == year, 'CPI_EST'] = y_pred

# Append the data for the current country to the new dataframe
df_new = pd.concat([df_new, df_country])

```

Oit [00:00, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is deprecated for better compatibility with scikit-learn, use `callbacks` in constructor or `set_params` instead.

```

warnings.warn(
52it [00:02, 24.37it/s]
100%| 179/179 [03:11<00:00, 1.07s/it]

```

[245]: # print the results of the build and evaluate model:

```

print(f"Training+Validation R^2: {results[0]}, RMSE: {results[2]}")
print(f"Testing R^2: {results[1]}, RMSE: {results[3]}")
print(f"Mean cross-validation score: {results[4]}\n")
print(results[5]) # Feature importances

```

Training+Validation R^2: 0.99942, RMSE: 0.45797
 Testing R^2: 0.98507, RMSE: 2.51129
 Mean cross-validation score: 0.98424

	Feature	Importance
518	CPI_EST_avg	0.8700
521	CPI_EST_prev	0.0307
519	NY_GDP_PCAP_KD_rel	0.0049
415	SP_POP_1564_MA_IN	0.0025
119	EN_ATM_CO2E_GF_KT	0.0023
..
256	NV_IND_TOTL_CD	0.0000
248	NV_AGR_TOTL_KN	0.0000
246	NV_AGR_TOTL_KD	0.0000
243	NE_TRD_GNFS_ZS	0.0000
261	NV_IND_TOTL_ZS	0.0000

[522 rows x 2 columns]

```
[248]: # Give a relative rank of CC_EST for each country in each year
df_new['CPI_EST_rank'] = df_new.groupby('year')['CPI_EST'].rank(pct=True)

# Give a rank of CC_EST for each country in each year as integers
df_new['CPI_EST_rank_int'] = df_new.groupby('year')['CPI_EST'].
    rank(method='dense', ascending=True)
```

	country	year	CPI_EST	CPI_EST_rank	CPI_EST_rank_int
9071	Nigeria	2022	76.000000	0.849162	57.0
9070	Nigeria	2021	76.000000	0.860335	56.0
9069	Nigeria	2020	75.000000	0.846369	54.0
9068	Nigeria	2019	74.000000	0.826816	53.0
9067	Nigeria	2018	73.000000	0.810056	57.0
9066	Nigeria	2017	73.000000	0.826816	52.0
9065	Nigeria	2016	72.000000	0.784916	60.0
9064	Nigeria	2015	74.000000	0.821229	65.0
9063	Nigeria	2014	73.000000	0.790503	59.0
9062	Nigeria	2013	75.000000	0.829609	60.0
9061	Nigeria	2012	73.000000	0.798883	61.0
9060	Nigeria	2011	73.476013	0.798883	143.0
9059	Nigeria	2010	74.291389	0.826816	148.0
9058	Nigeria	2009	75.248253	0.860335	154.0
9057	Nigeria	2008	74.567657	0.826816	148.0
9056	Nigeria	2007	74.602295	0.843575	151.0
9055	Nigeria	2006	71.902298	0.759777	136.0
9054	Nigeria	2005	72.747765	0.782123	140.0
9053	Nigeria	2004	72.019722	0.770950	138.0
9052	Nigeria	2003	71.641037	0.759777	136.0
9051	Nigeria	2002	74.344414	0.832402	149.0
9050	Nigeria	2001	72.654518	0.776536	139.0
9049	Nigeria	2000	72.278717	0.770950	138.0
9048	Nigeria	1999	74.733322	0.854749	153.0
9047	Nigeria	1998	72.792358	0.770950	138.0
9046	Nigeria	1997	74.552689	0.837989	150.0
9045	Nigeria	1996	74.114647	0.821229	147.0
9044	Nigeria	1995	74.162308	0.826816	148.0
9043	Nigeria	1994	74.374237	0.843575	151.0
9042	Nigeria	1993	74.356064	0.843575	151.0
9041	Nigeria	1992	74.278458	0.837989	150.0
9040	Nigeria	1991	74.719826	0.843575	151.0
9039	Nigeria	1990	72.311844	0.765363	137.0
9038	Nigeria	1989	74.460823	0.837989	150.0
9037	Nigeria	1988	74.888191	0.860335	154.0
9036	Nigeria	1987	74.582520	0.843575	151.0
9035	Nigeria	1986	74.401184	0.849162	152.0
9034	Nigeria	1985	71.701843	0.765363	137.0
9033	Nigeria	1984	73.724030	0.815642	146.0

9032	Nigeria	1983	74.154221	0.826816	148.0
9031	Nigeria	1982	74.754204	0.849162	152.0
9030	Nigeria	1981	74.558685	0.837989	150.0
9029	Nigeria	1980	74.477386	0.826816	148.0
9028	Nigeria	1979	74.035255	0.804469	144.0
9027	Nigeria	1978	74.517899	0.826816	148.0
9026	Nigeria	1977	74.475105	0.832402	149.0
9025	Nigeria	1976	73.852783	0.815642	146.0
9024	Nigeria	1975	74.463051	0.837989	150.0
9023	Nigeria	1974	74.454048	0.826816	148.0
9022	Nigeria	1973	73.920937	0.821229	147.0
9021	Nigeria	1972	74.052147	0.837989	150.0
9020	Nigeria	1971	74.219315	0.837989	150.0
9019	Nigeria	1970	73.885963	0.826816	148.0
9018	Nigeria	1969	74.599617	0.832402	149.0
9017	Nigeria	1968	73.677185	0.798883	143.0
9016	Nigeria	1967	73.968018	0.804469	144.0
9015	Nigeria	1966	74.454102	0.843575	151.0
9014	Nigeria	1965	73.626549	0.804469	144.0
9013	Nigeria	1964	73.878983	0.815642	146.0
9012	Nigeria	1963	73.503151	0.798883	143.0
9011	Nigeria	1962	74.140282	0.826816	148.0
9010	Nigeria	1961	73.751740	0.793296	142.0
9009	Nigeria	1960	73.526436	0.776536	139.0

[249]: # Append the new data of iso2c, year, CC_EST, CC_EST_rank, and CC_EST_rank_int
 ↪to a new csv file

```
df_new[['iso2c', 'year', 'CPI_EST', 'CPI_EST_rank', 'CPI_EST_rank_int']].  

    ↪to_csv('backcasted_corruption_data.csv', index=False)
```

[250]: # Print the mean, median, and range of predicted values for CC_EST in 1960,
 ↪with the countries that have the highest and lowest values

Get the mean, median, and range of predicted values for CC_EST in 1960

```
mean_1960 = round(df_new[df_new['year'] == 1960]['CPI_EST'].mean(), 3)
median_1960 = round(df_new[df_new['year'] == 1960]['CPI_EST'].median(), 3)
range_1960 = round(df_new[df_new['year'] == 1960]['CPI_EST'].max() -  

    ↪df_new[df_new['year'] == 1960]['CPI_EST'].min(), 3)
```

Get the countries with the highest and lowest predicted values for CC_EST in
 ↪1960

```
highest_1960 = df_new[df_new['year'] == 1960].sort_values('CPI_EST',  

    ↪ascending=False).iloc[0]['country']
lowest_1960 = df_new[df_new['year'] == 1960].sort_values('CPI_EST').  

    ↪iloc[0]['country']

print(f"The mean predicted value for CPI_EST in 1960 is {mean_1960}")
print(f"The median predicted value for CPI_EST in 1960 is {median_1960}")
```

```

print(f"The range of predicted values for CPI_EST in 1960 is {range_1960}")
print(f"The country with the highest predicted value for CPI_EST in 1960 is"
    ↪{highest_1960} with a value of {df_new[df_new['year'] == 1960].
    ↪sort_values('CPI_EST', ascending=False).iloc[0]['CPI_EST']}")
print(f"The country with the lowest predicted value for CPI_EST in 1960 is"
    ↪{lowest_1960} with a value of {df_new[df_new['year'] == 1960].
    ↪sort_values('CPI_EST').iloc[0]['CPI_EST']}")

```

The mean predicted value for CPI_EST in 1960 is 57.69
The median predicted value for CPI_EST in 1960 is 62.035
The range of predicted values for CPI_EST in 1960 is 67.717
The country with the highest predicted value for CPI_EST in 1960 is South Sudan
with a value of 84.58109283447266
The country with the lowest predicted value for CPI_EST in 1960 is New Zealand
with a value of 16.864179611206055

```

[251]: # Print the country that has gotten worse by the most and the country that has
    ↪improved by the most
# Calculate the change in CC_EST for each country between 1960 and 2022
df_change = df_new.pivot(index='iso2c', columns='year', values='CPI_EST')
df_change['change'] = df_change[2022] - df_change[1960]

# Get the country that has gotten worse by the most and the country that has
    ↪improved by the most
most_improved = df_change['change'].idxmax()
most_worsened = df_change['change'].idxmin()

print(f"The country that has improved the most between 1970 and 2022 is"
    ↪{df_new[df_new['iso2c'] == most_improved].iloc[0]['country']} with a change
    ↪of {df_change.loc[most_improved, 'change']}")
print(f"The country that has worsened the most between 1970 and 2022 is"
    ↪{df_new[df_new['iso2c'] == most_worsened].iloc[0]['country']} with a change
    ↪of {df_change.loc[most_worsened, 'change']})

# List worst to best
# Create a mapping of country codes to country names
country_mapping = df_new.set_index('iso2c')['country'].drop_duplicates()

# Map the country codes in df_change to country names
df_change['country'] = df_change.index.map(country_mapping)
# Sort df_change by 'change' in descending order and display the top 10
df_change.sort_values('change', ascending=True)[['country', 'change']].head(10)

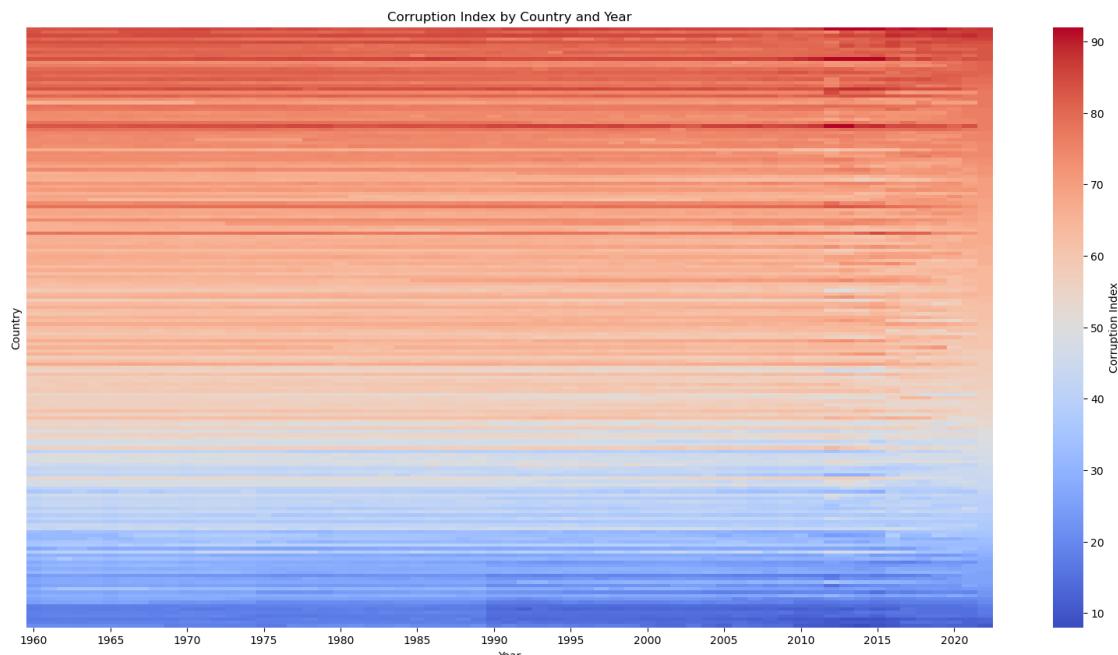
```

The country that has improved the most between 1970 and 2022 is Azerbaijan with
a change of 11.751197814941406
The country that has worsened the most between 1970 and 2022 is Angola with a
change of -11.129158020019531

```
[251]: year      country      change
iso2c
AO          Angola -11.129158
DK          Denmark -10.155420
IT          Italy   -9.794922
VN          Viet Nam -9.566093
UZ          Uzbekistan -9.026062
SC          Seychelles -8.288765
GR          Greece  -7.994835
TL          Timor-Leste -7.528000
AM          Armenia -7.403732
KR          Korea, Rep. -7.050106
```

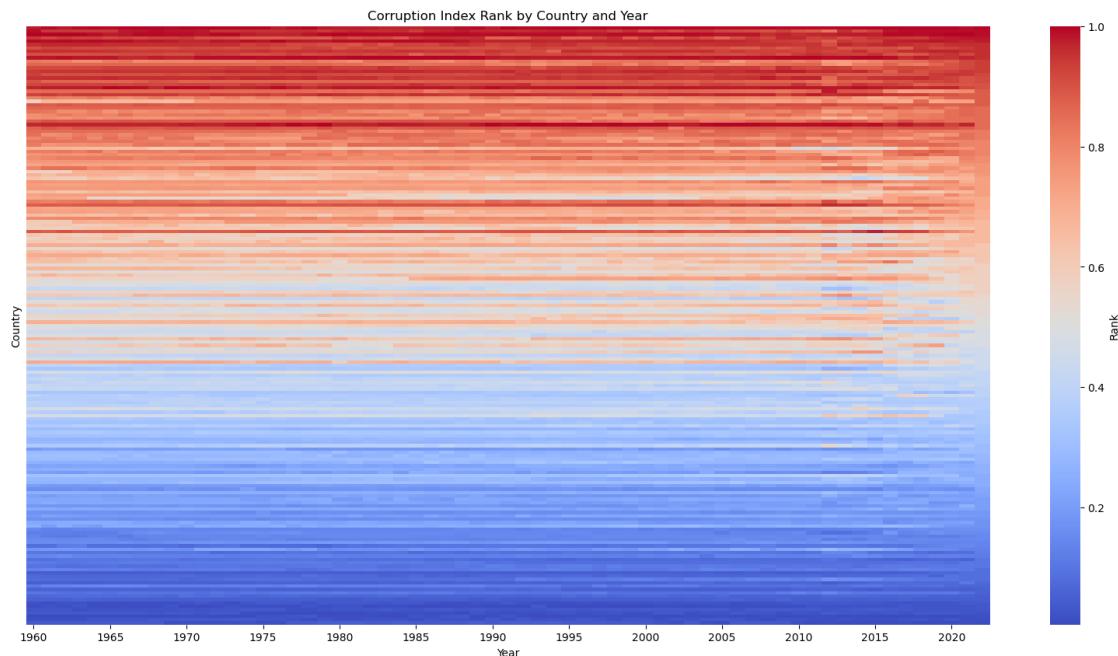
```
[252]: # Graphically illustrate the generale moves of country and corruption levels over time
# Create a pivot table of the data for the heatmap
df_heatmap = df_new.pivot(index='country', columns='year', values='CPI_EST')
df_heatmap = df_heatmap.sort_values(by=2022, ascending=False)

# Create a heatmap of the data, dont show country labels, and 1970 to 2022
plt.figure(figsize=(20, 10))
sns.heatmap(df_heatmap, cmap='coolwarm', cbar_kws={'label': 'Corruption Index'}, xticklabels=5, yticklabels=False)
plt.xlim(0, 63)
plt.title('Corruption Index by Country and Year')
plt.xlabel('Year')
plt.ylabel('Country')
plt.show()
```



```
[253]: # Plot changes in country ranks over time
# Create a pivot table of the data for the heatmap
df_heatmap = df_new.pivot(index='country', columns='year', u
                           ↴values='CPI_EST_rank')
df_heatmap = df_heatmap.sort_values(by=2022, ascending=False)

# Create a heatmap of the data, don't show country labels, and 1970 to 2022, u
                           ↴ranking from lowest to highest in 2022
plt.figure(figsize=(20, 10))
sns.heatmap(df_heatmap, cmap='coolwarm', cbar_kws={'label': 'Rank'}, u
                           ↴xticklabels=5, yticklabels=False)
plt.xlim(0, 63)
plt.title('Corruption Index Rank by Country and Year')
plt.xlabel('Year')
plt.ylabel('Country')
plt.show()
```



```
[254]: style.use('default')

# Plot graph of CC_EST_rank_int for the top 10 countries with the highest rank u
                           ↴in 2022
# Get the 40 countries around 150 countries with the highest rank in 2022
```

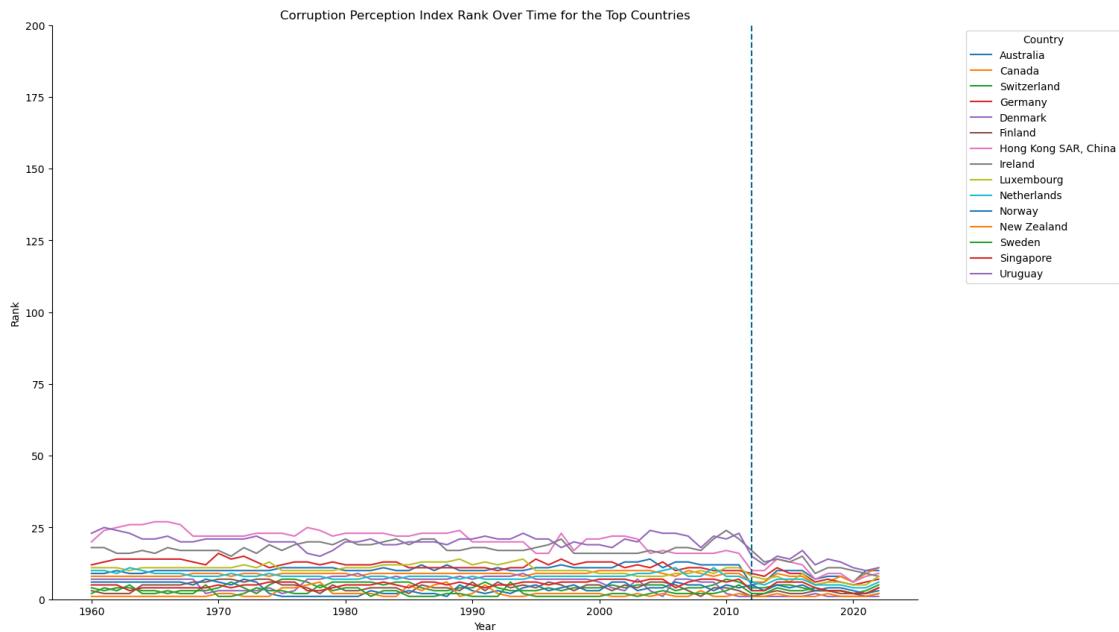
```

top_countries = df_heatmap[2022].sort_values(ascending=True).head(15).index

# Filter the data for the top 10 countries
df_top = df_new[df_new['country'].isin(top_countries)]

# Create a line plot of the rank over time for the top 10 countries
plt.figure(figsize=(15, 10))
sns.lineplot(data=df_top, x='year', y='CPI_EST_rank_int', hue='country', palette='tab10')
plt.axvline(x=2012, color="#00688B", linestyle='--')
plt.title('Corruption Perception Index Rank Over Time for the Top Countries')
plt.ylim(0, 200)
plt.xlabel('Year')
plt.ylabel('Rank')
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
sns.despine()
plt.show()

```



```

[255]: style.use('default')

# Plot graph of CC_EST_rank_int for the top 10 countries with the highest rank
# in 2022
# Get the top 10 countries with the highest rank in 2022
top_countries = df_heatmap[2022].sort_values(ascending=True).head(15).index

# Filter the data for the top 10 countries

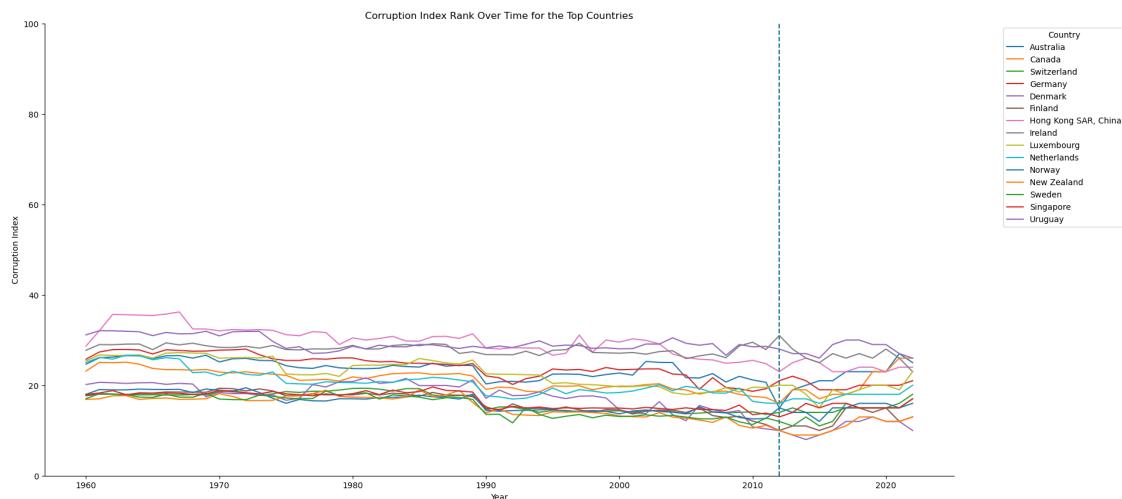
```

```

df_top = df_new[df_new['country'].isin(top_countries)]

# Create a line plot of the rank over time for the top 10 countries
plt.figure(figsize=(20, 10))
plt.axvline(x=2012, color='#00688B', linestyle='--')
sns.lineplot(data=df_top, x='year', y='CPI_EST', hue='country', palette='tab10')
plt.ylim(0, 100)
plt.title('Corruption Index Rank Over Time for the Top Countries')
plt.xlabel('Year')
plt.ylabel('Corruption Index')
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
sns.despine()
plt.show()

```

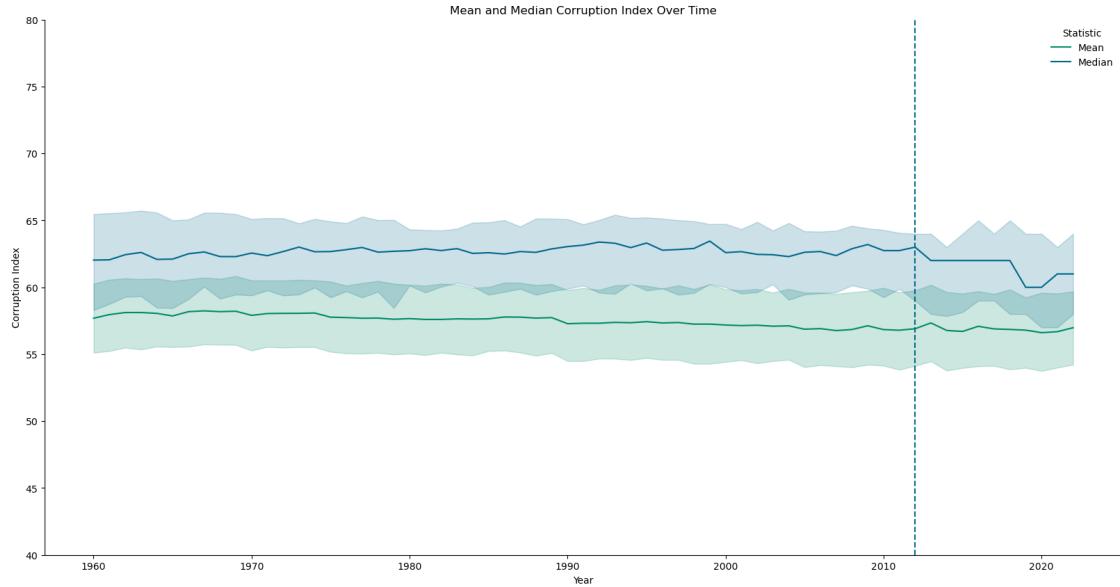


```

[256]: # Plot the aggregate mean and median CC_EST values (all countries together) over time in a line graph
# Create a line plot of the mean and median CC_EST values over time
plt.figure(figsize=(20, 10))
sns.lineplot(data=df_new, x='year', y='CPI_EST', estimator='mean', label='Mean', color = '#008B68')
sns.lineplot(data=df_new, x='year', y='CPI_EST', estimator='median', label='Median', color = '#00688B')
plt.axvline(x=2012, color='#00688B', linestyle='--')
plt.title('Mean and Median Corruption Index Over Time')
plt.ylim(40,80)
plt.xlabel('Year')
plt.ylabel('Corruption Index')
plt.legend(title='Statistic', frameon=False)
sns.despine()

```

```
plt.show()
```



```
[257]: # Plot a histogram of the average yearly change in CC_EST
plt.figure(figsize=(20, 10))
sns.histplot(df_change['change'], bins=20,color="#00688B", edgecolor="#00688B", kde=True)

# Calculate the mean and median
mean = df_change['change'].mean()
median = df_change['change'].median()

# Add lines for the mean and median
plt.axvline(mean, color='#00688B', linestyle='--', label=f'Mean: {mean:.3f}')
plt.axvline(median, color='#00688B', linestyle='-', label=f'Median: {median:.3f}')

# Customize the title and labels
plt.title('Change in Corruption Estimates (1960-2022)', fontsize=16)
plt.xlabel('Change', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.xlim(-30,30)

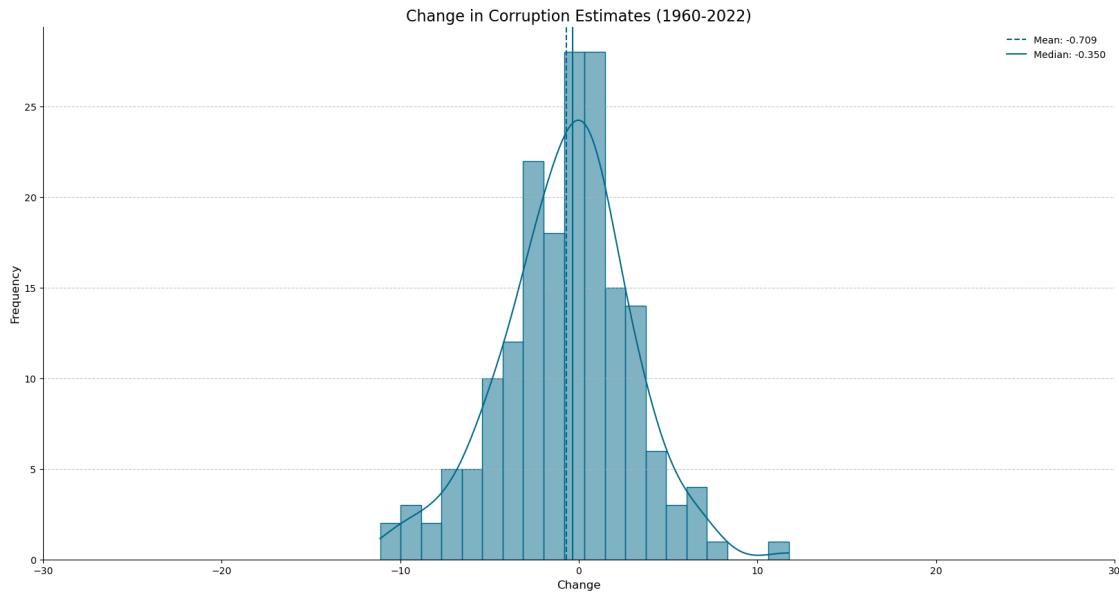
# Customize the grid
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend(frameon=False)

# Remove the top and right spines
```

```

sns.despine()
plt.show()

```



```

[258]: # Calculate the average change in CC_EST between years
df_change = df_new.pivot(index='iso2c', columns='year', values='CPI_EST')
df_change['change'] = (df_change[2022] - df_change[1960]) / (2022 - 1960)

# Sort the DataFrame in ascending order by 'change'
df_change = df_change.sort_values(by='change')

fig, ax = plt.subplots(figsize=(20, 10))

sns.barplot(x=df_change.index, y='change', data=df_change, ax=ax,
             palette='crest')

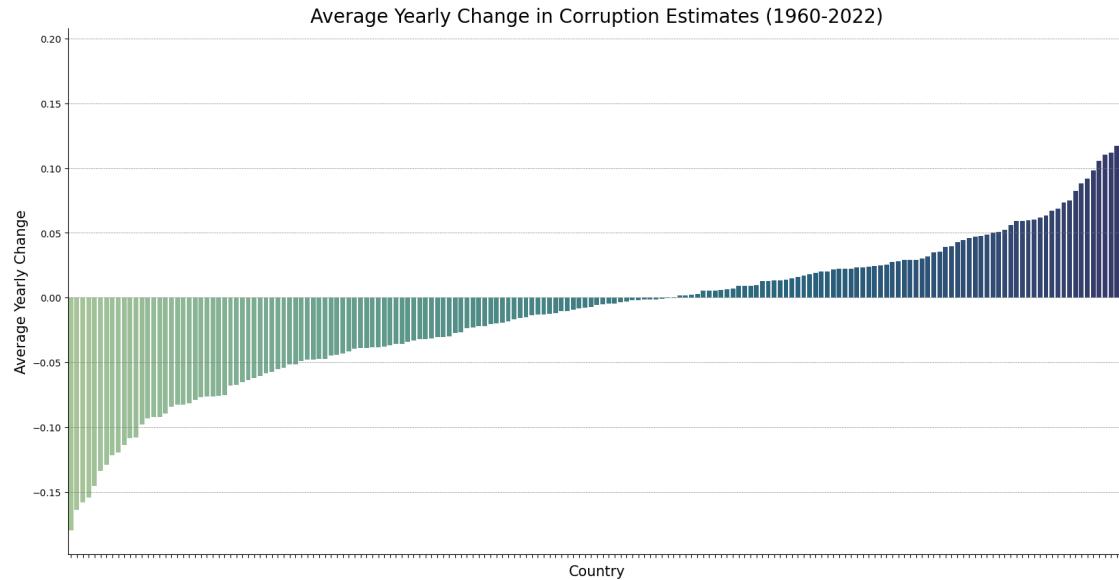
ax.set_title('Average Yearly Change in Corruption Estimates (1960-2022)', fontweight='bold', fontsize=20)
ax.set_xlabel('Country', fontweight='bold', fontsize=15)
ax.set_ylabel('Average Yearly Change', fontweight='bold', fontsize=15)

ax.set_xticklabels([])
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

ax.grid(color='gray', linestyle='--', linewidth=0.5, axis='y')

```

```
plt.show()
```



```
[259]: import matplotlib.cm as cm
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Convert the list of tuples to a dictionary
importances_dict = results[5].set_index('Feature')['Importance'].to_dict()

# Extract feature names
features = list(importances_dict.keys())

# Function to calculate mean of a list or a nested list
def calculate_mean(value):
    if isinstance(value, list):
        flattened_list = [item for sublist in value for item in sublist]
        return np.mean(flattened_list)
    else:
        return value

# Calculate importance scores and additional scores
importance_scores = [calculate_mean(importances_dict[feature]) for feature in
                     features]
additional_scores = [calculate_mean(importances_dict[feature]) for feature in
                     features]

# Create a colormap
```

```

cmap = cm.get_cmap('viridis')

# Normalize importance scores to range [0, 1] for coloring
norm = plt.Normalize(min(importance_scores), max(importance_scores))

# Create a 3D bar plot
fig = plt.figure(figsize=(20, 20))
ax = fig.add_subplot(111, projection='3d')

# Create a sequence of numbers for the x-axis
x = np.arange(len(features))

# Create the 3D bar plot
ax.bar3d(x, importance_scores, np.zeros_like(importance_scores), 1, 1, ↴
          additional_scores, color=cmap(norm(importance_scores)))

# Set the ticks on the x-axis to be the new labels and remove the lines (ticks)
ax.set_xticks(x, minor=False)
ax.set_xticklabels([])
ax.tick_params(axis='x', which='both', length=0)

# Set labels
ax.set_xlabel('Features')
ax.set_ylabel('Importance Scores')
ax.set_zlabel('Additional Scores')

# Remove the grid lines along the axes
ax.xaxis._axinfo['grid']['color'] = (1,1,1,0)

# Reverse the z-axis
#ax.invert_zaxis()

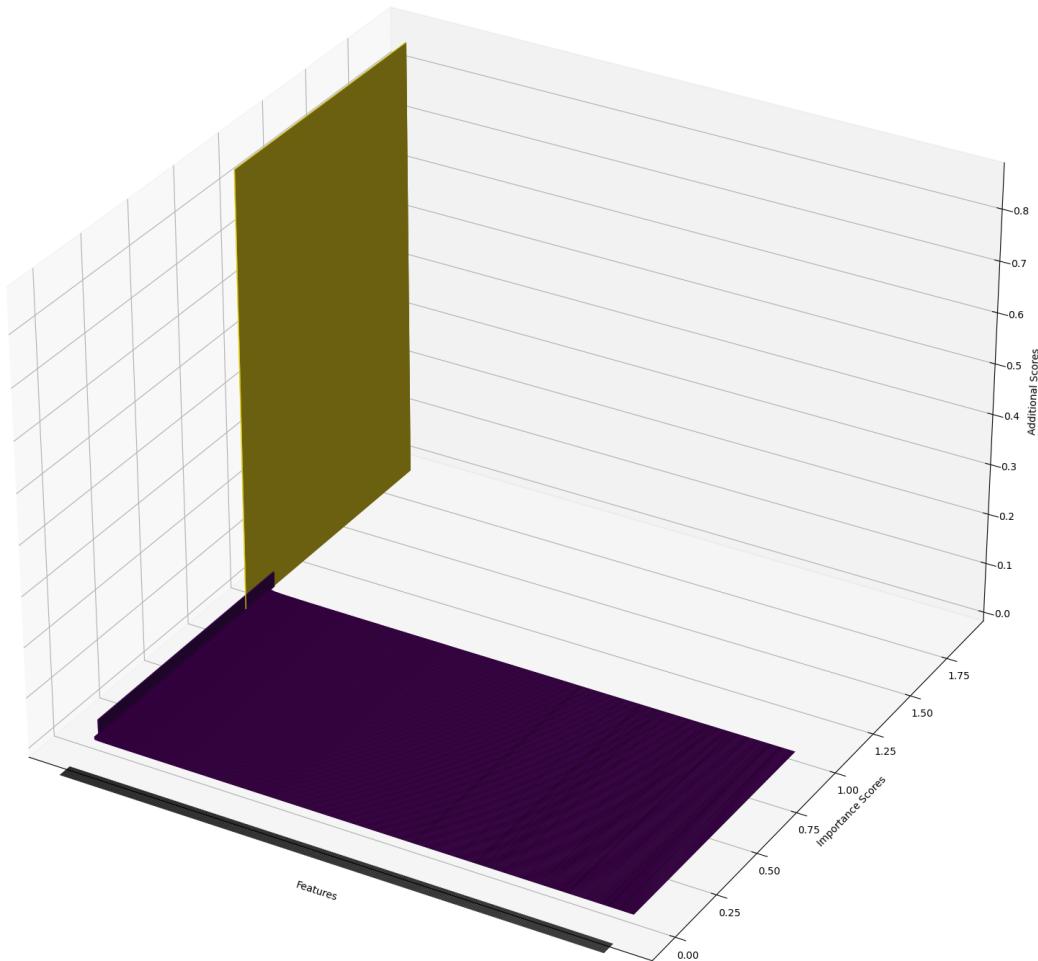
# Show the plot
plt.show()

```

```

/var/folders/1z/rmh3bk123qg9411_qfj8858000gn/T/ipykernel_37400/1061274750.py:2
4: MatplotlibDeprecationWarning: The get_cmap function was deprecated in
Matplotlib 3.7 and will be removed two minor releases later. Use
``matplotlib.colormaps[name]`` or ``matplotlib.colormaps.get_cmap(obj)`` instead.
cmap = cm.get_cmap('viridis')

```



```
[260]: # Convert the list of tuples to a dictionary
importances_dict = results[5].set_index('Feature')['Importance'].to_dict()

# Extract feature names
features = list(importances_dict.keys())

# Since all values are already floats, there's no need to calculate the mean
importance_scores = list(importances_dict.values())

# Create a 2D line plot
fig, ax = plt.subplots(figsize=(20, 10))

# Create a sequence of numbers for the x-axis
```

```

x = np.arange(len(features))

# Create the 2D line plot
ax.plot(x, importance_scores, color="#00688B")

# Set the ticks on the x-axis to be the new labels and remove the lines (ticks)
ax.set_xticks(x, minor=False)
ax.set_xticklabels([])
ax.tick_params(axis='x', which='both', length=0)

# Set labels
ax.set_xlabel('Features')
ax.set_ylabel('Importance Scores')

sns.despine()

# Show the plot
plt.show()

```



```

[261]: # create df_new_morphed that is df with cc_est replaced with CC_EST from df_new
df_new_morphed = df.copy()
df_new_morphed['CPI_EST'] = df_new['CPI_EST']

# Add columns for CC_EST_rank and CC_EST_rank_int
df_new_morphed['CP_EST_rank'] = df_new['CPI_EST_rank']
df_new_morphed['CP_EST_rank_int'] = df_new['CPI_EST_rank_int']

# Save the morphed data to a new CSV file

```

```

df_new_morphed.to_csv('df_model4.csv', index=False)

[262]: ### TAKES AROUND 12 MINUTES
# Calculate the absolute correlation of each variable with the target variable
df_numeric = df[vars_full].apply(pd.to_numeric, errors='coerce')

# Calculate the absolute correlation of each variable with the target variable
correl = df_numeric.corrwith(df['CPI_EST']).abs()

# Sort the variables by their correlation with the target variable
vars_sorted = correl.sort_values(ascending=False).index.tolist()

results = []
tree_depths = []
r2_scores = []

# Set the maximum number of variables to 10 or the total number of variables,
↳ whichever is smaller
max_vars = min(100, len(vars_sorted))

class TqdmCallback(xgb.callback.TrainingCallback):
    def __init__(self, bar):
        self._bar = bar

    def after_iteration(self, model, epoch, evals_log):
        self._bar.update(1)
        return False

# Create a progress bar
with tqdm(total=max_vars) as pbar:
    model = XGBRegressor(
        objective='reg:squarederror',
        random_state=0,
        alpha=1.0,
        reg_lambda=10.0,
        early_stopping_rounds=10,
        callbacks=[TqdmCallback(pbar)])
    )

# Create a progress bar
with tqdm(total=max_vars) as pbar:
    # Iterate over the number of variables to include in the model
    for i in range(1, max_vars + 1):
        # Select the top i variables
        vars_selected = vars_sorted[:i]

```

```

# Build and evaluate the model with the selected variables
try:
    result = build_and_evaluate_model(df, 'CPI_EST', vars_selected, n_leads=2)
except Exception as e:
    print(f"Error while building model with variables {vars_selected}: {e}")
    continue

# Store the result
results.append(result)

# Access the R^2 score for the test set
r2_test = result[1]

# Store the R^2 score
r2_scores.append(r2_test)

model = result[10]

# Retrieve the 'max_depth' parameter from the model's parameters
tree_depth = model.get_params().get('max_depth', 'Not set')

# Store the tree depth
tree_depths.append(tree_depth)

# Update the progress bar description
pbar.set_description(f"Number of variables: {i}, R^2: {r2_test:.2f}, Tree Depth: {tree_depth}")

# Update the progress bar
pbar.update()

```

```

0% | 0/100 [00:00<?, ?it/s]
0% | 0/100 [00:00<?,
?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.
warnings.warn(
89it [00:00, 1283.27it/s]
Number of variables: 1, R^2: 0.90, Tree Depth: None: 1% | 1/100
[00:00<00:28, 3.53it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

```

```

    warnings.warn(
31it [00:00, 682.85it/s]
Number of variables: 2, R^2: 0.99, Tree Depth: None:  2%|           | 2/100
[00:00<00:25,  3.91it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
30it [00:00, 1020.40it/s]
Number of variables: 3, R^2: 0.99, Tree Depth: None:  3%|           | 3/100
[00:00<00:25,  3.83it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
41it [00:00, 1113.41it/s]
Number of variables: 4, R^2: 0.99, Tree Depth: None:  4%|           | 4/100
[00:01<00:24,  3.98it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
28it [00:00, 1087.42it/s]
Number of variables: 5, R^2: 0.99, Tree Depth: None:  5%|           | 5/100
[00:01<00:22,  4.20it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
31it [00:00, 856.66it/s]
Number of variables: 6, R^2: 0.99, Tree Depth: None:  6%|           | 6/100
[00:01<00:23,  3.94it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
31it [00:00, 875.04it/s]
Number of variables: 7, R^2: 0.99, Tree Depth: None:  7%|           | 7/100
[00:01<00:23,  3.90it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
27it [00:00, 527.02it/s]
Number of variables: 8, R^2: 0.99, Tree Depth: None:  8%|           | 8/100
[00:02<00:28,  3.21it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in

```

```
constructor or `set_params` instead.

    warnings.warn(
31it [00:00, 496.01it/s]
Number of variables: 9, R^2: 0.98, Tree Depth: None:  9%|          | 9/100
[00:02<00:29,  3.04it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
36it [00:00, 646.31it/s]
Number of variables: 10, R^2: 0.99, Tree Depth: None: 10%|          | 10/100
[00:03<00:32,  2.73it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
68it [00:00, 585.66it/s]
Number of variables: 11, R^2: 0.99, Tree Depth: None: 11%|          | 11/100
[00:03<00:36,  2.44it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
29it [00:00, 595.22it/s]
Number of variables: 12, R^2: 0.99, Tree Depth: None: 12%|          | 12/100
[00:03<00:36,  2.40it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
52it [00:00, 632.46it/s]
Number of variables: 13, R^2: 0.99, Tree Depth: None: 13%|          | 13/100
[00:04<00:44,  1.94it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
37it [00:00, 551.74it/s]
Number of variables: 14, R^2: 0.99, Tree Depth: None: 14%|          | 14/100
[00:05<00:44,  1.95it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
27it [00:00, 440.64it/s]
Number of variables: 15, R^2: 0.99, Tree Depth: None: 15%|          | 15/100
[00:05<00:49,  1.72it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
```

```
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
52it [00:00, 561.39it/s]
Number of variables: 16, R^2: 0.99, Tree Depth: None: 16% | 16/100
[00:06<00:51, 1.64it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
32it [00:00, 549.45it/s]
Number of variables: 17, R^2: 0.99, Tree Depth: None: 17% | 17/100
[00:07<00:48, 1.72it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
62it [00:00, 186.30it/s]
Number of variables: 18, R^2: 0.99, Tree Depth: None: 18% | 18/100
[00:08<00:56, 1.46it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
39it [00:00, 475.84it/s]
Number of variables: 19, R^2: 0.98, Tree Depth: None: 19% | 19/100
[00:08<00:54, 1.49it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
30it [00:00, 454.12it/s]
Number of variables: 20, R^2: 0.98, Tree Depth: None: 20% | 20/100
[00:09<00:54, 1.46it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
51it [00:00, 370.67it/s]
Number of variables: 21, R^2: 0.98, Tree Depth: None: 21% | 21/100
[00:10<00:58, 1.35it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
35it [00:00, 253.90it/s]
Number of variables: 22, R^2: 0.98, Tree Depth: None: 22% | 22/100
[00:11<01:06, 1.17it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
```

```
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
67it [00:00, 433.92it/s]
Number of variables: 23, R^2: 0.99, Tree Depth: None: 23%|           | 23/100
[00:12<01:04,  1.19it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
46it [00:00, 415.94it/s]
Number of variables: 24, R^2: 0.99, Tree Depth: None: 24%|           | 24/100
[00:12<01:00,  1.25it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
30it [00:00, 374.07it/s]
Number of variables: 25, R^2: 0.98, Tree Depth: None: 25%|           | 25/100
[00:13<00:58,  1.28it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
46it [00:00, 396.29it/s]
Number of variables: 26, R^2: 0.99, Tree Depth: None: 26%|           | 26/100
[00:14<01:01,  1.21it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
66it [00:00, 219.26it/s]
Number of variables: 27, R^2: 0.99, Tree Depth: None: 27%|           | 27/100
[00:15<01:06,  1.10it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
64it [00:00, 302.77it/s]
Number of variables: 28, R^2: 0.99, Tree Depth: None: 28%|           | 28/100
[00:17<01:28,  1.23s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
62it [00:00, 275.10it/s]
Number of variables: 29, R^2: 0.99, Tree Depth: None: 29%|           | 29/100
```

```

[00:18<01:25,  1.21s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
57it [00:00, 298.52it/s]
Number of variables: 30, R^2: 0.99, Tree Depth: None: 30%|           | 30/100
[00:20<01:24,  1.20s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
40it [00:00, 311.87it/s]
Number of variables: 31, R^2: 0.99, Tree Depth: None: 31%|           | 31/100
[00:20<01:17,  1.12s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
35it [00:00, 300.41it/s]
Number of variables: 32, R^2: 0.99, Tree Depth: None: 32%|           | 32/100
[00:21<01:10,  1.04s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
86it [00:00, 349.90it/s]
Number of variables: 33, R^2: 0.99, Tree Depth: None: 33%|           | 33/100
[00:22<01:07,  1.01s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
60it [00:00, 326.57it/s]
Number of variables: 34, R^2: 0.99, Tree Depth: None: 34%|           | 34/100
[00:23<01:08,  1.04s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
93it [00:00, 280.48it/s]
Number of variables: 35, R^2: 0.99, Tree Depth: None: 35%|           | 35/100
[00:24<01:06,  1.02s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
51it [00:00, 294.30it/s]

```

```
Number of variables: 36, R^2: 0.99, Tree Depth: None: 36%| 36/100
[00:26<01:22, 1.29s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
53it [00:00, 182.30it/s]
Number of variables: 37, R^2: 0.99, Tree Depth: None: 37%| 37/100
[00:28<01:22, 1.31s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
54it [00:00, 235.43it/s]
Number of variables: 38, R^2: 0.99, Tree Depth: None: 38%| 38/100
[00:29<01:22, 1.33s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
84it [00:00, 182.71it/s]
Number of variables: 39, R^2: 0.99, Tree Depth: None: 39%| 39/100
[00:31<01:27, 1.43s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
48it [00:00, 95.87it/s]
Number of variables: 40, R^2: 0.99, Tree Depth: None: 40%| 40/100
[00:32<01:30, 1.51s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
64it [00:00, 96.15it/s]
Number of variables: 41, R^2: 0.99, Tree Depth: None: 41%| 41/100
[00:34<01:32, 1.57s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
82it [00:00, 259.65it/s]
Number of variables: 42, R^2: 0.99, Tree Depth: None: 42%| 42/100
[00:35<01:28, 1.53s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
```

```
51it [00:00, 277.59it/s]
Number of variables: 43, R^2: 0.99, Tree Depth: None: 43%| 43/100
[00:37<01:21, 1.44s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
64it [00:00, 259.12it/s]
Number of variables: 44, R^2: 0.99, Tree Depth: None: 44%| 44/100
[00:38<01:20, 1.45s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
45it [00:00, 79.08it/s]
Number of variables: 45, R^2: 0.99, Tree Depth: None: 45%| 45/100
[00:40<01:33, 1.69s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
92it [00:00, 207.15it/s]
Number of variables: 46, R^2: 0.99, Tree Depth: None: 46%| 46/100
[00:43<01:40, 1.87s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
97it [00:00, 195.11it/s]
Number of variables: 47, R^2: 0.99, Tree Depth: None: 47%| 47/100
[00:44<01:32, 1.75s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
72it [00:00, 173.91it/s]
Number of variables: 48, R^2: 0.99, Tree Depth: None: 48%| 48/100
[00:47<01:45, 2.02s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
62it [00:00, 197.01it/s]
Number of variables: 49, R^2: 0.99, Tree Depth: None: 49%| 49/100
[00:49<01:43, 2.02s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
```

```

    warnings.warn(
57it [00:00, 205.41it/s]
Number of variables: 50, R^2: 0.99, Tree Depth: None: 50%|      | 50/100
[00:50<01:33,  1.86s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:00, 115.56it/s]
Number of variables: 51, R^2: 0.99, Tree Depth: None: 51%|      | 51/100
[00:53<01:46,  2.18s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
66it [00:00, 180.69it/s]
Number of variables: 52, R^2: 0.99, Tree Depth: None: 52%|      | 52/100
[00:55<01:34,  1.98s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
60it [00:00, 115.03it/s]
Number of variables: 53, R^2: 0.99, Tree Depth: None: 53%|      | 53/100
[00:56<01:27,  1.87s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
58it [00:00, 189.44it/s]
Number of variables: 54, R^2: 0.99, Tree Depth: None: 54%|      | 54/100
[00:58<01:21,  1.77s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:00, 125.47it/s]
Number of variables: 55, R^2: 0.99, Tree Depth: None: 55%|      | 55/100
[01:00<01:21,  1.82s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
94it [00:00, 133.76it/s]
Number of variables: 56, R^2: 0.99, Tree Depth: None: 56%|      | 56/100
[01:03<01:31,  2.07s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in

```

```

constructor or `set_params` instead.

    warnings.warn(
47it [00:00, 153.41it/s]
Number of variables: 57, R^2: 0.99, Tree Depth: None: 57%|      | 57/100
[01:05<01:30,  2.12s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
70it [00:00, 203.04it/s]
Number of variables: 58, R^2: 0.99, Tree Depth: None: 58%|      | 58/100
[01:07<01:28,  2.10s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
51it [00:00, 164.88it/s]
Number of variables: 59, R^2: 0.99, Tree Depth: None: 59%|      | 59/100
[01:09<01:27,  2.12s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
62it [00:00, 161.71it/s]
Number of variables: 60, R^2: 0.99, Tree Depth: None: 60%|      | 60/100
[01:11<01:24,  2.11s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
93it [00:00, 189.98it/s]
Number of variables: 61, R^2: 0.99, Tree Depth: None: 61%|      | 61/100
[01:13<01:17,  2.00s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
100it [00:00, 123.85it/s]
Number of variables: 62, R^2: 0.99, Tree Depth: None: 62%|      | 62/100
[01:15<01:15,  1.98s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
81it [00:00, 126.31it/s]
Number of variables: 63, R^2: 0.99, Tree Depth: None: 63%|      | 63/100
[01:17<01:20,  2.19s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is

```

```
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:00, 187.30it/s]
Number of variables: 64, R^2: 0.99, Tree Depth: None: 64%|       | 64/100
[01:20<01:18,  2.19s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
67it [00:00, 175.51it/s]
Number of variables: 65, R^2: 0.99, Tree Depth: None: 65%|       | 65/100
[01:22<01:16,  2.17s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
81it [00:00, 132.17it/s]
Number of variables: 66, R^2: 0.99, Tree Depth: None: 66%|       | 66/100
[01:24<01:18,  2.31s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
81it [00:00, 159.48it/s]
Number of variables: 67, R^2: 0.99, Tree Depth: None: 67%|       | 67/100
[01:27<01:17,  2.34s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
43it [00:00, 142.52it/s]
Number of variables: 68, R^2: 0.99, Tree Depth: None: 68%|       | 68/100
[01:29<01:15,  2.36s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
86it [00:00, 164.67it/s]
Number of variables: 69, R^2: 0.99, Tree Depth: None: 69%|       | 69/100
[01:32<01:13,  2.39s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
51it [00:00, 135.62it/s]
Number of variables: 70, R^2: 0.99, Tree Depth: None: 70%|       | 70/100
[01:35<01:18,  2.61s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
```

```
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
61it [00:00, 163.90it/s]
Number of variables: 71, R^2: 0.99, Tree Depth: None: 71%|      | 71/100
[01:37<01:11,  2.48s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
100it [00:00, 169.05it/s]
Number of variables: 72, R^2: 0.99, Tree Depth: None: 72%|      | 72/100
[01:40<01:14,  2.66s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
67it [00:00, 145.91it/s]
Number of variables: 73, R^2: 0.99, Tree Depth: None: 73%|      | 73/100
[01:43<01:15,  2.80s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
51it [00:00, 102.20it/s]
Number of variables: 74, R^2: 0.99, Tree Depth: None: 74%|      | 74/100
[01:46<01:12,  2.78s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
58it [00:00, 146.34it/s]
Number of variables: 75, R^2: 0.99, Tree Depth: None: 75%|      | 75/100
[01:49<01:09,  2.78s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
100it [00:00, 139.45it/s]
Number of variables: 76, R^2: 0.99, Tree Depth: None: 76%|      | 76/100
[01:51<01:04,  2.70s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
64it [00:00, 140.51it/s]
Number of variables: 77, R^2: 0.99, Tree Depth: None: 77%|      | 77/100
```

```

[01:54<01:02,  2.71s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
78it [00:00, 123.51it/s]
Number of variables: 78, R^2: 0.99, Tree Depth: None: 78%|      | 78/100
[01:57<01:01,  2.78s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
62it [00:00, 149.93it/s]
Number of variables: 79, R^2: 0.99, Tree Depth: None: 79%|      | 79/100
[01:59<00:56,  2.69s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
67it [00:00, 132.29it/s]
Number of variables: 80, R^2: 0.99, Tree Depth: None: 80%|      | 80/100
[02:02<00:54,  2.74s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
81it [00:00, 162.06it/s]
Number of variables: 81, R^2: 0.99, Tree Depth: None: 81%|      | 81/100
[02:05<00:50,  2.68s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
61it [00:00, 125.05it/s]
Number of variables: 82, R^2: 0.99, Tree Depth: None: 82%|      | 82/100
[02:07<00:47,  2.63s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
44it [00:00, 126.55it/s]
Number of variables: 83, R^2: 0.99, Tree Depth: None: 83%|      | 83/100
[02:10<00:46,  2.74s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:00, 145.80it/s]

```

```
Number of variables: 84, R^2: 0.99, Tree Depth: None: 84%| 84/100
[02:13<00:44, 2.77s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
94it [00:00, 144.82it/s]
Number of variables: 85, R^2: 0.99, Tree Depth: None: 85%| 85/100
[02:16<00:43, 2.88s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
93it [00:00, 140.93it/s]
Number of variables: 86, R^2: 0.99, Tree Depth: None: 86%| 86/100
[02:19<00:40, 2.93s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:00, 121.69it/s]
Number of variables: 87, R^2: 0.99, Tree Depth: None: 87%| 87/100
[02:23<00:39, 3.06s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:00, 144.41it/s]
Number of variables: 88, R^2: 0.99, Tree Depth: None: 88%| 88/100
[02:26<00:37, 3.09s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
62it [00:00, 144.42it/s]
Number of variables: 89, R^2: 0.99, Tree Depth: None: 89%| 89/100
[02:29<00:33, 3.09s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
36it [00:00, 144.64it/s]
Number of variables: 90, R^2: 0.99, Tree Depth: None: 90%| 90/100
[02:31<00:28, 2.87s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
```

```
38it [00:00, 125.84it/s]
Number of variables: 91, R^2: 0.99, Tree Depth: None: 91%|      | 91/100
[02:34<00:24,  2.78s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
53it [00:00, 129.88it/s]
Number of variables: 92, R^2: 0.99, Tree Depth: None: 92%|      | 92/100
[02:37<00:22,  2.79s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
53it [00:00, 119.09it/s]
Number of variables: 93, R^2: 0.99, Tree Depth: None: 93%|      | 93/100
[02:39<00:19,  2.79s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
60it [00:00, 147.37it/s]
Number of variables: 94, R^2: 0.99, Tree Depth: None: 94%|      | 94/100
[02:42<00:17,  2.87s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
81it [00:00, 134.31it/s]
Number of variables: 95, R^2: 0.99, Tree Depth: None: 95%|      | 95/100
[02:45<00:14,  2.91s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
69it [00:00, 141.19it/s]
Number of variables: 96, R^2: 0.99, Tree Depth: None: 96%|      | 96/100
[02:49<00:11,  2.96s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
99it [00:00, 122.74it/s]
Number of variables: 97, R^2: 0.99, Tree Depth: None: 97%|      | 97/100
[02:52<00:09,  3.01s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
```

```

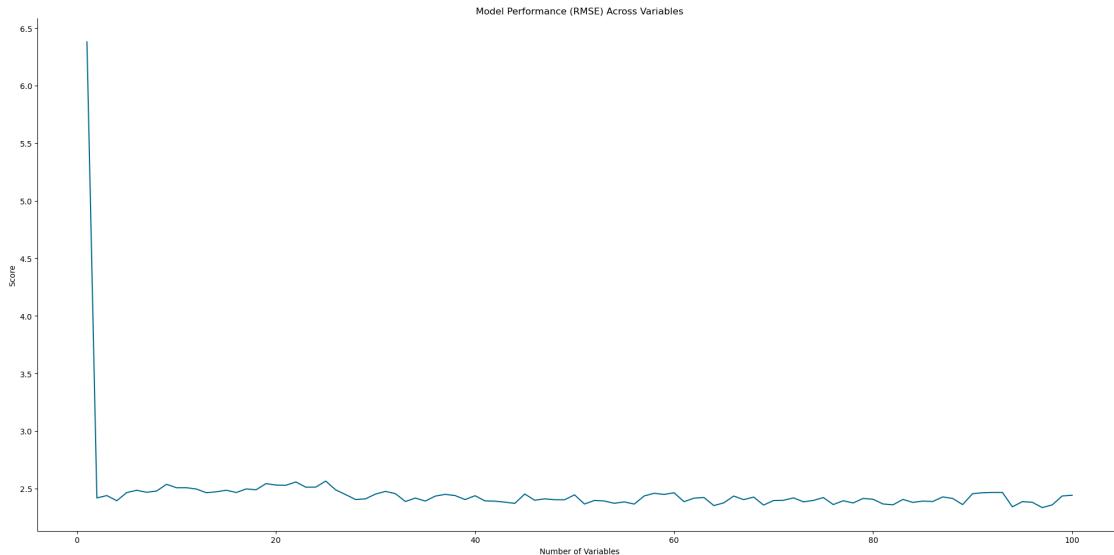
warnings.warn(
69it [00:00, 129.35it/s]
Number of variables: 98, R^2: 0.99, Tree Depth: None: 98%|      | 98/100
[02:54<00:05,  2.90s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
47it [00:00, 122.50it/s]
Number of variables: 99, R^2: 0.99, Tree Depth: None: 99%|      | 99/100
[02:58<00:03,  3.10s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
37it [00:00, 137.53it/s]
Number of variables: 100, R^2: 0.99, Tree Depth: None: 100%|      | 100/100
[03:00<00:00,  1.81s/it]

```

```
[263]: # Plot
fig, ax = plt.subplots(figsize=(20, 10))

# Line plot of R^2 and RMSE across different thresholds
r2_scores = [res[1] for res in results]
rmse_scores = [res[3] for res in results]
num_vars = range(1, max_vars + 1)
ax.plot(num_vars, rmse_scores,color='#00688B')
ax.set_xlabel('Number of Variables')
ax.set_ylabel('Score')
ax.set_title('Model Performance (RMSE) Across Variables')
sns.despine()

plt.tight_layout()
plt.show()
```



```
[264]: # Calculate the absolute differences for each model
abs_diffs = [abs(res[7] - res[8]) for res in results]

# Create a new subplot for the variance plot
fig, ax = plt.subplots(figsize=(20, 10))

# Get a color map
colors = sns.color_palette("mako", len(abs_diffs))

# KDE plot of the absolute differences for each model
for i, abs_diff in enumerate(abs_diffs):
    sns.kdeplot(abs_diff, color=colors[i], ax=ax)

# Calculate the aggregate mean and median of the absolute differences
mean_abs_diff = np.mean([np.mean(abs_diff) for abs_diff in abs_diffs])
median_abs_diff = np.median([np.median(abs_diff) for abs_diff in abs_diffs])

# Add a vertical line at the aggregate mean and median
ax.axvline(mean_abs_diff, color="#00688B", linestyle='--', label=f'Aggregate Mean: {mean_abs_diff:.2f}')
ax.axvline(median_abs_diff, color="#00688B", linestyle='-', label=f'Aggregate Median: {median_abs_diff:.2f}')

ax.set_xlabel('Absolute Difference')
ax.set_title('Distribution of Absolute Differences Across Different Number of Variables')

# Create a colorbar
```

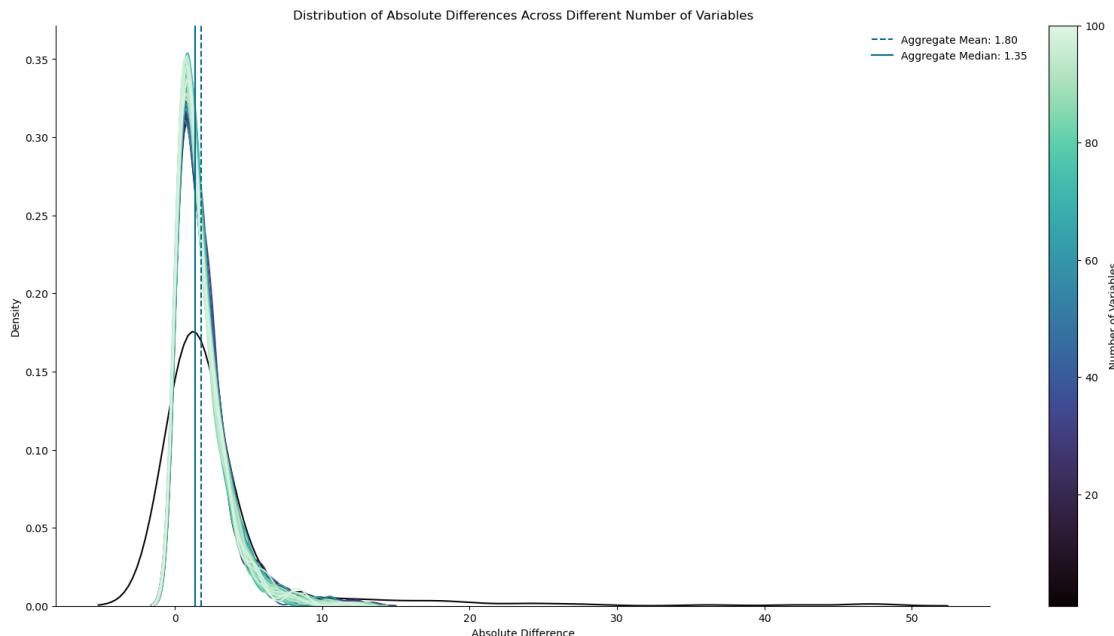
```

norm = Normalize(vmin=1, vmax=len(abs_diffs))
cbar = plt.colorbar(cm.ScalarMappable(norm=norm, cmap=sns.color_palette("mako", len(abs_diffs)), as_cmap=True)), ax=ax)
cbar.set_label('Number of Variables')

ax.legend(bbox_to_anchor=(1.05, 1), frameon=False)
sns.despine()

plt.show()

```



```

[265]: # Calculate the absolute differences for each model
abs_diffs = [abs(res[7] - res[8]) for res in results]

# Create a new subplot for the variance plot
fig, ax = plt.subplots(figsize=(20, 10))

# Get a color map
colors = sns.color_palette("mako", len(abs_diffs))

# KDE plot of the absolute differences for each model
for i, abs_diff in enumerate(abs_diffs):
    sns.kdeplot(abs_diff, color=colors[i], ax=ax)

# Calculate the aggregate mean and median of the absolute differences
mean_abs_diff = np.mean([np.mean(abs_diff) for abs_diff in abs_diffs])
median_abs_diff = np.median([np.median(abs_diff) for abs_diff in abs_diffs])

```

```

# Add a vertical line at the aggregate mean and median
ax.axvline(mean_abs_diff, color='#00688B', linestyle='--', label=f'Aggregate\u20d7Mean: {mean_abs_diff:.2f}')
ax.axvline(median_abs_diff, color='#00688B', linestyle='-', label=f'Aggregate\u20d7Median: {median_abs_diff:.2f}')

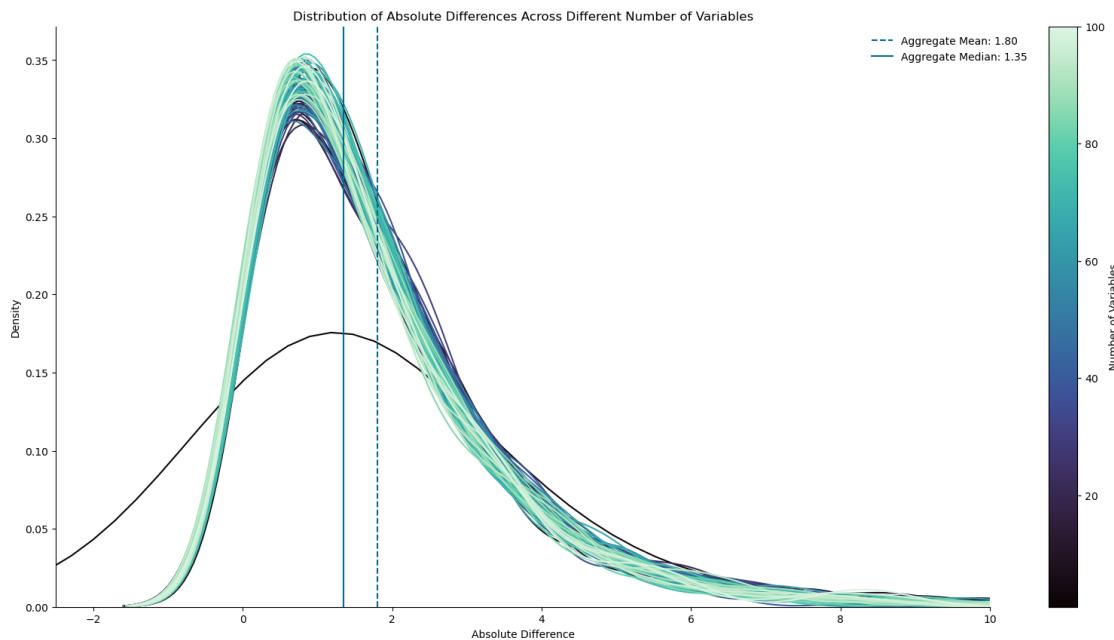
ax.set_xlabel('Absolute Difference')
ax.set_title('Distribution of Absolute Differences Across Different Number of\u20d7Variables')
ax.set_xlim(-2.5,10)

# Create a colorbar
norm = Normalize(vmin=1, vmax=len(abs_diffs))
cbar = plt.colorbar(cm.ScalarMappable(norm=norm, cmap=sns.color_palette("mako", len(abs_diffs)), as_cmap=True)), ax=ax)
cbar.set_label('Number of Variables')

ax.legend(bbox_to_anchor=(1.05, 1), frameon=False)
sns.despine()

plt.show()

```



```
[273]: # Create a new subplot for the scatter plot
fig, ax = plt.subplots(figsize=(20, 10))
```

```

# Create a colormap
cmap = sns.color_palette("mako", as_cmap=True)

# Calculate the number of lines to plot
num_lines = len(results)

# Scatter plot of the model accuracy over epochs for all models
for i, res in enumerate(results):
    # Calculate the color for the current line
    color = cmap(i / num_lines)

    ax.plot(range(len(res[10].evals_result()['validation_0']['rmse'])),
            res[10].evals_result()['validation_0']['rmse'],
            label=f'{len(res[5])} variables',
            color=color)

ax.set_xlabel('Epoch')
ax.set_ylabel('RMSE')
ax.set_title('Model Accuracy Over Epochs')
ax.set_ylim(2.2, 3.2)

# Create a colorbar
sm = plt.cm.ScalarMappable(cmap=cmap, norm=plt.Normalize(vmin=0, vmax=num_lines))
plt.colorbar(sm, label='Number of Variables')

plt.show()

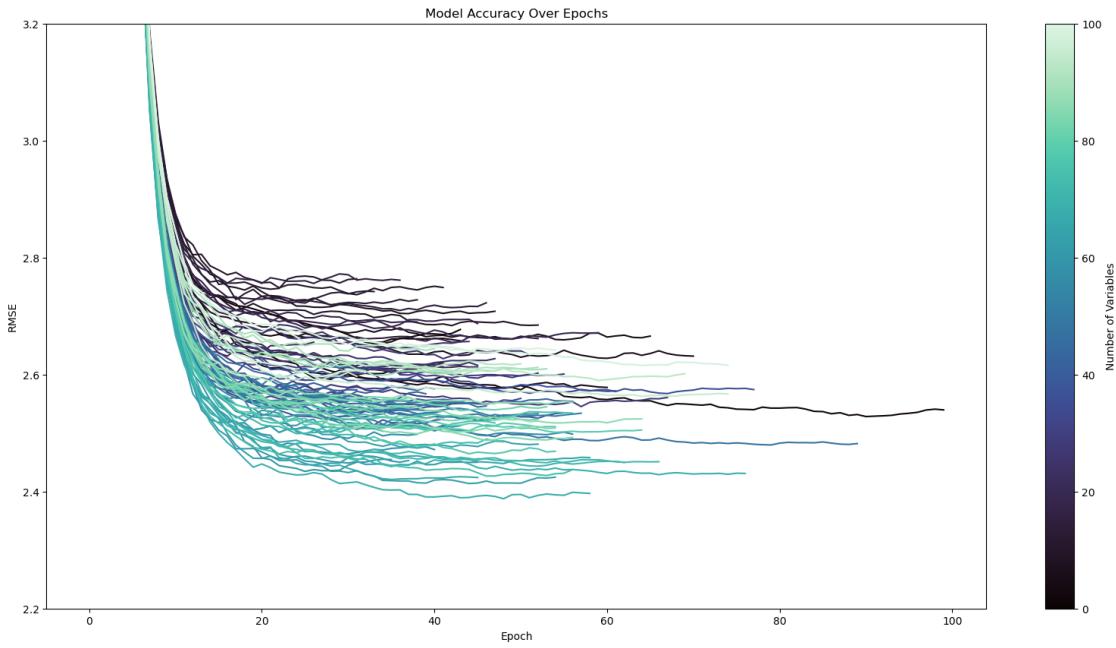
```

```

/var/folders/1z/rmh3bk123qg9411_qfj88580000gn/T/ipykernel_37400/1165927619.py:2
7: MatplotlibDeprecationWarning: Unable to determine Axes to steal space for
Colorbar. Using gca(), but will raise in the future. Either provide the *cax*
argument to use as the Axes for the Colorbar, provide the *ax* argument to steal
space from it, or add *mappable* to an Axes.

```

```
    plt.colorbar(sm, label='Number of Variables')
```



```
[267]: def custom_objective(y_true, y_pred):
    """
    Custom objective function that penalizes predictions close to or beyond the
    bounds of 0 and 100,
    penalizes large changes from the previous year's prediction, and penalizes
    an overall movement greater than 20 from the 2012 value.
    Also penalizes the overreliance on the 'CPI_EST_avg' feature.
    """
    # Convert y_true and y_pred to pandas Series
    y_true = pd.Series(y_true)
    y_pred = pd.Series(y_pred)

    # Calculate the squared error
    squared_error = (y_true - y_pred) ** 2

    # Calculate the penalty term for the overreliance on the 'CPI_EST_avg' feature
    penalty_CPI_EST_avg = np.abs(y_pred)

    # Increase the penalty for the overreliance on the 'CPI_EST_avg' feature
    penalty_CPI_EST_avg *= 100

    # Add the penalty terms to the squared error
    penalized_squared_error = squared_error + penalty_CPI_EST_avg

    # Calculate the first derivative (gradient) of the penalized squared error
    return 2 * (y_pred - y_true) + 200 * np.sign(penalty_CPI_EST_avg)
```

```

grad = -2 * (y_true - y_pred) + np.sign(y_pred)

# Calculate the second derivative (Hessian) of the penalized squared error
hess = np.ones_like(grad) * 2

return grad, hess

class TqdmCallback(xgb.callback.TrainingCallback):
    def __init__(self, bar):
        self._bar = bar

    def after_iteration(self, model, epoch, evals_log):
        self._bar.update(1)
        return False

    def build_and_evaluate_model(df, target_col_name, var_list, n_leads=2,
                                 max_depth=None):
        """
        Builds, evaluates, and returns results for an XGBoost model.
        """
        # Create a copy of the DataFrame to avoid modifying the original
        df = df.copy()

        # Exclude 'iso2c', 'country', target_col_name, and lead variables from
        # var_list
        var_list = [var for var in var_list if var not in ['iso2c', 'country',
                                                          target_col_name, 'year', 'iso3n'] + [f"{target_col_name}_lead{i}" for i in
                                                          range(1, n_leads + 1)]]
        if 'CPI_EST_avg' not in var_list:
            var_list.append('CPI_EST_avg')

        # Remove rows with invalid values in the target column
        df = df[np.isfinite(df[target_col_name]) & (abs(df[target_col_name]) <=
                                                      1e30)]

        # Add a new feature for the previous year's target value
        df[target_col_name + '_prev'] = df.groupby('iso2c')[target_col_name].shift()
        var_list.append(target_col_name + '_prev')

    X = df[var_list]

    df_train_val, df_test = train_test_split(df, test_size=0.1, random_state=0)

    x_train_val, x_test, y_train_val, y_test = train_test_split(X,
                                                               df[target_col_name], test_size=0.1, random_state=0)

    # Use the custom objective function in the XGBoost model

```

```

model = XGBRegressor(
    objective=custom_objective,
    random_state=0,
    alpha=1.0,
    reg_lambda=10.0,
    early_stopping_rounds=10,
    max_depth=max_depth,
    gamma=2.0
)

with tqdm(total=model.get_params()['n_estimators']) as pbar:
    model.fit(
        x_train_val,
        y_train_val,
        eval_set=[(x_test, y_test)],
        verbose=False,
        callbacks=[TqdmCallback(pbar)]
    )

y_train_val_pred = model.predict(x_train_val)
y_test_pred = model.predict(x_test)

# Clip the predictions to be within the desired range
y_train_val_pred = np.clip(y_train_val_pred, 0, 100)
y_test_pred = np.clip(y_test_pred, 0, 100)

r2_train_val = round(r2_score(y_train_val, y_train_val_pred), 5)
r2_test = round(r2_score(y_test, y_test_pred), 5)

rmse_train_val = round(np.sqrt(mean_squared_error(y_train_val, y_train_val_pred)), 5)
rmse_test = round(np.sqrt(mean_squared_error(y_test, y_test_pred)), 5)

cv_scores = custom_cross_val_score(model, x_train_val, y_train_val, cv=5)
mean_cv_score = round(np.mean(cv_scores), 5)

# Get the feature importances
feature_importances = model.feature_importances_

# Get the feature names from x_lead
feature_names = x_train_val.columns.tolist()

# Align feature importances with feature names
feature_importances_aligned = pd.Series(feature_importances, index=feature_names)

# Create a DataFrame with the aligned feature importances

```

```

feature_importances_df = pd.DataFrame({
    'Feature': feature_importances_aligned.index,
    'Importance': np.round(feature_importances_aligned.values, 4)
}).sort_values(by='Importance', ascending=False)

return r2_train_val, r2_test, rmse_train_val, rmse_test, mean_cv_score, □
feature_importances_df, x_test.index, y_test_pred, y_test, df, model

```

[268]: *### TAKES c.12 minutes!*

```

# Assuming df is your DataFrame and build_and_evaluate_model is a defined function
# Assuming vars_full is a list of column names in df

# Select the full set of variables
vars_selected = df[vars_full].columns.tolist()

# Define max depth values to test
max_depth_values = list(range(1, 21))

# Initialize lists to store R^2 scores for training and test sets
train_r2_scores = []
test_r2_scores = []

# Create a progress bar
with tqdm(total=len(max_depth_values)) as pbar:
    # Iterate over each max_depth value
    for max_depth in max_depth_values:
        # Build and evaluate the model with the full set of variables and current max_depth
        try:
            result = build_and_evaluate_model(df, 'CPI_EST', vars_selected, □
n_leads=2, max_depth=max_depth)
        except Exception as e:
            print(f"Error while building model at max_depth {max_depth}: {e}")
            continue

        # Access the R^2 scores for the training and test sets
        r2_train = result[0]
        r2_test = result[1]

        # Store the R^2 scores
        train_r2_scores.append(r2_train)
        test_r2_scores.append(r2_test)

    # Update the progress bar description

```

```

        pbar.set_description(f"Max Depth: {max_depth}, Train R^2: {r2_train:.2f}, Test R^2: {r2_test:.2f}")

    # Update the progress bar
    pbar.update()

0% | 0/20 [00:00<?, ?it/s]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
100it [00:00, 122.56it/s]
Max Depth: 1, Train R^2: 0.98, Test R^2: 0.98:  5%| 1/20
[00:05<01:38,  5.20s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
100it [00:00, 108.36it/s]
Max Depth: 2, Train R^2: 0.99, Test R^2: 0.98:  10%| 2/20
[00:10<01:38,  5.46s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
100it [00:01, 82.68it/s]
Max Depth: 3, Train R^2: 1.00, Test R^2: 0.99:  15%| 3/20
[00:18<01:51,  6.53s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
100it [00:02, 49.36it/s]
Max Depth: 4, Train R^2: 1.00, Test R^2: 0.99:  20%| 4/20
[00:28<02:03,  7.74s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
66it [00:01, 41.99it/s]
Max Depth: 5, Train R^2: 1.00, Test R^2: 0.99:  25%| 5/20
[00:37<02:04,  8.32s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or `set_params` instead.

    warnings.warn(
100it [00:02, 39.84it/s]

```

```

Max Depth: 6, Train R^2: 1.00, Test R^2: 0.99: 30%|       | 6/20
[00:48<02:08,  9.18s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:01, 50.77it/s]
Max Depth: 7, Train R^2: 1.00, Test R^2: 0.99: 35%|       | 7/20
[01:00<02:13, 10.24s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
40it [00:02, 16.93it/s]
Max Depth: 8, Train R^2: 1.00, Test R^2: 0.99: 40%|       | 8/20
[01:14<02:16, 11.38s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:02, 39.38it/s]
Max Depth: 9, Train R^2: 1.00, Test R^2: 0.99: 45%|       | 9/20
[01:27<02:09, 11.75s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:02, 39.65it/s]
Max Depth: 10, Train R^2: 1.00, Test R^2: 0.99: 50%|       | 10/20
[01:42<02:09, 12.96s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
43it [00:02, 18.07it/s]
Max Depth: 11, Train R^2: 1.00, Test R^2: 0.99: 55%|       | 11/20
[01:56<01:57, 13.02s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(
100it [00:02, 36.10it/s]
Max Depth: 12, Train R^2: 1.00, Test R^2: 0.99: 60%|       | 12/20
[02:14<01:57, 14.70s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.

    warnings.warn(

```

```
100it [00:02, 37.21it/s]
Max Depth: 13, Train R^2: 1.00, Test R^2: 0.99: 65%|       | 13/20
[02:30<01:45, 15.11s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
39it [00:03, 12.65it/s]
Max Depth: 14, Train R^2: 1.00, Test R^2: 0.99: 70%|       | 14/20
[02:45<01:29, 14.91s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
100it [00:02, 33.62it/s]
Max Depth: 15, Train R^2: 1.00, Test R^2: 0.99: 75%|       | 15/20
[02:59<01:14, 14.87s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
100it [00:02, 34.16it/s]
Max Depth: 16, Train R^2: 1.00, Test R^2: 0.99: 80%|       | 16/20
[03:17<01:03, 15.76s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
100it [00:03, 29.21it/s]
Max Depth: 17, Train R^2: 1.00, Test R^2: 0.99: 85%|       | 17/20
[03:33<00:47, 15.69s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
100it [00:02, 33.58it/s]
Max Depth: 18, Train R^2: 1.00, Test R^2: 0.99: 90%|       | 18/20
[03:49<00:31, 15.73s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
    warnings.warn(
100it [00:03, 28.03it/s]
Max Depth: 19, Train R^2: 1.00, Test R^2: 0.99: 95%|       | 19/20
[04:05<00:15, 15.97s/it]/Users/arthurjohnson/anaconda3/lib/python3.11/site-
packages/xgboost/sklearn.py:889: UserWarning: `callbacks` in `fit` method is
deprecated for better compatibility with scikit-learn, use `callbacks` in
constructor or`set_params` instead.
```

```
warnings.warn(
41it [00:03, 12.33it/s]
Max Depth: 20, Train R^2: 1.00, Test R^2: 0.99: 100%|      | 20/20
[04:21<00:00, 13.08s/it]
```

```
[269]: for result in results:
    print(f"Model with {len(result[5])} variables and max depth {result[10]}.
    ↪get_params()['max_depth']):")
    print(f"Training+Validation R^2: {result[0]}, RMSE: {result[2]}")
    print(f"Testing R^2: {result[1]}, RMSE: {result[3]}")
    print(f"Mean cross-validation score: {result[4]}\n")
    print(result[5])  # Feature importances
    print('\n')
```

```
Model with 1 variables and max depth None:
Training+Validation R^2: 0.89121, RMSE: 6.27465
Testing R^2: 0.90364, RMSE: 6.37987
Mean cross-validation score: 0.88905
```

	Feature Importance
0	CPI_EST_prev 1.0

```
Model with 2 variables and max depth None:
Training+Validation R^2: 0.98971, RMSE: 1.92938
Testing R^2: 0.98616, RMSE: 2.41794
Mean cross-validation score: 0.98429
```

	Feature Importance
0	CPI_EST_avg 0.9625
1	CPI_EST_prev 0.0375

```
Model with 3 variables and max depth None:
Training+Validation R^2: 0.99124, RMSE: 1.78045
Testing R^2: 0.98592, RMSE: 2.43862
Mean cross-validation score: 0.98456
```

	Feature Importance
0	CPI_EST_avg 0.9669
2	CPI_EST_prev 0.0310
1	NE_CON_PRVT_PC_KD 0.0022

```
Model with 4 variables and max depth None:
Training+Validation R^2: 0.99309, RMSE: 1.58153
Testing R^2: 0.98644, RMSE: 2.39371
Mean cross-validation score: 0.98409
```

	Feature	Importance
0	CPI_EST_avg	0.9675
3	CPI_EST_prev	0.0295
1	NE_CON_PRVT_PC_KD	0.0015
2	NY_ADJ_NNTY_PC_CD	0.0015

Model with 5 variables and max depth None:
 Training+Validation R^2: 0.99142, RMSE: 1.76257
 Testing R^2: 0.98562, RMSE: 2.46456
 Mean cross-validation score: 0.98373

	Feature	Importance
0	CPI_EST_avg	0.9648
4	CPI_EST_prev	0.0305
3	NY_GNP_PCAP_CD	0.0022
2	NY_ADJ_NNTY_PC_CD	0.0013
1	NE_CON_PRVT_PC_KD	0.0012

Model with 6 variables and max depth None:
 Training+Validation R^2: 0.99199, RMSE: 1.70267
 Testing R^2: 0.98539, RMSE: 2.48444
 Mean cross-validation score: 0.98405

	Feature	Importance
0	CPI_EST_avg	0.9626
5	CPI_EST_prev	0.0309
4	NY_GDP_PCAP_KD_rel	0.0023
3	NY_GNP_PCAP_CD	0.0017
1	NE_CON_PRVT_PC_KD	0.0013
2	NY_ADJ_NNTY_PC_CD	0.0013

Model with 7 variables and max depth None:
 Training+Validation R^2: 0.99265, RMSE: 1.6308
 Testing R^2: 0.98559, RMSE: 2.46718
 Mean cross-validation score: 0.98387

	Feature	Importance
0	CPI_EST_avg	0.9672
6	CPI_EST_prev	0.0256
4	NY_GDP_PCAP_KD_rel	0.0018
5	NY_GDP_PCAP_KD	0.0016
3	NY_GNP_PCAP_CD	0.0014
1	NE_CON_PRVT_PC_KD	0.0012
2	NY_ADJ_NNTY_PC_CD	0.0011

Model with 8 variables and max depth None:
 Training+Validation R^2: 0.9918, RMSE: 1.72233
 Testing R^2: 0.98546, RMSE: 2.47786
 Mean cross-validation score: 0.98415

	Feature	Importance
0	CPI_EST_avg	0.9647
7	CPI_EST_prev	0.0264
4	NY_GDP_PCAP_KD_rel	0.0019
5	NY_GDP_PCAP_KD	0.0018
6	NY_GDP_PCAP_CD	0.0015
3	NY_GNP_PCAP_CD	0.0014
1	NE_CON_PRVT_PC_KD	0.0012
2	NY_ADJ_NNTY_PC_CD	0.0011

Model with 9 variables and max depth None:
 Training+Validation R^2: 0.993, RMSE: 1.5917
 Testing R^2: 0.98477, RMSE: 2.53653
 Mean cross-validation score: 0.98354

	Feature	Importance
0	CPI_EST_avg	0.9624
8	CPI_EST_prev	0.0265
4	NY_GDP_PCAP_KD_rel	0.0023
5	NY_GDP_PCAP_KD	0.0023
7	IT_NET_USER_ZS	0.0016
3	NY_GNP_PCAP_CD	0.0014
6	NY_GDP_PCAP_CD	0.0014
1	NE_CON_PRVT_PC_KD	0.0011
2	NY_ADJ_NNTY_PC_CD	0.0010

Model with 10 variables and max depth None:
 Training+Validation R^2: 0.99402, RMSE: 1.47055
 Testing R^2: 0.98513, RMSE: 2.50629
 Mean cross-validation score: 0.98368

	Feature	Importance
0	CPI_EST_avg	0.9562
9	CPI_EST_prev	0.0332
5	NY_GDP_PCAP_KD	0.0017
3	NY_GNP_PCAP_CD	0.0016
4	NY_GDP_PCAP_KD_rel	0.0015
7	IT_NET_USER_ZS	0.0014
6	NY_GDP_PCAP_CD	0.0013

8	SP_DYN_LE00_MA_IN	0.0012
2	NY_ADJ_NNTY_PC_CD	0.0010
1	NE_CON_PRVT_PC_KD	0.0009

Model with 11 variables and max depth None:
 Training+Validation R^2: 0.99683, RMSE: 1.07164
 Testing R^2: 0.98512, RMSE: 2.50672
 Mean cross-validation score: 0.98357

	Feature	Importance
0	CPI_EST_avg	0.9602
10	CPI_EST_prev	0.0281
5	NY_GDP_PCAP_KD	0.0021
4	NY_GDP_PCAP_KD_rel	0.0016
9	SP_DYN_LE00_IN	0.0014
3	NY_GNP_PCAP_CD	0.0013
6	NY_GDP_PCAP_CD	0.0013
7	IT_NET_USER_ZS	0.0012
8	SP_DYN_LE00_MA_IN	0.0011
2	NY_ADJ_NNTY_PC_CD	0.0009
1	NE_CON_PRVT_PC_KD	0.0008

Model with 12 variables and max depth None:
 Training+Validation R^2: 0.99324, RMSE: 1.56355
 Testing R^2: 0.98526, RMSE: 2.4951
 Mean cross-validation score: 0.98376

	Feature	Importance
0	CPI_EST_avg	0.9569
11	CPI_EST_prev	0.0293
5	NY_GDP_PCAP_KD	0.0035
4	NY_GDP_PCAP_KD_rel	0.0015
3	NY_GNP_PCAP_CD	0.0014
9	SP_DYN_LE00_IN	0.0014
7	IT_NET_USER_ZS	0.0012
10	SP_POP_80UP_MA_5Y	0.0012
1	NE_CON_PRVT_PC_KD	0.0010
2	NY_ADJ_NNTY_PC_CD	0.0009
6	NY_GDP_PCAP_CD	0.0009
8	SP_DYN_LE00_MA_IN	0.0009

Model with 13 variables and max depth None:
 Training+Validation R^2: 0.99616, RMSE: 1.17908
 Testing R^2: 0.98564, RMSE: 2.46314
 Mean cross-validation score: 0.98367

	Feature	Importance
0	CPI_EST_avg	0.9543
12	CPI_EST_prev	0.0321
5	NY_GDP_PCAP_KD	0.0019
3	NY_GNP_PCAP_CD	0.0015
11	SP_POP_65UP_MA_ZS	0.0015
4	NY_GDP_PCAP_KD_rel	0.0014
9	SP_DYN_LEOO_IN	0.0014
10	SP_POP_80UP_MA_5Y	0.0012
7	IT_NET_USER_ZS	0.0011
8	SP_DYN_LEOO_MA_IN	0.0011
2	NY_ADJ_NNTY_PC_CD	0.0009
1	NE_CON_PRVT_PC_KD	0.0008
6	NY_GDP_PCAP_CD	0.0008

Model with 14 variables and max depth None:

Training+Validation R^2: 0.99463, RMSE: 1.39388

Testing R^2: 0.98554, RMSE: 2.47135

Mean cross-validation score: 0.98381

	Feature	Importance
0	CPI_EST_avg	0.9571
13	CPI_EST_prev	0.0297
12	SP_DYN_LEOO_FE_IN	0.0018
4	NY_GDP_PCAP_KD_rel	0.0015
3	NY_GNP_PCAP_CD	0.0013
5	NY_GDP_PCAP_KD	0.0013
11	SP_POP_65UP_MA_ZS	0.0013
10	SP_POP_80UP_MA_5Y	0.0011
7	IT_NET_USER_ZS	0.0010
8	SP_DYN_LEOO_MA_IN	0.0009
9	SP_DYN_LEOO_IN	0.0009
2	NY_ADJ_NNTY_PC_CD	0.0008
6	NY_GDP_PCAP_CD	0.0008
1	NE_CON_PRVT_PC_KD	0.0007

Model with 15 variables and max depth None:

Training+Validation R^2: 0.99276, RMSE: 1.61865

Testing R^2: 0.98538, RMSE: 2.48473

Mean cross-validation score: 0.98408

	Feature	Importance
0	CPI_EST_avg	0.9600
14	CPI_EST_prev	0.0276
12	SP_DYN_LEOO_FE_IN	0.0015

11	SP_POP_65UP_MA_ZS	0.0012
4	NY_GDP_PCAP_KD_rel	0.0011
3	NY_GNP_PCAP_CD	0.0010
5	NY_GDP_PCAP_KD	0.0010
13	SE_SEC_ENRR	0.0010
8	SP_DYN_LEOO_MA_IN	0.0009
9	SP_DYN_LEOO_IN	0.0009
6	NY_GDP_PCAP_CD	0.0008
7	IT_NET_USER_ZS	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
2	NY_ADJ_NNTY_PC_CD	0.0007
1	NE_CON_PRVT_PC_KD	0.0006

Model with 16 variables and max depth None:

Training+Validation R^2: 0.9967, RMSE: 1.09243

Testing R^2: 0.98561, RMSE: 2.46516

Mean cross-validation score: 0.98406

	Feature	Importance
0	CPI_EST_avg	0.9596
15	CPI_EST_prev	0.0269
12	SP_DYN_LEOO_FE_IN	0.0016
4	NY_GDP_PCAP_KD_rel	0.0014
3	NY_GNP_PCAP_CD	0.0012
11	SP_POP_65UP_MA_ZS	0.0011
9	SP_DYN_LEOO_IN	0.0010
13	SE_SEC_ENRR	0.0010
2	NY_ADJ_NNTY_PC_CD	0.0009
6	NY_GDP_PCAP_CD	0.0009
5	NY_GDP_PCAP_KD	0.0008
7	IT_NET_USER_ZS	0.0008
8	SP_DYN_LEOO_MA_IN	0.0008
14	SP_POP_0014_MA_ZS	0.0008
10	SP_POP_80UP_MA_5Y	0.0007
1	NE_CON_PRVT_PC_KD	0.0006

Model with 17 variables and max depth None:

Training+Validation R^2: 0.9945, RMSE: 1.41104

Testing R^2: 0.98525, RMSE: 2.49589

Mean cross-validation score: 0.98383

	Feature	Importance
0	CPI_EST_avg	0.9567
16	CPI_EST_prev	0.0288
4	NY_GDP_PCAP_KD_rel	0.0015
12	SP_DYN_LEOO_FE_IN	0.0013

11	SP_POP_65UP_MA_ZS	0.0012
13	SE_SEC_ENRR	0.0012
2	NY_ADJ_NNTY_PC_CD	0.0011
3	NY_GNP_PCAP_CD	0.0011
6	NY_GDP_PCAP_CD	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
15	SP_POP_0509_MA_5Y	0.0009
8	SP_DYN_LEOO_MA_IN	0.0009
7	IT_NET_USER_ZS	0.0008
9	SP_DYN_LEOO_IN	0.0008
5	NY_GDP_PCAP_KD	0.0007
14	SP_POP_0014_MA_ZS	0.0007
1	NE_CON_PRVT_PC_KD	0.0006

Model with 18 variables and max depth None:

Training+Validation R^2: 0.9976, RMSE: 0.93182

Testing R^2: 0.98533, RMSE: 2.48903

Mean cross-validation score: 0.98427

	Feature	Importance
0	CPI_EST_avg	0.9530
17	CPI_EST_prev	0.0303
4	NY_GDP_PCAP_KD_rel	0.0019
12	SP_DYN_LEOO_FE_IN	0.0018
3	NY_GNP_PCAP_CD	0.0013
11	SP_POP_65UP_MA_ZS	0.0011
13	SE_SEC_ENRR	0.0010
14	SP_POP_0014_MA_ZS	0.0010
2	NY_ADJ_NNTY_PC_CD	0.0009
5	NY_GDP_PCAP_KD	0.0009
7	IT_NET_USER_ZS	0.0009
8	SP_DYN_LEOO_MA_IN	0.0009
15	SP_POP_0509_MA_5Y	0.0009
9	SP_DYN_LEOO_IN	0.0009
6	NY_GDP_PCAP_CD	0.0008
1	NE_CON_PRVT_PC_KD	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
16	SP_POP_7074_MA_5Y	0.0007

Model with 19 variables and max depth None:

Training+Validation R^2: 0.99559, RMSE: 1.26341

Testing R^2: 0.9847, RMSE: 2.54237

Mean cross-validation score: 0.98423

	Feature	Importance
0	CPI_EST_avg	0.9550

18	CPI_EST_prev	0.0285
4	NY_GDP_PCAP_KD_rel	0.0017
5	NY_GDP_PCAP_KD	0.0016
12	SP_DYN_LE00_FE_IN	0.0013
17	SE_SEC_ENRR_FE	0.0012
11	SP_POP_65UP_MA_ZS	0.0011
10	SP_POP_80UP_MA_5Y	0.0010
3	NY_GNP_PCAP_CD	0.0010
13	SE_SEC_ENRR	0.0009
9	SP_DYN_LE00_IN	0.0009
8	SP_DYN_LE00_MA_IN	0.0008
7	IT_NET_USER_ZS	0.0008
6	NY_GDP_PCAP_CD	0.0008
14	SP_POP_0014_MA_ZS	0.0007
15	SP_POP_0509_MA_5Y	0.0007
16	SP_POP_7074_MA_5Y	0.0007
2	NY_ADJ_NNTY_PC_CD	0.0007
1	NE_CON_PRVT_PC_KD	0.0006

Model with 20 variables and max depth None:

Training+Validation R^2: 0.9943, RMSE: 1.43579

Testing R^2: 0.98485, RMSE: 2.52961

Mean cross-validation score: 0.98396

	Feature	Importance
0	CPI_EST_avg	0.9596
19	CPI_EST_prev	0.0255
4	NY_GDP_PCAP_KD_rel	0.0013
17	SE_SEC_ENRR_FE	0.0012
12	SP_DYN_LE00_FE_IN	0.0011
3	NY_GNP_PCAP_CD	0.0010
10	SP_POP_80UP_MA_5Y	0.0009
5	NY_GDP_PCAP_KD	0.0009
13	SE_SEC_ENRR	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
9	SP_DYN_LE00_IN	0.0008
8	SP_DYN_LE00_MA_IN	0.0007
7	IT_NET_USER_ZS	0.0007
14	SP_POP_0014_MA_ZS	0.0007
15	SP_POP_0509_MA_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
16	SP_POP_7074_MA_5Y	0.0006
18	SP_POP_7579_MA_5Y	0.0006
1	NE_CON_PRVT_PC_KD	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005

Model with 21 variables and max depth None:
 Training+Validation R^2: 0.99683, RMSE: 1.07151
 Testing R^2: 0.98487, RMSE: 2.52796
 Mean cross-validation score: 0.98431

	Feature	Importance
0	CPI_EST_avg	0.9401
20	CPI_EST_prev	0.0347
5	NY_GDP_PCAP_KD	0.0030
4	NY_GDP_PCAP_KD_rel	0.0027
17	SE_SEC_ENRR_FE	0.0023
12	SP_DYN_LE00_FE_IN	0.0016
19	SP_POP_80UP_FE_5Y	0.0015
11	SP_POP_65UP_MA_ZS	0.0012
14	SP_POP_0014_MA_ZS	0.0011
7	IT_NET_USER_ZS	0.0011
18	SP_POP_7579_MA_5Y	0.0011
16	SP_POP_7074_MA_5Y	0.0011
15	SP_POP_0509_MA_5Y	0.0011
10	SP_POP_80UP_MA_5Y	0.0010
13	SE_SEC_ENRR	0.0010
3	NY_GNP_PCAP_CD	0.0010
9	SP_DYN_LE00_IN	0.0009
8	SP_DYN_LE00_MA_IN	0.0009
6	NY_GDP_PCAP_CD	0.0009
1	NE_CON_PRVT_PC_KD	0.0008
2	NY_ADJ_NNTY_PC_CD	0.0008

Model with 22 variables and max depth None:
 Training+Validation R^2: 0.99517, RMSE: 1.32247
 Testing R^2: 0.98452, RMSE: 2.55682
 Mean cross-validation score: 0.98441

	Feature	Importance
0	CPI_EST_avg	0.9518
21	CPI_EST_prev	0.0301
4	NY_GDP_PCAP_KD_rel	0.0019
12	SP_DYN_LE00_FE_IN	0.0014
17	SE_SEC_ENRR_FE	0.0014
5	NY_GDP_PCAP_KD	0.0012
19	SP_POP_80UP_FE_5Y	0.0010
6	NY_GDP_PCAP_CD	0.0009
3	NY_GNP_PCAP_CD	0.0009
14	SP_POP_0014_MA_ZS	0.0008
20	SP_DYN_T065_MA_ZS	0.0008
18	SP_POP_7579_MA_5Y	0.0008
15	SP_POP_0509_MA_5Y	0.0008

11	SP_POP_65UP_MA_ZS	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
13	SE_SEC_ENRR	0.0007
8	SP_DYN_LE00_MA_IN	0.0007
7	IT_NET_USER_ZS	0.0007
1	NE_CON_PRVT_PC_KD	0.0006
9	SP_DYN_LE00_IN	0.0006
16	SP_POP_7074_MA_5Y	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005

Model with 23 variables and max depth None:

Training+Validation R^2: 0.99789, RMSE: 0.87399

Testing R^2: 0.98507, RMSE: 2.51114

Mean cross-validation score: 0.98429

	Feature	Importance
0	CPI_EST_avg	0.9503
22	CPI_EST_prev	0.0297
17	SE_SEC_ENRR_FE	0.0016
4	NY_GDP_PCAP_KD_rel	0.0014
12	SP_DYN_LE00_FE_IN	0.0013
19	SP_POP_80UP_FE_5Y	0.0012
5	NY_GDP_PCAP_KD	0.0011
6	NY_GDP_PCAP_CD	0.0011
9	SP_DYN_LE00_IN	0.0010
7	IT_NET_USER_ZS	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
20	SP_DYN_T065_MA_ZS	0.0009
3	NY_GNP_PCAP_CD	0.0009
13	SE_SEC_ENRR	0.0009
14	SP_POP_0014_MA_ZS	0.0009
21	SP_POP_1014_MA_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
1	NE_CON_PRVT_PC_KD	0.0008
8	SP_DYN_LE00_MA_IN	0.0008
16	SP_POP_7074_MA_5Y	0.0007
15	SP_POP_0509_MA_5Y	0.0007
2	NY_ADJ_NNTY_PC_CD	0.0006

Model with 24 variables and max depth None:

Training+Validation R^2: 0.99633, RMSE: 1.1531

Testing R^2: 0.98506, RMSE: 2.51213

Mean cross-validation score: 0.98413

	Feature	Importance
--	---------	------------

0	CPI_EST_avg	0.9473
23	CPI_EST_prev	0.0301
5	NY_GDP_PCAP_KD	0.0034
4	NY_GDP_PCAP_KD_rel	0.0018
17	SE_SEC_ENRR_FE	0.0014
19	SP_POP_80UP_FE_5Y	0.0014
12	SP_DYN_LE00_FE_IN	0.0012
6	NY_GDP_PCAP_CD	0.0010
22	SE_SEC_ENRR_MA	0.0010
3	NY_GNP_PCAP_CD	0.0010
18	SP_POP_7579_MA_5Y	0.0010
10	SP_POP_80UP_MA_5Y	0.0009
7	IT_NET_USER_ZS	0.0008
21	SP_POP_1014_MA_5Y	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
14	SP_POP_0014_MA_ZS	0.0008
13	SE_SEC_ENRR	0.0007
16	SP_POP_7074_MA_5Y	0.0007
15	SP_POP_0509_MA_5Y	0.0007
8	SP_DYN_LE00_MA_IN	0.0007
1	NE_CON_PRVT_PC_KD	0.0006
20	SP_DYN_TO65_MA_ZS	0.0006
9	SP_DYN_LE00_IN	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0005

Model with 25 variables and max depth None:

Training+Validation R^2: 0.99473, RMSE: 1.38134

Testing R^2: 0.98443, RMSE: 2.56425

Mean cross-validation score: 0.98432

	Feature	Importance
0	CPI_EST_avg	0.9467
24	CPI_EST_prev	0.0312
5	NY_GDP_PCAP_KD	0.0021
4	NY_GDP_PCAP_KD_rel	0.0016
12	SP_DYN_LE00_FE_IN	0.0015
17	SE_SEC_ENRR_FE	0.0015
19	SP_POP_80UP_FE_5Y	0.0014
22	SE_SEC_ENRR_MA	0.0011
6	NY_GDP_PCAP_CD	0.0010
20	SP_DYN_TO65_MA_ZS	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
9	SP_DYN_LE00_IN	0.0008
21	SP_POP_1014_MA_5Y	0.0008
3	NY_GNP_PCAP_CD	0.0008
14	SP_POP_0014_MA_ZS	0.0008
15	SP_POP_0509_MA_5Y	0.0008

7	IT_NET_USER_ZS	0.0008
18	SP_POP_7579_MA_5Y	0.0008
23	SP_POP_0014_TO_ZS	0.0007
13	SE_SEC_ENRR	0.0007
16	SP_POP_7074_MA_5Y	0.0007
11	SP_POP_65UP_MA_ZS	0.0007
8	SP_DYN_LE00_MA_IN	0.0007
2	NY_ADJ_NNTY_PC_CD	0.0006
1	NE_CON_PRVT_PC_KD	0.0005

Model with 26 variables and max depth None:

Training+Validation R^2: 0.99658, RMSE: 1.11215
 Testing R^2: 0.98535, RMSE: 2.48787
 Mean cross-validation score: 0.98401

	Feature	Importance
0	CPI_EST_avg	0.9481
25	CPI_EST_prev	0.0309
4	NY_GDP_PCAP_KD_rel	0.0016
17	SE_SEC_ENRR_FE	0.0014
12	SP_DYN_LE00_FE_IN	0.0014
19	SP_POP_80UP_FE_5Y	0.0011
3	NY_GNP_PCAP_CD	0.0010
14	SP_POP_0014_MA_ZS	0.0009
5	NY_GDP_PCAP_KD	0.0009
6	NY_GDP_PCAP_CD	0.0009
7	IT_NET_USER_ZS	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
20	SP_DYN_TO65_MA_ZS	0.0009
24	SP_POP_65UP_TO_ZS	0.0008
23	SP_POP_0014_TO_ZS	0.0008
22	SE_SEC_ENRR_MA	0.0008
13	SE_SEC_ENRR	0.0008
18	SP_POP_7579_MA_5Y	0.0007
15	SP_POP_0509_MA_5Y	0.0007
21	SP_POP_1014_MA_5Y	0.0007
1	NE_CON_PRVT_PC_KD	0.0006
9	SP_DYN_LE00_IN	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0006
16	SP_POP_7074_MA_5Y	0.0005

Model with 27 variables and max depth None:

Training+Validation R^2: 0.99812, RMSE: 0.82579
 Testing R^2: 0.98582, RMSE: 2.44698

Mean cross-validation score: 0.98386

	Feature	Importance
0	CPI_EST_avg	0.9484
26	CPI_EST_prev	0.0279
4	NY_GDP_PCAP_KD_rel	0.0019
17	SE_SEC_ENRR_FE	0.0016
5	NY_GDP_PCAP_KD	0.0014
12	SP_DYN_LE00_FE_IN	0.0014
19	SP_POP_80UP_FE_5Y	0.0013
20	SP_DYN_TO65_MA_ZS	0.0012
23	SP_POP_0014_TO_ZS	0.0011
22	SE_SEC_ENRR_MA	0.0010
11	SP_POP_65UP_MA_ZS	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
7	IT_NET_USER_ZS	0.0009
24	SP_POP_65UP_TO_ZS	0.0008
1	NE_CON_PRVT_PC_KD	0.0008
3	NY_GNP_PCAP_CD	0.0008
21	SP_POP_1014_MA_5Y	0.0008
25	SP_POP_6569_MA_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0008
14	SP_POP_0014_MA_ZS	0.0008
13	SE_SEC_ENRR	0.0007
16	SP_POP_7074_MA_5Y	0.0007
15	SP_POP_0509_MA_5Y	0.0007
9	SP_DYN_LE00_IN	0.0007
6	NY_GDP_PCAP_CD	0.0007
8	SP_DYN_LE00_MA_IN	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0005

Model with 28 variables and max depth None:

Training+Validation R^2: 0.99808, RMSE: 0.83438

Testing R^2: 0.98632, RMSE: 2.40397

Mean cross-validation score: 0.98451

	Feature	Importance
0	CPI_EST_avg	0.9573
27	CPI_EST_prev	0.0240
12	SP_DYN_LE00_FE_IN	0.0013
19	SP_POP_80UP_FE_5Y	0.0012
17	SE_SEC_ENRR_FE	0.0011
4	NY_GDP_PCAP_KD_rel	0.0010
3	NY_GNP_PCAP_CD	0.0010
5	NY_GDP_PCAP_KD	0.0008
22	SE_SEC_ENRR_MA	0.0008
6	NY_GDP_PCAP_CD	0.0007

7	IT_NET_USER_ZS	0.0007
26	IT_MLT_MAIN_P2	0.0007
23	SP_POP_0014_TO_ZS	0.0007
11	SP_POP_65UP_MA_ZS	0.0007
21	SP_POP_1014_MA_5Y	0.0007
15	SP_POP_0509_MA_5Y	0.0006
24	SP_POP_65UP_TO_ZS	0.0006
20	SP_DYN_TO65_MA_ZS	0.0006
14	SP_POP_0014_MA_ZS	0.0006
18	SP_POP_7579_MA_5Y	0.0006
13	SE_SEC_ENRR	0.0006
9	SP_DYN_LE00_IN	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
16	SP_POP_7074_MA_5Y	0.0005
1	NE_CON_PRVT_PC_KD	0.0005
25	SP_POP_6569_MA_5Y	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005
10	SP_POP_80UP_MA_5Y	0.0004

Model with 29 variables and max depth None:
 Training+Validation R^2: 0.99795, RMSE: 0.86132
 Testing R^2: 0.98625, RMSE: 2.41017
 Mean cross-validation score: 0.98423

	Feature	Importance
0	CPI_EST_avg	0.9517
28	CPI_EST_prev	0.0263
4	NY_GDP_PCAP_KD_rel	0.0014
12	SP_DYN_LE00_FE_IN	0.0012
3	NY_GNP_PCAP_CD	0.0012
17	SE_SEC_ENRR_FE	0.0012
19	SP_POP_80UP_FE_5Y	0.0011
5	NY_GDP_PCAP_KD	0.0011
22	SE_SEC_ENRR_MA	0.0011
20	SP_DYN_TO65_MA_ZS	0.0009
14	SP_POP_0014_MA_ZS	0.0008
7	IT_NET_USER_ZS	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
21	SP_POP_1014_MA_5Y	0.0008
27	SP_POP_0004_MA_5Y	0.0007
26	IT_MLT_MAIN_P2	0.0007
24	SP_POP_65UP_TO_ZS	0.0007
23	SP_POP_0014_TO_ZS	0.0007
15	SP_POP_0509_MA_5Y	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
18	SP_POP_7579_MA_5Y	0.0006

16	SP_POP_7074_MA_5Y	0.0006
1	NE_CON_PRVT_PC_KD	0.0006
13	SE_SEC_ENRR	0.0006
9	SP_DYN_LE00_IN	0.0006
25	SP_POP_6569_MA_5Y	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0005

Model with 30 variables and max depth None:
 Training+Validation R^2: 0.99746, RMSE: 0.9589
 Testing R^2: 0.98577, RMSE: 2.45168
 Mean cross-validation score: 0.98424

	Feature	Importance
0	CPI_EST_avg	0.9538
29	CPI_EST_prev	0.0254
12	SP_DYN_LE00_FE_IN	0.0016
4	NY_GDP_PCAP_KD_rel	0.0012
5	NY_GDP_PCAP_KD	0.0010
17	SE_SEC_ENRR_FE	0.0010
10	SP_POP_80UP_MA_5Y	0.0010
3	NY_GNP_PCAP_CD	0.0010
14	SP_POP_0014_MA_ZS	0.0009
19	SP_POP_80UP_FE_5Y	0.0009
20	SP_DYN_T065_MA_ZS	0.0008
7	IT_NET_USER_ZS	0.0008
22	SE_SEC_ENRR_MA	0.0007
11	SP_POP_65UP_MA_ZS	0.0007
21	SP_POP_1014_MA_5Y	0.0007
28	SP_POP_DPNP_DL	0.0007
25	SP_POP_6569_MA_5Y	0.0006
24	SP_POP_65UP_TO_ZS	0.0006
26	IT_MLT_MAIN_P2	0.0006
27	SP_POP_0004_MA_5Y	0.0006
16	SP_POP_7074_MA_5Y	0.0006
15	SP_POP_0509_MA_5Y	0.0006
18	SP_POP_7579_MA_5Y	0.0006
13	SE_SEC_ENRR	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
6	NY_GDP_PCAP_CD	0.0006
1	NE_CON_PRVT_PC_KD	0.0005
9	SP_DYN_LE00_IN	0.0005
23	SP_POP_0014_TO_ZS	0.0004
2	NY_ADJ_NNTY_PC_CD	0.0004

Model with 31 variables and max depth None:

Training+Validation R^2: 0.99552, RMSE: 1.27343
 Testing R^2: 0.9855, RMSE: 2.47474
 Mean cross-validation score: 0.98467

	Feature	Importance
0	CPI_EST_avg	0.9389
30	CPI_EST_prev	0.0325
5	NY_GDP_PCAP_KD	0.0034
12	SP_DYN_LEOO_FE_IN	0.0018
4	NY_GDP_PCAP_KD_rel	0.0015
19	SP_POP_80UP_FE_5Y	0.0015
17	SE_SEC_ENRR_FE	0.0013
3	NY_GNP_PCAP_CD	0.0012
10	SP_POP_80UP_MA_5Y	0.0010
21	SP_POP_1014_MA_5Y	0.0009
7	IT_NET_USER_ZS	0.0009
29	SP_POP_1014_FE_5Y	0.0009
20	SP_DYN_TO65_MA_ZS	0.0009
23	SP_POP_0014_TO_ZS	0.0009
27	SP_POP_0004_MA_5Y	0.0009
26	IT_MLT_MAIN_P2	0.0009
18	SP_POP_7579_MA_5Y	0.0008
24	SP_POP_65UP_TO_ZS	0.0008
25	SP_POP_6569_MA_5Y	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
28	SP_POP_DPND_DL	0.0008
6	NY_GDP_PCAP_CD	0.0008
16	SP_POP_7074_MA_5Y	0.0007
15	SP_POP_0509_MA_5Y	0.0007
22	SE_SEC_ENRR_MA	0.0007
1	NE_CON_PRVT_PC_KD	0.0007
13	SE_SEC_ENRR	0.0007
9	SP_DYN_LEOO_IN	0.0007
8	SP_DYN_LEOO_MA_IN	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0006
14	SP_POP_0014_MA_ZS	0.0004

Model with 32 variables and max depth None:
 Training+Validation R^2: 0.99537, RMSE: 1.29476
 Testing R^2: 0.98573, RMSE: 2.45537
 Mean cross-validation score: 0.98472

	Feature	Importance
0	CPI_EST_avg	0.9504
31	CPI_EST_prev	0.0245
5	NY_GDP_PCAP_KD	0.0030
4	NY_GDP_PCAP_KD_rel	0.0016

12	SP_DYN_LE00_FE_IN	0.0013
17	SE_SEC_ENRR_FE	0.0011
19	SP_POP_80UP_FE_5Y	0.0011
29	SP_POP_1014_FE_5Y	0.0010
23	SP_POP_0014_TO_ZS	0.0009
3	NY_GNP_PCAP_CD	0.0009
7	IT_NET_USER_ZS	0.0008
30	SP_POP_0014_FE_ZS	0.0008
22	SE_SEC_ENRR_MA	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
21	SP_POP_1014_MA_5Y	0.0008
20	SP_DYN_TO65_MA_ZS	0.0008
18	SP_POP_7579_MA_5Y	0.0007
26	IT_MLT_MAIN_P2	0.0007
27	SP_POP_0004_MA_5Y	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
1	NE_CON_PRVT_PC_KD	0.0006
28	SP_POP_DPND_DL	0.0006
13	SE_SEC_ENRR	0.0006
9	SP_DYN_LE00_IN	0.0006
24	SP_POP_65UP_TO_ZS	0.0006
25	SP_POP_6569_MA_5Y	0.0006
6	NY_GDP_PCAP_CD	0.0006
15	SP_POP_0509_MA_5Y	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0005
14	SP_POP_0014_MA_ZS	0.0005
8	SP_DYN_LE00_MA_IN	0.0005
16	SP_POP_7074_MA_5Y	0.0003

Model with 33 variables and max depth None:

Training+Validation R^2: 0.99894, RMSE: 0.61842

Testing R^2: 0.98651, RMSE: 2.38699

Mean cross-validation score: 0.98467

	Feature	Importance
0	CPI_EST_avg	0.9514
32	CPI_EST_prev	0.0259
19	SP_POP_80UP_FE_5Y	0.0012
4	NY_GDP_PCAP_KD_rel	0.0012
17	SE_SEC_ENRR_FE	0.0011
30	SP_POP_0014_FE_ZS	0.0011
12	SP_DYN_LE00_FE_IN	0.0011
5	NY_GDP_PCAP_KD	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
3	NY_GNP_PCAP_CD	0.0009
7	IT_NET_USER_ZS	0.0008

23	SP_POP_0014_TO_ZS	0.0008
29	SP_POP_1014_FE_5Y	0.0007
28	SP_POP_DPND_DL	0.0007
26	IT_MLT_MAIN_P2	0.0007
25	SP_POP_6569_MA_5Y	0.0007
20	SP_DYN_T065_MA_ZS	0.0007
2	NY_ADJ_NNTY_PC_CD	0.0007
9	SP_DYN_LEOO_IN	0.0007
18	SP_POP_7579_MA_5Y	0.0006
21	SP_POP_1014_MA_5Y	0.0006
22	SE_SEC_ENRR_MA	0.0006
15	SP_POP_0509_MA_5Y	0.0006
27	SP_POP_0004_MA_5Y	0.0006
6	NY_GDP_PCAP_CD	0.0006
31	SP_POP_0509_FE_5Y	0.0006
1	NE_CON_PRVT_PC_KD	0.0005
24	SP_POP_65UP_TO_ZS	0.0005
14	SP_POP_0014_MA_ZS	0.0005
13	SE_SEC_ENRR	0.0005
8	SP_DYN_LEOO_MA_IN	0.0005
16	SP_POP_7074_MA_5Y	0.0004

Model with 34 variables and max depth None:

Training+Validation R^2: 0.99772, RMSE: 0.90827

Testing R^2: 0.98617, RMSE: 2.41716

Mean cross-validation score: 0.9846

	Feature	Importance
0	CPI_EST_avg	0.9256
33	CPI_EST_prev	0.0393
17	SE_SEC_ENRR_FE	0.0019
4	NY_GDP_PCAP_KD_rel	0.0019
19	SP_POP_80UP_FE_5Y	0.0017
12	SP_DYN_LEOO_FE_IN	0.0016
29	SP_POP_1014_FE_5Y	0.0015
5	NY_GDP_PCAP_KD	0.0014
28	SP_POP_DPND_DL	0.0014
22	SE_SEC_ENRR_MA	0.0014
32	SP_POP_1519_MA_5Y	0.0013
3	NY_GNP_PCAP_CD	0.0013
18	SP_POP_7579_MA_5Y	0.0013
27	SP_POP_0004_MA_5Y	0.0012
25	SP_POP_6569_MA_5Y	0.0012
1	NE_CON_PRVT_PC_KD	0.0011
11	SP_POP_65UP_MA_ZS	0.0011
20	SP_DYN_T065_MA_ZS	0.0011
10	SP_POP_80UP_MA_5Y	0.0010

6	NY_GDP_PCAP_CD	0.0010
30	SP_POP_0014_FE_ZS	0.0010
7	IT_NET_USER_ZS	0.0010
24	SP_POP_65UP_TO_ZS	0.0010
13	SE_SEC_ENRR	0.0009
15	SP_POP_0509_MA_5Y	0.0009
16	SP_POP_7074_MA_5Y	0.0009
26	IT_MLT_MAIN_P2	0.0009
23	SP_POP_0014_TO_ZS	0.0008
14	SP_POP_0014_MA_ZS	0.0008
21	SP_POP_1014_MA_5Y	0.0007
8	SP_DYN_LE00_MA_IN	0.0006
31	SP_POP_0509_FE_5Y	0.0006
9	SP_DYN_LE00_IN	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005

Model with 35 variables and max depth None:

Training+Validation R^2: 0.99911, RMSE: 0.56658

Testing R^2: 0.98647, RMSE: 2.39079

Mean cross-validation score: 0.98437

	Feature	Importance
0	CPI_EST_avg	0.9250
34	CPI_EST_prev	0.0394
5	NY_GDP_PCAP_KD	0.0027
12	SP_DYN_LE00_FE_IN	0.0019
17	SE_SEC_ENRR_FE	0.0018
30	SP_POP_0014_FE_ZS	0.0017
4	NY_GDP_PCAP_KD_rel	0.0015
22	SE_SEC_ENRR_MA	0.0014
32	SP_POP_1519_MA_5Y	0.0012
19	SP_POP_80UP_FE_5Y	0.0012
3	NY_GNP_PCAP_CD	0.0012
18	SP_POP_7579_MA_5Y	0.0012
10	SP_POP_80UP_MA_5Y	0.0012
33	SP_POP_65UP_FE_ZS	0.0011
7	IT_NET_USER_ZS	0.0011
20	SP_DYN_TO65_MA_ZS	0.0011
29	SP_POP_1014_FE_5Y	0.0011
27	SP_POP_0004_MA_5Y	0.0011
11	SP_POP_65UP_MA_ZS	0.0010
1	NE_CON_PRVT_PC_KD	0.0010
28	SP_POP_DPND_DL	0.0010
26	IT_MLT_MAIN_P2	0.0009
6	NY_GDP_PCAP_CD	0.0009
16	SP_POP_7074_MA_5Y	0.0009
21	SP_POP_1014_MA_5Y	0.0009

25	SP_POP_6569_MA_5Y	0.0009
23	SP_POP_0014_TO_ZS	0.0008
24	SP_POP_65UP_TO_ZS	0.0008
13	SE_SEC_ENRR	0.0008
15	SP_POP_0509_MA_5Y	0.0007
8	SP_DYN_LE00_MA_IN	0.0006
14	SP_POP_0014_MA_ZS	0.0006
31	SP_POP_0509_FE_5Y	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005
9	SP_DYN_LE00_IN	0.0004

Model with 36 variables and max depth None:

Training+Validation R^2: 0.99758, RMSE: 0.9364

Testing R^2: 0.98598, RMSE: 2.4337

Mean cross-validation score: 0.98484

	Feature	Importance
0	CPI_EST_avg	0.9328
35	CPI_EST_prev	0.0386
12	SP_DYN_LE00_FE_IN	0.0018
4	NY_GDP_PCAP_KD_rel	0.0015
19	SP_POP_80UP_FE_5Y	0.0013
17	SE_SEC_ENRR_FE	0.0013
3	NY_GNP_PCAP_CD	0.0012
29	SP_POP_1014_FE_5Y	0.0011
11	SP_POP_65UP_MA_ZS	0.0011
22	SE_SEC_ENRR_MA	0.0011
32	SP_POP_1519_MA_5Y	0.0010
20	SP_DYN_T065_MA_ZS	0.0010
7	IT_NET_USER_ZS	0.0009
28	SP_POP_DPND_DL	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
27	SP_POP_0004_MA_5Y	0.0009
26	IT_MLT_MAIN_P2	0.0009
34	SP_POP_6064_MA_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0008
1	NE_CON_PRVT_PC_KD	0.0008
21	SP_POP_1014_MA_5Y	0.0007
5	NY_GDP_PCAP_KD	0.0007
23	SP_POP_0014_TO_ZS	0.0007
33	SP_POP_65UP_FE_ZS	0.0007
25	SP_POP_6569_MA_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
16	SP_POP_7074_MA_5Y	0.0007
30	SP_POP_0014_FE_ZS	0.0007
8	SP_DYN_LE00_MA_IN	0.0006
13	SE_SEC_ENRR	0.0006

24	SP_POP_65UP_TO_ZS	0.0006
31	SP_POP_0509_FE_5Y	0.0005
15	SP_POP_0509_MA_5Y	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0004
9	SP_DYN_LE00_IN	0.0003
14	SP_POP_0014_MA_ZS	0.0003

Model with 37 variables and max depth None:

Training+Validation R^2: 0.99743, RMSE: 0.96433

Testing R^2: 0.9858, RMSE: 2.4493

Mean cross-validation score: 0.98475

	Feature	Importance
0	CPI_EST_avg	0.9501
36	CPI_EST_prev	0.0284
12	SP_DYN_LE00_FE_IN	0.0011
17	SE_SEC_ENRR_FE	0.0010
19	SP_POP_80UP_FE_5Y	0.0009
4	NY_GDP_PCAP_KD_rel	0.0009
22	SE_SEC_ENRR_MA	0.0009
32	SP_POP_1519_MA_5Y	0.0008
20	SP_DYN_T065_MA_ZS	0.0008
33	SP_POP_65UP_FE_ZS	0.0007
7	IT_NET_USER_ZS	0.0007
27	SP_POP_0004_MA_5Y	0.0007
35	SP_POP_1519_FE_5Y	0.0007
1	NE_CON_PRVT_PC_KD	0.0007
5	NY_GDP_PCAP_KD	0.0007
30	SP_POP_0014_FE_ZS	0.0007
34	SP_POP_6064_MA_5Y	0.0006
21	SP_POP_1014_MA_5Y	0.0006
29	SP_POP_1014_FE_5Y	0.0006
28	SP_POP_DPND_DL	0.0006
26	IT_MLT_MAIN_P2	0.0006
18	SP_POP_7579_MA_5Y	0.0006
3	NY_GNP_PCAP_CD	0.0006
14	SP_POP_0014_MA_ZS	0.0006
11	SP_POP_65UP_MA_ZS	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
13	SE_SEC_ENRR	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
25	SP_POP_6569_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
15	SP_POP_0509_MA_5Y	0.0004
9	SP_DYN_LE00_IN	0.0004
6	NY_GDP_PCAP_CD	0.0004
2	NY_ADJ_NNTY_PC_CD	0.0004

23	SP_POP_0014_TO_ZS	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
31	SP_POP_0509_FE_5Y	0.0003

Model with 38 variables and max depth None:
 Training+Validation R^2: 0.99748, RMSE: 0.95463
 Testing R^2: 0.98592, RMSE: 2.43838
 Mean cross-validation score: 0.98489

	Feature	Importance
0	CPI_EST_avg	0.9329
37	CPI_EST_prev	0.0358
12	SP_DYN_LE00_FE_IN	0.0019
17	SE_SEC_ENRR_FE	0.0017
4	NY_GDP_PCAP_KD_rel	0.0015
20	SP_DYN_TO65_MA_ZS	0.0012
19	SP_POP_80UP_FE_5Y	0.0012
5	NY_GDP_PCAP_KD	0.0012
23	SP_POP_0014_TO_ZS	0.0011
22	SE_SEC_ENRR_MA	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
33	SP_POP_65UP_FE_ZS	0.0010
29	SP_POP_1014_FE_5Y	0.0010
36	SP_DYN_CBRT_IN	0.0009
35	SP_POP_1519_FE_5Y	0.0009
32	SP_POP_1519_MA_5Y	0.0009
15	SP_POP_0509_MA_5Y	0.0009
27	SP_POP_0004_MA_5Y	0.0009
3	NY_GNP_PCAP_CD	0.0009
28	SP_POP_DPND_DL	0.0008
24	SP_POP_65UP_TO_ZS	0.0008
7	IT_NET_USER_ZS	0.0008
14	SP_POP_0014_MA_ZS	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
21	SP_POP_1014_MA_5Y	0.0007
1	NE_CON_PRVT_PC_KD	0.0007
26	IT_MLT_MAIN_P2	0.0007
18	SP_POP_7579_MA_5Y	0.0007
25	SP_POP_6569_MA_5Y	0.0006
16	SP_POP_7074_MA_5Y	0.0006
13	SE_SEC_ENRR	0.0006
34	SP_POP_6064_MA_5Y	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0006
8	SP_DYN_LE00_MA_IN	0.0005
6	NY_GDP_PCAP_CD	0.0005
9	SP_DYN_LE00_IN	0.0005
30	SP_POP_0014_FE_ZS	0.0004

31 SP_POP_0509_FE_5Y 0.0004

Model with 39 variables and max depth None:
Training+Validation R^2: 0.99903, RMSE: 0.59366
Testing R^2: 0.98632, RMSE: 2.40372
Mean cross-validation score: 0.98539

	Feature	Importance
0	CPI_EST_avg	0.9327
38	CPI_EST_prev	0.0349
12	SP_DYN_LE00_FE_IN	0.0019
17	SE_SEC_ENRR_FE	0.0017
4	NY_GDP_PCAP_KD_rel	0.0015
20	SP_DYN_T065_MA_ZS	0.0012
35	SP_POP_1519_FE_5Y	0.0011
33	SP_POP_65UP_FE_ZS	0.0011
31	SP_POP_0509_FE_5Y	0.0011
10	SP_POP_80UP_MA_5Y	0.0011
22	SE_SEC_ENRR_MA	0.0011
36	SP_DYN_CBRT_IN	0.0010
25	SP_POP_6569_MA_5Y	0.0010
23	SP_POP_0014_TO_ZS	0.0010
18	SP_POP_7579_MA_5Y	0.0010
19	SP_POP_80UP_FE_5Y	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
5	NY_GDP_PCAP_KD	0.0009
21	SP_POP_1014_MA_5Y	0.0009
37	SP_POP_5559_MA_5Y	0.0008
7	IT_NET_USER_ZS	0.0008
3	NY_GNP_PCAP_CD	0.0008
32	SP_POP_1519_MA_5Y	0.0008
30	SP_POP_0014_FE_ZS	0.0008
26	IT_MLT_MAIN_P2	0.0008
27	SP_POP_0004_MA_5Y	0.0007
8	SP_DYN_LE00_MA_IN	0.0007
29	SP_POP_1014_FE_5Y	0.0007
9	SP_DYN_LE00_IN	0.0007
24	SP_POP_65UP_TO_ZS	0.0007
15	SP_POP_0509_MA_5Y	0.0006
6	NY_GDP_PCAP_CD	0.0006
13	SE_SEC_ENRR	0.0006
1	NE_CON_PRVT_PC_KD	0.0006
34	SP_POP_6064_MA_5Y	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0005
28	SP_POP_DPND_DL	0.0004
16	SP_POP_7074_MA_5Y	0.0004
14	SP_POP_0014_MA_ZS	0.0003

Model with 40 variables and max depth None:
 Training+Validation R^2: 0.99709, RMSE: 1.02541
 Testing R^2: 0.98594, RMSE: 2.43737
 Mean cross-validation score: 0.98514

	Feature	Importance
0	CPI_EST_avg	0.9429
39	CPI_EST_prev	0.0306
12	SP_DYN_LE00_FE_IN	0.0013
20	SP_DYN_TO65_MA_ZS	0.0011
4	NY_GDP_PCAP_KD_rel	0.0011
17	SE_SEC_ENRR_FE	0.0010
23	SP_POP_0014_TO_ZS	0.0010
22	SE_SEC_ENRR_MA	0.0010
19	SP_POP_80UP_FE_5Y	0.0010
24	SP_POP_65UP_TO_ZS	0.0009
36	SP_DYN_CBRT_IN	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
3	NY_GNP_PCAP_CD	0.0008
18	SP_POP_7579_MA_5Y	0.0008
33	SP_POP_65UP_FE_ZS	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
35	SP_POP_1519_FE_5Y	0.0008
5	NY_GDP_PCAP_KD	0.0008
32	SP_POP_1519_MA_5Y	0.0007
34	SP_POP_6064_MA_5Y	0.0007
7	IT_NET_USER_ZS	0.0007
31	SP_POP_0509_FE_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
29	SP_POP_1014_FE_5Y	0.0006
38	SP_POP_0004_FE_5Y	0.0006
27	SP_POP_0004_MA_5Y	0.0006
21	SP_POP_1014_MA_5Y	0.0006
25	SP_POP_6569_MA_5Y	0.0006
30	SP_POP_0014_FE_ZS	0.0005
15	SP_POP_0509_MA_5Y	0.0005
13	SE_SEC_ENRR	0.0005
9	SP_DYN_LE00_IN	0.0005
8	SP_DYN_LE00_MA_IN	0.0005
37	SP_POP_5559_MA_5Y	0.0005
28	SP_POP_DPNP_OL	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
6	NY_GDP_PCAP_CD	0.0004
2	NY_ADJ_NNTY_PC_CD	0.0004
16	SP_POP_7074_MA_5Y	0.0003
14	SP_POP_0014_MA_ZS	0.0002

Model with 41 variables and max depth None:
 Training+Validation R^2: 0.9983, RMSE: 0.78427
 Testing R^2: 0.98645, RMSE: 2.39213
 Mean cross-validation score: 0.98486

	Feature	Importance
0	CPI_EST_avg	0.9334
40	CPI_EST_prev	0.0310
4	NY_GDP_PCAP_KD_rel	0.0020
17	SE_SEC_ENRR_FE	0.0016
12	SP_DYN_LE00_FE_IN	0.0014
36	SP_DYN_CBRT_IN	0.0013
20	SP_DYN_T065_MA_ZS	0.0013
38	SP_POP_0004_FE_5Y	0.0012
22	SE_SEC_ENRR_MA	0.0012
39	SH_DYN_NMRT	0.0011
10	SP_POP_80UP_MA_5Y	0.0011
33	SP_POP_65UP_FE_ZS	0.0011
5	NY_GDP_PCAP_KD	0.0011
19	SP_POP_80UP_FE_5Y	0.0011
7	IT_NET_USER_ZS	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
35	SP_POP_1519_FE_5Y	0.0010
14	SP_POP_0014_MA_ZS	0.0010
31	SP_POP_0509_FE_5Y	0.0010
18	SP_POP_7579_MA_5Y	0.0010
3	NY_GNP_PCAP_CD	0.0010
23	SP_POP_0014_TO_ZS	0.0009
21	SP_POP_1014_MA_5Y	0.0009
15	SP_POP_0509_MA_5Y	0.0009
32	SP_POP_1519_MA_5Y	0.0008
6	NY_GDP_PCAP_CD	0.0008
37	SP_POP_5559_MA_5Y	0.0008
26	IT_MLT_MAIN_P2	0.0007
28	SP_POP_DPND_OL	0.0007
34	SP_POP_6064_MA_5Y	0.0007
8	SP_DYN_LE00_MA_IN	0.0006
9	SP_DYN_LE00_IN	0.0006
13	SE_SEC_ENRR	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0006
30	SP_POP_0014_FE_ZS	0.0006
29	SP_POP_1014_FE_5Y	0.0006
27	SP_POP_0004_MA_5Y	0.0006
25	SP_POP_6569_MA_5Y	0.0006
1	NE_CON_PRVT_PC_KD	0.0005
16	SP_POP_7074_MA_5Y	0.0004

24 SP_POP_65UP_TO_ZS 0.0003

Model with 42 variables and max depth None:
Training+Validation R^2: 0.99901, RMSE: 0.59729
Testing R^2: 0.98648, RMSE: 2.38984
Mean cross-validation score: 0.98484

	Feature	Importance
0	CPI_EST_avg	0.9262
41	CPI_EST_prev	0.0328
12	SP_DYN_LEOO_FE_IN	0.0020
4	NY_GDP_PCAP_KD_rel	0.0017
17	SE_SEC_ENRR_FE	0.0016
20	SP_DYN_TO65_MA_ZS	0.0016
36	SP_DYN_CBRT_IN	0.0015
35	SP_POP_1519_FE_5Y	0.0015
21	SP_POP_1014_MA_5Y	0.0014
39	SH_DYN_NMRT	0.0014
3	NY_GNP_PCAP_CD	0.0014
19	SP_POP_80UP_FE_5Y	0.0014
33	SP_POP_65UP_FE_ZS	0.0013
10	SP_POP_80UP_MA_5Y	0.0013
11	SP_POP_65UP_MA_ZS	0.0013
22	SE_SEC_ENRR_MA	0.0012
40	SP_POP_5054_MA_5Y	0.0011
38	SP_POP_0004_FE_5Y	0.0011
32	SP_POP_1519_MA_5Y	0.0011
18	SP_POP_7579_MA_5Y	0.0010
5	NY_GDP_PCAP_KD	0.0010
37	SP_POP_5559_MA_5Y	0.0010
7	IT_NET_USER_ZS	0.0010
30	SP_POP_0014_FE_ZS	0.0010
14	SP_POP_0014_MA_ZS	0.0009
2	NY_ADJ_NNTY_PC_CD	0.0008
6	NY_GDP_PCAP_CD	0.0008
34	SP_POP_6064_MA_5Y	0.0008
28	SP_POP_DPNP_DL	0.0008
15	SP_POP_0509_MA_5Y	0.0008
31	SP_POP_0509_FE_5Y	0.0007
29	SP_POP_1014_FE_5Y	0.0007
26	IT_MLT_MAIN_P2	0.0007
25	SP_POP_6569_MA_5Y	0.0007
9	SP_DYN_LEOO_IN	0.0007
1	NE_CON_PRVT_PC_KD	0.0007
27	SP_POP_0004_MA_5Y	0.0006
8	SP_DYN_LEOO_MA_IN	0.0006
23	SP_POP_0014_TO_ZS	0.0006

13	SE_SEC_ENRR	0.0006
24	SP_POP_65UP_TO_ZS	0.0005
16	SP_POP_7074_MA_5Y	0.0005

Model with 43 variables and max depth None:
 Training+Validation R^2: 0.99762, RMSE: 0.92882
 Testing R^2: 0.98658, RMSE: 2.38124
 Mean cross-validation score: 0.98483

	Feature	Importance
0	CPI_EST_avg	0.9328
42	CPI_EST_prev	0.0302
5	NY_GDP_PCAP_KD	0.0019
17	SE_SEC_ENRR_FE	0.0018
4	NY_GDP_PCAP_KD_rel	0.0016
19	SP_POP_80UP_FE_5Y	0.0016
12	SP_DYN_LEOO_FE_IN	0.0015
39	SH_DYN_NMRT	0.0014
14	SP_POP_0014_MA_ZS	0.0013
31	SP_POP_0509_FE_5Y	0.0012
41	SP_POP_DPND_YG	0.0012
20	SP_DYN_T065_MA_ZS	0.0011
30	SP_POP_0014_FE_ZS	0.0011
22	SE_SEC_ENRR_MA	0.0010
35	SP_POP_1519_FE_5Y	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
33	SP_POP_65UP_FE_ZS	0.0009
3	NY_GNP_PCAP_CD	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
18	SP_POP_7579_MA_5Y	0.0009
38	SP_POP_0004_FE_5Y	0.0009
37	SP_POP_5559_MA_5Y	0.0009
32	SP_POP_1519_MA_5Y	0.0008
6	NY_GDP_PCAP_CD	0.0007
36	SP_DYN_CBRT_IN	0.0007
7	IT_NET_USER_ZS	0.0007
25	SP_POP_6569_MA_5Y	0.0007
26	IT_MLT_MAIN_P2	0.0007
34	SP_POP_6064_MA_5Y	0.0007
28	SP_POP_DPND_DL	0.0007
9	SP_DYN_LEOO_IN	0.0007
40	SP_POP_5054_MA_5Y	0.0006
21	SP_POP_1014_MA_5Y	0.0006
29	SP_POP_1014_FE_5Y	0.0006
27	SP_POP_0004_MA_5Y	0.0006
1	NE_CON_PRVT_PC_KD	0.0006
15	SP_POP_0509_MA_5Y	0.0006

13	SE_SEC_ENRR	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
24	SP_POP_65UP_TO_ZS	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0004
16	SP_POP_7074_MA_5Y	0.0003
23	SP_POP_0014_TO_ZS	0.0002

Model with 44 variables and max depth None:

Training+Validation R^2: 0.99839, RMSE: 0.76395

Testing R^2: 0.98669, RMSE: 2.37092

Mean cross-validation score: 0.98513

	Feature	Importance
0	CPI_EST_avg	0.9332
43	CPI_EST_prev	0.0292
4	NY_GDP_PCAP_KD_rel	0.0024
38	SP_POP_0004_FE_5Y	0.0016
17	SE_SEC_ENRR_FE	0.0016
12	SP_DYN_LE00_FE_IN	0.0014
19	SP_POP_80UP_FE_5Y	0.0014
39	SH_DYN_NMRT	0.0013
10	SP_POP_80UP_MA_5Y	0.0012
36	SP_DYN_CBRT_IN	0.0011
3	NY_GNP_PCAP_CD	0.0011
21	SP_POP_1014_MA_5Y	0.0011
20	SP_DYN_TO65_MA_ZS	0.0011
5	NY_GDP_PCAP_KD	0.0011
42	SP_POP_7074_FE_5Y	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
35	SP_POP_1519_FE_5Y	0.0010
40	SP_POP_5054_MA_5Y	0.0010
31	SP_POP_0509_FE_5Y	0.0009
37	SP_POP_5559_MA_5Y	0.0009
7	IT_NET_USER_ZS	0.0009
32	SP_POP_1519_MA_5Y	0.0009
26	IT_MLT_MAIN_P2	0.0008
22	SE_SEC_ENRR_MA	0.0008
9	SP_DYN_LE00_IN	0.0008
15	SP_POP_0509_MA_5Y	0.0008
27	SP_POP_0004_MA_5Y	0.0007
29	SP_POP_1014_FE_5Y	0.0007
1	NE_CON_PRVT_PC_KD	0.0007
33	SP_POP_65UP_FE_ZS	0.0007
13	SE_SEC_ENRR	0.0007
24	SP_POP_65UP_TO_ZS	0.0006
25	SP_POP_6569_MA_5Y	0.0006
14	SP_POP_0014_MA_ZS	0.0006

28	SP_POP_DPND_OL	0.0006
6	NY_GDP_PCAP_CD	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
18	SP_POP_7579_MA_5Y	0.0006
41	SP_POP_DPND_YG	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005
34	SP_POP_6064_MA_5Y	0.0005
23	SP_POP_0014_TO_ZS	0.0005
30	SP_POP_0014_FE_ZS	0.0004
16	SP_POP_7074_MA_5Y	0.0003

Model with 45 variables and max depth None:

Training+Validation R^2: 0.99677, RMSE: 1.0809

Testing R^2: 0.98577, RMSE: 2.45206

Mean cross-validation score: 0.98493

	Feature	Importance
0	CPI_EST_avg	0.9444
44	CPI_EST_prev	0.0259
30	SP_POP_0014_FE_ZS	0.0016
4	NY_GDP_PCAP_KD_rel	0.0014
12	SP_DYN_LE00_FE_IN	0.0013
19	SP_POP_80UP_FE_5Y	0.0012
17	SE_SEC_ENRR_FE	0.0012
41	SP_POP_DPND_YG	0.0010
39	SH_DYN_NMRT	0.0009
3	NY_GNP_PCAP_CD	0.0009
18	SP_POP_7579_MA_5Y	0.0009
9	SP_DYN_LE00_IN	0.0008
20	SP_DYN_TO65_MA_ZS	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
37	SP_POP_5559_MA_5Y	0.0008
27	SP_POP_0004_MA_5Y	0.0007
22	SE_SEC_ENRR_MA	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
5	NY_GDP_PCAP_KD	0.0007
40	SP_POP_5054_MA_5Y	0.0006
42	SP_POP_7074_FE_5Y	0.0006
35	SP_POP_1519_FE_5Y	0.0006
38	SP_POP_0004_FE_5Y	0.0006
32	SP_POP_1519_MA_5Y	0.0006
31	SP_POP_0509_FE_5Y	0.0006
29	SP_POP_1014_FE_5Y	0.0006
7	IT_NET_USER_ZS	0.0006
21	SP_POP_1014_MA_5Y	0.0006
43	SP_DYN_AMRT_MA	0.0006
36	SP_DYN_CBRT_IN	0.0006

26	IT_MLT_MAIN_P2	0.0005
6	NY_GDP_PCAP_CD	0.0005
25	SP_POP_6569_MA_5Y	0.0005
24	SP_POP_65UP_TO_ZS	0.0005
8	SP_DYN_LE00_MA_IN	0.0005
1	NE_CON_PRVT_PC_KD	0.0005
33	SP_POP_65UP_FE_ZS	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0004
34	SP_POP_6064_MA_5Y	0.0004
28	SP_POP_DPND_DL	0.0004
16	SP_POP_7074_MA_5Y	0.0004
15	SP_POP_0509_MA_5Y	0.0004
13	SE_SEC_ENRR	0.0004
23	SP_POP_0014_TO_ZS	0.0004
14	SP_POP_0014_MA_ZS	0.0003

Model with 46 variables and max depth None:

Training+Validation R^2: 0.99933, RMSE: 0.49224

Testing R^2: 0.98639, RMSE: 2.3978

Mean cross-validation score: 0.98492

	Feature	Importance
0	CPI_EST_avg	0.9364
45	CPI_EST_prev	0.0280
17	SE_SEC_ENRR_FE	0.0017
12	SP_DYN_LE00_FE_IN	0.0016
39	SH_DYN_NMRT	0.0014
44	EG_USE_ELEC_KH_PC	0.0013
5	NY_GDP_PCAP_KD	0.0012
41	SP_POP_DPND_YG	0.0012
38	SP_POP_0004_FE_5Y	0.0011
4	NY_GDP_PCAP_KD_rel	0.0011
19	SP_POP_80UP_FE_5Y	0.0010
31	SP_POP_0509_FE_5Y	0.0010
35	SP_POP_1519_FE_5Y	0.0010
22	SE_SEC_ENRR_MA	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
37	SP_POP_5559_MA_5Y	0.0009
36	SP_DYN_CBRT_IN	0.0009
32	SP_POP_1519_MA_5Y	0.0009
43	SP_DYN_AMRT_MA	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
30	SP_POP_0014_FE_ZS	0.0008
23	SP_POP_0014_TO_ZS	0.0008
33	SP_POP_65UP_FE_ZS	0.0008
18	SP_POP_7579_MA_5Y	0.0008
42	SP_POP_7074_FE_5Y	0.0008

27	SP_POP_0004_MA_5Y	0.0007
28	SP_POP_DPND_DL	0.0007
24	SP_POP_65UP_TO_ZS	0.0007
25	SP_POP_6569_MA_5Y	0.0007
20	SP_DYN_TO65_MA_ZS	0.0007
7	IT_NET_USER_ZS	0.0007
26	IT_MLT_MAIN_P2	0.0006
3	NY_GNP_PCAP_CD	0.0006
6	NY_GDP_PCAP_CD	0.0006
21	SP_POP_1014_MA_5Y	0.0006
34	SP_POP_6064_MA_5Y	0.0006
16	SP_POP_7074_MA_5Y	0.0006
29	SP_POP_1014_FE_5Y	0.0006
14	SP_POP_0014_MA_ZS	0.0006
9	SP_DYN_LE00_IN	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005
13	SE_SEC_ENRR	0.0005
1	NE_CON_PRVT_PC_KD	0.0005
8	SP_DYN_LE00_MA_IN	0.0004
40	SP_POP_5054_MA_5Y	0.0003
15	SP_POP_0509_MA_5Y	0.0003

Model with 47 variables and max depth None:

Training+Validation R^2: 0.99935, RMSE: 0.4867

Testing R^2: 0.98625, RMSE: 2.41009

Mean cross-validation score: 0.98448

	Feature	Importance
0	CPI_EST_avg	0.9187
46	CPI_EST_prev	0.0363
4	NY_GDP_PCAP_KD_rel	0.0022
19	SP_POP_80UP_FE_5Y	0.0022
12	SP_DYN_LE00_FE_IN	0.0018
17	SE_SEC_ENRR_FE	0.0018
45	SP_POP_6569_FE_5Y	0.0017
44	EG_USE_ELEC_KH_PC	0.0017
39	SH_DYN_NMRT	0.0015
22	SE_SEC_ENRR_MA	0.0014
41	SP_POP_DPND_YG	0.0013
11	SP_POP_65UP_MA_ZS	0.0012
42	SP_POP_7074_FE_5Y	0.0012
10	SP_POP_80UP_MA_5Y	0.0011
3	NY_GNP_PCAP_CD	0.0011
38	SP_POP_0004_FE_5Y	0.0010
43	SP_DYN_AMRT_MA	0.0010
35	SP_POP_1519_FE_5Y	0.0010
34	SP_POP_6064_MA_5Y	0.0010

7	IT_NET_USER_ZS	0.0010
36	SP_DYN_CBRT_IN	0.0010
32	SP_POP_1519_MA_5Y	0.0009
23	SP_POP_0014_TO_ZS	0.0009
30	SP_POP_0014_FE_ZS	0.0009
21	SP_POP_1014_MA_5Y	0.0009
20	SP_DYN_T065_MA_ZS	0.0009
9	SP_DYN_LEOO_IN	0.0009
5	NY_GDP_PCAP_KD	0.0009
37	SP_POP_5559_MA_5Y	0.0008
31	SP_POP_0509_FE_5Y	0.0008
40	SP_POP_5054_MA_5Y	0.0008
2	NY_ADJ_NNTY_PC_CD	0.0008
26	IT_MLT_MAIN_P2	0.0008
18	SP_POP_7579_MA_5Y	0.0008
16	SP_POP_7074_MA_5Y	0.0008
13	SE_SEC_ENRR	0.0008
8	SP_DYN_LEOO_MA_IN	0.0008
29	SP_POP_1014_FE_5Y	0.0007
27	SP_POP_0004_MA_5Y	0.0007
25	SP_POP_6569_MA_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
33	SP_POP_65UP_FE_ZS	0.0005
1	NE_CON_PRVT_PC_KD	0.0005
15	SP_POP_0509_MA_5Y	0.0005
14	SP_POP_0014_MA_ZS	0.0004
28	SP_POP_DPND_DL	0.0003
24	SP_POP_65UP_TO_ZS	0.0003

Model with 48 variables and max depth None:

Training+Validation R^2: 0.99875, RMSE: 0.6732

Testing R^2: 0.98634, RMSE: 2.40246

Mean cross-validation score: 0.98457

	Feature	Importance
0	CPI_EST_avg	0.9313
47	CPI_EST_prev	0.0303
5	NY_GDP_PCAP_KD	0.0022
44	EG_USE_ELEC_KH_PC	0.0017
39	SH_DYN_NMRT	0.0014
4	NY_GDP_PCAP_KD_rel	0.0014
12	SP_DYN_LEOO_FE_IN	0.0014
17	SE_SEC_ENRR_FE	0.0014
38	SP_POP_0004_FE_5Y	0.0012
46	SP_DYN_T065_FE_ZS	0.0012
35	SP_POP_1519_FE_5Y	0.0012
19	SP_POP_80UP_FE_5Y	0.0012

42	SP_POP_7074_FE_5Y	0.0012
45	SP_POP_6569_FE_5Y	0.0011
22	SE_SEC_ENRR_MA	0.0011
32	SP_POP_1519_MA_5Y	0.0010
11	SP_POP_65UP_MA_ZS	0.0009
27	SP_POP_0004_MA_5Y	0.0009
28	SP_POP_DPND_OL	0.0009
23	SP_POP_0014_TO_ZS	0.0008
7	IT_NET_USER_ZS	0.0008
31	SP_POP_0509_FE_5Y	0.0007
40	SP_POP_5054_MA_5Y	0.0007
18	SP_POP_7579_MA_5Y	0.0007
20	SP_DYN_TO65_MA_ZS	0.0007
21	SP_POP_1014_MA_5Y	0.0007
43	SP_DYN_AMRT_MA	0.0007
36	SP_DYN_CBRT_IN	0.0007
33	SP_POP_65UP_FE_ZS	0.0007
2	NY_ADJ_NNTY_PC_CD	0.0007
34	SP_POP_6064_MA_5Y	0.0006
25	SP_POP_6569_MA_5Y	0.0006
30	SP_POP_0014_FE_ZS	0.0006
26	IT_MLT_MAIN_P2	0.0006
3	NY_GNP_PCAP_CD	0.0006
13	SE_SEC_ENRR	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
29	SP_POP_1014_FE_5Y	0.0005
37	SP_POP_5559_MA_5Y	0.0005
14	SP_POP_0014_MA_ZS	0.0005
41	SP_POP_DPND_YG	0.0005
9	SP_DYN_LE00_IN	0.0005
6	NY_GDP_PCAP_CD	0.0005
24	SP_POP_65UP_TO_ZS	0.0005
1	NE_CON_PRVT_PC_KD	0.0004
16	SP_POP_7074_MA_5Y	0.0003
15	SP_POP_0509_MA_5Y	0.0003

Model with 49 variables and max depth None:

Training+Validation R^2: 0.99832, RMSE: 0.78043

Testing R^2: 0.98634, RMSE: 2.402

Mean cross-validation score: 0.98466

	Feature	Importance
0	CPI_EST_avg	0.9468
48	CPI_EST_prev	0.0256
5	NY_GDP_PCAP_KD	0.0016
38	SP_POP_0004_FE_5Y	0.0012

4	NY_GDP_PCAP_KD_rel	0.0011
17	SE_SEC_ENRR_FE	0.0011
44	EG_USE_ELEC_KH_PC	0.0011
12	SP_DYN_LEOO_FE_IN	0.0010
45	SP_POP_6569_FE_5Y	0.0010
39	SH_DYN_NMRT	0.0010
19	SP_POP_80UP_FE_5Y	0.0009
42	SP_POP_7074_FE_5Y	0.0008
46	SP_DYN_TO65_FE_ZS	0.0008
32	SP_POP_1519_MA_5Y	0.0007
28	SP_POP_DPNP_DL	0.0007
11	SP_POP_65UP_MA_ZS	0.0006
8	SP_DYN_LEOO_MA_IN	0.0006
35	SP_POP_1519_FE_5Y	0.0006
20	SP_DYN_TO65_MA_ZS	0.0006
27	SP_POP_0004_MA_5Y	0.0006
6	NY_GDP_PCAP_CD	0.0006
18	SP_POP_7579_MA_5Y	0.0006
30	SP_POP_0014_FE_ZS	0.0005
37	SP_POP_5559_MA_5Y	0.0005
29	SP_POP_1014_FE_5Y	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005
40	SP_POP_5054_MA_5Y	0.0005
22	SE_SEC_ENRR_MA	0.0005
21	SP_POP_1014_MA_5Y	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
7	IT_NET_USER_ZS	0.0005
43	SP_DYN_AMRT_MA	0.0004
47	SP_DYN_IMRT_MA_IN	0.0004
36	SP_DYN_CBRT_IN	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
34	SP_POP_6064_MA_5Y	0.0004
26	IT_MLT_MAIN_P2	0.0004
3	NY_GNP_PCAP_CD	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
13	SE_SEC_ENRR	0.0004
33	SP_POP_65UP_FE_ZS	0.0003
31	SP_POP_0509_FE_5Y	0.0003
23	SP_POP_0014_TO_ZS	0.0003
15	SP_POP_0509_MA_5Y	0.0003
9	SP_DYN_LEOO_IN	0.0003
25	SP_POP_6569_MA_5Y	0.0003
41	SP_POP_DPNP_YG	0.0002
16	SP_POP_7074_MA_5Y	0.0002
14	SP_POP_0014_MA_ZS	0.0002

Model with 50 variables and max depth None:

Training+Validation R^2: 0.99802, RMSE: 0.84753
 Testing R^2: 0.98585, RMSE: 2.44467
 Mean cross-validation score: 0.98452

	Feature	Importance
0	CPI_EST_avg	0.9285
49	CPI_EST_prev	0.0303
17	SE_SEC_ENRR_FE	0.0017
39	SH_DYN_NMRT	0.0016
44	EG_USE_ELEC_KH_PC	0.0015
4	NY_GDP_PCAP_KD_rel	0.0014
12	SP_DYN_LEOO_FE_IN	0.0013
5	NY_GDP_PCAP_KD	0.0012
46	SP_DYN_T065_FE_ZS	0.0012
45	SP_POP_6569_FE_5Y	0.0012
19	SP_POP_80UP_FE_5Y	0.0012
33	SP_POP_65UP_FE_ZS	0.0012
38	SP_POP_0004_FE_5Y	0.0011
18	SP_POP_7579_MA_5Y	0.0011
22	SE_SEC_ENRR_MA	0.0010
20	SP_DYN_T065_MA_ZS	0.0010
42	SP_POP_7074_FE_5Y	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
35	SP_POP_1519_FE_5Y	0.0010
27	SP_POP_0004_MA_5Y	0.0010
36	SP_DYN_CBRT_IN	0.0010
32	SP_POP_1519_MA_5Y	0.0009
41	SP_POP_DPND_YG	0.0009
21	SP_POP_1014_MA_5Y	0.0009
25	SP_POP_6569_MA_5Y	0.0008
23	SP_POP_0014_TO_ZS	0.0008
8	SP_DYN_LEOO_MA_IN	0.0008
7	IT_NET_USER_ZS	0.0008
6	NY_GDP_PCAP_CD	0.0008
29	SP_POP_1014_FE_5Y	0.0008
2	NY_ADJ_NNTY_PC_CD	0.0007
26	IT_MLT_MAIN_P2	0.0007
28	SP_POP_DPND_DL	0.0007
3	NY_GNP_PCAP_CD	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
30	SP_POP_0014_FE_ZS	0.0006
48	SP_DYN_IMRT_IN	0.0006
47	SP_DYN_IMRT_MA_IN	0.0006
43	SP_DYN_AMRT_MA	0.0006
40	SP_POP_5054_MA_5Y	0.0006
37	SP_POP_5559_MA_5Y	0.0006
16	SP_POP_7074_MA_5Y	0.0005
13	SE_SEC_ENRR	0.0005

31	SP_POP_0509_FE_5Y	0.0005
24	SP_POP_65UP_TO_ZS	0.0004
9	SP_DYN_LE00_IN	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
34	SP_POP_6064_MA_5Y	0.0004
15	SP_POP_0509_MA_5Y	0.0003
14	SP_POP_0014_MA_ZS	0.0003

Model with 51 variables and max depth None:
 Training+Validation R^2: 0.99954, RMSE: 0.40895
 Testing R^2: 0.98676, RMSE: 2.36514
 Mean cross-validation score: 0.98471

	Feature	Importance
0	CPI_EST_avg	0.9150
50	CPI_EST_prev	0.0336
45	SP_POP_6569_FE_5Y	0.0022
4	NY_GDP_PCAP_KD_rel	0.0021
44	EG_USE_ELEC_KH_PC	0.0018
12	SP_DYN_LE00_FE_IN	0.0017
5	NY_GDP_PCAP_KD	0.0017
46	SP_DYN_T065_FE_ZS	0.0017
17	SE_SEC_ENRR_FE	0.0015
33	SP_POP_65UP_FE_ZS	0.0014
3	NY_GNP_PCAP_CD	0.0013
39	SH_DYN_NMRT	0.0013
6	NY_GDP_PCAP_CD	0.0013
41	SP_POP_DPNP_YG	0.0013
47	SP_DYN_IMRT_MA_IN	0.0012
49	SP_POP_7579_FE_5Y	0.0012
18	SP_POP_7579_MA_5Y	0.0012
19	SP_POP_80UP_FE_5Y	0.0012
22	SE_SEC_ENRR_MA	0.0012
30	SP_POP_0014_FE_ZS	0.0011
42	SP_POP_7074_FE_5Y	0.0011
31	SP_POP_0509_FE_5Y	0.0011
35	SP_POP_1519_FE_5Y	0.0011
27	SP_POP_0004_MA_5Y	0.0011
20	SP_DYN_T065_MA_ZS	0.0011
11	SP_POP_65UP_MA_ZS	0.0011
48	SP_DYN_IMRT_IN	0.0010
32	SP_POP_1519_MA_5Y	0.0010
7	IT_NET_USER_ZS	0.0010
36	SP_DYN_CBRT_IN	0.0010
26	IT_MLT_MAIN_P2	0.0010
10	SP_POP_80UP_MA_5Y	0.0009
9	SP_DYN_LE00_IN	0.0009

21	SP_POP_1014_MA_5Y	0.0008
40	SP_POP_5054_MA_5Y	0.0008
34	SP_POP_6064_MA_5Y	0.0008
43	SP_DYN_AMRT_MA	0.0007
38	SP_POP_0004_FE_5Y	0.0007
2	NY_ADJ_NNTY_PC_CD	0.0007
37	SP_POP_5559_MA_5Y	0.0007
29	SP_POP_1014_FE_5Y	0.0007
14	SP_POP_0014_MA_ZS	0.0007
13	SE_SEC_ENRR	0.0007
8	SP_DYN_LE00_MA_IN	0.0007
25	SP_POP_6569_MA_5Y	0.0007
28	SP_POP_DPND_DL	0.0006
24	SP_POP_65UP_TO_ZS	0.0006
15	SP_POP_0509_MA_5Y	0.0006
1	NE_CON_PRVT_PC_KD	0.0005
16	SP_POP_7074_MA_5Y	0.0005
23	SP_POP_0014_TO_ZS	0.0004

Model with 52 variables and max depth None:

Training+Validation R^2: 0.99852, RMSE: 0.73161

Testing R^2: 0.98641, RMSE: 2.39615

Mean cross-validation score: 0.98474

	Feature	Importance
0	CPI_EST_avg	0.9371
51	CPI_EST_prev	0.0286
17	SE_SEC_ENRR_FE	0.0013
44	EG_USE_ELEC_KH_PC	0.0013
33	SP_POP_65UP_FE_ZS	0.0013
12	SP_DYN_LE00_FE_IN	0.0011
4	NY_GDP_PCAP_KD_rel	0.0011
46	SP_DYN_TO65_FE_ZS	0.0010
19	SP_POP_80UP_FE_5Y	0.0010
39	SH_DYN_NMRT	0.0009
41	SP_POP_DPND_YG	0.0009
45	SP_POP_6569_FE_5Y	0.0009
22	SE_SEC_ENRR_MA	0.0009
5	NY_GDP_PCAP_KD	0.0009
11	SP_POP_65UP_MA_ZS	0.0008
49	SP_POP_7579_FE_5Y	0.0008
50	SP_DYN_IMRT_FE_IN	0.0008
36	SP_DYN_CBRT_IN	0.0008
42	SP_POP_7074_FE_5Y	0.0008
38	SP_POP_0004_FE_5Y	0.0007
35	SP_POP_1519_FE_5Y	0.0007
3	NY_GNP_PCAP_CD	0.0007

28	SP_POP_DPND_DL	0.0007
30	SP_POP_0014_FE_ZS	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
7	IT_NET_USER_ZS	0.0007
6	NY_GDP_PCAP_CD	0.0007
40	SP_POP_5054_MA_5Y	0.0006
31	SP_POP_0509_FE_5Y	0.0006
32	SP_POP_1519_MA_5Y	0.0006
27	SP_POP_0004_MA_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
21	SP_POP_1014_MA_5Y	0.0006
20	SP_DYN_TO65_MA_ZS	0.0006
18	SP_POP_7579_MA_5Y	0.0006
9	SP_DYN_LE00_IN	0.0006
34	SP_POP_6064_MA_5Y	0.0005
25	SP_POP_6569_MA_5Y	0.0005
24	SP_POP_65UP_TO_ZS	0.0005
37	SP_POP_5559_MA_5Y	0.0005
43	SP_DYN_AMRT_MA	0.0005
13	SE_SEC_ENRR	0.0005
29	SP_POP_1014_FE_5Y	0.0004
14	SP_POP_0014_MA_ZS	0.0004
2	NY_ADJ_NNTY_PC_CD	0.0004
15	SP_POP_0509_MA_5Y	0.0004
47	SP_DYN_IMRT_MA_IN	0.0003
48	SP_DYN_IMRT_IN	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
16	SP_POP_7074_MA_5Y	0.0003
23	SP_POP_0014_TO_ZS	0.0002

Model with 53 variables and max depth None:

Training+Validation R^2: 0.99841, RMSE: 0.75906

Testing R^2: 0.98646, RMSE: 2.39131

Mean cross-validation score: 0.98457

	Feature	Importance
0	CPI_EST_avg	0.9390
52	CPI_EST_prev	0.0273
4	NY_GDP_PCAP_KD_rel	0.0012
3	NY_GNP_PCAP_CD	0.0012
46	SP_DYN_TO65_FE_ZS	0.0012
12	SP_DYN_LE00_FE_IN	0.0011
17	SE_SEC_ENRR_FE	0.0011
44	EG_USE_ELEC_KH_PC	0.0010
39	SH_DYN_NMRT	0.0009
49	SP_POP_7579_FE_5Y	0.0009

5	NY_GDP_PCAP_KD	0.0008
24	SP_POP_65UP_TO_ZS	0.0008
33	SP_POP_65UP_FE_ZS	0.0008
19	SP_POP_80UP_FE_5Y	0.0007
41	SP_POP_DPNP_YG	0.0007
23	SP_POP_0014_TO_ZS	0.0007
22	SE_SEC_ENRR_MA	0.0007
20	SP_DYN_TO65_MA_ZS	0.0007
32	SP_POP_1519_MA_5Y	0.0007
45	SP_POP_6569_FE_5Y	0.0007
11	SP_POP_65UP_MA_ZS	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
51	NV_SRV_TOTL_ZS	0.0007
42	SP_POP_7074_FE_5Y	0.0007
31	SP_POP_0509_FE_5Y	0.0006
40	SP_POP_5054_MA_5Y	0.0006
35	SP_POP_1519_FE_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
25	SP_POP_6569_MA_5Y	0.0006
21	SP_POP_1014_MA_5Y	0.0006
38	SP_POP_0004_FE_5Y	0.0006
47	SP_DYN_IMRT_MA_IN	0.0006
7	IT_NET_USER_ZS	0.0006
6	NY_GDP_PCAP_CD	0.0006
36	SP_DYN_CBRT_IN	0.0006
43	SP_DYN_AMRT_MA	0.0005
48	SP_DYN_IMRT_IN	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005
37	SP_POP_5559_MA_5Y	0.0005
30	SP_POP_0014_FE_ZS	0.0005
28	SP_POP_DPNP_DL	0.0005
18	SP_POP_7579_MA_5Y	0.0005
13	SE_SEC_ENRR	0.0005
9	SP_DYN_LE00_IN	0.0005
8	SP_DYN_LE00_MA_IN	0.0005
27	SP_POP_0004_MA_5Y	0.0005
34	SP_POP_6064_MA_5Y	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
14	SP_POP_0014_MA_ZS	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004
29	SP_POP_1014_FE_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0003
15	SP_POP_0509_MA_5Y	0.0003

Model with 54 variables and max depth None:
 Training+Validation R^2: 0.99822, RMSE: 0.80186
 Testing R^2: 0.98669, RMSE: 2.37143

Mean cross-validation score: 0.98443

	Feature	Importance
0	CPI_EST_avg	0.9256
53	CPI_EST_prev	0.0332
41	SP_POP_DPNP_YG	0.0017
46	SP_DYN_T065_FE_ZS	0.0017
17	SE_SEC_ENRR_FE	0.0016
12	SP_DYN_LE00_FE_IN	0.0015
4	NY_GDP_PCAP_KD_rel	0.0015
44	EG_USE_ELEC_KH_PC	0.0015
39	SH_DYN_NMRT	0.0012
27	SP_POP_0004_MA_5Y	0.0011
18	SP_POP_7579_MA_5Y	0.0011
19	SP_POP_80UP_FE_5Y	0.0011
32	SP_POP_1519_MA_5Y	0.0010
20	SP_DYN_T065_MA_ZS	0.0010
22	SE_SEC_ENRR_MA	0.0010
11	SP_POP_65UP_MA_ZS	0.0009
30	SP_POP_0014_FE_ZS	0.0009
7	IT_NET_USER_ZS	0.0009
49	SP_POP_7579_FE_5Y	0.0008
48	SP_DYN_IMRT_IN	0.0008
50	SP_DYN_IMRT_FE_IN	0.0008
5	NY_GDP_PCAP_KD	0.0008
51	NV_SRV_TOTL_ZS	0.0008
3	NY_GNP_PCAP_CD	0.0008
52	SP_ADO_TFRT	0.0007
34	SP_POP_6064_MA_5Y	0.0007
38	SP_POP_0004_FE_5Y	0.0007
45	SP_POP_6569_FE_5Y	0.0007
42	SP_POP_7074_FE_5Y	0.0007
13	SE_SEC_ENRR	0.0007
21	SP_POP_1014_MA_5Y	0.0007
36	SP_DYN_CBRT_IN	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
47	SP_DYN_IMRT_MA_IN	0.0007
35	SP_POP_1519_FE_5Y	0.0007
40	SP_POP_5054_MA_5Y	0.0006
2	NY_ADJ_NNTY_PC_CD	0.0006
6	NY_GDP_PCAP_CD	0.0006
26	IT_MLT_MAIN_P2	0.0006
37	SP_POP_5559_MA_5Y	0.0005
33	SP_POP_65UP_FE_ZS	0.0005
8	SP_DYN_LE00_MA_IN	0.0005
9	SP_DYN_LE00_IN	0.0005
23	SP_POP_0014_TO_ZS	0.0005
28	SP_POP_DPNP_DL	0.0005

43	SP_DYN_AMRT_MA	0.0005
15	SP_POP_0509_MA_5Y	0.0004
14	SP_POP_0014_MA_ZS	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
29	SP_POP_1014_FE_5Y	0.0004
31	SP_POP_0509_FE_5Y	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
25	SP_POP_6569_MA_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0002

Model with 55 variables and max depth None:

Training+Validation R^2: 0.99958, RMSE: 0.38974

Testing R^2: 0.98655, RMSE: 2.3835

Mean cross-validation score: 0.98451

	Feature	Importance
0	CPI_EST_avg	0.9191
54	CPI_EST_prev	0.0343
46	SP_DYN_TO65_FE_ZS	0.0023
17	SE_SEC_ENRR_FE	0.0019
4	NY_GDP_PCAP_KD_rel	0.0017
5	NY_GDP_PCAP_KD	0.0015
18	SP_POP_7579_MA_5Y	0.0014
44	EG_USE_ELEC_KH_PC	0.0013
39	SH_DYN_NMRT	0.0013
19	SP_POP_80UP_FE_5Y	0.0013
49	SP_POP_7579_FE_5Y	0.0012
27	SP_POP_0004_MA_5Y	0.0012
33	SP_POP_65UP_FE_ZS	0.0012
20	SP_DYN_TO65_MA_ZS	0.0011
12	SP_DYN_LE00_FE_IN	0.0011
32	SP_POP_1519_MA_5Y	0.0010
53	SP_POP_5054_FE_5Y	0.0010
38	SP_POP_0004_FE_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0009
51	NV_SRV_TOTL_ZS	0.0009
52	SP_ADO_TFRT	0.0009
50	SP_DYN_IMRT_FE_IN	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
22	SE_SEC_ENRR_MA	0.0009
41	SP_POP_DPND_YG	0.0009
6	NY_GDP_PCAP_CD	0.0009
42	SP_POP_7074_FE_5Y	0.0008
40	SP_POP_5054_MA_5Y	0.0008
36	SP_DYN_CBRT_IN	0.0008
35	SP_POP_1519_FE_5Y	0.0008
43	SP_DYN_AMRT_MA	0.0008

28	SP_POP_DPND_OL	0.0008
21	SP_POP_1014_MA_5Y	0.0008
25	SP_POP_6569_MA_5Y	0.0008
7	IT_NET_USER_ZS	0.0008
23	SP_POP_0014_TO_ZS	0.0008
10	SP_POP_80UP_MA_5Y	0.0007
13	SE_SEC_ENRR	0.0007
48	SP_DYN_IMRT_IN	0.0007
31	SP_POP_0509_FE_5Y	0.0007
26	IT_MLT_MAIN_P2	0.0007
2	NY_ADJ_NNTY_PC_CD	0.0007
15	SP_POP_0509_MA_5Y	0.0006
37	SP_POP_5559_MA_5Y	0.0006
3	NY_GNP_PCAP_CD	0.0006
24	SP_POP_65UP_TO_ZS	0.0005
30	SP_POP_0014_FE_ZS	0.0005
9	SP_DYN_LE00_IN	0.0005
34	SP_POP_6064_MA_5Y	0.0005
1	NE_CON_PRVT_PC_KD	0.0004
16	SP_POP_7074_MA_5Y	0.0004
14	SP_POP_0014_MA_ZS	0.0004
29	SP_POP_1014_FE_5Y	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
47	SP_DYN_IMRT_MA_IN	0.0002

Model with 56 variables and max depth None:

Training+Validation R^2: 0.9994, RMSE: 0.46447

Testing R^2: 0.98676, RMSE: 2.36462

Mean cross-validation score: 0.98483

	Feature	Importance
0	CPI_EST_avg	0.9329
55	CPI_EST_prev	0.0301
41	SP_POP_DPND_YG	0.0015
4	NY_GDP_PCAP_KD_rel	0.0014
44	EG_USE_ELEC_KH_PC	0.0013
19	SP_POP_80UP_FE_5Y	0.0012
22	SE_SEC_ENRR_MA	0.0011
52	SP_ADO_TFRT	0.0011
17	SE_SEC_ENRR_FE	0.0011
54	SP_POP_5559_FE_5Y	0.0010
46	SP_DYN_TO65_FE_ZS	0.0010
12	SP_DYN_LE00_FE_IN	0.0010
39	SH_DYN_NMRT	0.0009
35	SP_POP_1519_FE_5Y	0.0009
49	SP_POP_7579_FE_5Y	0.0009
18	SP_POP_7579_MA_5Y	0.0008

33	SP_POP_65UP_FE_ZS	0.0008
20	SP_DYN_TO65_MA_ZS	0.0008
42	SP_POP_7074_FE_5Y	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
51	NV_SRV_TOTL_ZS	0.0007
21	SP_POP_1014_MA_5Y	0.0007
43	SP_DYN_AMRT_MA	0.0007
45	SP_POP_6569_FE_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
32	SP_POP_1519_MA_5Y	0.0007
50	SP_DYN_IMRT_FE_IN	0.0006
36	SP_DYN_CBRT_IN	0.0006
9	SP_DYN_LE00_IN	0.0006
34	SP_POP_6064_MA_5Y	0.0006
3	NY_GNP_PCAP_CD	0.0006
30	SP_POP_0014_FE_ZS	0.0006
47	SP_DYN_IMRT_MA_IN	0.0006
5	NY_GDP_PCAP_KD	0.0006
53	SP_POP_5054_FE_5Y	0.0006
23	SP_POP_0014_TO_ZS	0.0006
7	IT_NET_USER_ZS	0.0006
40	SP_POP_5054_MA_5Y	0.0005
31	SP_POP_0509_FE_5Y	0.0005
27	SP_POP_0004_MA_5Y	0.0005
25	SP_POP_6569_MA_5Y	0.0005
29	SP_POP_1014_FE_5Y	0.0004
28	SP_POP_DPND_DL	0.0004
38	SP_POP_0004_FE_5Y	0.0004
37	SP_POP_5559_MA_5Y	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
26	IT_MLT_MAIN_P2	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
15	SP_POP_0509_MA_5Y	0.0004
14	SP_POP_0014_MA_ZS	0.0004
13	SE_SEC_ENRR	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
48	SP_DYN_IMRT_IN	0.0003
16	SP_POP_7074_MA_5Y	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0003

Model with 57 variables and max depth None:
 Training+Validation R^2: 0.99765, RMSE: 0.9213
 Testing R^2: 0.98596, RMSE: 2.4354
 Mean cross-validation score: 0.98479

Feature Importance

0	CPI_EST_avg	0.9157
56	CPI_EST_prev	0.0379
33	SP_POP_65UP_FE_ZS	0.0031
17	SE_SEC_ENRR_FE	0.0019
4	NY_GDP_PCAP_KD_rel	0.0019
46	SP_DYN_T065_FE_ZS	0.0015
39	SH_DYN_NMRT	0.0015
44	EG_USE_ELEC_KH_PC	0.0013
54	SP_POP_5559_FE_5Y	0.0013
12	SP_DYN_LE00_FE_IN	0.0013
32	SP_POP_1519_MA_5Y	0.0012
42	SP_POP_7074_FE_5Y	0.0011
49	SP_POP_7579_FE_5Y	0.0011
11	SP_POP_65UP_MA_ZS	0.0011
51	NV_SRV_TOTL_ZS	0.0011
19	SP_POP_80UP_FE_5Y	0.0010
20	SP_DYN_T065_MA_ZS	0.0010
45	SP_POP_6569_FE_5Y	0.0009
24	SP_POP_65UP_TO_ZS	0.0009
6	NY_GDP_PCAP_CD	0.0009
30	SP_POP_0014_FE_ZS	0.0009
18	SP_POP_7579_MA_5Y	0.0009
52	SP_ADO_TFRT	0.0009
22	SE_SEC_ENRR_MA	0.0008
27	SP_POP_0004_MA_5Y	0.0008
3	NY_GNP_PCAP_CD	0.0008
13	SE_SEC_ENRR	0.0008
50	SP_DYN_IMRT_FE_IN	0.0008
36	SP_DYN_CBRT_IN	0.0007
41	SP_POP_DPND_YG	0.0007
55	SH_DYN_1519	0.0007
35	SP_POP_1519_FE_5Y	0.0007
7	IT_NET_USER_ZS	0.0007
43	SP_DYN_AMRT_MA	0.0007
5	NY_GDP_PCAP_KD	0.0007
29	SP_POP_1014_FE_5Y	0.0006
47	SP_DYN_IMRT_MA_IN	0.0006
53	SP_POP_5054_FE_5Y	0.0006
40	SP_POP_5054_MA_5Y	0.0006
28	SP_POP_DPND_OL	0.0006
38	SP_POP_0004_FE_5Y	0.0006
21	SP_POP_1014_MA_5Y	0.0006
9	SP_DYN_LE00_IN	0.0006
31	SP_POP_0509_FE_5Y	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
25	SP_POP_6569_MA_5Y	0.0006
14	SP_POP_0014_MA_ZS	0.0005

48	SP_DYN_IMRT_IN	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005
37	SP_POP_5559_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
34	SP_POP_6064_MA_5Y	0.0004
15	SP_POP_0509_MA_5Y	0.0003
23	SP_POP_0014_TO_ZS	0.0003
8	SP_DYN_LE00_MA_IN	0.0003

Model with 58 variables and max depth None:

Training+Validation R^2: 0.99893, RMSE: 0.62108

Testing R^2: 0.98569, RMSE: 2.45822

Mean cross-validation score: 0.98473

	Feature	Importance
0	CPI_EST_avg	0.9327
57	CPI_EST_prev	0.0300
46	SP_DYN_T065_FE_ZS	0.0013
17	SE_SEC_ENRR_FE	0.0013
44	EG_USE_ELEC_KH_PC	0.0012
54	SP_POP_5559_FE_5Y	0.0012
4	NY_GDP_PCAP_KD_rel	0.0011
5	NY_GDP_PCAP_KD	0.0011
33	SP_POP_65UP_FE_ZS	0.0011
12	SP_DYN_LE00_FE_IN	0.0010
20	SP_DYN_T065_MA_ZS	0.0010
32	SP_POP_1519_MA_5Y	0.0009
49	SP_POP_7579_FE_5Y	0.0009
39	SH_DYN_NMRT	0.0009
27	SP_POP_0004_MA_5Y	0.0009
11	SP_POP_65UP_MA_ZS	0.0008
42	SP_POP_7074_FE_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0008
19	SP_POP_80UP_FE_5Y	0.0008
51	NV_SRV_TOTL_ZS	0.0008
52	SP_ADO_TFRT	0.0007
43	SP_DYN_AMRT_MA	0.0007
45	SP_POP_6569_FE_5Y	0.0007
3	NY_GNP_PCAP_CD	0.0007
53	SP_POP_5054_FE_5Y	0.0007
22	SE_SEC_ENRR_MA	0.0007
55	SH_DYN_1519	0.0007
14	SP_POP_0014_MA_ZS	0.0007
9	SP_DYN_LE00_IN	0.0007
36	SP_DYN_CBRT_IN	0.0006
23	SP_POP_0014_TO_ZS	0.0006

28	SP_POP_DPND_OL	0.0006
6	NY_GDP_PCAP_CD	0.0006
50	SP_DYN_IMRT_FE_IN	0.0006
7	IT_NET_USER_ZS	0.0006
56	SP_POP_6064_FE_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
41	SP_POP_DPND_YG	0.0006
40	SP_POP_5054_MA_5Y	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
38	SP_POP_0004_FE_5Y	0.0005
37	SP_POP_5559_MA_5Y	0.0005
35	SP_POP_1519_FE_5Y	0.0005
13	SE_SEC_ENRR	0.0005
21	SP_POP_1014_MA_5Y	0.0005
47	SP_DYN_IMRT_MA_IN	0.0004
30	SP_POP_0014_FE_ZS	0.0004
34	SP_POP_6064_MA_5Y	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
25	SP_POP_6569_MA_5Y	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
29	SP_POP_1014_FE_5Y	0.0004
8	SP_DYN_LE00_MA_IN	0.0003
48	SP_DYN_IMRT_IN	0.0002
31	SP_POP_0509_FE_5Y	0.0002
16	SP_POP_7074_MA_5Y	0.0002
15	SP_POP_0509_MA_5Y	0.0002

Model with 59 variables and max depth None:

Training+Validation R^2: 0.99802, RMSE: 0.84569

Testing R^2: 0.98581, RMSE: 2.44853

Mean cross-validation score: 0.98483

	Feature	Importance
0	CPI_EST_avg	0.9147
58	CPI_EST_prev	0.0303
3	NY_GNP_PCAP_CD	0.0124
4	NY_GDP_PCAP_KD_rel	0.0017
17	SE_SEC_ENRR_FE	0.0016
12	SP_DYN_LE00_FE_IN	0.0013
20	SP_DYN_TO65_MA_ZS	0.0013
38	SP_POP_0004_FE_5Y	0.0012
39	SH_DYN_NMRT	0.0012
19	SP_POP_80UP_FE_5Y	0.0012
46	SP_DYN_TO65_FE_ZS	0.0012
33	SP_POP_65UP_FE_ZS	0.0012
11	SP_POP_65UP_MA_ZS	0.0012

44	EG_USE_ELEC_KH_PC	0.0011
54	SP_POP_5559_FE_5Y	0.0011
22	SE_SEC_ENRR_MA	0.0011
49	SP_POP_7579_FE_5Y	0.0010
18	SP_POP_7579_MA_5Y	0.0009
21	SP_POP_1014_MA_5Y	0.0009
57	SP_POP_4549_MA_5Y	0.0009
27	SP_POP_0004_MA_5Y	0.0009
28	SP_POP_DPND_DL	0.0009
5	NY_GDP_PCAP_KD	0.0009
45	SP_POP_6569_FE_5Y	0.0008
36	SP_DYN_CBRT_IN	0.0008
29	SP_POP_1014_FE_5Y	0.0008
32	SP_POP_1519_MA_5Y	0.0007
52	SP_ADO_TFRT	0.0007
53	SP_POP_5054_FE_5Y	0.0007
43	SP_DYN_AMRT_MA	0.0007
51	NV_SRV_TOTL_ZS	0.0007
35	SP_POP_1519_FE_5Y	0.0007
30	SP_POP_0014_FE_ZS	0.0007
6	NY_GDP_PCAP_CD	0.0007
7	IT_NET_USER_ZS	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
31	SP_POP_0509_FE_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
56	SP_POP_6064_FE_5Y	0.0006
55	SH_DYN_1519	0.0006
41	SP_POP_DPND_YG	0.0006
42	SP_POP_7074_FE_5Y	0.0006
25	SP_POP_6569_MA_5Y	0.0005
40	SP_POP_5054_MA_5Y	0.0005
13	SE_SEC_ENRR	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
9	SP_DYN_LE00_IN	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004
14	SP_POP_0014_MA_ZS	0.0004
47	SP_DYN_IMRT_MA_IN	0.0004
34	SP_POP_6064_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
15	SP_POP_0509_MA_5Y	0.0003
23	SP_POP_0014_TO_ZS	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
37	SP_POP_5559_MA_5Y	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0003
48	SP_DYN_IMRT_IN	0.0002

Model with 60 variables and max depth None:
 Training+Validation R^2: 0.99866, RMSE: 0.69621
 Testing R^2: 0.98564, RMSE: 2.46287
 Mean cross-validation score: 0.98465

	Feature	Importance
0	CPI_EST_avg	0.9121
59	CPI_EST_prev	0.0318
3	NY_GNP_PCAP_CD	0.0108
4	NY_GDP_PCAP_KD_rel	0.0020
12	SP_DYN_LE00_FE_IN	0.0019
45	SP_POP_6569_FE_5Y	0.0016
44	EG_USE_ELEC_KH_PC	0.0016
20	SP_DYN_T065_MA_ZS	0.0015
46	SP_DYN_T065_FE_ZS	0.0015
17	SE_SEC_ENRR_FE	0.0014
23	SP_POP_0014_TO_ZS	0.0014
19	SP_POP_80UP_FE_5Y	0.0013
39	SH_DYN_NMRT	0.0012
11	SP_POP_65UP_MA_ZS	0.0012
33	SP_POP_65UP_FE_ZS	0.0011
54	SP_POP_5559_FE_5Y	0.0010
42	SP_POP_7074_FE_5Y	0.0009
22	SE_SEC_ENRR_MA	0.0009
18	SP_POP_7579_MA_5Y	0.0009
27	SP_POP_0004_MA_5Y	0.0008
36	SP_DYN_CBRT_IN	0.0008
49	SP_POP_7579_FE_5Y	0.0008
51	NV_SRV_TOTL_ZS	0.0008
7	IT_NET_USER_ZS	0.0008
57	SP_POP_4549_MA_5Y	0.0008
58	SE_PRE_ENRR	0.0008
38	SP_POP_0004_FE_5Y	0.0007
16	SP_POP_7074_MA_5Y	0.0007
52	SP_ADO_TFRT	0.0007
53	SP_POP_5054_FE_5Y	0.0007
28	SP_POP_DPND_OL	0.0007
29	SP_POP_1014_FE_5Y	0.0007
32	SP_POP_1519_MA_5Y	0.0007
35	SP_POP_1519_FE_5Y	0.0007
30	SP_POP_0014_FE_ZS	0.0007
43	SP_DYN_AMRT_MA	0.0006
40	SP_POP_5054_MA_5Y	0.0006
31	SP_POP_0509_FE_5Y	0.0006
1	NE_CON_PRVT_PC_KD	0.0006
25	SP_POP_6569_MA_5Y	0.0006
5	NY_GDP_PCAP_KD	0.0006
21	SP_POP_1014_MA_5Y	0.0006

6	NY_GDP_PCAP_CD	0.0006
55	SH_DYN_1519	0.0005
8	SP_DYN_LEOO_MA_IN	0.0005
26	IT_MLT_MAIN_P2	0.0005
56	SP_POP_6064_FE_5Y	0.0005
34	SP_POP_6064_MA_5Y	0.0005
50	SP_DYN_IMRT_FE_IN	0.0005
13	SE_SEC_ENRR	0.0005
10	SP_POP_80UP_MA_5Y	0.0004
9	SP_DYN_LEOO_IN	0.0004
37	SP_POP_5559_MA_5Y	0.0004
48	SP_DYN_IMRT_IN	0.0004
41	SP_POP_DPND_YG	0.0004
47	SP_DYN_IMRT_MA_IN	0.0004
2	NY_ADJ_NNTY_PC_CD	0.0003
15	SP_POP_0509_MA_5Y	0.0003
24	SP_POP_65UP_TO_ZS	0.0002
14	SP_POP_0014_MA_ZS	0.0002

Model with 61 variables and max depth None:

Training+Validation R^2: 0.99954, RMSE: 0.40749

Testing R^2: 0.98653, RMSE: 2.38547

Mean cross-validation score: 0.98493

	Feature	Importance
0	CPI_EST_avg	0.9065
60	CPI_EST_prev	0.0339
3	NY_GNP_PCAP_CD	0.0106
4	NY_GDP_PCAP_KD_rel	0.0024
17	SE_SEC_ENRR_FE	0.0019
44	EG_USE_ELEC_KH_PC	0.0019
12	SP_DYN_LEOO_FE_IN	0.0016
39	SH_DYN_NMRT	0.0015
19	SP_POP_80UP_FE_5Y	0.0015
45	SP_POP_6569_FE_5Y	0.0014
46	SP_DYN_T065_FE_ZS	0.0014
42	SP_POP_7074_FE_5Y	0.0014
18	SP_POP_7579_MA_5Y	0.0014
57	SP_POP_4549_MA_5Y	0.0011
54	SP_POP_5559_FE_5Y	0.0011
11	SP_POP_65UP_MA_ZS	0.0011
22	SE_SEC_ENRR_MA	0.0010
20	SP_DYN_T065_MA_ZS	0.0010
49	SP_POP_7579_FE_5Y	0.0010
51	NV_SRV_TOTL_ZS	0.0010
52	SP_ADO_TFRT	0.0010
6	NY_GDP_PCAP_CD	0.0010

5	NY_GDP_PCAP_KD	0.0010
58	SE_PRE_ENRR	0.0009
53	SP_POP_5054_FE_5Y	0.0009
23	SP_POP_0014_TO_ZS	0.0009
1	NE_CON_PRVT_PC_KD	0.0008
56	SP_POP_6064_FE_5Y	0.0008
40	SP_POP_5054_MA_5Y	0.0008
59	NV_AGR_TOTL_ZS	0.0008
27	SP_POP_0004_MA_5Y	0.0007
26	IT_MLT_MAIN_P2	0.0007
32	SP_POP_1519_MA_5Y	0.0007
33	SP_POP_65UP_FE_ZS	0.0007
35	SP_POP_1519_FE_5Y	0.0007
7	IT_NET_USER_ZS	0.0007
55	SH_DYN_1519	0.0007
47	SP_DYN_IMRT_MA_IN	0.0006
43	SP_DYN_AMRT_MA	0.0006
29	SP_POP_1014_FE_5Y	0.0006
28	SP_POP_DPND_OL	0.0006
24	SP_POP_65UP_TO_ZS	0.0006
16	SP_POP_7074_MA_5Y	0.0006
14	SP_POP_0014_MA_ZS	0.0006
13	SE_SEC_ENRR	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
38	SP_POP_0004_FE_5Y	0.0005
36	SP_DYN_CBRT_IN	0.0005
25	SP_POP_6569_MA_5Y	0.0005
48	SP_DYN_IMRT_IN	0.0005
31	SP_POP_0509_FE_5Y	0.0005
41	SP_POP_DPND_YG	0.0004
37	SP_POP_5559_MA_5Y	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004
10	SP_POP_80UP_MA_5Y	0.0004
9	SP_DYN_LE00_IN	0.0004
2	NY_ADJ_NNTY_PC_CD	0.0003
34	SP_POP_6064_MA_5Y	0.0003
21	SP_POP_1014_MA_5Y	0.0003
15	SP_POP_0509_MA_5Y	0.0003
30	SP_POP_0014_FE_ZS	0.0002

Model with 62 variables and max depth None:
 Training+Validation R^2: 0.99962, RMSE: 0.36927
 Testing R^2: 0.98618, RMSE: 2.41584
 Mean cross-validation score: 0.98488

	Feature	Importance
0	CPI_EST_avg	0.9094

61	CPI_EST_prev	0.0322
3	NY_GNP_PCAP_CD	0.0108
4	NY_GDP_PCAP_KD_rel	0.0027
17	SE_SEC_ENRR_FE	0.0016
44	EG_USE_ELEC_KH_PC	0.0016
54	SP_POP_5559_FE_5Y	0.0016
12	SP_DYN_LE00_FE_IN	0.0016
39	SH_DYN_NMRT	0.0014
30	SP_POP_0014_FE_ZS	0.0014
46	SP_DYN_T065_FE_ZS	0.0013
45	SP_POP_6569_FE_5Y	0.0013
49	SP_POP_7579_FE_5Y	0.0011
20	SP_DYN_T065_MA_ZS	0.0011
19	SP_POP_80UP_FE_5Y	0.0011
42	SP_POP_7074_FE_5Y	0.0010
6	NY_GDP_PCAP_CD	0.0010
27	SP_POP_0004_MA_5Y	0.0009
18	SP_POP_7579_MA_5Y	0.0009
40	SP_POP_5054_MA_5Y	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
29	SP_POP_1014_FE_5Y	0.0009
57	SP_POP_4549_MA_5Y	0.0008
52	SP_ADO_TFRT	0.0008
7	IT_NET_USER_ZS	0.0008
59	NV_AGR_TOTL_ZS	0.0008
38	SP_POP_0004_FE_5Y	0.0008
22	SE_SEC_ENRR_MA	0.0008
55	SH_DYN_1519	0.0008
58	SE_PRE_ENRR	0.0008
51	NV_SRV_TOTL_ZS	0.0008
56	SP_POP_6064_FE_5Y	0.0007
53	SP_POP_5054_FE_5Y	0.0007
47	SP_DYN_IMRT_MA_IN	0.0007
8	SP_DYN_LE00_MA_IN	0.0007
35	SP_POP_1519_FE_5Y	0.0007
26	IT_MLT_MAIN_P2	0.0007
32	SP_POP_1519_MA_5Y	0.0006
36	SP_DYN_CBRT_IN	0.0006
34	SP_POP_6064_MA_5Y	0.0006
1	NE_CON_PRVT_PC_KD	0.0006
28	SP_POP_DPND_DL	0.0006
14	SP_POP_0014_MA_ZS	0.0006
60	SH_DYN_MORT_MA	0.0006
5	NY_GDP_PCAP_KD	0.0006
24	SP_POP_65UP_TO_ZS	0.0006
48	SP_DYN_IMRT_IN	0.0005
31	SP_POP_0509_FE_5Y	0.0005
43	SP_DYN_AMRT_MA	0.0005

23	SP_POP_0014_TO_ZS	0.0005
21	SP_POP_1014_MA_5Y	0.0005
16	SP_POP_7074_MA_5Y	0.0005
13	SE_SEC_ENRR	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
37	SP_POP_5559_MA_5Y	0.0004
15	SP_POP_0509_MA_5Y	0.0004
9	SP_DYN_LEOO_IN	0.0004
2	NY_ADJ_NNTY_PC_CD	0.0004
33	SP_POP_65UP_FE_ZS	0.0003
25	SP_POP_6569_MA_5Y	0.0003
41	SP_POP_DPND_YG	0.0002
50	SP_DYN_IMRT_FE_IN	0.0001

Model with 63 variables and max depth None:

Training+Validation R^2: 0.99927, RMSE: 0.51306

Testing R^2: 0.98612, RMSE: 2.42177

Mean cross-validation score: 0.98485

	Feature	Importance
0	CPI_EST_avg	0.9187
62	CPI_EST_prev	0.0311
3	NY_GNP_PCAP_CD	0.0100
46	SP_DYN_TO65_FE_ZS	0.0016
30	SP_POP_0014_FE_ZS	0.0016
17	SE_SEC_ENRR_FE	0.0015
4	NY_GDP_PCAP_KD_rel	0.0014
54	SP_POP_5559_FE_5Y	0.0012
12	SP_DYN_LEOO_FE_IN	0.0012
39	SH_DYN_NMRT	0.0011
45	SP_POP_6569_FE_5Y	0.0011
19	SP_POP_80UP_FE_5Y	0.0010
55	SH_DYN_1519	0.0010
44	EG_USE_ELEC_KH_PC	0.0010
41	SP_POP_DPND_YG	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
22	SE_SEC_ENRR_MA	0.0009
29	SP_POP_1014_FE_5Y	0.0008
57	SP_POP_4549_MA_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0008
51	NV_SRV_TOTL_ZS	0.0008
49	SP_POP_7579_FE_5Y	0.0008
42	SP_POP_7074_FE_5Y	0.0008
27	SP_POP_0004_MA_5Y	0.0008
20	SP_DYN_TO65_MA_ZS	0.0007
59	NV_AGR_TOTL_ZS	0.0007
7	IT_NET_USER_ZS	0.0007

40	SP_POP_5054_MA_5Y	0.0007
53	SP_POP_5054_FE_5Y	0.0006
32	SP_POP_1519_MA_5Y	0.0006
56	SP_POP_6064_FE_5Y	0.0006
58	SE_PRE_ENRR	0.0006
60	SH_DYN_MORT_MA	0.0006
61	SP_DYN_TFRT_IN	0.0006
6	NY_GDP_PCAP_CD	0.0006
52	SP_ADO_TFRT	0.0006
43	SP_DYN_AMRT_MA	0.0005
25	SP_POP_6569_MA_5Y	0.0005
35	SP_POP_1519_FE_5Y	0.0005
34	SP_POP_6064_MA_5Y	0.0005
1	NE_CON_PRVT_PC_KD	0.0005
28	SP_POP_DPND_DL	0.0005
26	IT_MLT_MAIN_P2	0.0005
48	SP_DYN_IMRT_IN	0.0005
24	SP_POP_65UP_TO_ZS	0.0005
21	SP_POP_1014_MA_5Y	0.0005
5	NY_GDP_PCAP_KD	0.0005
47	SP_DYN_IMRT_MA_IN	0.0005
2	NY_ADJ_NNTY_PC_CD	0.0004
36	SP_DYN_CBRT_IN	0.0004
23	SP_POP_0014_TO_ZS	0.0004
13	SE_SEC_ENRR	0.0004
9	SP_DYN_LEOO_IN	0.0004
8	SP_DYN_LEOO_MA_IN	0.0004
38	SP_POP_0004_FE_5Y	0.0003
50	SP_DYN_IMRT_FE_IN	0.0003
37	SP_POP_5559_MA_5Y	0.0003
14	SP_POP_0014_MA_ZS	0.0003
10	SP_POP_80UP_MA_5Y	0.0003
31	SP_POP_0509_FE_5Y	0.0003
33	SP_POP_65UP_FE_ZS	0.0002
16	SP_POP_7074_MA_5Y	0.0002
15	SP_POP_0509_MA_5Y	0.0002

Model with 64 variables and max depth None:

Training+Validation R^2: 0.99959, RMSE: 0.3841

Testing R^2: 0.98691, RMSE: 2.35171

Mean cross-validation score: 0.98518

	Feature	Importance
0	CPI_EST_avg	0.9172
63	CPI_EST_prev	0.0327
3	NY_GNP_PCAP_CD	0.0083
4	NY_GDP_PCAP_KD_rel	0.0023

12	SP_DYN_LE00_FE_IN	0.0016
17	SE_SEC_ENRR_FE	0.0016
54	SP_POP_5559_FE_5Y	0.0013
39	SH_DYN_NMRT	0.0012
45	SP_POP_6569_FE_5Y	0.0012
55	SH_DYN_1519	0.0012
44	EG_USE_ELEC_KH_PC	0.0011
49	SP_POP_7579_FE_5Y	0.0011
11	SP_POP_65UP_MA_ZS	0.0010
46	SP_DYN_T065_FE_ZS	0.0009
19	SP_POP_80UP_FE_5Y	0.0009
22	SE_SEC_ENRR_MA	0.0009
29	SP_POP_1014_FE_5Y	0.0009
57	SP_POP_4549_MA_5Y	0.0009
40	SP_POP_5054_MA_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0008
25	SP_POP_6569_MA_5Y	0.0007
51	NV_SRV_TOTL_ZS	0.0007
42	SP_POP_7074_FE_5Y	0.0007
38	SP_POP_0004_FE_5Y	0.0007
52	SP_ADO_TFRT	0.0007
21	SP_POP_1014_MA_5Y	0.0007
20	SP_DYN_T065_MA_ZS	0.0007
7	IT_NET_USER_ZS	0.0007
62	SH_DYN_2024	0.0007
26	IT_MLT_MAIN_P2	0.0006
41	SP_POP_DPND_YG	0.0006
56	SP_POP_6064_FE_5Y	0.0006
31	SP_POP_0509_FE_5Y	0.0006
58	SE_PRE_ENRR	0.0006
59	NV_AGR_TOTL_ZS	0.0006
60	SH_DYN_MORT_MA	0.0006
35	SP_POP_1519_FE_5Y	0.0006
61	SP_DYN_TFRT_IN	0.0006
6	NY_GDP_PCAP_CD	0.0006
32	SP_POP_1519_MA_5Y	0.0006
53	SP_POP_5054_FE_5Y	0.0005
33	SP_POP_65UP_FE_ZS	0.0005
27	SP_POP_0004_MA_5Y	0.0005
36	SP_DYN_CBRT_IN	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
14	SP_POP_0014_MA_ZS	0.0005
37	SP_POP_5559_MA_5Y	0.0004
13	SE_SEC_ENRR	0.0004
16	SP_POP_7074_MA_5Y	0.0004
43	SP_DYN_AMRT_MA	0.0004
2	NY_ADJ_NNTY_PC_CD	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004

1	NE_CON_PRVT_PC_KD	0.0004
34	SP_POP_6064_MA_5Y	0.0004
28	SP_POP_DPNP_OL	0.0003
30	SP_POP_0014_FE_ZS	0.0003
48	SP_DYN_IMRT_IN	0.0003
15	SP_POP_0509_MA_5Y	0.0003
47	SP_DYN_IMRT_MA_IN	0.0003
9	SP_DYN_LEOO_IN	0.0003
8	SP_DYN_LEOO_MA_IN	0.0003
5	NY_GDP_PCAP_KD	0.0003
24	SP_POP_65UP_TO_ZS	0.0002
23	SP_POP_0014_TO_ZS	0.0001

Model with 65 variables and max depth None:

Training+Validation R^2: 0.99887, RMSE: 0.63896

Testing R^2: 0.98664, RMSE: 2.37535

Mean cross-validation score: 0.98511

	Feature	Importance
0	CPI_EST_avg	0.9104
64	CPI_EST_prev	0.0312
5	NY_GDP_PCAP_KD	0.0187
4	NY_GDP_PCAP_KD_rel	0.0013
17	SE_SEC_ENRR_FE	0.0013
19	SP_POP_80UP_FE_5Y	0.0012
44	EG_USE_ELEC_KH_PC	0.0012
33	SP_POP_65UP_FE_ZS	0.0011
11	SP_POP_65UP_MA_ZS	0.0011
46	SP_DYN_T065_FE_ZS	0.0010
55	SH_DYN_1519	0.0010
39	SH_DYN_NMRT	0.0010
63	FS_AST_PRVT_GD_ZS	0.0009
54	SP_POP_5559_FE_5Y	0.0009
30	SP_POP_0014_FE_ZS	0.0009
41	SP_POP_DPNP_YG	0.0009
6	NY_GDP_PCAP_CD	0.0009
50	SP_DYN_IMRT_FE_IN	0.0008
20	SP_DYN_T065_MA_ZS	0.0008
57	SP_POP_4549_MA_5Y	0.0008
42	SP_POP_7074_FE_5Y	0.0008
12	SP_DYN_LEOO_FE_IN	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0007
45	SP_POP_6569_FE_5Y	0.0007
35	SP_POP_1519_FE_5Y	0.0007
51	NV_SRV_TOTL_ZS	0.0007
58	SE_PRE_ENRR	0.0007

56	SP_POP_6064_FE_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
49	SP_POP_7579_FE_5Y	0.0006
15	SP_POP_0509_MA_5Y	0.0006
27	SP_POP_0004_MA_5Y	0.0006
25	SP_POP_6569_MA_5Y	0.0006
23	SP_POP_0014_TO_ZS	0.0006
22	SE_SEC_ENRR_MA	0.0006
9	SP_DYN_LE00_IN	0.0006
62	SH_DYN_2024	0.0006
7	IT_NET_USER_ZS	0.0005
52	SP_ADO_TFRT	0.0005
13	SE_SEC_ENRR	0.0005
43	SP_DYN_AMRT_MA	0.0005
29	SP_POP_1014_FE_5Y	0.0005
61	SP_DYN_TFRT_IN	0.0005
28	SP_POP_DPNND_OL	0.0005
60	SH_DYN_MORT_MA	0.0005
53	SP_POP_5054_FE_5Y	0.0005
36	SP_DYN_CBRT_IN	0.0005
59	NV_AGR_TOTL_ZS	0.0005
40	SP_POP_5054_MA_5Y	0.0005
32	SP_POP_1519_MA_5Y	0.0004
38	SP_POP_0004_FE_5Y	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
31	SP_POP_0509_FE_5Y	0.0004
21	SP_POP_1014_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
37	SP_POP_5559_MA_5Y	0.0003
3	NY_GNP_PCAP_CD	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
14	SP_POP_0014_MA_ZS	0.0003
47	SP_DYN_IMRT_MA_IN	0.0002
34	SP_POP_6064_MA_5Y	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
48	SP_DYN_IMRT_IN	0.0001

Model with 66 variables and max depth None:

Training+Validation R^2: 0.99931, RMSE: 0.49891

Testing R^2: 0.98597, RMSE: 2.43444

Mean cross-validation score: 0.98505

	Feature	Importance
0	CPI_EST_avg	0.8888
65	CPI_EST_prev	0.0353
5	NY_GDP_PCAP_KD	0.0214

4	NY_GDP_PCAP_KD_rel	0.0020
6	NY_GDP_PCAP_CD	0.0018
19	SP_POP_80UP_FE_5Y	0.0016
17	SE_SEC_ENRR_FE	0.0016
44	EG_USE_ELEC_KH_PC	0.0015
55	SH_DYN_1519	0.0014
50	SP_DYN_IMRT_FE_IN	0.0014
39	SH_DYN_NMRT	0.0014
12	SP_DYN_LE00_FE_IN	0.0013
54	SP_POP_5559_FE_5Y	0.0013
46	SP_DYN_T065_FE_ZS	0.0013
33	SP_POP_65UP_FE_ZS	0.0012
63	FS_AST_PRVT_GD_ZS	0.0012
11	SP_POP_65UP_MA_ZS	0.0012
64	SH_DYN_MORT	0.0012
41	SP_POP_DPNP_YG	0.0011
30	SP_POP_0014_FE_ZS	0.0011
25	SP_POP_6569_MA_5Y	0.0010
35	SP_POP_1519_FE_5Y	0.0010
29	SP_POP_1014_FE_5Y	0.0010
18	SP_POP_7579_MA_5Y	0.0010
10	SP_POP_80UP_MA_5Y	0.0010
40	SP_POP_5054_MA_5Y	0.0009
26	IT_MLT_MAIN_P2	0.0009
27	SP_POP_0004_MA_5Y	0.0009
51	NV_SRV_TOTL_ZS	0.0009
57	SP_POP_4549_MA_5Y	0.0009
49	SP_POP_7579_FE_5Y	0.0009
58	SE_PRE_ENRR	0.0009
45	SP_POP_6569_FE_5Y	0.0009
20	SP_DYN_T065_MA_ZS	0.0009
7	IT_NET_USER_ZS	0.0008
43	SP_DYN_AMRT_MA	0.0008
42	SP_POP_7074_FE_5Y	0.0008
62	SH_DYN_2024	0.0008
56	SP_POP_6064_FE_5Y	0.0008
22	SE_SEC_ENRR_MA	0.0008
23	SP_POP_0014_TO_ZS	0.0007
31	SP_POP_0509_FE_5Y	0.0007
36	SP_DYN_CBRT_IN	0.0007
53	SP_POP_5054_FE_5Y	0.0007
52	SP_ADO_TFRT	0.0007
28	SP_POP_DPNP_DL	0.0007
21	SP_POP_1014_MA_5Y	0.0007
9	SP_DYN_LE00_IN	0.0006
38	SP_POP_0004_FE_5Y	0.0006
60	SH_DYN_MORT_MA	0.0006
59	NV_AGR_TOTL_ZS	0.0006

13	SE_SEC_ENRR	0.0006
61	SP_DYN_TFRT_IN	0.0006
32	SP_POP_1519_MA_5Y	0.0006
1	NE_CON_PRVT_PC_KD	0.0005
15	SP_POP_0509_MA_5Y	0.0005
16	SP_POP_7074_MA_5Y	0.0005
34	SP_POP_6064_MA_5Y	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
14	SP_POP_0014_MA_ZS	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
47	SP_DYN_IMRT_MA_IN	0.0003
37	SP_POP_5559_MA_5Y	0.0003
3	NY_GNP_PCAP_CD	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0002
48	SP_DYN_IMRT_IN	0.0001

Model with 67 variables and max depth None:

Training+Validation R^2: 0.99927, RMSE: 0.51293

Testing R^2: 0.98633, RMSE: 2.40298

Mean cross-validation score: 0.98489

	Feature	Importance
0	CPI_EST_avg	0.9186
66	CPI_EST_prev	0.0300
5	NY_GDP_PCAP_KD	0.0135
19	SP_POP_80UP_FE_5Y	0.0013
17	SE_SEC_ENRR_FE	0.0013
39	SH_DYN_NMRT	0.0012
31	SP_POP_0509_FE_5Y	0.0011
12	SP_DYN_LE00_FE_IN	0.0010
29	SP_POP_1014_FE_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0010
44	EG_USE_ELEC_KH_PC	0.0010
46	SP_DYN_TO65_FE_ZS	0.0010
11	SP_POP_65UP_MA_ZS	0.0009
42	SP_POP_7074_FE_5Y	0.0008
4	NY_GDP_PCAP_KD_rel	0.0008
10	SP_POP_80UP_MA_5Y	0.0007
27	SP_POP_0004_MA_5Y	0.0007
61	SP_DYN_TFRT_IN	0.0007
63	FS_AST_PRVT_GD_ZS	0.0007
35	SP_POP_1519_FE_5Y	0.0007
18	SP_POP_7579_MA_5Y	0.0007
38	SP_POP_0004_FE_5Y	0.0007
54	SP_POP_5559_FE_5Y	0.0007
53	SP_POP_5054_FE_5Y	0.0007
30	SP_POP_0014_FE_ZS	0.0007

51	NV_SRV_TOTL_ZS	0.0007
41	SP_POP_DPND_YG	0.0006
33	SP_POP_65UP_FE_ZS	0.0006
49	SP_POP_7579_FE_5Y	0.0006
50	SP_DYN_IMRT_FE_IN	0.0006
52	SP_ADO_TFRT	0.0006
57	SP_POP_4549_MA_5Y	0.0006
58	SE_PRE_ENRR	0.0006
59	NV_AGR_TOTL_ZS	0.0006
7	IT_NET_USER_ZS	0.0006
65	SE_TER_ENRR	0.0006
20	SP_DYN_TO65_MA_ZS	0.0006
55	SH_DYN_1519	0.0005
34	SP_POP_6064_MA_5Y	0.0005
22	SE_SEC_ENRR_MA	0.0005
26	IT_MLT_MAIN_P2	0.0005
6	NY_GDP_PCAP_CD	0.0005
3	NY_GNP_PCAP_CD	0.0005
21	SP_POP_1014_MA_5Y	0.0005
25	SP_POP_6569_MA_5Y	0.0005
40	SP_POP_5054_MA_5Y	0.0004
23	SP_POP_0014_TO_ZS	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
56	SP_POP_6064_FE_5Y	0.0004
13	SE_SEC_ENRR	0.0004
47	SP_DYN_IMRT_MA_IN	0.0004
62	SH_DYN_2024	0.0004
32	SP_POP_1519_MA_5Y	0.0004
8	SP_DYN_LE00_MA_IN	0.0003
64	SH_DYN_MORT	0.0003
60	SH_DYN_MORT_MA	0.0003
9	SP_DYN_LE00_IN	0.0003
37	SP_POP_5559_MA_5Y	0.0003
15	SP_POP_0509_MA_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0003
48	SP_DYN_IMRT_IN	0.0003
28	SP_POP_DPND_OL	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
43	SP_DYN_AMRT_MA	0.0003
36	SP_DYN_CBRT_IN	0.0003
14	SP_POP_0014_MA_ZS	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002

Model with 68 variables and max depth None:
 Training+Validation R^2: 0.9976, RMSE: 0.93166
 Testing R^2: 0.98607, RMSE: 2.42532
 Mean cross-validation score: 0.98494

	Feature	Importance
0	CPI_EST_avg	0.9024
67	CPI_EST_prev	0.0357
5	NY_GDP_PCAP_KD	0.0181
44	EG_USE_ELEC_KH_PC	0.0014
12	SP_DYN_LE00_FE_IN	0.0013
11	SP_POP_65UP_MA_ZS	0.0013
4	NY_GDP_PCAP_KD_rel	0.0013
46	SP_DYN_T065_FE_ZS	0.0012
30	SP_POP_0014_FE_ZS	0.0012
17	SE_SEC_ENRR_FE	0.0012
19	SP_POP_80UP_FE_5Y	0.0012
39	SH_DYN_NMRT	0.0011
54	SP_POP_5559_FE_5Y	0.0011
64	SH_DYN_MORT	0.0010
49	SP_POP_7579_FE_5Y	0.0009
10	SP_POP_80UP_MA_5Y	0.0009
42	SP_POP_7074_FE_5Y	0.0008
20	SP_DYN_T065_MA_ZS	0.0008
45	SP_POP_6569_FE_5Y	0.0008
59	NV_AGR_TOTL_ZS	0.0008
35	SP_POP_1519_FE_5Y	0.0008
61	SP_DYN_TFRT_IN	0.0008
65	SE_TER_ENRR	0.0008
63	FS_AST_PRVT_GD_ZS	0.0008
53	SP_POP_5054_FE_5Y	0.0007
29	SP_POP_1014_FE_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
51	NV_SRV_TOTL_ZS	0.0007
31	SP_POP_0509_FE_5Y	0.0007
58	SE_PRE_ENRR	0.0007
57	SP_POP_4549_MA_5Y	0.0007
55	SH_DYN_1519	0.0006
22	SE_SEC_ENRR_MA	0.0006
52	SP_ADO_TFRT	0.0006
26	IT_MLT_MAIN_P2	0.0006
25	SP_POP_6569_MA_5Y	0.0006
48	SP_DYN_IMRT_IN	0.0006
38	SP_POP_0004_FE_5Y	0.0006
66	SH_DYN_MORT_FE	0.0006
7	IT_NET_USER_ZS	0.0006
41	SP_POP_DPND_YG	0.0006
43	SP_DYN_AMRT_MA	0.0006
60	SH_DYN_MORT_MA	0.0006
62	SH_DYN_2024	0.0005
56	SP_POP_6064_FE_5Y	0.0005
34	SP_POP_6064_MA_5Y	0.0005

32	SP_POP_1519_MA_5Y	0.0005
33	SP_POP_65UP_FE_ZS	0.0005
18	SP_POP_7579_MA_5Y	0.0005
21	SP_POP_1014_MA_5Y	0.0005
27	SP_POP_0004_MA_5Y	0.0005
15	SP_POP_0509_MA_5Y	0.0005
3	NY_GNP_PCAP_CD	0.0005
37	SP_POP_5559_MA_5Y	0.0005
1	NE_CON_PRVT_PC_KD	0.0004
13	SE_SEC_ENRR	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004
47	SP_DYN_IMRT_MA_IN	0.0004
9	SP_DYN_LE00_IN	0.0004
36	SP_DYN_CBRT_IN	0.0004
16	SP_POP_7074_MA_5Y	0.0004
40	SP_POP_5054_MA_5Y	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
23	SP_POP_0014_TO_ZS	0.0003
28	SP_POP_DPND_DL	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0003
14	SP_POP_0014_MA_ZS	0.0001

Model with 69 variables and max depth None:

Training+Validation R^2: 0.9994, RMSE: 0.46583

Testing R^2: 0.98686, RMSE: 2.3561

Mean cross-validation score: 0.98495

	Feature	Importance
0	CPI_EST_avg	0.9148
68	CPI_EST_prev	0.0298
5	NY_GDP_PCAP_KD	0.0167
4	NY_GDP_PCAP_KD_rel	0.0016
39	SH_DYN_NMRT	0.0012
12	SP_DYN_LE00_FE_IN	0.0011
17	SE_SEC_ENRR_FE	0.0011
11	SP_POP_65UP_MA_ZS	0.0010
44	EG_USE_ELEC_KH_PC	0.0010
38	SP_POP_0004_FE_5Y	0.0009
45	SP_POP_6569_FE_5Y	0.0009
46	SP_DYN_TO65_FE_ZS	0.0009
19	SP_POP_80UP_FE_5Y	0.0009
54	SP_POP_5559_FE_5Y	0.0009
61	SP_DYN_TFRT_IN	0.0008
41	SP_POP_DPND_YG	0.0008
49	SP_POP_7579_FE_5Y	0.0008
42	SP_POP_7074_FE_5Y	0.0008

20	SP_DYN_T065_MA_ZS	0.0008
53	SP_POP_5054_FE_5Y	0.0007
57	SP_POP_4549_MA_5Y	0.0007
22	SE_SEC_ENRR_MA	0.0007
18	SP_POP_7579_MA_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
59	NV_AGR_TOTL_ZS	0.0007
63	FS_AST_PRVT_GD_ZS	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
51	NV_SRV_TOTL_ZS	0.0006
7	IT_NET_USER_ZS	0.0006
3	NY_GNP_PCAP_CD	0.0006
65	SE_TER_ENRR	0.0006
35	SP_POP_1519_FE_5Y	0.0006
31	SP_POP_0509_FE_5Y	0.0006
29	SP_POP_1014_FE_5Y	0.0006
27	SP_POP_0004_MA_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
52	SP_ADO_TFRT	0.0005
66	SH_DYN_MORT_FE	0.0005
67	SP_POP_4549_FE_5Y	0.0005
56	SP_POP_6064_FE_5Y	0.0005
43	SP_DYN_AMRT_MA	0.0005
58	SE_PRE_ENRR	0.0005
34	SP_POP_6064_MA_5Y	0.0005
30	SP_POP_0014_FE_ZS	0.0005
25	SP_POP_6569_MA_5Y	0.0005
36	SP_DYN_CBRT_IN	0.0005
21	SP_POP_1014_MA_5Y	0.0005
37	SP_POP_5559_MA_5Y	0.0005
60	SH_DYN_MORT_MA	0.0004
64	SH_DYN_MORT	0.0004
55	SH_DYN_1519	0.0004
40	SP_POP_5054_MA_5Y	0.0004
32	SP_POP_1519_MA_5Y	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004
33	SP_POP_65UP_FE_ZS	0.0004
9	SP_DYN_LE00_IN	0.0004
15	SP_POP_0509_MA_5Y	0.0004
62	SH_DYN_2024	0.0003
13	SE_SEC_ENRR	0.0003
8	SP_DYN_LE00_MA_IN	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
14	SP_POP_0014_MA_ZS	0.0002
24	SP_POP_65UP_TO_ZS	0.0002
16	SP_POP_7074_MA_5Y	0.0002
28	SP_POP_DPND_OL	0.0002
48	SP_DYN_IMRT_IN	0.0002

47	SP_DYN_IMRT_MA_IN	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
23	SP_POP_0014_TO_ZS	0.0001

Model with 70 variables and max depth None:
 Training+Validation R^2: 0.99807, RMSE: 0.83637
 Testing R^2: 0.98642, RMSE: 2.39514
 Mean cross-validation score: 0.98487

	Feature	Importance
0	CPI_EST_avg	0.9207
69	CPI_EST_prev	0.0292
5	NY_GDP_PCAP_KD	0.0168
4	NY_GDP_PCAP_KD_rel	0.0012
12	SP_DYN_LEOO_FE_IN	0.0011
44	EG_USE_ELEC_KH_PC	0.0011
46	SP_DYN_TO65_FE_ZS	0.0011
19	SP_POP_80UP_FE_5Y	0.0011
17	SE_SEC_ENRR_FE	0.0011
54	SP_POP_5559_FE_5Y	0.0010
39	SH_DYN_NMRT	0.0010
41	SP_POP_DPND_YG	0.0009
61	SP_DYN_TFRT_IN	0.0008
11	SP_POP_65UP_MA_ZS	0.0007
60	SH_DYN_MORT_MA	0.0007
45	SP_POP_6569_FE_5Y	0.0007
57	SP_POP_4549_MA_5Y	0.0006
6	NY_GDP_PCAP_CD	0.0006
63	FS_AST_PRVT_GD_ZS	0.0006
65	SE_TER_ENRR	0.0006
49	SP_POP_7579_FE_5Y	0.0006
51	NV_SRV_TOTL_ZS	0.0005
3	NY_GNP_PCAP_CD	0.0005
7	IT_NET_USER_ZS	0.0005
29	SP_POP_1014_FE_5Y	0.0005
27	SP_POP_0004_MA_5Y	0.0005
25	SP_POP_6569_MA_5Y	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
22	SE_SEC_ENRR_MA	0.0005
40	SP_POP_5054_MA_5Y	0.0005
20	SP_DYN_TO65_MA_ZS	0.0005
55	SH_DYN_1519	0.0005
56	SP_POP_6064_FE_5Y	0.0005
58	SE_PRE_ENRR	0.0005
59	NV_AGR_TOTL_ZS	0.0005
35	SP_POP_1519_FE_5Y	0.0005
52	SP_ADO_TFRT	0.0004

53	SP_POP_5054_FE_5Y	0.0004
43	SP_DYN_AMRT_MA	0.0004
42	SP_POP_7074_FE_5Y	0.0004
32	SP_POP_1519_MA_5Y	0.0004
38	SP_POP_0004_FE_5Y	0.0004
68	SP_DYN_AMRT_FE	0.0004
33	SP_POP_65UP_FE_ZS	0.0004
31	SP_POP_0509_FE_5Y	0.0004
14	SP_POP_0014_MA_ZS	0.0004
15	SP_POP_0509_MA_5Y	0.0004
26	IT_MLT_MAIN_P2	0.0004
18	SP_POP_7579_MA_5Y	0.0004
21	SP_POP_1014_MA_5Y	0.0003
67	SP_POP_4549_FE_5Y	0.0003
13	SE_SEC_ENRR	0.0003
16	SP_POP_7074_MA_5Y	0.0003
62	SH_DYN_2024	0.0003
23	SP_POP_0014_TO_ZS	0.0003
37	SP_POP_5559_MA_5Y	0.0003
50	SP_DYN_IMRT_FE_IN	0.0003
28	SP_POP_DPND_DL	0.0003
30	SP_POP_0014_FE_ZS	0.0003
34	SP_POP_6064_MA_5Y	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
64	SH_DYN_MORT	0.0002
66	SH_DYN_MORT_FE	0.0002
36	SP_DYN_CBRT_IN	0.0002
47	SP_DYN_IMRT_MA_IN	0.0002
24	SP_POP_65UP_TO_ZS	0.0002
9	SP_DYN_LEOO_IN	0.0002
8	SP_DYN_LEOO_MA_IN	0.0002
48	SP_DYN_IMRT_IN	0.0001
2	NY_ADJ_NNTY_PC_CD	0.0001

Model with 71 variables and max depth None:

Training+Validation R^2: 0.9988, RMSE: 0.65957

Testing R^2: 0.98639, RMSE: 2.39747

Mean cross-validation score: 0.98452

	Feature	Importance
0	CPI_EST_avg	0.8934
70	CPI_EST_prev	0.0296
5	NY_GDP_PCAP_KD	0.0279
4	NY_GDP_PCAP_KD_rel	0.0020
17	SE_SEC_ENRR_FE	0.0016
39	SH_DYN_NMRT	0.0015
12	SP_DYN_LEOO_FE_IN	0.0013

44	EG_USE_ELEC_KH_PC	0.0013
19	SP_POP_80UP_FE_5Y	0.0012
54	SP_POP_5559_FE_5Y	0.0012
55	SH_DYN_1519	0.0012
43	SP_DYN_AMRT_MA	0.0011
46	SP_DYN_T065_FE_ZS	0.0011
11	SP_POP_65UP_MA_ZS	0.0011
51	NV_SRV_TOTL_ZS	0.0011
22	SE_SEC_ENRR_MA	0.0010
63	FS_AST_PRVT_GD_ZS	0.0010
6	NY_GDP_PCAP_CD	0.0009
42	SP_POP_7074_FE_5Y	0.0009
7	IT_NET_USER_ZS	0.0009
23	SP_POP_0014_TO_ZS	0.0009
49	SP_POP_7579_FE_5Y	0.0009
52	SP_ADO_TFRT	0.0009
20	SP_DYN_T065_MA_ZS	0.0009
57	SP_POP_4549_MA_5Y	0.0009
45	SP_POP_6569_FE_5Y	0.0009
15	SP_POP_0509_MA_5Y	0.0009
40	SP_POP_5054_MA_5Y	0.0009
31	SP_POP_0509_FE_5Y	0.0008
59	NV_AGR_TOTL_ZS	0.0008
32	SP_POP_1519_MA_5Y	0.0008
38	SP_POP_0004_FE_5Y	0.0008
29	SP_POP_1014_FE_5Y	0.0008
27	SP_POP_0004_MA_5Y	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
35	SP_POP_1519_FE_5Y	0.0007
53	SP_POP_5054_FE_5Y	0.0007
3	NY_GNP_PCAP_CD	0.0007
26	IT_MLT_MAIN_P2	0.0007
65	SE_TER_ENRR	0.0007
21	SP_POP_1014_MA_5Y	0.0007
18	SP_POP_7579_MA_5Y	0.0007
58	SE_PRE_ENRR	0.0007
56	SP_POP_6064_FE_5Y	0.0006
61	SP_DYN_TFRT_IN	0.0006
50	SP_DYN_IMRT_FE_IN	0.0006
68	SP_DYN_AMRT_FE	0.0006
69	FD_AST_PRVT_GD_ZS	0.0006
25	SP_POP_6569_MA_5Y	0.0005
1	NE_CON_PRVT_PC_KD	0.0005
48	SP_DYN_IMRT_IN	0.0005
13	SE_SEC_ENRR	0.0005
62	SH_DYN_2024	0.0005
67	SP_POP_4549_FE_5Y	0.0004
36	SP_DYN_CBRT_IN	0.0004

34	SP_POP_6064_MA_5Y	0.0004
28	SP_POP_DPND_DL	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
33	SP_POP_65UP_FE_ZS	0.0003
60	SH_DYN_MORT_MA	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
16	SP_POP_7074_MA_5Y	0.0003
66	SH_DYN_MORT_FE	0.0003
9	SP_DYN_LE00_IN	0.0003
47	SP_DYN_IMRT_MA_IN	0.0002
41	SP_POP_DPND_YG	0.0002
37	SP_POP_5559_MA_5Y	0.0002
14	SP_POP_0014_MA_ZS	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
64	SH_DYN_MORT	0.0001
30	SP_POP_0014_FE_ZS	0.0000

Model with 72 variables and max depth None:

Training+Validation R^2: 0.99971, RMSE: 0.32136

Testing R^2: 0.98615, RMSE: 2.41844

Mean cross-validation score: 0.98485

	Feature	Importance
0	CPI_EST_avg	0.9118
71	CPI_EST_prev	0.0275
5	NY_GDP_PCAP_KD	0.0229
4	NY_GDP_PCAP_KD_rel	0.0014
17	SE_SEC_ENRR_FE	0.0011
12	SP_DYN_LE00_FE_IN	0.0011
46	SP_DYN_TO65_FE_ZS	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
44	EG_USE_ELEC_KH_PC	0.0009
55	SH_DYN_1519	0.0009
54	SP_POP_5559_FE_5Y	0.0009
19	SP_POP_80UP_FE_5Y	0.0009
39	SH_DYN_NMRT	0.0009
38	SP_POP_0004_FE_5Y	0.0009
70	FM_AST_PRVT_GD_ZS	0.0008
51	NV_SRV_TOTL_ZS	0.0008
20	SP_DYN_TO65_MA_ZS	0.0008
49	SP_POP_7579_FE_5Y	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
45	SP_POP_6569_FE_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
31	SP_POP_0509_FE_5Y	0.0007
59	NV_AGR_TOTL_ZS	0.0006
41	SP_POP_DPND_YG	0.0006

61	SP_DYN_TFRT_IN	0.0006
63	FS_AST_PRVT_GD_ZS	0.0006
57	SP_POP_4549_MA_5Y	0.0006
22	SE_SEC_ENRR_MA	0.0006
26	IT_MLT_MAIN_P2	0.0006
65	SE_TER_ENRR	0.0006
18	SP_POP_7579_MA_5Y	0.0006
7	IT_NET_USER_ZS	0.0006
27	SP_POP_0004_MA_5Y	0.0005
29	SP_POP_1014_FE_5Y	0.0005
32	SP_POP_1519_MA_5Y	0.0005
52	SP_ADO_TFRT	0.0005
34	SP_POP_6064_MA_5Y	0.0005
43	SP_DYN_AMRT_MA	0.0005
21	SP_POP_1014_MA_5Y	0.0005
15	SP_POP_0509_MA_5Y	0.0005
42	SP_POP_7074_FE_5Y	0.0005
53	SP_POP_5054_FE_5Y	0.0004
56	SP_POP_6064_FE_5Y	0.0004
30	SP_POP_0014_FE_ZS	0.0004
40	SP_POP_5054_MA_5Y	0.0004
3	NY_GNP_PCAP_CD	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
35	SP_POP_1519_FE_5Y	0.0004
58	SE_PRE_ENRR	0.0004
28	SP_POP_DPNP_DL	0.0004
67	SP_POP_4549_FE_5Y	0.0004
69	FD_AST_PRVT_GD_ZS	0.0004
13	SE_SEC_ENRR	0.0004
9	SP_DYN_LEOO_IN	0.0004
8	SP_DYN_LEOO_MA_IN	0.0004
60	SH_DYN_MORT_MA	0.0003
66	SH_DYN_MORT_FE	0.0003
37	SP_POP_5559_MA_5Y	0.0003
36	SP_DYN_CBRT_IN	0.0003
50	SP_DYN_IMRT_FE_IN	0.0003
48	SP_DYN_IMRT_IN	0.0003
33	SP_POP_65UP_FE_ZS	0.0003
25	SP_POP_6569_MA_5Y	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
23	SP_POP_0014_TO_ZS	0.0003
16	SP_POP_7074_MA_5Y	0.0003
47	SP_DYN_IMRT_MA_IN	0.0002
62	SH_DYN_2024	0.0002
68	SP_DYN_AMRT_FE	0.0002
14	SP_POP_0014_MA_ZS	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
64	SH_DYN_MORT	0.0001

Model with 73 variables and max depth None:
 Training+Validation R^2: 0.999, RMSE: 0.60198
 Testing R^2: 0.98654, RMSE: 2.38428
 Mean cross-validation score: 0.98497

	Feature	Importance
0	CPI_EST_avg	0.9209
72	CPI_EST_prev	0.0284
5	NY_GDP_PCAP_KD	0.0152
4	NY_GDP_PCAP_KD_rel	0.0016
17	SE_SEC_ENRR_FE	0.0010
39	SH_DYN_NMRT	0.0009
12	SP_DYN_LE00_FE_IN	0.0009
54	SP_POP_5559_FE_5Y	0.0009
19	SP_POP_80UP_FE_5Y	0.0009
22	SE_SEC_ENRR_MA	0.0008
44	EG_USE_ELEC_KH_PC	0.0008
70	FM_AST_PRVT_GD_ZS	0.0008
49	SP_POP_7579_FE_5Y	0.0008
6	NY_GDP_PCAP_CD	0.0008
11	SP_POP_65UP_MA_ZS	0.0007
46	SP_DYN_T065_FE_ZS	0.0007
18	SP_POP_7579_MA_5Y	0.0007
45	SP_POP_6569_FE_5Y	0.0007
33	SP_POP_65UP_FE_ZS	0.0006
48	SP_DYN_IMRT_IN	0.0006
51	NV_SRV_TOTL_ZS	0.0006
53	SP_POP_5054_FE_5Y	0.0006
55	SH_DYN_1519	0.0006
42	SP_POP_7074_FE_5Y	0.0006
66	SH_DYN_MORT_FE	0.0006
57	SP_POP_4549_MA_5Y	0.0006
16	SP_POP_7074_MA_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0005
40	SP_POP_5054_MA_5Y	0.0005
56	SP_POP_6064_FE_5Y	0.0005
7	IT_NET_USER_ZS	0.0005
71	TM_VAL_MRCH_HI_ZS	0.0005
65	SE_TER_ENRR	0.0005
52	SP_ADO_TFRT	0.0005
20	SP_DYN_T065_MA_ZS	0.0005
21	SP_POP_1014_MA_5Y	0.0005
41	SP_POP_DPND_YG	0.0005
63	FS_AST_PRVT_GD_ZS	0.0005
29	SP_POP_1014_FE_5Y	0.0005
58	SE_PRE_ENRR	0.0005

35	SP_POP_1519_FE_5Y	0.0005
32	SP_POP_1519_MA_5Y	0.0005
15	SP_POP_0509_MA_5Y	0.0004
68	SP_DYN_AMRT_FE	0.0004
28	SP_POP_DPND_DL	0.0004
31	SP_POP_0509_FE_5Y	0.0004
10	SP_POP_80UP_MA_5Y	0.0004
69	FD_AST_PRVT_GD_ZS	0.0004
30	SP_POP_0014_FE_ZS	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
59	NV_AGR_TOTL_ZS	0.0004
67	SP_POP_4549_FE_5Y	0.0004
43	SP_DYN_AMRT_MA	0.0004
27	SP_POP_0004_MA_5Y	0.0004
61	SP_DYN_TFRT_IN	0.0004
3	NY_GNP_PCAP_CD	0.0004
62	SH_DYN_2024	0.0003
60	SH_DYN_MORT_MA	0.0003
37	SP_POP_5559_MA_5Y	0.0003
47	SP_DYN_IMRT_MA_IN	0.0003
38	SP_POP_0004_FE_5Y	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
25	SP_POP_6569_MA_5Y	0.0003
23	SP_POP_0014_TO_ZS	0.0003
13	SE_SEC_ENRR	0.0003
9	SP_DYN_LE00_IN	0.0003
34	SP_POP_6064_MA_5Y	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0002
50	SP_DYN_IMRT_FE_IN	0.0002
8	SP_DYN_LE00_MA_IN	0.0002
36	SP_DYN_CBRT_IN	0.0002
64	SH_DYN_MORT	0.0001
14	SP_POP_0014_MA_ZS	0.0001

Model with 74 variables and max depth None:

Training+Validation R^2: 0.99822, RMSE: 0.80276

Testing R^2: 0.98641, RMSE: 2.39627

Mean cross-validation score: 0.98485

	Feature	Importance
0	CPI_EST_avg	0.8997
73	CPI_EST_prev	0.0353
5	NY_GDP_PCAP_KD	0.0170
4	NY_GDP_PCAP_KD_rel	0.0020
39	SH_DYN_NMRT	0.0020
46	SP_DYN_TO65_FE_ZS	0.0012
11	SP_POP_65UP_MA_ZS	0.0012

44	EG_USE_ELEC_KH_PC	0.0012
31	SP_POP_0509_FE_5Y	0.0012
12	SP_DYN_LE00_FE_IN	0.0011
30	SP_POP_0014_FE_ZS	0.0011
17	SE_SEC_ENRR_FE	0.0011
19	SP_POP_80UP_FE_5Y	0.0011
49	SP_POP_7579_FE_5Y	0.0010
54	SP_POP_5559_FE_5Y	0.0009
42	SP_POP_7074_FE_5Y	0.0009
20	SP_DYN_T065_MA_ZS	0.0009
35	SP_POP_1519_FE_5Y	0.0009
33	SP_POP_65UP_FE_ZS	0.0009
65	SE_TER_ENRR	0.0008
45	SP_POP_6569_FE_5Y	0.0008
55	SH_DYN_1519	0.0008
40	SP_POP_5054_MA_5Y	0.0008
57	SP_POP_4549_MA_5Y	0.0008
63	FS_AST_PRVT_GD_ZS	0.0008
72	SP_POP_2024_FE_5Y	0.0008
41	SP_POP_DPND_YG	0.0007
29	SP_POP_1014_FE_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
53	SP_POP_5054_FE_5Y	0.0007
66	SH_DYN_MORT_FE	0.0007
51	NV_SRV_TOTL_ZS	0.0007
69	FD_AST_PRVT_GD_ZS	0.0007
67	SP_POP_4549_FE_5Y	0.0006
59	NV_AGR_TOTL_ZS	0.0006
58	SE_PRE_ENRR	0.0006
52	SP_ADO_TFRT	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
43	SP_DYN_AMRT_MA	0.0006
70	FM_AST_PRVT_GD_ZS	0.0006
38	SP_POP_0004_FE_5Y	0.0006
18	SP_POP_7579_MA_5Y	0.0006
22	SE_SEC_ENRR_MA	0.0006
27	SP_POP_0004_MA_5Y	0.0006
23	SP_POP_0014_TO_ZS	0.0006
32	SP_POP_1519_MA_5Y	0.0006
36	SP_DYN_CBRT_IN	0.0005
56	SP_POP_6064_FE_5Y	0.0005
61	SP_DYN_TFRT_IN	0.0005
60	SH_DYN_MORT_MA	0.0005
28	SP_POP_DPND_DL	0.0005
7	IT_NET_USER_ZS	0.0005
13	SE_SEC_ENRR	0.0005
71	TM_VAL_MRCH_HI_ZS	0.0005

15	SP_POP_0509_MA_5Y	0.0005
3	NY_GNP_PCAP_CD	0.0004
14	SP_POP_0014_MA_ZS	0.0004
48	SP_DYN_IMRT_IN	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
1	NE_CON_PRVT_PC_KD	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004
62	SH_DYN_2024	0.0004
64	SH_DYN_MORT	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0003
47	SP_DYN_IMRT_MA_IN	0.0003
34	SP_POP_6064_MA_5Y	0.0003
25	SP_POP_6569_MA_5Y	0.0003
21	SP_POP_1014_MA_5Y	0.0003
9	SP_DYN_LE00_IN	0.0003
37	SP_POP_5559_MA_5Y	0.0003
68	SP_DYN_AMRT_FE	0.0002
16	SP_POP_7074_MA_5Y	0.0002
8	SP_DYN_LE00_MA_IN	0.0002

Model with 75 variables and max depth None:

Training+Validation R^2: 0.99867, RMSE: 0.69483

Testing R^2: 0.98612, RMSE: 2.42136

Mean cross-validation score: 0.985

	Feature	Importance
0	CPI_EST_avg	0.9206
74	CPI_EST_prev	0.0313
5	NY_GDP_PCAP_KD	0.0122
39	SH_DYN_NMRT	0.0015
4	NY_GDP_PCAP_KD_rel	0.0013
17	SE_SEC_ENRR_FE	0.0012
38	SP_POP_0004_FE_5Y	0.0011
33	SP_POP_65UP_FE_ZS	0.0010
49	SP_POP_7579_FE_5Y	0.0009
54	SP_POP_5559_FE_5Y	0.0009
12	SP_DYN_LE00_FE_IN	0.0009
11	SP_POP_65UP_MA_ZS	0.0008
46	SP_DYN_TO65_FE_ZS	0.0008
63	FS_AST_PRVT_GD_ZS	0.0007
51	NV_SRV_TOTL_ZS	0.0007
18	SP_POP_7579_MA_5Y	0.0007
19	SP_POP_80UP_FE_5Y	0.0007
44	EG_USE_ELEC_KH_PC	0.0007
23	SP_POP_0014_TO_ZS	0.0006
35	SP_POP_1519_FE_5Y	0.0006
45	SP_POP_6569_FE_5Y	0.0006

52	SP_ADO_TFRT	0.0006
58	SE_PRE_ENRR	0.0006
43	SP_DYN_AMRT_MA	0.0006
65	SE_TER_ENRR	0.0006
6	NY_GDP_PCAP_CD	0.0006
59	NV_AGR_TOTL_ZS	0.0006
72	SP_POP_2024_FE_5Y	0.0006
20	SP_DYN_TO65_MA_ZS	0.0006
67	SP_POP_4549_FE_5Y	0.0006
32	SP_POP_1519_MA_5Y	0.0005
40	SP_POP_5054_MA_5Y	0.0005
42	SP_POP_7074_FE_5Y	0.0005
61	SP_DYN_TFRT_IN	0.0005
55	SH_DYN_1519	0.0005
56	SP_POP_6064_FE_5Y	0.0005
22	SE_SEC_ENRR_MA	0.0005
57	SP_POP_4549_MA_5Y	0.0004
69	FD_AST_PRVT_GD_ZS	0.0004
70	FM_AST_PRVT_GD_ZS	0.0004
53	SP_POP_5054_FE_5Y	0.0004
73	SE_PRM_ENRL_TC_ZS	0.0004
37	SP_POP_5559_MA_5Y	0.0004
21	SP_POP_1014_MA_5Y	0.0004
10	SP_POP_80UP_MA_5Y	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
25	SP_POP_6569_MA_5Y	0.0004
26	IT_MLT_MAIN_P2	0.0004
13	SE_SEC_ENRR	0.0004
29	SP_POP_1014_FE_5Y	0.0004
9	SP_DYN_LE00_IN	0.0004
7	IT_NET_USER_ZS	0.0004
50	SP_DYN_IMRT_FE_IN	0.0003
31	SP_POP_0509_FE_5Y	0.0003
64	SH_DYN_MORT	0.0003
28	SP_POP_DPNP_OL	0.0003
71	TM_VAL_MRCH_HI_ZS	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
66	SH_DYN_MORT_FE	0.0003
60	SH_DYN_MORT_MA	0.0003
62	SH_DYN_2024	0.0003
8	SP_DYN_LE00_MA_IN	0.0002
14	SP_POP_0014_MA_ZS	0.0002
41	SP_POP_DPNP_YG	0.0002
15	SP_POP_0509_MA_5Y	0.0002
16	SP_POP_7074_MA_5Y	0.0002
3	NY_GNP_PCAP_CD	0.0002
34	SP_POP_6064_MA_5Y	0.0002
47	SP_DYN_IMRT_MA_IN	0.0002

36	SP_DYN_CBRT_IN	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
27	SP_POP_0004_MA_5Y	0.0001
68	SP_DYN_AMRT_FE	0.0001
48	SP_DYN_IMRT_IN	0.0001
30	SP_POP_0014_FE_ZS	0.0000

Model with 76 variables and max depth None:

Training+Validation R^2: 0.99974, RMSE: 0.30759

Testing R^2: 0.98681, RMSE: 2.36051

Mean cross-validation score: 0.9846

	Feature	Importance
0	CPI_EST_avg	0.9195
75	CPI_EST_prev	0.0384
39	SH_DYN_NMRT	0.0015
4	NY_GDP_PCAP_KD_rel	0.0015
17	SE_SEC_ENRR_FE	0.0015
54	SP_POP_5559_FE_5Y	0.0014
19	SP_POP_80UP_FE_5Y	0.0012
63	FS_AST_PRVT_GD_ZS	0.0011
74	SP_RUR_TOTL_ZS	0.0009
44	EG_USE_ELEC_KH_PC	0.0008
45	SP_POP_6569_FE_5Y	0.0008
51	NV_SRV_TOTL_ZS	0.0008
53	SP_POP_5054_FE_5Y	0.0008
12	SP_DYN_LE00_FE_IN	0.0008
30	SP_POP_0014_FE_ZS	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
46	SP_DYN_T065_FE_ZS	0.0007
55	SH_DYN_1519	0.0007
57	SP_POP_4549_MA_5Y	0.0007
59	NV_AGR_TOTL_ZS	0.0007
72	SP_POP_2024_FE_5Y	0.0007
42	SP_POP_7074_FE_5Y	0.0007
38	SP_POP_0004_FE_5Y	0.0007
22	SE_SEC_ENRR_MA	0.0007
20	SP_DYN_T065_MA_ZS	0.0007
18	SP_POP_7579_MA_5Y	0.0007
49	SP_POP_7579_FE_5Y	0.0006
37	SP_POP_5559_MA_5Y	0.0006
70	FM_AST_PRVT_GD_ZS	0.0006
65	SE_TER_ENRR	0.0006
32	SP_POP_1519_MA_5Y	0.0006
7	IT_NET_USER_ZS	0.0006
6	NY_GDP_PCAP_CD	0.0006
10	SP_POP_80UP_MA_5Y	0.0006

56	SP_POP_6064_FE_5Y	0.0006
58	SE_PRE_ENRR	0.0006
52	SP_ADO_TFRT	0.0005
41	SP_POP_DPND_YG	0.0005
47	SP_DYN_IMRT_MA_IN	0.0005
43	SP_DYN_AMRT_MA	0.0005
61	SP_DYN_TFRT_IN	0.0005
21	SP_POP_1014_MA_5Y	0.0005
40	SP_POP_5054_MA_5Y	0.0005
5	NY_GDP_PCAP_KD	0.0005
35	SP_POP_1519_FE_5Y	0.0005
31	SP_POP_0509_FE_5Y	0.0004
9	SP_DYN_LE00_IN	0.0004
26	IT_MLT_MAIN_P2	0.0004
73	SE_PRM_ENRL_TC_ZS	0.0004
13	SE_SEC_ENRR	0.0004
29	SP_POP_1014_FE_5Y	0.0004
71	TM_VAL_MRCH_HI_ZS	0.0004
3	NY_GNP_PCAP_CD	0.0004
66	SH_DYN_MORT_FE	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
48	SP_DYN_IMRT_IN	0.0004
67	SP_POP_4549_FE_5Y	0.0004
69	FD_AST_PRVT_GD_ZS	0.0004
60	SH_DYN_MORT_MA	0.0004
68	SP_DYN_AMRT_FE	0.0003
62	SH_DYN_2024	0.0003
23	SP_POP_0014_TO_ZS	0.0003
8	SP_DYN_LE00_MA_IN	0.0003
15	SP_POP_0509_MA_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
33	SP_POP_65UP_FE_ZS	0.0003
27	SP_POP_0004_MA_5Y	0.0003
64	SH_DYN_MORT	0.0002
50	SP_DYN_IMRT_FE_IN	0.0002
36	SP_DYN_CBRT_IN	0.0002
34	SP_POP_6064_MA_5Y	0.0002
28	SP_POP_DPND_OL	0.0002
25	SP_POP_6569_MA_5Y	0.0002
14	SP_POP_0014_MA_ZS	0.0001
2	NY_ADJ_NNTY_PC_CD	0.0001

Model with 77 variables and max depth None:

Training+Validation R^2: 0.99903, RMSE: 0.59173

Testing R^2: 0.98644, RMSE: 2.39346

Mean cross-validation score: 0.9846

	Feature	Importance
0	CPI_EST_avg	0.9177
76	CPI_EST_prev	0.0364
4	NY_GDP_PCAP_KD_rel	0.0018
39	SH_DYN_NMRT	0.0016
17	SE_SEC_ENRR_FE	0.0015
12	SP_DYN_LE00_FE_IN	0.0013
54	SP_POP_5559_FE_5Y	0.0012
44	EG_USE_ELEC_KH_PC	0.0011
63	FS_AST_PRVT_GD_ZS	0.0011
11	SP_POP_65UP_MA_ZS	0.0010
45	SP_POP_6569_FE_5Y	0.0010
42	SP_POP_7074_FE_5Y	0.0010
49	SP_POP_7579_FE_5Y	0.0009
46	SP_DYN_T065_FE_ZS	0.0009
19	SP_POP_80UP_FE_5Y	0.0009
72	SP_POP_2024_FE_5Y	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
75	SP_URB_TOTL_IN_ZS	0.0008
32	SP_POP_1519_MA_5Y	0.0008
31	SP_POP_0509_FE_5Y	0.0008
59	NV_AGR_TOTL_ZS	0.0008
51	NV_SRV_TOTL_ZS	0.0007
53	SP_POP_5054_FE_5Y	0.0007
35	SP_POP_1519_FE_5Y	0.0007
57	SP_POP_4549_MA_5Y	0.0007
65	SE_TER_ENRR	0.0007
23	SP_POP_0014_TO_ZS	0.0007
22	SE_SEC_ENRR_MA	0.0007
66	SH_DYN_MORT_FE	0.0007
7	IT_NET_USER_ZS	0.0007
74	SP_RUR_TOTL_ZS	0.0007
70	FM_AST_PRVT_GD_ZS	0.0007
24	SP_POP_65UP_TO_ZS	0.0006
61	SP_DYN_TFRT_IN	0.0006
6	NY_GDP_PCAP_CD	0.0006
43	SP_DYN_AMRT_MA	0.0006
40	SP_POP_5054_MA_5Y	0.0006
50	SP_DYN_IMRT_FE_IN	0.0006
18	SP_POP_7579_MA_5Y	0.0006
56	SP_POP_6064_FE_5Y	0.0006
33	SP_POP_65UP_FE_ZS	0.0006
20	SP_DYN_T065_MA_ZS	0.0006
69	FD_AST_PRVT_GD_ZS	0.0005
52	SP_ADO_TFRT	0.0005
55	SH_DYN_1519	0.0005
71	TM_VAL_MRCH_HI_ZS	0.0005

58	SE_PRE_ENRR	0.0005
67	SP_POP_4549_FE_5Y	0.0005
38	SP_POP_0004_FE_5Y	0.0005
15	SP_POP_0509_MA_5Y	0.0005
29	SP_POP_1014_FE_5Y	0.0004
68	SP_DYN_AMRT_FE	0.0004
16	SP_POP_7074_MA_5Y	0.0004
21	SP_POP_1014_MA_5Y	0.0004
25	SP_POP_6569_MA_5Y	0.0004
47	SP_DYN_IMRT_MA_IN	0.0004
27	SP_POP_0004_MA_5Y	0.0004
13	SE_SEC_ENRR	0.0004
26	IT_MLT_MAIN_P2	0.0004
37	SP_POP_5559_MA_5Y	0.0004
3	NY_GNP_PCAP_CD	0.0004
73	SE_PRM_ENRL_TC_ZS	0.0004
30	SP_POP_0014_FE_ZS	0.0003
8	SP_DYN_LEOO_MA_IN	0.0003
62	SH_DYN_2024	0.0003
36	SP_DYN_CBRT_IN	0.0003
64	SH_DYN_MORT	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
5	NY_GDP_PCAP_KD	0.0003
9	SP_DYN_LEOO_IN	0.0002
48	SP_DYN_IMRT_IN	0.0002
28	SP_POP_DPND_OL	0.0002
34	SP_POP_6064_MA_5Y	0.0002
41	SP_POP_DPND_YG	0.0002
14	SP_POP_0014_MA_ZS	0.0001
60	SH_DYN_MORT_MA	0.0001
2	NY_ADJ_NNTY_PC_CD	0.0001

Model with 78 variables and max depth None:

Training+Validation R^2: 0.99933, RMSE: 0.49083

Testing R^2: 0.98665, RMSE: 2.37437

Mean cross-validation score: 0.98501

	Feature	Importance
0	CPI_EST_avg	0.9078
77	CPI_EST_prev	0.0411
4	NY_GDP_PCAP_KD_rel	0.0021
17	SE_SEC_ENRR_FE	0.0019
12	SP_DYN_LEOO_FE_IN	0.0018
39	SH_DYN_NMRT	0.0017
19	SP_POP_80UP_FE_5Y	0.0016
45	SP_POP_6569_FE_5Y	0.0012
30	SP_POP_0014_FE_ZS	0.0012

44	EG_USE_ELEC_KH_PC	0.0011
61	SP_DYN_TFRT_IN	0.0011
76	SH_DYN_1014	0.0011
54	SP_POP_5559_FE_5Y	0.0011
74	SP_RUR_TOTL_ZS	0.0010
63	FS_AST_PRVT_GD_ZS	0.0010
46	SP_DYN_T065_FE_ZS	0.0010
72	SP_POP_2024_FE_5Y	0.0009
51	NV_SRV_TOTL_ZS	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
18	SP_POP_7579_MA_5Y	0.0009
27	SP_POP_0004_MA_5Y	0.0009
50	SP_DYN_IMRT_FE_IN	0.0008
22	SE_SEC_ENRR_MA	0.0008
57	SP_POP_4549_MA_5Y	0.0008
65	SE_TER_ENRR	0.0008
53	SP_POP_5054_FE_5Y	0.0007
52	SP_ADO_TFRT	0.0007
26	IT_MLT_MAIN_P2	0.0007
59	NV_AGR_TOTL_ZS	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
32	SP_POP_1519_MA_5Y	0.0007
42	SP_POP_7074_FE_5Y	0.0007
35	SP_POP_1519_FE_5Y	0.0007
3	NY_GNP_PCAP_CD	0.0007
69	FD_AST_PRVT_GD_ZS	0.0006
38	SP_POP_0004_FE_5Y	0.0006
58	SE_PRE_ENRR	0.0006
40	SP_POP_5054_MA_5Y	0.0006
6	NY_GDP_PCAP_CD	0.0006
24	SP_POP_65UP_TO_ZS	0.0006
7	IT_NET_USER_ZS	0.0006
71	TM_VAL_MRCH_HI_ZS	0.0005
49	SP_POP_7579_FE_5Y	0.0005
67	SP_POP_4549_FE_5Y	0.0005
20	SP_DYN_T065_MA_ZS	0.0005
56	SP_POP_6064_FE_5Y	0.0005
55	SH_DYN_1519	0.0005
70	FM_AST_PRVT_GD_ZS	0.0005
13	SE_SEC_ENRR	0.0005
73	SE_PRM_ENRL_TC_ZS	0.0005
47	SP_DYN_IMRT_MA_IN	0.0005
29	SP_POP_1014_FE_5Y	0.0005
75	SP_URB_TOTL_IN_ZS	0.0005
43	SP_DYN_AMRT_MA	0.0005
41	SP_POP_DPND_YG	0.0005
5	NY_GDP_PCAP_KD	0.0005
34	SP_POP_6064_MA_5Y	0.0004

1	NE_CON_PRVT_PC_KD	0.0004
62	SH_DYN_2024	0.0004
37	SP_POP_5559_MA_5Y	0.0004
64	SH_DYN_MORT	0.0003
8	SP_DYN_LEOO_MA_IN	0.0003
68	SP_DYN_AMRT_FE	0.0003
66	SH_DYN_MORT_FE	0.0003
16	SP_POP_7074_MA_5Y	0.0003
15	SP_POP_0509_MA_5Y	0.0003
21	SP_POP_1014_MA_5Y	0.0003
23	SP_POP_0014_TO_ZS	0.0003
25	SP_POP_6569_MA_5Y	0.0003
28	SP_POP_DPND_DL	0.0003
48	SP_DYN_IMRT_IN	0.0003
31	SP_POP_0509_FE_5Y	0.0003
33	SP_POP_65UP_FE_ZS	0.0003
14	SP_POP_0014_MA_ZS	0.0002
60	SH_DYN_MORT_MA	0.0002
36	SP_DYN_CBRT_IN	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
9	SP_DYN_LEOO_IN	0.0001

Model with 79 variables and max depth None:

Training+Validation R^2: 0.99891, RMSE: 0.62674

Testing R^2: 0.98621, RMSE: 2.41369

Mean cross-validation score: 0.985

	Feature	Importance
0	CPI_EST_avg	0.9207
78	CPI_EST_prev	0.0348
4	NY_GDP_PCAP_KD_rel	0.0023
54	SP_POP_5559_FE_5Y	0.0015
17	SE_SEC_ENRR_FE	0.0013
45	SP_POP_6569_FE_5Y	0.0012
63	FS_AST_PRVT_GD_ZS	0.0012
39	SH_DYN_NMRT	0.0011
33	SP_POP_65UP_FE_ZS	0.0010
12	SP_DYN_LEOO_FE_IN	0.0010
44	EG_USE_ELEC_KH_PC	0.0010
70	FM_AST_PRVT_GD_ZS	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
61	SP_DYN_TFRT_IN	0.0008
74	SP_RUR_TOTL_ZS	0.0008
51	NV_SRV_TOTL_ZS	0.0008
72	SP_POP_2024_FE_5Y	0.0008
19	SP_POP_80UP_FE_5Y	0.0008
57	SP_POP_4549_MA_5Y	0.0008

30	SP_POP_0014_FE_ZS	0.0008
35	SP_POP_1519_FE_5Y	0.0007
49	SP_POP_7579_FE_5Y	0.0007
59	NV_AGR_TOTL_ZS	0.0007
38	SP_POP_0004_FE_5Y	0.0007
56	SP_POP_6064_FE_5Y	0.0006
32	SP_POP_1519_MA_5Y	0.0006
65	SE_TER_ENRR	0.0006
42	SP_POP_7074_FE_5Y	0.0006
55	SH_DYN_1519	0.0006
67	SP_POP_4549_FE_5Y	0.0006
69	FD_AST_PRVT_GD_ZS	0.0006
23	SP_POP_0014_TO_ZS	0.0006
22	SE_SEC_ENRR_MA	0.0006
18	SP_POP_7579_MA_5Y	0.0006
46	SP_DYN_T065_FE_ZS	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
75	SP_URB_TOTL_IN_ZS	0.0006
5	NY_GDP_PCAP_KD	0.0006
58	SE_PRE_ENRR	0.0006
53	SP_POP_5054_FE_5Y	0.0005
26	IT_MLT_MAIN_P2	0.0005
37	SP_POP_5559_MA_5Y	0.0005
29	SP_POP_1014_FE_5Y	0.0005
21	SP_POP_1014_MA_5Y	0.0005
73	SE_PRM_ENRL_TC_ZS	0.0005
13	SE_SEC_ENRR	0.0005
76	SH_DYN_1014	0.0005
7	IT_NET_USER_ZS	0.0005
77	EG_USE_PCAP_KG_OE	0.0005
71	TM_VAL_MRCH_HI_ZS	0.0004
52	SP_ADO_TFRT	0.0004
40	SP_POP_5054_MA_5Y	0.0004
9	SP_DYN_LEOO_IN	0.0004
47	SP_DYN_IMRT_MA_IN	0.0004
43	SP_DYN_AMRT_MA	0.0004
6	NY_GDP_PCAP_CD	0.0004
8	SP_DYN_LEOO_MA_IN	0.0004
36	SP_DYN_CBRT_IN	0.0004
20	SP_DYN_T065_MA_ZS	0.0004
27	SP_POP_0004_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
34	SP_POP_6064_MA_5Y	0.0003
25	SP_POP_6569_MA_5Y	0.0003
68	SP_DYN_AMRT_FE	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
31	SP_POP_0509_FE_5Y	0.0003
62	SH_DYN_2024	0.0003

60	SH_DYN_MORT_MA	0.0003
3	NY_GNP_PCAP_CD	0.0003
15	SP_POP_0509_MA_5Y	0.0002
50	SP_DYN_IMRT_FE_IN	0.0002
28	SP_POP_DPNP_DL	0.0002
48	SP_DYN_IMRT_IN	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
41	SP_POP_DPNP_YG	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
66	SH_DYN_MORT_FE	0.0001
14	SP_POP_0014_MA_ZS	0.0001
64	SH_DYN_MORT	0.0001

Model with 80 variables and max depth None:

Training+Validation R^2: 0.99905, RMSE: 0.58664

Testing R^2: 0.98629, RMSE: 2.40644

Mean cross-validation score: 0.98487

	Feature	Importance
0	CPI_EST_avg	0.9287
79	CPI_EST_prev	0.0329
4	NY_GDP_PCAP_KD_rel	0.0013
30	SP_POP_0014_FE_ZS	0.0013
17	SE_SEC_ENRR_FE	0.0011
39	SH_DYN_NMRT	0.0011
49	SP_POP_7579_FE_5Y	0.0010
44	EG_USE_ELEC_KH_PC	0.0009
45	SP_POP_6569_FE_5Y	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
63	FS_AST_PRVT_GD_ZS	0.0008
74	SP_RUR_TOTL_ZS	0.0008
75	SP_URB_TOTL_IN_ZS	0.0008
12	SP_DYN_LE00_FE_IN	0.0008
54	SP_POP_5559_FE_5Y	0.0007
57	SP_POP_4549_MA_5Y	0.0007
19	SP_POP_80UP_FE_5Y	0.0007
51	NV_SRV_TOTL_ZS	0.0007
72	SP_POP_2024_FE_5Y	0.0007
42	SP_POP_7074_FE_5Y	0.0007
46	SP_DYN_TO65_FE_ZS	0.0007
18	SP_POP_7579_MA_5Y	0.0006
76	SH_DYN_1014	0.0006
50	SP_DYN_IMRT_FE_IN	0.0006
70	FM_AST_PRVT_GD_ZS	0.0006
52	SP_ADO_TFRT	0.0005
67	SP_POP_4549_FE_5Y	0.0005
33	SP_POP_65UP_FE_ZS	0.0005

32	SP_POP_1519_MA_5Y	0.0005
31	SP_POP_0509_FE_5Y	0.0005
66	SH_DYN_MORT_FE	0.0005
53	SP_POP_5054_FE_5Y	0.0005
26	IT_MLT_MAIN_P2	0.0005
20	SP_DYN_TO65_MA_ZS	0.0005
58	SE_PRE_ENRR	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
78	SG_LAW_INDX	0.0005
61	SP_DYN_TFRT_IN	0.0005
65	SE_TER_ENRR	0.0005
23	SP_POP_0014_TO_ZS	0.0005
55	SH_DYN_1519	0.0004
56	SP_POP_6064_FE_5Y	0.0004
59	NV_AGR_TOTL_ZS	0.0004
69	FD_AST_PRVT_GD_ZS	0.0004
62	SH_DYN_2024	0.0004
73	SE_PRM_ENRL_TC_ZS	0.0004
22	SE_SEC_ENRR_MA	0.0004
16	SP_POP_7074_MA_5Y	0.0004
3	NY_GNP_PCAP_CD	0.0004
37	SP_POP_5559_MA_5Y	0.0004
6	NY_GDP_PCAP_CD	0.0004
35	SP_POP_1519_FE_5Y	0.0004
7	IT_NET_USER_ZS	0.0004
77	EG_USE_PCAP_KG_OE	0.0004
21	SP_POP_1014_MA_5Y	0.0004
5	NY_GDP_PCAP_KD	0.0003
8	SP_DYN_LE00_MA_IN	0.0003
13	SE_SEC_ENRR	0.0003
43	SP_DYN_AMRT_MA	0.0003
71	TM_VAL_MRCH_HI_ZS	0.0003
27	SP_POP_0004_MA_5Y	0.0003
28	SP_POP_DPND_DL	0.0003
29	SP_POP_1014_FE_5Y	0.0003
36	SP_DYN_CBRT_IN	0.0003
38	SP_POP_0004_FE_5Y	0.0003
68	SP_DYN_AMRT_FE	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
40	SP_POP_5054_MA_5Y	0.0002
60	SH_DYN_MORT_MA	0.0002
48	SP_DYN_IMRT_IN	0.0002
47	SP_DYN_IMRT_MA_IN	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
34	SP_POP_6064_MA_5Y	0.0002
25	SP_POP_6569_MA_5Y	0.0002
24	SP_POP_65UP_TO_ZS	0.0002
15	SP_POP_0509_MA_5Y	0.0002

9	SP_DYN_LE00_IN	0.0002
64	SH_DYN_MORT	0.0001
14	SP_POP_0014_MA_ZS	0.0001
41	SP_POP_DPND_YG	0.0001

Model with 81 variables and max depth None:
 Training+Validation R^2: 0.99934, RMSE: 0.49024
 Testing R^2: 0.98675, RMSE: 2.36556
 Mean cross-validation score: 0.98459

	Feature	Importance
0	CPI_EST_avg	0.9069
80	CPI_EST_prev	0.0382
4	NY_GDP_PCAP_KD_rel	0.0025
5	NY_GDP_PCAP_KD	0.0025
12	SP_DYN_LE00_FE_IN	0.0017
19	SP_POP_80UP_FE_5Y	0.0014
44	EG_USE_ELEC_KH_PC	0.0014
63	FS_AST_PRVT_GD_ZS	0.0013
17	SE_SEC_ENRR_FE	0.0013
39	SH_DYN_NMRT	0.0013
38	SP_POP_0004_FE_5Y	0.0012
54	SP_POP_5559_FE_5Y	0.0011
57	SP_POP_4549_MA_5Y	0.0011
50	SP_DYN_IMRT_FE_IN	0.0011
22	SE_SEC_ENRR_MA	0.0010
46	SP_DYN_T065_FE_ZS	0.0010
74	SP_RUR_TOTL_ZS	0.0010
33	SP_POP_65UP_FE_ZS	0.0009
20	SP_DYN_T065_MA_ZS	0.0009
72	SP_POP_2024_FE_5Y	0.0009
59	NV_AGR_TOTL_ZS	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
55	SH_DYN_1519	0.0008
49	SP_POP_7579_FE_5Y	0.0008
45	SP_POP_6569_FE_5Y	0.0008
51	NV_SRV_TOTL_ZS	0.0008
76	SH_DYN_1014	0.0008
53	SP_POP_5054_FE_5Y	0.0008
10	SP_POP_80UP_MA_5Y	0.0008
78	SG_LAW_INDX	0.0008
56	SP_POP_6064_FE_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0007
42	SP_POP_7074_FE_5Y	0.0007
67	SP_POP_4549_FE_5Y	0.0007
65	SE_TER_ENRR	0.0007
61	SP_DYN_TFRT_IN	0.0007

32	SP_POP_1519_MA_5Y	0.0007
58	SE_PRE_ENRR	0.0007
77	EG_USE_PCAP_KG_OE	0.0007
43	SP_DYN_AMRT_MA	0.0006
6	NY_GDP_PCAP_CD	0.0006
7	IT_NET_USER_ZS	0.0006
70	FM_AST_PRVT_GD_ZS	0.0006
60	SH_DYN_MORT_MA	0.0006
35	SP_POP_1519_FE_5Y	0.0006
75	SP_URB_TOTL_IN_ZS	0.0006
28	SP_POP_DPND_DL	0.0006
73	SE_PRM_ENRL_TC_ZS	0.0006
68	SP_DYN_AMRT_FE	0.0005
52	SP_ADO_TFRT	0.0005
37	SP_POP_5559_MA_5Y	0.0005
71	TM_VAL_MRCH_HI_ZS	0.0005
79	SP_POP_2024_MA_5Y	0.0005
34	SP_POP_6064_MA_5Y	0.0005
31	SP_POP_0509_FE_5Y	0.0005
29	SP_POP_1014_FE_5Y	0.0005
26	IT_MLT_MAIN_P2	0.0005
9	SP_DYN_LEOO_IN	0.0005
13	SE_SEC_ENRR	0.0005
8	SP_DYN_LEOO_MA_IN	0.0004
69	FD_AST_PRVT_GD_ZS	0.0004
25	SP_POP_6569_MA_5Y	0.0004
21	SP_POP_1014_MA_5Y	0.0004
66	SH_DYN_MORT_FE	0.0004
48	SP_DYN_IMRT_IN	0.0003
62	SH_DYN_2024	0.0003
16	SP_POP_7074_MA_5Y	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
23	SP_POP_0014_TO_ZS	0.0003
30	SP_POP_0014_FE_ZS	0.0003
36	SP_DYN_CBRT_IN	0.0003
40	SP_POP_5054_MA_5Y	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0002
47	SP_DYN_IMRT_MA_IN	0.0002
3	NY_GNP_PCAP_CD	0.0002
27	SP_POP_0004_MA_5Y	0.0002
15	SP_POP_0509_MA_5Y	0.0002
64	SH_DYN_MORT	0.0001
24	SP_POP_65UP_TO_ZS	0.0001
14	SP_POP_0014_MA_ZS	0.0000
41	SP_POP_DPND_YG	0.0000

Model with 82 variables and max depth None:

Training+Validation R^2: 0.99885, RMSE: 0.64593
 Testing R^2: 0.98683, RMSE: 2.35829
 Mean cross-validation score: 0.98457

	Feature	Importance
0	CPI_EST_avg	0.9095
81	CPI_EST_prev	0.0371
4	NY_GDP_PCAP_KD_rel	0.0021
54	SP_POP_5559_FE_5Y	0.0017
20	SP_DYN_T065_MA_ZS	0.0015
39	SH_DYN_NMRT	0.0014
44	EG_USE_ELEC_KH_PC	0.0014
17	SE_SEC_ENRR_FE	0.0013
46	SP_DYN_T065_FE_ZS	0.0013
45	SP_POP_6569_FE_5Y	0.0013
12	SP_DYN_LE00_FE_IN	0.0012
63	FS_AST_PRVT_GD_ZS	0.0011
49	SP_POP_7579_FE_5Y	0.0011
72	SP_POP_2024_FE_5Y	0.0010
74	SP_RUR_TOTL_ZS	0.0010
22	SE_SEC_ENRR_MA	0.0009
36	SP_DYN_CBRT_IN	0.0009
33	SP_POP_65UP_FE_ZS	0.0009
42	SP_POP_7074_FE_5Y	0.0009
31	SP_POP_0509_FE_5Y	0.0009
59	NV_AGR_TOTL_ZS	0.0009
11	SP_POP_65UP_MA_ZS	0.0009
61	SP_DYN_TFRT_IN	0.0008
51	NV_SRV_TOTL_ZS	0.0008
32	SP_POP_1519_MA_5Y	0.0008
43	SP_DYN_AMRT_MA	0.0008
27	SP_POP_0004_MA_5Y	0.0008
65	SE_TER_ENRR	0.0008
76	SH_DYN_1014	0.0008
19	SP_POP_80UP_FE_5Y	0.0008
57	SP_POP_4549_MA_5Y	0.0007
70	FM_AST_PRVT_GD_ZS	0.0007
18	SP_POP_7579_MA_5Y	0.0007
38	SP_POP_0004_FE_5Y	0.0007
13	SE_SEC_ENRR	0.0007
64	SH_DYN_MORT	0.0007
58	SE_PRE_ENRR	0.0007
53	SP_POP_5054_FE_5Y	0.0006
80	NE_CON_PRVT_ZS	0.0006
7	IT_NET_USER_ZS	0.0006
55	SH_DYN_1519	0.0006
56	SP_POP_6064_FE_5Y	0.0006
40	SP_POP_5054_MA_5Y	0.0006

60	SH_DYN_MORT_MA	0.0006
62	SH_DYN_2024	0.0006
26	IT_MLT_MAIN_P2	0.0006
52	SP_ADO_TFRT	0.0006
73	SE_PRM_ENRL_TC_ZS	0.0005
71	TM_VAL_MRCH_HI_ZS	0.0005
75	SP_URB_TOTL_IN_ZS	0.0005
69	FD_AST_PRVT_GD_ZS	0.0005
67	SP_POP_4549_FE_5Y	0.0005
79	SP_POP_2024_MA_5Y	0.0005
41	SP_POP_DPND_YG	0.0005
23	SP_POP_0014_TO_ZS	0.0005
5	NY_GDP_PCAP_KD	0.0005
3	NY_GNP_PCAP_CD	0.0005
30	SP_POP_0014_FE_ZS	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
68	SP_DYN_AMRT_FE	0.0004
6	NY_GDP_PCAP_CD	0.0004
37	SP_POP_5559_MA_5Y	0.0004
77	EG_USE_PCAP_KG_OE	0.0004
34	SP_POP_6064_MA_5Y	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
28	SP_POP_DPND_DL	0.0004
21	SP_POP_1014_MA_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0003
66	SH_DYN_MORT_FE	0.0003
29	SP_POP_1014_FE_5Y	0.0003
35	SP_POP_1519_FE_5Y	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
24	SP_POP_65UP_TO_ZS	0.0002
15	SP_POP_0509_MA_5Y	0.0002
9	SP_DYN_LE00_IN	0.0002
78	SG_LAW_INDX	0.0002
48	SP_DYN_IMRT_IN	0.0001
14	SP_POP_0014_MA_ZS	0.0001
25	SP_POP_6569_MA_5Y	0.0001
50	SP_DYN_IMRT_FE_IN	0.0001
47	SP_DYN_IMRT_MA_IN	0.0000

Model with 83 variables and max depth None:
 Training+Validation R^2: 0.99801, RMSE: 0.84929
 Testing R^2: 0.98631, RMSE: 2.40511
 Mean cross-validation score: 0.98454

	Feature	Importance
0	CPI_EST_avg	0.9213

82	CPI_EST_prev	0.0338
4	NY_GDP_PCAP_KD_rel	0.0022
46	SP_DYN_T065_FE_ZS	0.0014
12	SP_DYN_LE00_FE_IN	0.0013
17	SE_SEC_ENRR_FE	0.0013
39	SH_DYN_NMRT	0.0012
74	SP_RUR_TOTL_ZS	0.0011
44	EG_USE_ELEC_KH_PC	0.0011
19	SP_POP_80UP_FE_5Y	0.0011
54	SP_POP_5559_FE_5Y	0.0010
49	SP_POP_7579_FE_5Y	0.0010
63	FS_AST_PRVT_GD_ZS	0.0010
64	SH_DYN_MORT	0.0008
38	SP_POP_0004_FE_5Y	0.0008
65	SE_TER_ENRR	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
75	SP_URB_TOTL_IN_ZS	0.0007
45	SP_POP_6569_FE_5Y	0.0007
13	SE_SEC_ENRR	0.0007
20	SP_DYN_T065_MA_ZS	0.0007
43	SP_DYN_AMRT_MA	0.0007
72	SP_POP_2024_FE_5Y	0.0007
76	SH_DYN_1014	0.0007
51	NV_SRV_TOTL_ZS	0.0007
42	SP_POP_7074_FE_5Y	0.0007
31	SP_POP_0509_FE_5Y	0.0006
59	NV_AGR_TOTL_ZS	0.0006
58	SE_PRE_ENRR	0.0006
55	SH_DYN_1519	0.0006
22	SE_SEC_ENRR_MA	0.0006
80	NE_CON_PRVT_ZS	0.0006
70	FM_AST_PRVT_GD_ZS	0.0006
18	SP_POP_7579_MA_5Y	0.0006
56	SP_POP_6064_FE_5Y	0.0005
5	NY_GDP_PCAP_KD	0.0005
53	SP_POP_5054_FE_5Y	0.0005
7	IT_NET_USER_ZS	0.0005
79	SP_POP_2024_MA_5Y	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
61	SP_DYN_TFRT_IN	0.0005
21	SP_POP_1014_MA_5Y	0.0005
30	SP_POP_0014_FE_ZS	0.0005
40	SP_POP_5054_MA_5Y	0.0005
26	IT_MLT_MAIN_P2	0.0005
27	SP_POP_0004_MA_5Y	0.0005
36	SP_DYN_CBRT_IN	0.0005
33	SP_POP_65UP_FE_ZS	0.0005
32	SP_POP_1519_MA_5Y	0.0005

62	SH_DYN_2024	0.0004
71	TM_VAL_MRCH_HI_ZS	0.0004
73	SE_PRM_ENRL_TC_ZS	0.0004
69	FD_AST_PRVT_GD_ZS	0.0004
67	SP_POP_4549_FE_5Y	0.0004
57	SP_POP_4549_MA_5Y	0.0004
68	SP_DYN_AMRT_FE	0.0004
41	SP_POP_DPND_YG	0.0004
52	SP_ADO_TFRT	0.0004
3	NY_GNP_PCAP_CD	0.0004
28	SP_POP_DPND_DL	0.0004
34	SP_POP_6064_MA_5Y	0.0004
29	SP_POP_1014_FE_5Y	0.0003
81	SH_DYN_0509	0.0003
6	NY_GDP_PCAP_CD	0.0003
8	SP_DYN_LEOO_MA_IN	0.0003
77	EG_USE_PCAP_KG_OE	0.0003
16	SP_POP_7074_MA_5Y	0.0003
23	SP_POP_0014_TO_ZS	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0003
37	SP_POP_5559_MA_5Y	0.0003
60	SH_DYN_MORT_MA	0.0003
66	SH_DYN_MORT_FE	0.0002
24	SP_POP_65UP_TO_ZS	0.0002
35	SP_POP_1519_FE_5Y	0.0002
15	SP_POP_0509_MA_5Y	0.0002
9	SP_DYN_LEOO_IN	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
78	SG_LAW_INDX	0.0002
25	SP_POP_6569_MA_5Y	0.0002
50	SP_DYN_IMRT_FE_IN	0.0001
14	SP_POP_0014_MA_ZS	0.0001
47	SP_DYN_IMRT_MA_IN	0.0001
48	SP_DYN_IMRT_IN	0.0001

Model with 84 variables and max depth None:

Training+Validation R^2: 0.99973, RMSE: 0.313

Testing R^2: 0.9866, RMSE: 2.37915

Mean cross-validation score: 0.98487

	Feature	Importance
0	CPI_EST_avg	0.9238
83	CPI_EST_prev	0.0293
39	SH_DYN_NMRT	0.0016
75	SP_URB_TOTL_IN_ZS	0.0015
4	NY_GDP_PCAP_KD_rel	0.0013
17	SE_SEC_ENRR_FE	0.0013

44	EG_USE_ELEC_KH_PC	0.0012
12	SP_DYN_LE00_FE_IN	0.0010
19	SP_POP_80UP_FE_5Y	0.0010
54	SP_POP_5559_FE_5Y	0.0010
63	FS_AST_PRVT_GD_ZS	0.0010
74	SP_RUR_TOTL_ZS	0.0009
49	SP_POP_7579_FE_5Y	0.0009
46	SP_DYN_T065_FE_ZS	0.0009
11	SP_POP_65UP_MA_ZS	0.0008
27	SP_POP_0004_MA_5Y	0.0008
51	NV_SRV_TOTL_ZS	0.0008
45	SP_POP_6569_FE_5Y	0.0008
76	SH_DYN_1014	0.0008
42	SP_POP_7074_FE_5Y	0.0008
65	SE_TER_ENRR	0.0007
61	SP_DYN_TFRT_IN	0.0007
60	SH_DYN_MORT_MA	0.0007
70	FM_AST_PRVT_GD_ZS	0.0007
72	SP_POP_2024_FE_5Y	0.0007
43	SP_DYN_AMRT_MA	0.0007
22	SE_SEC_ENRR_MA	0.0007
57	SP_POP_4549_MA_5Y	0.0006
82	EG_ELC_LOSS_ZS	0.0006
18	SP_POP_7579_MA_5Y	0.0006
21	SP_POP_1014_MA_5Y	0.0006
31	SP_POP_0509_FE_5Y	0.0006
32	SP_POP_1519_MA_5Y	0.0006
52	SP_ADO_TFRT	0.0006
69	FD_AST_PRVT_GD_ZS	0.0006
36	SP_DYN_CBRT_IN	0.0006
20	SP_DYN_T065_MA_ZS	0.0006
38	SP_POP_0004_FE_5Y	0.0006
59	NV_AGR_TOTL_ZS	0.0006
58	SE_PRE_ENRR	0.0006
26	IT_MLT_MAIN_P2	0.0005
53	SP_POP_5054_FE_5Y	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
13	SE_SEC_ENRR	0.0005
7	IT_NET_USER_ZS	0.0005
5	NY_GDP_PCAP_KD	0.0005
79	SP_POP_2024_MA_5Y	0.0005
80	NE_CON_PRVT_ZS	0.0005
50	SP_DYN_IMRT_FE_IN	0.0005
55	SH_DYN_1519	0.0005
37	SP_POP_5559_MA_5Y	0.0005
40	SP_POP_5054_MA_5Y	0.0005
35	SP_POP_1519_FE_5Y	0.0005
29	SP_POP_1014_FE_5Y	0.0005

3	NY_GNP_PCAP_CD	0.0005
67	SP_POP_4549_FE_5Y	0.0004
56	SP_POP_6064_FE_5Y	0.0004
30	SP_POP_0014_FE_ZS	0.0004
41	SP_POP_DPND_YG	0.0004
77	EG_USE_PCAP_KG_OE	0.0004
62	SH_DYN_2024	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
73	SE_PRM_ENRL_TC_ZS	0.0004
71	TM_VAL_MRCH_HI_ZS	0.0004
9	SP_DYN_LE00_IN	0.0003
78	SG_LAW_INDX	0.0003
33	SP_POP_65UP_FE_ZS	0.0003
6	NY_GDP_PCAP_CD	0.0003
8	SP_DYN_LE00_MA_IN	0.0003
16	SP_POP_7074_MA_5Y	0.0003
68	SP_DYN_AMRT_FE	0.0003
25	SP_POP_6569_MA_5Y	0.0003
66	SH_DYN_MORT_FE	0.0003
15	SP_POP_0509_MA_5Y	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
23	SP_POP_0014_TO_ZS	0.0002
34	SP_POP_6064_MA_5Y	0.0002
28	SP_POP_DPND_DL	0.0002
14	SP_POP_0014_MA_ZS	0.0002
47	SP_DYN_IMRT_MA_IN	0.0002
81	SH_DYN_0509	0.0001
48	SP_DYN_IMRT_IN	0.0001
64	SH_DYN_MORT	0.0001
2	NY_ADJ_NNTY_PC_CD	0.0001

Model with 85 variables and max depth None:

Training+Validation R^2: 0.99961, RMSE: 0.37468

Testing R^2: 0.98647, RMSE: 2.39055

Mean cross-validation score: 0.98499

	Feature	Importance
0	CPI_EST_avg	0.9241
84	CPI_EST_prev	0.0312
4	NY_GDP_PCAP_KD_rel	0.0015
39	SH_DYN_NMRT	0.0013
12	SP_DYN_LE00_FE_IN	0.0013
11	SP_POP_65UP_MA_ZS	0.0010
44	EG_USE_ELEC_KH_PC	0.0010
46	SP_DYN_T065_FE_ZS	0.0010
76	SH_DYN_1014	0.0010
17	SE_SEC_ENRR_FE	0.0010

45	SP_POP_6569_FE_5Y	0.0009
74	SP_RUR_TOTL_ZS	0.0009
54	SP_POP_5559_FE_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0008
63	FS_AST_PRVT_GD_ZS	0.0008
55	SH_DYN_1519	0.0008
48	SP_DYN_IMRT_IN	0.0008
49	SP_POP_7579_FE_5Y	0.0008
51	NV_SRV_TOTL_ZS	0.0007
31	SP_POP_0509_FE_5Y	0.0007
27	SP_POP_0004_MA_5Y	0.0007
42	SP_POP_7074_FE_5Y	0.0007
19	SP_POP_80UP_FE_5Y	0.0007
70	FM_AST_PRVT_GD_ZS	0.0007
5	NY_GDP_PCAP_KD	0.0007
72	SP_POP_2024_FE_5Y	0.0007
32	SP_POP_1519_MA_5Y	0.0006
40	SP_POP_5054_MA_5Y	0.0006
30	SP_POP_0014_FE_ZS	0.0006
22	SE_SEC_ENRR_MA	0.0006
57	SP_POP_4549_MA_5Y	0.0006
59	NV_AGR_TOTL_ZS	0.0006
65	SE_TER_ENRR	0.0006
20	SP_DYN_T065_MA_ZS	0.0006
79	SP_POP_2024_MA_5Y	0.0006
80	NE_CON_PRVT_ZS	0.0006
41	SP_POP_DPND_YG	0.0006
58	SE_PRE_ENRR	0.0005
52	SP_ADO_TFRT	0.0005
75	SP_URB_TOTL_IN_ZS	0.0005
82	EG_ELC_LOSS_ZS	0.0005
83	SP_POP_1564_MA_ZS	0.0005
43	SP_DYN_AMRT_MA	0.0005
38	SP_POP_0004_FE_5Y	0.0005
6	NY_GDP_PCAP_CD	0.0005
13	SE_SEC_ENRR	0.0005
21	SP_POP_1014_MA_5Y	0.0005
33	SP_POP_65UP_FE_ZS	0.0005
23	SP_POP_0014_TO_ZS	0.0005
7	IT_NET_USER_ZS	0.0005
26	IT_MLT_MAIN_P2	0.0005
60	SH_DYN_MORT_MA	0.0004
56	SP_POP_6064_FE_5Y	0.0004
61	SP_DYN_TFRT_IN	0.0004
16	SP_POP_7074_MA_5Y	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
73	SE_PRM_ENRL_TC_ZS	0.0004
68	SP_DYN_AMRT_FE	0.0004

29	SP_POP_1014_FE_5Y	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004
36	SP_DYN_CBRT_IN	0.0004
10	SP_POP_80UP_MA_5Y	0.0004
62	SH_DYN_2024	0.0004
15	SP_POP_0509_MA_5Y	0.0003
71	TM_VAL_MRCH_HI_ZS	0.0003
8	SP_DYN_LE00_MA_IN	0.0003
69	FD_AST_PRVT_GD_ZS	0.0003
3	NY_GNP_PCAP_CD	0.0003
67	SP_POP_4549_FE_5Y	0.0003
66	SH_DYN_MORT_FE	0.0003
25	SP_POP_6569_MA_5Y	0.0003
28	SP_POP_DPND_DL	0.0003
53	SP_POP_5054_FE_5Y	0.0003
34	SP_POP_6064_MA_5Y	0.0003
35	SP_POP_1519_FE_5Y	0.0003
37	SP_POP_5559_MA_5Y	0.0003
9	SP_DYN_LE00_IN	0.0003
78	SG_LAW_INDX	0.0002
64	SH_DYN_MORT	0.0002
77	EG_USE_PCAP_KG_OE	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
47	SP_DYN_IMRT_MA_IN	0.0002
14	SP_POP_0014_MA_ZS	0.0001
81	SH_DYN_0509	0.0001
2	NY_ADJ_NNTY_PC_CD	0.0001

Model with 86 variables and max depth None:

Training+Validation R^2: 0.99962, RMSE: 0.37195

Testing R^2: 0.98651, RMSE: 2.38717

Mean cross-validation score: 0.98484

	Feature	Importance
0	CPI_EST_avg	0.9146
85	CPI_EST_prev	0.0352
46	SP_DYN_T065_FE_ZS	0.0021
39	SH_DYN_NMRT	0.0016
4	NY_GDP_PCAP_KD_rel	0.0015
45	SP_POP_6569_FE_5Y	0.0013
17	SE_SEC_ENRR_FE	0.0012
54	SP_POP_5559_FE_5Y	0.0012
44	EG_USE_ELEC_KH_PC	0.0011
42	SP_POP_7074_FE_5Y	0.0011
19	SP_POP_80UP_FE_5Y	0.0011
31	SP_POP_0509_FE_5Y	0.0010
12	SP_DYN_LE00_FE_IN	0.0010

74	SP_RUR_TOTL_ZS	0.0010
20	SP_DYN_TO65_MA_ZS	0.0010
41	SP_POP_DPND_YG	0.0009
72	SP_POP_2024_FE_5Y	0.0009
76	SH_DYN_1014	0.0008
75	SP_URB_TOTL_IN_ZS	0.0008
82	EG_ELC_LOSS_ZS	0.0008
49	SP_POP_7579_FE_5Y	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
63	FS_AST_PRVT_GD_ZS	0.0008
52	SP_ADO_TFRT	0.0008
51	NV_SRV_TOTL_ZS	0.0008
27	SP_POP_0004_MA_5Y	0.0007
57	SP_POP_4549_MA_5Y	0.0007
22	SE_SEC_ENRR_MA	0.0007
13	SE_SEC_ENRR	0.0006
61	SP_DYN_TFRT_IN	0.0006
40	SP_POP_5054_MA_5Y	0.0006
55	SH_DYN_1519	0.0006
38	SP_POP_0004_FE_5Y	0.0006
65	SE_TER_ENRR	0.0006
80	NE_CON_PRVT_ZS	0.0006
26	IT_MLT_MAIN_P2	0.0006
70	FM_AST_PRVT_GD_ZS	0.0006
59	NV_AGR_TOTL_ZS	0.0006
53	SP_POP_5054_FE_5Y	0.0006
23	SP_POP_0014_TO_ZS	0.0006
47	SP_DYN_IMRT_MA_IN	0.0005
7	IT_NET_USER_ZS	0.0005
84	EN_ATM_CO2E_PC	0.0005
58	SE_PRE_ENRR	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
18	SP_POP_7579_MA_5Y	0.0005
32	SP_POP_1519_MA_5Y	0.0005
21	SP_POP_1014_MA_5Y	0.0005
78	SG_LAW_INDX	0.0005
77	EG_USE_PCAP_KG_OE	0.0005
62	SH_DYN_2024	0.0004
79	SP_POP_2024_MA_5Y	0.0004
67	SP_POP_4549_FE_5Y	0.0004
69	FD_AST_PRVT_GD_ZS	0.0004
71	TM_VAL_MRCH_HI_ZS	0.0004
73	SE_PRM_ENRL_TC_ZS	0.0004
64	SH_DYN_MORT	0.0004
43	SP_DYN_AMRT_MA	0.0004
29	SP_POP_1014_FE_5Y	0.0004
28	SP_POP_DPND_OL	0.0004
5	NY_GDP_PCAP_KD	0.0004

3	NY_GNP_PCAP_CD	0.0004
36	SP_DYN_CBRT_IN	0.0004
35	SP_POP_1519_FE_5Y	0.0004
16	SP_POP_7074_MA_5Y	0.0004
33	SP_POP_65UP_FE_ZS	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
25	SP_POP_6569_MA_5Y	0.0004
83	SP_POP_1564_MA_ZS	0.0003
6	NY_GDP_PCAP_CD	0.0003
8	SP_DYN_LE00_MA_IN	0.0003
9	SP_DYN_LE00_IN	0.0003
15	SP_POP_0509_MA_5Y	0.0003
56	SP_POP_6064_FE_5Y	0.0003
50	SP_DYN_IMRT_FE_IN	0.0003
34	SP_POP_6064_MA_5Y	0.0003
68	SP_DYN_AMRT_FE	0.0003
66	SH_DYN_MORT_FE	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
37	SP_POP_5559_MA_5Y	0.0002
81	SH_DYN_0509	0.0002
60	SH_DYN_MORT_MA	0.0002
14	SP_POP_0014_MA_ZS	0.0001
48	SP_DYN_IMRT_IN	0.0001
2	NY_ADJ_NNTY_PC_CD	0.0001
30	SP_POP_0014_FE_ZS	0.0000

Model with 87 variables and max depth None:

Training+Validation R^2: 0.99973, RMSE: 0.31322

Testing R^2: 0.98606, RMSE: 2.42674

Mean cross-validation score: 0.98465

	Feature	Importance
0	CPI_EST_avg	0.9245
86	CPI_EST_prev	0.0308
5	NY_GDP_PCAP_KD	0.0016
19	SP_POP_80UP_FE_5Y	0.0015
39	SH_DYN_NMRT	0.0013
4	NY_GDP_PCAP_KD_rel	0.0013
31	SP_POP_0509_FE_5Y	0.0013
44	EG_USE_ELEC_KH_PC	0.0012
46	SP_DYN_TO65_FE_ZS	0.0011
17	SE_SEC_ENRR_FE	0.0011
45	SP_POP_6569_FE_5Y	0.0010
12	SP_DYN_LE00_FE_IN	0.0010
76	SH_DYN_1014	0.0009
54	SP_POP_5559_FE_5Y	0.0009
74	SP_RUR_TOTL_ZS	0.0008

61	SP_DYN_TFRT_IN	0.0008
63	FS_AST_PRVT_GD_ZS	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
40	SP_POP_5054_MA_5Y	0.0007
51	NV_SRV_TOTL_ZS	0.0007
20	SP_DYN_T065_MA_ZS	0.0007
59	NV_AGR_TOTL_ZS	0.0007
72	SP_POP_2024_FE_5Y	0.0007
65	SE_TER_ENRR	0.0006
49	SP_POP_7579_FE_5Y	0.0006
69	FD_AST_PRVT_GD_ZS	0.0006
18	SP_POP_7579_MA_5Y	0.0006
42	SP_POP_7074_FE_5Y	0.0006
75	SP_URB_TOTL_IN_ZS	0.0006
84	EN_ATM_CO2E_PC	0.0006
47	SP_DYN_IMRT_MA_IN	0.0005
36	SP_DYN_CBRT_IN	0.0005
57	SP_POP_4549_MA_5Y	0.0005
58	SE_PRE_ENRR	0.0005
26	IT_MLT_MAIN_P2	0.0005
30	SP_POP_0014_FE_ZS	0.0005
67	SP_POP_4549_FE_5Y	0.0005
22	SE_SEC_ENRR_MA	0.0005
16	SP_POP_7074_MA_5Y	0.0005
15	SP_POP_0509_MA_5Y	0.0005
13	SE_SEC_ENRR	0.0005
80	NE_CON_PRVT_ZS	0.0005
7	IT_NET_USER_ZS	0.0005
73	SE_PRM_ENRL_TC_ZS	0.0004
56	SP_POP_6064_FE_5Y	0.0004
79	SP_POP_2024_MA_5Y	0.0004
55	SH_DYN_1519	0.0004
82	EG_ELC_LOSS_ZS	0.0004
53	SP_POP_5054_FE_5Y	0.0004
85	SP_POP_4044_FE_5Y	0.0004
52	SP_ADO_TFRT	0.0004
43	SP_DYN_AMRT_MA	0.0004
32	SP_POP_1519_MA_5Y	0.0004
21	SP_POP_1014_MA_5Y	0.0004
6	NY_GDP_PCAP_CD	0.0004
10	SP_POP_80UP_MA_5Y	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
35	SP_POP_1519_FE_5Y	0.0004
27	SP_POP_0004_MA_5Y	0.0004
34	SP_POP_6064_MA_5Y	0.0004
33	SP_POP_65UP_FE_ZS	0.0004
29	SP_POP_1014_FE_5Y	0.0004
83	SP_POP_1564_MA_ZS	0.0003

77	EG_USE_PCAP_KG_OE	0.0003
71	TM_VAL_MRCH_HI_ZS	0.0003
70	FM_AST_PRVT_GD_ZS	0.0003
28	SP_POP_DPND_OL	0.0003
8	SP_DYN_LE00_MA_IN	0.0003
62	SH_DYN_2024	0.0003
38	SP_POP_0004_FE_5Y	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
3	NY_GNP_PCAP_CD	0.0003
78	SG_LAW_INDX	0.0003
68	SP_DYN_AMRT_FE	0.0002
9	SP_DYN_LE00_IN	0.0002
60	SH_DYN_MORT_MA	0.0002
81	SH_DYN_0509	0.0002
37	SP_POP_5559_MA_5Y	0.0002
14	SP_POP_0014_MA_ZS	0.0001
50	SP_DYN_IMRT_FE_IN	0.0001
23	SP_POP_0014_TO_ZS	0.0001
25	SP_POP_6569_MA_5Y	0.0001
48	SP_DYN_IMRT_IN	0.0001
66	SH_DYN_MORT_FE	0.0001
64	SH_DYN_MORT	0.0001
41	SP_POP_DPND_YG	0.0001
2	NY_ADJ_NNTY_PC_CD	0.0001

Model with 88 variables and max depth None:

Training+Validation R^2: 0.99976, RMSE: 0.29225

Testing R^2: 0.98621, RMSE: 2.41347

Mean cross-validation score: 0.98454

	Feature	Importance
0	CPI_EST_avg	0.9192
87	CPI_EST_prev	0.0330
4	NY_GDP_PCAP_KD_rel	0.0016
44	EG_USE_ELEC_KH_PC	0.0013
46	SP_DYN_T065_FE_ZS	0.0013
17	SE_SEC_ENRR_FE	0.0013
39	SH_DYN_NMRT	0.0011
81	SH_DYN_0509	0.0011
19	SP_POP_80UP_FE_5Y	0.0009
55	SH_DYN_1519	0.0009
12	SP_DYN_LE00_FE_IN	0.0009
74	SP_RUR_TOTL_ZS	0.0009
63	FS_AST_PRVT_GD_ZS	0.0008
70	FM_AST_PRVT_GD_ZS	0.0008
72	SP_POP_2024_FE_5Y	0.0008
54	SP_POP_5559_FE_5Y	0.0008

51	NV_SRV_TOTL_ZS	0.0008
49	SP_POP_7579_FE_5Y	0.0008
75	SP_URB_TOTL_IN_ZS	0.0008
84	EN_ATM_CO2E_PC	0.0008
42	SP_POP_7074_FE_5Y	0.0008
40	SP_POP_5054_MA_5Y	0.0008
45	SP_POP_6569_FE_5Y	0.0008
59	NV_AGR_TOTL_ZS	0.0007
57	SP_POP_4549_MA_5Y	0.0007
11	SP_POP_65UP_MA_ZS	0.0007
65	SE_TER_ENRR	0.0007
48	SP_DYN_IMRT_IN	0.0007
22	SE_SEC_ENRR_MA	0.0006
58	SE_PRE_ENRR	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
18	SP_POP_7579_MA_5Y	0.0006
7	IT_NET_USER_ZS	0.0006
20	SP_DYN_TO65_MA_ZS	0.0006
41	SP_POP_DPND_YG	0.0006
26	IT_MLT_MAIN_P2	0.0006
79	SP_POP_2024_MA_5Y	0.0006
32	SP_POP_1519_MA_5Y	0.0006
35	SP_POP_1519_FE_5Y	0.0005
31	SP_POP_0509_FE_5Y	0.0005
56	SP_POP_6064_FE_5Y	0.0005
62	SH_DYN_2024	0.0005
52	SP_ADO_TFRT	0.0005
68	SP_DYN_AMRT_FE	0.0005
69	FD_AST_PRVT_GD_ZS	0.0005
36	SP_DYN_CBRT_IN	0.0005
80	NE_CON_PRVT_ZS	0.0005
43	SP_DYN_AMRT_MA	0.0005
61	SP_DYN_TFRT_IN	0.0005
76	SH_DYN_1014	0.0005
38	SP_POP_0004_FE_5Y	0.0005
13	SE_SEC_ENRR	0.0005
8	SP_DYN_LE00_MA_IN	0.0004
6	NY_GDP_PCAP_CD	0.0004
67	SP_POP_4549_FE_5Y	0.0004
5	NY_GDP_PCAP_KD	0.0004
85	SP_POP_4044_FE_5Y	0.0004
78	SG_LAW_INDX	0.0004
77	EG_USE_PCAP_KG_OE	0.0004
86	SH_IMM_IDPT	0.0004
23	SP_POP_0014_TO_ZS	0.0004
27	SP_POP_0004_MA_5Y	0.0004
53	SP_POP_5054_FE_5Y	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004

29	SP_POP_1014_FE_5Y	0.0004
33	SP_POP_65UP_FE_ZS	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
60	SH_DYN_MORT_MA	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
25	SP_POP_6569_MA_5Y	0.0003
28	SP_POP_DPND_DL	0.0003
83	SP_POP_1564_MA_ZS	0.0003
82	EG_ELC_LOSS_ZS	0.0003
37	SP_POP_5559_MA_5Y	0.0003
21	SP_POP_1014_MA_5Y	0.0003
3	NY_GNP_PCAP_CD	0.0003
73	SE_PRM_ENRL_TC_ZS	0.0003
9	SP_DYN_LEOO_IN	0.0003
71	TM_VAL_MRCH_HI_ZS	0.0003
34	SP_POP_6064_MA_5Y	0.0002
30	SP_POP_0014_FE_ZS	0.0002
47	SP_DYN_IMRT_MA_IN	0.0002
16	SP_POP_7074_MA_5Y	0.0002
15	SP_POP_0509_MA_5Y	0.0002
66	SH_DYN_MORT_FE	0.0001
14	SP_POP_0014_MA_ZS	0.0001
2	NY_ADJ_NNTY_PC_CD	0.0001
64	SH_DYN_MORT	0.0000

Model with 89 variables and max depth None:

Training+Validation R^2: 0.99887, RMSE: 0.63991

Testing R^2: 0.98681, RMSE: 2.36038

Mean cross-validation score: 0.98456

	Feature	Importance
0	CPI_EST_avg	0.9058
88	CPI_EST_prev	0.0364
12	SP_DYN_LEOO_FE_IN	0.0019
44	EG_USE_ELEC_KH_PC	0.0016
4	NY_GDP_PCAP_KD_rel	0.0016
47	SP_DYN_IMRT_MA_IN	0.0015
17	SE_SEC_ENRR_FE	0.0014
87	IT_CEL_SETS_P2	0.0012
75	SP_URB_TOTL_IN_ZS	0.0012
46	SP_DYN_TO65_FE_ZS	0.0012
11	SP_POP_65UP_MA_ZS	0.0012
19	SP_POP_80UP_FE_5Y	0.0012
63	FS_AST_PRVT_GD_ZS	0.0011
51	NV_SRV_TOTL_ZS	0.0011
22	SE_SEC_ENRR_MA	0.0011
45	SP_POP_6569_FE_5Y	0.0011

81	SH_DYN_0509	0.0010
39	SH_DYN_NMRT	0.0010
41	SP_POP_DPNP_YG	0.0010
54	SP_POP_5559_FE_5Y	0.0009
72	SP_POP_2024_FE_5Y	0.0009
84	EN_ATM_CO2E_PC	0.0009
59	NV_AGR_TOTL_ZS	0.0008
65	SE_TER_ENRR	0.0008
49	SP_POP_7579_FE_5Y	0.0008
27	SP_POP_0004_MA_5Y	0.0008
79	SP_POP_2024_MA_5Y	0.0008
83	SP_POP_1564_MA_ZS	0.0008
21	SP_POP_1014_MA_5Y	0.0008
55	SH_DYN_1519	0.0008
5	NY_GDP_PCAP_KD	0.0008
62	SH_DYN_2024	0.0007
53	SP_POP_5054_FE_5Y	0.0007
10	SP_POP_80UP_MA_5Y	0.0007
74	SP_RUR_TOTL_ZS	0.0007
76	SH_DYN_1014	0.0007
52	SP_ADO_TFRT	0.0007
36	SP_DYN_CBRT_IN	0.0007
31	SP_POP_0509_FE_5Y	0.0007
20	SP_DYN_T065_MA_ZS	0.0007
23	SP_POP_0014_TO_ZS	0.0007
35	SP_POP_1519_FE_5Y	0.0007
57	SP_POP_4549_MA_5Y	0.0007
34	SP_POP_6064_MA_5Y	0.0007
43	SP_DYN_AMRT_MA	0.0006
30	SP_POP_0014_FE_ZS	0.0006
32	SP_POP_1519_MA_5Y	0.0006
80	NE_CON_PRVT_ZS	0.0006
28	SP_POP_DPNP_OL	0.0006
7	IT_NET_USER_ZS	0.0006
33	SP_POP_65UP_FE_ZS	0.0006
85	SP_POP_4044_FE_5Y	0.0006
56	SP_POP_6064_FE_5Y	0.0006
67	SP_POP_4549_FE_5Y	0.0006
71	TM_VAL_MRCH_HI_ZS	0.0005
70	FM_AST_PRVT_GD_ZS	0.0005
61	SP_DYN_TFRT_IN	0.0005
86	SH_IMM_IDPT	0.0005
16	SP_POP_7074_MA_5Y	0.0005
58	SE_PRE_ENRR	0.0005
26	IT_MLT_MAIN_P2	0.0005
42	SP_POP_7074_FE_5Y	0.0005
69	FD_AST_PRVT_GD_ZS	0.0004
18	SP_POP_7579_MA_5Y	0.0004

14	SP_POP_0014_MA_ZS	0.0004
82	EG_ELC_LOSS_ZS	0.0004
29	SP_POP_1014_FE_5Y	0.0004
78	SG_LAW_INDX	0.0004
40	SP_POP_5054_MA_5Y	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
9	SP_DYN_LE00_IN	0.0004
68	SP_DYN_AMRT_FE	0.0003
77	EG_USE_PCAP_KG_OE	0.0003
13	SE_SEC_ENRR	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
25	SP_POP_6569_MA_5Y	0.0003
3	NY_GNP_PCAP_CD	0.0003
37	SP_POP_5559_MA_5Y	0.0003
6	NY_GDP_PCAP_CD	0.0003
73	SE_PRM_ENRL_TC_ZS	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
50	SP_DYN_IMRT_FE_IN	0.0002
15	SP_POP_0509_MA_5Y	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
38	SP_POP_0004_FE_5Y	0.0001
66	SH_DYN_MORT_FE	0.0000
48	SP_DYN_IMRT_IN	0.0000
64	SH_DYN_MORT	0.0000
60	SH_DYN_MORT_MA	0.0000

Model with 90 variables and max depth None:

Training+Validation R^2: 0.99673, RMSE: 1.08863

Testing R^2: 0.98573, RMSE: 2.45496

Mean cross-validation score: 0.98464

	Feature	Importance
0	CPI_EST_avg	0.9271
89	CPI_EST_prev	0.0278
46	SP_DYN_TO65_FE_ZS	0.0014
4	NY_GDP_PCAP_KD_rel	0.0014
17	SE_SEC_ENRR_FE	0.0012
39	SH_DYN_NMRT	0.0012
44	EG_USE_ELEC_KH_PC	0.0012
56	SP_POP_6064_FE_5Y	0.0011
11	SP_POP_65UP_MA_ZS	0.0011
12	SP_DYN_LE00_FE_IN	0.0011
54	SP_POP_5559_FE_5Y	0.0009
76	SH_DYN_1014	0.0009
75	SP.URB.TOTL_IN_ZS	0.0008
88	SP_POP_4044_MA_5Y	0.0008
41	SP_POP_DPND_YG	0.0008

87	IT_CEL_SETS_P2	0.0008
51	NV_SRV_TOTL_ZS	0.0008
38	SP_POP_0004_FE_5Y	0.0007
84	EN_ATM_CO2E_PC	0.0007
63	FS_AST_PRVT_GD_ZS	0.0007
19	SP_POP_80UP_FE_5Y	0.0007
35	SP_POP_1519_FE_5Y	0.0006
22	SE_SEC_ENRR_MA	0.0006
49	SP_POP_7579_FE_5Y	0.0006
18	SP_POP_7579_MA_5Y	0.0006
65	SE_TER_ENRR	0.0006
42	SP_POP_7074_FE_5Y	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
52	SP_ADO_TFRT	0.0006
53	SP_POP_5054_FE_5Y	0.0006
83	SP_POP_1564_MA_ZS	0.0006
32	SP_POP_1519_MA_5Y	0.0006
55	SH_DYN_1519	0.0006
72	SP_POP_2024_FE_5Y	0.0006
50	SP_DYN_IMRT_FE_IN	0.0005
86	SH_IMM_IDPT	0.0005
57	SP_POP_4549_MA_5Y	0.0005
58	SE_PRE_ENRR	0.0005
59	NV_AGR_TOTL_ZS	0.0005
61	SP_DYN_TFRT_IN	0.0005
69	FD_AST_PRVT_GD_ZS	0.0005
78	SG_LAW_INDX	0.0005
79	SP_POP_2024_MA_5Y	0.0005
80	NE_CON_PRVT_ZS	0.0005
71	TM_VAL_MRCH_HI_ZS	0.0005
45	SP_POP_6569_FE_5Y	0.0005
20	SP_DYN_T065_MA_ZS	0.0005
5	NY_GDP_PCAP_KD	0.0005
43	SP_DYN_AMRT_MA	0.0005
23	SP_POP_0014_TO_ZS	0.0005
26	IT_MLT_MAIN_P2	0.0005
29	SP_POP_1014_FE_5Y	0.0005
25	SP_POP_6569_MA_5Y	0.0004
27	SP_POP_0004_MA_5Y	0.0004
77	EG_USE_PCAP_KG_OE	0.0004
74	SP_RUR_TOTL_ZS	0.0004
30	SP_POP_0014_FE_ZS	0.0004
81	SH_DYN_0509	0.0004
82	EG_ELC_LOSS_ZS	0.0004
21	SP_POP_1014_MA_5Y	0.0004
40	SP_POP_5054_MA_5Y	0.0004
7	IT_NET_USER_ZS	0.0004
9	SP_DYN_LEOO_IN	0.0004

3	NY_GNP_PCAP_CD	0.0003
13	SE_SEC_ENRR	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
8	SP_DYN_LEOO_MA_IN	0.0003
85	SP_POP_4044_FE_5Y	0.0003
70	FM_AST_PRVT_GD_ZS	0.0003
68	SP_DYN_AMRT_FE	0.0003
62	SH_DYN_2024	0.0003
36	SP_DYN_CBRT_IN	0.0003
34	SP_POP_6064_MA_5Y	0.0003
33	SP_POP_65UP_FE_ZS	0.0003
28	SP_POP_DPND_OL	0.0002
67	SP_POP_4549_FE_5Y	0.0002
6	NY_GDP_PCAP_CD	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
37	SP_POP_5559_MA_5Y	0.0002
31	SP_POP_0509_FE_5Y	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
15	SP_POP_0509_MA_5Y	0.0002
64	SH_DYN_MORT	0.0002
73	SE_PRM_ENRL_TC_ZS	0.0002
14	SP_POP_0014_MA_ZS	0.0001
60	SH_DYN_MORT_MA	0.0001
16	SP_POP_7074_MA_5Y	0.0001
48	SP_DYN_IMRT_IN	0.0001
47	SP_DYN_IMRT_MA_IN	0.0001
66	SH_DYN_MORT_FE	0.0000

Model with 91 variables and max depth None:

Training+Validation R^2: 0.99719, RMSE: 1.00888

Testing R^2: 0.98564, RMSE: 2.46276

Mean cross-validation score: 0.98473

	Feature	Importance
0	CPI_EST_avg	0.9080
90	CPI_EST_prev	0.0306
75	SP_URB_TOTL_IN_ZS	0.0024
41	SP_POP_DPND_YG	0.0020
46	SP_DYN_TO65_FE_ZS	0.0017
83	SP_POP_1564_MA_ZS	0.0016
39	SH_DYN_NMRT	0.0014
11	SP_POP_65UP_MA_ZS	0.0013
17	SE_SEC_ENRR_FE	0.0013
4	NY_GDP_PCAP_KD_rel	0.0013
54	SP_POP_5559_FE_5Y	0.0012
44	EG_USE_ELEC_KH_PC	0.0012
49	SP_POP_7579_FE_5Y	0.0011

87	ITCELSETS_P2	0.0011
12	SPDYNLE00_FE_IN	0.0011
35	SPPOP1519_FE_5Y	0.0011
55	SHDYN1519	0.0010
72	SPPOP2024_FE_5Y	0.0010
57	SPPOP4549_MA_5Y	0.0009
64	SHDYN_MORT	0.0009
53	SPPOP5054_FE_5Y	0.0009
69	FDAST_PRVT_GD_ZS	0.0009
51	NV_SRV_TOTL_ZS	0.0009
33	SPPOP65UP_FE_ZS	0.0009
65	SETER_ENRR	0.0008
63	FSAST_PRVT_GD_ZS	0.0008
88	SPPOP4044_MA_5Y	0.0008
78	SG_LAW_INDX	0.0008
50	SPDYN_IMRT_FE_IN	0.0008
56	SPPOP6064_FE_5Y	0.0008
31	SPPOP0509_FE_5Y	0.0008
84	EN_ATM_CO2E_PC	0.0008
52	SPADO_TFR	0.0007
23	SPPOP0014_TO_ZS	0.0007
22	SE_SEC_ENRR_MA	0.0007
19	SPPOP80UP_FE_5Y	0.0007
76	SHDYN1014	0.0007
81	SHDYN0509	0.0007
86	SHIMM_IDPT	0.0007
58	SEPRE_ENRR	0.0006
59	NVAGR_TOTL_ZS	0.0006
74	SPRUR_TOTL_ZS	0.0006
80	NECON_PRVT_ZS	0.0006
48	SPDYN_IMRT_IN	0.0006
45	SPPOP6569_FE_5Y	0.0006
7	ITNET_USER_ZS	0.0006
34	SPPOP6064_MA_5Y	0.0006
24	SPPOP65UP_TO_ZS	0.0006
20	SPDYN_T065_MA_ZS	0.0006
18	SPPOP7579_MA_5Y	0.0006
29	SPPOP1014_FE_5Y	0.0006
32	SPPOP1519_MA_5Y	0.0006
30	SPPOP0014_FE_ZS	0.0006
38	SPPOP0004_FE_5Y	0.0006
40	SPPOP5054_MA_5Y	0.0006
42	SPPOP7074_FE_5Y	0.0006
43	SPDYN_AMRT_MA	0.0006
36	SPDYN_CBRT_IN	0.0005
82	EGELC_LOSS_ZS	0.0005
61	SPDYN_TFR_IN	0.0005
71	TMVAL_MRCH_HI_ZS	0.0005

70	FM_AST_PRVT_GD_ZS	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
66	SH_DYN_MORT_FE	0.0005
21	SP_POP_1014_MA_5Y	0.0005
8	SP_DYN_LE00_MA_IN	0.0004
16	SP_POP_7074_MA_5Y	0.0004
9	SP_DYN_LE00_IN	0.0004
68	SP_DYN_AMRT_FE	0.0004
67	SP_POP_4549_FE_5Y	0.0004
85	SP_POP_4044_FE_5Y	0.0004
26	IT_MLT_MAIN_P2	0.0004
79	SP_POP_2024_MA_5Y	0.0004
27	SP_POP_0004_MA_5Y	0.0004
28	SP_POP_DPND_DL	0.0004
89	SP_POP_DPND	0.0004
13	SE_SEC_ENRR	0.0004
6	NY_GDP_PCAP_CD	0.0003
5	NY_GDP_PCAP_KD	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
77	EG_USE_PCAP_KG_OE	0.0003
15	SP_POP_0509_MA_5Y	0.0003
62	SH_DYN_2024	0.0003
73	SE_PRM_ENRL_TC_ZS	0.0002
3	NY_GNP_PCAP_CD	0.0002
37	SP_POP_5559_MA_5Y	0.0002
47	SP_DYN_IMRT_MA_IN	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
14	SP_POP_0014_MA_ZS	0.0001
25	SP_POP_6569_MA_5Y	0.0001
60	SH_DYN_MORT_MA	0.0000

Model with 92 variables and max depth None:

Training+Validation R^2: 0.99854, RMSE: 0.72765

Testing R^2: 0.98561, RMSE: 2.46555

Mean cross-validation score: 0.98432

	Feature	Importance
0	CPI_EST_avg	0.9218
91	CPI_EST_prev	0.0347
44	EG_USE_ELEC_KH_PC	0.0012
83	SP_POP_1564_MA_ZS	0.0012
17	SE_SEC_ENRR_FE	0.0011
46	SP_DYN_T065_FE_ZS	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
41	SP_POP_DPND_YG	0.0010
87	IT_CEL_SETS_P2	0.0010
39	SH_DYN_NMRT	0.0010

75	SP_URB_TOTL_IN_ZS	0.0010
4	NY_GDP_PCAP_KD_rel	0.0010
12	SP_DYN_LE00_FE_IN	0.0008
88	SP_POP_4044_MA_5Y	0.0008
32	SP_POP_1519_MA_5Y	0.0007
22	SE_SEC_ENRR_MA	0.0007
20	SP_DYN_T065_MA_ZS	0.0007
84	EN_ATM_CO2E_PC	0.0007
57	SP_POP_4549_MA_5Y	0.0007
65	SE_TER_ENRR	0.0007
74	SP_RUR_TOTL_ZS	0.0007
55	SH_DYN_1519	0.0006
63	FS_AST_PRVT_GD_ZS	0.0006
52	SP_ADO_TFRT	0.0006
72	SP_POP_2024_FE_5Y	0.0006
51	NV_SRV_TOTL_ZS	0.0006
42	SP_POP_7074_FE_5Y	0.0006
35	SP_POP_1519_FE_5Y	0.0006
53	SP_POP_5054_FE_5Y	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
19	SP_POP_80UP_FE_5Y	0.0006
76	SH_DYN_1014	0.0005
70	FM_AST_PRVT_GD_ZS	0.0005
59	NV_AGR_TOTL_ZS	0.0005
54	SP_POP_5559_FE_5Y	0.0005
49	SP_POP_7579_FE_5Y	0.0005
45	SP_POP_6569_FE_5Y	0.0005
40	SP_POP_5054_MA_5Y	0.0005
86	SH_IMM_IDPT	0.0005
90	NE_EXP_GNFS_KD	0.0005
7	IT_NET_USER_ZS	0.0005
89	SP_POP_DPNP	0.0004
58	SE_PRE_ENRR	0.0004
60	SH_DYN_MORT_MA	0.0004
61	SP_DYN_TFRT_IN	0.0004
62	SH_DYN_2024	0.0004
78	SG_LAW_INDX	0.0004
8	SP_DYN_LE00_MA_IN	0.0004
56	SP_POP_6064_FE_5Y	0.0004
67	SP_POP_4549_FE_5Y	0.0004
68	SP_DYN_AMRT_FE	0.0004
29	SP_POP_1014_FE_5Y	0.0004
71	TM_VAL_MRCH_HI_ZS	0.0004
77	EG_USE_PCAP_KG_OE	0.0004
13	SE_SEC_ENRR	0.0004
66	SH_DYN_MORT_FE	0.0004
43	SP_DYN_AMRT_MA	0.0004
18	SP_POP_7579_MA_5Y	0.0004

82	EG_ELC_LOSS_ZS	0.0004
5	NY_GDP_PCAP_KD	0.0004
26	IT_MLT_MAIN_P2	0.0004
25	SP_POP_6569_MA_5Y	0.0004
80	NE_CON_PRVT_ZS	0.0004
36	SP_DYN_CBRT_IN	0.0004
14	SP_POP_0014_MA_ZS	0.0003
9	SP_DYN_LEOO_IN	0.0003
69	FD_AST_PRVT_GD_ZS	0.0003
81	SH_DYN_0509	0.0003
31	SP_POP_0509_FE_5Y	0.0003
27	SP_POP_0004_MA_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0003
79	SP_POP_2024_MA_5Y	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
21	SP_POP_1014_MA_5Y	0.0003
85	SP_POP_4044_FE_5Y	0.0003
38	SP_POP_0004_FE_5Y	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0002
73	SE_PRM_ENRL_TC_ZS	0.0002
50	SP_DYN_IMRT_FE_IN	0.0002
48	SP_DYN_IMRT_IN	0.0002
3	NY_GNP_PCAP_CD	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
37	SP_POP_5559_MA_5Y	0.0002
34	SP_POP_6064_MA_5Y	0.0002
28	SP_POP_DPND_DL	0.0002
15	SP_POP_0509_MA_5Y	0.0002
6	NY_GDP_PCAP_CD	0.0002
64	SH_DYN_MORT	0.0001
33	SP_POP_65UP_FE_ZS	0.0001
47	SP_DYN_IMRT_MA_IN	0.0001
30	SP_POP_0014_FE_ZS	0.0000
23	SP_POP_0014_TO_ZS	0.0000

Model with 93 variables and max depth None:

Training+Validation R^2: 0.99854, RMSE: 0.72765

Testing R^2: 0.98561, RMSE: 2.46555

Mean cross-validation score: 0.98414

	Feature	Importance
0	CPI_EST_avg	0.9237
92	CPI_EST_prev	0.0315
46	SP_DYN_TO65_FE_ZS	0.0013
17	SE_SEC_ENRR_FE	0.0012
44	EG_USE_ELEC_KH_PC	0.0011
41	SP_POP_DPND_YG	0.0011

75	SP_URB_TOTL_IN_ZS	0.0010
39	SH_DYN_NMRT	0.0010
4	NY_GDP_PCAP_KD_rel	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
12	SP_DYN_LE00_FE_IN	0.0010
88	SP_POP_4044_MA_5Y	0.0009
83	SP_POP_1564_MA_ZS	0.0009
22	SE_SEC_ENRR_MA	0.0008
87	IT_CEL_SETS_P2	0.0008
74	SP_RUR_TOTL_ZS	0.0007
76	SH_DYN_1014	0.0007
84	EN_ATM_CO2E_PC	0.0007
35	SP_POP_1519_FE_5Y	0.0007
65	SE_TER_ENRR	0.0007
57	SP_POP_4549_MA_5Y	0.0006
55	SH_DYN_1519	0.0006
51	NV_SRV_TOTL_ZS	0.0006
7	IT_NET_USER_ZS	0.0006
52	SP_ADO_TFRT	0.0006
42	SP_POP_7074_FE_5Y	0.0006
53	SP_POP_5054_FE_5Y	0.0006
49	SP_POP_7579_FE_5Y	0.0006
54	SP_POP_5559_FE_5Y	0.0006
19	SP_POP_80UP_FE_5Y	0.0006
20	SP_DYN_T065_MA_ZS	0.0006
63	FS_AST_PRVT_GD_ZS	0.0006
89	SP_POP_DPND	0.0005
40	SP_POP_5054_MA_5Y	0.0005
10	SP_POP_80UP_MA_5Y	0.0005
71	TM_VAL_MRCH_HI_ZS	0.0005
45	SP_POP_6569_FE_5Y	0.0005
70	FM_AST_PRVT_GD_ZS	0.0005
80	NE_CON_PRVT_ZS	0.0005
90	NE_EXP_GNFS_KD	0.0005
59	NV_AGR_TOTL_ZS	0.0005
60	SH_DYN_MORT_MA	0.0005
36	SP_DYN_CBRT_IN	0.0005
66	SH_DYN_MORT_FE	0.0005
86	SH_IMM_IDPT	0.0005
32	SP_POP_1519_MA_5Y	0.0005
67	SP_POP_4549_FE_5Y	0.0004
82	EG_ELC_LOSS_ZS	0.0004
58	SE_PRE_ENRR	0.0004
72	SP_POP_2024_FE_5Y	0.0004
81	SH_DYN_0509	0.0004
61	SP_DYN_TFRT_IN	0.0004
77	EG_USE_PCAP_KG_OE	0.0004
78	SG_LAW_INDX	0.0004

62	SH_DYN_2024	0.0004
43	SP_DYN_AMRT_MA	0.0004
21	SP_POP_1014_MA_5Y	0.0004
8	SP_DYN_LEOO_MA_IN	0.0004
5	NY_GDP_PCAP_KD	0.0004
26	IT_MLT_MAIN_P2	0.0004
25	SP_POP_6569_MA_5Y	0.0004
24	SP_POP_65UP_TO_ZS	0.0004
18	SP_POP_7579_MA_5Y	0.0004
56	SP_POP_6064_FE_5Y	0.0003
9	SP_DYN_LEOO_IN	0.0003
13	SE_SEC_ENRR	0.0003
14	SP_POP_0014_MA_ZS	0.0003
15	SP_POP_0509_MA_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0003
68	SP_DYN_AMRT_FE	0.0003
69	FD_AST_PRVT_GD_ZS	0.0003
50	SP_DYN_IMRT_FE_IN	0.0003
27	SP_POP_0004_MA_5Y	0.0003
28	SP_POP_DPND_DL	0.0003
29	SP_POP_1014_FE_5Y	0.0003
31	SP_POP_0509_FE_5Y	0.0003
85	SP_POP_4044_FE_5Y	0.0003
38	SP_POP_0004_FE_5Y	0.0003
48	SP_DYN_IMRT_IN	0.0003
79	SP_POP_2024_MA_5Y	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0002
73	SE_PRM_ENRL_TC_ZS	0.0002
3	NY_GNP_PCAP_CD	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
37	SP_POP_5559_MA_5Y	0.0002
34	SP_POP_6064_MA_5Y	0.0002
33	SP_POP_65UP_FE_ZS	0.0002
6	NY_GDP_PCAP_CD	0.0002
30	SP_POP_0014_FE_ZS	0.0001
23	SP_POP_0014_TO_ZS	0.0001
47	SP_DYN_IMRT_MA_IN	0.0001
64	SH_DYN_MORT	0.0000
91	SP_POP_1564_TO_ZS	0.0000

Model with 94 variables and max depth None:
 Training+Validation R^2: 0.99885, RMSE: 0.64572
 Testing R^2: 0.98703, RMSE: 2.34089
 Mean cross-validation score: 0.98479

	Feature	Importance
0	CPI_EST_avg	0.9189

93	CPI_EST_prev	0.0327
46	SP_DYN_T065_FE_ZS	0.0021
17	SE_SEC_ENRR_FE	0.0012
30	SP_POP_0014_FE_ZS	0.0012
12	SP_DYN_LE00_FE_IN	0.0011
34	SP_POP_6064_MA_5Y	0.0011
76	SH_DYN_1014	0.0010
4	NY_GDP_PCAP_KD_rel	0.0010
41	SP_POP_DPND_YG	0.0010
75	SP_URB_TOTL_IN_ZS	0.0010
44	EG_USE_ELEC_KH_PC	0.0010
39	SH_DYN_NMRT	0.0010
72	SP_POP_2024_FE_5Y	0.0009
66	SH_DYN_MORT_FE	0.0009
55	SH_DYN_1519	0.0008
49	SP_POP_7579_FE_5Y	0.0008
54	SP_POP_5559_FE_5Y	0.0008
87	IT_CEL_SETS_P2	0.0008
45	SP_POP_6569_FE_5Y	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
19	SP_POP_80UP_FE_5Y	0.0007
28	SP_POP_DPND_DL	0.0007
38	SP_POP_0004_FE_5Y	0.0007
83	SP_POP_1564_MA_ZS	0.0006
84	EN_ATM_CO2E_PC	0.0006
22	SE_SEC_ENRR_MA	0.0006
42	SP_POP_7074_FE_5Y	0.0006
53	SP_POP_5054_FE_5Y	0.0006
57	SP_POP_4549_MA_5Y	0.0006
23	SP_POP_0014_TO_ZS	0.0006
65	SE_TER_ENRR	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
91	SP_POP_1564_TO_ZS	0.0006
74	SP_RUR_TOTL_ZS	0.0006
51	NV_SRV_TOTL_ZS	0.0005
50	SP_DYN_IMRT_FE_IN	0.0005
43	SP_DYN_AMRT_MA	0.0005
60	SH_DYN_MORT_MA	0.0005
61	SP_DYN_TFRT_IN	0.0005
62	SH_DYN_2024	0.0005
63	FS_AST_PRVT_GD_ZS	0.0005
82	EG_ELC_LOSS_ZS	0.0005
70	FM_AST_PRVT_GD_ZS	0.0005
25	SP_POP_6569_MA_5Y	0.0005
7	IT_NET_USER_ZS	0.0005
18	SP_POP_7579_MA_5Y	0.0005
35	SP_POP_1519_FE_5Y	0.0005
20	SP_DYN_T065_MA_ZS	0.0005

29	SP_POP_1014_FE_5Y	0.0005
56	SP_POP_6064_FE_5Y	0.0004
89	SP_POP_DPND	0.0004
21	SP_POP_1014_MA_5Y	0.0004
59	NV_AGR_TOTL_ZS	0.0004
58	SE_PRE_ENRR	0.0004
80	NE_CON_PRVT_ZS	0.0004
48	SP_DYN_IMRT_IN	0.0004
8	SP_DYN_LEOO_MA_IN	0.0004
26	IT_MLT_MAIN_P2	0.0004
92	TM_VAL_TRAN_ZS_WT	0.0004
52	SP_ADO_TFRT	0.0004
31	SP_POP_0509_FE_5Y	0.0004
77	EG_USE_PCAP_KG_OE	0.0004
78	SG_LAW_INDX	0.0004
5	NY_GDP_PCAP_KD	0.0004
90	NE_EXP_GNFS_KD	0.0004
79	SP_POP_2024_MA_5Y	0.0003
88	SP_POP_4044_MA_5Y	0.0003
85	SP_POP_4044_FE_5Y	0.0003
71	TM_VAL_MRCH_HI_ZS	0.0003
86	SH_IMM_IDPT	0.0003
47	SP_DYN_IMRT_MA_IN	0.0003
69	FD_AST_PRVT_GD_ZS	0.0003
67	SP_POP_4549_FE_5Y	0.0003
9	SP_DYN_LEOO_IN	0.0003
13	SE_SEC_ENRR	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
40	SP_POP_5054_MA_5Y	0.0003
36	SP_DYN_CBRT_IN	0.0003
32	SP_POP_1519_MA_5Y	0.0003
27	SP_POP_0004_MA_5Y	0.0002
6	NY_GDP_PCAP_CD	0.0002
14	SP_POP_0014_MA_ZS	0.0002
15	SP_POP_0509_MA_5Y	0.0002
16	SP_POP_7074_MA_5Y	0.0002
33	SP_POP_65UP_FE_ZS	0.0002
73	SE_PRM_ENRL_TC_ZS	0.0002
81	SH_DYN_0509	0.0002
37	SP_POP_5559_MA_5Y	0.0002
3	NY_GNP_PCAP_CD	0.0002
64	SH_DYN_MORT	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
24	SP_POP_65UP_TO_ZS	0.0001
68	SP_DYN_AMRT_FE	0.0001

Model with 95 variables and max depth None:

Training+Validation R^2: 0.99948, RMSE: 0.43525
 Testing R^2: 0.98653, RMSE: 2.38573
 Mean cross-validation score: 0.98496

	Feature	Importance
0	CPI_EST_avg	0.8880
94	CPI_EST_prev	0.0374
4	NY_GDP_PCAP_KD_rel	0.0033
12	SP_DYN_LEOO_FE_IN	0.0025
44	EG_USE_ELEC_KH_PC	0.0024
46	SP_DYN_T065_FE_ZS	0.0019
17	SE_SEC_ENRR_FE	0.0019
39	SH_DYN_NMRT	0.0018
19	SP_POP_80UP_FE_5Y	0.0015
84	EN_ATM_CO2E_PC	0.0015
42	SP_POP_7074_FE_5Y	0.0014
41	SP_POP_DPND_YG	0.0014
20	SP_DYN_T065_MA_ZS	0.0013
87	IT_CEL_SETS_P2	0.0012
49	SP_POP_7579_FE_5Y	0.0012
60	SH_DYN_MORT_MA	0.0012
92	TM_VAL_TRAN_ZS_WT	0.0012
69	FD_AST_PRVT_GD_ZS	0.0012
35	SP_POP_1519_FE_5Y	0.0012
51	NV_SRV_TOTL_ZS	0.0011
72	SP_POP_2024_FE_5Y	0.0011
65	SE_TER_ENRR	0.0011
82	EG_ELC_LOSS_ZS	0.0011
63	FS_AST_PRVT_GD_ZS	0.0010
74	SP_RUR_TOTL_ZS	0.0010
58	SE_PRE_ENRR	0.0010
57	SP_POP_4549_MA_5Y	0.0010
27	SP_POP_0004_MA_5Y	0.0010
54	SP_POP_5559_FE_5Y	0.0010
10	SP_POP_80UP_MA_5Y	0.0010
11	SP_POP_65UP_MA_ZS	0.0010
93	SH_IMM_MEAS	0.0010
5	NY_GDP_PCAP_KD	0.0009
40	SP_POP_5054_MA_5Y	0.0009
22	SE_SEC_ENRR_MA	0.0009
18	SP_POP_7579_MA_5Y	0.0009
78	SG_LAW_INDX	0.0009
59	NV_AGR_TOTL_ZS	0.0009
45	SP_POP_6569_FE_5Y	0.0008
67	SP_POP_4549_FE_5Y	0.0008
55	SH_DYN_1519	0.0008
56	SP_POP_6064_FE_5Y	0.0008
53	SP_POP_5054_FE_5Y	0.0008

83	SP_POP_1564_MA_ZS	0.0008
34	SP_POP_6064_MA_5Y	0.0008
28	SP_POP_DPND_OL	0.0007
85	SP_POP_4044_FE_5Y	0.0007
88	SP_POP_4044_MA_5Y	0.0007
50	SP_DYN_IMRT_FE_IN	0.0007
71	TM_VAL_MRCH_HI_ZS	0.0007
80	NE_CON_PRVT_ZS	0.0007
32	SP_POP_1519_MA_5Y	0.0007
21	SP_POP_1014_MA_5Y	0.0007
29	SP_POP_1014_FE_5Y	0.0007
70	FM_AST_PRVT_GD_ZS	0.0007
24	SP_POP_65UP_TO_ZS	0.0007
90	NE_EXP_GNFS_KD	0.0006
13	SE_SEC_ENRR	0.0006
86	SH_IMM_IDPT	0.0006
61	SP_DYN_TFRT_IN	0.0006
38	SP_POP_0004_FE_5Y	0.0006
7	IT_NET_USER_ZS	0.0006
68	SP_DYN_AMRT_FE	0.0006
89	SP_POP_DPND	0.0006
52	SP_ADO_TFRT	0.0006
76	SH_DYN_1014	0.0006
36	SP_DYN_CBRT_IN	0.0006
8	SP_DYN_LE00_MA_IN	0.0006
79	SP_POP_2024_MA_5Y	0.0005
73	SE_PRM_ENRL_TC_ZS	0.0005
26	IT_MLT_MAIN_P2	0.0005
77	EG_USE_PCAP_KG_OE	0.0005
37	SP_POP_5559_MA_5Y	0.0005
43	SP_DYN_AMRT_MA	0.0004
64	SH_DYN_MORT	0.0004
62	SH_DYN_2024	0.0004
33	SP_POP_65UP_FE_ZS	0.0004
31	SP_POP_0509_FE_5Y	0.0004
25	SP_POP_6569_MA_5Y	0.0004
15	SP_POP_0509_MA_5Y	0.0004
1	NE_CON_PRVT_PC_KD	0.0003
9	SP_DYN_LE00_IN	0.0003
6	NY_GDP_PCAP_CD	0.0003
47	SP_DYN_IMRT_MA_IN	0.0003
2	NY_ADJ_NNTY_PC_CD	0.0002
81	SH_DYN_0509	0.0002
3	NY_GNP_PCAP_CD	0.0002
16	SP_POP_7074_MA_5Y	0.0001
14	SP_POP_0014_MA_ZS	0.0001
75	SP_URB_TOTL_IN_ZS	0.0000
66	SH_DYN_MORT_FE	0.0000

30	SP_POP_0014_FE_ZS	0.0000
23	SP_POP_0014_TO_ZS	0.0000
91	SP_POP_1564_TO_ZS	0.0000
48	SP_DYN_IMRT_IN	0.0000

Model with 96 variables and max depth None:
 Training+Validation R^2: 0.9991, RMSE: 0.57212
 Testing R^2: 0.98659, RMSE: 2.38023
 Mean cross-validation score: 0.98465

	Feature	Importance
0	CPI_EST_avg	0.9289
95	CPI_EST_prev	0.0296
49	SP_POP_7579_FE_5Y	0.0012
44	EG_USE_ELEC_KH_PC	0.0012
17	SE_SEC_ENRR_FE	0.0011
4	NY_GDP_PCAP_KD_rel	0.0010
45	SP_POP_6569_FE_5Y	0.0009
72	SP_POP_2024_FE_5Y	0.0008
42	SP_POP_7074_FE_5Y	0.0008
46	SP_DYN_T065_FE_ZS	0.0008
11	SP_POP_65UP_MA_ZS	0.0008
12	SP_DYN_LE00_FE_IN	0.0008
75	SP_URB_TOTL_IN_ZS	0.0008
57	SP_POP_4549_MA_5Y	0.0007
41	SP_POP_DPND_YG	0.0007
87	IT_CEL_SETS_P2	0.0007
27	SP_POP_0004_MA_5Y	0.0007
54	SP_POP_5559_FE_5Y	0.0007
39	SH_DYN_NMRT	0.0007
63	FS_AST_PRVT_GD_ZS	0.0007
93	SH_IMM_MEAS	0.0007
19	SP_POP_80UP_FE_5Y	0.0006
76	SH_DYN_1014	0.0006
22	SE_SEC_ENRR_MA	0.0006
53	SP_POP_5054_FE_5Y	0.0006
51	NV_SRV_TOTL_ZS	0.0006
32	SP_POP_1519_MA_5Y	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
66	SH_DYN_MORT_FE	0.0005
67	SP_POP_4549_FE_5Y	0.0005
43	SP_DYN_AMRT_MA	0.0005
59	NV_AGR_TOTL_ZS	0.0005
55	SH_DYN_1519	0.0005
79	SP_POP_2024_MA_5Y	0.0005
52	SP_ADO_TFRT	0.0005
78	SG_LAW_INDX	0.0005

65	SE_TER_ENRR	0.0005
74	SP_RUR_TOTL_ZS	0.0005
94	TX_VAL_SERV_CD_WT	0.0005
84	EN_ATM_CO2E_PC	0.0005
20	SP_DYN_T065_MA_ZS	0.0005
83	SP_POP_1564_MA_ZS	0.0004
88	SP_POP_4044_MA_5Y	0.0004
5	NY_GDP_PCAP_KD	0.0004
70	FM_AST_PRVT_GD_ZS	0.0004
7	IT_NET_USER_ZS	0.0004
64	SH_DYN_MORT	0.0004
92	TM_VAL_TRAN_ZS_WT	0.0004
61	SP_DYN_TFRT_IN	0.0004
18	SP_POP_7579_MA_5Y	0.0004
36	SP_DYN_CBRT_IN	0.0004
56	SP_POP_6064_FE_5Y	0.0004
86	SH_IMM_IDPT	0.0004
50	SP_DYN_IMRT_FE_IN	0.0004
35	SP_POP_1519_FE_5Y	0.0004
80	NE_CON_PRVT_ZS	0.0004
34	SP_POP_6064_MA_5Y	0.0004
82	EG_ELC_LOSS_ZS	0.0004
13	SE_SEC_ENRR	0.0003
40	SP_POP_5054_MA_5Y	0.0003
71	TM_VAL_MRCH_HI_ZS	0.0003
69	FD_AST_PRVT_GD_ZS	0.0003
77	EG_USE_PCAP_KG_OE	0.0003
33	SP_POP_65UP_FE_ZS	0.0003
8	SP_DYN_LE00_MA_IN	0.0003
89	SP_POP_DPNP	0.0003
23	SP_POP_0014_TO_ZS	0.0003
15	SP_POP_0509_MA_5Y	0.0003
60	SH_DYN_MORT_MA	0.0003
58	SE_PRE_ENRR	0.0003
26	IT_MLT_MAIN_P2	0.0003
21	SP_POP_1014_MA_5Y	0.0003
38	SP_POP_0004_FE_5Y	0.0003
85	SP_POP_4044_FE_5Y	0.0002
90	NE_EXP_GNFS_KD	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
29	SP_POP_1014_FE_5Y	0.0002
6	NY_GDP_PCAP_CD	0.0002
3	NY_GNP_PCAP_CD	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
9	SP_DYN_LE00_IN	0.0002
37	SP_POP_5559_MA_5Y	0.0002
16	SP_POP_7074_MA_5Y	0.0002
24	SP_POP_65UP_TO_ZS	0.0002

25	SP_POP_6569_MA_5Y	0.0001
14	SP_POP_0014_MA_ZS	0.0001
91	SP_POP_1564_TO_ZS	0.0001
73	SE_PRM_ENRL_TC_ZS	0.0001
28	SP_POP_DPNP_OL	0.0001
68	SP_DYN_AMRT_FE	0.0001
31	SP_POP_0509_FE_5Y	0.0001
47	SP_DYN_IMRT_MA_IN	0.0001
81	SH_DYN_0509	0.0001
62	SH_DYN_2024	0.0001
30	SP_POP_0014_FE_ZS	0.0000
48	SP_DYN_IMRT_IN	0.0000

Model with 97 variables and max depth None:

Training+Validation R^2: 0.99969, RMSE: 0.33611

Testing R^2: 0.9871, RMSE: 2.33402

Mean cross-validation score: 0.98482

	Feature	Importance
0	CPI_EST_avg	0.8915
96	CPI_EST_prev	0.0367
4	NY_GDP_PCAP_KD_rel	0.0033
41	SP_POP_DPNP_YG	0.0026
44	EG_USE_ELEC_KH_PC	0.0025
17	SE_SEC_ENRR_FE	0.0017
72	SP_POP_2024_FE_5Y	0.0015
66	SH_DYN_MORT_FE	0.0015
11	SP_POP_65UP_MA_ZS	0.0014
87	IT_CEL_SETS_P2	0.0013
82	EG_ELC_LOSS_ZS	0.0012
12	SP_DYN_LEOO_FE_IN	0.0012
22	SE_SEC_ENRR_MA	0.0012
76	SH_DYN_1014	0.0012
50	SP_DYN_IMRT_FE_IN	0.0012
24	SP_POP_65UP_TO_ZS	0.0011
74	SP_RUR_TOTL_ZS	0.0011
5	NY_GDP_PCAP_KD	0.0011
95	TM_VAL_SERV_CD_WT	0.0010
79	SP_POP_2024_MA_5Y	0.0010
45	SP_POP_6569_FE_5Y	0.0010
57	SP_POP_4549_MA_5Y	0.0010
27	SP_POP_0004_MA_5Y	0.0009
39	SH_DYN_NMRT	0.0009
60	SH_DYN_MORT_MA	0.0009
85	SP_POP_4044_FE_5Y	0.0009
49	SP_POP_7579_FE_5Y	0.0009
69	FD_AST_PRVT_GD_ZS	0.0009

94	TX_VAL_SERV_CD_WT	0.0009
46	SP_DYN_T065_FE_ZS	0.0009
20	SP_DYN_T065_MA_ZS	0.0009
93	SH_IMM_MEAS	0.0009
54	SP_POP_5559_FE_5Y	0.0008
73	SE_PRM_ENRL_TC_ZS	0.0008
42	SP_POP_7074_FE_5Y	0.0008
63	FS_AST_PRVT_GD_ZS	0.0008
51	NV_SRV_TOTL_ZS	0.0008
65	SE_TER_ENRR	0.0008
36	SP_DYN_CBRT_IN	0.0008
7	IT_NET_USER_ZS	0.0008
84	EN_ATM_CO2E_PC	0.0008
92	TM_VAL_TRAN_ZS_WT	0.0008
19	SP_POP_80UP_FE_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0007
6	NY_GDP_PCAP_CD	0.0007
62	SH_DYN_2024	0.0007
9	SP_DYN_LE00_IN	0.0007
55	SH_DYN_1519	0.0007
53	SP_POP_5054_FE_5Y	0.0007
90	NE_EXP_GNFS_KD	0.0007
52	SP_ADO_TFRT	0.0007
88	SP_POP_4044_MA_5Y	0.0007
80	NE_CON_PRVT_ZS	0.0007
70	FM_AST_PRVT_GD_ZS	0.0006
71	TM_VAL_MRCH_HI_ZS	0.0006
89	SP_POP_DPND	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
58	SE_PRE_ENRR	0.0006
59	NV_AGR_TOTL_ZS	0.0006
32	SP_POP_1519_MA_5Y	0.0006
40	SP_POP_5054_MA_5Y	0.0006
67	SP_POP_4549_FE_5Y	0.0006
34	SP_POP_6064_MA_5Y	0.0006
77	EG_USE_PCAP_KG_OE	0.0005
86	SH_IMM_IDPT	0.0005
68	SP_DYN_AMRT_FE	0.0005
48	SP_DYN_IMRT_IN	0.0005
13	SE_SEC_ENRR	0.0005
29	SP_POP_1014_FE_5Y	0.0005
56	SP_POP_6064_FE_5Y	0.0005
15	SP_POP_0509_MA_5Y	0.0005
43	SP_DYN_AMRT_MA	0.0005
21	SP_POP_1014_MA_5Y	0.0005
35	SP_POP_1519_FE_5Y	0.0005
28	SP_POP_DPND_OL	0.0005
31	SP_POP_0509_FE_5Y	0.0005

91	SP_POP_1564_TO_ZS	0.0004
25	SP_POP_6569_MA_5Y	0.0004
26	IT_MLT_MAIN_P2	0.0004
78	SG_LAW_INDX	0.0004
8	SP_DYN_LEOO_MA_IN	0.0004
61	SP_DYN_TFRT_IN	0.0004
16	SP_POP_7074_MA_5Y	0.0003
81	SH_DYN_0509	0.0003
47	SP_DYN_IMRT_MA_IN	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
3	NY_GNP_PCAP_CD	0.0003
83	SP_POP_1564_MA_ZS	0.0002
33	SP_POP_65UP_FE_ZS	0.0002
37	SP_POP_5559_MA_5Y	0.0002
75	SP_URB_TOTL_IN_ZS	0.0002
14	SP_POP_0014_MA_ZS	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
23	SP_POP_0014_TO_ZS	0.0001
38	SP_POP_0004_FE_5Y	0.0001
64	SH_DYN_MORT	0.0000
30	SP_POP_0014_FE_ZS	0.0000

Model with 98 variables and max depth None:

Training+Validation R^2: 0.99916, RMSE: 0.55046

Testing R^2: 0.98684, RMSE: 2.35741

Mean cross-validation score: 0.98484

	Feature	Importance
0	CPI_EST_avg	0.8919
97	CPI_EST_prev	0.0367
4	NY_GDP_PCAP_KD_rel	0.0028
41	SP_POP_DPNP_YG	0.0026
44	EG_USE_ELEC_KH_PC	0.0023
17	SE_SEC_ENRR_FE	0.0017
82	EG_ELC_LOSS_ZS	0.0015
66	SH_DYN_MORT_FE	0.0015
72	SP_POP_2024_FE_5Y	0.0015
87	IT_CEL_SETS_P2	0.0013
11	SP_POP_65UP_MA_ZS	0.0013
22	SE_SEC_ENRR_MA	0.0012
50	SP_DYN_IMRT_FE_IN	0.0012
12	SP_DYN_LEOO_FE_IN	0.0012
76	SH_DYN_1014	0.0012
74	SP_RUR_TOTL_ZS	0.0011
5	NY_GDP_PCAP_KD	0.0011
79	SP_POP_2024_MA_5Y	0.0010
69	FD_AST_PRVT_GD_ZS	0.0010

95	TM_VAL_SERV_CD_WT	0.0010
57	SP_POP_4549_MA_5Y	0.0010
24	SP_POP_65UP_TO_ZS	0.0010
45	SP_POP_6569_FE_5Y	0.0010
39	SH_DYN_NMRT	0.0009
85	SP_POP_4044_FE_5Y	0.0009
93	SH_IMM_MEAS	0.0009
94	TX_VAL_SERV_CD_WT	0.0009
60	SH_DYN_MORT_MA	0.0009
46	SP_DYN_T065_FE_ZS	0.0009
49	SP_POP_7579_FE_5Y	0.0009
20	SP_DYN_T065_MA_ZS	0.0009
27	SP_POP_0004_MA_5Y	0.0009
63	FS_AST_PRVT_GD_ZS	0.0008
84	EN_ATM_CO2E_PC	0.0008
54	SP_POP_5559_FE_5Y	0.0008
65	SE_TER_ENRR	0.0008
19	SP_POP_80UP_FE_5Y	0.0008
42	SP_POP_7074_FE_5Y	0.0008
18	SP_POP_7579_MA_5Y	0.0008
36	SP_DYN_CBRT_IN	0.0008
92	TM_VAL_TRAN_ZS_WT	0.0008
7	IT_NET_USER_ZS	0.0008
51	NV_SRV_TOTL_ZS	0.0008
80	NE_CON_PRVT_ZS	0.0007
73	SE_PRM_ENRL_TC_ZS	0.0007
53	SP_POP_5054_FE_5Y	0.0007
9	SP_DYN_LEOO_IN	0.0007
70	FM_AST_PRVT_GD_ZS	0.0007
88	SP_POP_4044_MA_5Y	0.0007
90	NE_EXP_GNFS_KD	0.0007
62	SH_DYN_2024	0.0007
55	SH_DYN_1519	0.0007
6	NY_GDP_PCAP_CD	0.0007
52	SP_ADO_TFRT	0.0007
43	SP_DYN_AMRT_MA	0.0006
10	SP_POP_80UP_MA_5Y	0.0006
67	SP_POP_4549_FE_5Y	0.0006
34	SP_POP_6064_MA_5Y	0.0006
40	SP_POP_5054_MA_5Y	0.0006
89	SP_POP_DPND	0.0006
71	TM_VAL_MRCH_HI_ZS	0.0006
59	NV_AGR_TOTL_ZS	0.0006
58	SE_PRE_ENRR	0.0006
32	SP_POP_1519_MA_5Y	0.0006
35	SP_POP_1519_FE_5Y	0.0005
15	SP_POP_0509_MA_5Y	0.0005
29	SP_POP_1014_FE_5Y	0.0005

28	SP_POP_DPND_OL	0.0005
86	SH_IMM_IDPT	0.0005
77	EG_USE_PCAP_KG_OE	0.0005
68	SP_DYN_AMRT_FE	0.0005
31	SP_POP_0509_FE_5Y	0.0005
48	SP_DYN_IMRT_IN	0.0005
21	SP_POP_1014_MA_5Y	0.0005
56	SP_POP_6064_FE_5Y	0.0005
13	SE_SEC_ENRR	0.0005
26	IT_MLT_MAIN_P2	0.0004
25	SP_POP_6569_MA_5Y	0.0004
3	NY_GNP_PCAP_CD	0.0004
91	SP_POP_1564_TO_ZS	0.0004
61	SP_DYN_TFRT_IN	0.0004
78	SG_LAW_INDX	0.0004
8	SP_DYN_LEOO_MA_IN	0.0004
16	SP_POP_7074_MA_5Y	0.0003
81	SH_DYN_0509	0.0003
47	SP_DYN_IMRT_MA_IN	0.0003
1	NE_CON_PRVT_PC_KD	0.0003
83	SP_POP_1564_MA_ZS	0.0002
37	SP_POP_5559_MA_5Y	0.0002
14	SP_POP_0014_MA_ZS	0.0002
75	SP_URB_TOTL_IN_ZS	0.0002
33	SP_POP_65UP_FE_ZS	0.0002
2	NY_ADJ_NNTY_PC_CD	0.0002
38	SP_POP_0004_FE_5Y	0.0001
96	BX_GSR_NFSV_CD	0.0001
23	SP_POP_0014_TO_ZS	0.0001
30	SP_POP_0014_FE_ZS	0.0000
64	SH_DYN_MORT	0.0000

Model with 99 variables and max depth None:

Training+Validation R^2: 0.99806, RMSE: 0.83716

Testing R^2: 0.98597, RMSE: 2.43476

Mean cross-validation score: 0.98459

	Feature	Importance
0	CPI_EST_avg	0.9349
98	CPI_EST_prev	0.0289
4	NY_GDP_PCAP_KD_rel	0.0010
46	SP_DYN_TO65_FE_ZS	0.0009
11	SP_POP_65UP_MA_ZS	0.0008
44	EG_USE_ELEC_KH_PC	0.0008
76	SH_DYN_1014	0.0008
72	SP_POP_2024_FE_5Y	0.0007
87	IT_CEL_SETS_P2	0.0007

95	TM_VAL_SERV_CD_WT	0.0007
12	SP_DYN_LE00_FE_IN	0.0007
39	SH_DYN_NMRT	0.0007
10	SP_POP_80UP_MA_5Y	0.0006
70	FM_AST_PRVT_GD_ZS	0.0006
45	SP_POP_6569_FE_5Y	0.0006
17	SE_SEC_ENRR_FE	0.0006
19	SP_POP_80UP_FE_5Y	0.0006
20	SP_DYN_T065_MA_ZS	0.0006
65	SE_TER_ENRR	0.0006
63	FS_AST_PRVT_GD_ZS	0.0006
54	SP_POP_5559_FE_5Y	0.0005
43	SP_DYN_AMRT_MA	0.0005
7	IT_NET_USER_ZS	0.0005
32	SP_POP_1519_MA_5Y	0.0005
51	NV_SRV_TOTL_ZS	0.0005
53	SP_POP_5054_FE_5Y	0.0005
28	SP_POP_DPND_OL	0.0005
26	IT_MLT_MAIN_P2	0.0005
79	SP_POP_2024_MA_5Y	0.0005
22	SE_SEC_ENRR_MA	0.0005
15	SP_POP_0509_MA_5Y	0.0005
93	SH_IMM_MEAS	0.0005
57	SP_POP_4549_MA_5Y	0.0004
58	SE_PRE_ENRR	0.0004
80	NE_CON_PRVT_ZS	0.0004
48	SP_DYN_IMRT_IN	0.0004
42	SP_POP_7074_FE_5Y	0.0004
78	SG_LAW_INDX	0.0004
61	SP_DYN_TFRT_IN	0.0004
77	EG_USE_PCAP_KG_OE	0.0004
49	SP_POP_7579_FE_5Y	0.0004
74	SP_RUR_TOTL_ZS	0.0004
84	EN_ATM_CO2E_PC	0.0004
85	SP_POP_4044_FE_5Y	0.0004
97	TM_VAL_OTHZ_ZS_WT	0.0004
89	SP_POP_DPND	0.0004
94	TX_VAL_SERV_CD_WT	0.0004
23	SP_POP_0014_TO_ZS	0.0004
37	SP_POP_5559_MA_5Y	0.0003
24	SP_POP_65UP_TO_ZS	0.0003
68	SP_DYN_AMRT_FE	0.0003
67	SP_POP_4549_FE_5Y	0.0003
6	NY_GDP_PCAP_CD	0.0003
9	SP_DYN_LE00_IN	0.0003
13	SE_SEC_ENRR	0.0003
62	SH_DYN_2024	0.0003
18	SP_POP_7579_MA_5Y	0.0003

55	SH_DYN_1519	0.0003
40	SP_POP_5054_MA_5Y	0.0003
56	SP_POP_6064_FE_5Y	0.0003
5	NY_GDP_PCAP_KD	0.0003
71	TM_VAL_MRCH_HI_ZS	0.0003
29	SP_POP_1014_FE_5Y	0.0003
52	SP_ADO_TFRT	0.0003
30	SP_POP_0014_FE_ZS	0.0003
81	SH_DYN_0509	0.0003
88	SP_POP_4044_MA_5Y	0.0003
47	SP_DYN_IMRT_MA_IN	0.0003
83	SP_POP_1564_MA_ZS	0.0003
75	SP_URB_TOTL_IN_ZS	0.0003
82	EG_ELC_LOSS_ZS	0.0002
90	NE_EXP_GNFS_KD	0.0002
92	TM_VAL_TRAN_ZS_WT	0.0002
50	SP_DYN_IMRT_FE_IN	0.0002
73	SE_PRM_ENRL_TC_ZS	0.0002
27	SP_POP_0004_MA_5Y	0.0002
66	SH_DYN_MORT_FE	0.0002
8	SP_DYN_LEOO_MA_IN	0.0002
59	NV_AGR_TOTL_ZS	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
41	SP_POP_DPND_YG	0.0002
36	SP_DYN_CBRT_IN	0.0002
21	SP_POP_1014_MA_5Y	0.0002
69	FD_AST_PRVT_GD_ZS	0.0002
33	SP_POP_65UP_FE_ZS	0.0002
31	SP_POP_0509_FE_5Y	0.0002
96	BX_GSR_NFSV_CD	0.0001
14	SP_POP_0014_MA_ZS	0.0001
16	SP_POP_7074_MA_5Y	0.0001
25	SP_POP_6569_MA_5Y	0.0001
86	SH_IMM_IDPT	0.0001
34	SP_POP_6064_MA_5Y	0.0001
35	SP_POP_1519_FE_5Y	0.0001
38	SP_POP_0004_FE_5Y	0.0001
3	NY_GNP_PCAP_CD	0.0001
60	SH_DYN_MORT_MA	0.0001
64	SH_DYN_MORT	0.0001
2	NY_ADJ_NNTY_PC_CD	0.0001
91	SP_POP_1564_TO_ZS	0.0000

Model with 100 variables and max depth None:

Training+Validation R^2: 0.99677, RMSE: 1.08174

Testing R^2: 0.98589, RMSE: 2.44136

Mean cross-validation score: 0.98462

	Feature	Importance
0	CPI_EST_avg	0.9249
99	CPI_EST_prev	0.0292
12	SP_DYN_LE00_FE_IN	0.0013
20	SP_DYN_T065_MA_ZS	0.0012
11	SP_POP_65UP_MA_ZS	0.0011
4	NY_GDP_PCAP_KD_rel	0.0011
33	SP_POP_65UP_FE_ZS	0.0009
19	SP_POP_80UP_FE_5Y	0.0009
17	SE_SEC_ENRR_FE	0.0009
95	TM_VAL_SERV_CD_WT	0.0009
44	EG_USE_ELEC_KH_PC	0.0009
72	SP_POP_2024_FE_5Y	0.0009
39	SH_DYN_NMRT	0.0008
46	SP_DYN_T065_FE_ZS	0.0008
65	SE_TER_ENRR	0.0008
93	SH_IMM_MEAS	0.0008
87	IT_CEL_SETS_P2	0.0008
76	SH_DYN_1014	0.0008
10	SP_POP_80UP_MA_5Y	0.0007
98	BM_GSR_NFSV_CD	0.0007
79	SP_POP_2024_MA_5Y	0.0007
57	SP_POP_4549_MA_5Y	0.0007
22	SE_SEC_ENRR_MA	0.0007
45	SP_POP_6569_FE_5Y	0.0007
51	NV_SRV_TOTL_ZS	0.0006
53	SP_POP_5054_FE_5Y	0.0006
54	SP_POP_5559_FE_5Y	0.0006
84	EN_ATM_CO2E_PC	0.0006
70	FM_AST_PRVT_GD_ZS	0.0006
74	SP_RUR_TOTL_ZS	0.0006
63	FS_AST_PRVT_GD_ZS	0.0006
42	SP_POP_7074_FE_5Y	0.0006
26	IT_MLT_MAIN_P2	0.0006
32	SP_POP_1519_MA_5Y	0.0005
24	SP_POP_65UP_TO_ZS	0.0005
89	SP_POP_DPND	0.0005
30	SP_POP_0014_FE_ZS	0.0005
58	SE_PRE_ENRR	0.0005
97	TM_VAL_OTHZ_ZS_WT	0.0005
43	SP_DYN_AMRT_MA	0.0005
88	SP_POP_4044_MA_5Y	0.0005
40	SP_POP_5054_MA_5Y	0.0005
28	SP_POP_DPND_DL	0.0005
77	EG_USE_PCAP_KG_OE	0.0005
9	SP_DYN_LE00_IN	0.0005
80	NE_CON_PRVT_ZS	0.0005

61	SP_DYN_TFRT_IN	0.0004
62	SH_DYN_2024	0.0004
15	SP_POP_0509_MA_5Y	0.0004
56	SP_POP_6064_FE_5Y	0.0004
55	SH_DYN_1519	0.0004
59	NV_AGR_TOTL_ZS	0.0004
49	SP_POP_7579_FE_5Y	0.0004
7	IT_NET_USER_ZS	0.0004
47	SP_DYN_IMRT_MA_IN	0.0004
82	EG_ELC_LOSS_ZS	0.0004
78	SG_LAW_INDX	0.0004
94	TX_VAL_SERV_CD_WT	0.0004
85	SP_POP_4044_FE_5Y	0.0004
73	SE_PRM_ENRL_TC_ZS	0.0003
8	SP_DYN_LE00_MA_IN	0.0003
13	SE_SEC_ENRR	0.0003
83	SP_POP_1564_MA_ZS	0.0003
71	TM_VAL_MRCH_HI_ZS	0.0003
6	NY_GDP_PCAP_CD	0.0003
69	FD_AST_PRVT_GD_ZS	0.0003
5	NY_GDP_PCAP_KD	0.0003
21	SP_POP_1014_MA_5Y	0.0003
67	SP_POP_4549_FE_5Y	0.0003
16	SP_POP_7074_MA_5Y	0.0003
18	SP_POP_7579_MA_5Y	0.0003
23	SP_POP_0014_TO_ZS	0.0003
52	SP_ADO_TFRT	0.0003
25	SP_POP_6569_MA_5Y	0.0003
92	TM_VAL_TRAN_ZS_WT	0.0003
96	BX_GSR_NFSV_CD	0.0003
37	SP_POP_5559_MA_5Y	0.0003
35	SP_POP_1519_FE_5Y	0.0003
34	SP_POP_6064_MA_5Y	0.0003
81	SH_DYN_0509	0.0003
86	SH_IMM_IDPT	0.0002
90	NE_EXP_GNFS_KD	0.0002
50	SP_DYN_IMRT_FE_IN	0.0002
64	SH_DYN_MORT	0.0002
60	SH_DYN_MORT_MA	0.0002
3	NY_GNP_PCAP_CD	0.0002
1	NE_CON_PRVT_PC_KD	0.0002
48	SP_DYN_IMRT_IN	0.0002
68	SP_DYN_AMRT_FE	0.0002
27	SP_POP_0004_MA_5Y	0.0002
38	SP_POP_0004_FE_5Y	0.0002
75	SP_URB_TOTL_IN_ZS	0.0001
31	SP_POP_0509_FE_5Y	0.0001
14	SP_POP_0014_MA_ZS	0.0001

```
29    SP_POP_1014_FE_5Y      0.0001
91    SP_POP_1564_TO_ZS      0.0001
36      SP_DYN_CBRT_IN       0.0001
41      SP_POP_DPND_YG       0.0001
66      SH_DYN_MORT_FE       0.0001
2      NY_ADJ_NNTY_PC_CD     0.0001
```

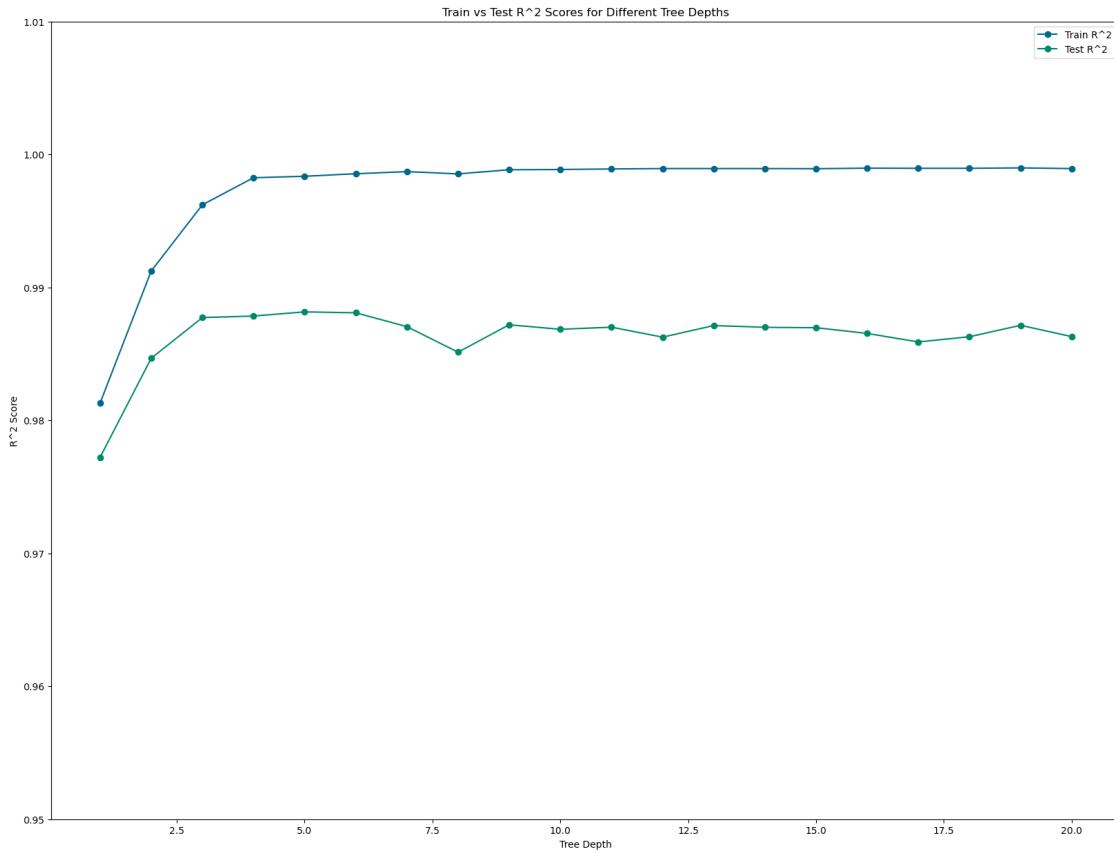
```
[270]: # Plotting
fig, ax = plt.subplots(figsize=(20, 15))

# Plot the R^2 scores for the training set
ax.plot(max_depth_values, train_r2_scores, marker='o', color='#00688B', □
         ↪label='Train R^2')

# Plot the R^2 scores for the test set
ax.plot(max_depth_values, test_r2_scores, marker='o', color='#008B68', □
         ↪label='Test R^2')

ax.set_xlabel('Tree Depth')
ax.set_ylabel('R^2 Score')
ax.set_title('Train vs Test R^2 Scores for Different Tree Depths')
ax.set_ylim(0.95,1.01)
ax.legend()

plt.show()
```



```
[274]: # Plot the difference between the training and test R^2 scores
fig, ax = plt.subplots(figsize=(20, 10))

# Calculate the difference between the training and test R^2 scores
r2_diff = np.array(train_r2_scores) - np.array(test_r2_scores)

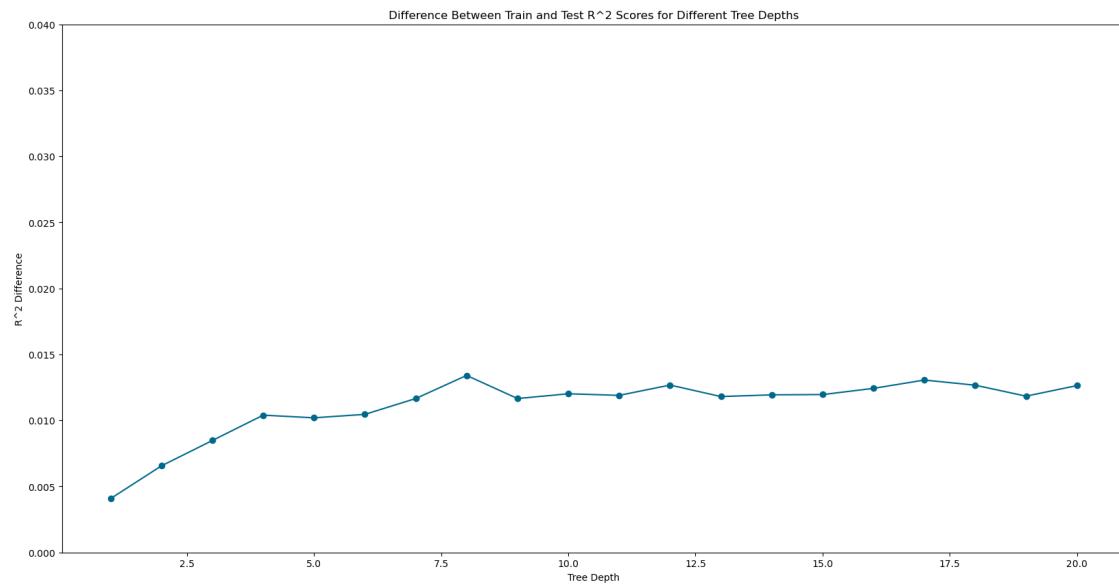
# Plot the difference between the training and test R^2 scores
ax.plot(max_depth_values, r2_diff, marker='o', color='#00688B')

ax.set_xlabel('Tree Depth')
ax.set_ylabel('R^2 Difference')
ax.set_title('Difference Between Train and Test R^2 Scores for Different Tree Depths')
ax.set_yticks([0, 0.04])

plt.show()

# Print the maximum R^2 score for the test set and the corresponding tree depth
max_r2_test = max(test_r2_scores)
max_r2_test_depth = max_depth_values[test_r2_scores.index(max_r2_test)]
```

```
print(f"The maximum R^2 score for the test set is {max_r2_test:.5f} at a tree depth of {max_r2_test_depth}")
```



The maximum R² score for the test set is 0.98816 at a tree depth of 5