

Practical Exam Python

Part 2. Practical Question

This dataset contains HR Data Analytics. By using features, predict whether employee will leave company or not. ("left" is target column). You are asked to build model for predicting patients having heart disease.

1. Import data (HR_data.csv) and get familiarized with it. (show statistical summary, percentage of missing values in each column and check duplicate values.) (6 points)
2. Show distribution of variables, correlation between them and check imbalance problem. (6 points)
3. Prepare data for modelling. (7 points)
4. Fit data into different models – Logistic regression, Decision Tree, Naive Bayes and XGBoost models. (10 points)
5. Print detailed reports regarding accuracy metrics for each model. (8 points)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import copy
import warnings
warnings.filterwarnings('ignore')
import plotly.express as px
```

```
df = pd.read_csv('/content/HR_data - HR_data - HR_data - HR_data -  
df
```

	Id	satisfaction_level	last_evaluation	number_project	ave
0	1	0.38	0.53	2	
1	2	0.80	0.86	5	
2	3	0.11	0.88	7	
3	4	0.72	0.87	5	
4	5	0.37	0.52	2	
...
14994	14995	0.40	0.57	2	
14995	14996	0.37	0.48	2	
14996	14997	0.37	0.53	2	
14997	14998	0.11	0.96	6	
14998	14999	0.37	0.52	2	

14999 rows x 12 columns

Next steps:

[Generate code with df](#)[New interactive sheet](#)

```
df.describe()
```

	Id	satisfaction_level	last_evaluation	number_projects
count	14999.000000	14999.000000	14999.000000	14999.000000
mean	7500.000000	0.612834	0.716102	3.803000
std	4329.982679	0.248631	0.171169	1.232500
min	1.000000	0.090000	0.360000	2.000000
25%	3750.500000	0.440000	0.560000	3.000000
50%	7500.000000	0.640000	0.720000	4.000000
75%	11249.500000	0.820000	0.870000	5.000000
max	14999.000000	1.000000	1.000000	7.000000

```
df.isnull().sum()
```

	0
Id	0
satisfaction_level	0
last_evaluation	0
number_project	0
average_monthly_hours	0
time_spend_company	0
Work_accident	0
promotion_last_5years	0
Department	0
salary	0
Age	14841
left	0

```
dtype: int64
```

```
df.nunique()
```

	0
Id	14999
satisfaction_level	92
last_evaluation	65
number_project	6
average_montly_hours	215
time_spend_company	8
Work_accident	2
promotion_last_5years	2
Department	10
salary	3
Age	10
left	2

dtype: int64

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Id                                     14999 non-null  int64
1   satisfaction_level                    14999 non-null  float64
2   last_evaluation                      14999 non-null  float64
3   number_project                      14999 non-null  int64
4   average_montly_hours                14999 non-null  int64
5   time_spend_company                  14999 non-null  int64
6   Work_accident                      14999 non-null  int64
7   promotion_last_5years                14999 non-null  int64
8   Department                          14999 non-null  object
9   salary                              14999 non-null  object
10  Age                                  158 non-null    float64
11  left                                14999 non-null  int64
dtypes: float64(3), int64(7), object(2)
memory usage: 1.4+ MB
```

```
df['Age']
```

	Age
0	23.0
1	34.0
2	19.0
3	35.0
4	45.0
...	...
14994	25.0
14995	23.0
14996	34.0
14997	19.0
14998	35.0

14999 rows × 1 columns

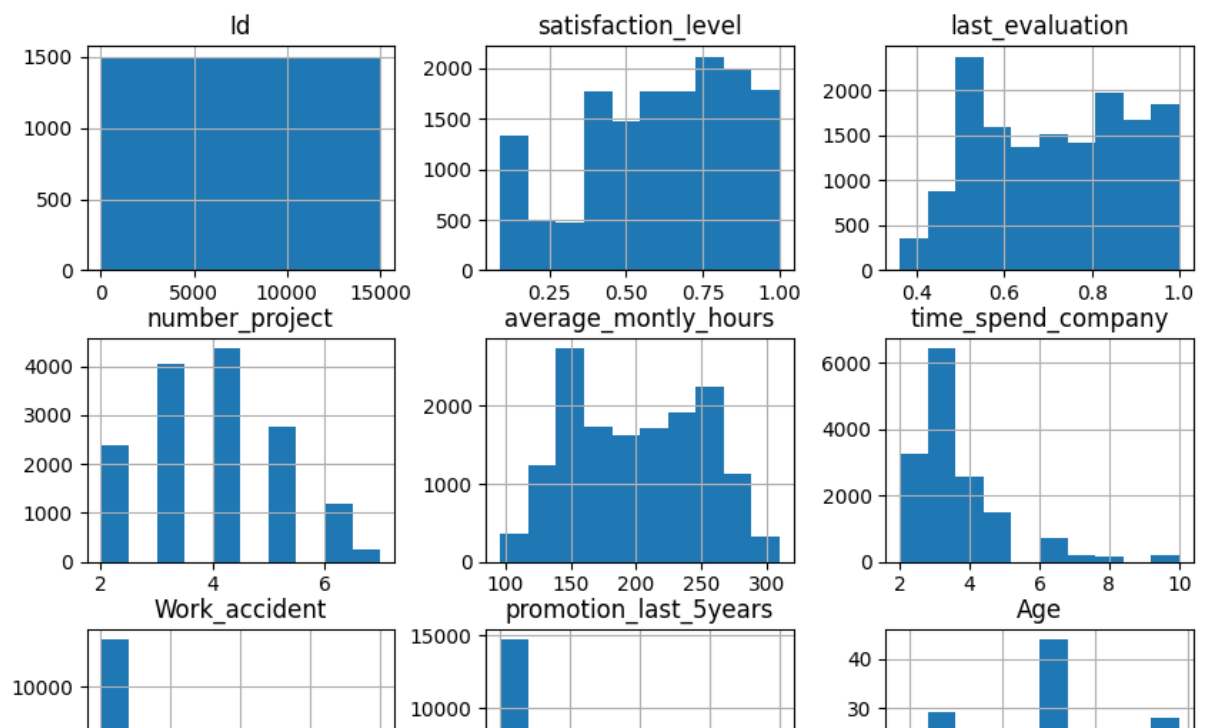
dtype: float64

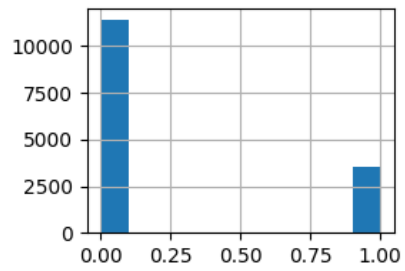
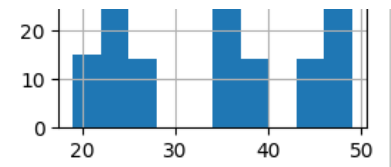
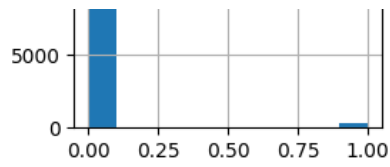
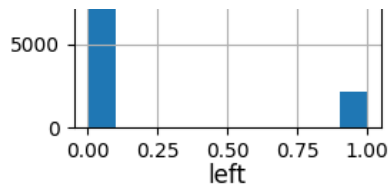
```
df.isnull().mean() * 100
```

	0
Id	0.000000
satisfaction_level	0.000000
last_evaluation	0.000000
number_project	0.000000
average_monthly_hours	0.000000
time_spend_company	0.000000
Work_accident	0.000000
promotion_last_5years	0.000000
Department	0.000000
salary	0.000000
Age	98.946596
left	0.000000

dtype: float64

```
df.hist(figsize=(10,10))
plt.show()
```





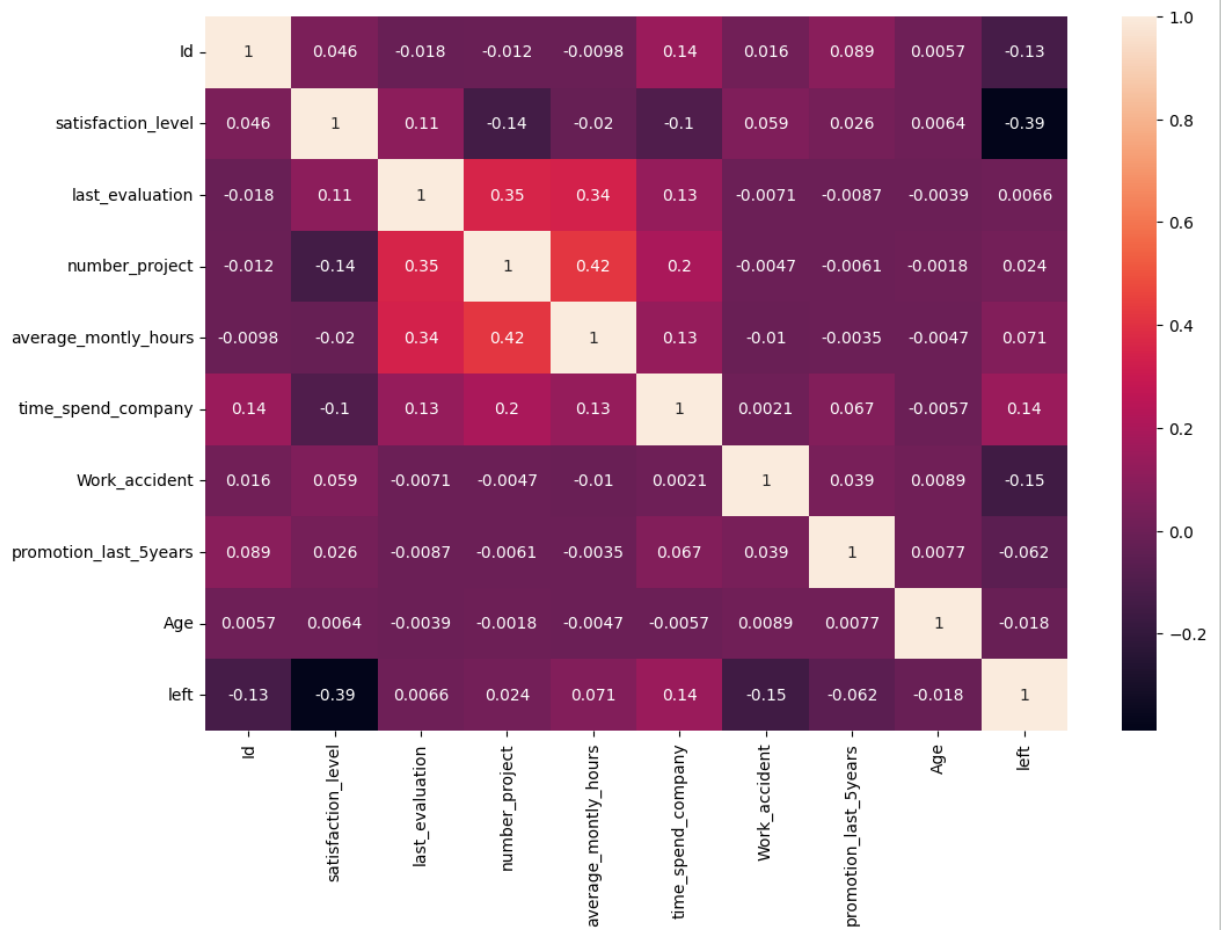
```
df["Age"] = df["Age"].fillna(df["Age"].median())
```

```
df.info()
```

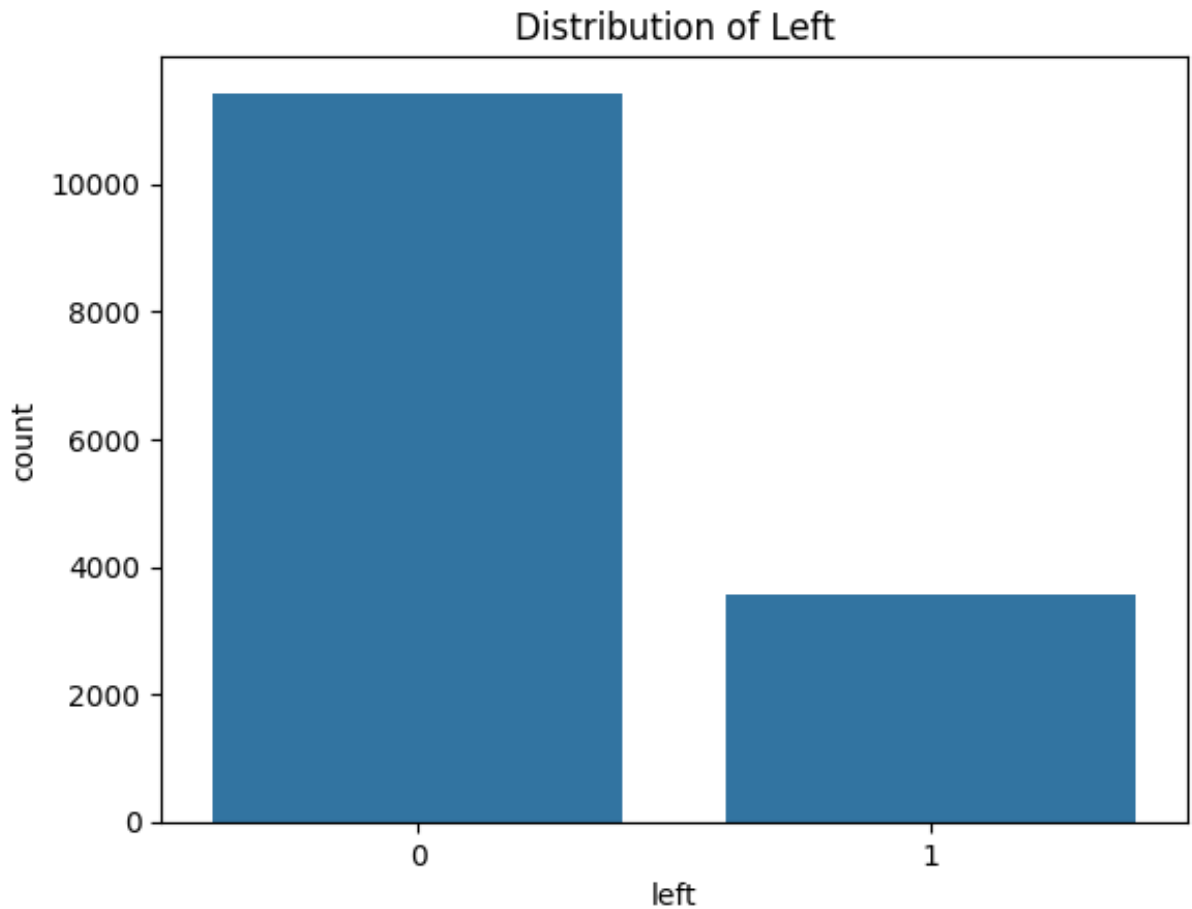
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Id                                    14999 non-null  int64
1   satisfaction_level                    14999 non-null  float64
2   last_evaluation                       14999 non-null  float64
3   number_project                       14999 non-null  int64
4   average_monthly_hours                14999 non-null  int64
5   time_spend_company                  14999 non-null  int64
6   Work_accident                       14999 non-null  int64
7   promotion_last_5years                14999 non-null  int64
8   Department                           14999 non-null  object
9   salary                              14999 non-null  object
10  Age                                  14999 non-null  float64
11  left                                14999 non-null  int64
dtypes: float64(3), int64(7), object(2)
memory usage: 1.4+ MB
```



```
plt.figure(figsize=(12, 8))
sns.heatmap(df.select_dtypes(include = 'number').corr(),annot=True)
plt.show()
```



```
sns.countplot(x = 'left', data = df)
plt.title('Distribution of Left')
plt.show()
```



```
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split

df_encoded = pd.get_dummies(df, columns=['salary', 'Department'])

X = df_encoded.drop('left', axis=1)
y = df_encoded['left']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
```

```
#Logistic regression, Decision Tree, Naive Bayes and XGBoost models
```

```

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, classification_report,

```

```

from sklearn.metrics import accuracy_score, classification_report,

models = {
    'Logistic Regression': LogisticRegression(),
    'Decision Tree': DecisionTreeClassifier(),
    'Naive Bayes': GaussianNB(),
    'XGBoost': XGBClassifier()
}

fit_models = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    fit_models[name] = model

    y_pred = model.predict(X_test)

    print(f'\nModel: {name}')
    print('-----')

    score = accuracy_score(y_test, y_pred)
    print(f"Accuracy Score: {score:.4f}")

    print(classification_report(y_test, y_pred))
    print(confusion_matrix(y_test, y_pred))

```

Model: Logistic Regression

Accuracy Score: 0.7847

	precision	recall	f1-score	support
0	0.81	0.94	0.87	2294
1	0.60	0.26	0.37	706
accuracy			0.78	3000
macro avg	0.70	0.60	0.62	3000
weighted avg	0.76	0.78	0.75	3000

[[2167 127]

```
[ 519  187]]
```

Model: Decision Tree

Accuracy Score: 1.0000

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2294
1	1.00	1.00	1.00	706
accuracy			1.00	3000
macro avg	1.00	1.00	1.00	3000
weighted avg	1.00	1.00	1.00	3000

```
[[2294  0]
 [  0 706]]
```

Model: Naive Bayes

Accuracy Score: 0.8403

	precision	recall	f1-score	support
0	0.83	0.99	0.90	2294
1	0.93	0.35	0.51	706
accuracy			0.84	3000
macro avg	0.88	0.67	0.71	3000
weighted avg	0.85	0.84	0.81	3000

```
[[2274  20]
 [ 459 247]]
```

Model: XGBoost

Accuracy Score: 0.9993

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2294
1	1.00	1.00	1.00	706
accuracy			1.00	3000
macro avg	1.00	1.00	1.00	3000
weighted avg	1.00	1.00	1.00	3000

```
[[2293  1]
 [  1 705]]
```

PART 2: Practical question

1. Import data (pop.csv) and get familiarized with it. (show statistics summary, search for missing values etc.) (5 points)
2. Visualize 'value' column according to time. Find correlation between them. (3 points)
3. Prepare data for modelling. (5 points)
4. Define model parameters. Fit data into Prophet model. (7 points)
5. Plot components of Prophet model, visualize predicted and actual values. (3 points)

	realtime_start	value	date	realtime_end	
0	2019-12-06	156309.000	1952-01-01	2019-12-06	
1	2019-12-06	156527.000	1952-02-01	2019-12-06	
2	2019-12-06	156731.000	1952-03-01	2019-12-06	
3	2019-12-06	156943.000	1952-04-01	2019-12-06	
4	2019-12-06	157140.000	1952-05-01	2019-12-06	
...	
811	2019-12-06	329591.333	2019-08-01	2019-12-06	
812	2019-12-06	329785.872	2019-09-01	2019-12-06	
813	2019-12-06	329982.035	2019-10-01	2019-12-06	
814	2019-12-06	330154.949	2019-11-01	2019-12-06	
815	2019-12-06	330309.946	2019-12-01	2019-12-06	



Next steps:

New interactive sheet

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   realtime_start  816 non-null   object
1   value           816 non-null   float64
2   date            816 non-null   object
3   realtime_end    816 non-null   object
dtypes: float64(1), object(3)
memory usage: 25.6+ KB
```

```
df1.describe()
```

	value	
count	816.000000	
mean	243847.767826	
std	50519.140567	
min	156309.000000	
25%	201725.250000	
50%	239557.500000	
75%	289364.250000	
max	330309.946000	

```
df1.isna().sum()*100
```

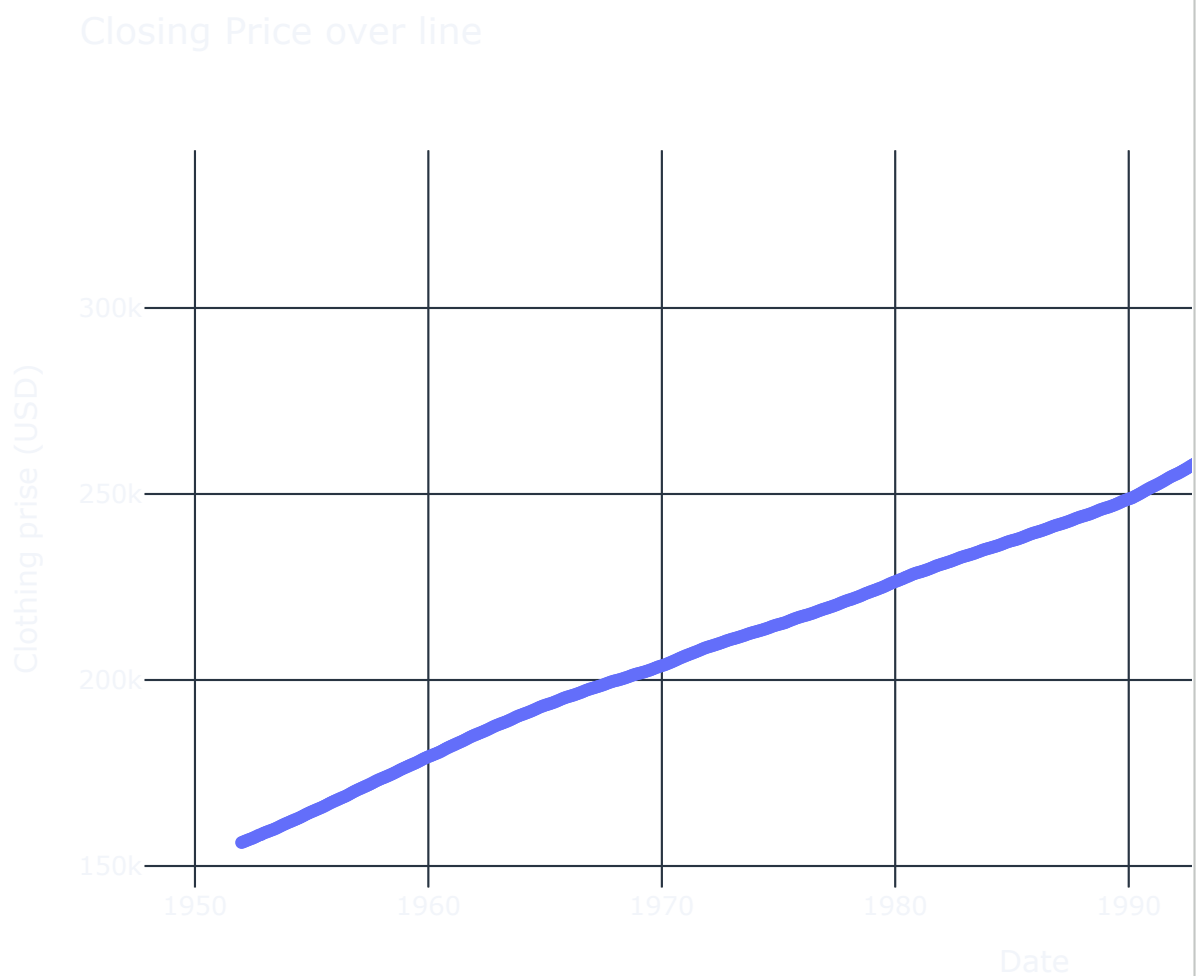
```

0
realtime_start  0
value          0
date           0
realtime_end    0

dtype: int64
```

```
df1['date'] = pd.to_datetime(df1['date'])
```

```
fig = px.line(  
    df1,  
    x = 'date',  
    y = 'value',  
    title = 'Closing Price over line',  
    markers = True  
)  
  
fig.update_layout(  
    template = 'plotly_dark',  
    xaxis_title = 'Date',  
    yaxis_title = 'Clothing prise (USD)',  
    hovermode = 'x unified'  
)  
  
fig.show()
```



```
pip install bayesian-optimization
```

```
Requirement already satisfied: bayesian-optimization in /usr/local/lib/python3.8/site-packages (1.0.0)
Requirement already satisfied: colorama>=0.4.6 in /usr/local/lib/python3.8/site-packages (0.4.6)
Requirement already satisfied: numpy>=1.25 in /usr/local/lib/python3.8/site-packages (1.25.0)
Requirement already satisfied: scikit-learn>=1.0.0 in /usr/local/lib/python3.8/site-packages (1.0.0)
Requirement already satisfied: scipy>=1.0.0 in /usr/local/lib/python3.8/site-packages (1.0.0)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.8/site-packages (1.2.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.8/site-packages (3.1.0)
```

```
import logging
import pandas as pd
import numpy as np
from prophet import Prophet
from bayes_opt import BayesianOptimization
from sklearn.metrics import mean_absolute_error
```

```
df1 = df1[['date', 'value']]
df1.columns = ['ds', 'y']
df1['ds'] = pd.to_datetime(df1['ds'])
```


df1

	ds	y	
0	1952-01-01	156309.000	
1	1952-02-01	156527.000	
2	1952-03-01	156731.000	
3	1952-04-01	156943.000	
4	1952-05-01	157140.000	
...	
811	2019-08-01	329591.333	
812	2019-09-01	329785.872	
813	2019-10-01	329982.035	
814	2019-11-01	330154.949	
815	2019-12-01	330309.946	



816 rows × 2 columns

Next steps:

[Generate code with df1](#)[New interactive sheet](#)

```
train_size = int(0.97 * len(df1))
train_df = df1.iloc[:train_size]
test_df = df1.iloc[train_size:]
print(len(test_df))
test_df
```

25

	ds	y	
791	2017-12-01	326301.399	
792	2018-01-01	326454.123	
793	2018-02-01	326600.823	
794	2018-03-01	326736.690	
795	2018-04-01	326887.866	
796	2018-05-01	327048.704	
797	2018-06-01	327219.140	

798	2018-07-01	327403.909
799	2018-08-01	327600.250
800	2018-09-01	327794.788
801	2018-10-01	327990.950
802	2018-11-01	328163.864
803	2018-12-01	328318.861
804	2019-01-01	328467.812
805	2019-02-01	328610.744
806	2019-03-01	328742.843
807	2019-04-01	328890.250
808	2019-05-01	329047.319
809	2019-06-01	329213.989
810	2019-07-01	329394.993
811	2019-08-01	329591.333
812	2019-09-01	329785.872
813	2019-10-01	329982.035
814	2019-11-01	330154.949
815	2019-12-01	330309.946

Next steps:

[Generate code with test_df](#)[New interactive sheet](#)

```
def Prophet_cv(changepoint_prior_scale, seasonality_prior_scale, hc
    model = Prophet(
        changepoint_prior_scale = changepoint_prior_scale,
        seasonality_prior_scale = seasonality_prior_scale,
        holidays_prior_scale = holidays_prior_scale
    )
    model.fit(train_df)
    future = model.make_future_dataframe(periods = len(test_df))
    forecast = model.predict(future)
    y_pred = forecast['yhat'].iloc[-len(test_df):].values
    y_true = test_df['y'].values
    mae = mean_absolute_error(y_true, y_pred)
    return -mae

pbounds = {
```

```

    'changepoint_prior_scale': (0.001, 0.5),
    'seasonality_prior_scale': (0.1, 20),
    'holidays_prior_scale': (0.1, 20)
}

```

```

optimizer = BayesianOptimization(
    f = Prophet_cv,
    pbounds = pbounds,
    random_state = 42
)

```

```

optimizer.maximize(init_points = 5, n_iter = 25)
best_params = optimizer.max['params']
print('Best params is:', best_params)

```

```

| 11          | -1387.500 | 0.0544436 | 10.553315 | 3.2748893 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 12          | -1376.358 | 0.1682215 | 6.9867828 | 18.985463 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 13          | -1383.002 | 0.0837318 | 4.1586805 | 15.973067 |
| 14          | -1375.227 | 0.3141185 | 10.966887 | 16.372748 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 15          | -1380.651 | 0.2291995 | 9.3771636 | 16.924638 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 16          | -1380.280 | 0.2384266 | 11.359894 | 15.048670 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 17          | -1379.625 | 0.5       | 9.4364520 | 14.988893 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 18          | -1381.830 | 0.4779900 | 6.1442911 | 20.0       |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 19          | -1290.041 | 0.001     | 6.7453180 | 17.772244 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 20          | -1328.888 | 0.0010247 | 6.7394248 | 17.760149 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 21          | -1384.794 | 0.1132860 | 6.6023155 | 17.775058 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_s
| 22          | -1381.215 | 0.1720976 | 10.458435 | 15.702177 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly.

```

```

INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality
| 23          | -1378.379 | 0.0569323 | 6.8573071 | 17.834816 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality
| 24          | -1372.928 | 0.3285945 | 2.0037934 | 18.615393 |
| 25          | -1385.560 | 0.4048916 | 14.927716 | 15.588149 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality
| 26          | -1382.162 | 0.2777512 | 10.941510 | 16.348337 |
| 27          | -1381.835 | 0.2558631 | 15.096773 | 13.237178 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality
| 28          | -1365.610 | 0.0062361 | 6.7494767 | 17.771851 |
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality
| 29          | -1382.647 | 0.1661617 | 19.973530 | 14.795155 |
| 30          | -1385.086 | 0.0488270 | 6.7167462 | 17.758623 |

```

```

best_param = optimizer.max['params']
best_model = Prophet(
    changepoint_prior_scale = best_param['changepoint_prior_scale']
    seasonality_prior_scale = best_param['seasonality_prior_scale']
    holidays_prior_scale = best_param['holidays_prior_scale']
)
best_model.fit(df1)

```

```

future = best_model.make_future_dataframe(periods=len(test_df))
forecast = best_model.predict(future)

```

forecast

```

INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality

```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper
0	1952-01-01	156101.141046	155493.966099	156777.543029	156101.141046	156101.141046
1	1952-02-01	156346.942931	155687.918283	156993.246506	156346.942931	156346.942931
2	1952-03-01	156576.886629	155879.857099	157220.309170	156576.886629	156576.886629
3	1952-04-01	156822.688514	156153.420073	157397.515374	156822.688514	156822.688514
4	1952-05-01	157073.688514	156404.420073	157648.515374	157073.688514	157073.688514

4	2017-05-01	157060.561305	156358.921346	157658.006793	157060.561305	1570
...
836	2019-12-22	331446.625427	331509.629403	332802.605491	331446.625427	3314
837	2019-12-23	331453.068784	331524.051031	332780.239151	331453.068784	3314
838	2019-12-24	331459.512142	331411.893539	332721.808680	331459.512142	3314
839	2019-12-25	331465.955499	331438.439572	332662.363004	331465.955499	3314
840	2019-12-26	331472.398856	331304.062438	332564.576273	331472.398856	3314

841 rows x 16 columns

Next steps:

[Generate code with forecast](#)[New interactive sheet](#)

```
comparison_df = pd.DataFrame({
    'ds': test_df['ds'].values,
    'y_true': test_df['y'].values,
    'y_pred': forecast['yhat'].iloc[-len(test_df):].values
})
```

comparison_df

	ds	y_true	y_pred
0	2017-12-01	326301.399	331376.078837
1	2018-01-01	326454.123	331434.823752
2	2018-02-01	326600.823	331499.140651
3	2018-03-01	326736.690	331567.954578
4	2018-04-01	326887.866	331640.033799
5	2018-05-01	327048.704	331714.019100
6	2018-06-01	327219.140	331788.457205
7	2018-07-01	327403.909	331861.837476
8	2018-08-01	327600.250	331932.630905

9	2018-09-01	327794.788	331999.330362
10	2018-10-01	327990.950	332060.490983
11	2018-11-01	328163.864	332114.769599
12	2018-12-01	328318.861	332160.962077
13	2019-01-01	328467.812	332198.037528
14	2019-02-01	328610.744	332225.168368
15	2019-03-01	328742.843	332241.755371
16	2019-04-01	328890.250	332247.446923
17	2019-05-01	329047.319	332242.151897
18	2019-06-01	329213.989	332226.045686
19	2019-07-01	329394.993	332199.569155
20	2019-08-01	329591.333	332163.420445
21	2019-09-01	329785.872	332118.539762
22	2019-10-01	329982.035	332066.087488
23	2019-11-01	330154.949	332007.416122
24	2019-12-01	330309.946	331944.036765

Next steps:

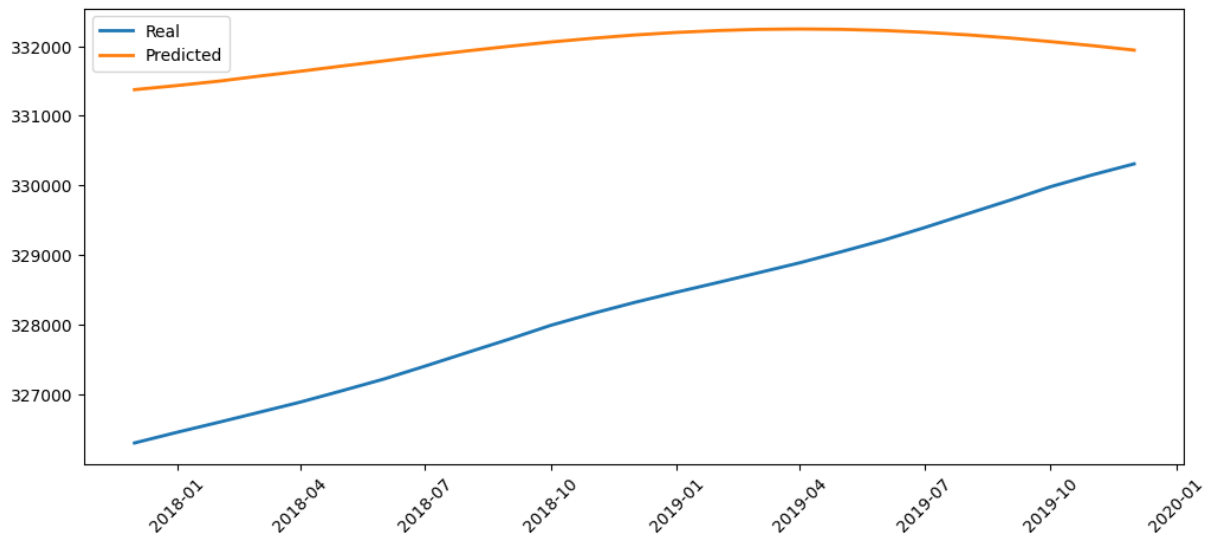
[Generate code with comparison_df](#)[New interactive sheet](#)

```
from sklearn.metrics import mean_absolute_error
```

```
mae = mean_absolute_error(comparison_df['y_true'], comparison_df['y  
mae
```

```
3692.670113446417
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(12,5))
plt.plot(comparison_df['ds'], comparison_df['y_true'], label='Real')
plt.plot(comparison_df['ds'], comparison_df['y_pred'], label='Predicted')
plt.legend()
plt.xticks(rotation=45)
plt.show()
```



```
!pip install neuralprophet
```

```
Collecting neuralprophet
  Downloading neuralprophet-0.8.0-py3-none-any.whl.metadata (9.1 kB)
Collecting captum>=0.6.0 (from neuralprophet)
  Downloading captum-0.8.0-py3-none-any.whl.metadata (26 kB)
Requirement already satisfied: holidays>=0.41 in /usr/local/lib/python3.10/site-packages (from neuralprophet)
Requirement already satisfied: matplotlib<4.0.0,>=3.5.3 in /usr/local/lib/python3.10/site-packages (from neuralprophet)
Requirement already satisfied: nbformat<6.0.0,>=5.8.0 in /usr/local/lib/python3.10/site-packages (from neuralprophet)
Collecting numpy<2.0.0,>=1.25.0 (from neuralprophet)
  Downloading numpy-1.26.4-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (61.0/61.0 kB 5.1 MB/s)
Requirement already satisfied: pandas<3.0.0,>=2.0.0 in /usr/local/lib/python3.10/site-packages (from neuralprophet)
Requirement already satisfied: plotly<6.0.0,>=5.13.1 in /usr/local/lib/python3.10/site-packages (from neuralprophet)
```

```

Collecting pytorch-lightning<2.0.0,>=1.9.4 (from neuralprophet)
  Downloading pytorch_lightning-1.9.5-py3-none-any.whl.metadata (21
Requirement already satisfied: tensorboard<3.0.0,>=2.11.2 in /usr/
Requirement already satisfied: torch<3.0.0,>=2.0.0 in /usr/local/li
Collecting torchmetrics<2.0.0,>=1.0.0 (from neuralprophet)
  Downloading torchmetrics-1.8.2-py3-none-any.whl.metadata (22 kB)
Requirement already satisfied: typing-extensions<5.0.0,>=4.5.0 in
Requirement already satisfied: packaging in /usr/local/lib/python3.
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/di
Requirement already satisfied: python-dateutil in /usr/local/lib/py
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/p
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/pytho
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/p
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/l
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/py
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in /usr/
Requirement already satisfied: traitlets>=5.1 in /usr/local/lib/pyt
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/pytho
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/pyt
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/py
Requirement already satisfied: PyYAML>=5.4 in /usr/local/lib/pythor
Requirement already satisfied: fsspec>2021.06.0 in /usr/local/lib/p
Collecting lightning-utilities>=0.6.0.post0 (from pytorch-lightning)
  Downloading lightning_utilities-0.15.2-py3-none-any.whl.metadata
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/pytho
Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/pyt
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/py
Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in /usr/lo
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/li
Requirement already satisfied: six>1.9 in /usr/local/lib/python3.12
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/py
Requirement already satisfied: filelock in /usr/local/lib/python3.1
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/pyth
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/py
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/
Requirement already satisfied: aiohttp!=4.0.0a0,!4.0.0a1 in /usr/
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/pyth
Requirement already satisfied: jsonschema-specifications>=2023.03.0
Requirement already satisfied: referencing>=0.28.4 in /usr/local/li
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/pyt
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/li
Requirement already satisfied: markupsafe>=2.1.1 in /usr/local/lib/
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in /usr/loc
Requirement already satisfied: aiosignal>=1.4.0 in /usr/local/lib/p
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/li
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/p
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.

```



```

-----
Downloading neuralprophet-0.8.0-py3-none-any.whl (145 kB)
----- 145.4/145.4 kB 7.4 MB/s
Downloading captum-0.8.0-py3-none-any.whl (1.4 MB)
----- 1.4/1.4 MB 34.8 MB/s et
Downloading numpy-1.26.4-cp312-cp312-manylinux_2_17_x86_64.manylin
----- 18.0/18.0 MB 92.7 MB/s
Downloading pytorch_lightning-1.9.5-py3-none-any.whl (829 kB)
----- 829.5/829.5 kB 48.7 MB/s
Downloading torchmetrics-1.8.2-py3-none-any.whl (983 kB)
----- 983.2/983.2 kB 57.3 MB/s
Downloading lightning_utilities-0.15.2-py3-none-any.whl (29 kB)
Installing collected packages: numpy, lightning-utilities, torchmet
  Attempting uninstall: numpy
    Found existing installation: numpy 2.0.2
    Uninstalling numpy-2.0.2:
      Successfully uninstalled numpy-2.0.2
ERROR: pip's dependency resolver does not currently take into account
pytensor 2.35.1 requires numpy>=2.0, but you have numpy 1.26.4 which
jaxlib 0.7.2 requires numpy>=2.0, but you have numpy 1.26.4 which i
jax 0.7.2 requires numpy>=2.0, but you have numpy 1.26.4 which is i
opencv-python 4.12.0.88 requires numpy<2.3.0,>=2; python_version >=
shap 0.50.0 requires numpy>=2, but you have numpy 1.26.4 which is i
opencv-contrib-python 4.12.0.88 requires numpy<2.3.0,>=2; python_ve
opencv-python-headless 4.12.0.88 requires numpy<2.3.0,>=2; python_v
Successfully installed captum-0.8.0 lightning-utilities-0.15.2 neu
WARNING: The following packages were previously imported in this
runtime:
  [numpy]
You must restart the runtime in order to use newly installed
versions.

```

RESTART SESSION

```

from neuralprophet import NeuralProphet

print("NeuralProphet imported successfully!")

```

```

-----
KeyboardInterrupt                                Traceback (most recent
call last)
/tmp/ipython-input-2069683832.py in <cell line: 0>()
----> 1 from neuralprophet import NeuralProphet
      2
      3 print("NeuralProphet imported successfully!")

----- 20 frames -----
/usr/local/lib/python3.12/dist-packages/torch/_library/autograd.py
in <module>

```

```
!pip install neuralprophet
```

```
from neuralprophet import NeuralProphet

print("NeuralProphet imported successfully!")
```

KeyboardInterrupt

```
conda create -n prophet python=3.10
conda activate prophet
pip install neuralprophet
```

```
!pip install neuralforecast
```

```
Requirement already satisfied: neuralforecast in /usr/local/lib/py
Requirement already satisfied: coreforecast>=0.0.6 in /usr/local/l
Requirement already satisfied: fsspec in /usr/local/lib/python3.12
Requirement already satisfied: numpy>=1.21.6 in /usr/local/lib/pytl
Requirement already satisfied: pandas>=1.3.5 in /usr/local/lib/pytl
Requirement already satisfied: torch>=2.4.0 in /usr/local/lib/pytho
Requirement already satisfied: pytorch-lightning>=2.0.0 in /usr/lo
Requirement already satisfied: ray>=2.2.0 in /usr/local/lib/python.
Requirement already satisfied: optuna in /usr/local/lib/python3.12
Requirement already satisfied: utilsforecast>=0.2.3 in /usr/local/
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/loca
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/pytho
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/py
Requirement already satisfied: tqdm>=4.57.0 in /usr/local/lib/pytho
Requirement already satisfied: PyYAML>5.4 in /usr/local/lib/python.
Requirement already satisfied: torchmetrics>0.7.0 in /usr/local/lil
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/p
Requirement already satisfied: typing-extensions>4.5.0 in /usr/loc
Requirement already satisfied: lightning-utilities>=0.10.0 in /usr
Requirement already satisfied: click!=8.3.*,>=7.0 in /usr/local/lil
Requirement already satisfied: filelock in /usr/local/lib/python3.
Requirement already satisfied: jsonschema in /usr/local/lib/python.
Requirement already satisfied: msgpack<2.0.0,>=1.0.0 in /usr/local
Requirement already satisfied: protobuf>=3.20.3 in /usr/local/lib/
Requirement already satisfied: requests in /usr/local/lib/python3.
Requirement already satisfied: pydantic!=2.0.*,!=2.1.*,!=2.2.*,!=2
Requirement already satisfied: tensorboardX>=1.9 in /usr/local/lib
Requirement already satisfied: pyarrow>=9.0.0 in /usr/local/lib/py
Requirement already satisfied: setuptools in /usr/local/lib/python.
```

```

Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12.5/site-packages/sympy-1.13.3-py3.12.egg (from tensorflow)
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12.5/site-packages/networkx-2.8.8-py3.12.egg (from tensorflow)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12.5/site-packages/jinja2-3.1.2-py3.12.egg (from tensorflow)
Requirement already satisfied: narwhals>=2.0 in /usr/local/lib/python3.12.5/site-packages/narwhals-1.17.0-py3.12.egg (from tensorflow)
Requirement already satisfied: alembic>=1.5.0 in /usr/local/lib/python3.12.5/site-packages/alembic-1.13.1-py3.12.egg (from tensorflow)
Requirement already satisfied: colorlog in /usr/local/lib/python3.12.5/site-packages/colorlog-6.8.2-py3.12.egg (from tensorflow)
Requirement already satisfied: sqlalchemy>=1.4.2 in /usr/local/lib/python3.12.5/site-packages/sqlalchemy-2.0.25-py3.12.egg (from tensorflow)
Requirement already satisfied: Mako in /usr/local/lib/python3.12.5/site-packages/Mako-1.3.10-py3.12.egg (from tensorflow)
Requirement already satisfied: aiohttp!=4.0.0a0,!4.0.0a1 in /usr/local/lib/python3.12.5/site-packages/aiohttp-3.9.5-py3.12.egg (from tensorflow)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12.5/site-packages/annotated_types-0.6.0-py3.12.egg (from tensorflow)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12.5/site-packages/pydantic_core-2.41.4-py3.12.egg (from tensorflow)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12.5/site-packages/typing_inspection-0.4.2-py3.12.egg (from tensorflow)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12.5/site-packages/six-1.17.0-py3.12.egg (from tensorflow)
Requirement already satisfied: greenlet>=1 in /usr/local/lib/python3.12.5/site-packages/greenlet-3.0.3-py3.12.egg (from tensorflow)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12.5/site-packages/mpmath-1.3.1-py3.12.egg (from tensorflow)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12.5/site-packages/MarkupSafe-2.1.5-py3.12.egg (from tensorflow)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.12.5/site-packages/attrs-23.2.0-py3.12.egg (from tensorflow)
Requirement already satisfied: jsonschema-specifications>=2023.03.0 in /usr/local/lib/python3.12.5/site-packages/jsonschema_specifications-2023.12.14-py3.12.egg (from tensorflow)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.12.5/site-packages/referencing-0.35.1-py3.12.egg (from tensorflow)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.12.5/site-packages/rpds_py-0.12.1-py3.12.egg (from tensorflow)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12.5/site-packages/charset_normalizer-3.3.0-py3.12.egg (from tensorflow)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12.5/site-packages/idna-3.10-py3.12.egg (from tensorflow)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12.5/site-packages/urllib3-2.2.1-py3.12.egg (from tensorflow)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12.5/site-packages/certifi-2024.7.4-py3.12.egg (from tensorflow)
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in /usr/local/lib/python3.12.5/site-packages/aiohappyeyeballs-2.5.0-py3.12.egg (from tensorflow)
Requirement already satisfied: aiosignal>=1.4.0 in /usr/local/lib/python3.12.5/site-packages/aiosignal-1.4.0-py3.12.egg (from tensorflow)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.12.5/site-packages/frozenlist-1.4.1-py3.12.egg (from tensorflow)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.12.5/site-packages/multidict-6.0.5-py3.12.egg (from tensorflow)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.12.5/site-packages/propcache-0.2.0-py3.12.egg (from tensorflow)

```

```

df_fc = df1[['ds', 'y']].copy()
df_fc['unique_id'] = 'series_1'

```

```

H = len(test_df)

train_fc = df_fc.iloc[:H]
test_fc = df_fc.iloc[H:]

```

```
df_fc.dtypes
```

```
      0
ds      datetime64[ns]
y      float64
unique_id  object
dtype: object
```

```

from neuralforecast import NeuralForecast
from neuralforecast.models import NHITS

model = NHITS(
    h=H,
    input_size=36,
    max_steps=300
)

fcst = NeuralForecast(
    models=[model],
    freq='M'
)

fcst.fit(train_fc)
forecast = fcst.predict()

```

```

INFO:lightning_fabric.utilities.seed:Seed set to 1
INFO:pytorch_lightning.utilities.rank_zero:GPU available: False, use
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, use

```

	Name	Type	Params	Mode	FLOPs
0	loss	MAE	0	train	0
1	padder_train	ConstantPad1d	0	train	0
2	scaler	TemporalNorm	0	train	0
3	blocks	ModuleList	2.5 M	train	0

Trainable params: 2.5 M

Non-trainable params: 0

Total params: 2.5 M

Total estimated model params size (MB): 9

Modules in train mode: 34

Modules in eval mode: 0

Total FLOPs: 0

Epoch 299/-2  1/1 0:00:00 • 0:00:00

```

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `m

```

```

INFO:pytorch_lightning.utilities.rank_zero:GPU available: False, use

```

```

INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, use

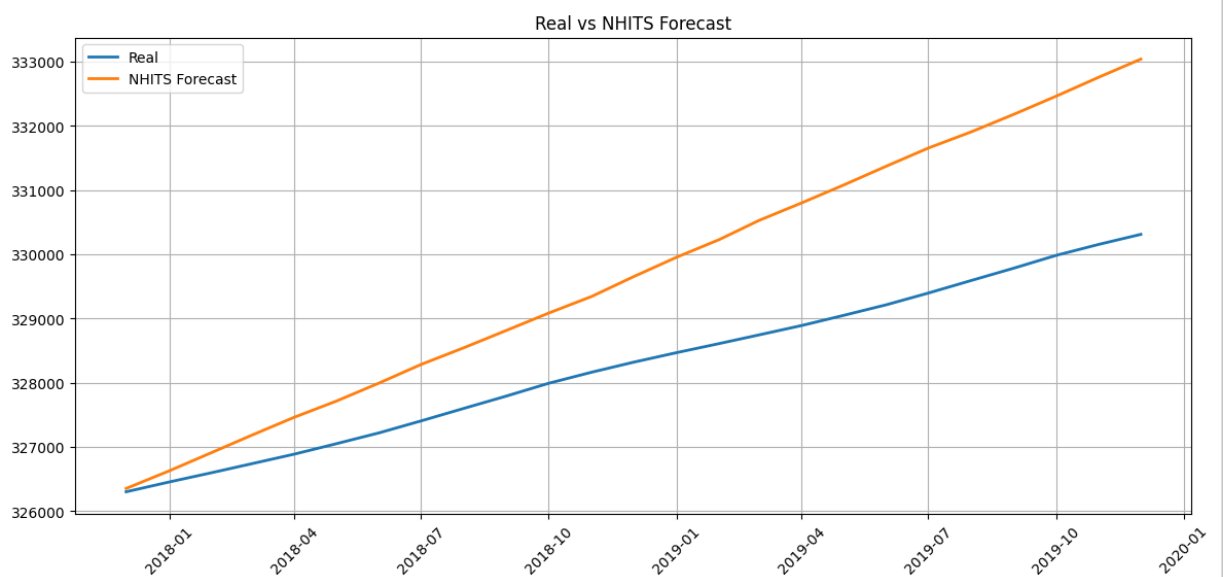
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(14,6))

plt.plot(test_fc['ds'], test_fc['y'], label='Real', linewidth=2)
plt.plot(test_fc['ds'], forecast['NHITS'], label='NHITS Forecast', lw=2)

plt.title("Real vs NHITS Forecast")
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.show()
```



0

