```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from sklearn import preprocessing, linear_model, model_selection, metrics
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
import warnings
from sklearn.feature_selection import SelectKBest, mutual_info_regression
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor  # Added
from sklearn.metrics import r2_score, mean_squared_error
from scipy.stats import norm
from scipy import stats
```

NOTE: Conclusion is mentioned in last

Without Cross Validation

Insights after Performing Data Science Life Cycle on Life Expectancy Dataset Dataset Without cv

- HISTORGRAM PLOT

- In Historgram plot we can see that, This distribution suggests that the majority of your data is concentrated in one range or or multiple range: Normalized Loss: Sharp peak on the left side, indicating high frequency of low values. X-axis ranges from 0 to 100,000. Y-axis ranges from 0 to 600.
- Life Expectancy Histogram: Data distributed between 0 and about 100 on the x-axis. Y-axis ranges from 0 to about 100.
- Adult molarity Histogram: Cyan-colored bars with a sharp peak on the left side (similar to alcohal but smaller scale). X-axis ranges from 0 to about 25. Y-axis ranges up to approximately 500.
- Infant death Histogram: Cyan-colored bars showing another sharp initial some peak that declines rapidly. X-axis ranging from around 0 to 1700. Y-axis peaking at just over 1600.

- BOX PLOT

- In boxplot we can see that there are few outlier in our dataset in columns percent of expenditure and instant death and others
- For removing oultier i use IQR method and after remove we can see new boxplot does contain few amount of oultiers.

- Model Accuracy without cross valiation

  I implemented three Regression model linear, KNN and random forest and overall we can see that random forest is very good

  Linear Regression Results: R-squared: 0.7489039180158683 Mean Squared Error: 0.2862529432844452

  KNN Regression Results: R-squared: 0.8392975221199117 Mean Squared Error: 0.18320300708310472

  Random Forest Regression Results: R-squared: 0.905235897625214 Mean Squared Error: 0.10803236358026995 overall overmodel accuracy are very good R-square represent that over mdel accuracy is very good and mean square error. showing over model are very good for prediction

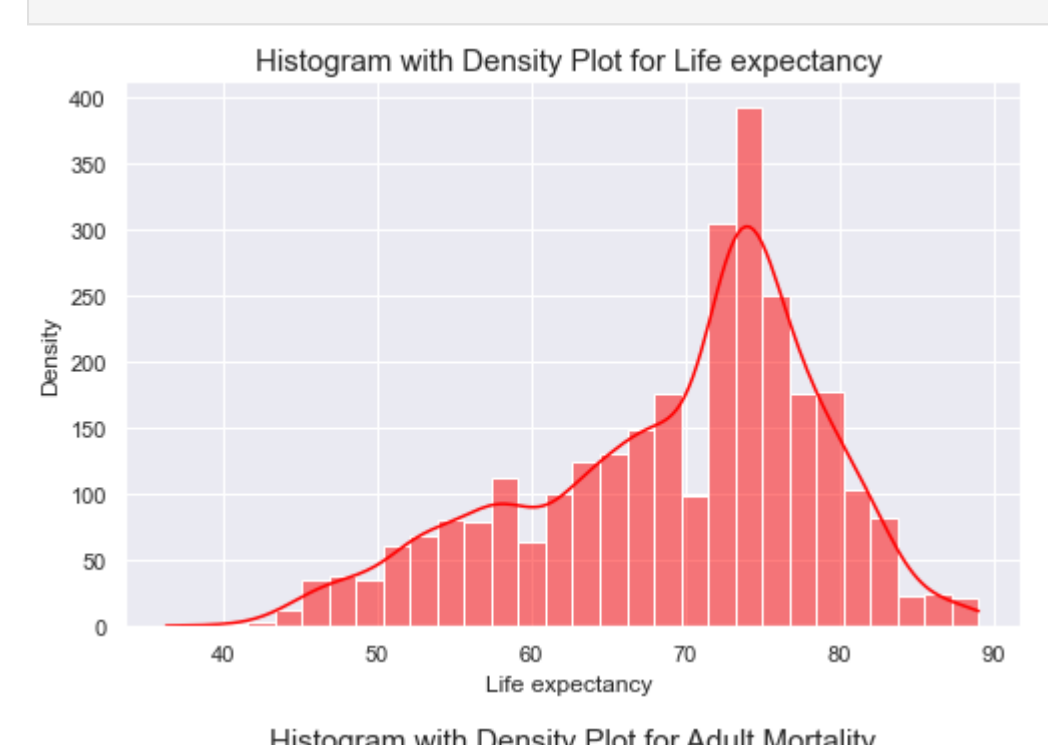
- Regression Plot

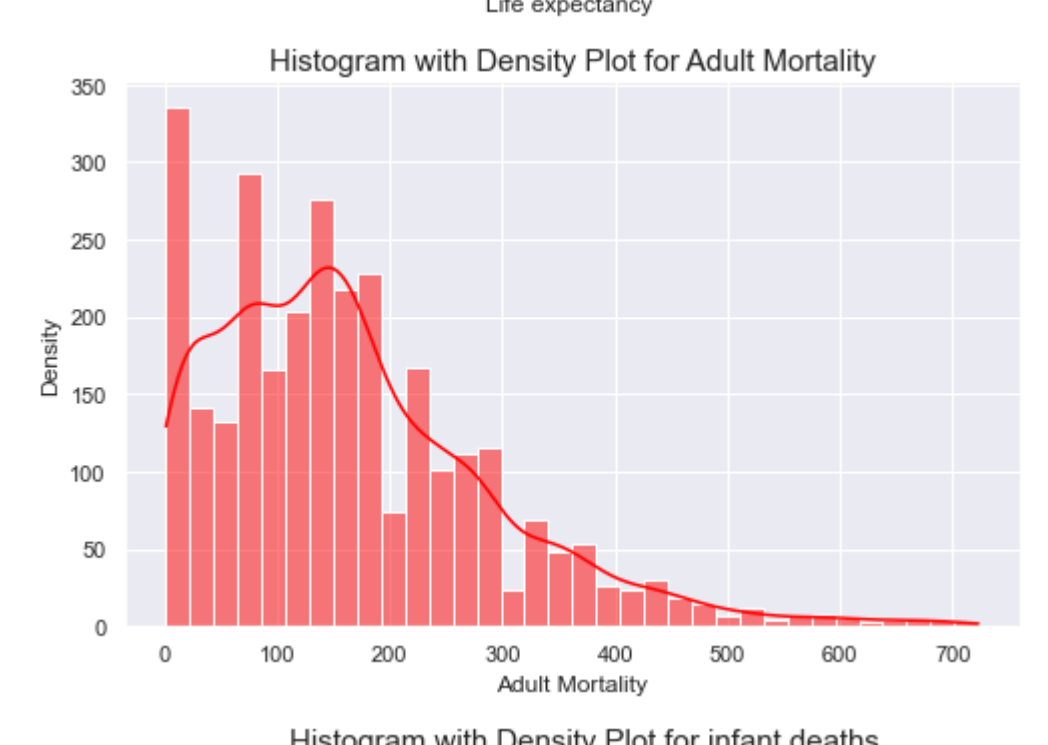
In our regression plot we can see that our model prediction are very good for all three models

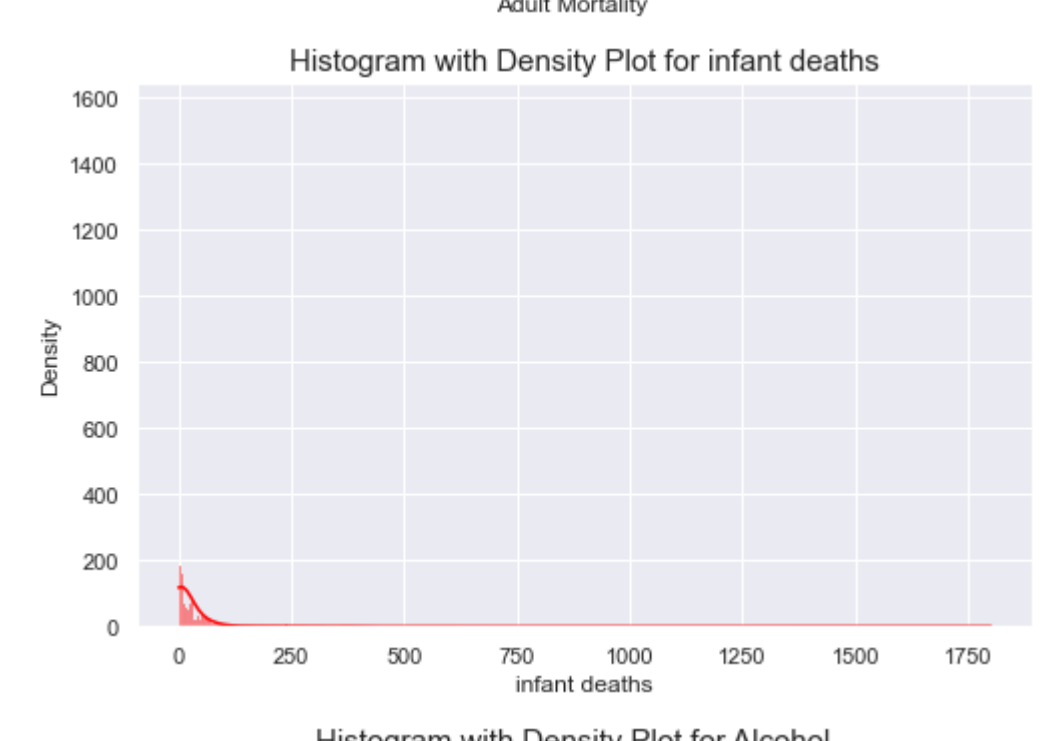
- OLS Regression Model

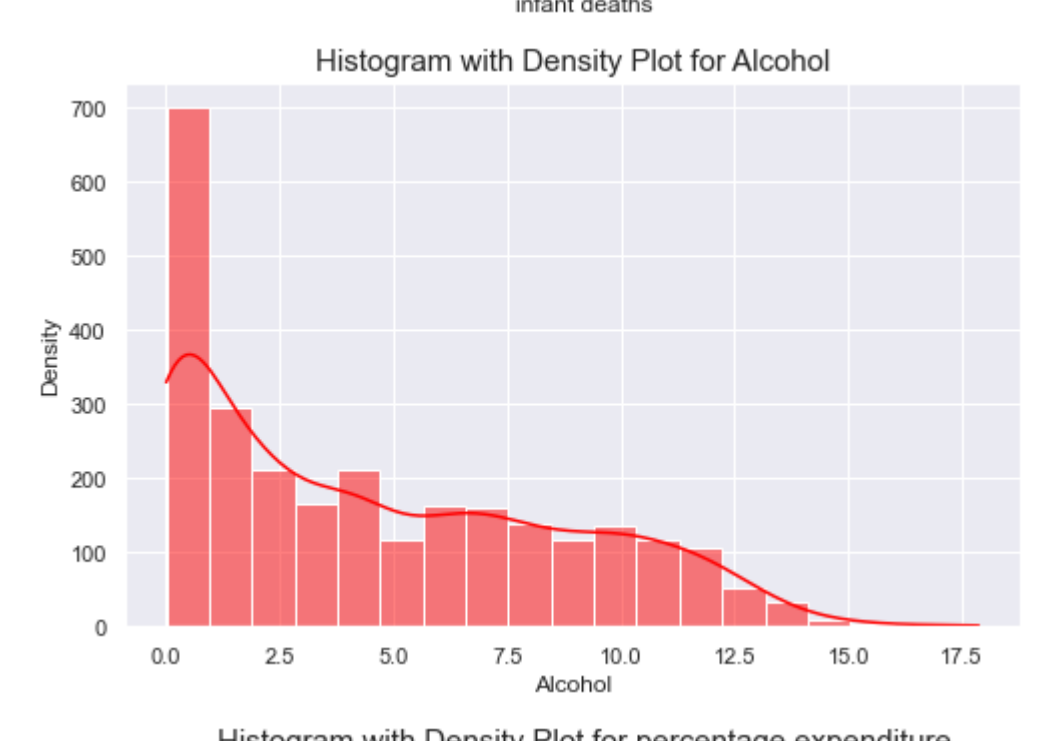
The model predicts life expectancy (dependent variable) based on country, year, and 18 other factors. The model explains 77.5% of the variation in life expectancy. There is a statistically significant relationship between the variables (very low p-value). However, some factors like country and adult mortality have high p-values, so their influence on life expectancy is less clear. Year, infant deaths, hepatitis B, and several other factors significantly influence life expectancy (very low p-values)

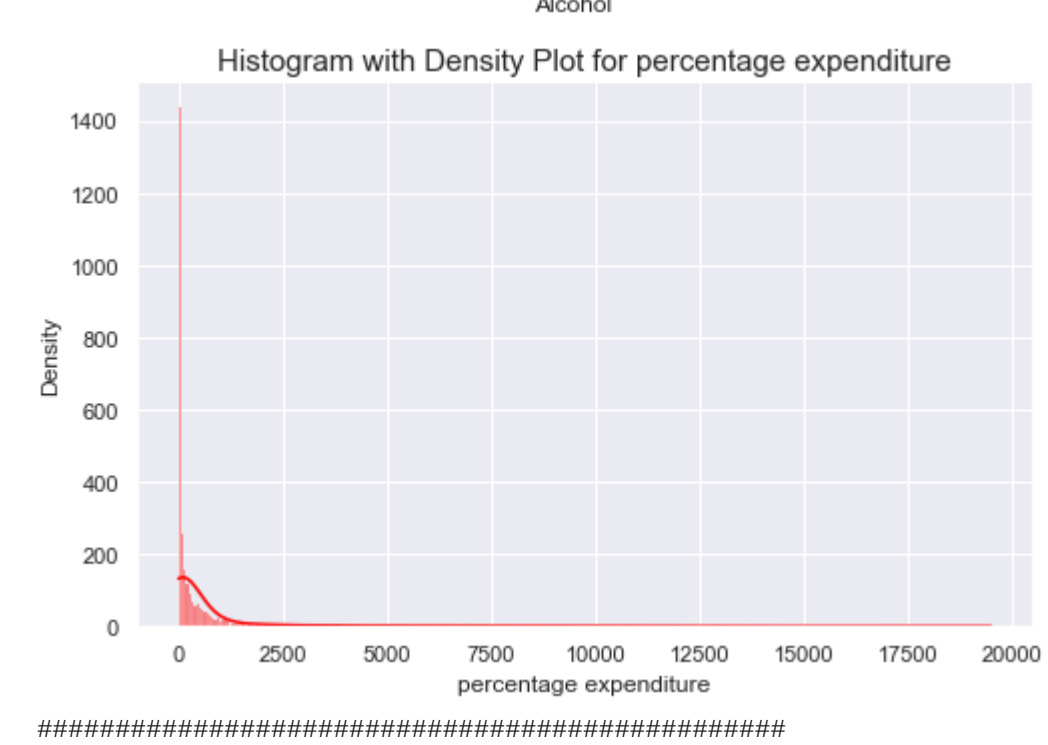
```python
def regression_analysis(target_column, file_path):
    # Step 1: Load dataset
    df = pd.read_csv(file_path)
    numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
    for column in numeric_columns[1:6]:
        plt.figure(figsize=(8, 5))
        sns.histplot(df[column], kde=True, color='red')
        plt.title(f'Histogram with Density Plot for {column}', fontsize=15)
        plt.xlabel(column, fontsize=12)
        plt.ylabel('Density', fontsize=12)
        plt.grid(True)
        plt.show()
    print('################################################\n\n')
    print('perfrom eda using boxplot for each column \n\n')
    print('################################################\n\n')

    for column in numeric_columns[1:6]:
        plt.figure(figsize=(8, 5))
        sns.boxplot(x=df[column], color='skyblue')
        plt.title(f'Box Plot for {column}', fontsize=15)
        plt.xlabel(column, fontsize=12)
        plt.grid(True)
        plt.show()
    missing_values = df.isnull().sum()
    print("Missing Values:")
    print(missing_values)
    df.fillna(df.mean(), inplace=True)
    Q1 = df.quantile(0.25)
    Q3 = df.quantile(0.75)
    IQR = Q3 - Q1
    df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]
    categorical_cols = list(df.select_dtypes(include=['object']).columns)
    for col in categorical_cols:
        df[col] = LabelEncoder().fit_transform(df[col])
    scaler = StandardScaler()
    df_scaled = scaler.fit_transform(df)
    df_scaled = pd.DataFrame(df_scaled, columns=df.columns)

    X = df_scaled.drop(columns=[target_column])
    y = df_scaled[target_column]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=21)
    lr_model = LinearRegression()
    knn_model = KNeighborsRegressor()
    rf_model = RandomForestRegressor(n_estimators=100, random_state=42)  # Added

    lr_model.fit(X_train, y_train)
    knn_model.fit(X_train, y_train)
    rf_model.fit(X_train, y_train)
    lr_predictions = lr_model.predict(X_test)
    knn_predictions = knn_model.predict(X_test)
    rf_predictions = rf_model.predict(X_test)

    print("Linear Regression Results:")
    print("R-squared:", r2_score(y_test, lr_predictions))
    print("Mean Squared Error:", mean_squared_error(y_test, lr_predictions))

    print("\nKNN Regression Results:")
    print("R-squared:", r2_score(y_test, knn_predictions))
    print("Mean Squared Error:", mean_squared_error(y_test, knn_predictions))

    print("\nRandom Forest Regression Results:")
    print("R-squared:", r2_score(y_test, rf_predictions))
    print("Mean Squared Error:", mean_squared_error(y_test, rf_predictions))


    plt.figure(figsize=(10,6))

    plt.scatter(y_test, lr_predictions, color='blue', label='Linear Regression')
    lr_line = np.polyfit(y_test, lr_predictions, 1)
    plt.plot(y_test, lr_line[0] * y_test + lr_line[1], color='blue')

    plt.scatter(y_test, knn_predictions, color='red', label='KNN Regression')
    knn_line = np.polyfit(y_test, knn_predictions, 1)
    plt.plot(y_test, knn_line[0] * y_test + knn_line[1], color='red')

    plt.scatter(y_test, rf_predictions, color='green', label='Random Forest Regression')
    rf_line = np.polyfit(y_test, rf_predictions, 1)
    plt.plot(y_test, rf_line[0] * y_test + rf_line[1], color='green')

    plt.xlabel('True Values')
    plt.ylabel('Predictions')
    plt.title('Regression Predictions vs True Values')
    plt.legend()

    X_train_ols = sm.add_constant(X_test)
    ols_model = sm.OLS(y_test, X_train_ols).fit()
    print(ols_model.summary())



regression_analysis('Life expectancy ', r'C:\Users\Hamza\Desktop\Regression.Life.Expectancy.csv')
```



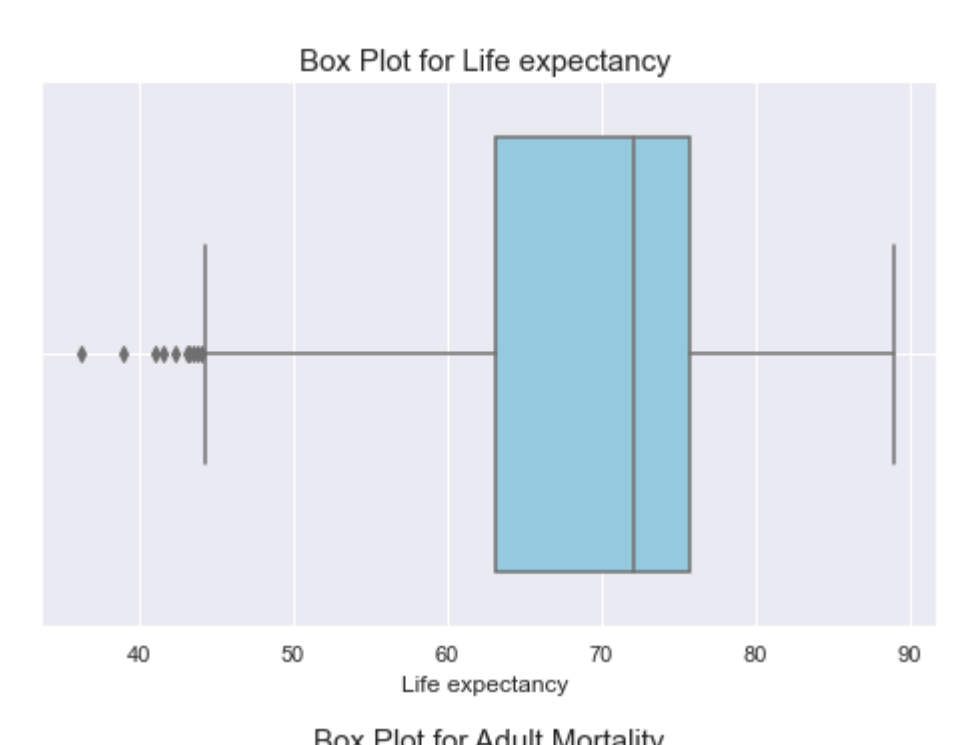


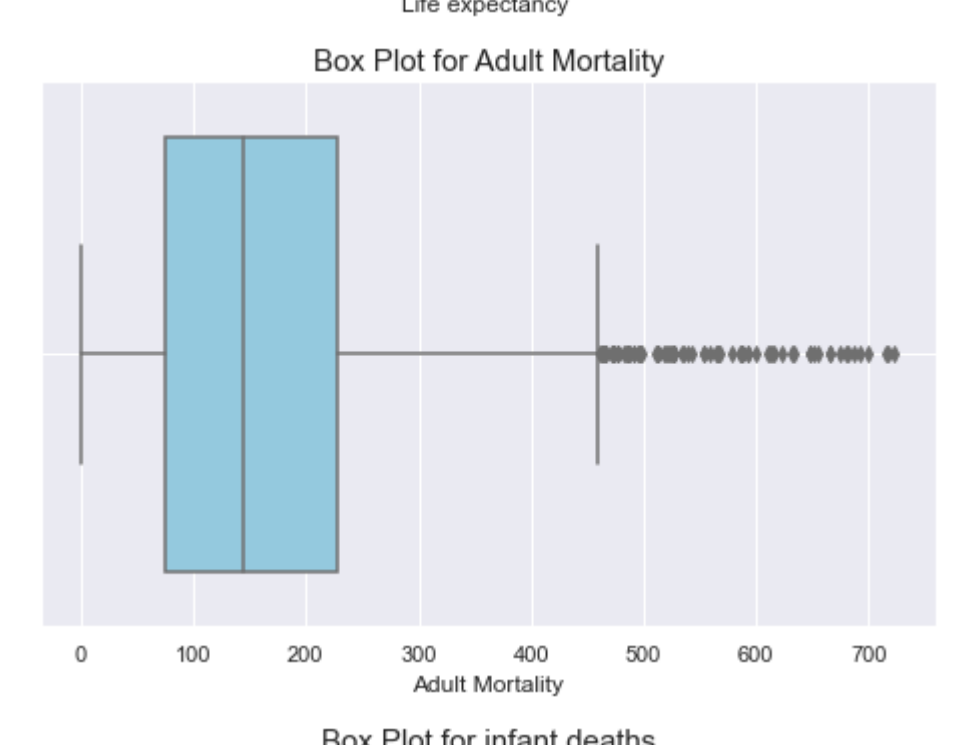


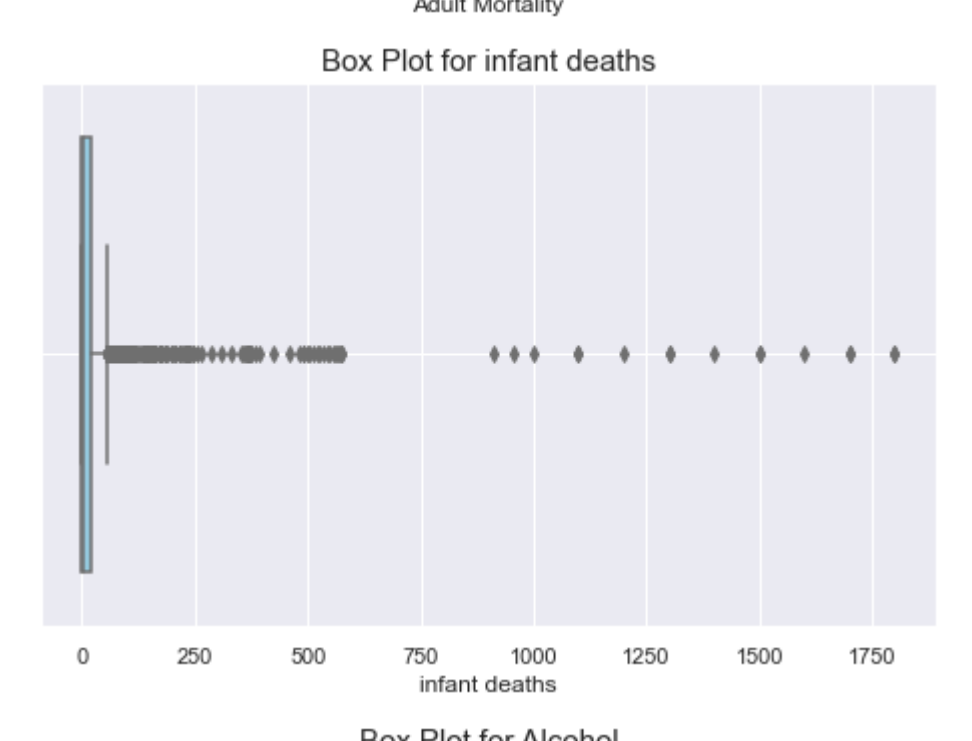


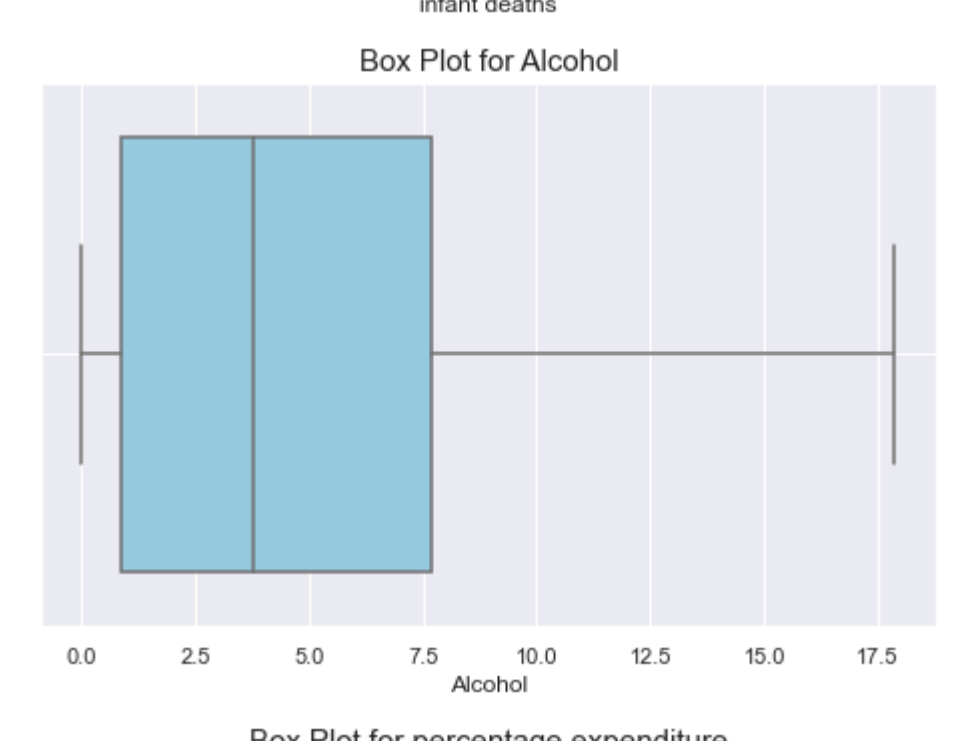



```
################################################

perfrom eda using boxplot for each column

################################################
```










```
Missing Values:
Country                            0
Year                               0
Status                             0
Life expectancy                   10
Adult Mortality                   10
infant deaths                      0
Alcohol                          194
percentage expenditure             0
Hepatitis B                      553
Measles                            0
 BMI                              34
under-five deaths                  0
Polio                             19
Total expenditure                226
Diphtheria                        19
 HIV/AIDS                          0
GDP                              448
Population                       652
 thinness  1-19 years             34
 thinness 5-9 years               34
Income composition of resources  167
Schooling                        163
dtype: int64
Linear Regression Results:
R-squared: 0.7489039180158683
Mean Squared Error: 0.2862529432844452

KNN Regression Results:
R-squared: 0.8392975221199117
Mean Squared Error: 0.18320300708310472

Random Forest Regression Results:
R-squared: 0.905235897625214
Mean Squared Error: 0.10803236358026995
                            OLS Regression Results
==============================================================================
Dep. Variable:       Life expectancy    R-squared:                       0.775
Model:                           OLS    Adj. R-squared:                  0.754
Method:                Least Squares    F-statistic:                     35.69
Date:               Tue, 26 Mar 2024    Prob (F-statistic):           1.13e-58
Time:                       19:39:37    Log-Likelihood:                -176.29
No. Observations:                239    AIC:                             396.6
Df Residuals:                    217    BIC:                             473.1
Df Model:                         21
Covariance Type:           nonrobust
===================================================================================================
                                      coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
const                               0.0246      0.037      0.662      0.509      -0.049       0.098
Country                            -0.0002      0.036     -0.005      0.996      -0.071       0.070
Year                                0.0241      0.040      0.598      0.551      -0.055       0.103
Status                             -0.0853      0.051     -1.687      0.093      -0.185       0.014
Adult Mortality                    -0.2972      0.045     -6.615      0.000      -0.386      -0.209
infant deaths                       0.7109      0.285      2.498      0.013       0.150       1.272
Alcohol                            -0.0149      0.054     -0.278      0.781      -0.121       0.091
percentage expenditure              0.0183      0.042      0.437      0.663      -0.064       0.101
Hepatitis B                        -0.1296      0.064     -2.016      0.045      -0.256      -0.003
Measles                            -0.0458      0.032     -1.418      0.158      -0.109       0.018
 BMI                               -0.0919      0.042     -2.190      0.030      -0.175      -0.009
under-five deaths                  -0.7685      0.292     -2.631      0.009      -1.344      -0.193
Polio                              -0.0754      0.144     -0.525      0.600      -0.358       0.208
Total expenditure                   0.1092      0.038      2.850      0.005       0.034       0.185
Diphtheria                          0.1391      0.167      0.832      0.406      -0.190       0.469
 HIV/AIDS                          -0.0541      0.039     -1.371      0.172      -0.132       0.024
GDP                                -0.0156      0.046     -0.340      0.734      -0.106       0.075
Population                          0.0487      0.042      1.151      0.251      -0.035       0.132
 thinness  1-19 years               0.8657      0.700      1.236      0.218      -0.514       2.246
 thinness 5-9 years                -0.9306      0.698     -1.333      0.184      -2.307       0.446
Income composition of resources     0.6528      0.087      7.497      0.000       0.481       0.824
Schooling                          -0.1165      0.075     -1.557      0.121      -0.264       0.031
==============================================================================
Omnibus:                       12.847   Durbin-Watson:                   2.108
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               30.519
Skew:                           0.098   Prob(JB):                     2.36e-07
Kurtosis:                       4.740   Cond. No.                         67.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

With Cross Validation

Insights after Performing Data Science Life Cycle on Life Expectancy Dataset Dataset with cross validataion

- Model Accuracy with cross valiation

  I implemented three Regression model linear, KNN and random forest and overall we can see that random forest is very good

  Linear Regression Cross-Validation R-squared: 0.6459528928385387 KNN Regression Cross-Validation R-squared: 0.6159441219329578 Random Forest Regression Cross-Validation R-squared: 0.7591699247994296

  Linear Regression Results: R-squared: 0.7489039180158683 Mean Squared Error: 0.2862529432844452

  KNN Regression Results: R-squared: 0.8392975221199117 Mean Squared Error: 0.18320300708310472

  Random Forest Regression Results: R-squared: 0.905235897625214 Mean Squared Error: 0.10803236358026995 overall overmodel accuracy are very good R-square represent that over mdel accuracy is very good and mean square error. showing over model are very good for prediction and not a big impact of cross validation

- Regression Plot

In our regression plot we can see that our model prediction are very good for all three models

- OLS Regression Model

The model predicts life expectancy (dependent variable) for people in different countries. Factors like adult mortality, under-five deaths, and BMI seem to significantly affect life expectancy (low p-values). However, the influence of factors like infant deaths and expenditure on health is less clear (higher p-values). Overall, the model explains 77.5% of the variation in life expectancy. There is a statistically significant relationship between the variables (very low p-value).

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_squared_error
from scipy.stats import norm
from scipy import stats
import statsmodels.api as sm

def regression_analysis(target_column, file_path):
    # Step 1: Load dataset
    df = pd.read_csv(file_path)
    missing_values = df.isnull().sum()
    print("Missing Values:")
    print(missing_values)
    df.fillna(df.mean(), inplace=True)

    Q1 = df.quantile(0.25)
    Q3 = df.quantile(0.75)
    IQR = Q3 - Q1
    df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]
    categorical_cols = list(df.select_dtypes(include=['object']).columns)
    for col in categorical_cols:
        df[col] = LabelEncoder().fit_transform(df[col])

    scaler = StandardScaler()
    df_scaled = scaler.fit_transform(df)
    df_scaled = pd.DataFrame(df_scaled, columns=df.columns)

    X = df_scaled.drop(columns=[target_column])
    y = df_scaled[target_column]
    lr_model = LinearRegression()
    knn_model = KNeighborsRegressor()
    rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

    lr_scores = cross_val_score(lr_model, X, y, cv=5, scoring='r2')
    knn_scores = cross_val_score(knn_model, X, y, cv=5, scoring='r2')
    rf_scores = cross_val_score(rf_model, X, y, cv=5, scoring='r2')

    print("Linear Regression Cross-Validation R-squared:", lr_scores.mean())
    print("KNN Regression Cross-Validation R-squared:", knn_scores.mean())
    print("Random Forest Regression Cross-Validation R-squared:", rf_scores.mean())

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=21)

    lr_model.fit(X_train, y_train)
    knn_model.fit(X_train, y_train)
    rf_model.fit(X_train, y_train)

    lr_predictions = lr_model.predict(X_test)
    knn_predictions = knn_model.predict(X_test)
    rf_predictions = rf_model.predict(X_test)

    print("\nLinear Regression Results:")
    print("R-squared:", r2_score(y_test, lr_predictions))
    print("Mean Squared Error:", mean_squared_error(y_test, lr_predictions))

    print("\nKNN Regression Results:")
    print("R-squared:", r2_score(y_test, knn_predictions))
    print("Mean Squared Error:", mean_squared_error(y_test, knn_predictions))

    print("\nRandom Forest Regression Results:")
    print("R-squared:", r2_score(y_test, rf_predictions))
    print("Mean Squared Error:", mean_squared_error(y_test, rf_predictions))


    plt.figure(figsize=(10,6))

    plt.scatter(y_test, lr_predictions, color='blue', label='Linear Regression')
    lr_line = np.polyfit(y_test, lr_predictions, 1)
    plt.plot(y_test, lr_line[0] * y_test + lr_line[1], color='blue')

    plt.scatter(y_test, knn_predictions, color='red', label='KNN Regression')
    knn_line = np.polyfit(y_test, knn_predictions, 1)
    plt.plot(y_test, knn_line[0] * y_test + knn_line[1], color='red')

    plt.scatter(y_test, rf_predictions, color='green', label='Random Forest Regression')
    rf_line = np.polyfit(y_test, rf_predictions, 1)
    plt.plot(y_test, rf_line[0] * y_test + rf_line[1], color='green')
    plt.show()

    X_train_ols = sm.add_constant(X_test)
    ols_model = sm.OLS(y_test, X_train_ols).fit()
    print(ols_model.summary())

regression_analysis('Life expectancy ', r'C:\Users\Hamza\Desktop\Regression.Life.Expectancy.csv')
```

```
Missing Values:
Country                            0
Year                               0
Status                             0
Life expectancy                   10
Adult Mortality                   10
infant deaths                      0
Alcohol                          194
percentage expenditure             0
Hepatitis B                      553
Measles                            0
 BMI                              34
under-five deaths                  0
Polio                             19
Total expenditure                226
Diphtheria                        19
 HIV/AIDS                          0
GDP                              448
Population                       652
 thinness  1-19 years             34
 thinness 5-9 years               34
Income composition of resources  167
Schooling                        163
dtype: int64
Linear Regression Cross-Validation R-squared: 0.6459528928385387
KNN Regression Cross-Validation R-squared: 0.6159441219329578
Random Forest Regression Cross-Validation R-squared: 0.7591699247994296

Linear Regression Results:
R-squared: 0.7489039180158683
Mean Squared Error: 0.2862529432844452

KNN Regression Results:
R-squared: 0.8392975221199117
Mean Squared Error: 0.18320300708310472

Random Forest Regression Results:
R-squared: 0.905235897625214
Mean Squared Error: 0.10803236358026995
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Life expectancy    R-squared:                       0.775
Model:                           OLS    Adj. R-squared:                  0.754
Method:                Least Squares    F-statistic:                     35.69
Date:               Tue, 26 Mar 2024    Prob (F-statistic):           1.13e-58
Time:                       19:39:59    Log-Likelihood:                -176.29
No. Observations:                239    AIC:                             396.6
Df Residuals:                    217    BIC:                             473.1
Df Model:                         21
Covariance Type:           nonrobust
===================================================================================================
                                      coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
const                               0.0246      0.037      0.662      0.509      -0.049       0.098
Country                            -0.0002      0.036     -0.005      0.996      -0.071       0.070
Year                                0.0241      0.040      0.598      0.551      -0.055       0.103
Status                             -0.0853      0.051     -1.687      0.093      -0.185       0.014
Adult Mortality                    -0.2972      0.045     -6.615      0.000      -0.386      -0.209
infant deaths                       0.7109      0.285      2.498      0.013       0.150       1.272
Alcohol                            -0.0149      0.054     -0.278      0.781      -0.121       0.091
percentage expenditure              0.0183      0.042      0.437      0.663      -0.064       0.101
Hepatitis B                        -0.1296      0.064     -2.016      0.045      -0.256      -0.003
Measles                            -0.0458      0.032     -1.418      0.158      -0.109       0.018
 BMI                               -0.0919      0.042     -2.190      0.030      -0.175      -0.009
under-five deaths                  -0.7685      0.292     -2.631      0.009      -1.344      -0.193
Polio                              -0.0754      0.144     -0.525      0.600      -0.358       0.208
Total expenditure                   0.1092      0.038      2.850      0.005       0.034       0.185
Diphtheria                          0.1391      0.167      0.832      0.406      -0.190       0.469
 HIV/AIDS                          -0.0541      0.039     -1.371      0.172      -0.132       0.024
GDP                                -0.0156      0.046     -0.340      0.734      -0.106       0.075
Population                          0.0487      0.042      1.151      0.251      -0.035       0.132
 thinness  1-19 years               0.8657      0.700      1.236      0.218      -0.514       2.246
 thinness 5-9 years                -0.9306      0.698     -1.333      0.184      -2.307       0.446
Income composition of resources     0.6528      0.087      7.497      0.000       0.481       0.824
Schooling                          -0.1165      0.075     -1.557      0.121      -0.264       0.031
==============================================================================
Omnibus:                       12.847   Durbin-Watson:                   2.108
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               30.519
Skew:                           0.098   Prob(JB):                     2.36e-07
Kurtosis:                       4.740   Cond. No.                         67.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Conclusion

We are implemented regression model with different ML model by using without and with cross validataion but one thing that we

can see the there are not a big accuracy different in both strategy , becuase cross validatoin is commonly use with hyper paramenter tuning , in our case we are not implement with hyper paramenter,that why may be we are not get a large accuracy difference but in our all case we can see that random forest regression provide use very good accuracy R-squared: 0.905235897625214 Mean Squared Error: 0.10803236358026995 and if we see knn regressor this is also good model for us R-squared: 0.8392975221199117 Mean Squared Error: 0.18320300708310472 and if we see linear regression accuracy R-squared: 0.7489039180158683 Mean Squared Error: 0.2862529432844452 but at end we can see random forest and knn have good accuracy