Electricity Price Prediction using RandomForestRegressor

import pandas as pd
import numpy as np
data = pd.read_csv(r"C:\Users\Admin\Downloads\electricity.csv")
data.head()

C:\Users\Admin\AppData\Local\Temp\ipykernel_11804\1934404862.py:3: DtypeWarning: Columns (9,10,11,14,15,16,17) have mixed types. Specify dtype option on import or set low _memory=False.

data = pd.read_csv(r"C:\Users\Admin\Downloads\electricity.csv")

Out[2]:		DateTime	Holiday	HolidayFlag	DayOfWeek	WeekOfYear	Day	Month	Year	PeriodOfDay	Fore
	0	01/11/2011 00:00	NaN	0	1	44	1	11	2011	0	
	1	01/11/2011 00:30	NaN	0	1	44	1	11	2011	1	
	2	01/11/2011 01:00	NaN	0	1	44	1	11	2011	2	
	3	01/11/2011 01:30	NaN	0	1	44	1	11	2011	3	
	4	01/11/2011 02:00	NaN	0	1	44	1	11	2011	4	
	_										

In [3]: # show all the columns of this dataset
 data.info()

Out[4]:

In [6]:

```
<class 'pandas.core.frame.DataFrame'>
        RangeIndex: 38014 entries, 0 to 38013
        Data columns (total 18 columns):
         #
             Column
                                     Non-Null Count Dtype
             -----
                                     -----
         0
             DateTime
                                     38014 non-null object
         1
             Holiday
                                                     object
                                     1536 non-null
         2
             HolidayFlag
                                     38014 non-null int64
         3
             DayOfWeek
                                     38014 non-null int64
         4
             WeekOfYear
                                     38014 non-null int64
         5
             Day
                                     38014 non-null int64
         6
             Month
                                     38014 non-null int64
         7
             Year
                                     38014 non-null int64
         8
             PeriodOfDay
                                     38014 non-null int64
         9
             ForecastWindProduction 38014 non-null object
         10 SystemLoadEA
                                     38014 non-null object
         11 SMPEA
                                     38014 non-null object
         12 ORKTemperature
                                     38014 non-null object
         13 ORKWindspeed
                                     38014 non-null object
         14 CO2Intensity
                                     38014 non-null object
         15 ActualWindProduction
                                     38014 non-null object
         16 SystemLoadEP2
                                     38014 non-null object
         17 SMPEP2
                                     38014 non-null object
        dtypes: int64(7), object(11)
        memory usage: 5.2+ MB
In [4]:
        # to check the null values
        data.isnull().sum()
        DateTime
                                      0
                                  36478
        Holiday
        HolidayFlag
                                      0
        DayOfWeek
                                      0
        WeekOfYear
                                      0
                                      0
        Day
        Month
                                      0
                                      0
        Year
        PeriodOfDay
                                      0
        ForecastWindProduction
                                      0
        SystemLoadEA
                                      0
        SMPEA
                                      0
        ORKTemperature
                                      0
        ORKWindspeed
                                      0
        CO2Intensity
                                      0
                                      0
        ActualWindProduction
        SystemLoadEP2
                                      0
        SMPEP2
                                      0
        dtype: int64
In [5]: # fill the missing values
        data['Holiday'].fillna(data['Holiday'].mode()[0],inplace=True)
```

data.isnull().sum()

```
0
         DateTime
Out[6]:
         Holiday
                                    0
         HolidayFlag
                                    0
         DayOfWeek
                                    0
         WeekOfYear
                                    0
         Day
                                    0
         Month
                                    0
         Year
                                    0
         PeriodOfDay
                                    0
         ForecastWindProduction
                                    0
         SystemLoadEA
         SMPEA
                                    0
         ORKTemperature
                                    0
         ORKWindspeed
                                    0
         CO2Intensity
                                    0
         ActualWindProduction
                                    0
         SystemLoadEP2
                                    0
         SMPEP2
                                    0
         dtype: int64
         # to drop all these rows containing null values from the dataset
In [7]:
         # data = data.dropna()
         # # converting categories it number --- Label Encoder
In [8]:
         from sklearn.preprocessing import LabelEncoder
         label encoder = LabelEncoder()
In [10]:
         data['DateTime'] = label encoder.fit transform(data['DateTime'])
         data['Holiday'] = label_encoder.fit_transform(data['Holiday'])
         # Convert all values in 'ForecastWindProduction' column to strings
         data['ForecastWindProduction'] = data['ForecastWindProduction'].astype(str)
         data['ForecastWindProduction'] = label_encoder.fit_transform(data['ForecastWindProduct')
         data['SystemLoadEA'] = data['SystemLoadEA'].astype(str)
In [13]:
         data['SystemLoadEA'] = label_encoder.fit_transform(data['SystemLoadEA'])
         data['SMPEA'] = data['SMPEA'].astype(str)
         data['SMPEA'] = label_encoder.fit_transform(data['SMPEA'])
         data['ORKTemperature'] = data['ORKTemperature'].astype(str)
         data['ORKTemperature'] = label_encoder.fit_transform(data['ORKTemperature'])
In [14]:
         data['ORKWindspeed'] = data['ORKWindspeed'].astype(str)
         data['ORKWindspeed'] = label encoder.fit transform(data['ORKWindspeed'])
         data['CO2Intensity'] = data['CO2Intensity'].astype(str)
         data['CO2Intensity'] = label_encoder.fit_transform(data['CO2Intensity'])
         data['ActualWindProduction'] = data['ActualWindProduction'].astype(str)
         data['ActualWindProduction'] = label_encoder.fit_transform(data['ActualWindProduction']
```

```
In [15]:
              data['SystemLoadEP2'] = data['SystemLoadEP2'].astype(str)
              data['SystemLoadEP2'] = label_encoder.fit_transform(data['SystemLoadEP2'])
              data['SMPEP2'] = data['SMPEP2'].astype(str)
              data['SMPEP2'] = label encoder.fit transform(data['SMPEP2'])
             # to know the correlation between all the columns in the dataset
In [16]:
              import seaborn as sns
              import matplotlib.pyplot as plt
              correlations = data.corr(method='pearson')
              plt.figure(figsize=(16, 12))
              sns.heatmap(correlations, cmap="coolwarm", annot=True)
              plt.show()
                                    0.066 0.046 -0.0042 0.093
                                               -0.14 -0.025 0.066 0.022 -0.019 -1.3e-05 -0.035 -0.03 -0.011 0.059 -0.008 -0.068 0.0081 -0.065 -0.008
                        Holiday - 0.066
                                               -0.12 -0.0033 0.046 0.032 -0.023 -1.6e-05 -0.043 -0.044 -0.0085 0.075 0.02 -0.065 -0.011 -0.089 -0.013
                     HolidayFlag -
                               0.046
                                                                                                                                       0.8
                               0.0042 -0.14 -0.12
                     DavOfWeek -
                               0.093 -0.025 -0.0033 0.0075
                                                                    -0.24 -8.3e-05 -0.025 -0.00066 -0.014 -0.13 0.031 -0.1 0.0014 -0.044 -0.007
                                    0.066 0.046 -0.0044 0.061
                                                                                                                                       0.6
                                                                        -8.7e-05 -0.022 -0.0043 -0.014 -0.12 0.03 -0.11 0.0037 -0.045 -0.0014
                         Month -
                                                          0.0018 -0.24
                           Year -0.0073 -0.019 -0.023 -0.0012 -0.24
                                                                         .9e-05 0.0028 -0.06 0.08
                                                                                             -0.041 0.016 -0.19 -0.026 -0.056 0.039
                                                                                        0.28
             ForecastWindProduction - -0.045 -0.035 -0.043 -0.048 -0.025 -0.045 -0.022 0.0028 -0.016
                                                                                   -0.0018 0.0086 -0.019 -0.099 0.065 0.44
                    SystemLoadEA - -0.014 -0.03 -0.044 -0.11 -0.00066 -0.013 -0.0043 -0.06 -0.25 -0.0018
                                                                                              0.14 0.032 0.026 0.02 0.0081 -0.16
                               -0.014 -0.011 -0.0085 -0.051 -0.014 -0.014 -0.014 0.08 0.28 0.0086
                                                                                                                  0.32 0.36
                                                                                                                                       - 0.2
                  ORKTemperature - -0.038 | 0.059 | 0.075 | 0.014 | -0.13 | -0.035 | -0.12 | -0.041 | -0.018 | -0.019 | 0.14
                                                                                                    0.04 0.006 0.0056 0.12 -0.054
                               0.018 -0.008 0.02 0.0089 0.031 0.017 0.03 0.016 -0.032 -0.099 0.032 -0.035 0.04
                                                                                                         0.022 -0.097 -0.03 -0.034
                     CO2Intensity - 0.084 0.068 0.065 0.03 0.1 0.08 0.11 0.19 0.16 0.065 0.026 0.052 0.006 0.022
                                                                                                              .00082 -0.092 0.036
               ActualWindProduction - -0.043 0.0081 -0.011 -0.03 0.0014 -0.042 0.0037 -0.026 -0.029
                                                                              0.44
                                                                                        -0.024 0.0056 -0.097-0.0008
                                                                         0.58
                                                                              6.8e-05 0.0081 0.32 0.12
                        SMPEP2 - -0.046 -0.0082 -0.013 -0.057 -0.0074 -0.046 -0.0014 0.039
                                                                         0.2
                                                                              0.036 -0.16
                                                                                         0.36
                                                                               ForecastWindProduction
                                                                                                               ActualWindProduction
In [17]:
             # Assign the values of X and y
              x = data[["Day", "Month", "ForecastWindProduction", "SystemLoadEA",
                             "SMPEA", "ORKTemperature", "ORKWindspeed", "CO2Intensity",
                            "ActualWindProduction", "SystemLoadEP2"]]
              y = data["SMPEP2"]
In [18]: # split the data set to test set and train set
              from sklearn.model selection import train test split
              xtrain, xtest, ytrain, ytest = train test split(x, y,
                                                                                    test size=0.2,
                                                                                    random state=42)
```

```
Electricity Price Prediction with Machine Learning
In [19]:
          # model selection
          from sklearn.ensemble import RandomForestRegressor
          model = RandomForestRegressor()
          model.fit(xtrain, ytrain)
Out[19]:
          ▼ RandomForestRegressor
          RandomForestRegressor()
          y_pred = model.predict(xtest)
In [22]:
          y_pred
          array([5221.63, 3960.59, 4406.46, ..., 1720.7 , 4768.07, 3495.71])
Out[22]:
In [25]:
          import pandas as pd
          # Assuming ytest and y pred are your lists or arrays of actual and predicted values
          df = pd.DataFrame({'Actual': ytest, 'Predicted': y_pred})
          df
Out[25]:
                 Actual Predicted
          35833
                  5109
                          5221.63
            198
                  3752
                          3960.59
          36547
                  4911
                          4406.46
          26373
                  3728
                          3715.99
          21156
                  5050
                          4394.12
          13927
                  3708
                          3941.14
```

7603 rows × 2 columns

5883

471

7420

3541

5729.91

1720.70

4768.07

3495.71

16926

24520

9059

9786

```
# Calculate evaluation metrics
In [35]:
         from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
         mae = mean_absolute_error(ytest, y_pred)
         mse = mean_squared_error(ytest, y_pred)
         rmse = np.sqrt(mse)
         r2 = r2_score(ytest, y_pred)
         print("Mean Absolute Error:", mae)
         print("Mean Squared Error:", mse)
         print("Root Mean Squared Error:", rmse)
         print("R-squared:", r2)
```

Mean Absolute Error: 752.619826384322 Mean Squared Error: 1417762.0598960149 Root Mean Squared Error: 1190.6981397046084

R-squared: 0.5660115459731367

In []: