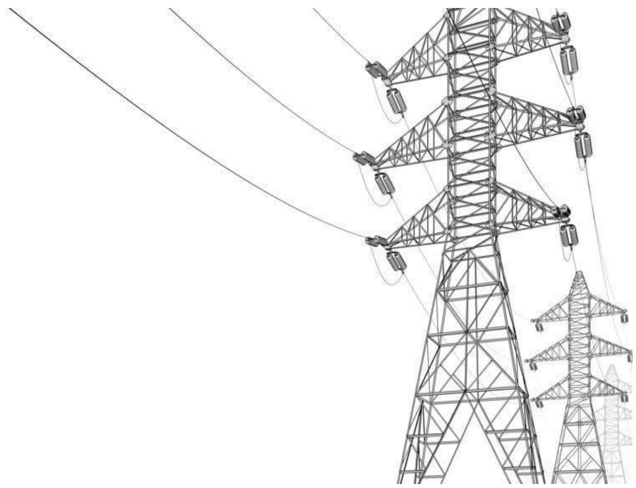
ELECTRICITY PRICE PREDICTION USING MACHINE LEARNING

BATCH MEMBER

613021104015: CHAITHANYA N

Phase 3 submission document

Project title: Electricity Price Prediction



INTRODUTION:

- The electricity market is a complex, evolutionary, and dynamic environment.
- Forecasting electricity prices is an important issue for all electricity market participants.
- In this study, we shed light on how to improve electricity price forecasting accuracy through the use of a machine learning technique—namely, a novel genetic programming approach.
- Drawing on empirical data from the largest EU energy markets, we propose a forecasting model that considers variables related to weather conditions, oil prices, and CO_2 coupons and predicts energy prices 24 hours ahead.

- We show that the proposed model provides more accurate predictions of future electricity prices than existing prediction methods.

- Our important findings will assist the electricity market participants in forecasting future price movements.

GIVEN DATASET

26	828.09	5018.8	56.25	12	20.4	513.98	683
27	873.81	4916.93	56.25	11	24.1	518.96	711
28	920.69	4933.87	56.25	12	22.2	525.69	761
29	985.09	4978.87	56.25	11	25.9	528.47	750
30	1044.37	5013.1	56.25	11	22.2	528.17	758
31	1098.97	5061.1	56.25	11	24.1	513.22	805
32	1148.24	5137.37	56.25	11	27.8	509.16	782
33	1191.7	5325.37	56.25	11	27.8	495.57	861
34	1229.56	5628.84	160.62	11	31.5	479.82	1014
35	1281.93	5887.54	247.04	11	29.6	460.51	1065
36	1328.42	5871.14	196.43	11	33.3	450.42	1138
37	1370.76	5727.71	61.89	11	35.2	460.87	1191
38	1407.5	5532.86	56.25	12	35.2	463.02	1230
39	1438.24	5401.07	56.25	12	37	457.03	1279
40	1464.71	5182.14	49.66	12	37	457.09	1278

NECESSARY STEPS TO FOLLOW

1.Import libraries:

Program:

```
import pandas as pd

import numpy as np
```

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

2.Load the Dataset:

Load your dataset into a Pandas DataFrame. You can typically find electricity price datasets in CSV format, but you can adapt this code to other formats as needed.

```
df = pd.read_csv(' C:\Electricity.csv')
```

3.Exploratory Data Analysis (EDA):

Perform EDA to understand your data better. This includes checking for missing values, exploring the data's statistics, and visualizing it to identify patterns.

```
# Check for missing values
print(df.isnull().sum())
```

Program:

```
Pd.read()
```

Program:

```
# Explore statistics
print(df.describe())
# Visualize the data (e.g., histograms, scatter plots, etc.)
```

4.Feature Engineering:

Depending on your dataset, you may need to create new features or transform existing ones. This can involve one-hot encoding categorical variables, handling date/time data, or scaling numerical features.

Program:

df = pd.get_dummies(df, columns=[' SystemLoadEA, SMPEA’])

5. Split the Data:

Split your dataset into training and testing sets. This helps you evaluate

your model's performance later. X = df.drop('price', axis=1) # Features

y = df['price'] # Target variable

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

6.Feature Scaling:

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 and std=1) is a common choice.

Program:

scaler = StandardScaler()

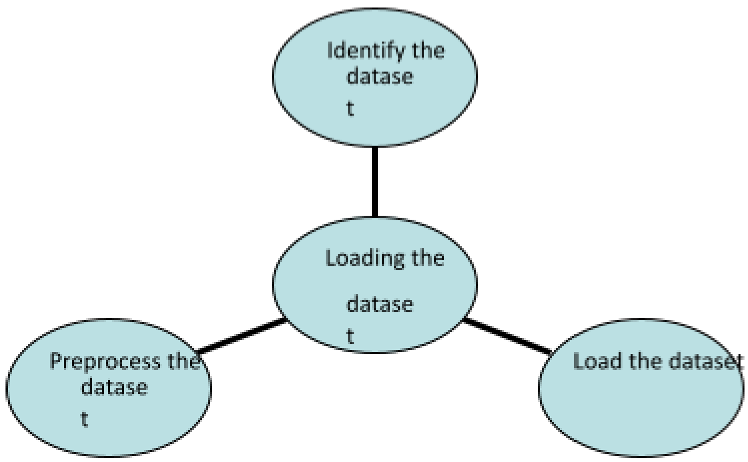
X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)

7.Importance of loading and processing dataset:

Loading and preprocessing the dataset is an important first step in building any machine learning model. However, it is especially important for electricity price prediction models, as electricity price datasets are often complex and noisy.

By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.



Program:

```
import numpy as np
import pandas as pd


import os

for dirname, _, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))

df = pd.read_csv("/kaggle/input/electrity-prices/electricity_prices.csv")
```

```
df.head()

df.info()

df.shape

import missingno


missingno.bar(df)

cols = ["ForecastWindProduction", "SystemLoadEA", "SMPEA", "ORKTemperature",
"ORKWindspeed",

        "CO2Intensity", "ActualWindProduction", "SystemLoadEP2", "SMPEP2"]


for col in cols:

    df[col] = pd.to_numeric(df[col], errors="coerce")

df.isnull().sum()


df = df.dropna()

import matplotlib.pyplot as plt

import seaborn as sns


plt.figure(figsize=(15, 15))

sns.heatmap(df.corr(method="pearson"), annot=True, fmt=".0f", cbar=False)

plt.show()

df.corr()[["SMPEP2"]].sort_values("SMPEP2", ascending=False)

df = df.drop(["Year", "HolidayFlag", "ORKTemperature", "Day", "Month", "WeekOfYear",

      "CO2Intensity", "ORKWindspeed", "DayOfWeek", "ForecastWindProduction",

      "ActualWindProduction", "Holiday", "DateTime"], axis=1)

X = df.drop("SMPEP2", axis=1)

y = df["SMPEP2"]

from sklearn.model_selection import train_test_split


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

In [14]:

linkcode

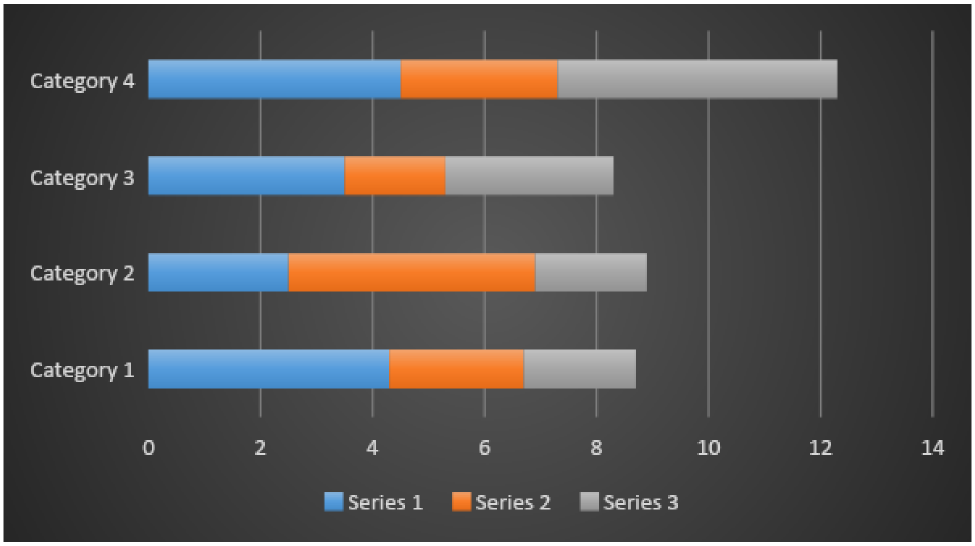
```
from sklearn.ensemble import RandomForestRegressor

rf = RandomForestRegressor()
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
rf.score(X_test, y_test)
import matplotlib.pyplot

plt.figure()
plt.scatter(y_test, y_pred)
plt.scatter(y_test, y_test)
plt.plot(y_test, y_test)
 plt.legend(["Predicted", "Original", "Regression Line"])
plt.show()
```

Visualisation and Pre-Processing of Data:

In [1]: `sns.histplot(dataset, x='Price', bins=50, color='y')` Out[1]:



In [2]: sns.boxplot(dataset, x='Price',

palette='Blues')

Out[2]:

<Axes: xlabel='Price'>

Out[2]:

<seaborn.axisgrid.PairGrid at 0x7caf0c2ac550>