IBM NAAN MUDHALVAN

ELECTRICITY PRICES

PREDICTION

|  |  |
| --- | --- |
| DOMAIN | APPLIED DATA SCIENCE |
| PROJECT TOPIC | ELECTRICITY PRICES PREDICTION |
| TEAM MEMBERS&  REGISTER NUMBER | ARAVINDHATCHAN S[ 420421104007]  ANBUMANI P[ 420421104006 ]  VIGNESHWAR T [ 420421104086]  KOLLURU LAKSHMI MANI [ 420421104037 ] |

PHASE\_3 : SUBMISSION DOCUMENT

Introduction:

* **Predicting electricity prices is a common task in energy economics, finance, and energy management. Accurate predictions can help energy companies, consumers, and policymakers make informed decisions. Here's a high-level overview of the steps involved in predicting electricity prices**

Import Libraries:

* Numpy
* Pandas
* Matplotlib
* Seaborn

1. **Dataset Loading and Exploration**:
   * Import necessary libraries.
   * Load the dataset.
   * Explore the dataset's structure and basic statistics.
   * Identify missing values and outliers, if any.
2. **Data Preprocessing**:
   * Handle missing values (impute or remove data as necessary).
   * Address outliers using appropriate techniques.
   * Convert categorical features into numerical representations (if applicable).
   * Standardize or normalize numerical features.
3. **Data Analysis**:
   * Explore the distribution of electricity prices.
   * Analyze temporal patterns (e.g., hourly, daily, monthly trends).
   * Examine the correlation between variables.
   * Visualize the data to gain insights.
4. **Conclusion:**

* Summarize the findings, discuss the model's performance, and provide recommendations or future work.

Data Set Link:

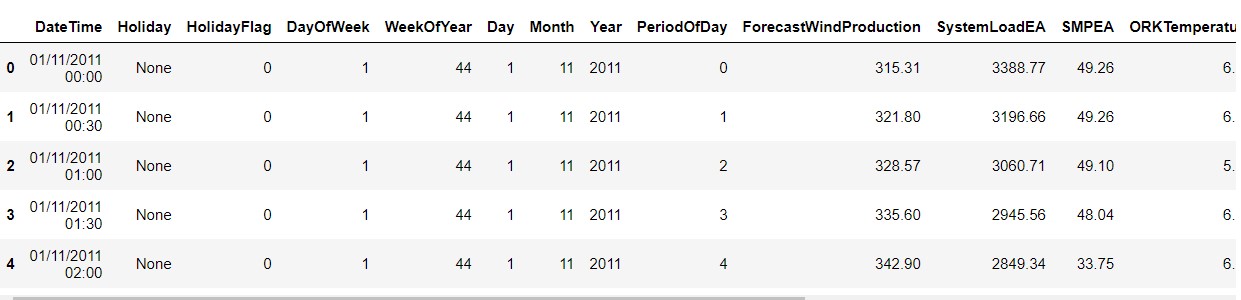
[**https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction**](https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction) 

**PROGRAM:**

**READING FILE:**

|  |
| --- |
| df=pd.read\_csv("Electricity.csv", low\_memory=False)  df.head() |

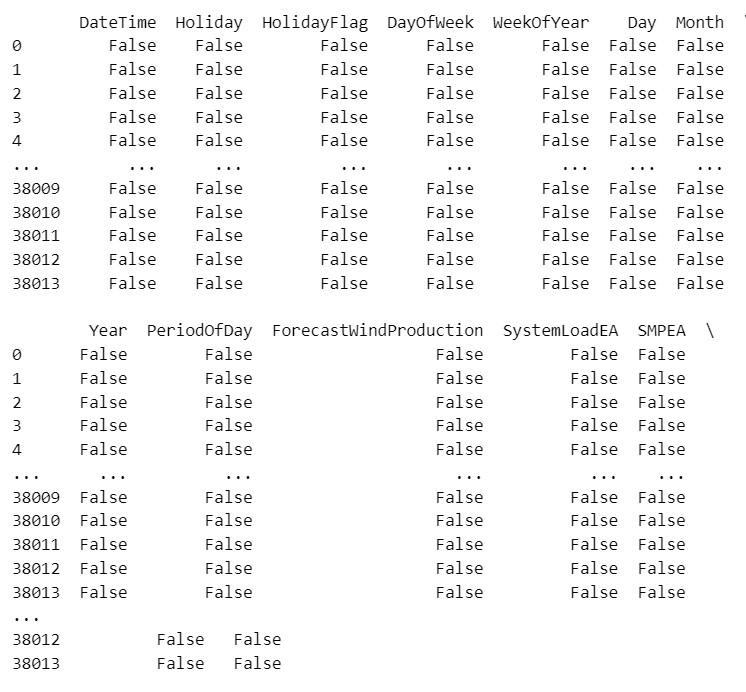
**OUTPUT:**

****

**HANDLING DATASET:**

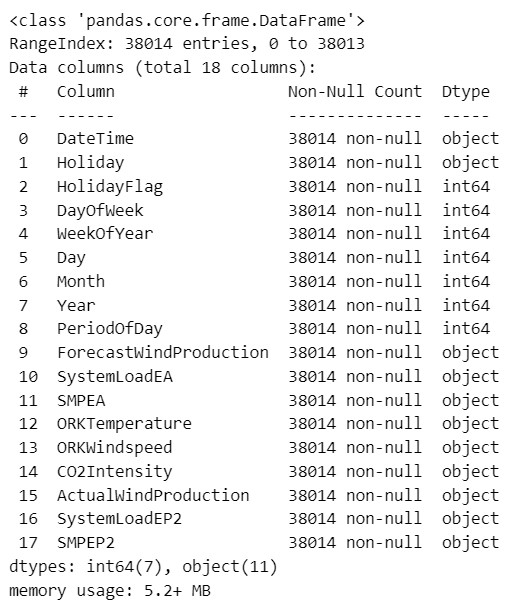
|  |
| --- |
| ##Missing Data  import pandas as pd  # Load your data into a DataFrame  df = pd.read\_csv("Electricity.csv")  # Check for missing data  missing\_data = df.isnull()  print(missing\_data)  # Remove rows with missing data  df.dropna(axis=0, inplace=True)  # Remove columns with missing data  df.dropna(axis=1, inplace=True) |

**OUTPUT:**

****

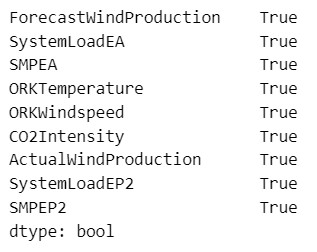
|  |
| --- |
| df.info() |

**OUTPUT:**

****

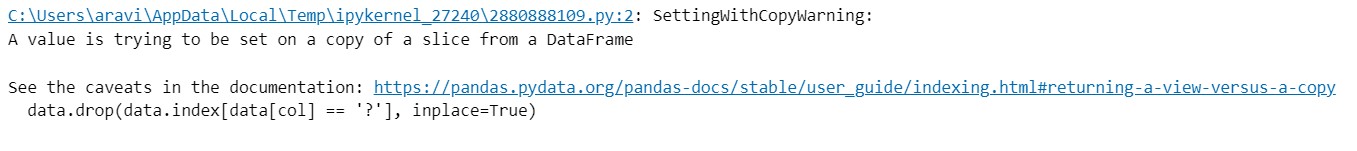
|  |
| --- |
| data.isin(['?']).any() |

**OUTPUT:**

****

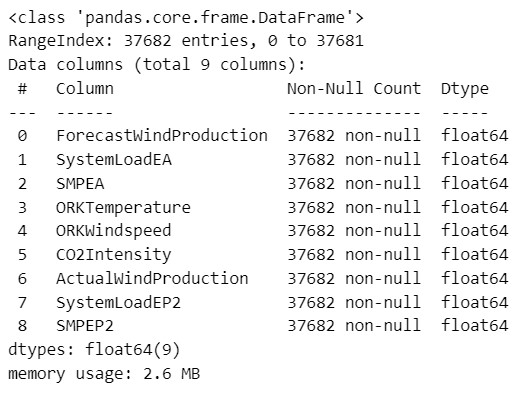
|  |
| --- |
| for col in data.columns:      data.drop(data.index[data[col] == '?'], inplace=True) |

**OUTPUT:**



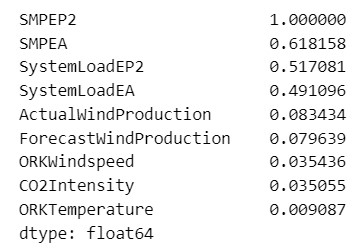
|  |
| --- |
| data=data.apply(pd.to\_numeric)  data=data.reset\_index()  data.drop('index', axis=1, inplace=True)  data.info() |

**OUTPUT:**



|  |
| --- |
| data.corrwith(data['SMPEP2']).abs().sort\_values(ascending=False) |

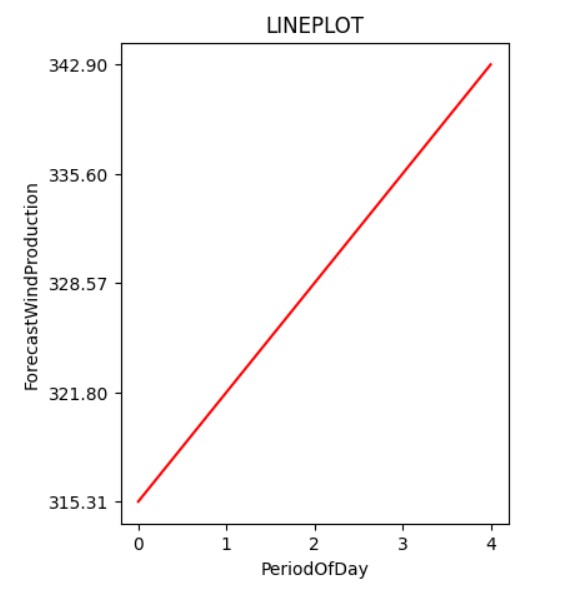
**OUTPUT:**



**VISUALIZATION OF DATASET:**

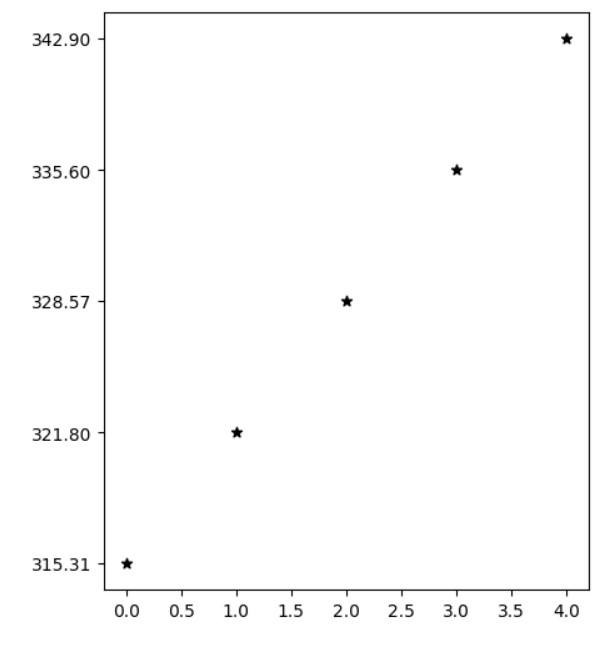
|  |
| --- |
| **##Lineplot**  a=df['PeriodOfDay'].head(5)  df1=df['ForecastWindProduction'].head(5)  fig = plt.figure(figsize =(8, 7))  plt.plot(a, df1,color='red')  plt.title("LINEPLOT")  plt.xlabel("PeriodOfDay")  plt.ylabel("ForecastWindProduction")  plt.show() |

**OUTPUT:**



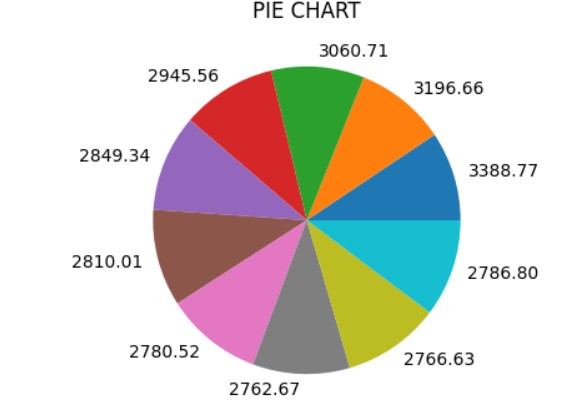
|  |
| --- |
| **##Scatterplot**  a=df['PeriodOfDay'].head()  df1=df['ForecastWindProduction'].head()  fig = plt.figure(figsize =(8, 7))  plt.scatter(a, df1,marker='\*',color='black')  plt.show("SCATTERPLOT")  plt.show() |

**OUTPUT:**



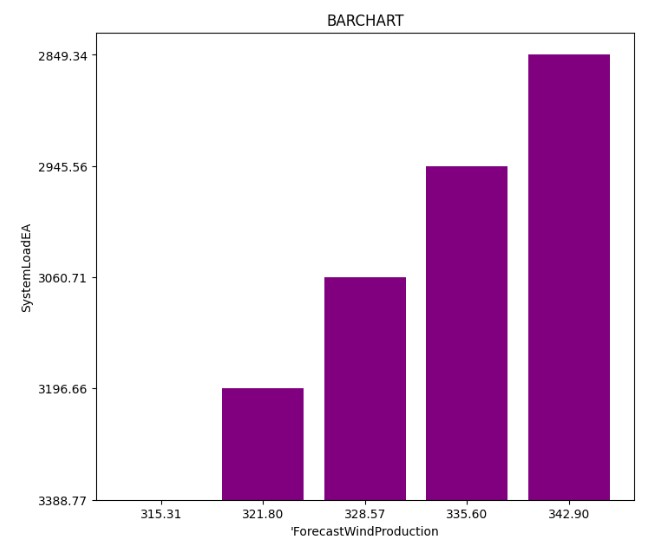
|  |
| --- |
| **#Pie Chart**  a=df['ForecastWindProduction'].head(10)  df1=df['SystemLoadEA'].head(10)  fig = plt.figure(figsize =(4,4))  plt.pie(a, labels= df1)  plt.title("PIE CHART")  plt.show() |

**OUTPUT:**



|  |
| --- |
| **#BAR PLOT**  a=df['ForecastWindProduction'].head(5)  df1=df['SystemLoadEA'].head(5)  fig = plt.figure(figsize =(8, 7))  plt.bar(a, df1,color='purple')  plt.title("BARCHART")  plt.xlabel("TV")  plt.ylabel("Radio")  plt.show() |

**OUTPUT:**



Conclusion:

The dataset has been preprocessed and visualization in the phase executed successfully.