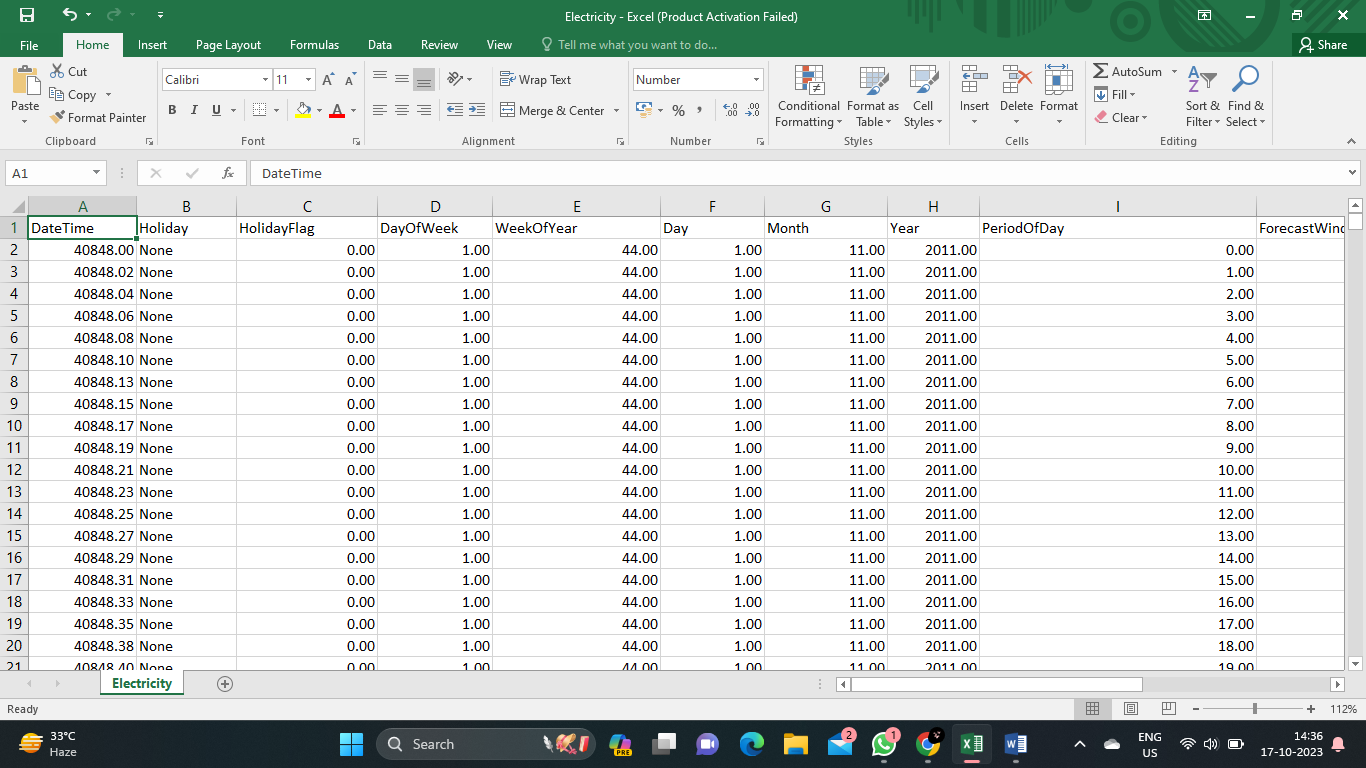
**TITLE: ELECTRICITY PRICE PREDICTION**

**INTRODUCTION:**

In this part of the project, the main goal is to build a prediction model for electricity prices. This report outlines the steps involved in loading and preprocessing the dataset to prepare it for analysis.

**DATASET DESCRIPTION:**

The dataset used for this project contains historical electricity prices. It includes various features such as date, time, and the corresponding electricity price for each observation.



**ALGORITHM:**

**1) Loading the Dataset:**

The first step in the development process is to load the historical electricity prices dataset. This can be done using a variety of tools and programming languages, such as Python and its data manipulation libraries like Pandas.

**2)** **Handling Missing Values:**

Missing values can be filled in or removed depending on the nature and significance of the missing data. This can be done using techniques such as mean imputation or interpolation.

**3) Converting Data Types:**

Sometimes, the dataset may contain columns with incorrect data types. For example, a column that represents dates may be stored as strings instead of datetime objects. In such cases, it is important to convert the data types to the appropriate format.

**4) Removing Outliers:**

Outliers are data points that deviate significantly from the rest of the dataset. These outliers can distort the analysis and prediction models. Removing outliers can be done by using statistical techniques such as z-score or interquartile range.

**5) Scaling and Normalization:**

Scaling and normalization are important preprocessing steps to ensure that all features are on a similar scale. This helps in preventing certain features from dominating the analysis or prediction models.

**PROGRAM:**

import pandas as pd**# Load the dataset**dataset = pd.read\_csv('/Electricity.csv')

**# Handle missing values**

dataset = dataset.fillna(method='ffill')  **# Forward fill missing values**print(dataset)

**OUTPUT:**

DateTime Holiday HolidayFlag DayOfWeek \0 1970-01-01 00:00:00.000040848 None 0.0 1.0 1 1970-01-01 00:00:00.000040848 None 0.0 1.0 2 1970-01-01 00:00:00.000040848 None 0.0 1.0 3 1970-01-01 00:00:00.000040848 None 0.0 1.0 4 1970-01-01 00:00:00.000040848 None 0.0 1.0 ... ... ... ... ... 38009 1970-01-01 00:00:00.000041639 New Year's Eve 1.0 1.0 38010 1970-01-01 00:00:00.000041639 New Year's Eve 1.0 1.0 38011 1970-01-01 00:00:00.000041639 New Year's Eve 1.0 1.0 38012 1970-01-01 00:00:00.000041639 New Year's Eve 1.0 1.0 38013 1970-01-01 00:00:00.000041639 New Year's Eve 1.0 1.0 WeekOfYear Day Month Year PeriodOfDay ForecastWindProduction \0 44.0 1.0 11.0 2011.0 -1.696450 315.31 1 44.0 1.0 11.0 2011.0 -1.624264 321.80 2 44.0 1.0 11.0 2011.0 -1.552078 328.57 3 44.0 1.0 11.0 2011.0 -1.479892 335.60 4 44.0 1.0 11.0 2011.0 -1.407706 342.90 ... ... ... ... ... ... ... 38009 1.0 31.0 12.0 2013.0 1.407547 1179.14 38010 1.0 31.0 12.0 2013.0 1.479733 1152.01 38011 1.0 31.0 12.0 2013.0 1.551919 1123.67 38012 1.0 31.0 12.0 2013.0 1.624105 1094.24 38013 1.0 31.0 12.0 2013.0 1.696290 1064.0 SystemLoadEA SMPEA ORKTemperature ORKWindspeed CO2Intensity \0 3388.77 49.26 6.00 9.30 600.71 1 3196.66 49.26 6.00 11.10 605.42 2 3060.71 49.10 5.00 11.10 589.97 3 2945.56 48.04 6.00 9.30 585.94 4 2849.34 33.75 6.00 11.10 571.52 ... ... ... ... ... ... 38009 3932.22 34.51 6.00 22.20 285.31 38010 3821.44 33.83 5.00 24.10 278.31 38011 3724.21 31.75 4.00 20.40 280.91 38012 3638.16 33.83 5.00 14.80 302.46 38013 3624.25 33.83 5.00 16.70 308.01 ActualWindProduction SystemLoadEP2 SMPEP2 0 356.00 3159.60 54.32 1 317.00 2973.01 54.23 2 311.00 2834.00 54.23 3 313.00 2725.99 53.47 4 346.00 2655.64 39.87 ... ... ... ... 38009 812.0 3692.95 42.45 38010 852.0 3571.0 33.83 38011 962.0 3460.29 31.75 38012 950.0 3563.99 50.6 38013 1020.0 3517.08 34.9 [38014 rows x 18 columns]

**# Convert data types**

dataset['DateTime'] = pd.to\_datetime(dataset['DateTime'])print(dataset['DateTime'])

**OUTPUT:**

0 1970-01-01 00:00:00.0000408481 1970-01-01 00:00:00.0000408482 1970-01-01 00:00:00.0000408483 1970-01-01 00:00:00.0000408484 1970-01-01 00:00:00.000040848 ... 38009 1970-01-01 00:00:00.00004163938010 1970-01-01 00:00:00.00004163938011 1970-01-01 00:00:00.00004163938012 1970-01-01 00:00:00.00004163938013 1970-01-01 00:00:00.000041639Name: DateTime, Length: 38014, dtype: datetime64[ns]

**#REMOVE OUTLIERS**

outliers = dataset['PeriodOfDay'].between(0, 47) # Define a range for valid pricesdataset = dataset[outliers]print(outliers)

**OUTPUT:**

24 True25 True26 True27 True28 True ... 38009 True38010 True38011 True38012 True38013 TrueName: PeriodOfDay, Length: 19008, dtype: bool

**#Scale and normalize feature**

dataset['PeriodOfDay'] = (dataset['PeriodOfDay'] - dataset['PeriodOfDay'].mean()) / dataset['PeriodOfDay'].std()print(dataset['PeriodOfDay'])

**OUTPUT:**

24 -1.66128125 -1.51682226 -1.37236327 -1.22790328 -1.083444 ... 38009 1.08344438010 1.22790338011 1.37236338012 1.51682238013 1.661281Name: PeriodOfDay, Length: 19008, dtype: float64

**CONCLUSION:**

In this part of the project, we have successfully loaded and preprocessed the historical electricity prices dataset. The dataset is now ready for further analysis and the development of a prediction model for electricity prices.