**Electricity Prices Prediction**

**Introduction:**

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

**Data:**

DateTime Holiday HolidayFlag DayOfWeek WeekOfYear \

0 1/11/2011 0:00 None 0 1 44

1 1/11/2011 0:30 None 0 1 44

2 1/11/2011 1:00 None 0 1 44

3 1/11/2011 1:30 None 0 1 44

4 1/11/2011 2:00 None 0 1 44

... ... ... ... ... ...

38009 31/12/2013 21:30 New Year's Eve 1 1 1

38010 31/12/2013 22:00 New Year's Eve 1 1 1

38011 31/12/2013 22:30 New Year's Eve 1 1 1

38012 31/12/2013 23:00 New Year's Eve 1 1 1

38013 31/12/2013 23:30 New Year's Eve 1 1 1

Day Month Year PeriodOfDay ForecastWindProduction SystemLoadEA \

0 1 11 2011 0 315.31 3388.77

1 1 11 2011 1 321.8 3196.66

2 1 11 2011 2 328.57 3060.71

3 1 11 2011 3 335.6 2945.56

4 1 11 2011 4 342.9 2849.34

... ... ... ... ... ... ...

38009 31 12 2013 43 1179.14 3932.22

38010 31 12 2013 44 1152.01 3821.44

38011 31 12 2013 45 1123.67 3724.21

38012 31 12 2013 46 1094.24 3638.16

38013 31 12 2013 47 1064.0 3624.25

SMPEA ORKTemperature ORKWindspeed CO2Intensity ActualWindProduction \

0 49.26 6 9.3 600.71 356

1 49.26 6 11.1 605.42 317

2 49.1 5 11.1 589.97 311

3 48.04 6 9.3 585.94 313

4 33.75 6 11.1 571.52 346

... ... ... ... ... ...

38009 34.51 6 22.2 285.31 812

38010 33.83 5 24.1 278.31 852

38011 31.75 4 20.4 280.91 962

38012 33.83 5 14.8 302.46 950

38013 33.83 5 16.7 308.01 1020

SystemLoadEP2 SMPEP2

0 3159.6 54.32

1 2973.01 54.23

2 2834 54.23

3 2725.99 53.47

4 2655.64 39.87

... ... ...

38009 3692.95 42.45

38010 3571.0 33.83

38011 3460.29 31.75

38012 3563.99 50.6

38013 3517.08 34.9

[38014 rows x 18 columns]

<ipython-input-8-c4c64ba6fca1>:2: 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(io.BytesIO(uploaded["Electricity.csv"]))

**Necessary step to follow:**

**1.Import Libraries:**

Start by importing the necessary 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 house price datasets in CSV format, but you can adapt this code to other formats as needed.

Program:

df = pd.read\_csv(' E:\Electricity prices prediction.csv ')

Pd.read()

**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.

Program:

# Check for missing values

print(df.isnull().sum())

# 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:

# Example: One-hot encoding for categorical variables

df = pd.get\_dummies(df, columns=[' Avg. Area Income ', ' Avg. AreaHouse Age '])

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 andstd=1) is a common choice.

Program:

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

Importance of loading and processing dataset:

Loading and processing a dataset is a crucial step in data analysis and machine learning. The importance of this step cannot be overstated, as it directly impacts the quality of insights or predictions that can be derived from the data. Here are some key reasons highlighting the importance of loading and processing datasets:

1. \*\*Data Quality Assurance:\*\* Loading a dataset involves inspecting and cleaning the data. This process helps identify and rectify errors, missing values, outliers, and inconsistencies in the data. Ensuring data quality is essential for reliable and accurate analysis or modeling.

2. \*\*Data Integration:\*\* In many real-world scenarios, data comes from various sources and formats. Loading and processing data allow for the integration of diverse datasets into a unified format. This is particularly important in data warehousing, business intelligence, and analytics applications.

3. \*\*Feature Engineering:\*\* Processing the dataset involves selecting relevant features (variables) and engineering new features that can improve model performance. Proper feature engineering can greatly enhance the effectiveness of machine learning algorithms.

4. \*\*Dimensionality Reduction:\*\* In cases where datasets have a large number of features, dimensionality reduction techniques can be applied during data processing to reduce the complexity of the data while preserving important information.

5. \*\*Normalization and Scaling:\*\* Normalizing or scaling data is essential for algorithms that are sensitive to the scale of features, such as distance-based methods and gradient-based optimization. It ensures that all features contribute equally to the analysis.

6. \*\*Data Exploration and Visualization:\*\* Loading and processing data enables exploratory data analysis (EDA). EDA is crucial for gaining an understanding of the data's characteristics, distributions, and relationships between variables. Visualization of processed data aids in identifying patterns and trends.

7. \*\*Preventing Data Leakage:\*\* When working with time-series data or datasets with temporal elements, proper handling of the data is necessary to prevent data leakage, which occurs when future information is inadvertently included in the training data.

8. \*\*Imbalanced Data Handling:\*\* In classification problems, datasets are often imbalanced, with one class significantly outnumbering the others. Processing data may involve techniques such as oversampling, undersampling, or generating synthetic data to address this imbalance.

9. \*\*Data Security and Privacy:\*\* Processing data may include anonymizing or obfuscating sensitive information to ensure data security and privacy, particularly when dealing with personal or confidential data.

10. \*\*Optimizing Performance:\*\* Loading and processing data can be tailored to specific machine learning algorithms, optimizing data structures and representations to improve model training and inference speed.

11. \*\*Model Interpretability:\*\* Properly processed data can make it easier to interpret and explain the results of machine learning models, which is critical for applications where model transparency is necessary, such as in healthcare or finance.

12. \*\*Error Handling and Robustness:\*\* Data processing can involve error-handling mechanisms to handle unexpected issues gracefully, ensuring the robustness of data pipelines and reducing the risk of failures during analysis.

In summary, loading and processing a dataset is a foundational step in data analysis and machine learning, serving as the basis for all subsequent tasks. Ensuring data quality, making data suitable for modeling, and addressing specific challenges associated with different datasets are vital for achieving meaningful and actionable insights from data. Proper data handling enhances the reliability, accuracy, and effectiveness of analyses and models.