**Libraries for Electricity price prediction**

import csv

from sklearn.model\_selection import train\_test\_split

Nmpy

Pandas

from sklearn.preprocessing import onehotEncoder

from sklearn.preprocessing import LabelEncoder

import matplotlib.pyplot

from sklearn.datasets import load\_iris

from sklearn.linear\_model import linearRegression

from sklearn.preprocessing import standard scaler

from sklearn.ensemble import randomforestclassifier

**Importing the Data Set:**

import csv

file = open('data.csv', 'rU')

reader = csv.reader(file)

matrix=[]

for row in reader:

matrix.append(row)

print(matrix)

**Handling the Missing Data:**

In this we use the sklearn.preprocessing import imputer it is used to fill the missing data.

The imputer is an estimator used to fill the missing values in datasets. For numerical values, it uses mean, median, and constant. For categorical values, it uses the most frequently used and constant value.

**Encode categorical Data:**

For Encoding the categorical data using the **one-hot encoding** and it is used to convert a categorical variables into a binary matrix.

This is commonly used in machine learning algorithms that cannot work with categorical data directly.

We can use libraries like scikit-learn or keras to perform one-hot encoding easily.

**Splitting the Data Set:**

In this we are using the **from sklearn.model\_selection import train\_test\_split** library for splitting the data set as the test data and the trained data from the data set.

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

The dataframe gets divided into X\_train,X\_test , y\_train and y\_test. **X\_train and y\_train** sets are used for training and fitting the model. The **X\_test and y\_test** sets are used for testing the model if it's predicting the right outputs/labels.

**Feature Scaling:**

Sklearn library offers us with **StandardScaler()** function to standardize the data values into a standard format.

The library is **from sklearn.preprocessing import standard scaler**.

**Syntax:**

object = StandardScaler()

object.fit\_transform(data)

**Csv File:**

