ELECTRICITY PRICES PREDICTION

Creating a predictive program for electricity prices in python typically involves using times Series forecasting techniques. Get started with the following steps using libraries like Pandas, NumPy, and Scikit-learn.

DATA COLLECTION:

Gather historical electricity price data. You can often find such Data form government sources, utility companies, or online database.

DATA PREPROCESSING:

Clean and pre-process the data, handling missing values and converting timestamp if needed.

FEATURE ENGINEERING:

Create relevant features like time of day of the week, holidays, etc., which can impact electricity Prices.

SPILT DATA:

Spilt your dataset into training and testing sets to evaluate your predictive model.

CHOOSE A MODEL:

Select a suitable time series forecasting model. Some common choices include ARIMA, SARIMA, Or machine learning models like XG Boost or LSTM for deep learning.

HYPERPARAMETER TUNING:

If using machine learning models, tune hyperparameter to optimize the models' performance.

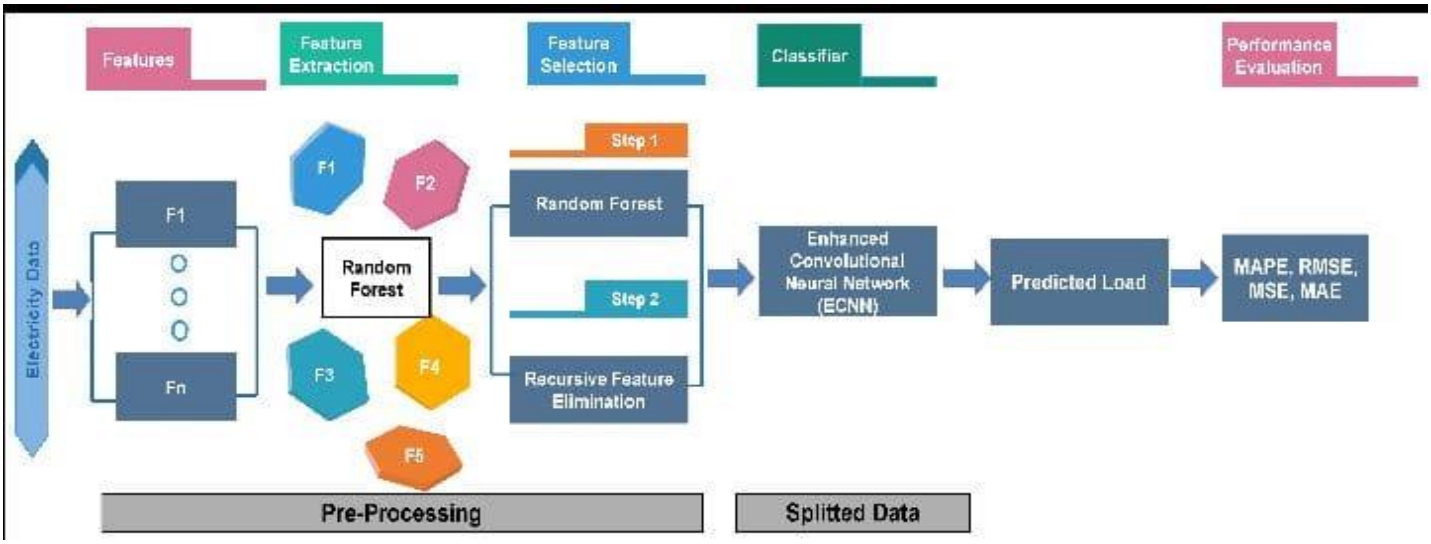
EVALUATE THE MODEL:

Use the testing data to evaluate the models' accuracy and performance using metric like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).

VIUALIZATION:

Visualize the predictions against the actual electricity prices to assess the models' performance.

MODEL ELECTRICITY PRICE AND LOAD:



Here 's simple example using scikit - learn 's Linear Regression for illustration:

```
import pandas as pd
from sk learn .model _selection import train _test _split
from sk learn .linear _model import Linear Regression
from sk learn .metrics import mean _squared _error
#Load your electricity price data into a Data frame
data = pd .read _csv('electricity_prices.csv')
#Preprocess and feature engineering steps here….
#Spilt the data into training and testing sets
X _train, X _test, y _train, y _test= train _test _spilt(features, target, test _size=0.2,random_state=42)
#Initialize and train the model
Model = Linear Regression()
Model .fit ( X _train,  y _train)
#Make predictions
Y _ pred = mode l .predict(  X _test)
#Evaluate the model
 Mse = mean  _squared _error(y _test, y _pred)
 Rmse= mse ** 0.5
 Print ( f "Root Mean Squared Error:{rmse}")
```

In the analysis, we're going to pull the times series for electricity prices for the state of Texas into Python for analysis, as shown below:

PROGRAM:

```
def retrieve _ time _series (api, series _ID):
Return the time series data frame, based on API and unique  Series ID
api: API that we're connected to
series _ID: string. Name  of the series that we want to pull from the EIA  API

#Retrieve Data By Series ID
    Series _search = api. data _by _series (series =series _ID)
##Create a  pandas data frame from the retrieved  time series
   df= pd. data frame (series_ search) return df

### Execute in the main block
#Create EIA API using  your specific API key
Api _key = "YOR API KEY HERE"
api = eia. API(api _key)I
 #Pull the electricity price data series _ID='ELEC.PRICE.TX-ALL.M'
Electricity  _df= retrieve _time _series(api, series _ID)
Electricity _df .reset _index (level=0, in place= True)
#Rename the columns for easer analysis
Electricity _df .rename(columns={'index ':'Date',
        Electricity _df .columns [1]: ' Electricity _Price'},
 In place=True)
```

OUTPUT:

	Date	Electricity_Price
Date		
2001-01-01	2001-01-01	6.90
2001-02-01	2001-02-01	6.91
2001-03-01	2001-03-01	7.02
2001-04-01	2001-04-01	7.04
2001-05-01	2001-05-01	7.34
2001-06-01	2001-06-01	7.90
2001-07-01	2001-07-01	7.98
2001-08-01	2001-08-01	8.00
2001-09-01	2001-09-01	7.65
2001-10-01	2001-10-01	7.38
2001-11-01	2001-11-01	6.92
2001-12-01	2001-12-01	6.93
2002-01-01	2002-01-01	6.73
2002-02-01	2002-02-01	7.06
2002-03-01	2002-03-01	6.64
2002-04-01	2002-04-01	6.53
2002-05-01	2002-05-01	6.32
2002-06-01	2002-06-01	6.91

Snapshot of the time series data for electricity prices, pulled via the EIA API

ELECTRICITY PRICE PREDICTION MODEL:

The task of training an electricity prediction model. I will first add all the important features to X And the target column to y, and then I will split the data into training and the sets.

```
X= data [['Day', "Month", "Forecasting Wind Production", "System Load EA",
        ",SMPEA", " ORKT Temperature", ORK Windspeed", "Co2Intensity',
        "Actual Wind Production", "System Load EP2"]]
Y=data ["MPEP2"]
  From sk learn .model _selection import train _test _spilt
X train, x test, y train, y Test = train _test _spilt (x, y, test_ size =0.2. random _state=42)
```

As this is the problem of regressions, so here I will choose the Random Forest regressions algorithm To train the electricity price prediction model:

PROGRAM:

From sk learn .ensemble import Random Forest Regressor

Model = Random Forest Regressor()

Model .fit (x train, y train)

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=100, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

PYTHON LIBRARIES USED:

Numpy, Scipy, Matplotlib, Pandas, Time, Seaborn, Requests,

JSON, Datatime, IPython, Statsmodels, Scikit Learn.

Electricity price prediciton depends on different factors like national wind, wind production and National factors etc.,,, the average cost of satate electricity supplied in India was 6.19 Indian rupees per kilowatt- hours