**PROBLEM STATEMENT**

**TNSDC NAAN MUDHALVAN (IBM)**

**Project Name: electricity price prediction**

**College code: 1108**

**College Name: Jaya Engineering College**



**Team Members:**

1. **Kaviya .A(B.Tech[Information Technology])**
2. **Varsha. R(B.Tech[Information Technology])**
3. **Jananee.G(B.Tech[Information Technology])**

**Introduction:** Electricity price forecasting (EPF) is a branch of energy forecasting which focuses on predicting the spot and forward prices in wholesale electricity markets. Over the last 15 years electricity price forecasts have become a fundamental input to energy companies' decision-making mechanisms at the corporate level.

**Program**

**Df=pd.read\_csv(“/kaggle/input/electric/electricity.gui”)**

**Df.head()**

**output:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 01/11/2011 00:00 | None | 0 | 1 | 44 | 1 | 11 | 2011 | 0 | 315.31 | 3388.77 | 49.26 | 6.00 | 9.30 | 600.71 | 356.00 | 3159.60 | 54.32 |
| 1 | 01/11/2011 00:30 | None | 0 | 1 | 44 | 1 | 11 | 2011 | 1 | 321.80 | 3196.66 | 49.26 | 6.00 | 11.10 | 605.42 | 317.00 | 2973.01 | 54.23 |
| 2 | 01/11/2011 01:00 | None | 0 | 1 | 44 | 1 | 11 | 2011 | 2 | 328.57 | 3060.71 | 49.10 | 5.00 | 11.10 | 589.97 | 311.00 | 2834.00 | 54.23 |
| 3 | 01/11/2011 01:30 | None | 0 | 1 | 44 | 1 | 11 | 2011 | 3 | 335.60 | 2945.56 | 48.04 | 6.00 | 9.30 | 585.94 | 313.00 | 2725.99 | 53.47 |
| 4 | 01/11/2011 02:00 | None | 0 | 1 | 44 | 1 | 11 | 2011 | 4 |  |  |  |  |  |  |  |  |  |

About dataset

The price of electricity depends on many factors. Predicting the price of electricity helps many businesses understand how much electricity they have to pay each year. The Electricity Price Prediction task is based on a case study where you need to predict the daily price of electricity based on the daily consumption of heavy machinery used by businesses

**Data loading**

**df=pd.read\_csv("/kaggle/input/electric/electricity.gui")**

**df.head()**

**Output**

01/11/2011 00:00

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01/11/2011 00:30 | None | 0 | 1 | 44 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01/11/2011 01:00 | None | 0 | 1 | 44 |

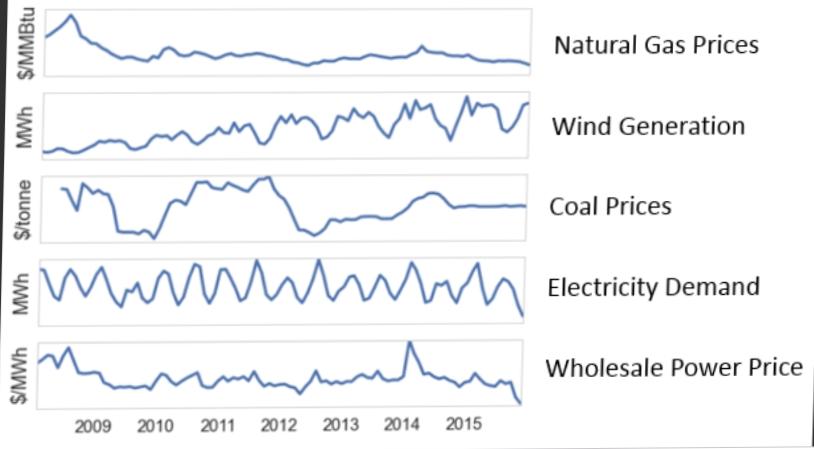
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01/11/2011 00:00 | None | 0 | 1 | 44 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01/11/2011 02:00 | None | 0 | 1 | 44 |
|  |  |  |  |  |

Predicting Electricity Prices

The final prediction model captured 93% of the variance in wholesale electricity prices.

The project team evaluated a number of analytical techniques to predict wholesale electricity prices using the available data for the years 2008–2015 for overall demand, natural gas prices, coal prices, and wind generation capacity.

**Data processing**

*from sklearn.decomposition import PCA*

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

*from math import sqrt*

*%matplotlib inline*

*for dirname, \_, filenames in os.walk('/kaggle/input'):*

*for filename in filenames:*

*print(os.path.join(dirname, filename))*

*/kaggle/input/energy-consumption-generation-prices-and-weather/energy\_dataset.csv*

*/kaggle/input/energy-consumption-generation-prices-and-weather/weather\_features.csv*

*#read the dataset*

*df\_weather = pd.read\_csv(*

*'/kaggle/input/energy-consumption-generation-prices-and-weather/weather\_features.csv',*

*parse\_dates=['dt\_iso']*

*)*

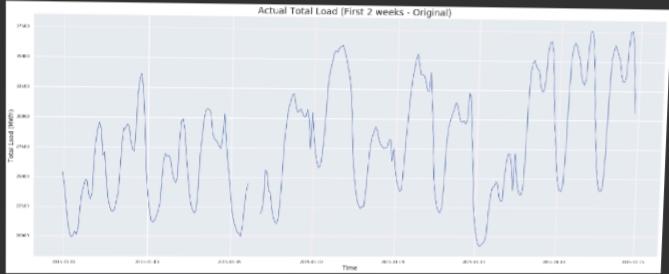
*df\_energy = pd.read\_csv(*

*'/kaggle/input/energy-consumption-generation-prices-and-weather/energy\_dataset.csv',*

*parse\_dates=['time']*

*)*

Output



**Modelling**

In [68]:

X=df.drop("SMPEP2",axis=1) #bağımsızz değişkenler

y=df["SMPEP2"] # bağımlı değişkenler

In [69]:

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,random\_state=0)

In [70]:

!pip install catboost

unfold\_moreShow hidden output

In [71]:

!pip install lightgbm

unfold\_moreShow hidden output

In [72]:

!pip install xgboost

unfold\_moreShow hidden output

In [75]:

from xgboost import XGBRegressor

from catboost import CatBoostRegressor

from lightgbm import LGBMRegressor

In [74]:

ridge=Ridge().fit(X\_train,y\_train)

lasso=Lasso().fit(X\_train,y\_train)

enet=ElasticNet().fit(X\_train,y\_train)

knn=KNeighborsRegressor().fit(X\_train,y\_train)

ada=AdaBoostRegressor().fit(X\_train,y\_train)

In [76]:

svm=SVR().fit(X\_train,y\_train)

mlpc=MLPRegressor().fit(X\_train,y\_train)

dtc=DecisionTreeRegressor().fit(X\_train,y\_train)

rf=RandomForestRegressor().fit(X\_train,y\_train)

xgb=XGBRegressor().fit(X\_train,y\_train)

gbm=GradientBoostingRegressor().fit(X\_train,y\_train)

lgb=LGBMRegressor().fit(X\_train,y\_train)

catbost=CatBoostRegressor().fit(X\_train,y\_train)

Learning rate set to 0.06876

0: learn: 34.5202564 total: 64.1ms remaining: 1m 4s

1: learn: 33.6345770 total: 70ms remaining: 34.9s

2: learn: 32.8676895 total: 74.8ms remaining: 24.8s

3: learn: 32.1602282 total: 79.9ms remaining: 19.9s

4: learn: 31.4994441 total: 85.2ms remaining: 17s

5: learn: 30.9246676 total: 91ms remaining: 15.1s

6: learn: 30.4012671 total: 96.3ms remaining: 13.7s

7: learn: 29.9286504 total: 101ms remaining: 12.5s

8: learn: 29.5172364 total: 106ms remaining: 11.7s

9: learn: 29.1386738 total: 111ms remaining: 11s

10: learn: 28.7562568 total: 116ms remaining: 10.4s

11: learn: 28.4486528 total: 122ms remaining: 10s

12: learn: 28.1657127 total: 128ms remaining: 9.7s

13: learn: 27.8831288 total: 133ms remaining: 9.38s

14: learn: 27.6408000 total: 138ms remaining: 9.07s

15: learn: 27.4359313 total: 143ms remaining: 8.8s

16: learn: 27.2173157 total: 148ms re

Implementation and Algorithms:

5.1 **Data pre-processing**

A dataset is a collection of data. With tabular data, each table row corresponds to a

specific record of the data set, and each column to a single variable. A data set is related to

one or more database tables[15] .

The Kaggle website is where the electricity dataset was found. This dataset has about

40,000items and includes the following columns: datetime, id, name, geoid, geoname, and

value. Each of these fields has a distinct meaning. dataset is a collection of data. With tabular data, each table row corresponds to a

specific record of the data set, and each column to a single variable. A data set is related to

one or more database tables[15] .

The Kaggle website is where the electricity dataset was found. This dataset has about

40,000items and includes the following columns: datetime, id, name, geoid, geoname, and

value. Each of these fields has a distinct meaning

Data processing

In: mport os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import statsmodels.api as sm

import tensorflow as tf

import xgboost as xgb

import os

import warnings

from tensorflow.keras.layers import Dense, LSTM, Conv1D, MaxPooling1D, TimeDistributed, Flatten, Dropout, RepeatVector

from statsmodels.graphics.tsaplots import plot\_acf, plot\_pacf

from statsmodels.tsa.stattools import adfuller, kpss, ccf

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.preprocessing import LabelEncoder, StandardScaler, MinMaxScaler

from sklearn.decomposition import PCA

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

from math import sqrt

%matplotlib inline

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

/kaggle/input/energy-consumption-generation-prices-and-weather/energy\_dataset.csv

/kaggle/input/energy-consumption-generation-prices-and-weather/weather\_features.csv

#read the dataset

df\_weather = pd.read\_csv(

'/kaggle/input/energy-consumption-generation-prices-and-weather/weather\_features.csv',

parse\_dates=['dt\_iso']

)

df\_energy = pd.read\_csv(

'/kaggle/input/energy-consumption-generation-prices-and-weather/energy\_dataset.csv',

parse\_dates=['time']

**Methodology**

4.1 **Problem Statement**

Predicting electricity usage is a major issue in energy management. For effective energy

management, accurate electricity consumption forecasting is crucial because it enables

energy suppliers to optimise energy distribution, cut down on energy waste, and avoid

overloading the power system[11]. The accuracy and scalability of traditional techniques

of predicting power use are constrained. Consequently, a reliable and effective way of

predicting power use is required.

The goal of this work is to create a machine learning-based method for precisely and

effectively forecasting power use[12] . Large data volumes, handling missing values and

outliers, and extracting pertinent characteristics from the data should all be capabilities of

the method. The method must to be able to decide which model performs the best and

anticipate power use with accuracy. To ascertain the suggested approach's efficacy in

forecasting powerusage, various assessment indicators should be used. The project seeks

to advance energy management by offering a precise and effective approach for

forecasting power usage.

The formula for KNN regression is as follows:

y\_hat = (1/K) \* ∑(yi), i=1 to KWhere:

y\_hat is the predicted value of the target variable for a given observation.

K is the number of nearest neighbors that will be used to make the prediction.

yi is the value of the target variable for the i-th nearest neighbor to the observation

**Modules**

1. The power utility provider is contacted in the first stage to obtain historical

information on electricity use. The information comprises of a year's worth of

hourly power use[14] .

2. The gathered data is preprocessed to deal with missing values and outliers in

the second stage. This entails locating missing data and substituting an

acceptable value for it. Once outliers have been located, they are either

eliminated or replaced with more typical values.

3. To extract pertinent features from the preprocessed data, feature engineering

is carried out in the third stage. In order to increase the model's precision,

featureslike the time of day, the day of the week, and seasonality are retrieved

from thedata[13].

4. The fourth stage involves training machine learning models on the

preprocessed and feature-engineered data, including linear regression,

decision trees, random forests, and artificial neural networks. To determine

which model is the best performer, the models are assessed using several

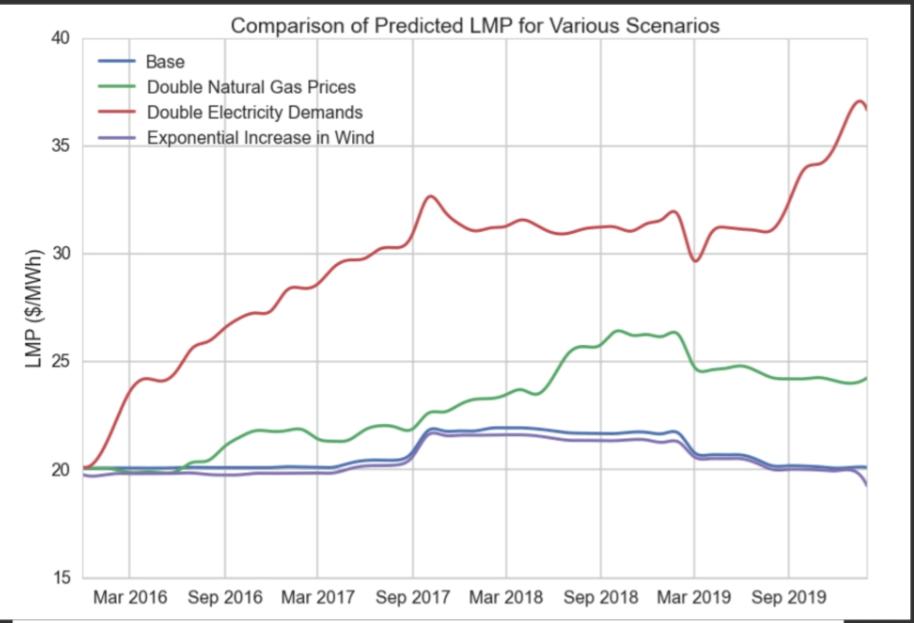
evaluation measures includingMAE, RMSE, and R2.

5. The final step is predicting power use using the chosen model. The model

predicts the amount of power used based on the important characteristics

taken from the current data.

**Forecasting Future Scenarios**

Wholesale electricity prices rise dramatically as demand increases.

To test the model, we created several hypothetical scenarios including: 1) a doubling of electricity demand over the next 5 years, 2) a doubling of the natural gas price, which is roughly equivalent to a $50/tonne carbon tax, and 3) a tenfold increase in wind generation. The results are displayed in the graph below – in each case we see that future prices vary significantly f

rom the baseline.

**Electricity prices prediction and innovation**

The final prediction model captured 93% of the variance in wholesale electricity prices. The project team evaluated a number of analytical techniques to predict wholesale electricity prices using the available data for the years 2008–2015 for overall demand, natural gas prices, coal prices, and wind generation capacity Design the system to handle an increasing volume of transactions as the business grows

**Scalability**:

CODING:

Import numpy as np

Import pandas as pd

Import seaborn as sns

Import matplotlib.pyplot as plt

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

From imblearn.over\_sampling import SMOTE

From sklearn.preprocessing import StandardScaler

From sklearn.ensemble import RandomForestClassifier

From sklearn.metrics import confusion\_matrix, classification\_report

PRE-PROCESSING:

Df=pd.read\_csv(‘/kaggle/input/creditcardfraud/creditcard.csv’)

Feature Engineering:

Create new features or manipulate existing ones that might help in fraud detection, such as transaction amounts, time of day, and more.

Data Splitting:

Divide the dataset into training and testing sets to evaluate the model’s performance.

Model Selection:

Choose an appropriate machine learning or deep learning algorithm for fraud detection. Common choices include logistic regression, decision trees, random forests, or neural networks.

Model Training:

Train the selected model on the training data.

Model Evaluation:

Assess the model’s performance on the testing data, using metrics like accuracy, precision, recall, F1-score, and ROC curves.

Tuning and Optimization:

Fine-tune hyperparameters to improve model performance.

Deployment:

Implement the model in a real-time or batch processing system for live fraud detection.

Monitoring and Maintenance:

Continuously monitor the model’s performance and retrain it as new data becomes available.

Expandability and Interpretability:

Ensure that the model’s decisions can be explained to stakeholders and auditors.

Documentation and Reporting:

Maintain detailed documentation of the project, including methodologies, findings, and model performance.

User Interface (UI):

Develop a user-friendly interface for stakeholders to interact with the fraud detection system.

Training and Education:

Train relevant personnel in the use of the system and educate them about fraud detection techniques.

Legal and Compliance:

Ensure compliance with data protection and privacy regulations.

Feedback Loop:

Permutation Importance Conclusions -

Since shap doesn't seem to agree with RandomForestRegressor.. these are the results for the second best model.

Generation biomass seems to have the highest importance, followed by generation nuclear.

The feature engineered column seems to have the highest permutation importance value, followed by generation fossil gas