GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY, TIRUTTANI - 631209

Approved by AICTE, New Delhi Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PHASE 2

PROJECT TITLE

Electricity Price Prediction Using Machine Learning

COLLEGE CODE : 1103

LEELAVATHI N V

3rd yr, 5th sem

Reg no : 110321104025

nvleela2004@gmail.com

2.1 SHORT EXPLAINATION ABOUT ELECTRICITY PRICE PREDICTION USING DATA SCIENCE

Predicting electrical prices can be a bit tricky, but it often involves analyzing various factors that influence supply and demand in the energy market. Some key elements to consider include:

1. Supply and Demand: The basic economic principle of supply and demand plays a significant role. If demand for electricity increases and the supply cannot keep up, prices tend to rise.

2. Weather Conditions: Weather can have a substantial impact on electricity prices. Extreme temperatures, especially during peak seasons, can lead to higher demand for heating or cooling, affecting prices.

3. Fuel Prices: The cost of fuels used for power generation, such as natural gas or coal, can influence electricity prices. Fluctuations in these fuel prices can directly impact the cost of producing electricity.

4.Renewable Energy Production: The availability and production of renewable energy sources, like solar and wind, can also impact prices. For example, abundant sunlight or strong winds can lead to increased renewable energy generation, potentially lowering prices

5. Regulatory Policies: Government regulations and policies regarding the energy market can impact prices. Changes in regulations, subsidies for certain types of energy, or the introduction of carbon pricing mechanisms can all have effects.

6.Infrastructure and Transmission Costs: Upgrades or issues with the electrical grid and transmission infrastructure can affect prices. The cost of transmitting electricity from where it's generated to where it's consumed can be a significant factor.

7.Geopolitical Events: Events on the global stage, such as geopolitical tensions or changes in energy policies of major producers, can influence energy prices.

Predicting prices with precision is challenging due to the complex interplay of these factors. Analysts often use a combination of historical data, economic models, and expert judgment to make predictions. Machine learning and artificial intelligence techniques are also being increasingly employed for more accurate forecasting.

2.2 WHERE I GOT THE DATASETS AND ITS DETAILS

You can find datasets for electricity prediction and various other data science projects from several reputable sources.

KAGGLE : Kaggle is a popular platform for data science competitions and dataset sharing. It hosts a wide range of datasets on various topics, including customer data. You can browse datasets, read their descriptions, and download them for free. Kaggle also provides a community where you can discuss and collaborate on data science projects.

Website : https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction

NAME OF THE DATASET : electicity-price-prediction

DATA DESCRIPTION :

When predicting electricity prices using artificial neural networks (ANNs) or other machine learning techniques, the quality and relevance of the data are crucial. Here are some key types of data that are typically used in electricity price prediction:

1.Historical Electricity Prices:

Time-series data of past electricity prices is fundamental. This includes hourly or daily prices over an extended period.

2. Demand Data:

- Historical data on electricity demand is essential. This might include daily or hourly demand patterns, seasonal variations, and special events that could impact demand.

3. Weather Data:

- Weather conditions can have a significant impact on electricity prices. Include data such as temperature, humidity, wind speed, and precipitation. For example, extreme temperatures may lead to higher demand for heating or cooling.

4. Generation Data:

- Information about the types and amounts of energy generation, including both traditional sources (coal, natural gas) and renewable sources (solar, wind). The availability of renewable energy can influence prices.

5. Fuel Prices:

- The cost of fuels used for power generation, such as natural gas or coal. Fluctuations in fuel prices can directly affect the cost of producing electricity.

6. Regulatory Data:

- Information on regulations and policies affecting the energy market. Changes in regulations, subsidies, or carbon pricing mechanisms can impact prices.

7. Transmission and Grid Data:

- Data related to the infrastructure of the electrical grid, including transmission capabilities and any upgrades or maintenance activities.

8. Market and Economic Indicators:

- Economic factors like inflation rates, GDP growth, and market trends can influence electricity prices.

9. Public Holidays and Events:

- Special events or holidays that might impact electricity demand and pricing.

10. Outages and Maintenance Data:

- Information on planned or unplanned outages of power plants or parts of the grid. These events can impact supply and, consequently, prices.

11. Exchange Rates (if applicable):

- If your electricity market is influenced by international factors, consider including exchange rates in your dataset.

Remember, the key is to create a comprehensive dataset that captures the various factors influencing electricity prices. Data preprocessing steps, such as handling missing values, scaling, and feature engineering, are also crucial to ensure the effectiveness of your prediction model. Additionally, keeping the dataset up-to-date is essential for maintaining the accuracy of your predictions over time.

2.3 DETAILS ABOUT COLUMNS

To develop an electricity price prediction model, relevant columns from the dataset would typically include:

- DateTime: Essential for time-series analysis.
- Holiday and HolidayFlag: These could impact electricity demand and prices.
- DayOfWeek and WeekOfYear: Capture weekly patterns.
- Day, Month, and Year: Provide temporal features.
- PeriodOfDay: Indicates different time intervals affecting consumption.

➢ ForecastWindProduction, ActualWindProduction: Wind production influences overall supply.

➢ SystemLoadEA, SystemLoadEP2: Current and predicted system loads impact prices.

SMPEA, SMPEP2: Price indicators**.**

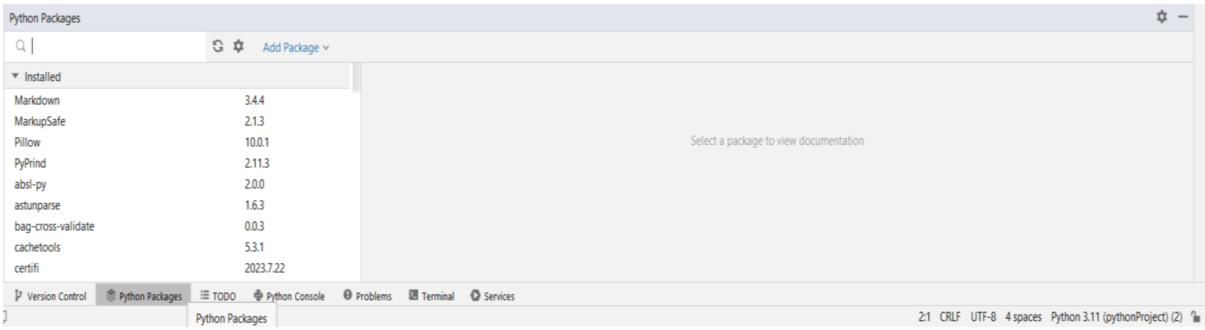
2.4 DETAILS OF LIBRARIES TO BE USED AND WAY TO DOWNLOAD

LIBRARIES TO BE USED

- import numpy as np
- import pandas as pd
- import matplotlib.pyplot as plt
- from sklearn.model_selection import train_test_split
- from sklearn.ensemble import RandomForestRegressor
- from fbprophet import Prophet

WAY TO DOWNLOAD THE LIBRARIES

1. Click the python packages in the bottom of your project in pycharm



2. Type the required library in the search box and click install package in the right end top of the python packages.

3. After installation process finished it shows the package was installed in the python packages.

2.5 HOW TO TRAIN AND TEST THE DATASET

To train and test a machine learning model using a dataset of mall customers with the given column names (DateTime, Holiday, HolidayFlag, DayOfWeek, WeekOfYear, Day, MonthandYear,ForecastWindProduction,SystemLoadEA,SMPEA,ActualWindProduction,SystemLoadEP2,SMPEP2)), you can follow these steps:

Data Preprocessing :

Load your dataset into a data analysis or machine learning environment (e.g., Python with libraries like pandas and scikit-learn).

Explore and clean the data to handle any missing values, duplicates, or outliers.

- ❖ Handle Missing Data:Check for missing values in your dataset and decide on a strategy for handling them. You can choose to drop missing values, impute them with mean or median, or use more advanced imputation techniques.
- ❖ Convert DateTime Column:Ensure that the DateTime column is in the correct format and set it as the index of the DataFrame, etc.,

Splitting the Data :

Divide your dataset into two parts: a training set and a testing set. A common split is 80% for training and 20% for testing, but you can adjust this ratio as needed.

Ensure that the split maintains a representative distribution of data, especially if you have imbalanced classes or segments.

Selecting a Machine Learning Model :

Choose an appropriate machine learning model for your task. Since you want to electricity price prediction choose machine earning algorithms like clustering (e.g., Random

Encode Categorical Variables:

If there are categorical variables like 'Holiday,' you may need to encode them for the model. In the case of binary categories (e.g., 'HolidayFlag'), you might not need encoding.

```
df['Holiday'] = df['Holiday'].astype('category').cat.codes
```

Feature Selection:

Select the relevant features for your model. In the case of Prophet, you would typically include the DateTime column and the target variable ('y').

```
selected_features = df[['DateTime', 'SMPEA']]
```

Train-Test Split:

Split your data into training and testing sets. This is crucial for evaluating the performance of your model.

```
from sklearn.model_selection import train_test_split

X = df.drop(columns=['SMPEA'])

y = df['SMPEA']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

2.6 REST OF EXPLAINATION

Additional Feature Engineering:

Depending on your domain knowledge, you may need to perform additional feature engineering. For example, creating lag features, extracting time-related features, or transforming variables.

Scale/Normalize Data:

Depending on the algorithm used, you may want to scale or normalize your features. Some algorithms, like neural networks, can benefit from scaled inputs.

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)
```

Instantiate and Fit the Prophet Model:

Instantiate a Prophet model and fit it to the training data.

```
from fbprophet import Prophet

model.fit(X_train)
```

Create a DataFrame for Future Dates:

Create a DataFrame containing future dates for which you want to make predictions.

```
future = model.make_future_dataframe(periods=len(X_test), freq='D')
```

Generate Forecasts:

Use the fitted model to generate forecasts for the future dates.

```
forecast = model.predict(future)
```

Evaluate the Model:

Evaluate the performance of the model using metrics relevant to time series forecasting. Common metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

```
from sklearn.metrics import mean_absolute_error, mean_squared_error

y_true = y_test.values

y_pred = forecast[-len(X_test):]['yhat'].values

mae = mean_absolute_error(y_true, y_pred)

mse = mean_squared_error(y_true, y_pred)

rmse = mse ** 0.5

print(f'MAE: {mae}, MSE: {mse}, RMSE: {rmse}')
```

Visualize the Results:

Plot the actual vs. predicted values to visually inspect the model's performance.

```
fig = model.plot(forecast)
```

Explore Forecast Components:

Prophet provides a built-in method for exploring the components of the forecast, including trend, seasonality, and holidays.

```
fig_components = model.plot_components(forecast)
```

Adjust Hyperparameters:

Optionally, you can experiment with adjusting hyperparameters of the Prophet model, such as seasonality parameters or the changepoint parameter, based on the performance on your specific dataset.

```
model.add_seasonality(name='weekly_custom', period=7, fourier_order=5)
```

Fine-Tuning and Optimization (Optional):

Depending on the performance of the initial model, you may perform further fine-tuning and optimization. This may involve adjusting the feature set, exploring different model configurations, or considering additional preprocessing steps.

These steps provide a comprehensive guide for using the Prophet algorithm for time series forecasting.

2.7 WHAT METRICS USED FOR THE ACCURACY CHECK

Prophet is particularly advantageous for electricity price prediction due to its specific features:

Time Series Emphasis: Prophet is designed with a focus on time series data, making it inherently suitable for capturing temporal patterns present in electricity prices.**Automatic Seasonality Detection:** The algorithm can automatically detect and model seasonality, an essential characteristic in electricity prices with recurring daily and weekly patterns.**Holiday and Event Effects:** Including holiday effects is crucial in energy markets, and Prophet allows for the explicit modeling of these effects, enhancing the accuracy of predictions during special events.**Changepoint Detection:** Prophet's ability to automatically detect changepoints allows it to adapt to structural changes in electricity prices, a common occurrence in dynamic markets.**Uncertainty Quantification:** Providing uncertainty estimates is vital in volatile energy markets, helping stakeholders make informed decisions based on the reliability of predictions.**Robust Handling of Data Imperfections**: Prophet's robustness to missing data and outliers ensures that it can effectively model electricity price data, which may often contain irregularities.**User-Friendly Interface:** The algorithm's simplicity and user-friendly interface make it accessible to users with varying levels of expertise, facilitating quick model development and deployment.

While other machine learning algorithms may offer flexibility and complexity, Prophet's specialized features align well with the unique characteristics and requirements of electricity price prediction, making it a pragmatic choice for this specific application.