**ELECTRICITY PRICE PREDICTION**

PHASE – 5



**INTRODUCTION**

Electricity price prediction is the process of using historical and real-time data to forecast future electricity prices in the energy market. This forecasting task is essential for energy market participants, utility companies, investors, and consumers as it provides insights into the future cost of electricity, enabling more informed decision-making. Electricity price prediction leverages data analysis, statistical methods, and machine learning to provide valuable insights into the future cost of electricity. This information is crucial for energy market participants to optimize their strategies, for utilities to manage the grid effectively, and for consumers to make informed decisions about their energy consumption, ultimately contributing to a more sustainable and economically efficient energy ecosystem. Advanced techniques and the availability of comprehensive data sources continue to enhance the accuracy and relevance of electricity price predictions.

**1.PROBLEM STATEMENT AND DESIGN THINKING**

Problem Statement:

Create a predictive model that utilizes historical electricity prices and relevant factors to forecast future electricity prices, assisting energy providers and consumers in making informed decisions regarding consumption and investment.

Problem Definition:

 The problem is to develop a predictive model that uses historical electricity prices and relevant factors to forecast future electricity prices. The objective is to create a tool that assists both energy providers and consumers in making informed decisions regarding consumption and investment by predicting future electricity prices. This project involves data preprocessing, feature engineering, model selection, training and evaluation.

**Design thinking:**

1**.Data source:**

In electricity price prediction, the choice of data sources is critical for building accurate and robust predictive models. Here are some common data sources used in electricity price prediction

* **Historical Electricity Price Data:**

Historical price data is the backbone of any electricity price prediction model. It includes information about electricity prices at different time intervals (e.g., hourly, daily) for a specific geographical region or market.

* **Weather Data:**

Weather conditions have a significant impact on electricity demand and supply. Data sources such as temperature, humidity, wind speed, and precipitation can be used to account for weather-related factors that influence electricity prices.

* **Demand Data**:

Information on electricity demand patterns is crucial. This data can come from utility companies and may include historical demand levels at different times of the day, week, or year.

* **Supply Data:**

Data on electricity generation and supply sources are important. This can include information on power plant capacity, availability, and generation methods (e.g., renewable, fossil fuels).

* **Economic Indicators:**

Macroeconomic data, such as GDP growth rates, inflation rates, and industrial production figures, can influence electricity demand and, subsequently, prices.

**2.Data preprocessing:**

Data preprocessing is a crucial step in electricity price prediction as it ensures that the data used for modeling is clean, relevant, and suitable for the predictive task. Here are key data preprocessing steps in electricity price prediction.

* **Data Cleaning**:
* Handle missing values: Identify and fill in or impute missing data points in the price, weather, or other relevant datasets. Common methods include mean imputation or interpolation.
* Outlier detection: Identify and address outliers in the data, which can significantly impact model performance. Outliers in price data, for example, could distort predictions.
* **Time Series Transformation**:
* Ensure stationary data: Many time series models assume stationarity. You may need to perform differencing or other transformations to make the time series data stationary.
* Seasonal decomposition: Decompose the time series into its seasonal, trend, and residual components using methods like seasonal decomposition of time series (STL).
* **Handling Imbalanced Data:**

Electricity price data can sometimes be imbalanced, with periods of stable prices and occasional spikes. Consider techniques like oversampling or under sampling to balance the dataset.

* **Data Splitting:**

Split the data into training, validation, and test sets. Ensure that the time order is preserved to simulate real-world forecasting scenarios.

**3.Feature engineering:**

Feature engineering is a crucial step in electricity price prediction as it involves transforming raw data into informative features that can improve the accuracy of predictive models. Here are some common feature engineering techniques and considerations specific to electricity price prediction.

* **Time-Based Features:**
* Time of day: Extracting features like hour of the day, day of the week, or month can capture daily, weekly, and seasonal patterns in electricity prices.
* Holidays and special events: Including binary flags for holidays or significant events that affect electricity demand and pricing.
* **Lagged Variables:**
* Lagged electricity prices: Adding lagged (previous time steps) prices as features can help capture temporal dependencies and autocorrelation in price data.
* Weather data: Incorporating lagged weather variables such as temperature, humidity, and wind speed can account for weather-related effects on electricity demand and supply.
* **Price Spread and Volatility**:
* Price spread: Calculating the difference between high and low prices within a time period can indicate market volatility.
* Volatility measures: Incorporating metrics like historical price volatility or implied volatility can provide insights into price uncertainty.

**4.Model selection:**

Selecting the right model for electricity price prediction is crucial for accurate and reliable forecasts. Model selection involves choosing an appropriate machine learning or statistical model that can capture the underlying patterns in the data. Here are some commonly used models for electricity price prediction.

* **Time Series Models:**
* ARIMA (Auto Regressive Integrated Moving Average): ARIMA models are widely used for time series forecasting, including electricity prices. They can capture seasonality and trends in price data.
* Seasonal Decomposition of Time Series (STL): STL decomposes time series data into seasonal, trend, and residual components, making it easier to model each component separately.
* Prophet: Developed by Facebook, Prophet is designed for forecasting with daily observations that display patterns on different time scales. It can handle holidays and special events, which are relevant in electricity price prediction.
* **Machine Learning Models:**
* Random Forest: Random Forest is an ensemble learning method that can capture complex relationships in the data. It is often used when there are nonlinear patterns in electricity price data.
* Gradient Boosting (e.g., XGBoost, LightGBM): Gradient boosting algorithms are powerful for regression tasks. They can handle large datasets and capture both linear and nonlinear patterns.
* **Support Vector Machines (SVM):**
* SVMs can be used for regression tasks, but they are less common in time series forecasting for electricity prices compared to other models.

* **Proximity-Based Models**:
* k-Nearest Neighbors (k-NN): k-NN can be used for regression by averaging the k-nearest neighbors' values. It's simple but may not capture complex temporal patterns well.

**5.Model training:**

Training a model for electricity price prediction typically involves the following steps:

* **Data Collection:**

Gather historical data on electricity prices. This data should include timestamps, price values, and potentially relevant features like weather conditions, demand, or generation sources.

* **Data Preprocessing:**

Clean and preprocess the data. This involves handling missing values, normalizing or scaling the data, and encoding categorical variables.

* **Deployment:** Once satisfied with the model's performance, deploy it to make real-time predictions on electricity prices.
* **Monitoring and Maintenance:** Continuously monitor the model's performance in a production environment and retrain it periodically with new data to ensure its accuracy remains high.

**6.Evaluation:**

In electricity price prediction, evaluating the performance of your predictive model is crucial to ensure its accuracy and reliability. Here are some common evaluation metrics and techniques used for assessing the quality of electricity price predictions:

* **Mean Absolute Error (MAE):**

MAE measures the average absolute difference between the predicted and actual prices. It provides a simple and interpretable measure of prediction error.

* **Mean Squared Error (MSE):**

MSE calculates the average squared difference between predicted and actual prices. It penalizes larger errors more heavily than MAE, making it sensitive to outliers.

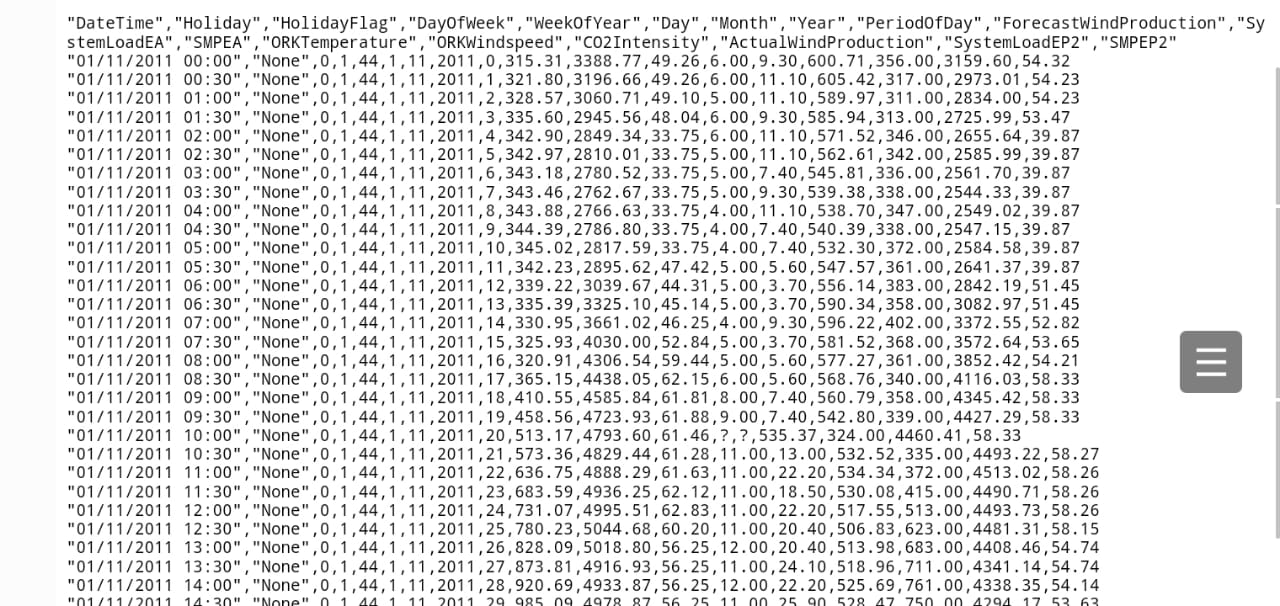
* **Root Mean Squared Error (RMSE):**

RMSE is the square root of MSE and provides a measure of the typical prediction error. It's in the same unit as the target variable (electricity price) and is commonly used in regression tasks.

**DATASET**

The dataset used for electricity price prediction typically consists of historical data related to electricity prices and various relevant factors that can influence those prices. Some of the key components of the dataset are Price data , Demand data, Supply data, weather data, Market data, Calendar data, Economic data.

<http://raw.githubusercontent.com/amankharwal/website-data/master/electricity.csv>



**2. DESIGN INTO INNOVATION**

LIBRARIES

Python libraries are used to perform various tasks, including data manipulation, modeling, evaluation, and visualization. Here is the list of libraries used in the code:

1.Pandas (import pandas as pd):

Pandas is a powerful library for data manipulation and analysis. It provides data structures like DataFrames that are used to load, manipulate, and preprocess the dataset.

2.NumPy (import numpy as np):

NumPy is a fundamental library for numerical computing in Python. It is often used for mathematical operations and working with arrays and matrices.

3.Matplotlib (import matplotlib.pyplot as plt):

Matplotlib is a widely used library for creating visualizations and plots. In the code, it's used for visualizing the actual vs. predicted electricity prices.

4.Scikit-Learn (from sklearn.model\_selection import train\_test\_split, from sklearn.linear\_model import LinearRegression, from sklearn.ensemble import RandomForestRegressor, from sklearn.metrics import mean\_squared\_error, r2\_score):

Scikit-Learn is a comprehensive machine learning library. It provides tools for data splitting, model training, evaluation, and metrics calculation. In the code examples, Scikit-Learn is used to split the dataset, train machine learning models (Linear Regression and Random Forest), and calculate evaluation metrics (MSE and R2 score)

COLUMNS

In the coding examples provided for electricity price prediction, various columns from the dataset are used as features (X) and the target variable (y) for the machine learning model. . By using these features in a machine learning model, we aim to learn patterns and relationships that allows us to predict future electricity prices based on historical data.

1.Hour:

This column represents the hour of the day when the electricity price data was recorded. It's a categorical or numerical feature that helps capture the diurnal patterns in electricity prices. For example, electricity prices might be higher during peak demand hours.

2.DayOfWeek:

This column represents the day of the week when the electricity price data was recorded. It's typically encoded as a numerical feature (e.g., 0 for Monday, 1 for Tuesday, etc.) and helps account for weekly patterns in electricity prices. For instance, weekdays might have different price patterns compared to weekends.

3.Temperature:

This column represents the temperature at the time of recording. Temperature is often considered an important explanatory variable in electricity price prediction because it can affect both electricity demand (e.g., more air conditioning in hot weather) and supply (e.g., temperature-related changes in generation capacity).

4.Price:

This is the target variable that we are trying to predict. It represents the electricity price at the given time. It's what we want our machine learning model to predict based on the other features. Prices can vary significantly depending on factors such as demand, generation mix, and market conditions.

5.Date and Time:

These columns provide the timestamp for each data point. They are often used to organize the data temporally. Depending on the dataset, we might need to preprocess these columns to extract useful information like the day of the week or the hour.

METRICS

We use two common metrics to evaluate the performance of the predictive models: Mean Squared Error (MSE) and R-squared (R2) score. These metrics provide insights into how well the model is performing in terms of prediction accuracy and variance explained.

1.Mean squared error(MSE):

The MSE is a measure of the average squared difference between the predicted values and the actual target values. It quantifies the average of the squared errors or residuals.

Where,

• n is the number of data points

• yi^ is the predicted value for the i-th data point.

• yi is the actual (observed) value for the i-th data point.

2.R squared score:

The R-squared score, also known as the coefficient of determination, quantifies the proportion of the variance in the target variable that is explained by the model. It is a value between 0 and 1, where 1 indicates a perfect fit, and 0 indicates that the model does not explain any of the variance.

Where,

• n is the number of data points.

• yi^ is the predicted value for the i-th data point.

• yi is the actual (observed) value for the i-th data point.

• yˉ is the mean of the actual values.

Training and testing data:

Training and testing your machine learning model is a crucial step in the predictive modeling process. It involves splitting your dataset into two parts: a training set and a testing (or validation) set. The training set is used to train the model, while the testing set is used to evaluate the model's performance.

**1.Import the necessary libraries:**

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

from sklearn.linear\_model import LinearRegression

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

**2.Load and preprocess your dataset**:

As explained earlier, load the data set, and preprocess it as needed. Ensure that it has feature matrix x and target variable y.

**3.Split the data into training and testing sets:**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**4.Create and train the machine learning model:**

model = LinearRegression()

model.fit(X\_train, y\_train)

**5.Make predictions using the trained model:**

y\_pred = model.predict(X\_test)

**6.Evaluate the model:**

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f'Mean Squared Error (MSE): {mse:.2f}')

print(f'R-squared (R2) Score: {r2:.2f}'

Data visualization:

Visualization is a crucial part of understanding and communicating the results of electricity price prediction models. Here are some common types of visualizations used in the context of electricity price prediction:

**Time Series Plots:** Time series plots are fundamental for visualizing historical electricity prices. A simple line chart can display how prices have fluctuated over time. You can use different colors or annotations to highlight specific events or periods.

**Forecasting vs. Actual Prices**: To assess the accuracy of your price predictions, you can create a chart that overlays the predicted prices with the actual prices. This allows you to see where your model's predictions align with or deviate from the real prices.

**Error Metrics Over Time:** Plots of error metrics (e.g., MAE, MSE, RMSE) over time can help you understand the performance of your model. It can show if your predictions are becoming more accurate or if there are periods of consistently high errors.

**Price Distribution Histograms**: Histograms can illustrate the distribution of electricity prices, helping to identify patterns like seasonality, skewness, and outliers in the data.

**Seasonal Decomposition**: Seasonal decomposition charts (e.g., using the additive or multiplicative decomposition) can reveal the underlying trends, seasonal components, and residual fluctuations in electricity prices.

**Heatmaps**: Heatmaps can visualize the correlation between electricity prices and various factors like temperature, demand, or market indices. This can help identify which variables have the most significant influence on prices.

**Candlestick Charts:** Candlestick charts are often used in finance and energy markets to represent price movements over time. They provide information on open, close, high, and low prices within a specified time period.

**Box Plots**: Box plots can display the distribution of electricity prices within specific time intervals, allowing you to see price quartiles, outliers, and variability.

**Scatter Plots:** Scatter plots can show the relationship between electricity prices and other variables. For example, you can use a scatter plot to visualize the relationship between temperature and electricity prices, which can be particularly useful for identifying weather-related patterns.

**LSTM Model Predictions**: If you're using LSTM or other deep learning models, you can create plots showing how well your model predicts future electricity prices compared to actual prices. This can help you assess model performance over time.

**Interactive Dashboards**: Interactive dashboards, built using tools like Tableau, Power BI, or Plotly, allow users to explore and interact with the data. You can include various visualizations and filtering options for a more comprehensive understanding of electricity price dynamics.

**Geospatial Visualizations**: If relevant, geospatial visualizations can show electricity price variations across regions or locations. These maps can help energy companies optimize their operations.

**Anomaly Detection:** Visualizations can highlight anomalies or unusual price movements, helping grid operators and traders identify potential issues or opportunities.

**Predicted Price Confidence Intervals:** Plotting the predicted price along with confidence intervals can show the uncertainty associated with predictions. This can help users understand the range of possible price outcomes.

**Customized Visualization:** Depending on the specific goals and requirements of your electricity price prediction project, you may need to create custom visualizations to showcase particular aspects or insights from your data.

**3.DEVELOPMENT PART 1**

**1.Importing libraries and loading dataset:**

import pandas as p import numpy as np import seaborn as sns import matplotlib.pyplot as plt data = pd.read\_csv("http://raw.githubusercontent.com/amankharwal/website-data/master/electricity.csv") print(data.head())

Output:

DateTime Holiday HolidayFlag DayOfWeek WeekOfYear Day Month \

0 01/11/2011 00:00 NaN 0 1 44 1 11

1 01/11/2011 00:30 NaN 0 1 44 1 11

2 01/11/2011 01:00 NaN 0 1 44 1 11

3 01/11/2011 01:30 NaN 0 1 44 1 11

4 01/11/2011 02:00 NaN 0 1 44 1 11

Year PeriodOfDay ForecastWindProduction SystemLoadEA SMPEA \

0 2011 0 315.31 3388.77 49.26

1 2011 1 321.80 3196.66 49.26

2 2011 2 328.57 3060.71 49.10

3 2011 3 335.60 2945.56 48.04

4 2011 4 342.90 2849.34 33.75

ORKTemperature ORKWindspeed CO2Intensity ActualWindProduction SystemLoadEP2 \

0 6.00 9.30 600.71 356.00 3159.60

1 6.00 11.10 605.42 317.00 2973.01

2 5.00 11.10 589.97 311.00 2834.00

3 6.00 9.30 585.94 313.00 2725.99

4 6.00 11.10 571.52 346.00 2655.64

SMPEP2

0 54.32

1 54.23

2 54.23

3 53.47

4 39.87

**2. Exploratory Data Analysis(EDA):**

2.1 data.info()

data = data.dropna()

Output:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 38014 entries, 0 to 38013

Data columns (total 18 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 DateTime 38014 non-null object

1 Holiday 1536 non-null object

2 HolidayFlag 38014 non-null int64

3 DayOfWeek 38014 non-null int64

4 WeekOfYear 38014 non-null int64

5 Day 38014 non-null int64

6 Month 38014 non-null int64

7 Year 38014 non-null int64

8 PeriodOfDay 38014 non-null int64

9 ForecastWindProduction 38014 non-null object

10 SystemLoadEA 38014 non-null object

11 SMPEA 38014 non-null object

12 ORKTemperature 38014 non-null object

13 ORKWindspeed 38014 non-null object

14 CO2Intensity 38014 non-null object

15 ActualWindProduction 38014 non-null object

16 SystemLoadEP2 38014 non-null object

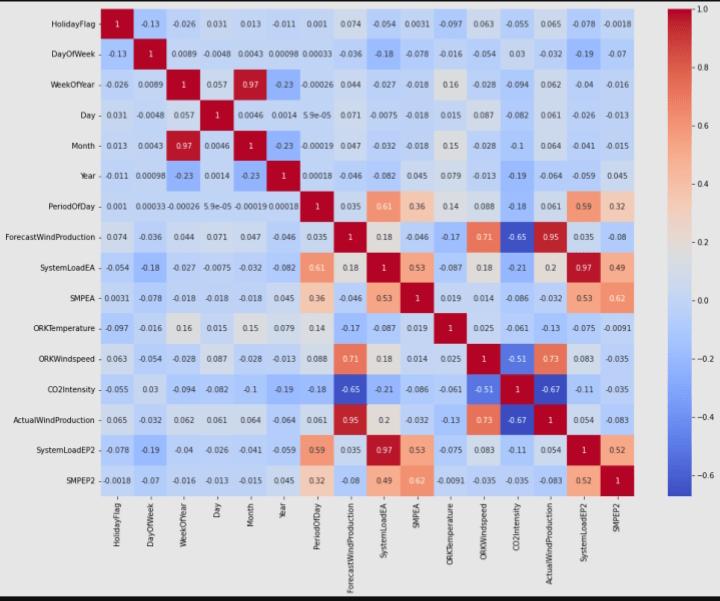
17 SMPEP2 38014 non-null object

dtypes: int64(7), object(11)

memory usage: 5.2+ MB

2.2

correlations = data.corr(method='pearson plt.figure(figsize=(16, 12)) sns.heatmap(correlations, cmap="coolwarm", annpt=True) plt.show()



**3.Prediction model:**

3.1 x=data[["Day","Month","ForecastWindProduction","SystemLoadEA","SMPEA","ORKTemperature","ORKWindspeed","CO2Intensity","ActualWindProduction","SystemLoadEP2"]] y = data["SMPEP2"] from sklearn.model\_selection import train\_test\_split xtrain,xtest,ytrain,ytest = train\_test\_split(x,y,test\_size = 0.2,random\_state=42)

3.2

from sklearn.ensemble import RandomForestRegressor model = RandomForestRegressor() model.fit(xtrain,ytrain)

Output:

RandomForestRegressor()

**4.Features:**

features = np.array([[10, 12, 54.10, 4241.05, 49.56, 9.0, 14.8, 491.32, 54.0, 4426.84]]) model.predict(features)

Output:

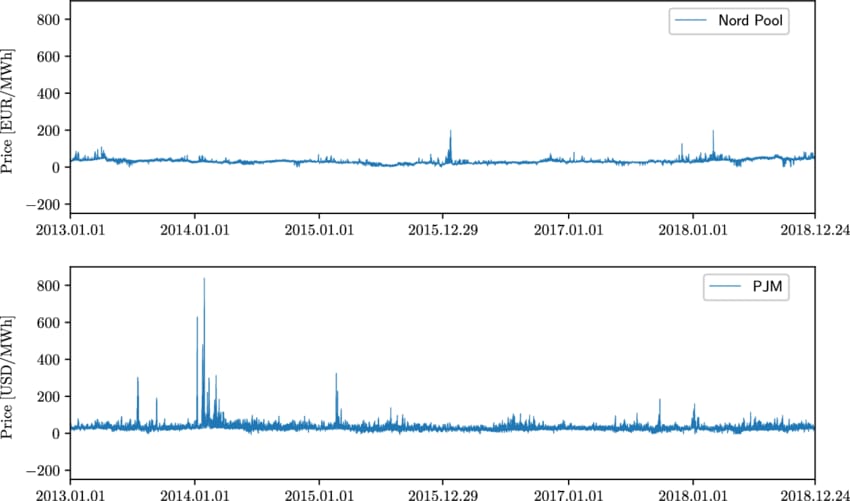
array([100.9588])

DIFFERENT ANALYSIS

1.Time series analysis:

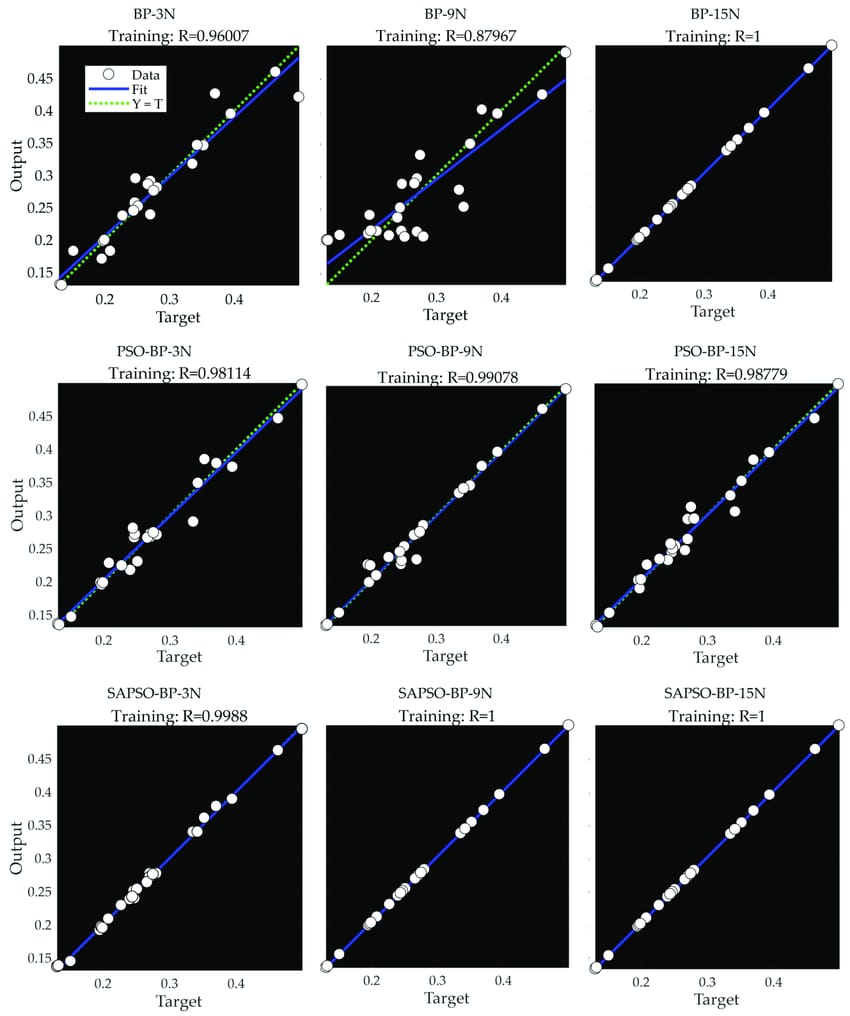
ARIMA (AutoRegressive Integrated Moving Average): This is a classic method for time series forecasting. It decomposes historical data into components like trend, seasonality, and noise, making it easier to make predictions.

Seasonal Decomposition of Time Series (STL): STL is a more advanced technique that can handle data with irregular seasonality and trends.

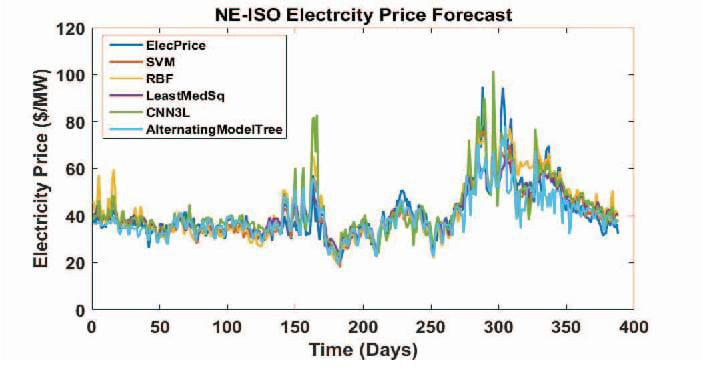


2. Machine Learning Models:

Regression Models: Linear regression, polynomial regression, or other regression techniques can be used to predict electricity prices by considering various features like historical prices, weather data, and demand.



Support Vector Machines (SVM): SVM can be applied for regression to predict electricity prices, especially when dealing with nonlinear relationships.



Random Forests and Gradient Boosting: These ensemble methods are powerful for capturing complex patterns in the data. They can handle various features and are robust to overfitting.

**4. DEVELOPMENT PART 2**

**Overview of the process**:

The following is an overview of the process of electricity

price prediction model by feature selection, model training, and

evaluation:

**1. Prepare the data:** This includes cleaning the data, removing

outliers, and handling missing values.

**2. Perform feature selection**: This can be done using a variety of

methods, such as correlation analysis, information gain, and recursive

feature elimination.

**3. Train the model:** There are many different machine learning

algorithms that can be used for electricity price prediction. Some popular

choices include linear regression, random forests, and gradient boosting

machines.

**4. Evaluate the model:** This can be done by calculating the mean

squared error (MSE) or the root mean squared error (RMSE) of the

model's predictions on the held-out test set.

**5. Deploy the model:** Once the model has been evaluated and found

to be performing well, it can be deployed to production so that it can be

used to predict the electricity prices.

FEATURE ENGINEERING

Feature engineering for electricity price prediction typically involves data preprocessing and transformation tasks that can be implemented in various programming languages like Python.

**1.Historical Price Data**:

Lagged Values: Include past electricity prices as features, which can help capture seasonality and trends.

df['price\_lag\_1'] = df['price'].shift(1)

df['price\_lag\_7'] = df['price'].shift(7)

Rolling Averages: Calculate rolling averages or moving averages of past prices over specific time windows.

df['price\_rolling\_7'] = df['price'].rolling(window=7).mean()

**2.Time-Related Features:**

Date and Time Components: Extract components like day of the week, month, hour, or season to account for time-based patterns and seasonality.

df['hour'] = df['timestamp'].dt.hour

df['day\_of\_week'] = df['timestamp'].dt.dayofweek

df['month'] = df['timestamp'].dt.month

**3.Weather Data:**

Temperature: Incorporate historical and forecasted temperature data since weather conditions can impact electricity demand and supply.

df['temperature'] = weather\_df['temperature']

Precipitation: Include information on rainfall or snowfall, which can affect energy consumption and generation.

**4.Demand and Supply Data:**

Electricity Demand: Historical and forecasted demand data can be essential in understanding price fluctuations.

Generation Capacity: Include data on available generation capacity and production from various sources (e.g., renewables, fossil fuels).

**5.Market Data:**

Fuel Prices: The cost of fuels used for electricity generation, such as natural gas or coal, can impact prices.

Transmission Line Data: Information about congestion on transmission lines and grid conditions.

**6.Economic Indicators:**

Economic Variables: Features related to economic factors like GDP, unemployment rates, or industrial production can affect electricity consumption.

**7.Event-Based Features:**

Holidays: Include information about holidays, as electricity consumption patterns often change during holidays.

Special Events: Consider major events or incidents that could impact electricity supply or demand (e.g., heatwaves, natural disasters).

**8.Technical Indicators:**

Moving Averages: Calculate various moving averages of prices or other relevant metrics.

Relative Strength Index (RSI), MACD, or other financial indicators that capture market sentiment.

**9.Categorical Features:**

Market Zones: If the electricity market is divided into zones, include the zone as a categorical feature.

Day Type: Encode weekdays, weekends, or holidays as categorical variables.

**10.Feature Scaling and Transformation**:

Normalize or standardize numerical features to ensure that they have similar scales.

Logarithmic or other transformations may be applied to features if their distributions are skewed.

df = pd.get\_dummies(df, columns=['day\_of\_week'])

**11.Feature Interactions:**

Create interaction features to capture relationships between different features. For example, the interaction between temperature and electricity demand.

df['temperature\_demand\_interaction'] = df['temperature'] \* df['demand']

**12.Feature Selection:**

Use techniques like feature importance from machine learning models or correlation analysis to select the most relevant features.

feature\_importances = model.feature\_importances\_

print("Feature Importances:", feature\_importances)

MODEL TRAINING

Training a model for electricity price prediction involves several steps, from data preparation to selecting the appropriate machine learning algorithm and fine-tuning the model.

**1.Data Collection and Preprocessing**:

Collect historical data on electricity prices, weather conditions, demand, and other relevant factors. Ensure the data is clean, organized, and stored in a suitable format (e.g., CSV or a database).

Handle missing values, outliers, and anomalies in the data.

**2.Data Splitting:**

Divide your data into training, validation, and testing sets. In time series data, it's important to respect the temporal order, with the training set containing earlier data and the validation/testing sets containing more recent data.

X\_train = series\_reshaped[:43800, :n\_steps]

X\_valid = series\_reshaped[43800:52560, :n\_steps]

X\_test = series\_reshaped[52560:, :n\_steps]

Y = np.empty((61134, n\_steps, 24))

for step\_ahead in range(1, 24 + 1):

Y[..., step\_ahead - 1] = series\_reshaped[..., step\_ahead:step\_ahead + n\_steps, 0]

Y\_train = Y[:43800]

Y\_valid = Y[43800:52560]

Y\_test = Y[52560:]

print("Shape of X\_train=", X\_train.shape, "\n",

" X\_valid=", X\_valid.shape, "\n",

" X\_test=", X\_test.shape, "\n",

" Y\_train=", Y\_train.shape, "\n",

" Y\_valid=", Y\_valid.shape, "\n",

" Y\_test=", Y\_test.shape)

**3.Feature Scaling:**

Normalize or standardize the numerical features to ensure they have similar scales. This can help the model converge faster and perform better**.**

**4.Select a Machine Learning Algorithm:**

* Linear Regression

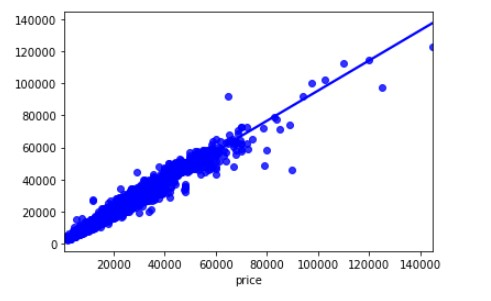
lr\_model = LinearRegression()

lr\_model.fit(train[['ForecastWindProduction', 'CO2Intensity']], train['Price'])

lr\_forecast = lr\_model.predict(test[['ForecastWindProduction', 'CO2Intensity']])

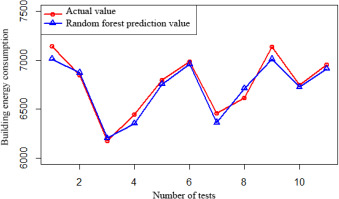
lr\_mse = mean\_squared\_error(test['Price'], lr\_forecast)

print(f'Linear Regression Mean Squared Error: {lr\_mse}')



* Random Forest

rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=0) rf\_model.fit(train[['ForecastWindProduction', 'CO2Intensity']], train['Price']) rf\_forecast = rf\_model.predict(test[['ForecastWindProduction', 'CO2Intensity']]) rf\_mse = mean\_squared\_error(test['Price'], rf\_forecast) print(f'Random Forest Mean Squared Error: {rf\_mse}')



* ARIMA

arima\_model = ARIMA(train['Price'], order=(5,1,0)) arima\_result = arima\_model.fit(disp=0) arima\_forecast = arima\_result.forecast(steps=len(test)) arima\_mse = mean\_squared\_error(test['Price'], arima\_forecast) print(f'ARIMA Mean Squared Error: {arima\_mse}')

* PROPHET

prophet\_model = Prophet() prophet\_model.fit(train.rename(columns={'DateTime': 'ds', 'Price': 'y'})) future = prophet\_model.make\_future\_dataframe(periods=len(test)) prophet\_forecast = prophet\_model.predict(future) prophet\_mse = mean\_squared\_error(test['Price'], prophet\_forecast['yhat'][-len(test):]) print(f'Prophet Mean Squared Error: {prophet\_mse}')

**5.Model Training:**

Fit the selected model to the training dataset using the features and target variable (electricity prices).Perform hyperparameter tuning to optimize the model's performance. You can use techniques like grid search or random search.

MODEL EVALUATION

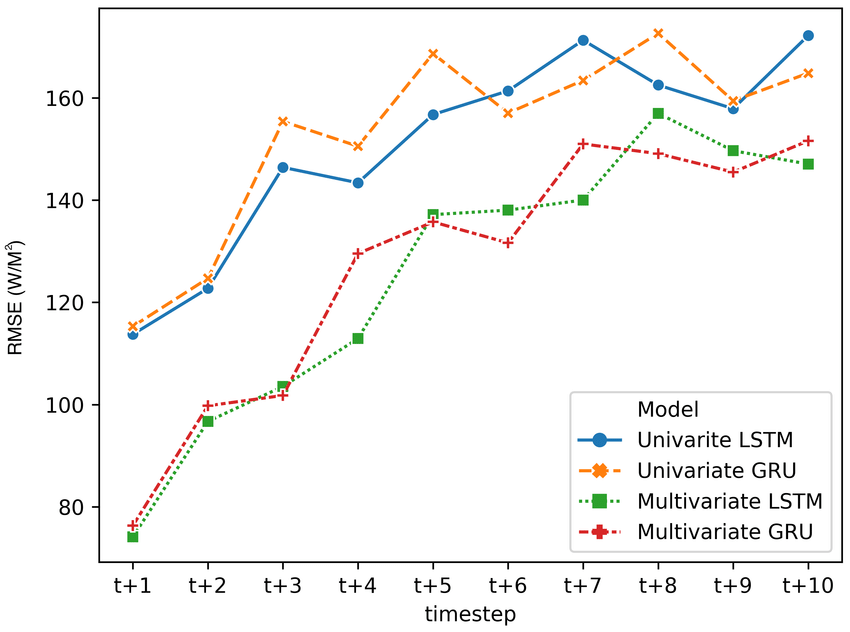
Evaluating the performance of an electricity price prediction model is essential to assess its accuracy and suitability for real-world applications. Several evaluation metrics and techniques can be used, depending on the nature of the problem and the specific goals of your prediction model.

* Mean Absolute Error (MAE):

MAE is the average absolute difference between the predicted and actual prices. It measures the average prediction error in the same units as the target variable.

**MAE = (1/n) \* Σ|actual - predicted**

Lower MAE indicates better model performance.

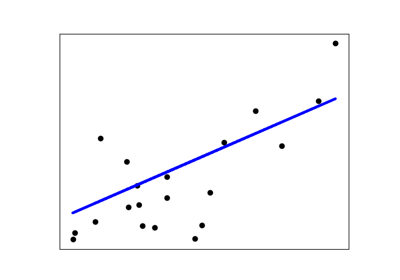


* Mean Squared Error (MSE):

MSE measures the average squared difference between predicted and actual prices. It penalizes larger errors more than MAE.

**MSE = (1/n) \* Σ(actual - predicted)^2**

RMSE (Root Mean Squared Error) is the square root of MSE and provides an interpretable metric in the same units as the target variable.



* Mean Absolute Percentage Error (MAPE):

MAPE calculates the percentage error for each prediction and then takes the average of these errors.

**MAPE = (1/n) \* Σ|(actual - predicted) / actual| \* 100%**

It provides insights into the relative error compared to the actual values.

* R-squared (R²) or Coefficient of Determination:

R² measures the proportion of the variance in the dependent variable (electricity prices) that is predictable from the independent variables (features) in your model.

R² ranges from 0 to 1, where 1 indicates a perfect fit, and 0 indicates no predictive power.

# Assuming you have predictions (y\_pred) and actual prices (y\_true)

y\_true = [100.0, 110.0, 120.0, 130.0, 140.0]

y\_pred = [98.0, 112.0, 118.0, 129.0, 142.0]

# Calculate MAE

mae = mean\_absolute\_error(y\_true, y\_pred)

print(f”Mean Absolute Error (MAE): {mae}")

# Calculate MSE

mse = mean\_squared\_error(y\_true, y\_pred)

print(f"Mean Squared Error (MSE): {mse}")

# Calculate RMSE

rmse = np.sqrt(mse)

print(f"Root Mean Squared Error (RMSE): {rmse}")

# Calculate R-squared (R2)

r2 = r2\_score(y\_true, y\_pred)

print(f"R-squared (R2): {r2}")

**Advantages of electricity price prediction**

Electricity price prediction offers numerous advantages for various stakeholders in the energy industry, as well as for consumers and policymakers. Some of the key advantages of electricity price prediction include:

* Cost Savings: Electricity price prediction enables consumers and businesses to make informed decisions about when to use electricity, allowing them to shift consumption to times when prices are lower. This can lead to significant cost savings on electricity bills.
* Optimized Energy Consumption: Predicting electricity prices helps consumers and businesses optimize their energy consumption patterns. For instance, they can schedule energy-intensive tasks during off-peak hours, reducing the overall demand on the grid.
* Energy Market Efficiency: Energy market participants, such as generators, retailers, and traders, can use price predictions to make better decisions about energy production, purchasing, and trading. This leads to more efficient market operations and potentially higher profits.
* Grid Management: Utilities and grid operators use electricity price forecasts to manage the grid more effectively. This includes balancing supply and demand, reducing grid congestion, and avoiding costly peak demand situations.
* Renewable Energy Integration: Electricity price predictions are crucial for optimizing the integration of renewable energy sources like solar and wind. By forecasting electricity prices, operators can adjust renewable energy generation to coincide with peak price periods, making renewables more economically viable.
* Risk Mitigation: Investors and energy companies can use price predictions to assess the financial risks associated with energy-related projects. This helps in making informed investment decisions and managing price volatility.
* Policy and Regulation: Policymakers and regulators can benefit from electricity price predictions to design and implement effective energy policies and regulations. These forecasts provide insights into market dynamics and the impact of policy changes.
* Environmental Benefits: By optimizing energy consumption and renewable energy integration, electricity price prediction contributes to reducing greenhouse gas emissions. It encourages the use of cleaner energy sources and the reduction of fossil fuel consumption during peak demand periods.
* Resource Allocation: Utilities can allocate resources more efficiently based on predicted electricity prices. This can help them determine the optimal mix of energy sources for power generation, including fossil fuels, renewables, and energy storage.
* Consumer Engagement: Predicted electricity prices can engage consumers by raising awareness of the cost implications of their energy use. This can lead to more responsible energy consumption and increased interest in energy efficiency measures.
* Economic Planning: Electricity price forecasts provide valuable information for businesses and governments to plan for future energy costs. This is particularly important for industries with high energy consumption, such as manufacturing and data centers.
* Grid Resilience: Predictive models can help grid operators anticipate and manage potential disruptions or grid stability issues during periods of high electricity demand and price volatility.

**Disadvantages of electricity price prediction**

While electricity price prediction offers many advantages, it also comes with certain disadvantages and challenges. Some of the notable disadvantages and limitations of electricity price prediction include

* Inaccuracy: Predicting electricity prices is a complex task, and predictions may not always be highly accurate. Variability in factors like weather, demand, and market conditions can lead to errors in forecasts.
* Market Volatility: Energy markets can be highly volatile, making price predictions particularly challenging. Sudden events, such as changes in fuel prices, geopolitical developments, or extreme weather, can lead to unexpected price spikes.
* Data Quality: The accuracy of price predictions heavily depends on the quality and availability of historical and real-time data. Inaccurate or incomplete data can result in unreliable forecasts.
* Model Complexity: Developing accurate prediction models often requires sophisticated algorithms and significant computational resources. Complex models can be challenging to implement, maintain, and interpret.
* Data Requirements: Effective electricity price prediction often demands a large volume of data, including historical price data, weather data, and more. Access to and management of such data can be a logistical challenge.
* Model Overfitting: Overfitting occurs when a prediction model is too complex and fits the training data too closely, resulting in poor generalization to unseen data. Striking the right balance between model complexity and performance can be challenging.
* Lack of Transparency: Some machine learning models, such as deep neural networks, can be seen as "black boxes" with limited interpretability. This can make it difficult to understand why a model makes a particular prediction.
* High Computational Resources: Developing and running complex prediction models can be computationally intensive, potentially requiring high-performance hardware and substantial energy consumption.
* Market Manipulation: Electricity price predictions can be vulnerable to market manipulation, where traders or organizations may attempt to exploit predicted price trends for their benefit.
* Regulatory Changes: Predictive models are based on historical data and may not account for regulatory changes or policy shifts that can impact electricity prices unexpectedly.
* Data Privacy Concerns: Handling large volumes of data can raise data privacy and security concerns. Protecting sensitive information in the energy sector is paramount.
* Behavioral Factors: Predicting consumer behavior is challenging, as it depends on various factors beyond economic considerations. Consumer sentiment, social and cultural factors, and government policies can all affect consumption patterns.
* Model Maintenance: Over time, prediction models may require ongoing maintenance and updating as market dynamics change, and new data becomes available.
* Overreliance on Predictions: Relying too heavily on predictions without considering uncertainties or having contingency plans in place can lead to adverse consequences when forecasts are inaccurate.
* Ethical Considerations: The use of electricity price predictions can raise ethical questions, particularly when it involves influencing consumer behavior or market decisions for financial gain

**CONCLUSION**

Predicting electricity prices is a complex and crucial task, with implications for both consumers and energy providers. In conclusion, accurate electricity price prediction is essential for a variety of stakeholders, including businesses, households, and policymakers. Here are some key points to consider:

**Volatility and Uncertainty**: Electricity prices are subject to a multitude of variables, including supply and demand fluctuations, weather conditions, government policies, and market dynamics. Predicting prices accurately is challenging due to these volatile and uncertain factors.

**Data and Technology**: Advanced data analytics, machine learning, and AI technologies have improved the accuracy of electricity price predictions. These methods can analyze large datasets, identify patterns, and make real-time adjustments, enhancing the quality of forecasts.

**Energy Transition**: The transition to renewable energy sources, such as solar and wind power, adds an additional layer of complexity to price predictions. The intermittent nature of these sources makes it critical to integrate them into forecasting models effectively.

**Risk Mitigation:** Accurate price predictions enable energy market participants to make informed decisions, manage risks, and optimize their energy consumption or generation. This, in turn, can result in cost savings and more efficient energy utilization.

**Market Integration**: Integration of regional and international energy markets is becoming more common. Accurate price predictions are crucial for participants in interconnected markets, allowing them to respond to price differentials and make strategic investments.

**Policy and Regulation**: Government policies and regulations significantly influence electricity prices, such as subsidies, taxes, and environmental standards. Anticipating these changes is vital for energy planning and decision-making.

**Environmental Impact**: Accurate price predictions can encourage sustainable practices by helping consumers choose cleaner, more cost-effective energy sources. This, in turn, can contribute to reducing greenhouse gas emissions and mitigating climate change.

In conclusion, predicting electricity prices is a dynamic and multifaceted endeavor. The integration of advanced technology and data analytics, as well as consideration of factors like renewable energy adoption and policy changes, is essential for achieving more precise forecasts. Accurate predictions empower market participants to make informed decisions, manage risk, and contribute to a more sustainable energy future. However, it's important to recognize that while prediction models can enhance decision-making, there will always be an inherent level of uncertainty in energy markets.