Electricity price prediction is the process of forecasting the future costs of electricity. It is a crucial task for various stakeholders in the energy industry, including consumers, energy suppliers, and grid operators, as it helps them make informed decisions regarding energy consumption, production, and trading.

* **Data Collection**: To make accurate predictions, historical data related to electricity prices is collected. This data typically includes information on electricity demand, supply, weather conditions, and market dynamics.
* **Feature Engineering**: Relevant features, such as time of day, day of the week, season, weather data, and market factors like fuel prices or renewable energy generation, are extracted from the historical data. These features play a crucial role in predicting electricity prices.
* **Model Selection**: Various machine learning and statistical models can be employed for electricity price prediction. Common models include time series forecasting (e.g., ARIMA or LSTM), regression analysis, and machine learning techniques like random forests or gradient boosting.
* **Training and Testing** The historical data is split into training and testing datasets. The model is trained on the historical data and tested to evaluate its accuracy in making predictions.
* **Evaluation and Tuning** The model's performance is assessed using metrics like Mean Absolute Error (MAE) or Root Mean Square Error (RMSE). If the model's performance is not satisfactory, parameters are tuned, or alternative models are considered.
* **Incorporating External Factors** Some predictions may require considering external factors like government policies, economic trends, or geopolitical events that can impact energy markets.
* **Real-Time Data** For short-term predictions, real-time data is crucial. This includes up-to-the-minute information on factors like demand and supply fluctuations, sudden weather changes, and unexpected events.
* **Deploymen**t Once a model proves accurate and reliable, it can be deployed in practice. It can be integrated into energy management systems, trading platforms, or used by consumers to make informed decisions about their energy use.
* **Continuous Monitoring and** **Updating** Electricity price prediction models need to be regularly updated to adapt to changing market conditions, new data, and evolving energy policies.

Accurate electricity price prediction is vital for optimizing energy consumption, managing costs, and promoting the integration of renewable energy sources into the grid, as it allows stakeholders to make well-informed decisions in a dynamic and complex energy market.  
 **DATA SET FROM:**  
  
<https://www.kaggle.com/code/qahramonuktamov2205/electricity-price-prediction>  
**COLUMN DESCRIPTION:**

* **DateTime**: String, defines date and time of sample
* **Holiday**: String, gives name of holiday if day is a bank holiday
* **HolidayFlag**: integer, 1 if day is a bank holiday, zero otherwise
* **DayOfWeek:** integer (0-6), 0 monday, day of week
* **WeekOfYear:** integer, running week within year of this date
* **Day integer**: day of the date
* **Month integer**: month of the date
* **Year integer**: year of the date
* **PeriodOfDay integer**: denotes half hour period of day (0-47)
* **ForecastWindProduction**: the forecasted wind production for this period
* **SystemLoadEA**: the national load forecast for this period
* **SMPEA**: the price forecast for this period
* **ORKTemperature:** the actual temperature measured at Cork airport
* **ORKWindspeed**: the actual windspeed measured at Cork airport
* **CO2Intensity**: the actual CO2 intensity in (g/kWh) for the electricity produced
* **ActualWindProduction**: the actual wind energy production for this period
* **SystemLoadEP2**: the actual national system load for this period  
   **LIBRARIES**
* **Pandas**:Pandas is a powerful library for data manipulation and analysis. You can use it to read datasets from various file formats like CSV, Excel, SQL databases, and more. To download datasets, you may need to use other libraries or websites to fetch data files.

To read a CSV file using Pandas: python

import pandas as pd

df = pd.read\_csv('your\_dataset.csv')

* **NumPy**:NumPy is commonly used for numerical operations. You can use it to create and manipulate arrays, which are often used for storing and working with data.
* **Scikit-learn**:Scikit-learn is a machine learning library that includes several datasets for practice and experimentation. You can load them using the `datasets` module. Example:

Python

from sklearn import dataset

iris = datasets.load\_iris()

* **Seaborn and Matplotlib**: These libraries are commonly used for data visualization, but they also include sample datasets that you can load for plotting and analysis. Example: python

import seaborn as sn

tips = sns.load\_dataset('tips')

To download datasets from the internet, you can use various methods depending on the source. For example, you can use:

The `requests` library to fetch data from a web URL.

Specific dataset loading functions provided by libraries like Scikit-learn, TensorFlow, or PyTorch.

APIs or data sources that provide data access through web requests.  
EXAMPLE:

python

import requests

url = 'https://example.com/your\_dataset.csv'

response = requests.get(url)

with open('your\_dataset.csv', 'wb') as file:

file.write(response.content)

**TRAIN AND TEST**  
  
Training and testing an electricity price prediction model involves several steps, including data preparation, model selection, training, and evaluation. Here's a general outline of the process:

* **Data Collection**: Gather historical data related to electricity prices. This dataset should include features such as time, weather conditions, demand, and any other relevant factors.
* **Data Preprocessing:**

- Clean the data by handling missing values and outliers.

- Feature engineering: Create new features if necessary or transform existing ones.

- Normalize or scale the data to ensure that all features have the same scale.

- Split the data into training and testing sets.

* **Choose a Model:** Select a suitable machine learning model for electricity price prediction. Common choices include linear regression, decision trees, random forests, support vector machines, and neural networks. The choice of the model depends on the complexity of the problem and the characteristics of your dataset.
* **Training the Model**

- Fit the selected model to the training data.

- Tune hyperparameters for the model (e.g., learning rates, regularization parameters) using techniques like cross-validation.

- Train the model on the training data.

* **Testing the Model**:

- Use the trained model to make predictions on the test dataset.

- Evaluate the model's performance using appropriate metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or R-squared (R2).

**Example:** (using scikit-learn for a simple linear regression model):

python

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

model = LinearRegression()

model.fit(X\_train, y\_train) # X\_train and y\_train are your training data

y\_pred = model.predict(X\_test) # X\_test is your test data

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

* **Hyperparameter Tuning** Fine-tune the model's hyperparameters to optimize its performance. You can use techniques like grid search or random search to find the best hyperparameters.
* **Model Deployment** Once you're satisfied with the model's performance, you can deploy it to make real-time predictions or use it in a production environment.
* **Monitoring and Maintenance** Continuously monitor the model's performance and retrain it periodically with new data to keep it accurate.
* **Visualization and Interpretation**: Visualize the model's predictions and explore the importance of features to gain insights into the factors affecting electricity prices.
* **Documentation** Document your model, including the preprocessing steps, model details, and performance metrics, so that others can understand and use it.

**ACCURACY CHECK**  
. Here are some commonly used metrics:

* **Mean Absolute Error (MAE)**: MAE measures the average absolute difference between the actual and predicted values. It gives you an idea of the model's accuracy in terms of the actual electricity prices.

Python

from sklearn.metrics import mean\_absolute\_error

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

* **Mean Squared Error (MSE)** MSE measures the average of the squared differences between actual and predicted values. It penalizes larger errors more heavily than MAE.

Python

from sklearn.metrics import mean\_squared\_error

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

* **Root Mean Squared Error (RMSE):** RMSE is the square root of MSE. It's often used because it's in the same units as the target variable (electricity price) and provides a more interpretable error metric.

Python

import numpy as np

rmse = np.sqrt(mean\_squared\_error(y\_true, y\_pred))

* **R-squared (R2)** R-squared measures the proportion of the variance in the dependent variable (electricity price) that is predictable from the independent variables (features). A higher R2 indicates a better model fit.

Python

from sklearn.metrics import r2\_score

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

* **Explained Variance Score** This metric quantifies the proportion of variance in the target variable that the model explains.

Python

from sklearn.metrics import explained\_variance\_score

explained\_variance = explained\_variance\_score(y\_true, y\_pred)

* **Mean Absolute Percentage Error (MAPE):** MAPE calculates the average percentage difference between the actual and predicted values. It is useful when you want to understand the error in terms of percentage.

Python

def mean\_absolute\_percentage\_error(y\_true, y\_pred):

return np.mean(np.abs((y\_true - y\_pred) / y\_true)) \* 100

mape = mean\_absolute\_percentage\_error(y\_true, y\_pred)

The choice of metric depends on the specific context and requirements of your electricity price prediction task. It's a good practice to use a combination of these metrics to get a comprehensive view of your model's performance. For example, while RMSE and MAE indicate prediction errors in price units, R-squared and explained variance provide insights into the overall model fit.