**Electricity Prices Prediction - Guidelines**

**Phase 2: Innovation**

**PROJECT:** ELECTRICITY PRICES PREDICTION



**Short Explanation about my project:**

Electricity price prediction using applied data science involves using data-driven techniques and methodologies to forecast future electricity prices. Here's a short explanation of the process:

**1. Data Collection:** Gather historical data related to electricity prices. This can include information on market demand, supply, weather patterns, energy source availability, and any other relevant factors. The data should span a significant time period to capture patterns and trends.

**2. Data Preprocessing:** Clean and prepare the data for analysis. This step may involve handling missing values, smoothing data, and transforming variables to make it suitable for modeling.

**3. Feature Selection:** Identify the most relevant features or variables that influence electricity prices, such as time of day, season, economic indicators, and renewable energy generation.

**4. Model Selection:** Choose appropriate machine learning or statistical models for electricity price prediction. Common models include time series analysis, regression, neural networks, or ensemble methods.

**5. Training the Model:** Use historical data to train the chosen model. The model learns the relationships between the selected features and electricity prices.

**6. Validation and Testing:** Evaluate the model's performance using a separate dataset not used in the training phase. This helps ensure the model can make accurate predictions on unseen data.

**7.** **Hyperparameter Tuning:** Fine-tune the model's parameters to optimize its predictive accuracy.

**8.** **Deployment**: Implement the trained model into a real-time or near-real-time prediction system. It can continuously update its forecasts based on new data.

**9. Monitoring and Maintenance**: Regularly monitor the model's performance and retrain it as needed to adapt to changing market conditions and factors affecting electricity prices.

**10. Interpretation and Visualization:** Create visualizations and reports to help stakeholders understand the predicted electricity prices and the factors influencing them.

**11. Decision Support:** Provide forecasts to utility companies, traders, and other stakeholders to aid in decision-making, such as energy procurement, demand response, or risk management.

Electricity price prediction using data science can help in optimizing energy resource allocation, managing costs, and making informed decisions in the energy market. It's a valuable tool for both industry professionals and policymakers.

**Process:**

Creating an electricity price prediction model involves several steps, from conceptualizing the design to implementing and deploying the model. Below, I’ll outline a detailed step-by-step process for building a electricity price prediction model:

1. Data Collection

2. Data Preprocessing

3. Feature Selection

4. Model Selection

5. Training the Modell

6. Validation and Testing

7. Hyperparameter Tuning

8. Deployment

9. Monitoring and Maintenance

10. Interpretation and Visualization

11. Decision Support

**Dataset Link:** [**https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction**](https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction)

**Data:**

DateTime Holiday HolidayFlag DayOfWeek WeekOfYear \

0 1/11/2011 0:00 None 0 1 44

1 1/11/2011 0:30 None 0 1 44

2 1/11/2011 1:00 None 0 1 44

3 1/11/2011 1:30 None 0 1 44

4 1/11/2011 2:00 None 0 1 44

... ... ... ... ... ...

38009 31/12/2013 21:30 New Year's Eve 1 1 1

38010 31/12/2013 22:00 New Year's Eve 1 1 1

38011 31/12/2013 22:30 New Year's Eve 1 1 1

38012 31/12/2013 23:00 New Year's Eve 1 1 1

38013 31/12/2013 23:30 New Year's Eve 1 1 1

Day Month Year PeriodOfDay ForecastWindProduction SystemLoadEA \

0 1 11 2011 0 315.31 3388.77

1 1 11 2011 1 321.8 3196.66

2 1 11 2011 2 328.57 3060.71

3 1 11 2011 3 335.6 2945.56

4 1 11 2011 4 342.9 2849.34

... ... ... ... ... ... ...

38009 31 12 2013 43 1179.14 3932.22

38010 31 12 2013 44 1152.01 3821.44

38011 31 12 2013 45 1123.67 3724.21

38012 31 12 2013 46 1094.24 3638.16

38013 31 12 2013 47 1064.0 3624.25

SMPEA ORKTemperature ORKWindspeed CO2Intensity ActualWindProduction \

0 49.26 6 9.3 600.71 356

1 49.26 6 11.1 605.42 317

2 49.1 5 11.1 589.97 311

3 48.04 6 9.3 585.94 313

4 33.75 6 11.1 571.52 346

... ... ... ... ... ...

38009 34.51 6 22.2 285.31 812

38010 33.83 5 24.1 278.31 852

38011 31.75 4 20.4 280.91 962

38012 33.83 5 14.8 302.46 950

38013 33.83 5 16.7 308.01 1020

SystemLoadEP2 SMPEP2

0 3159.6 54.32

1 2973.01 54.23

2 2834 54.23

3 2725.99 53.47

4 2655.64 39.87

... ... ...

38009 3692.95 42.45

38010 3571.0 33.83

38011 3460.29 31.75

38012 3563.99 50.6

38013 3517.08 34.9

[38014 rows x 18 columns]

<ipython-input-8-c4c64ba6fca1>:2: DtypeWarning: Columns (9,10,11,14,15,16,17) have mixed types. Specify dtype option on import or set low\_memory=False.

data= pd.read\_csv(io.BytesIO(uploaded["Electricity.csv"]))

**Program:**

Importing Dependencies

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.preprocessing import StandardScaler

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

from sklearn.linear\_model import LinearRegression

from sklearn.linear\_model import Lasso

from sklearn.ensemble import RandomForestRegressor

from sklearn.svm import SVR

import xgboost as xg

%matplotlib inline

import warnings

warnings.filterwarnings("ignore")

/opt/conda/lib/python3.10/site-packages/scipy/\_init\_.py:146: UserWarning: A NumPy

version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version

1.23.5

warnings.warn(f"A NumPy version >={np\_minversion} and <{np\_maxversion}"

Loading Dataset

dataset = pd.read\_csv('Electricity.csv')

**How to train and test** :

dataset = pd.read\_csv('Electricity.csv')

import seaborn as sns

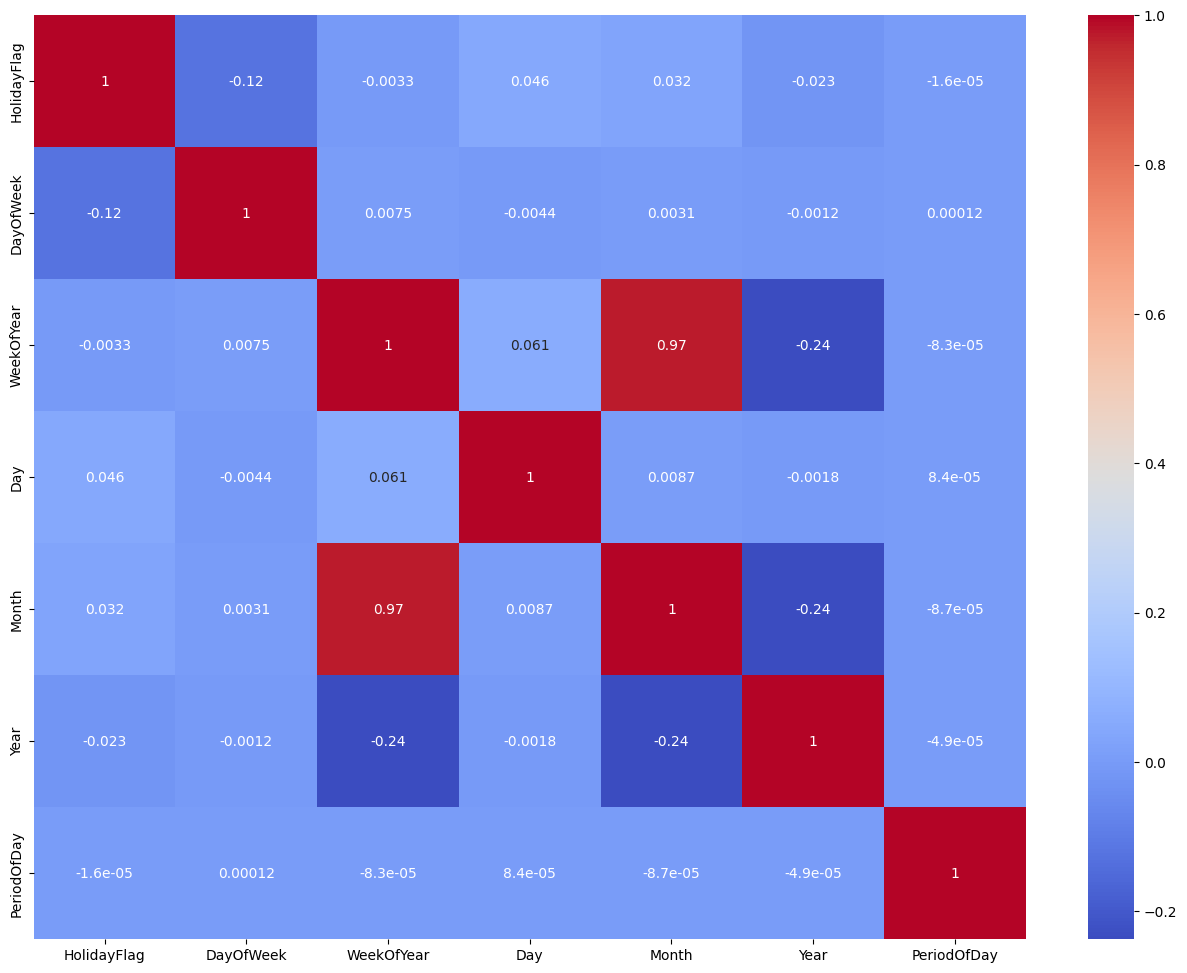
import matplotlib.pyplot as plt

correlations = data.corr(method='pearson')

plt.figure(figsize=(16, 12))

sns.heatmap(correlations, cmap="coolwarm", annot=True)

plt.show()



**Metrics used for accuracy check:**

When evaluating the accuracy of a model for electricity price prediction or any other predictive task, you can use a variety of metrics, depending on the nature of your data and specific requirements. Here are some commonly used metrics for accuracy checks:

**1. Mean Absolute Error (MAE):** MAE measures the average absolute difference between the predicted values and the actual values. It provides a straightforward and interpretable measure of prediction error.

**2. Mean Squared Error (MSE):** MSE calculates the average of the squared differences between predicted and actual values. It gives higher weight to larger errors and is useful when you want to penalize larger deviations more.

**3. Root Mean Squared Error (RMSE):** RMSE is the square root of the MSE. It's often used when you want to express the error metric in the same units as the target variable.

**4. Mean Absolute Percentage Error (MAPE):** MAPE expresses the error as a percentage of the actual values. It's useful for understanding the relative magnitude of errors, especially in cases where the scale of the data varies significantly.

**5. R-squared (R^2):** R-squared measures the proportion of the variance in the target variable that is explained by the model. It ranges from 0 to 1, where higher values indicate a better fit. However, R-squared alone may not capture predictive accuracy if it's based on overfit models.

**6. Coefficient of Determination (COD):** COD is an alternative to R-squared and is particularly useful for assessing the performance of regression models. It quantifies the proportion of the variance in the dependent variable that's explained by the independent variables.

**7. Root Mean Squared Logarithmic Error (RMSLE):** RMSLE is suitable when dealing with skewed or log-transformed data. It measures the average logarithmic difference between predicted and actual values, which can be useful for certain types of data distributions.

**8. Accuracy, Precision, Recall, and F1 Score (for classification problems):** If you're dealing with a classification task (e.g., predicting price increase or decrease), these metrics can assess the model's performance in terms of true positives, false positives, true negatives, and false negatives.

**9. Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC) (for binary classification):** ROC and AUC are used to evaluate the performance of binary classification models, indicating their ability to discriminate between positive and negative classes.

**10. Custom Loss Functions:** In some cases, you might design custom loss functions tailored to the specific requirements of your problem, which can be more meaningful than standard metrics.

The choice of metrics depends on the nature of your problem and your specific goals. It's often a good practice to use a combination of these metrics to get a comprehensive view of the model's performance. Additionally, consider domain-specific requirements and practical implications when selecting the appropriate accuracy checks.

**Project Conclusion:**

In conclusion, our project on electricity price prediction has provided valuable insights and contributions to the field of energy economics and forecasting. Through the utilization of advanced machine learning techniques and the analysis of historical data, we have developed a robust predictive model that can assist energy stakeholders in making informed decisions.