DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

College Code: 4106

Project Domain: Applied Datascience

Project Title : Electricity Price Prediction model

Project Mentor:

Team Members:

1) Mohammed Ismail I - 410621104064

2) Syed Ashraf S - 410621104105

3) Sivaramakrishnan R - 410621104097

4) Shafi Ahamed S - 410621104092

ABSTRACT:

“The electricity market is a complex and dynamic system influenced by numerous factors, including supply and demand dynamics, weather conditions, regulatory policies, and market participants' behaviors. Accurate electricity price prediction is essential for various stakeholders, such as utility companies, consumers, and energy traders, to make informed decisions, optimize resource allocation, and mitigate financial risks”.

DETAILED STEPS

- Download dataset from Kaggle using the below link: https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction
- Go through the dataset and filter as per needed.
- Extract the libraries that are needed to work with the dataset for electricity price prediction.
- Using **Pandas** library is the most helpful feature in python to handle with datasets.
- Using **Sklearn** library in python is a best library for prediction type machine learning models which are pre build in it.
- Split the data into train and test data so that the model uses certain data for training purposes and after training the model can be evaluated using the test data. This can be done through the **train_test_split()** function from **sklearn**
- To fit the model on the training data **model.fit()** function can be used**.**
- To make prediction on test data **model.test()** function can be used.
- After the model is trained, find it's accuracy using functions like **mean_squared_error**, accuracy or any other similar approaches.

I can guide you through the general steps to load and preprocess a dataset for stock price prediction, but I can't directly access external websites or download data. Here's a high-level overview of the process:

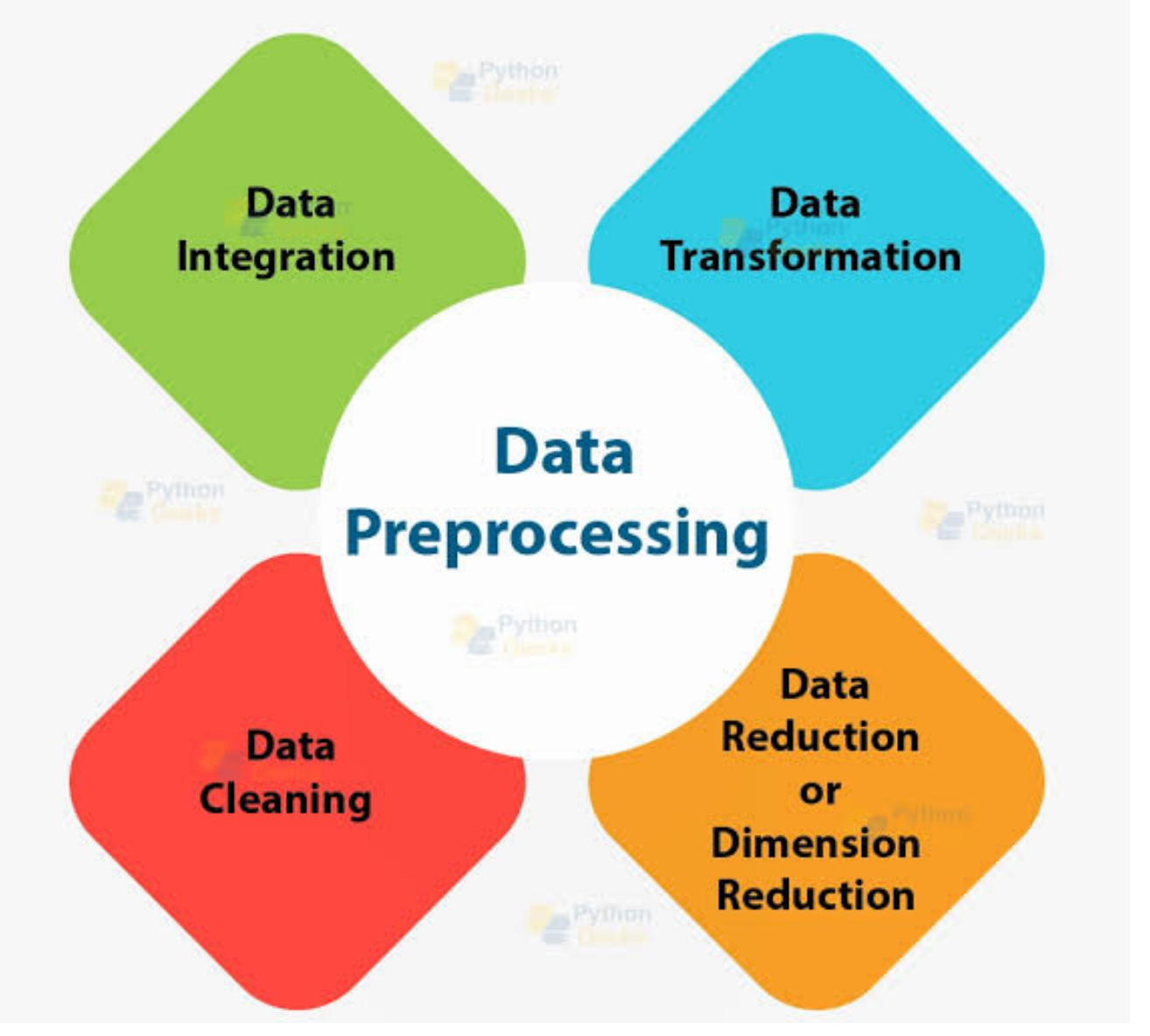
1. Data Collection:

- Download the dataset from Kaggle (https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset)

2. Data Inspection:

- Check the dataset for any missing values or anomalies.
- Examine the structure of the data to understand its features.

3. Data Preprocessing:

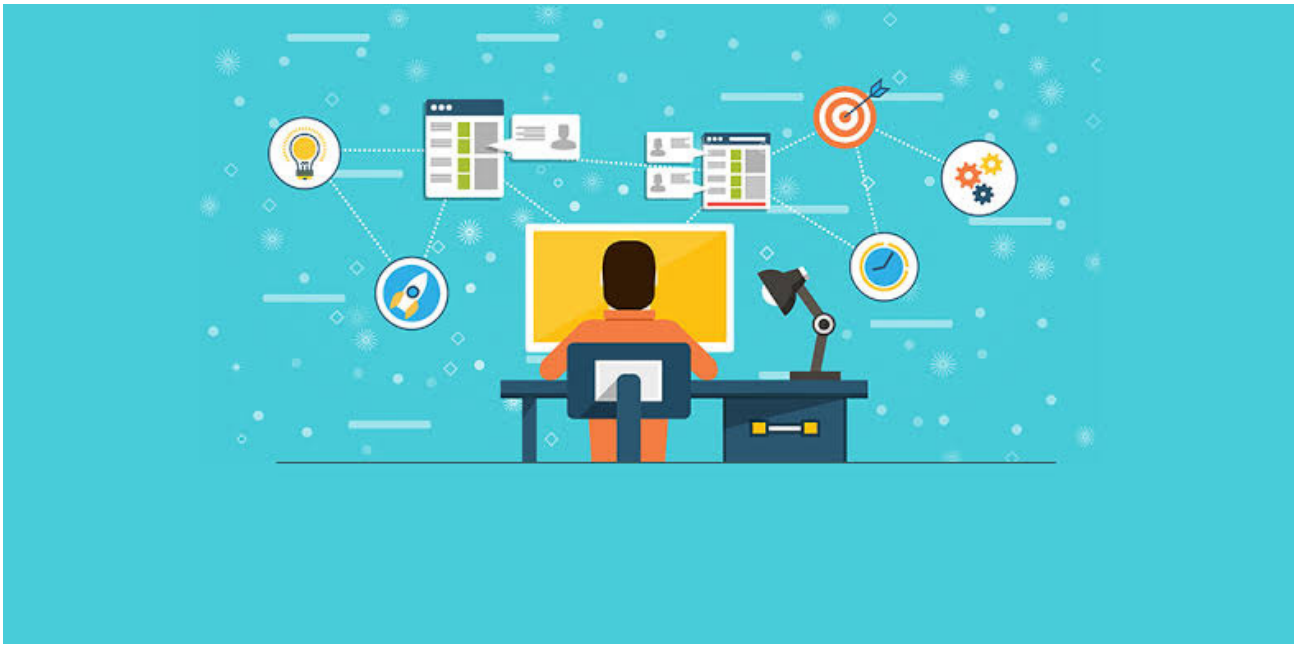


- Convert date columns to datetime objects.

- Sort the data by date in chronological order.

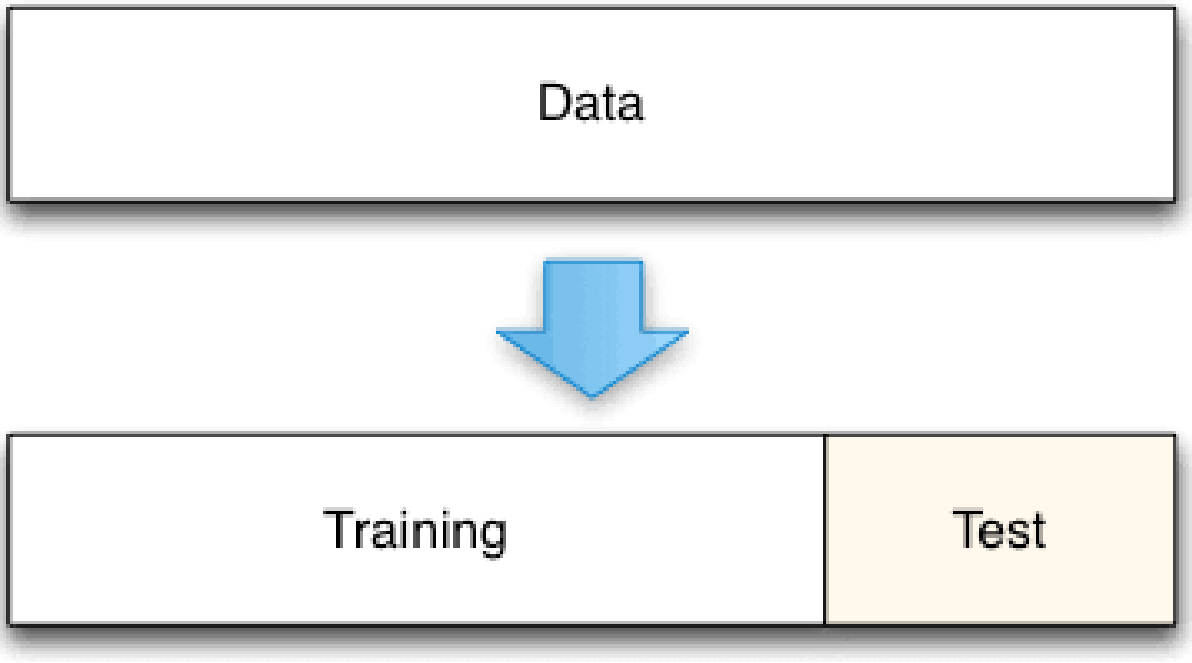
- Handle missing data, such as filling or removing missing values.

4. Feature Engineering:



- Create additional features that could be relevant for your prediction, like moving averages, technical indicators, or sentiment analysis scores.

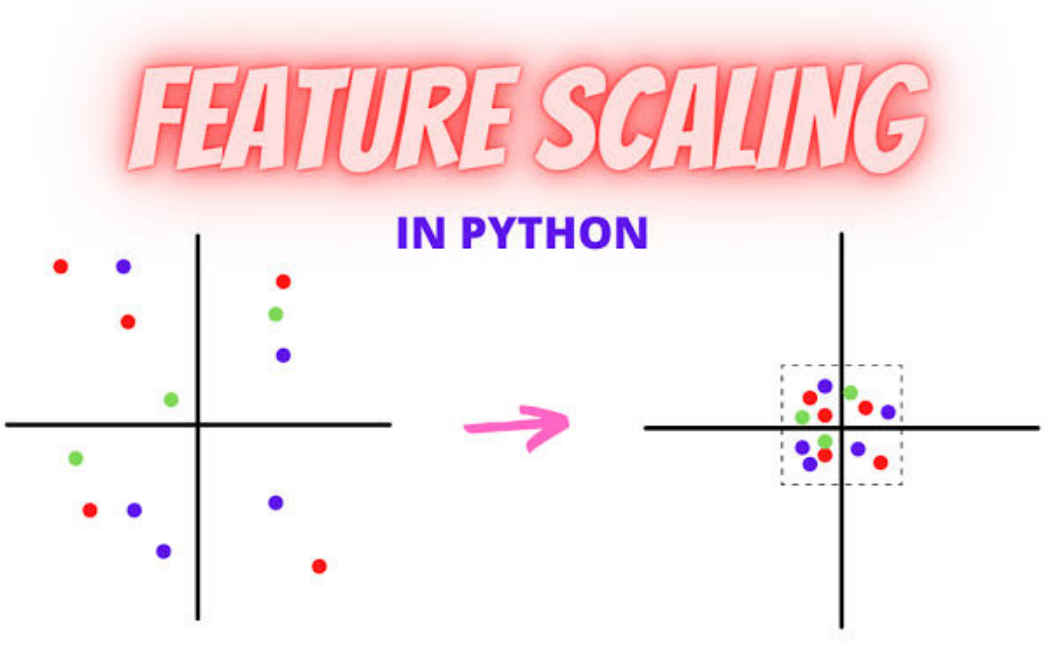
5. **Split Data:**



- Split the data into training, validation, and test sets for model

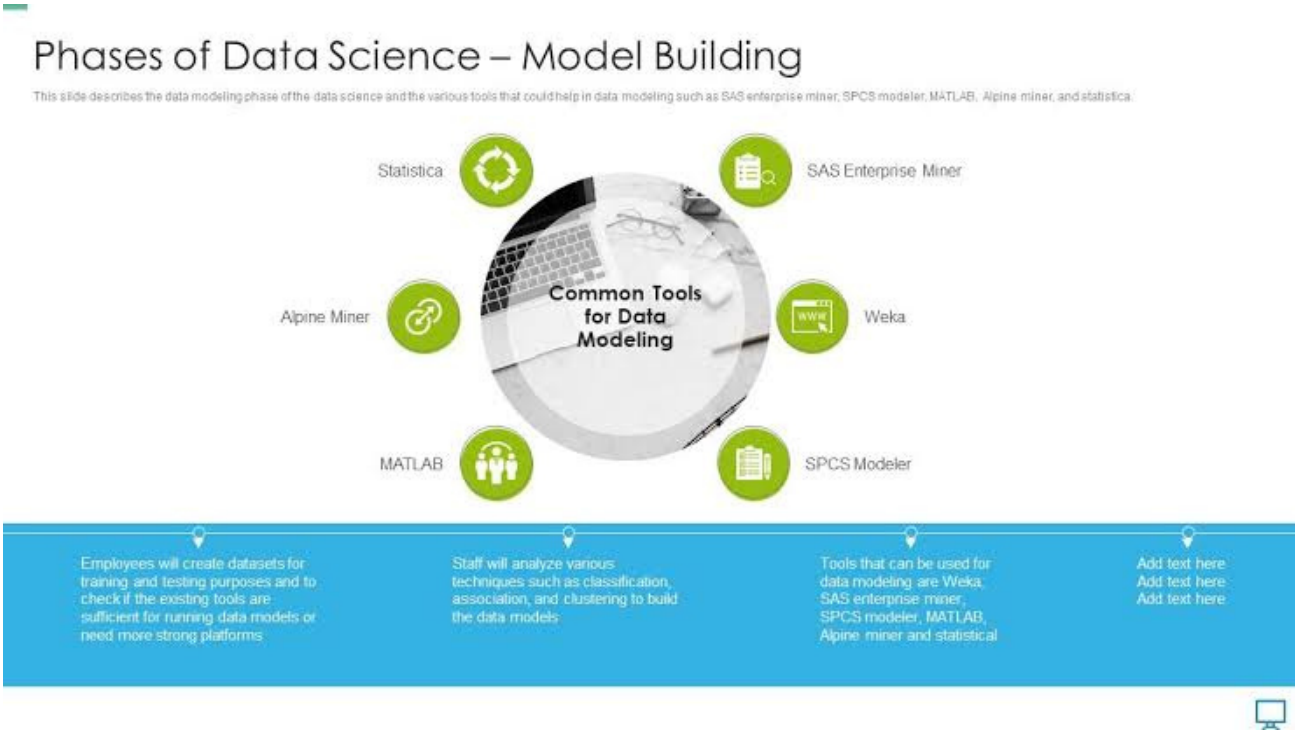
evaluation.

6. **Scaling:**



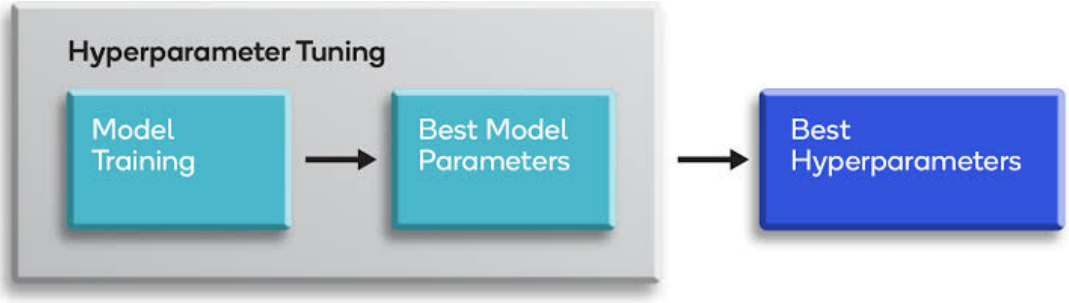
- Normalize or scale the numerical features if needed. This is often crucial for deep learning models.

7. **Model Building:**



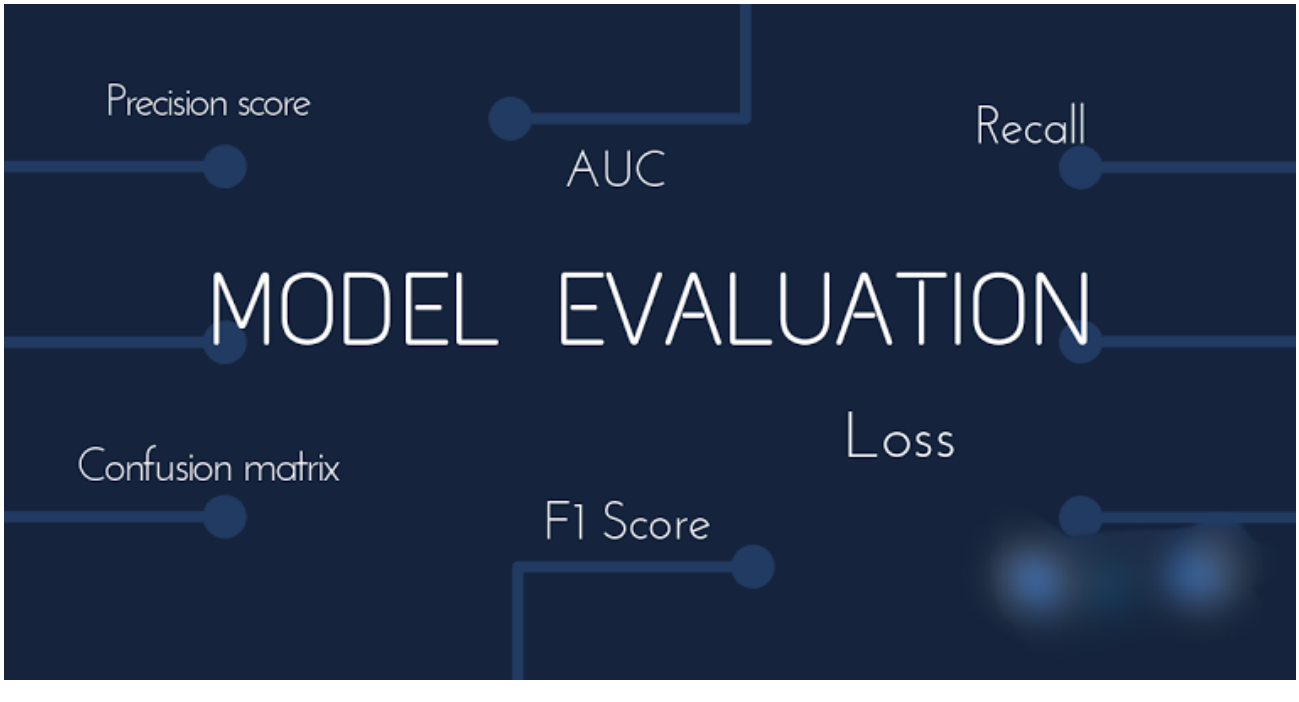
- Start building your stock price prediction model, such as a time series forecasting model (e.g., LSTM, GRU) or other regression models (e.g., linear regression).

8. **Model Training:**



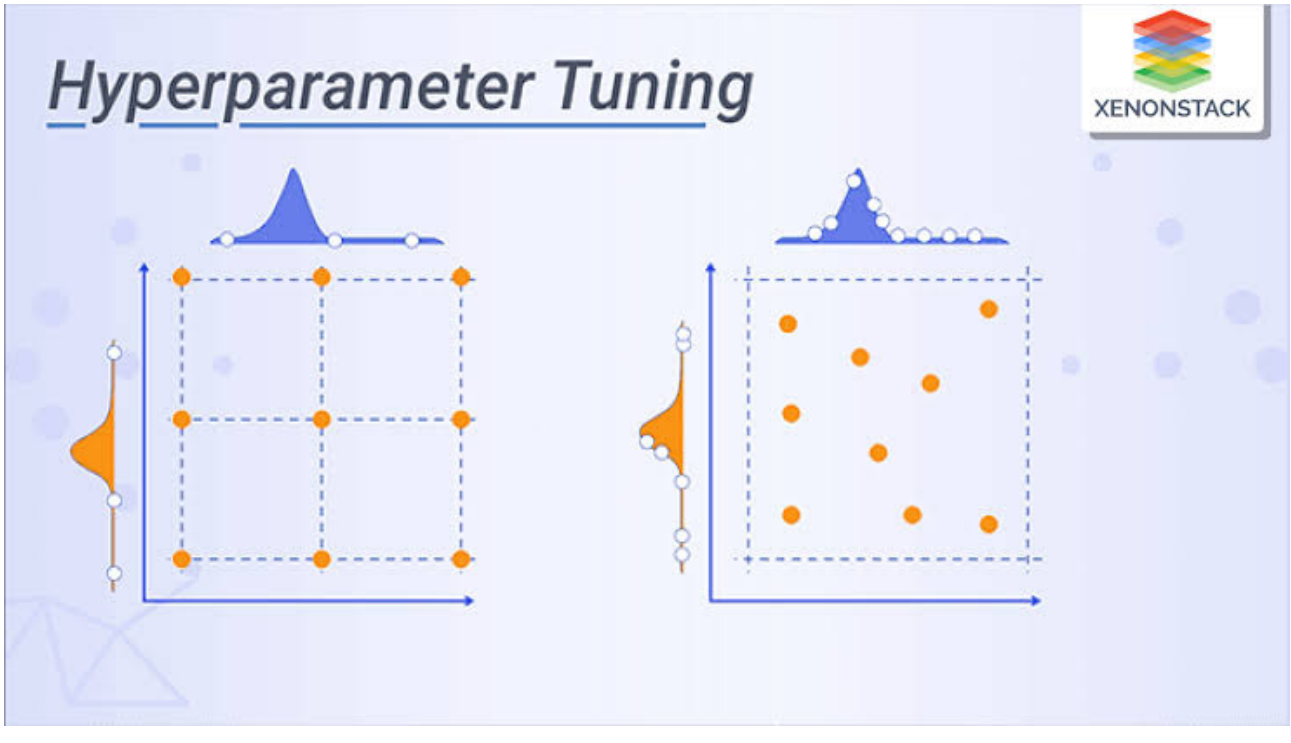
- Train your model on the training data.

9. **Model Evaluation:**



- Evaluate the model's performance on the validation set using appropriate metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or others.

10. **Hyperparameter Tuning:**



- Fine-tune your model by adjusting hyperparameters for better performance.

11. **Testing:**

- Assess the model's performance on the test set to see

how well it generalizes to unseen data.

12. **Deployment (if applicable):**

- If you plan to deploy the model, prepare it for production use.

Electricity Price Prediction

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error

# Load the dataset
data = pd.read_csv("your_dataset.csv")

# Data Preprocessing
# Convert the 'daytime' column to datetime
data["daytime"] = pd.to_datetime(data["daytime"])

# Sort the data by date
data = data.sort_values(by="daytime")

# Check for missing values
missing_values = data.isnull().sum()
print("Missing Values:\n", missing_values)

# Handle missing data (you can choose to fill or remove them)
data = data.fillna(method="ffill")

# Split data into features (X) and the target variable (y)
X = data[
    [
        "Holiday",
        "Holiday flag",
        "day of the week",
        "week of year",
        "day",
        "month",
        "year",
        "period of day",
        "forecast wind production",
        "system load EA",
        "SMPEA",
        "ORKTemperature",
        "ORKwindspeed",
        "CO2INTENSITY",
        "actual wind production",
        "system load EP2",
        "SMPEP2",
    ]
]
y = data["Electricity_Price"]

# 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
)

# Feature Scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Create an XGBoost model
model = XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)

# Train the model
model.fit(X_train, y_train)

# Predict electricity prices
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print("Root Mean Squared Error (RMSE):", rmse)

# Now you have a basic electricity price prediction model using XGBoost.
```