**ADS\_Phase 4:Development Part 2**

**APPLIED DATA SCIENCE – ELECTRICITY PRICES PREDICTION**

### Objective:

The objective of this phase is to continue building the electricity prices prediction model by focusing on feature engineering, model training, and model evaluation.

### **Feature Engineering:**

**Data Preparation:**

* Verify dataset cleanliness and structure.
* Handle missing values and outliers.

Python code

# Assuming 'data' is your dataset

data.info() # Check for missing values

data.describe() # Explore data statistics

**Feature Selection:**

Select relevant features.

Python code:

# Assuming 'data' is your dataset

selected\_features = data[['Feature1', 'Feature2', ...]]

3. **emporal Features:**

-Create time-related features.

Python code:

data['DateTime'] = pd.to\_datetime(data['DateTime'])

data['Hour'] = data['DateTime'].dt.hour

data['DayOfWeek'] = data['DateTime'].dt.dayofweek

data['Month'] = data['DateTime'].dt.month

**4. Lagged Features:**

Generate lag features to capture time dependencies.

Python code:

data['Price\_Lagged\_1'] = data['ElectricityPrice'].shift(1)

data['Price\_Lagged\_7'] = data['ElectricityPrice'].shift(7)

**5. Weather Data:**

Integrate historical weather data.

Python code:

# Assuming you have a weather dataset

data = pd.merge(data, weather\_data, on='DateTime', how='left')

**6. Events:**

- Include features representing events.

Python code:

data = pd.merge(data, holiday\_data, on='DateTime', how='left')

**7.Normalization and Scaling:**

- Normalize or scale features.

Python code:

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

data['ScaledFeature'] = scaler.fit\_transform(data[['Feature']])

**Model Training:**

**1. Select Algorithms:**

- Choose regression algorithms (e.g., Linear Regression, Random Forest, XGBoost).

python code:

from sklearn.linear\_model import LinearRegression

model = LinearRegression()

**2. Split Data:**

- Divide the dataset into training, validation, and test sets.

Python code:

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

X = data[selected\_features]

Y = data['ElectricityPrice']

X\_train, X\_temp, Y\_train, Y\_temp = train\_test\_split(X, Y, test\_size=0.3, random\_state=42)

X\_val, X\_test, Y\_val, Y\_test = train\_test\_split(X\_temp, Y\_temp, test\_size=0.5, random\_state=42)

**3. Model Training:**

- Train the selected model using the training data.

Python code:

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

**Evaluation:**

**1. Validation Set Evaluation:**

- Assess the model's performance on the validation set.

Python code:

Y\_val\_pred = model.predict(X\_val)

# Evaluate using appropriate regression metrics

**2. Test Set Evaluation:**

- Evaluate the model on an independent test set.

Python code:

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

# Evaluate using appropriate regression metrics

**3. Interpretability:**

- Interpret model predictions and feature importance.

Python code:

# For linear models

feature\_importance = model.coef\_

# For tree-based models

feature\_importance = model.feature\_importances\_

**Conclusion:**

In Phase 4, we continued to enhance the electricity prices prediction model by performing feature engineering, model training, and evaluation. These steps are crucial for improving the model's accuracy and reliability. The project is progressing well, and we're prepared for further phases and model deployment.