Phase 3

Development Part 1

In This part you will begin building your project by loading pre-processing the dataset

Import necessary libraries with dataset:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Load dataset from CSV file
data = pd.read_csv("electricity.xlsx")
```

Exploring data analysis:

Explore the dataset to get an understanding of its structure and characteristics.

```
# Display the first few rows of the dataset
print("Original Dataset:")
print(data.head())

# Preprocessing

# Handling Missing Values (if any)
data.dropna(inplace=True)  # Drop rows with missing values
```

Data pre-processing:

Data pre-processing is a critical step that involves handling missing values, encoding categorical variables, and scaling/normalizing numerical features.

```
# Encoding Categorical Variables (if any)
label_encoders = {}  # Dictionary to store label encoders for categorical columns
categorical_columns = ['categorical_column1', 'categorical_column2']  # List of categorical columns

for col in categorical_columns:
    label_encoders[col] = LabelEncoder()
    data[col] = label_encoders[col].fit_transform(data[col])
```

Splitting the dataset:

It involves machine learning, split your dataset into training and testing sets. This is crucial for model evaluation

```
for col in categorical_columns:
    label_encoders[col] = LabelEncoder()
    data[col] = label_encoders[col].fit_transform(data[col])

# Splitting into Features (X) and Target (y)
X = data.drop('target_column', axis=1)  # Features
y = data['target_column']  # Target variable
```

Building and training the models:

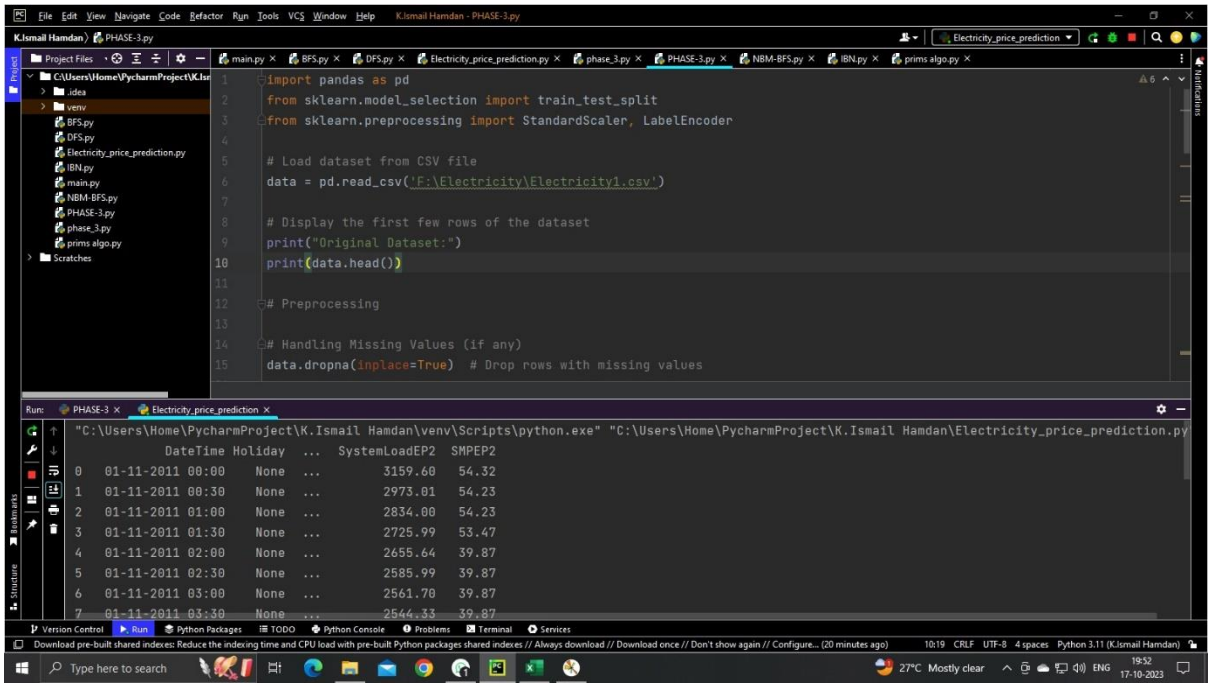
Depending on the project, we might build and train machine learning models using libraries like "Scikit-Learn or deep learning frameworks like TensorFlow or PyTorch".

```
# Splitting 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)

# Display preprocessed data
print("\nPreprocessed Dataset:")
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

After training the model, evaluate its performance using appropriate metrics

Output:



Team Members:

Team Leader: k.Ismail Hamdan

Team Member: H.Izhan Ur Rahman

Team Member:P.Mohammed zakwan

Team Member:K.A Hashir kaamran

Team Member:G.Mohammed Faheem