Phase 3
Development part 1

This section will involve loading and pre-processing the dataset as you start to construct your project.

Import datasets and the necessary libraries:

```
1  import pandas as pd
2  from sklearn.model_selection import train_test_split
3  from sklearn.preprocessing import StandardScaler, LabelEncoder
4
5  data = pd.read_csv('F:\Electricity\Electricity1.csv')
```

Exploring data analysis:

Investigate the dataset to learn about its properties and structure.

```
8  print("Original Dataset:")
9  print(data.head())
```

Handling the missing data:

When dealing with missing data, data scientists can use two primary methods to solve the error: Imputation or the removal

of data. The imputation method develops reasonable guesses for missing data. It’s most useful when the percentage of missing data. It’s most useful when the percentage of missing data is low

```
10
11  data.dropna(inplace=True)
12
```

Encoding Categorical Data:

Data Encoding is an important pre-processing step in Machine Learning. It refers to the process of converting categorical or textual data into numerical format, so that it can be used as input for algorithms to process.

```
13  label_encoders = {}
14  categorical_columns = ['categorical_column1', 'categorical_column2']
15
16  for col in categorical_columns:
17      label_encoders[col] = LabelEncoder()
18      data[col] = label_encoders[col].fit_transform(data[col])
```

Splitting the dataset:

Data Splitting is when data is divided into two or more subsets. Typically, with a two-part split, one part is used to evaluate or

test the data and the other to train the model. Data splitting is an important aspect of data science, particularly for creating models based on data.

```
X = data.drop('target_column', axis=1)
y = data['target_column']

scaler = StandardScaler()
X = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Feature Scaling:

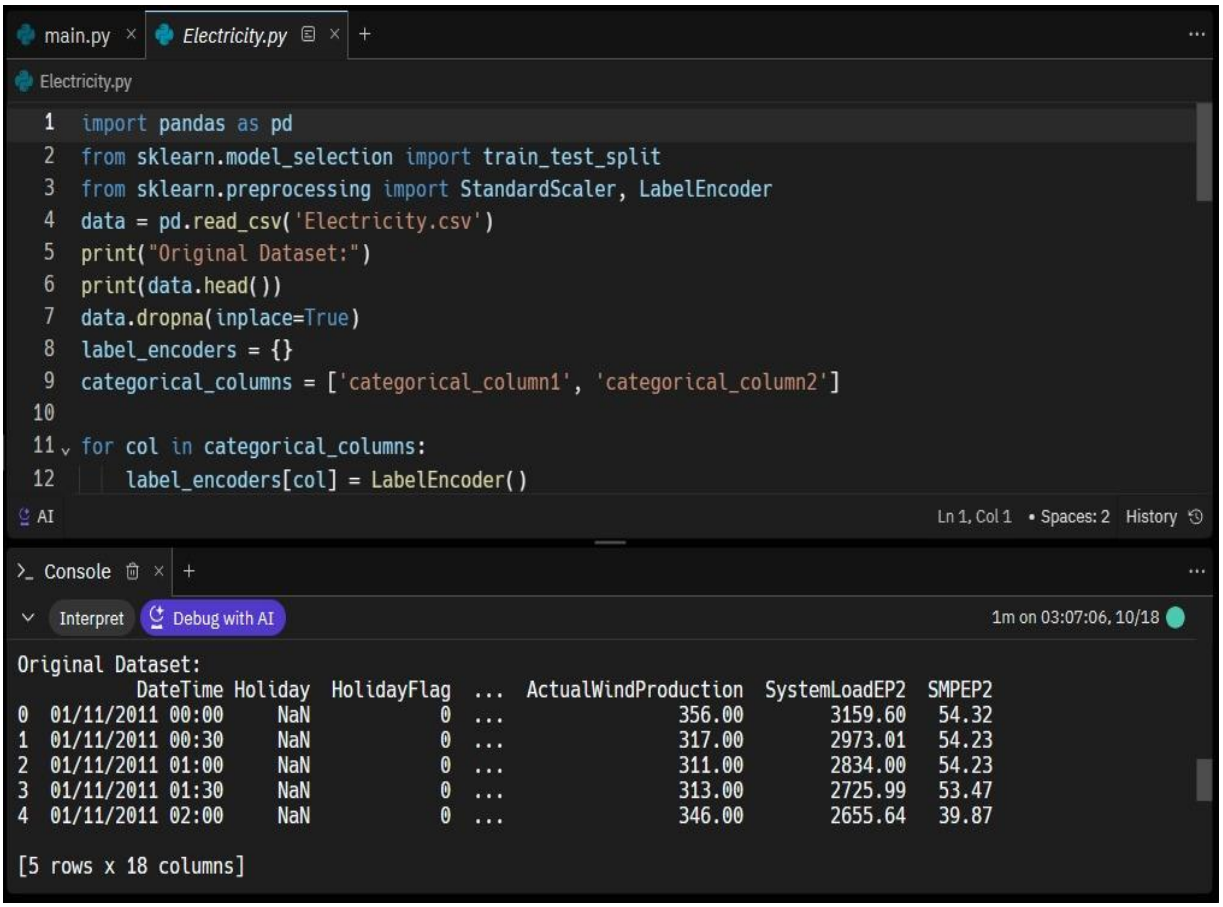
Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.

```

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)

```

Output:



Team Leader: Ismail Hamdan K

Team Member: Hashir Kaamran K.A

Team Member: Izhan Ur Rahman H

Team Member: Mohammed Faheem G

Team Member: Mohammed Zakwan P