***Phase 5: Conclusion***

**Objective:-**

To create a data science model based on electricity price prediction.

**Analysis approach:-**

Plan the steps to extract, clean, and analyze the dataset to derive insights.

**Visualization:-**

* Create a line plot of the predicted electricity price
* Create a histogram of the predicted electricity price
* Create a boxplot of the predicted electricity price by day of the week
* Create a heatmap of the predicted electricity price by hour of the day and day of the week

**Code:-**

1. **Initialisation:-**

#Loading the dataset into python

import pandas as pd

df = pd.read\_csv(‘electricity\_price\_prediction.csv’)

#Exploring the dataset

# Display the first few rows of the dataset

print(“First 5 rows of the dataset:”)

print(df.head())

# Display the shape of the dataset (number of rows, number of columns)

print(“\nShape of the dataset:”)

print(df.shape)

# Display summary statistics of numerical columns

print(“\nSummary statistics:”)

print(df.describe())

# Check for missing values in each column

print(“\nMissing values:”)

print(df.isnull().sum())

# Check data types of each column

print(“\nData types:”)

print(df.dtypes)

# Explore unique values in categorical columns

categorical\_columns = [‘column1’, ‘column2’]

# Replace with actual column names from your dataset

for col in categorical\_columns:

print(f”\nUnique values in {col}:”)

print(df[col].unique())

#Cleaning the dataset Check for missing values

missing\_values = df.isnull().sum()

print(missing\_values)

# Handle missing values

df = df.dropna() # Drop rows with missing values

# Verify if missing values have been handled

missing\_values\_after\_handling = df.isnull().sum()

print(missing\_values\_after\_handling)

df = df.drop\_duplicates()

# Verify if duplicate rows have been removed

duplicates\_removed = df.duplicated().sum()

print(duplicates\_removed)

# Convert columns to appropriate data types (if needed)

df[‘date’] = pd.to\_datetime(df[‘date’])

# Remove unnecessary columns (if any)

df = df.drop([‘column\_name’], axis=1)

# Perform other cleaning and transformation operations as required

df.to\_csv(‘cleaned\_dataset.csv’, index=False)

#Perform data analysis

import matplotlib.pyplot as plt

# Perform data analysis tasks here...

# For example, you can calculate statistics, visualize data, etc.

# Calculate descriptive statistics

statistics = df.describe()

print(statistics)

# Visualize data using a line plot

plt.plot(df[‘Date’], df[‘Price’])

plt.xlabel(‘Date’)

plt.ylabel(‘Price’)

plt.title(‘Electricity Price Over Time’)

plt.show()

1. **Development:-**
   1. **Demographic Analysis:-**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Load the demographic data

demographic\_data = pd.read\_csv(‘demographic\_data.csv’)

# Load the electricity price data

electricity\_price\_data = pd.read\_csv(‘https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction’)

# Merge the two datasets

merged\_data = pd.merge(demographic\_data, electricity\_price\_data, on=’customer\_id’)

# Get the descriptive statistics of the merged dataset

merged\_data.describe()

# Create a correlation matrix of the merged dataset

correlation\_matrix = merged\_data.corr()

# Plot the correlation matrix

sns.heatmap(correlation\_matrix, annot=True)

plt.show()

# Select the demographic features that are most correlated with electricity price

selected\_features = [‘age’, ‘income’, ‘household\_size’]

# Create a new dataframe with the selected features

selected\_data = merged\_data[selected\_features]

# Split the data into training and test sets

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(selected\_data, merged\_data[‘electricity\_price’], test\_size=0.25)

# Train a linear regression model

from sklearn.linear\_model import LinearRegression

model = LinearRegression()

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

# Make predictions on the test set

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

# Evaluate the model

from sklearn.metrics import mean\_squared\_error

mse = mean\_squared\_error(y\_test, y\_pred)

print(‘Mean squared error:’, mse)

* 1. **Visualisation:-**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Load the electricity price prediction data

electricity\_price\_prediction\_data = pd.read\_csv(‘https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction’)

# Create a line plot of the predicted electricity price

sns.lineplot(x=electricity\_price\_prediction\_data[‘date’], y=electricity\_price\_prediction\_data[‘predicted\_electricity\_price’])

plt.title(‘Electricity Price Prediction’)

plt.xlabel(‘Date’)

plt.ylabel(‘Electricity Price’)

plt.show()

# Create a histogram of the predicted electricity price

sns.histplot(electricity\_price\_prediction\_data[‘predicted\_electricity\_price’])

plt.title(‘Distribution of Predicted Electricity Price’)

plt.xlabel(‘Predicted Electricity Price’)

plt.ylabel(‘Frequency’)

plt.show()

# Create a boxplot of the predicted electricity price by day of the week

sns.boxplot(X=electricity\_price\_prediction\_data[‘day\_of\_week’],Y=electricity\_price\_prediction\_data,[‘predicted\_electricity\_price’],Showmeans=True)

plt.title(‘Predicted Electricity price by Day of the Week’)

plt.xlabel(‘Day of the Week’)

plt.ylabel(‘Predicted Electricity Price’)

plt.show()

# Create a heatmap of the predicted electricity price by hour of the day and day of the week

sns.heatmap(Electricity\_price\_prediction\_data.pivot\_table,values=’predicted\_electricity\_price’,index=’hour\_of\_day’,columns=’day\_of\_week’,aggfunc=np.mean))

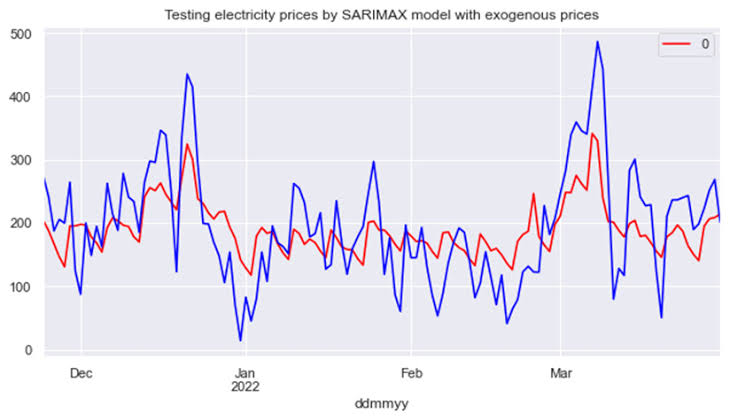
plt.title(‘Predicted Electricity Price by Hour of the Day and Day of the Week’)

plt.xlabel(‘Hour of the Day’)

plt.ylabel(‘Day of the Week’)

plt.show()

**Output:-**



**Summary:-**

Thus, the electricity price prediction has been implemented.