

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import
Linear Regression
from sklearn.model_selection import
train_test_split
# Load the dataset
df =
pd.read_csv('https://www.kaggle.com/data sets/chakradharmattapalli/electricity-price
-prediction/download')
# Preprocess the data
# Convert the date column to datetime
format
df['Date'] = pd.to_datetime(df['Date'])
# Create a new column for the time of day df['Time'] = df['Date'].dt.hour
# Create a new column for the day of the week
df['Day of Week'] = df['Date'].dt.dayofweek
# Create a new column for the month df['Month'] = df['Date'].dt.month
# Drop the Date column
df.drop('Date', axis=1, inplace=True)
# Set the index to the Time column df.set_index('Time', inplace=True)
# Fill in missing values
df.fillna(method='ffill', inplace=True)
# Convert the target variable to a NumPy array
target = df['Electricity Price'].to_numpy()
# Convert the remaining features to a
NumPy array
features = df.drop('Electricity Price', axis=1).to_numpy()
# Split the data into training and testing
sets
X_train, X_test, y_train, y_test =
train_test_split(features, target,
test_size=0.25, random_state=42)
# Train a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Evaluate the model on the test data
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
# Print the evaluation metrics
print('MSE:', mse)
print('RMSE:', rmse)
print('R2:', r2)
# Make predictions on the test data
y_test_pred = model.predict(X_test)

```

```
# Visualize the predictions
plt.scatter(X_test[:, 1], y_test, color='blue',
            label='Actual')
plt.scatter(X_test[:, 1], y_test_pred,
            color='red', label='Predicted')
plt.legend()
plt.title('Electricity Price Prediction')
plt.xlabel('Time of Day (Hour)')
plt.ylabel('Electricity Price') plt.show\(\)
```