This code imports necessary libraries and reads in the data from the CSV file **prices.csv**.

These lines split the data into train and test sets by taking the first 80% of rows as training data and the remaining 20% as test data.

# Split the data into train and test sets

train\_size = int(0.8 \* n)

test\_size = n - train\_size

train\_data = data[:train\_size]

test\_data = data[train\_size:]

This code converts the 'Date' column in the **train\_data** dataframe to a datetime object and sets it as the index for the dataframe.

# Convert the date column to a datetime object and set it as the index

train\_data['Date'] = pd.to\_datetime(train\_data['Date'])

train\_data.set\_index('Date', inplace=True)

This code creates a new column in **train\_data** called 'Day of Week', which contains the day of the week (as a number, where Monday = 0 and Sunday = 6) corresponding to each date in the index.

# Create a new column for the day of the week

train\_data['Day of Week'] = train\_data.index.dayofweek

This code creates a new column in **train\_data** called 'Price Lag 1', which contains the previous day's closing price for each row. The **shift(1)** method shifts the 'Price' column down by 1 row, effectively shifting each day's closing price to the following day. The first row, which has no previous day's closing price, is dropped using **dropna()**

# Create a lagged version of the price column

train\_data['Price Lag 1'] = train\_data['Price'].shift(1)

train\_data.dropna(inplace=True)

These lines create two new variables, **X\_train** and **y\_train**, which contain the feature matrix (X\_train) and target vector (y\_train) for training the linear regression model. The feature matrix contains the 'Day of Week' and 'Price Lag 1' columns from **train\_data**, while the target vector contains the 'Price' column from **train\_data**.

# Split the data into features (X) and target

X\_train = train\_data[['Day of Week', 'Price Lag 1']]

y\_train = train\_data['Price']

These lines create two new variables, **X\_train** and **y\_train**, which contain the feature matrix (X\_train) and target vector (y\_train) for training the linear regression model. The feature matrix contains the 'Day of Week' and 'Price Lag 1' columns from **train\_data**, while the target vector contains the 'Price' column from **train\_data**.

These lines create a new instance of the linear regression model and train it on the training data using **fit()**.

# Create an instance of the linear regression model

model = LinearRegression()

# Train the model on the training data

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

This code converts the 'Date' column in the **test\_data** dataframe to a datetime object and sets it as the index for the dataframe.

test\_data['Date'] = pd.to\_datetime(test\_data['Date'])

test\_data.set\_index('Date', inplace=True)

* **test\_data['Date'] = pd.to\_datetime(test\_data['Date'])**: Convert the "Date" column in **test\_data** from a string to a datetime object using the **pd.to\_datetime()** function and store the result in a new column called "Date".
* **test\_data.set\_index('Date', inplace=True)**: Set the "Date" column as the index of the **test\_data** DataFrame.
* **test\_data['Day of Week'] = test\_data.index.dayofweek**: Create a new column called "Day of Week" in **test\_data** which contains the day of the week for each date in the index.
* **test\_data['Price Lag 1'] = test\_data['Price'].shift(1)**: Create a new column called "Price Lag 1" in **test\_data** which contains the previous day's closing price for each day in the "Price" column. This is done using the **shift()** method which shifts the data in the "Price" column up by one row.
* **test\_data.dropna(inplace=True)**: Drop any rows in **test\_data** which contain missing values.
* **X\_test = test\_data[['Day of Week', 'Price Lag 1']]**: Create a new DataFrame called **X\_test** which contains the "Day of Week" and "Price Lag 1" columns from **test\_data**.
* **y\_test = test\_data['Price']**: Create a new Series called **y\_test** which contains the values from the "Price" column in **test\_data**.
* **y\_pred = model.predict(X\_test)**: Make predictions on the test data using the linear regression model trained on the training data. The predicted values are stored in the **y\_pred** variable.
* **mse = mean\_squared\_error(y\_test, y\_pred)**: Calculate the mean squared error (MSE) between the actual values (**y\_test**) and the predicted values (**y\_pred**) using the **mean\_squared\_error()** function from scikit-learn. The result is stored in the **mse** variable.
* **print("Mean Squared Error:", mse)**: Print the value of the MSE.
* **print(y\_pred)**: Print the predicted values.