In [1]:
```
import pandas as pd
```

In [2]:
```
# Load the Dataset
df=pd.read_csv("stock-market data.csv")
```

In [3]:
```
df
```

Out[3]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	21-07-2010	25.100000	25.100000	24.700001	24.700001	23.343714	42000
1	22-07-2010	25.420000	25.420000	25.129999	25.260000	23.872967	17500
2	23-07-2010	25.540001	25.540001	25.080000	25.280001	23.891865	8600
3	26-07-2010	25.400000	25.400000	25.219999	25.370001	23.976921	18900
4	27-07-2010	25.250000	25.290001	25.200001	25.290001	23.901318	8200
...	...	...	...	...	...	...	...
2437	26-03-2020	41.200001	42.290001	41.200001	42.090000	42.090000	19600
2438	27-03-2020	40.360001	40.500000	39.900002	40.419998	40.419998	6900
2439	30-03-2020	40.549999	40.709999	39.970001	40.709999	40.709999	3000
2440	31-03-2020	40.500000	41.230000	40.250000	40.340000	40.340000	4500
2441	01-04-2020	39.169998	39.169998	38.599998	39.029999	39.029999	5700

2442 rows × 7 columns

In [4]:
```
X = df[['Open', 'Volume']]
y = df['Close']
```

In [5]:
```
# Adding lagged values of the target variable (Close price)
for i in range(1, 4):
    df[f'Close_lagged_{i}'] = df['Close'].shift(i)
```

In [6]:
```
# Drop rows with NaN resulting from lagging
df.dropna(inplace=True)
X = df[['Open', 'Volume', 'Close_lagged_1', 'Close_lagged_2', 'Close_lagged_3'
y = df['Close']
```

In [7]:
```
# Splitting the data into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rando
```

In [8]:
```
# model training
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[8]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [9]:
```
# Model evaluation
from sklearn.metrics import mean_squared_error
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 0.10141538500108284
```

In [10]:
```
# Prediction for the next 3 days
# Get the last available values to predict the next 3 days
last_available_values = X.tail(1)
last_available_values
```

Out[10]:

	Open	Volume	Close_lagged_1	Close_lagged_2	Close_lagged_3
2441	39.169998	5700	40.34	40.709999	40.419998

In [11]:
```
#next_3_days_pred = model.predict(last_available_values)
#print("Predicted Close Prices for the Next 3 Days:", next_3_days_pred)
```

In [12]:
```
next_3_days_pred = []
for i in range(3):
    # Predict for the next day
    next_day_pred = model.predict(last_available_values)
    next_3_days_pred.append(next_day_pred[0])  # Append the predicted value to

    # Update the features for the next prediction
    last_available_values['Open'] = [next_day_pred[0]]  # Update Open with pre
    last_available_values['Volume'] = [20000]  # You can update Volume based o
    for j in range(3, 1, -1):
        last_available_values[f'Close_lagged_{j}'] = last_available_values[f'C
    last_available_values['Close_lagged_1'] = [next_day_pred[0]]  # Update Clo

print("Predicted Close Prices for the Next 3 Days:", next_3_days_pred)
```

```
Predicted Close Prices for the Next 3 Days: [39.32931469452693, 39.2638141165
06104, 39.204469315399685]
```

```
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:8: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  last_available_values['Open'] = [next_day_pred[0]]  # Update Open with pred
icted Close
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:9: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  last_available_values['Volume'] = [20000]  # You can update Volume based on
your data
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:11: S
ettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  last_available_values[f'Close_lagged_{j}'] = last_available_values[f'Close_
lagged_{j-1}']
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:12: S
ettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  last_available_values['Close_lagged_1'] = [next_day_pred[0]]  # Update Clos
e_lagged_1 with predicted Close
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:8: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  last_available_values['Open'] = [next_day_pred[0]]  # Update Open with pred
icted Close
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:9: Se
ttingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  last_available_values['Volume'] = [20000]  # You can update Volume based on your data
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  last_available_values[f'Close_lagged_{j}'] = last_available_values[f'Close_lagged_{j-1}']
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  last_available_values['Close_lagged_1'] = [next_day_pred[0]]  # Update Close_lagged_1 with predicted Close
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  last_available_values['Open'] = [next_day_pred[0]]  # Update Open with predicted Close
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  last_available_values['Volume'] = [20000]  # You can update Volume based on your data
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  last_available_values[f'Close_lagged_{j}'] = last_available_values[f'Close_lagged_{j-1}']
C:\Users\himanshi bajaj\AppData\Local\Temp\ipykernel_2912\2454145894.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  last_available_values['Close_lagged_1'] = [next_day_pred[0]]  # Update Close_lagged_1 with predicted Close
```

In [16]:
```
# Predicted close prices for the next 3 days
next_3_days_pred = [39.32931469452693, 39.263814116506104, 39.204469315399685]
```

In [24]:
```
df.tail(1)
```

Out[24]:

	Date	Open	High	Low	Close	Adj Close	Volume	Close_lagged_1	Cl
2441	01-04-2020	39.169998	39.169998	38.599998	39.029999	39.029999	5700	40.34	

In [17]:
```
# Dates for the next 3 days
dates = ["02-02-2020", "03-02-2020", "04-02-2020"]
```

In [20]:
```
# Create a DataFrame with dates and predicted close prices
predictions_df = pd.DataFrame({"Date": dates, "Predicted Stock Prices": next_3
```

In [21]:
```
predictions_df
```

Out[21]:

	Date	Predicted Stock Prices
0	02-02-2020	39.329315
1	03-02-2020	39.263814
2	04-02-2020	39.204469

In [22]:
```
# Export the DataFrame to an Excel file to import it to tableau public
predictions_df.to_excel("predicted_stock_prices.xlsx", index=False)
```