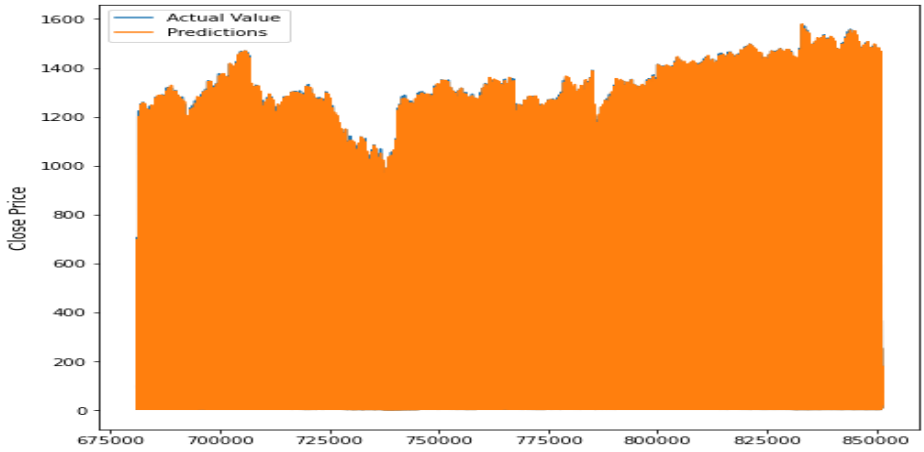
Blog Post

- **Motivation**



  -
  - The problem I am trying to solve now is when we try to show the graph of “Actual Value and Predictions lines”, what we get is not the line but the whole orange image that takes over almost the whole graph.
  - It's an interesting problem because we think the image is so important to express my idea and from the above graph, we can barely see the Actual Value line.
  - Since we can’t see the “Actual value line” clearly, we have no idea how close the Actual value line and Prediction line are.
- **Data**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 851264 entries, 0 to 851263
Data columns (total 7 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   date    851264 non-null  object
 1   symbol  851264 non-null  object
 2   open    851264 non-null  float64
 3   close   851264 non-null  float64
 4   low     851264 non-null  float64
 5   high    851264 non-null  float64
 6   volume  851264 non-null  float64
dtypes: float64(5), object(2)
memory usage: 45.5+ MB

 number of stocks:  501
```

- From the above image, we can see that our data has a total range index of 851264 rows and 7 columns (followed by date, symbol, open, close, low, high and volume). Among this huge data, we now do not have any missing data. From the image, we also can see that there are 501 unique symbols which allow us to predict the result much more accurately.
  - Data Dictionary:
    - Date: a particular period of time in a year.
    - Symbol: the shorthand way of describing a company's stock.
    - Open: the stock price when the market opens.
    - High: the highest price of stock in a day.
    - Low: the lowest price of stock in a day.
    - Close: the stock price at market close.
    - Volume: the number of shares traded.

```
[ ]  # check for null and duplicate
     print("null:", df.isnull().sum().sum())
     print("dupe:", df.duplicated().sum())

     # look at the size of data
     print("Shape:", df.shape)
```

```
null: 0
dupe: 0
Shape: (851264, 7)
```

In our code, we check the data and see if there is any null data. However, from the above image (from code), we can see that there is no missing value in all 851264 now. Finally, we can start our model without worrying about the accuracy of our result.

- **Evaluation**
  - The Prediction Model is by splitting data into the test train pairs.

```
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, shuffle=Fa

Acc = []
```

  - Then we use Linear Regression to show the Actual and Predicted values. And we can see from the image below and compare the each row value of Actual and Predicted, both values are very close. That means our model is in the right direction.

```
[31] # prediction
     y_pred_1 = model_1.predict(X_test)
     pred_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred_1})
     pred_df.head()
```

	Actual	Predicted
681011	97.750000	98.585524
681012	83.040001	84.471816
681013	39.750000	40.042867
681014	88.680000	89.898516
681015	42.630001	43.057401

- Then we are going to measure the Accuracy score by import r2_score from sklearn.metrics. Where we get the Accuracy score of the predictions: 0.9999… which means the results are very close.

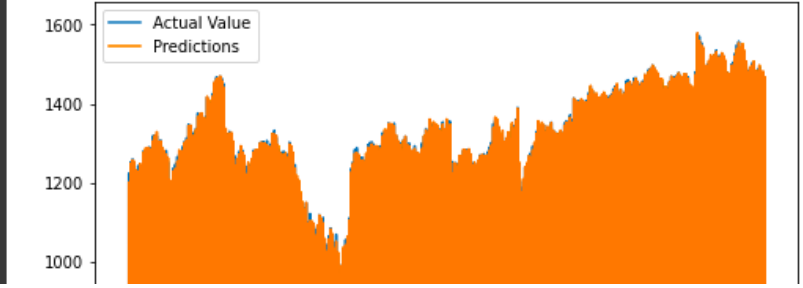
```
from sklearn.metrics import r2_score

print("Accuracy score of the predictions: {0}".format(r2_score(y_test, y_pred_1)))
Acc.append(r2_score(y_test, y_pred_1))
```

```
Accuracy score of the predictions: 0.9999431626682372
```

- Then we plot the graph that contains Actual Value and Predictions.

```
plt.figure(figsize=(8, 8))
plt.ylabel('Close Price', fontsize=12)
plt.plot(pred_df)
plt.legend(['Actual Value', 'Predictions'])
plt.show()
```



From the above graph, we can't tell how close are the Actual value and Predictions lines. But from the Accuracy score of prediction, we can deduce that the Actual Value and Predictions lines will not have much difference when they rolling out

- **Future Work**
    - Trying to fix the graph problem we mentioned at the beginning.
        - The image can clearly express what we want to say, so a clear graph is a must. And that graph isn't enough to explain the relationship between Actual Value and Predictions lines.
    - We are thinking about adding another model if we think it's necessary later.
        - We are not sure if it's necessary right now because we are focusing on the graph problem. And If we add another model, do we need to compare both models? Maybe we should.