FINAL PRESENTATION

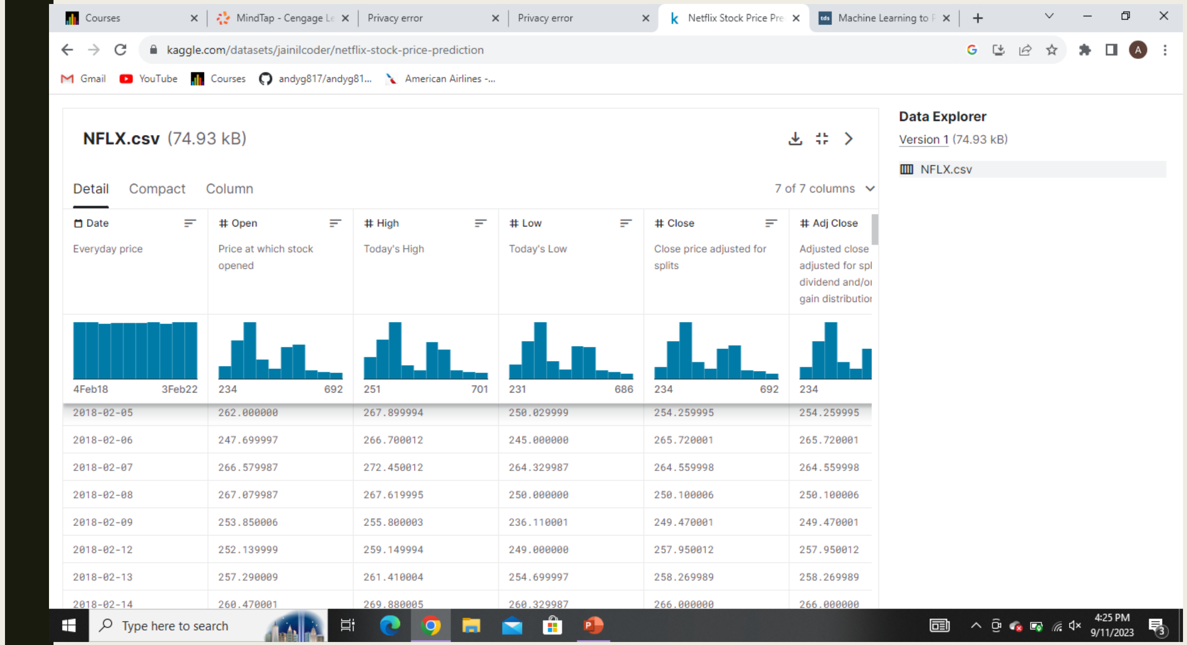
Andrew Garcia

Overview

- My project is to show the key differences between two different algorithms widely used today in Stock Market Prediction, Long Short Term Memory(LSTM) and Linear Regression
- I will show the differences between these two algorithms and highlight what their respective strengths and weaknesses are using selected stocks for the dataset

Implementation/ dataset

- I will be implementing and comparing these two algorithms on the same sets of stocks/ data sets such as Google or Netflix stock history



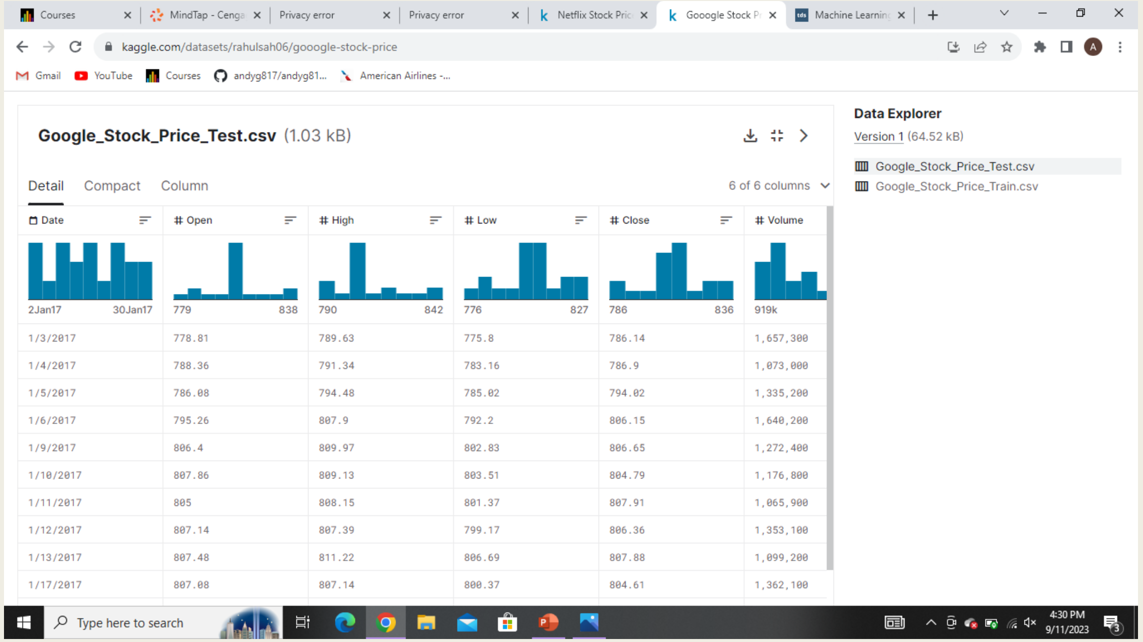
kaggle.com/datasets/jainilcoder/netflix-stock-price-prediction

NFLX.csv (74.93 kB)

Data Explorer
Version 1 (74.93 kB)
NFLX.csv

Detail Compact Column — 7 of 7 columns

Date	Open	High	Low	Close	Adj Close
Everyday price	Price at which stock opened	Today's High	Today's Low	Close price adjusted for splits	Adjusted close adjusted for spl dividend and/o gain distributio
4Feb18 – 3Feb22	234 – 692	251 – 701	231 – 686	234 – 692	234
2018-02-05	262.000000	267.899994	250.029999	254.259995	254.259995
2018-02-06	247.699997	266.700012	245.000000	265.720001	265.720001
2018-02-07	266.579987	272.450012	264.329987	264.559998	264.559998
2018-02-08	267.079987	267.619995	250.000000	250.100006	250.100006
2018-02-09	253.850006	255.800003	236.110001	249.470001	249.470001
2018-02-12	252.139999	259.149994	249.000000	257.950012	257.950012
2018-02-13	257.290009	261.410004	254.699997	258.269989	258.269989
2018-02-14	260.470001	269.880005	260.329987	266.000000	266.000000



kaggle.com/datasets/rahulsah06/gooogle-stock-price

Google_Stock_Price_Test.csv (1.03 kB)

Data Explorer
Version 1 (64.52 kB)
Google_Stock_Price_Test.csv
Google_Stock_Price_Train.csv

Detail Compact Column — 6 of 6 columns

Date	Open	High	Low	Close	Volume
2Jan17 – 30Jan17	779 – 838	790 – 842	776 – 827	786 – 836	919k
1/3/2017	778.81	789.63	775.8	786.14	1,657,300
1/4/2017	788.36	791.34	783.16	786.9	1,073,000
1/5/2017	786.08	794.48	785.02	794.02	1,335,200
1/6/2017	795.26	807.9	792.2	806.15	1,640,200
1/9/2017	806.4	809.97	802.83	806.65	1,272,400
1/10/2017	807.86	809.13	803.51	804.79	1,176,800
1/11/2017	805	808.15	801.37	807.91	1,065,900
1/12/2017	807.14	807.39	799.17	806.36	1,353,100
1/13/2017	807.48	811.22	806.69	807.88	1,099,200
1/17/2017	807.08	807.14	800.37	804.61	1,362,100

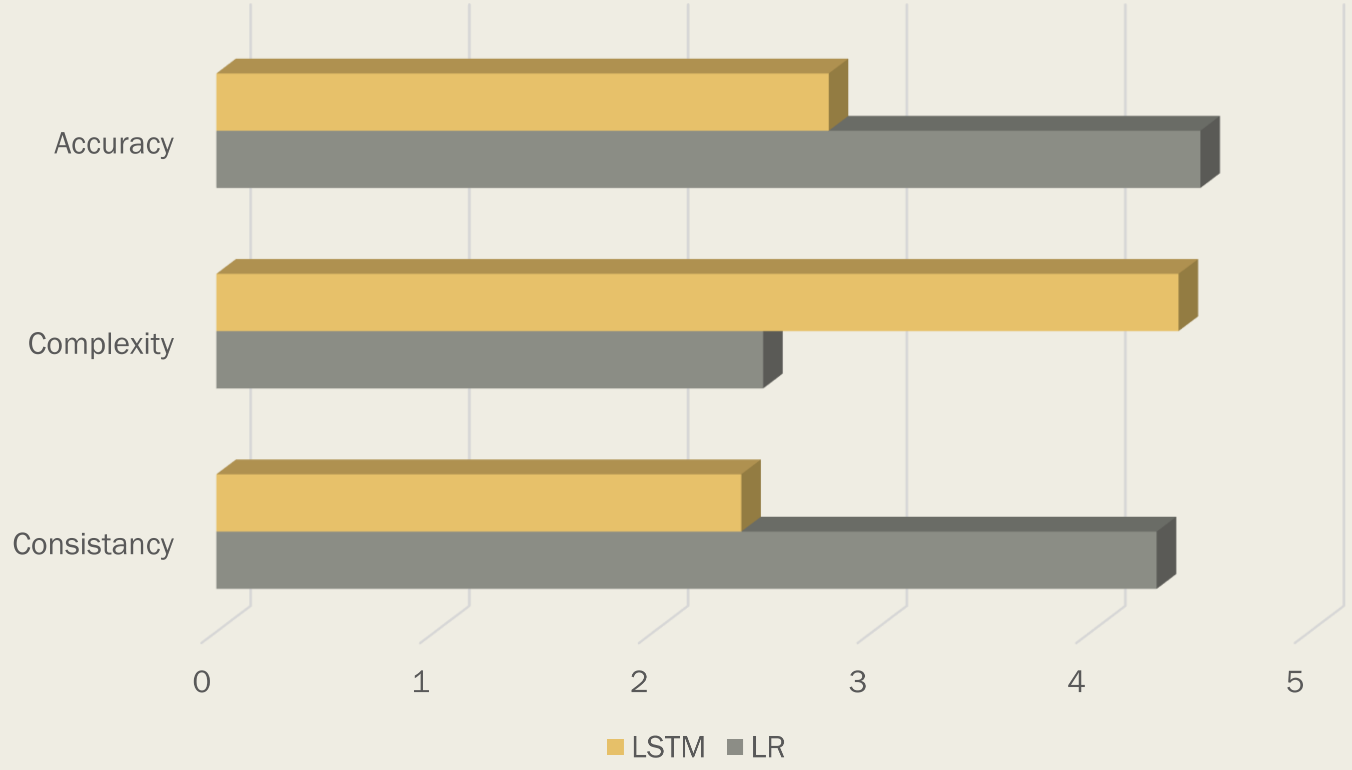
Metrics

- I will be evaluate Logistic regression and Long Short Term Memory in the Metrics of

consistency

complexity

accuracy



Methodology

LSTM

Pros: great for modeling long term dependencies in data such as stocks since it can remember and forget information, along with its ability to select important information

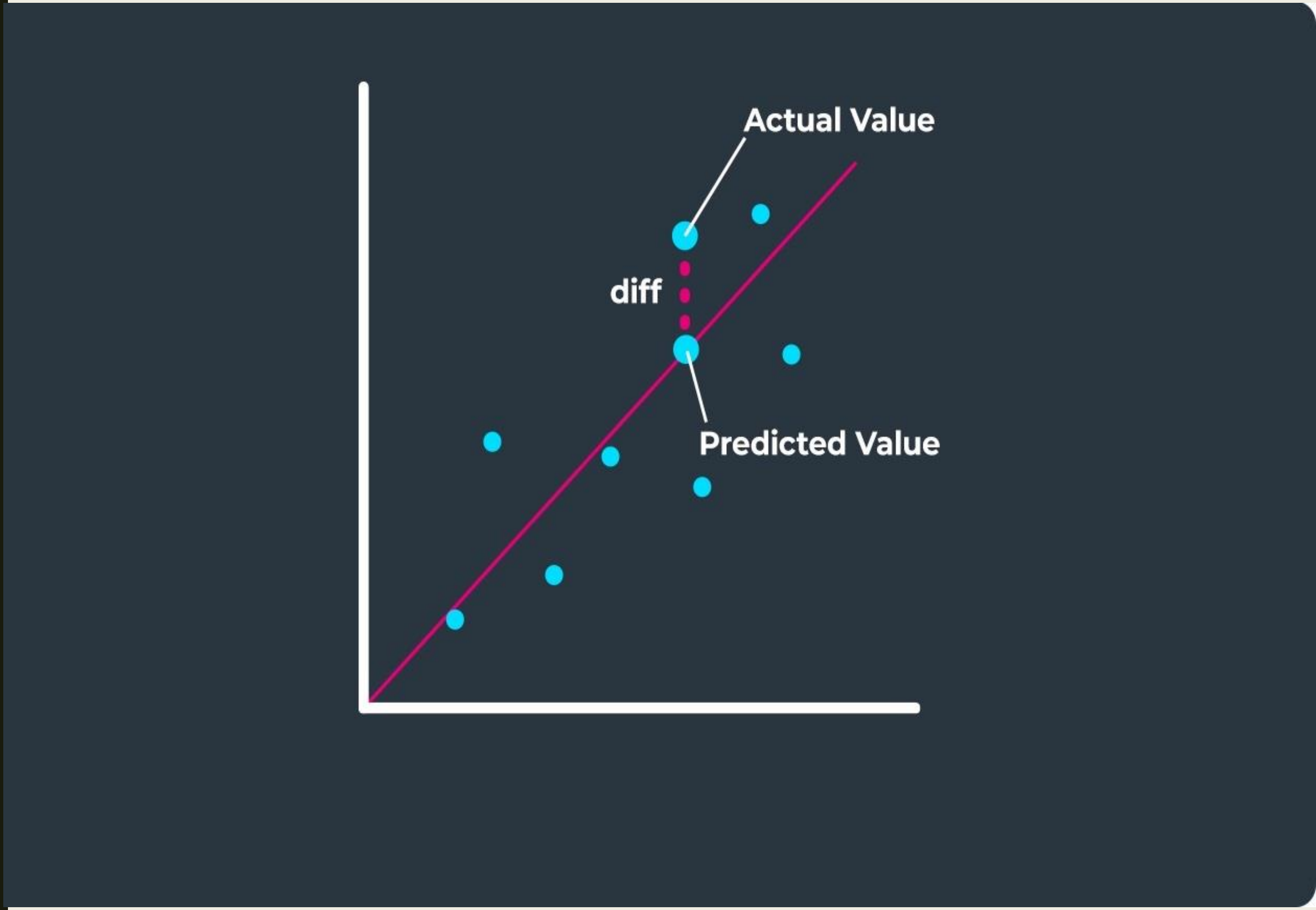
Cons: Complex and requires lots of data to be accurate

LR

Pros: Easy to implement and extend

Cons: struggles on very complex relationships

Both can be measured with Root mean Square Error which measures the average difference between a statistical models predicted values and the actual values



Implementation

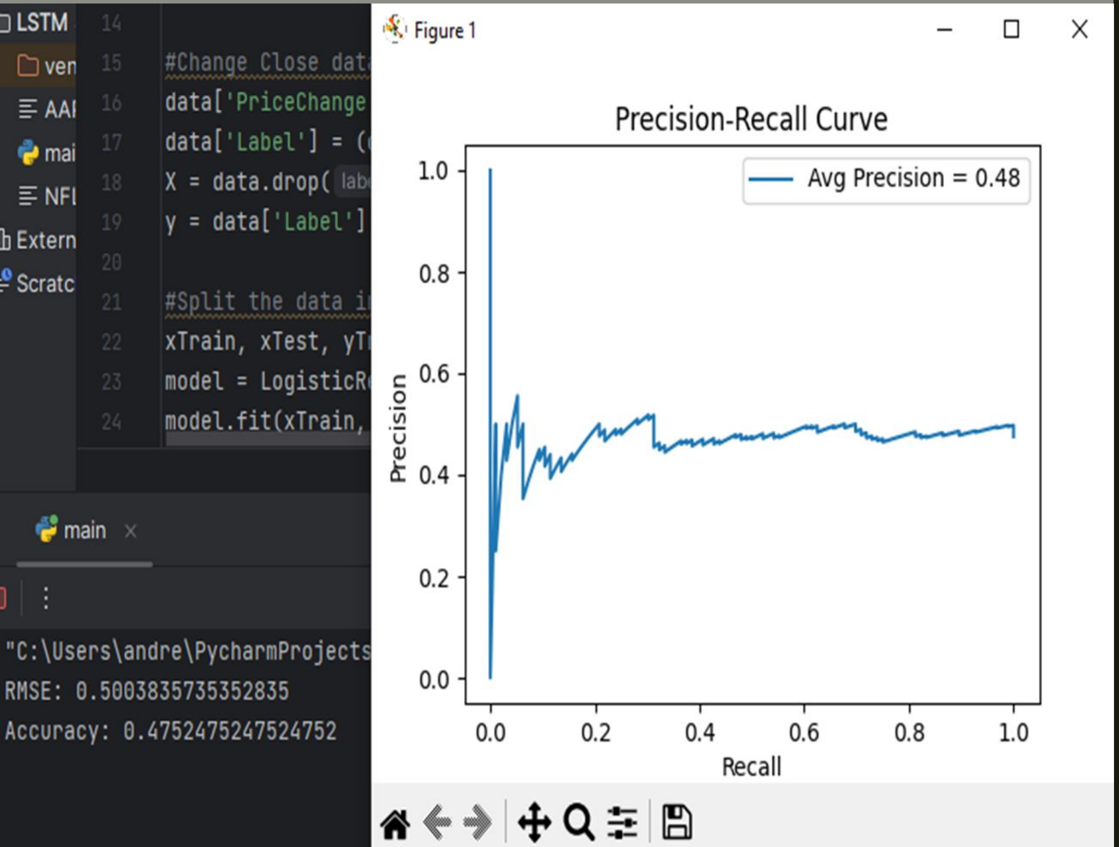
- Python
  - *pycharm*
- Packages
  - *Pandas : read in dataset*
  - *Matplotlib : plots data*
  - *Sklearn : calculations (split data, precision curve, average, mse etc)*
  - *Trnsorflow : imports LSTM*

Experimental Setup

- Dataset
  - *Netflix stocks : date, closing price*
  - *1010 data points*
- Both give graphs displaying model accuracy
- Comparison metrics
  - *Accuracy*
  - *RMSE*
  - *Consistancy*

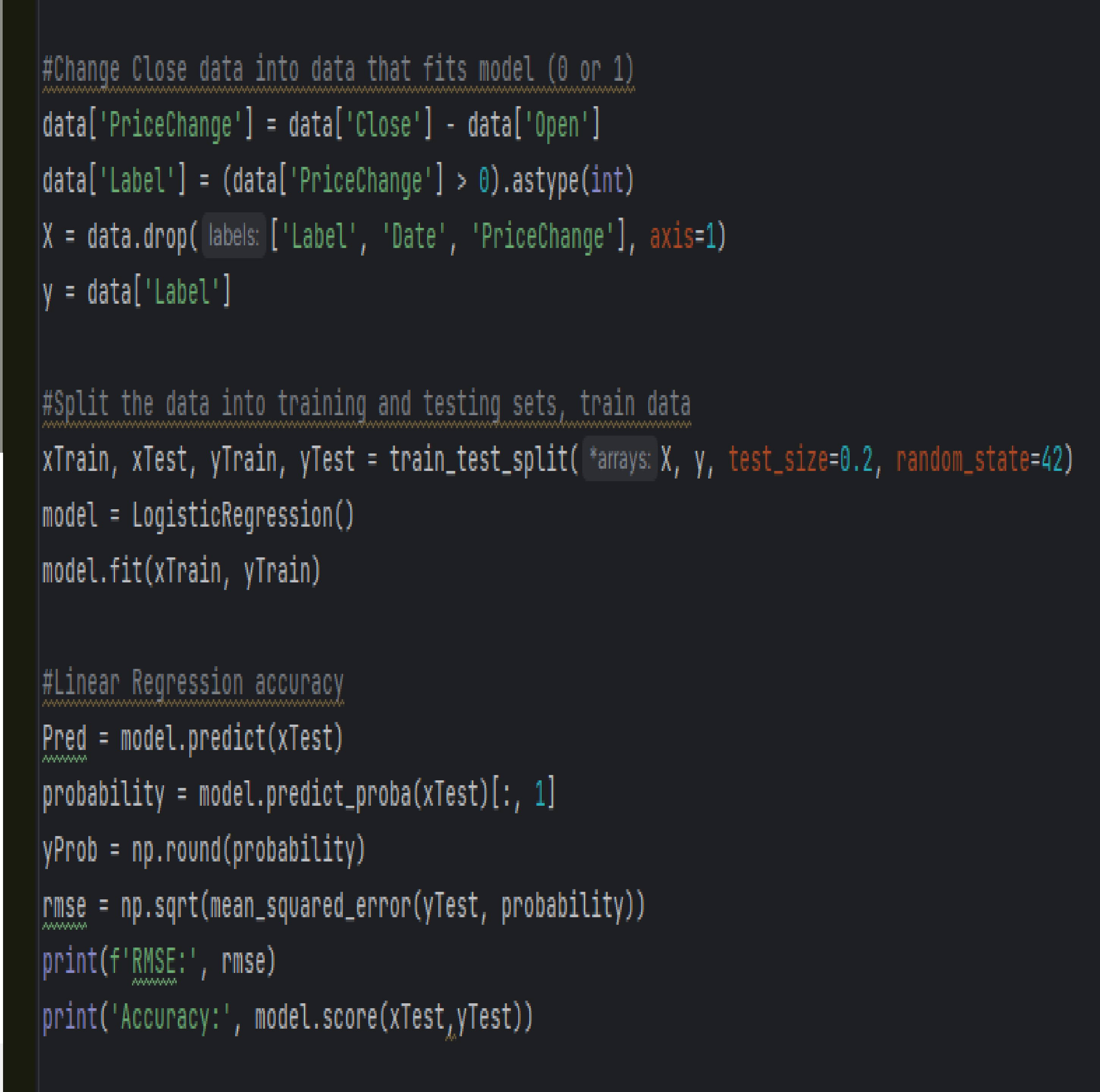
Implementation Code: Initial Results

Logistic Regression



```
#Change Close dat
data['PriceChange
data['Label'] = (
X = data.drop( labe
y = data['Label']

#Split the data i
xTrain, xTest, yT
model = LogisticR
model.fit(xTrain,
```

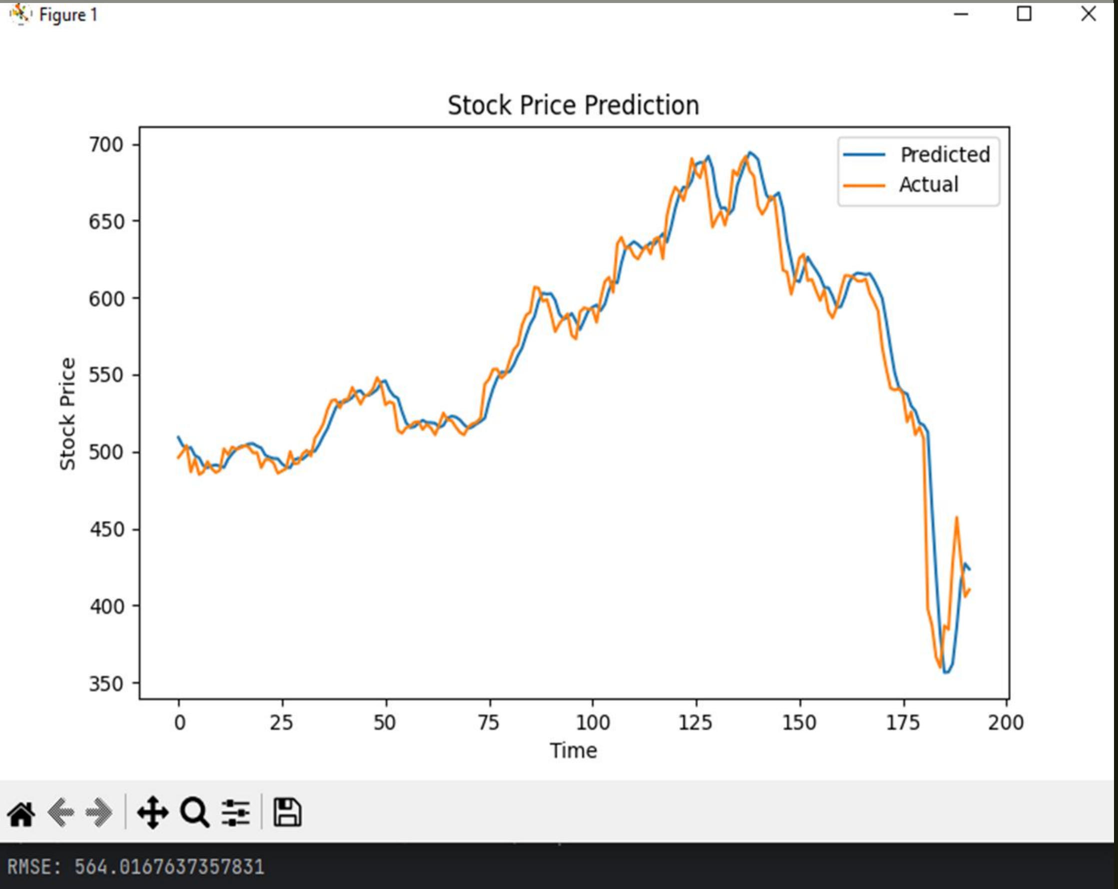
```
"C:\Users\andre\PycharmProjects
RMSE: 0.5003835735352835
Accuracy: 0.4752475247524752
```



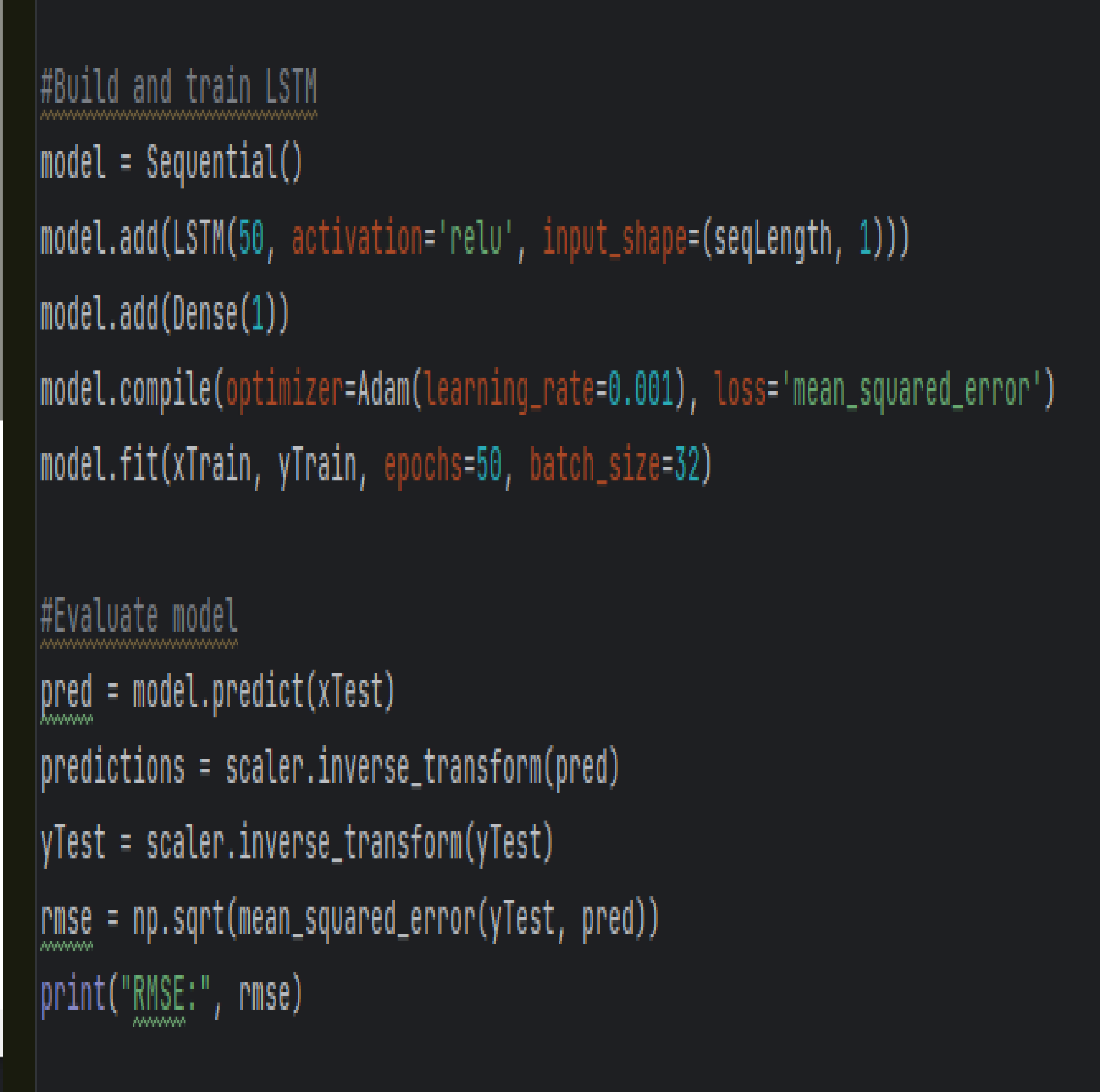
```
#Change Close data into data that fits model (0 or 1)
data['PriceChange'] = data['Close'] - data['Open']
data['Label'] = (data['PriceChange'] > 0).astype(int)
X = data.drop( labels: ['Label', 'Date', 'PriceChange'], axis=1)
y = data['Label']


#Split the data into training and testing sets, train data
xTrain, xTest, yTrain, yTest = train_test_split( *arrays: X, y, test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(xTrain, yTrain)


#Linear Regression accuracy
Pred = model.predict(xTest)
probability = model.predict_proba(xTest)[:, 1]
yProb = np.round(probability)
rmse = np.sqrt(mean_squared_error(yTest, probability))
print(f'RMSE:', rmse)
print('Accuracy:', model.score(xTest,yTest))
```

Implementation Code: Initial Results

Long Short Term Memory



```
RMSE: 564.0167637357831
```



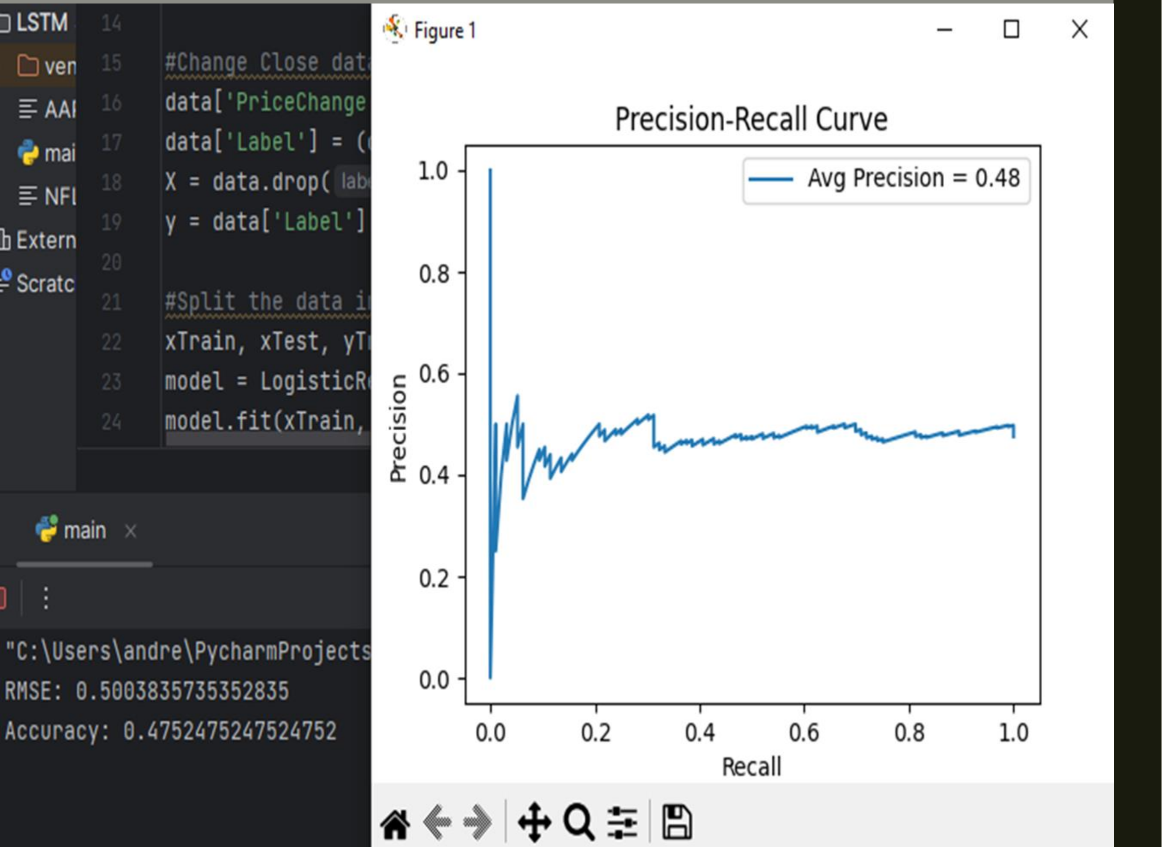
```
#Build and train LSTM
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(seqLength, 1)))
model.add(Dense(1))
model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
model.fit(xTrain, yTrain, epochs=50, batch_size=32)


#Evaluate model
pred = model.predict(xTest)
predictions = scaler.inverse_transform(pred)
yTest = scaler.inverse_transform(yTest)
rmse = np.sqrt(mean_squared_error(yTest, pred))
print("RMSE:", rmse)
```

Testing modifications

- I trained both the LSTM model and LR model with three different stock market datasets which include Google, Netflix, and Microsoft

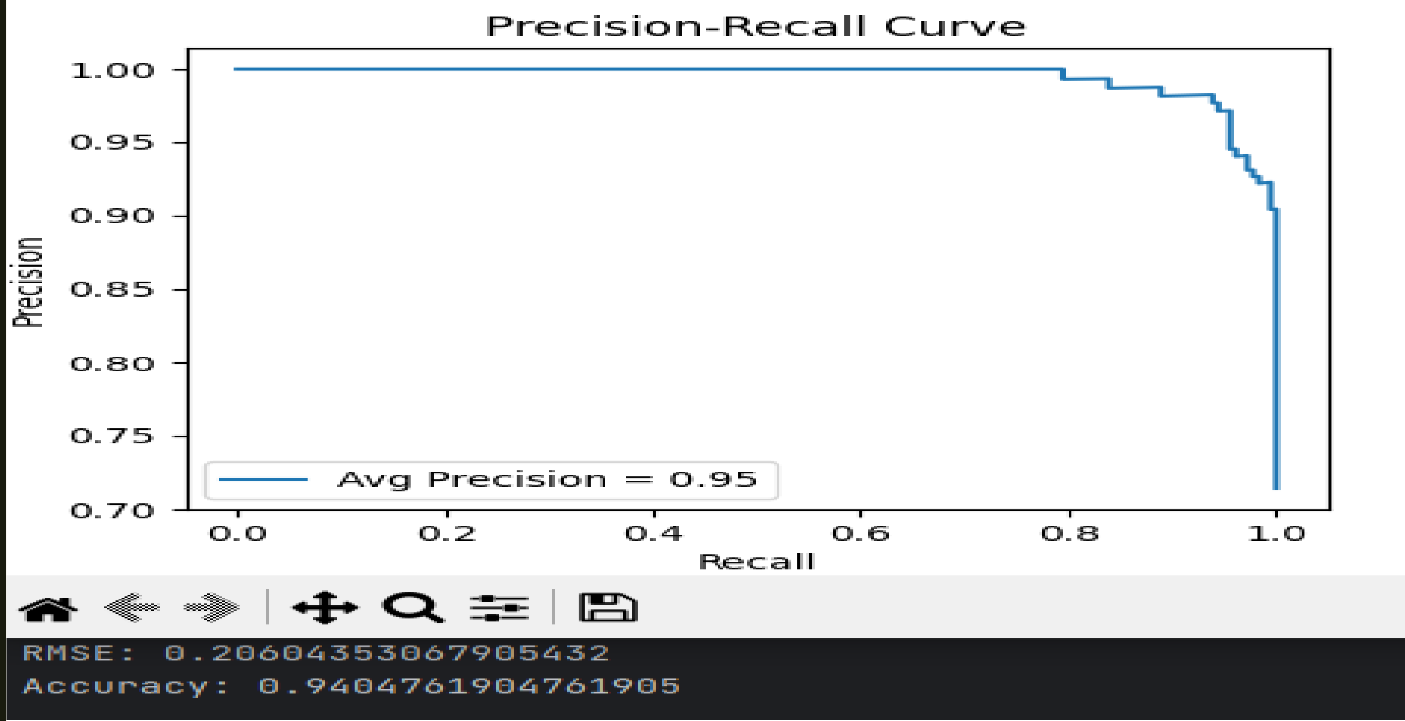
Implementation Code: Final Results

Logistic Regression

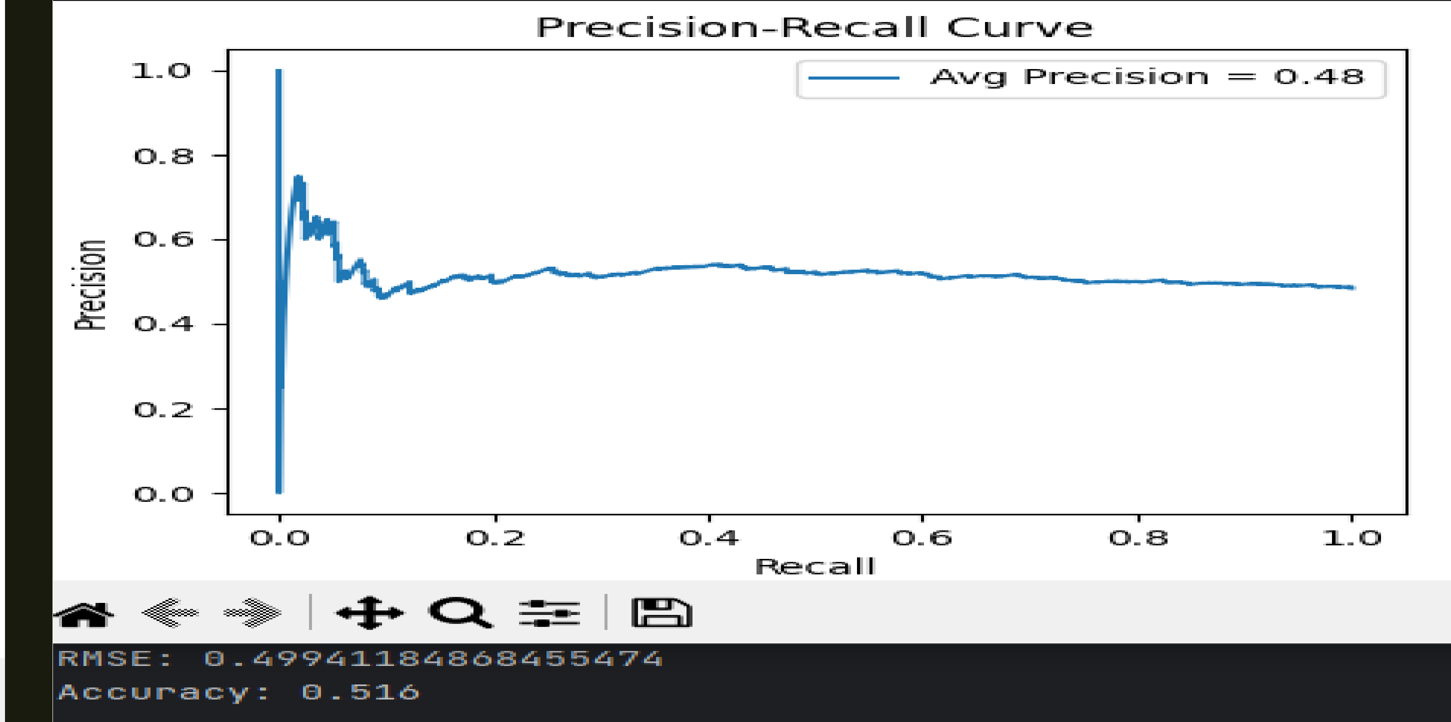


```
#Change Close dat
data['PriceChange
data['Label'] = (
X = data.drop( labe
y = data['Label']

#Split the data i
xTrain, xTest, yT
model = LogisticR
model.fit(xTrain,
```

```
"C:\Users\andre\PycharmProjects
RMSE: 0.5003835735352835
Accuracy: 0.4752475247524752
```



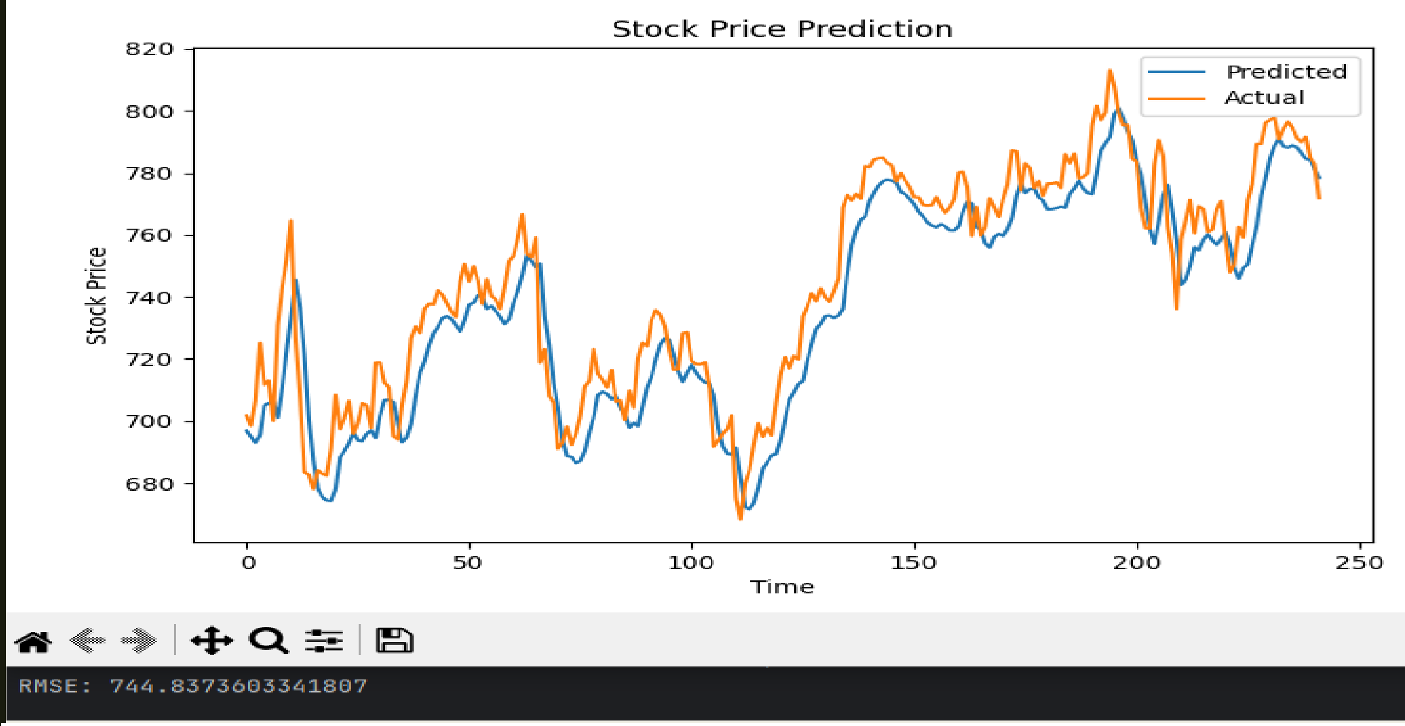
```
RMSE: 0.20604353067905432
Accuracy: 0.9404761904761905
```



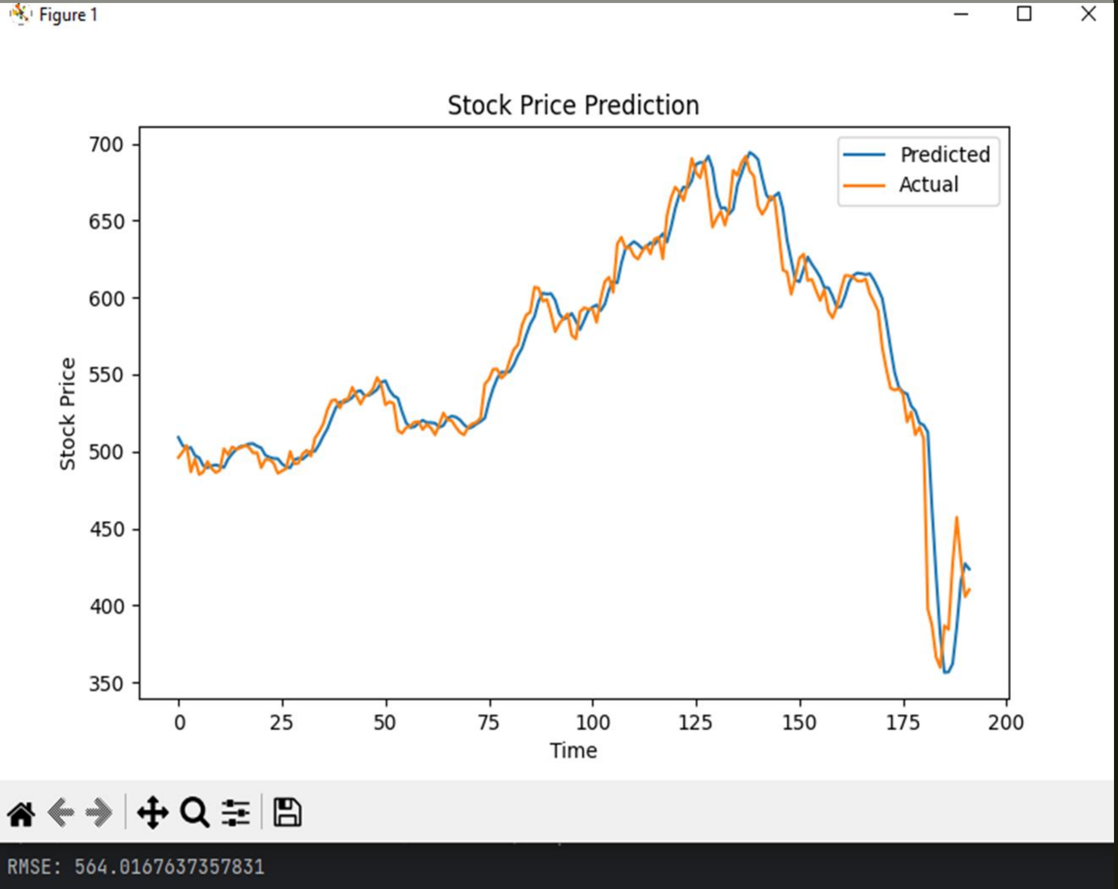
```
RMSE: 0.49941184868455474
Accuracy: 0.516
```

Implementation Code: Final Results

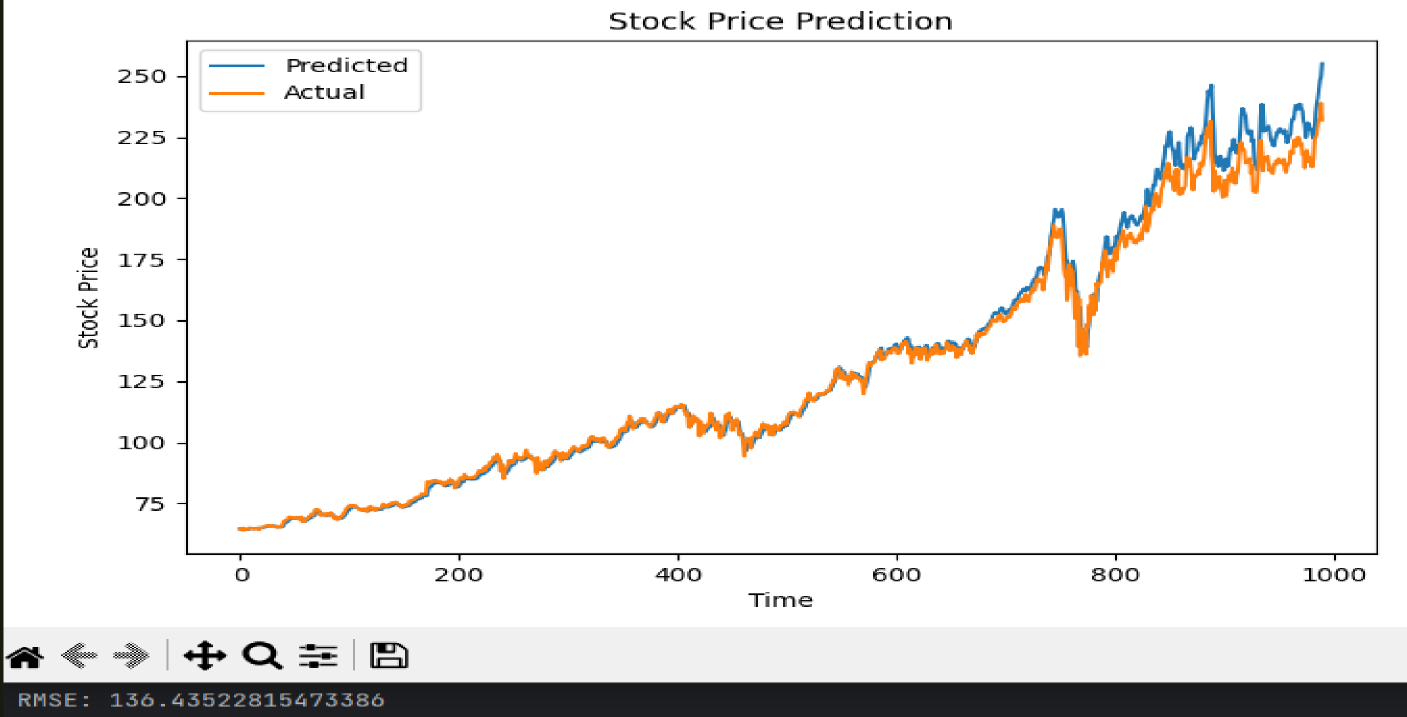
Long Short Term Memory



```
RMSE: 744.8373603341807
```



```
RMSE: 564.0167637357831
```



```
RMSE: 136.43522815473386
```

References

- M Siddharth. “Stock price using LSTM and its implementation” Analytics Vidhya, December 6, 2021, https://www.analyticsvidhya.com/blog/2021/12/stock-price-prediction-using-lstm/
- Bandara Isira, ‘Stock Market Prediction Using Linear Regression Modeling” medium.com, Aug 31, 2022, https://medium.com/@isirabandarafb/stock-market-prediction-using-linear-regression-modeling-5a1c9b510254
- https://medium.com/@anishnama20/understanding-lstm-architecture-pros-and-cons-and-implementation-3e0cca194094
- https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/#