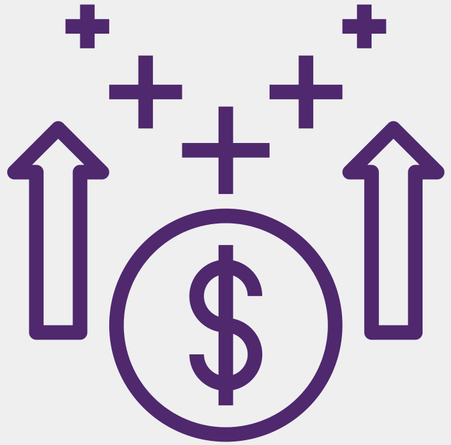
Pract Machine Deep Learning

Stock Price Prediction

Mohamed Ellethy - Mark Adil

Investing in Stocks

- Build Your Savings
- Protect Your Money from Inflation and Taxes
- Maximize income from your investments



Problem Definition

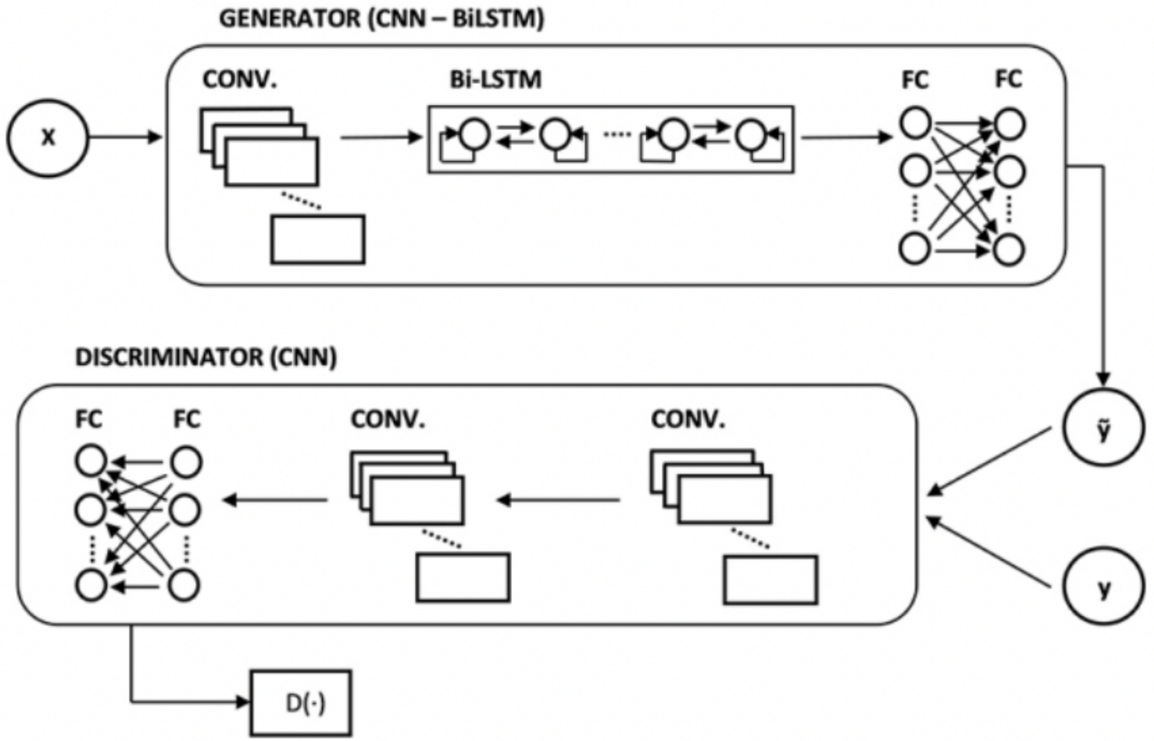
Stock Price Prediction

Politics

Fundamental Analysis

Technical Analysis

State of Art: Forecasting by a Deep Convolutional Generative Adversarial Network:

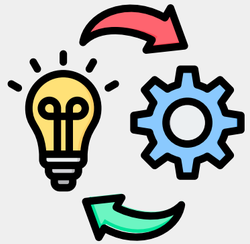


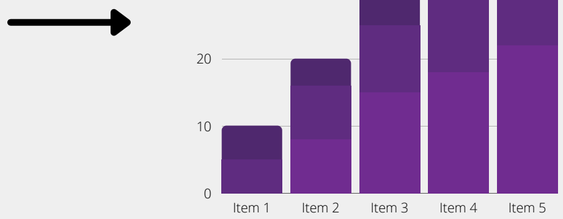
ARIMAX-SVR		RF Regressor	LSTM		Benchmark GAN		Our GAN	
(p,d,q)	Results	Results	Lags	Results	Lags	Results	Lags	Results
(5,1,0)	0.990	0.814 (0.060)	2	0.763 (0.015)	2	0.890 (0.120)	2	**0.631 (0.043)**
	0.820	0.543 (0.022)		0.501 (0.011)		0.694 (0.118)		**0.416 (0.029)**
	4.871	3.034 (0.177)		2.779 (0.052)		3.821 (0.829)		**2.294 (0.134)**
(5,1,0)	0.663	0.633 (0.084)	2	0.582 (0.004)	2	0.895 (0.149)	2	**0.541 (0.011)**
	0.530	0.460 (0.053)		0.403 (0.004)		0.788 (0.150)		**0.347 (0.003)**
	3.544	2.900 (0.329)		2.603 (0.035)		5.306 (1.050)		**2.329 (0.028)**
(4,1,0)	0.899	0.639 (0.038)	2	0.578 (0.006)	2	0.700 (0.106)	2	**0.506 (0.008)**
	0.796	0.479 (0.023)		0.426 (0.006)		0.560 (0.124)		**0.366 (0.006)**
	4.128	2.496 (0.117)		2.225 (0.032)		2.951 (0.629)		**1.911 (0.030)**
(2,1,0)	0.427	0.467 (0.022)	2	0.266 (0.023)	2	0.352 (0.097)	2	**0.180 (0.013)**
	0.313	0.274 (0.021)		0.179 (0.017)		0.179 (0.017)		**0.120 (0.011)**
	4.488	3.582 (0.017)		2.605 (0.213)		4.863 (0.635)		**1.797 (0.150)**
(1,1,0)	1.390	1.874 (0.016)	2	1.231 (0.064)	2	1.093 (0.260)	2	**0.391 (0.073)**
	1.124	1.152 (0.014)		0.838 (0.033)		0.951 (0.258)		**0.280 (0.059)**
	11.194	13.839 (0.144)		9.568 (0.441)		8.372 (2.065)		**2.688 (0.620)**
(1,1,0)	3.533	4.410 (0.014)	2	2.662 (0.186)	2	2.749 (0.636)	2	**1.930 (0.113)**
	2.635	2.828 (0.011)		1.945 (0.136)		2.250 (0.565)		**1.403 (0.132)**
	4.719	4.438 (0.028)		3.315 (0.214)		3.907 (0.975)		**2.454 (0.227)**

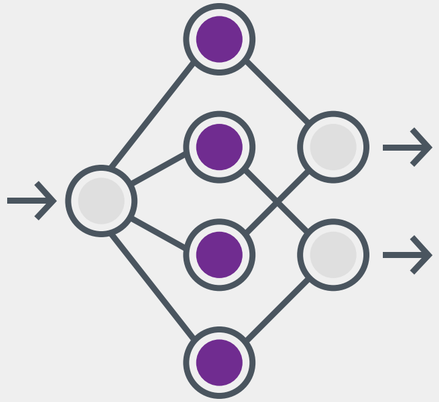
Comparing with four widely used models for forecasting Deep Convolutional Genera
Adversarial Network outperforms others in the evaluation metrics making it the best

Stock Market Analysis + Prediction using LSTM

Deployed Model





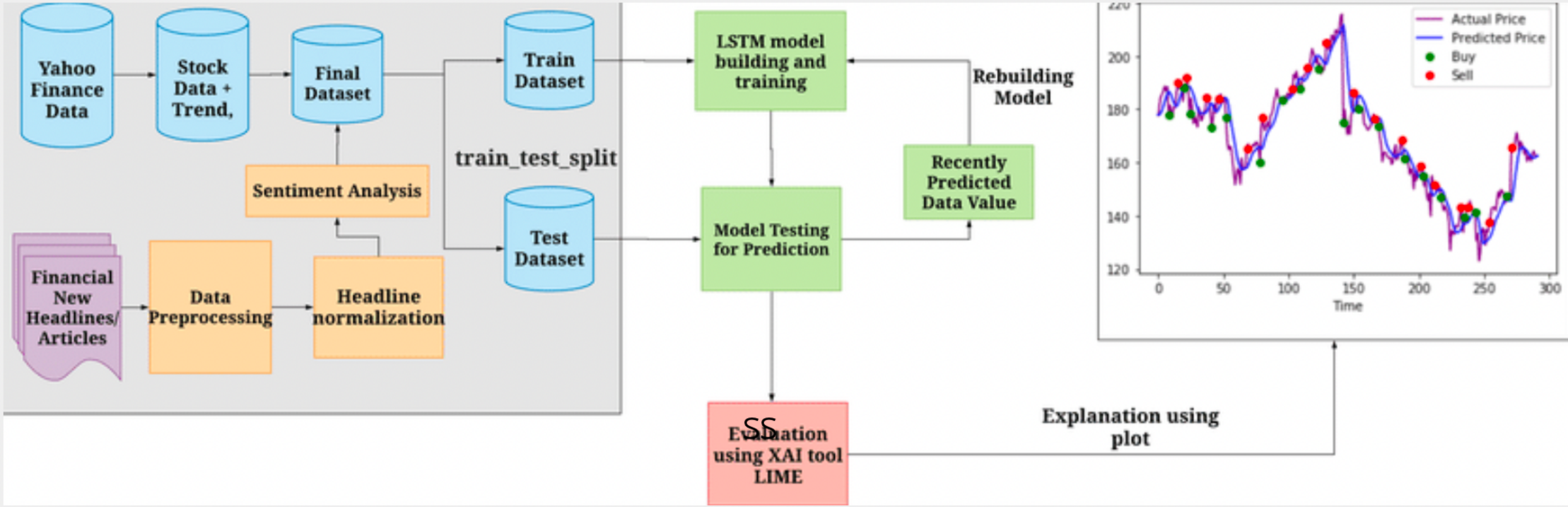


Data from Yahoo Finance

Graphical Representation

LSTM

Predicted Price



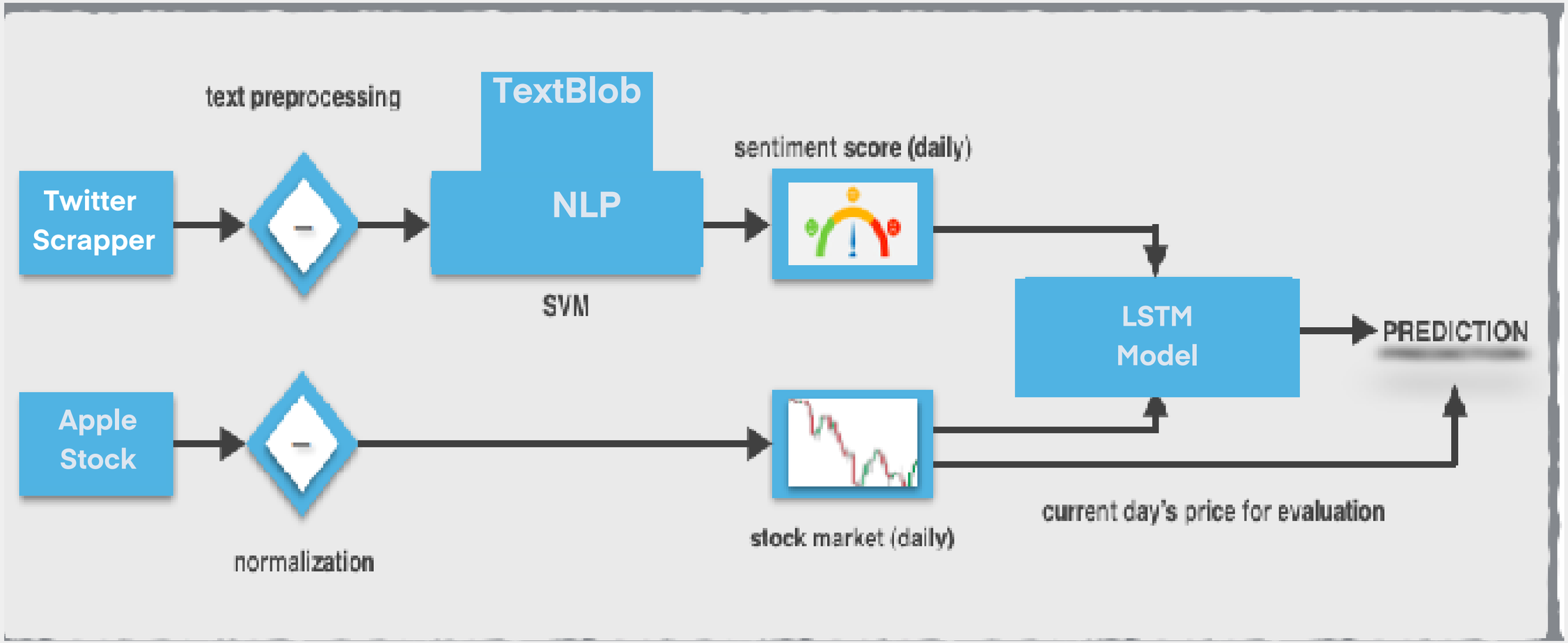
LSTM has three gates:

The input gate: The input gate adds information to the cell state,

The forget gate: It removes the information that is no longer required by the model,

The output gate: Output Gate at LSTM selects the information to be shown as output.

Our solution.



Adding Sentiment Analysis:

27.5K
Retweets

2,793
Quote Tweets

329.4K
Likes

3500% Increase

Progress Done

- Implementing a scrapper to collect Apple Tweets for period between 2010-2020 using Selenium and BeautifulSoup
- Feeding tweets into TextBlob to get the sentiment score for the Tweets
- Merging the stock prices with sentiment score and feeding it as new input into the model
- Hyper-paramater tuning

Scrapper Implementation

```
    url = 'https://twitter.com/search?q=%24AAPL&src=typeahead_click&f=top'

    browser = webdriver.Chrome(executable_path='/Users/leithy/Downloads/chromedriver')

    browser.get(url)
    time.sleep(30)


    elem = browser.find_element_by_tag_name("html")

    no_of_pagedowns = 10000

    f = open("aaplTweets.html", "w")

    while no_of_pagedowns:
        elem.send_keys(Keys.PAGE_DOWN)
        time.sleep(0.5)
        if(no_of_pagedowns%15==0):
            content = browser.page_source
            f.write(content)
        no_of_pagedowns-=1
        print(no_of_pagedowns)

    f.close()

    with open('aaplTweets.html', 'r') as f:

        contents = f.read()

    soup = BeautifulSoup(contents, 'html5lib')
    tweets = soup.find_all('div', attrs={'class':'css-901oao r-18jsvk2 r-37j5jr r-a023e6 r-16dba41 r-rjixqe r-bcqeeo r-bnwqim r-qvutc0'})
    ts = soup.find_all('time')
    print(len(tweets))
    print(len(ts))


    for i in range(len(tweets)):
        print(tweets[i].get_text())
        print(ts[i]['datetime'].partition('T')[0])
        tweet_dict = {
        'tweet' : cleanTweet(tweets[i].get_text().rstrip()),
        'date' : ts[i]['datetime'].partition('T')[0],
        }
        applStock.append(tweet_dict)

    df = pd.DataFrame(applStock)
```

Sentiment Analysis

```
[ ]  polarity=[]
     subjectivity=[]
     for d in df.itertuples():
       print(d.Tweet)
       try:
         strin=TextBlob((d.Tweet))
         polarity.append(strin.sentiment.polarity)
         subjectivity.append(strin.sentiment.subjectivity)
         #print(strin.sentiment)
       except:
         polarity.append(0)
         subjectivity.append(0)

     df['polarity']=polarity
     df['subjectivity']=subjectivity
```

Code Snippet

```
df=pd.read_csv("/content/prices.csv")
tweets_dataframe=pd.read_csv("/content/AAPL.csv")
df=pd.merge(df, tweets_dataframe, on='Date')
#df=df.merge(tweets_dataframe,on='Date')
df
```

New Input

	Close	ts_polarity
0	172.259995	0.095630
1	172.229996	0.112818
2	173.029999	0.131831
3	175.000000	0.105427
4	174.350006	0.143506
...	...	...
515	319.230011	0.048873
516	316.850006	0.005594
517	318.890015	0.061395
518	316.730011	0.052449

Results

Original model RMSE

```
# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
rmse
```

```
8.345904016049182
```

Our Model RMSE

```
# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
print(rmse)
```

```
2.6547111500702107
```

Graphs

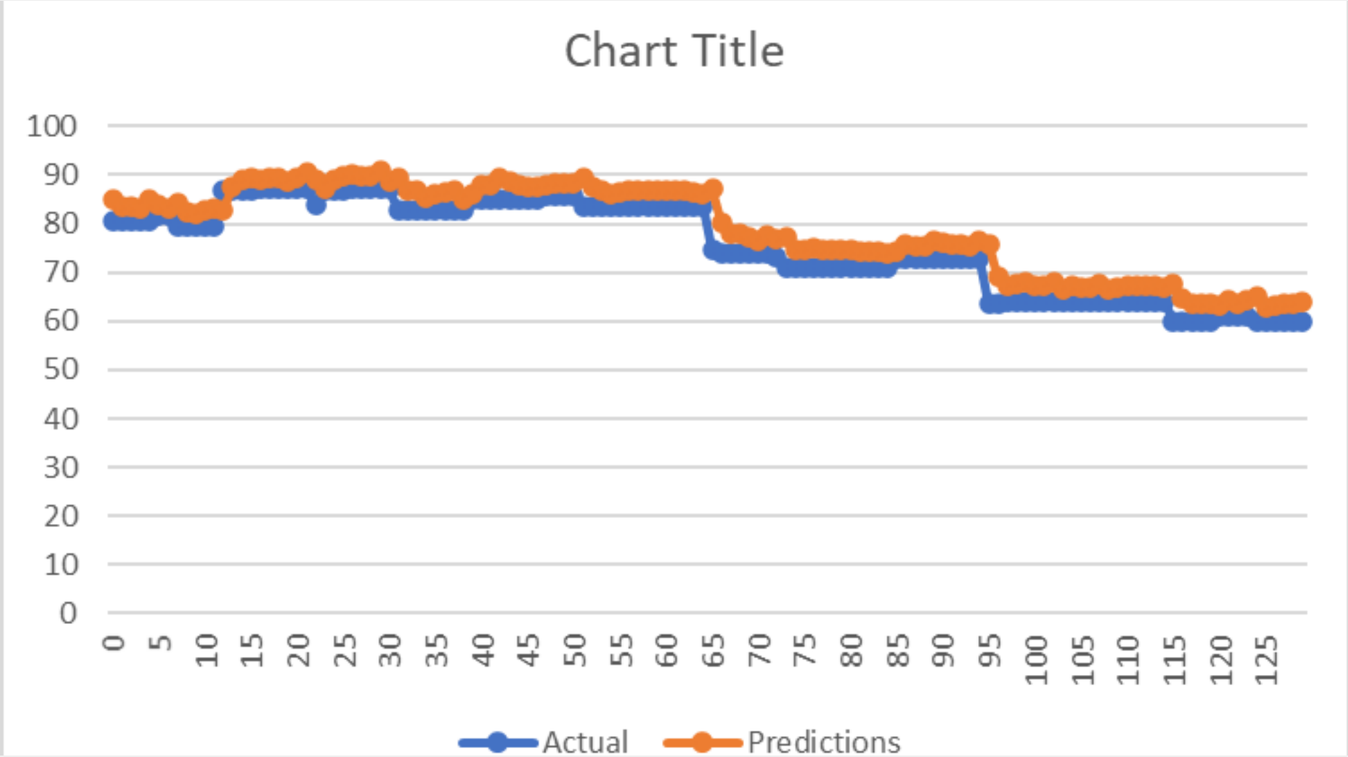


Chart Title
100
90
80
70
60
50
40
30
20
10
0
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
Actual
Predictions

Predictions 15-3-2022 to 12-5-2022

	polarity	Close	Predictions
227	0.200000	146.5	146.571594
228	0.000000	146.5	146.592697
229	0.000000	146.5	146.562317
230	0.500000	146.5	146.549683
231	-0.235185	146.5	146.604294
232	0.400000	146.5	146.481262
233	-0.155556	146.5	146.551407
234	0.000000	146.5	146.486435
235	0.175000	146.5	146.507355
236	0.375000	146.5	146.535583
237	0.000000	146.5	146.549789

Contribution

Mohamed
Fine tuning the hyper-parameters
Creating tweet scrapper using Selenuim and beautifulSoup
Creating tweet sentiment score by feeding tweets into tweet sentiment analysis model

Mark
Fine tuning the hyper-parameters
Integrating the tweet sentiment score to the original model and reshaping the model to accommodate the extra input
Creating tweet sentiment score by feeding tweets into tweet sentiment analysis model

References:

Kaggle.com. 2022. Stock Market Analysis + Prediction using LSTM. [online] Available at: <https://www.kaggle.com/faressayah/stock-market-analysis-prediction-using-lstm/comments>

Thank You