REPORT ON STOCK MARKET ANALYSIS

Aishwarya Gunasekar

agunasek@uncc.edu

Feb 12th 2019

CONTENTS

INTRODUCTION

Stock data of the following companies were collected from various sources for years 2013 - 2019 :

- Apple (AAPL)
- IBM
- Goldman Sachs (GS)
- Amazon (AMZN)
- General Electric (GE)
- Google (GOOG)

A custom web scraper was built to scrape news headlines and numerical data from various sources. The data went through an extensive data preparation procedure to be able to make relevant predictions.

Unusual market behavior was detected using Unsupervised Deep Learning Technique - Self Organized Maps.

Fundamental analysis was performed over the data to understand which stocks may cause loss/profit in the portfolio.

Technical Analysis was performed over the data to understand which indicators impact the market the most.

These indicators were used to build an LSTM network to predict short term prices.

An improvisation to the model was performed after extracting daily news and appending them to the numerical stock data. A significant improvement in the prediction of market trends was observed.

Multi-step algorithms of Recursive Strategy and Direct Strategy was implemented from the research paper given.

DATA COLLECTION AND PREPARATION

The train data consisted of years Jan 2013 - Dec 2018.
The test data consisted of Jan 2019.

Snippet of input data format

```
           Date       Open       High        Low      Close  Adj Close  \
146  2013-01-02  79.117142  79.285713  77.375717  78.432854  55.923737
147  2013-01-03  78.268570  78.524284  77.285713  77.442856  55.217865
148  2013-01-04  76.709999  76.947144  75.118568  75.285713  53.679771
149  2013-01-07  74.571426  75.614288  73.599998  74.842857  53.364014
150  2013-01-08  75.601425  75.984283  74.464287  75.044289  53.507637

        Volume        MA7       MA21       MA20       MA60       MA42  \
146  140129500  74.660203  76.123333  75.742428  81.852952  78.372075
147   88241300  75.124897  75.823401  75.501356  81.556047  78.186905
148  148583400  75.264284  75.491088  75.417142  81.257024  78.017517
149  121039100  75.486734  75.389795  75.250428  80.984952  77.810986
150  114676800  75.695917  75.240612  75.193714  80.721762  77.615272

          MA63
146  82.472539
147  82.206576
148  81.902018
149  81.567437
150  81.246598
```

Sources :

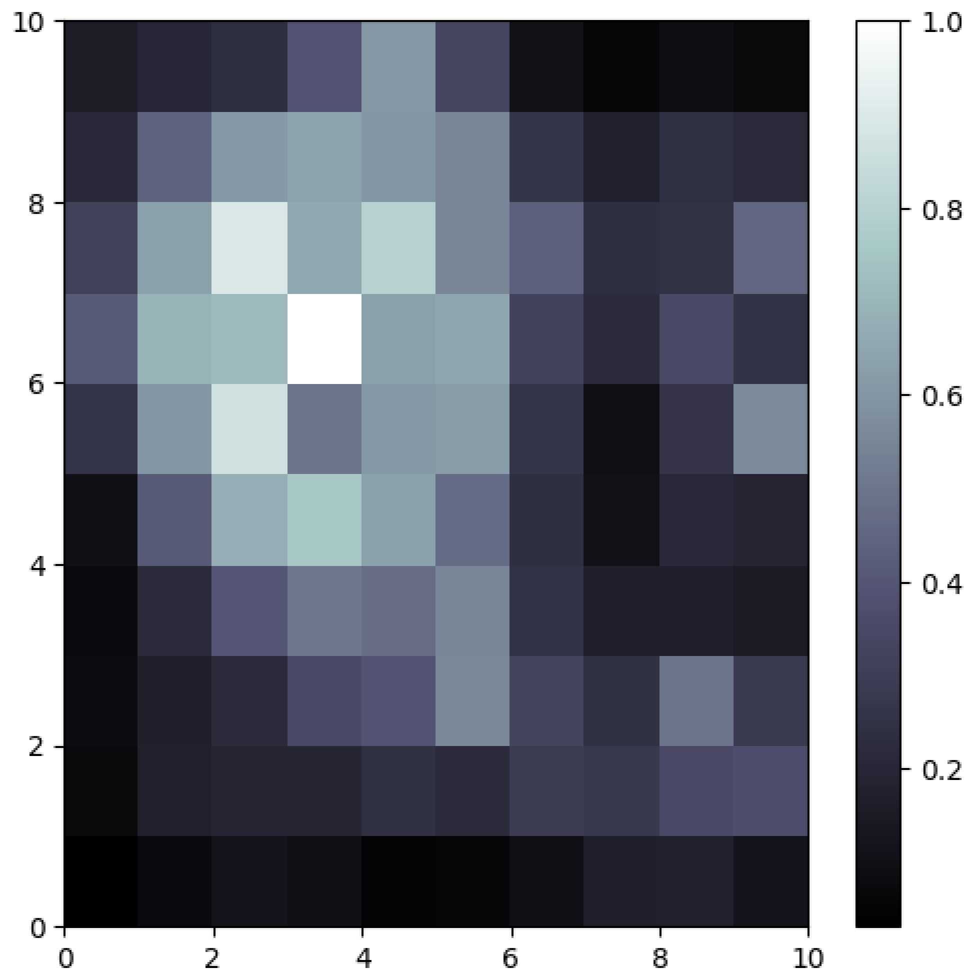
https://finance.yahoo.com/quote/
https://finviz.com
https://www.reuters.com/

DETECTION OF UNUSUAL MARKET BEHAVIOR

Technique - Self Organized Maps Unsupervised Deep Learning

- Each row observations are the inputs to the SOM.
- These input points are going to be mapped to an output space.
- We have a neural network composed of neurons between this input space and output space.
- Each neuron will be initialized with the as a vector of weights whose number = same as the vector size of a customer.
- Now, for each observation, a neuron will be picked that the customer is the closest to.
- This neuron is called the winning node. The winning node is the most similar neuron to the input observation.
- We can use Gaussian Neighbourhood function to update the weights of the neighbours of the winning node to move them closer to the point.
- We do this for all the observations and we repeat it many times.
- Each time , the output space dimension keeps decreasing and stops at a point.

The self organizing map returned on the Apple stock prices dataset



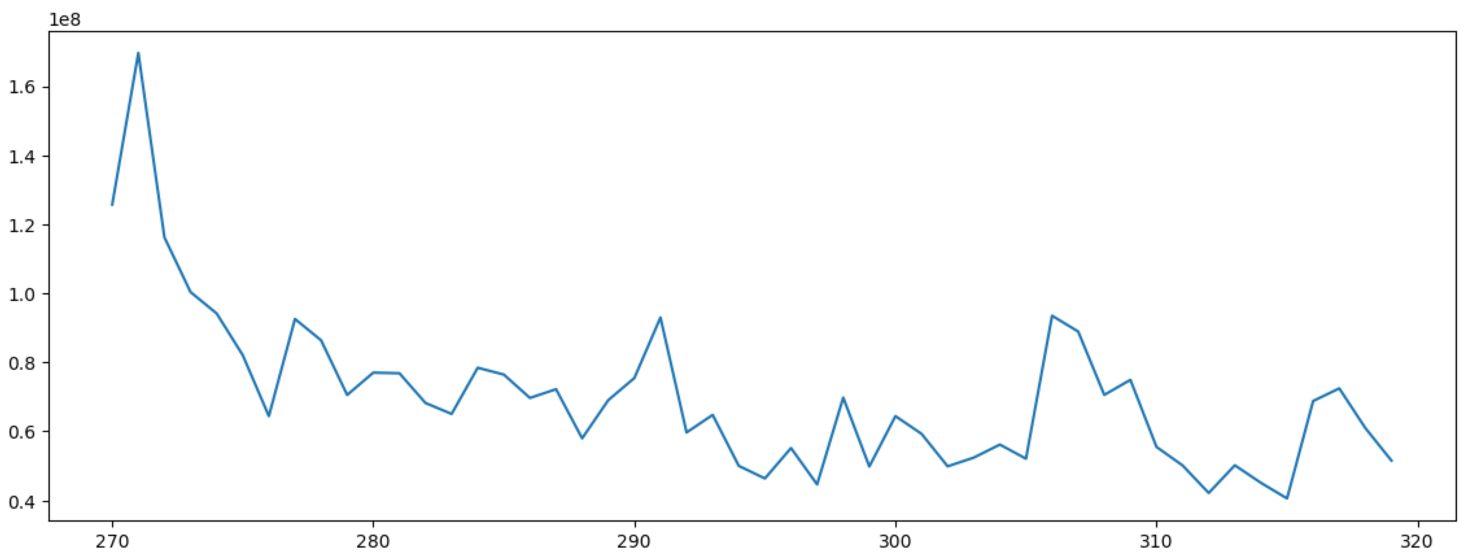
Example Detection of Unusual Behavior :

The following observations were detected as unusual by the Self Organizing Maps

25	270.0	71.992859	72.481430	71.231430	71.535713	125702500.0
26	272.0	70.739998	71.647141	70.507141	71.514282	116199300.0
27	273.0	71.801430	72.532860	71.328575	71.647141	100366000.0
28	274.0	72.264282	72.779999	71.822861	72.684288	94170300.0
29	277.0	74.482857	74.704285	73.911430	74.239998	92570100.0

The first column represents - Close price
The last column represents - Volume

We plot a graph of Close price vs Volume



We clearly verify that the market in fact showed unusual behavior around that range of days.

FUNDAMENTAL ANALYSIS - TO MAKE INVESTMENT DECISIONS

The following indicators and their values were chosen for fundamental analysis to determine which stocks are causing loss/profit in the portfolio.

	Company	P/B	P/E	Forward P/E	PEG	Debt/Eq	EPS (ttm)	Dividend %	ROE	ROI	EPS Q/Q	Insider Own
0	Apple	6.81	14.20	13.30	1.09	0.97	11.94	1.72%	50.90%	26.60%	0004.80%	0.07%
1	IBM	6.17	11.38	9.46	11.86	2.37	11.77	4.69%	31.10%	17.40%	0.70%	0.10%
2	Goldman Sachs	0.98	7.96	7.17	1.24	7.08	24.05	1.67%	6.80%	1.50%	25.20%	0.40%
3	Amazon	17.90	80.22	40.12	1.83	1.13	19.83	000	27.00%	11.90%	166.60%	16.10%
4	General Electric	2.77	000	10.95	000	3.66	0002.40	0.40%	000	0001.90%	000937.50%	0.15%
5	Google	4.29	25.06	23.27	1.53	000	43.70	000	000	000	000	000

The following criteria were chosen:

1. Businesses which are quoted at low valuations

P/E < 20
P/B < 3

2. Businesses which have demonstrated earning power¶

EPS Q/Q > 10%

3. Businesses earning good returns on equity while employing little or no debt

Debt/Eq < 1
ROE > 10%

4. Management having substantial ownership in the business¶

Insider own > 30%

Results

Companies likely to maximize profit in the Portfolio¶

- Goldman Sachs
- Amazon
- Apple

Companies likely to cause loss in the Portfolio

- IBM

TECHNICAL ANALYSIS - TO GET THE MOST IMPORTANT INDICATORS

Technique - Feature importances through XGBRegressor

a. The following technical indicators were derived from the OPEN, CLOSE, HIGH, LOW, VOLUME data.

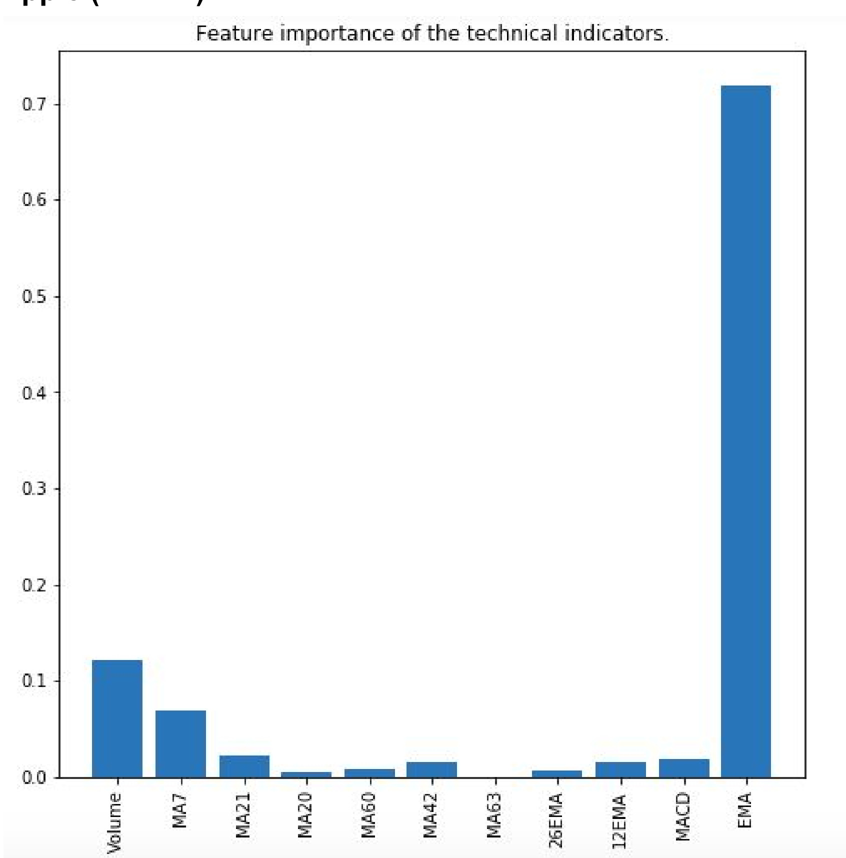
- moving average for 7 days
- moving average for 21 days
- moving average for 20 days
- moving average for 60 days
- moving average for 42 days
- moving average for 63 days
- Moving Average Convergence Divergence (12EMA - 26EMA)
- Exponential Moving Average

b. It was followed by fitting an XGBRegressor over it to obtain the most important features.

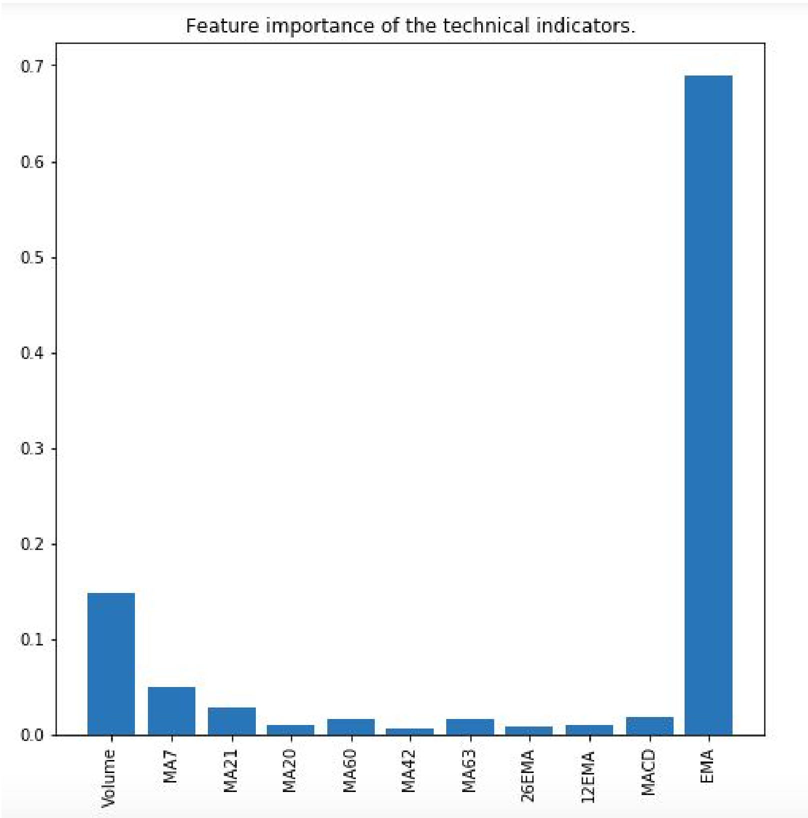
c. Based on the particular stock data, the most relevant indicators were chosen to be fed into the RNN - LSTM model.

Feature Importances of all stocks in Portfolio

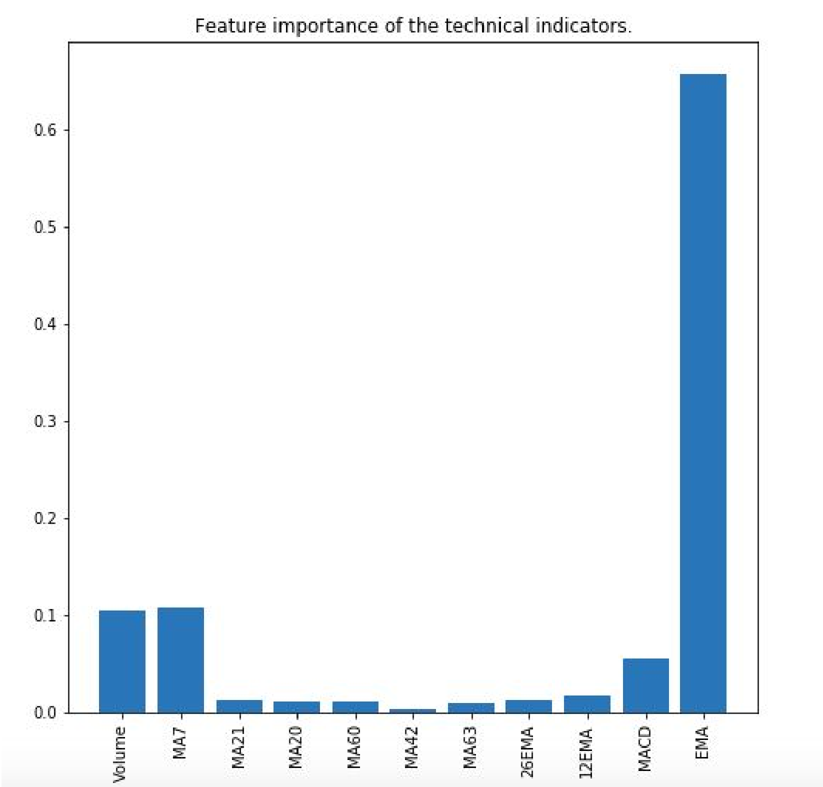
Apple (AAPL)



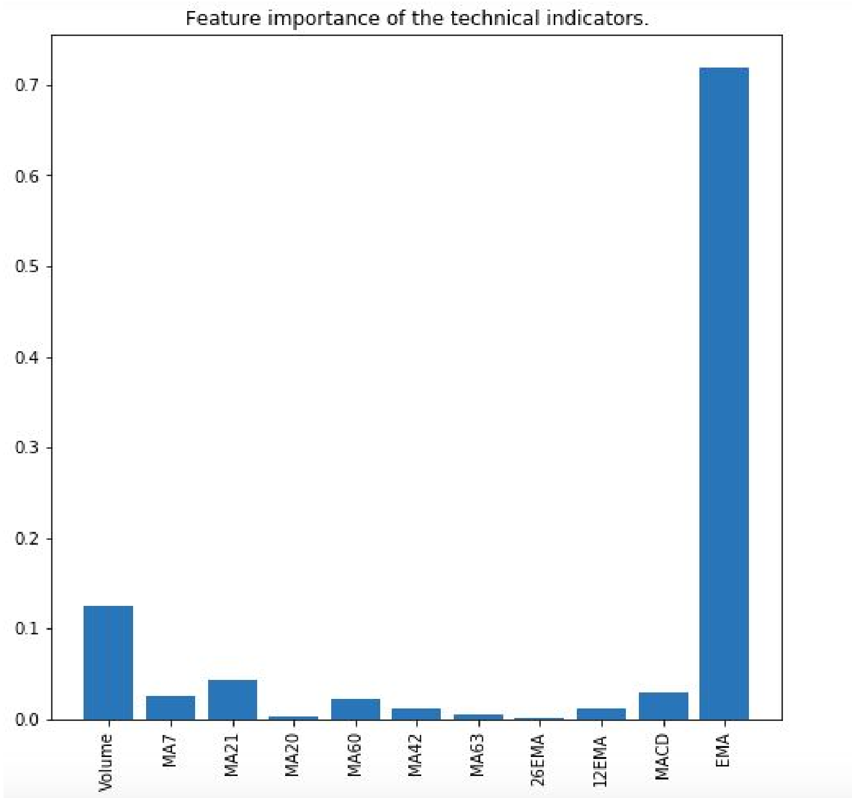
IBM



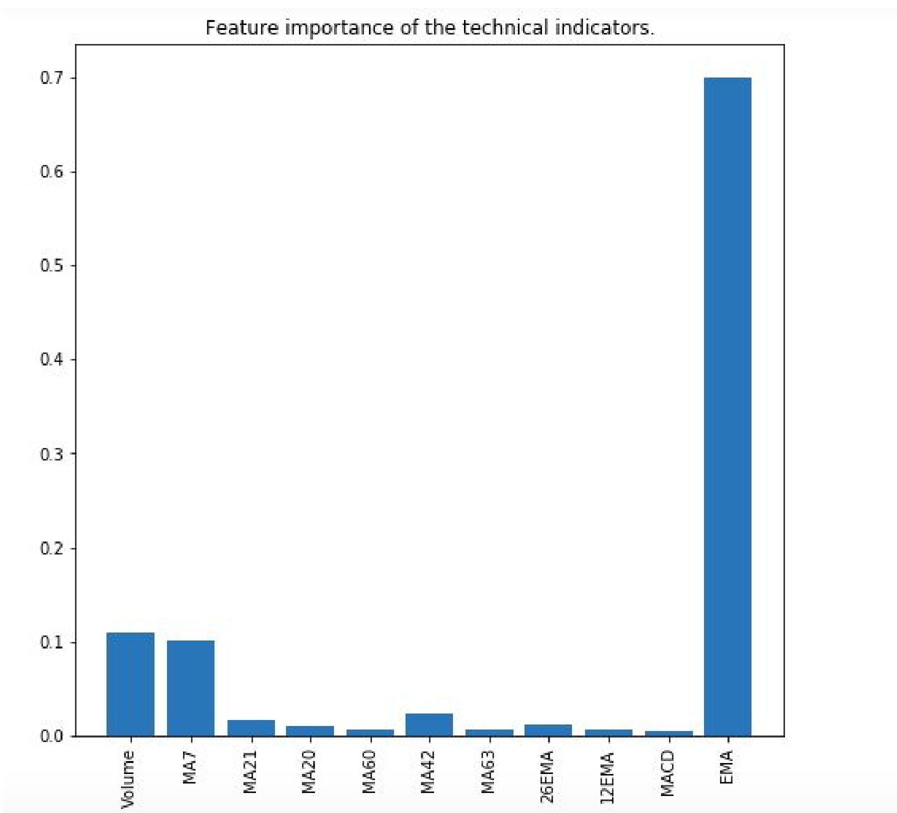
Goldman Sachs (GS)



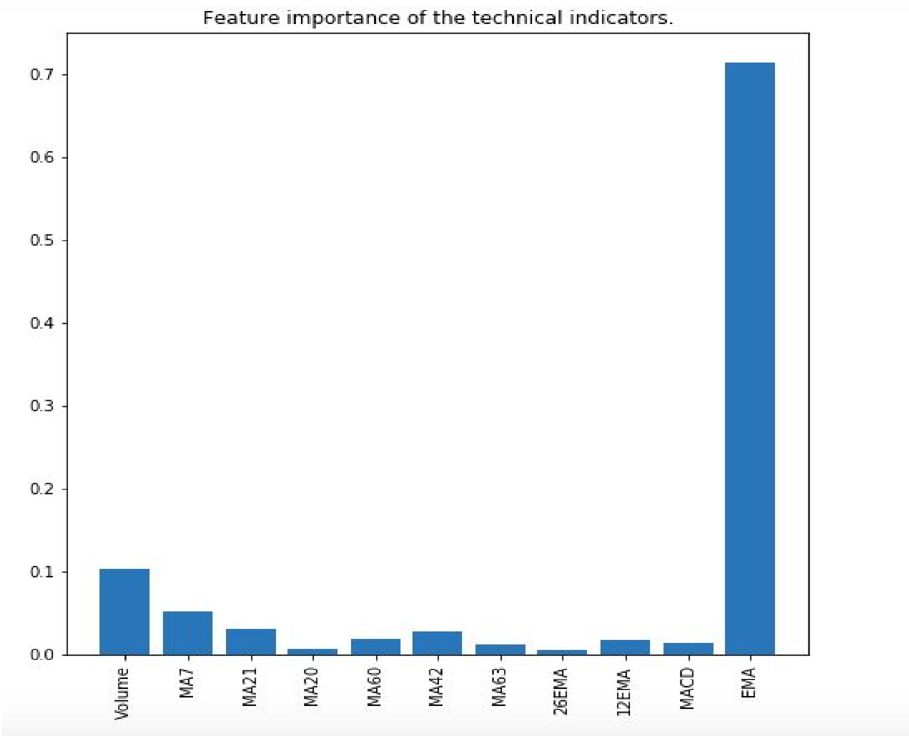
Amazon (AMZN)



General Electric (GE)



Google (GOOG)


Feature importance of the technical indicators.
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Volume
MA7
MA21
MA20
MA60
MA42
MA63
26EMA
12EMA
MACD
EMA

PREDICTIVE MODEL - SHORT TERM PRICE PREDICTION

Technique - RNN - LSTM

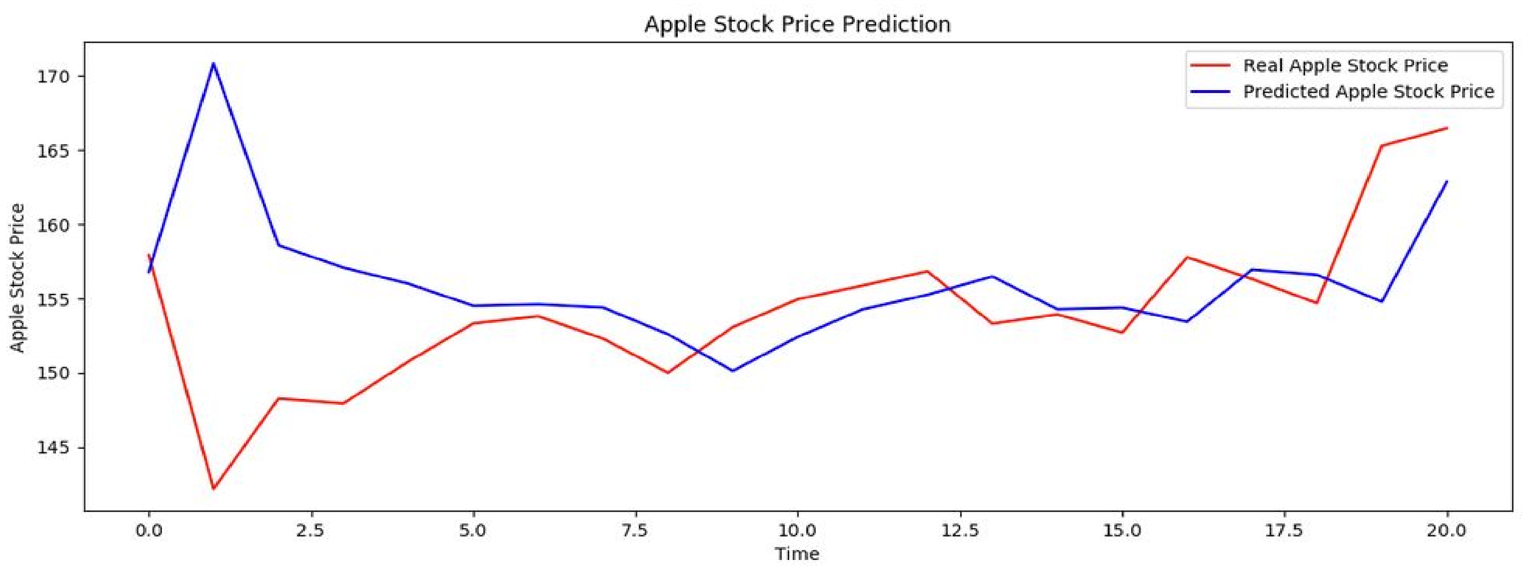
The following steps have been taken in creating a predictive stock market model:

- Data preparation for time - series stock market model
- Building the model
- Training the model
- Predicting the stock prices
- Evaluating the model
- Visualizing the model

The models were trained over the years Jan 2013 - Dec 2018
The prediction was done on Jan 2019

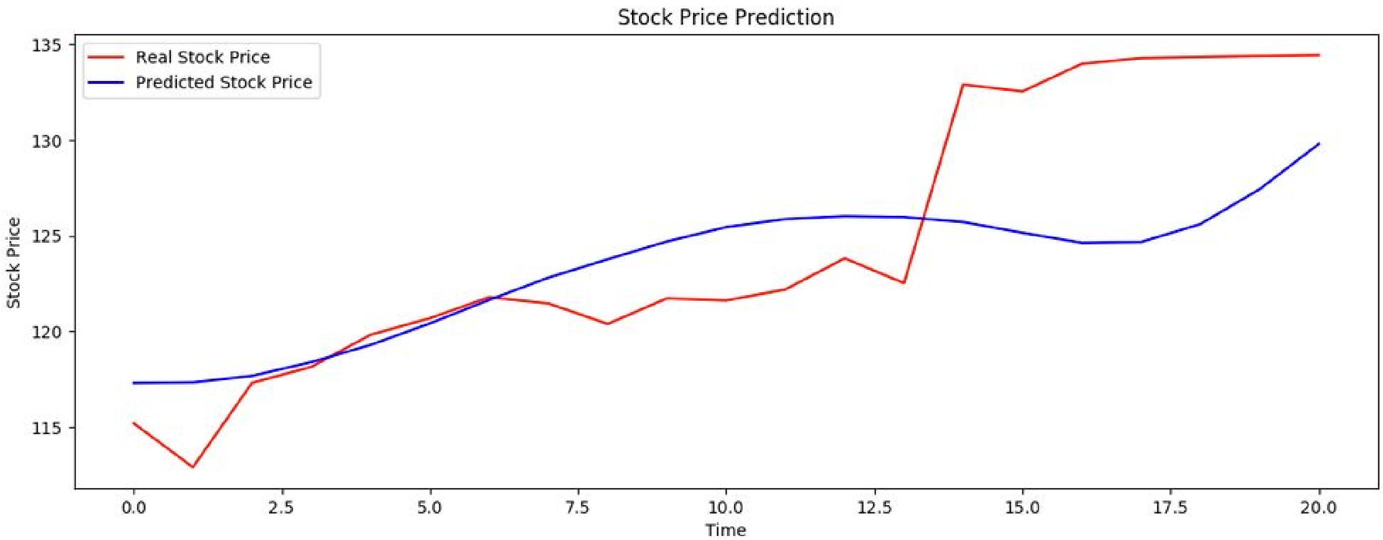
The following visualizations show the predicted and actual prices for 20 days.

Apple (AAPL)



Interpretation : We observe that the model is doing a pretty good job in predicting the trends except in a few places of unusual market behaviors. We can fix this by considering more factors such as breaking news headlines for Apple.

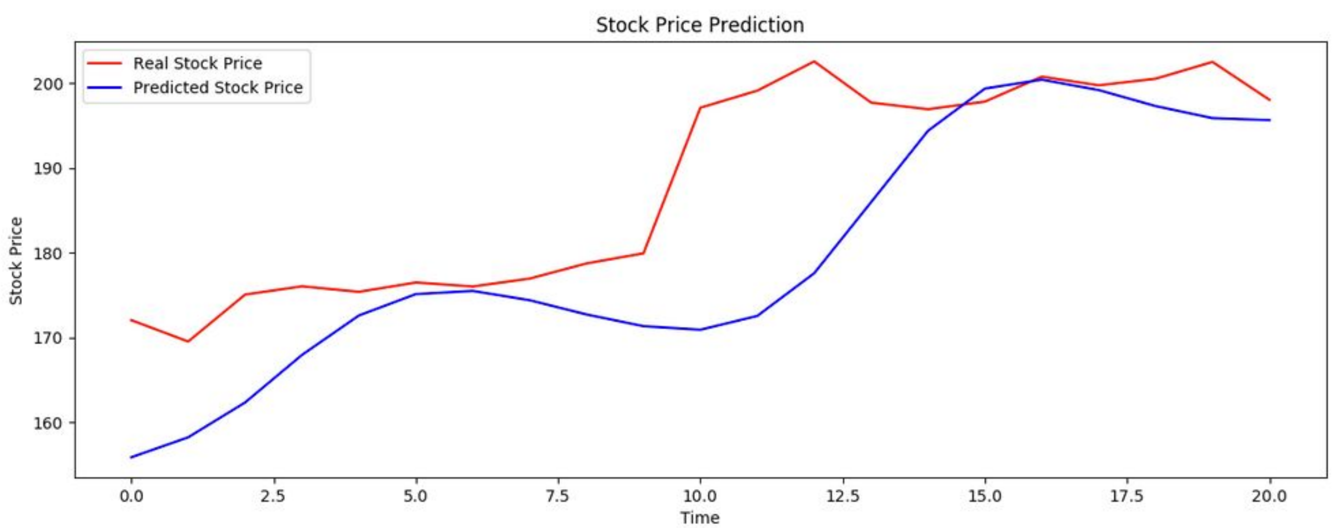
IBM




```
0.0400640889989
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1]
Predicted values matched the actual direction 80.95% of the time.
```

Interpretation : Overall, the model is doing a good job in predicting the market trends for IBM

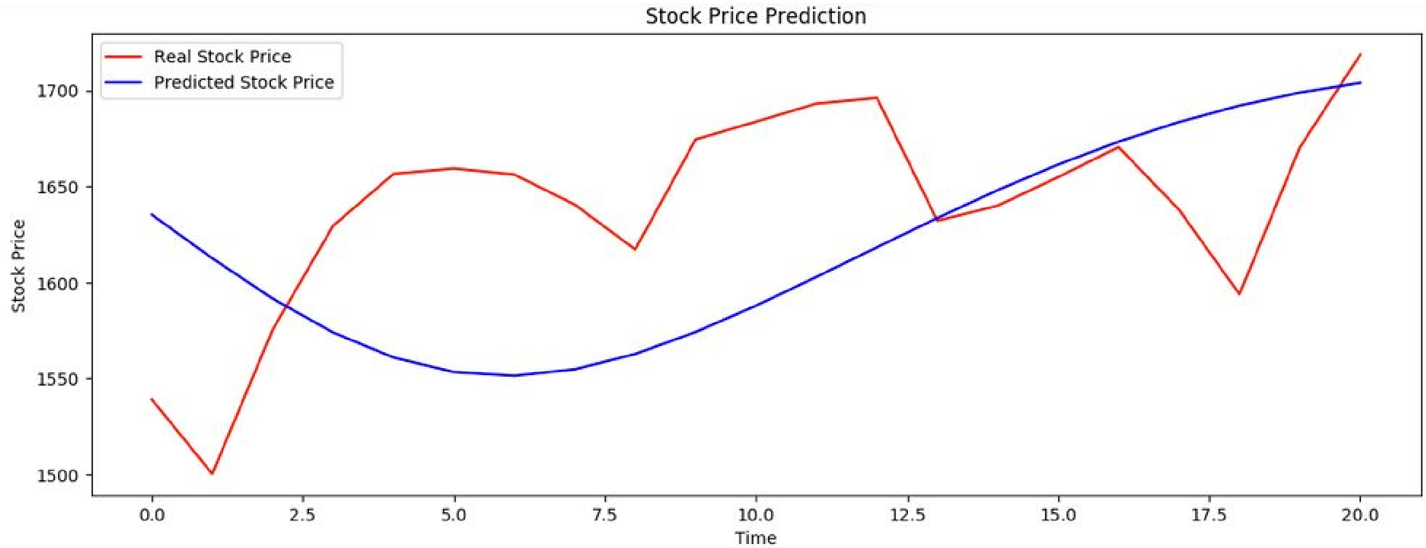
Goldman Sachs (GS)




```
0.0631925556634
Prediction Direction [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]
Real Direction [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Predicted values matched the actual direction 85.71% of the time.
```

Interpretation : Overall, the model is doing a good job in predicting the market trends for Goldman Sachs.

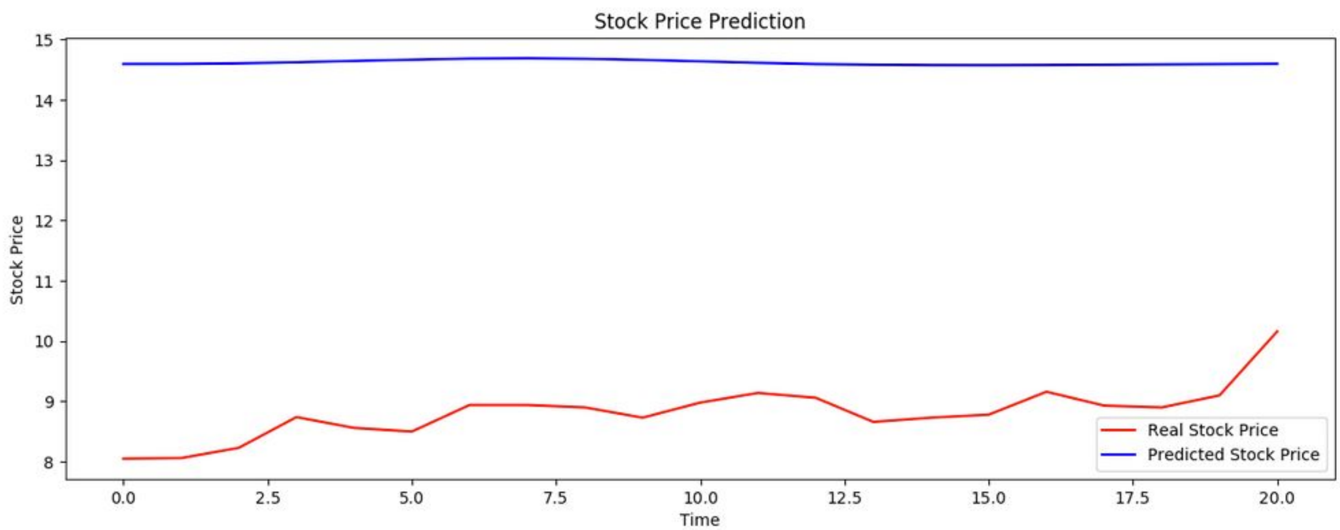
Amazon (AMZN)




```
0.0447874653288
Prediction Direction [1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Real Direction [0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Predicted values matched the actual direction 52.38% of the time.
```

Interpretation : The model seems to fail in predicting the trends for Amazon. We must include more features in training the model.

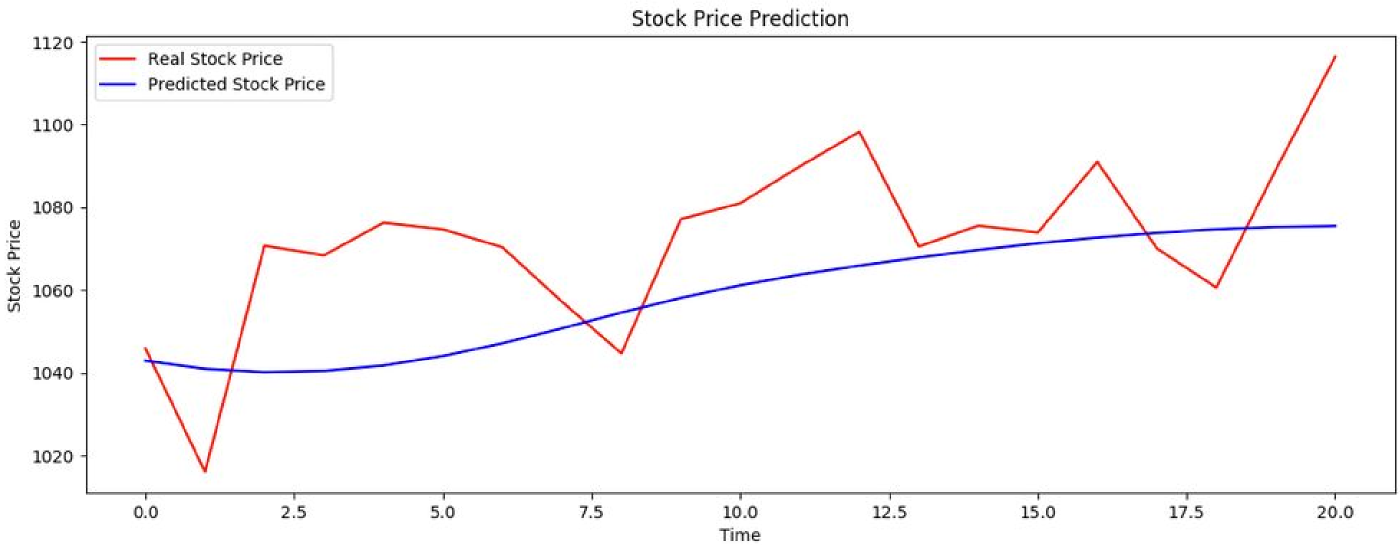
General Electric (GE)




```
0.658892774872
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Predicted values matched the actual direction 100.0% of the time.
```

Interpretation : Even though, the model seems to be performing poorly, it is in fact doing pretty well because the average difference between the predicted and actual prices is about 7 points. However the difference seems to be enlarged because General Electric has a very low range of CLOSE price. We may have to change the architecture of the model to fix this problem.

Google (GOOG)




```
0.020448544494
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Real Direction [1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Predicted values matched the actual direction 95.24000000000001% of the time.
```

Interpretation : We see that the model is doing a pretty good job in predicting the market trends for Google. However, it is perhaps generalizing a little too much. We can fix this by training the model over more number of epochs.

PREDICTIVE MODEL WITH NEWS HEADLINES

Technique - Bidirectional LSTM

We achieve the best results with this multi modal algorithm which combines news headlines, moving averages, and other factors for a specific stock into one model to predict the stock price on a given day.

The input data for the model is of the following format

	Date	Close	Volume	MA7	MA21	MA42	MACD	EMA	Article_Title
0	2018-01-02	172.259995	25555900.0	171.965714	171.909047	172.157619	0.386977	171.455446	Breakingviews - TV content wars will have gris...
1	2018-01-03	172.229996	29517900.0	171.568571	171.965238	172.284762	0.375116	171.971813	Breakingviews - Tech salad will come with a si...
2	2018-01-03	172.229996	29517900.0	171.568571	171.965238	172.284762	0.375116	171.971813	SEC mixes message on Apple shareholder proposa...
3	2018-01-03	172.229996	29517900.0	171.568571	171.965238	172.284762	0.375116	171.971813	SEC mixes message on Apple shareholder proposa...
4	2018-01-04	173.029999	22434600.0	171.285714	172.119047	172.401905	0.425366	172.677270	
5	2018-01-05	175.000000	23660000.0	171.918571	172.374285	172.461429	0.617039	174.225757	UPDATE 2-Apple to issue fix for iPhones, Macs ...
6	2018-01-05	175.000000	23660000.0	171.918571	172.374285	172.461429	0.617039	174.225757	Apple to issue fix for iPhones, Macs at risk f...
7	2018-01-05	175.000000	23660000.0	171.918571	172.374285	172.461429	0.617039	174.225757	Friday Morning Briefing
8	2018-01-08	174.350006	20567800.0	172.454285	172.628571	172.463810	0.708328	174.308590	Monday Morning Briefing
9	2018-01-08	174.350006	20567800.0	172.454285	172.628571	172.463810	0.708328	174.308590	REFILE-Apple should address youth phone addict...

The multi modal learning algorithm model consists of the following steps:

1. - Data Collection
2. - Formatting 'Date' from news headlines to match with 'Date' from numerical stock data
3. - Merging the two datasets on unique dates from stock data
4. - Cleaning the text to remove unwanted characters, contradictions, stopwords etc
5. - Creating a Dictionary for each unique word and it's counts
6. - Creating a Dictionary for each word in Glove embedding and it's embedding representation
7. - Creating a Dictionary to convert words to integer representations
8. - Creating a reverse mapping Dictionary to convert integer representations to their words
9. - Creating a word embedding matrix for each unique word in the dataset and its embedding representation
10. - Create integer representation of all the news headlines
11. - Normalize the data
12. - Build the architecture of the RNN model
13. - Train it
14. - Test and Evaluate

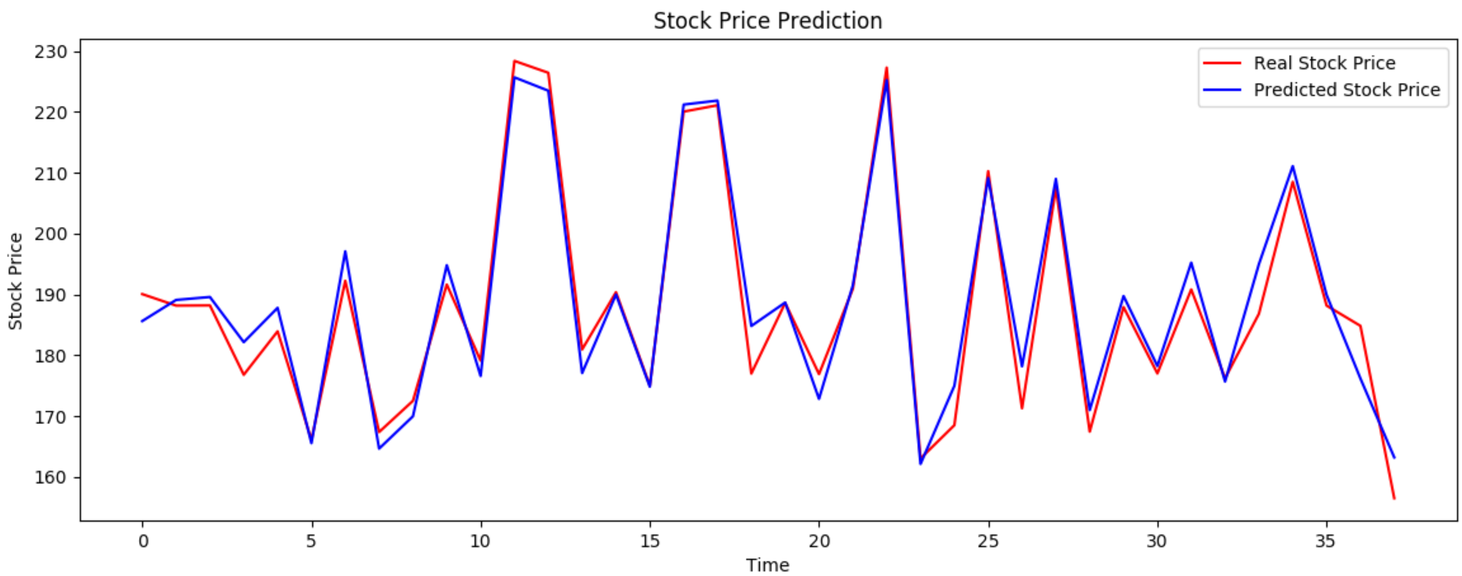
Market Trends for APPLE

```
print("Predicted Direction", direction_pred)
print("Actual Direction", direction_test)
```

```
Predicted Direction [0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1,
0, 1, 1, 1, 0, 0]
Actual Direction [1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0,
0, 1, 1, 0, 0]
```

```
direction = acc(direction_test, direction_pred)
direction = round(direction,4)*100
print("Predicted values matched the actual direction {}% of the time.".format(direction))
```

```
Predicted values matched the actual direction 92.11% of the time.
```

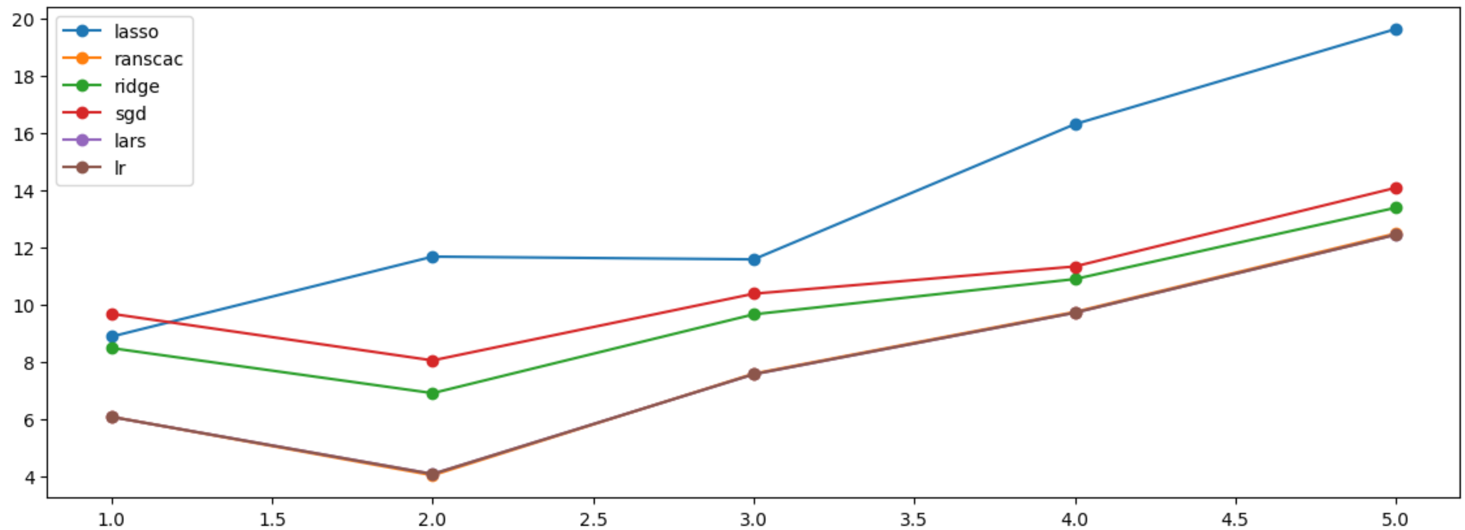


Interpretation : **As observed above, we notice that the algorithm is predicting the market trends of Apple 92.11% of the time which is excellent. This is clearly an improvement over the previous LSTM model which did not take news headlines into consideration.**

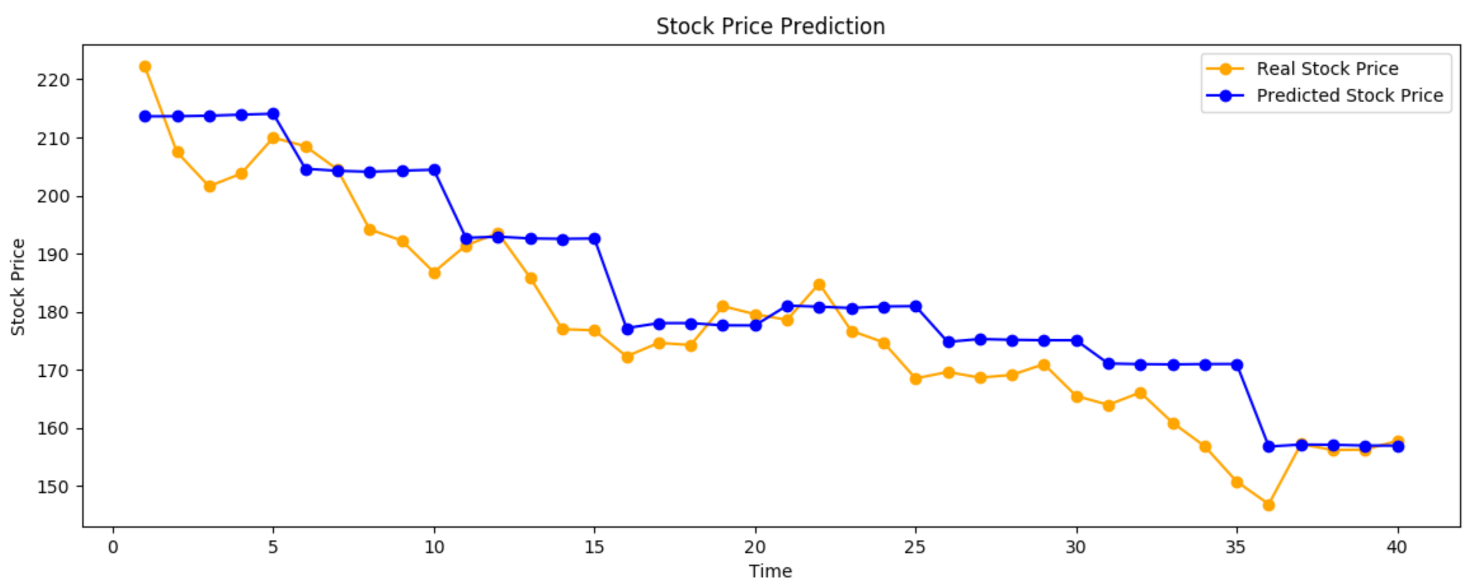
Recursive Strategy

Apple (AAPL)

Loss comparison of models



Actual vs Predicted Prices

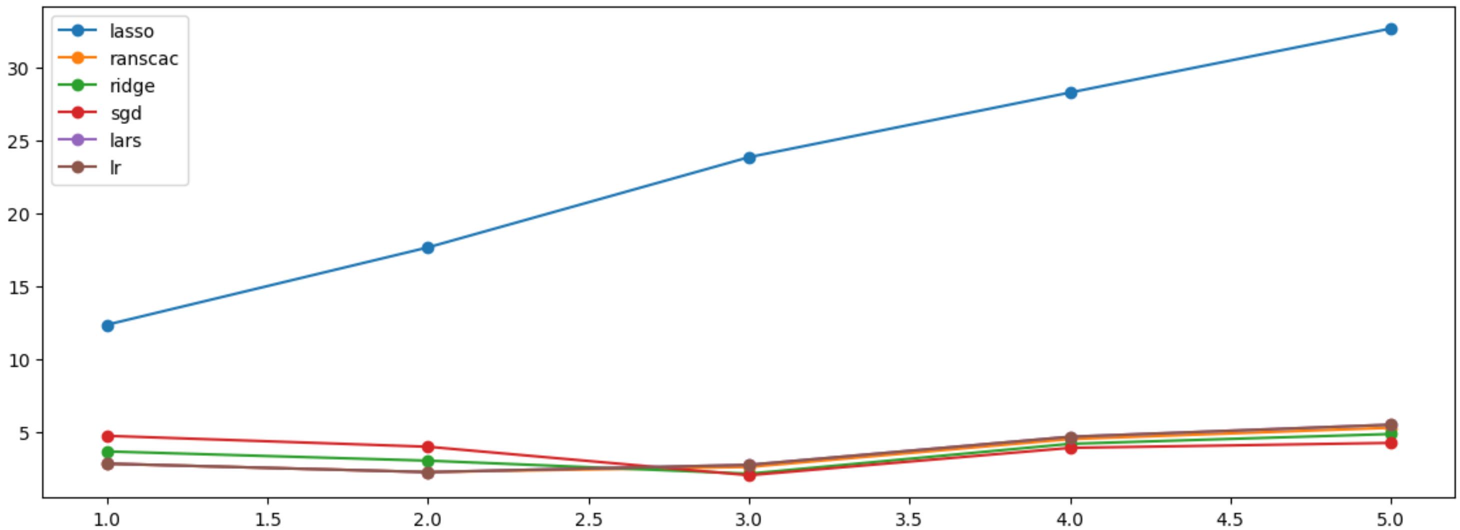



```
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 72.5% of the time.
```

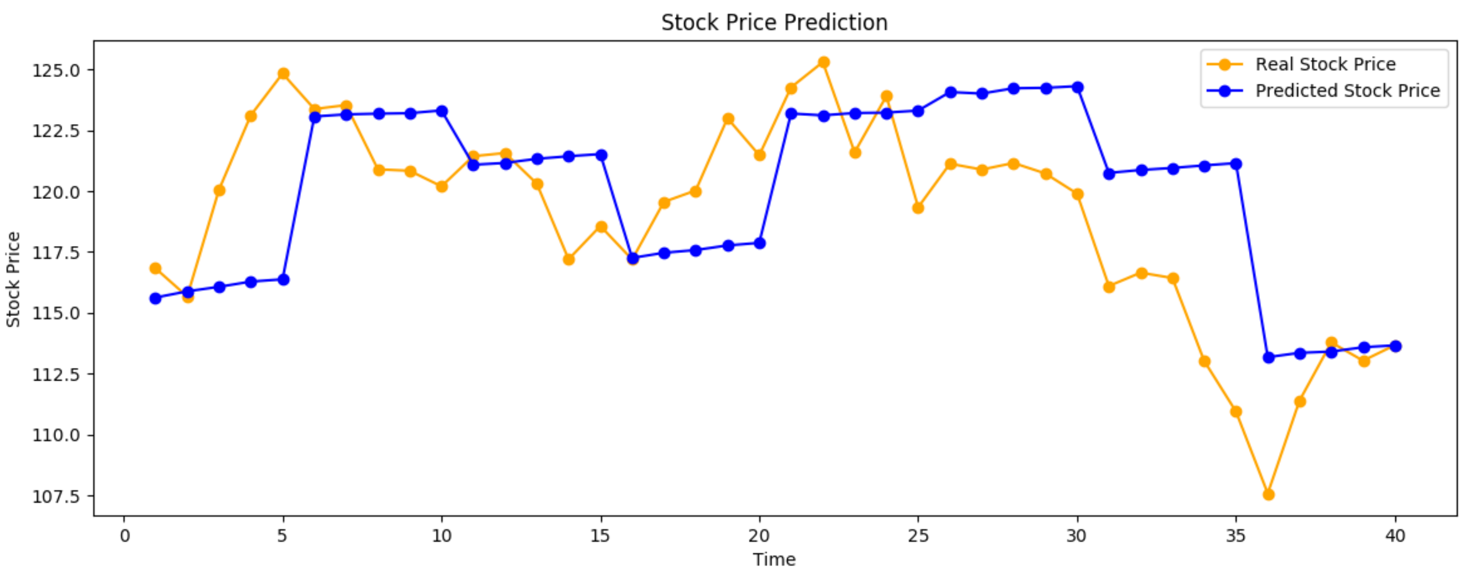
Interpretation

IBM

Loss comparison of models



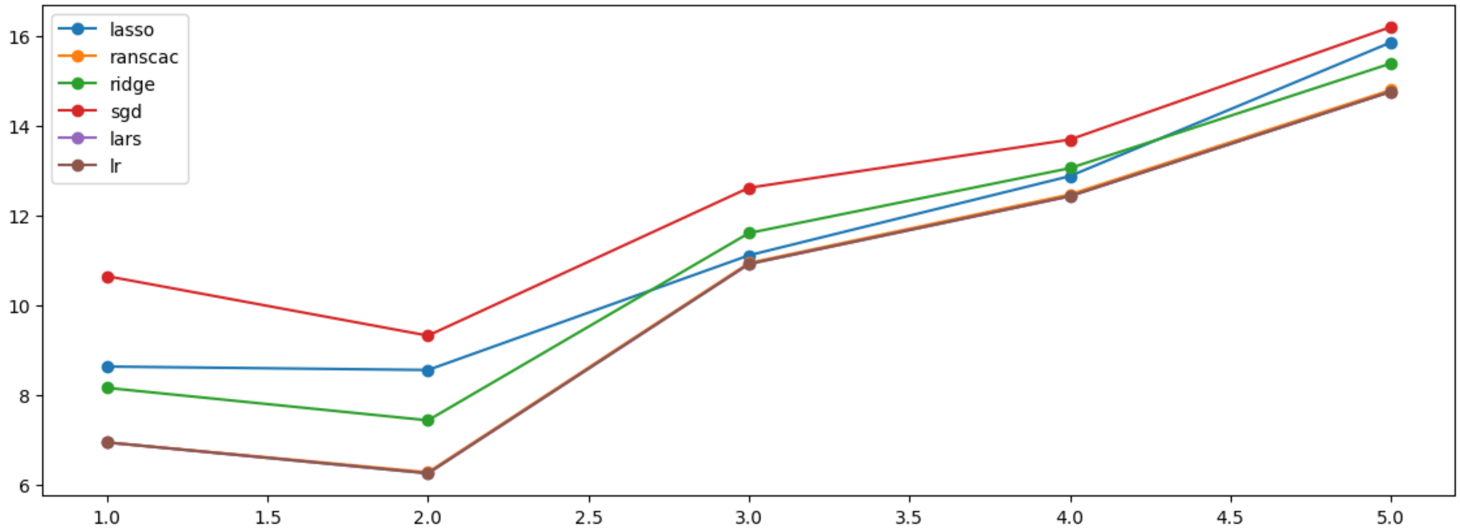
Actual vs Predicted Prices



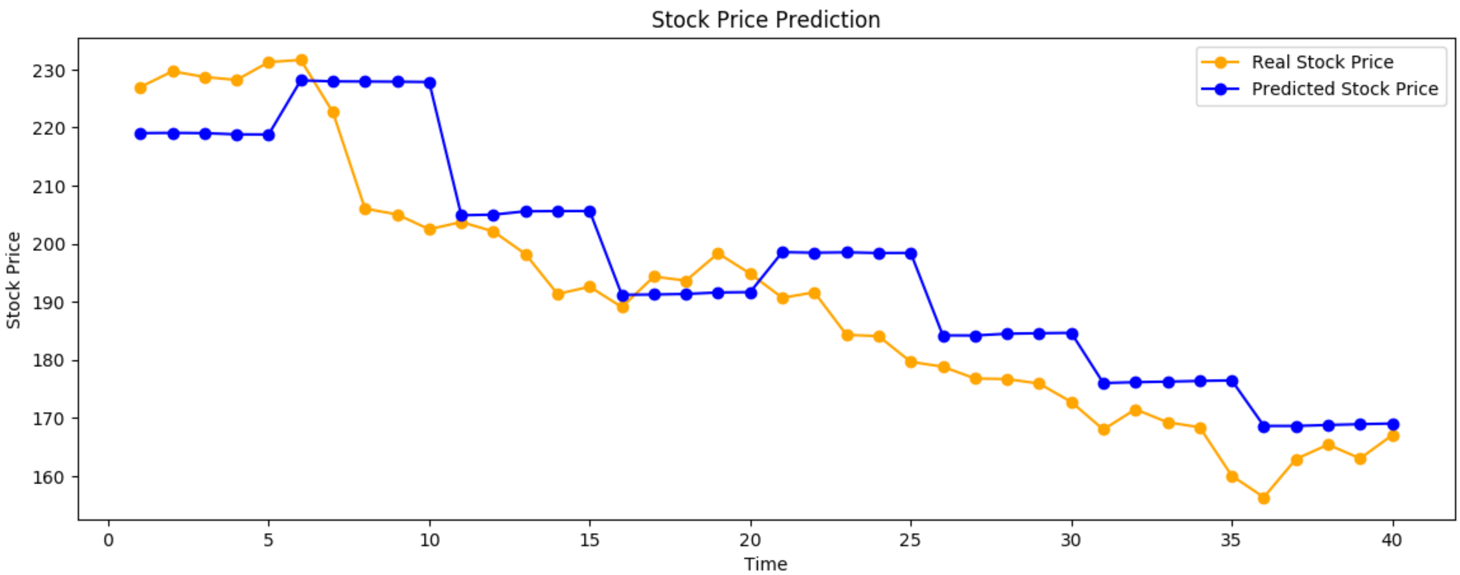

```
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 0, 0, 0, 0, 0]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 87.5% of the time.
```

Goldman Sachs (GS)

Loss comparison of models



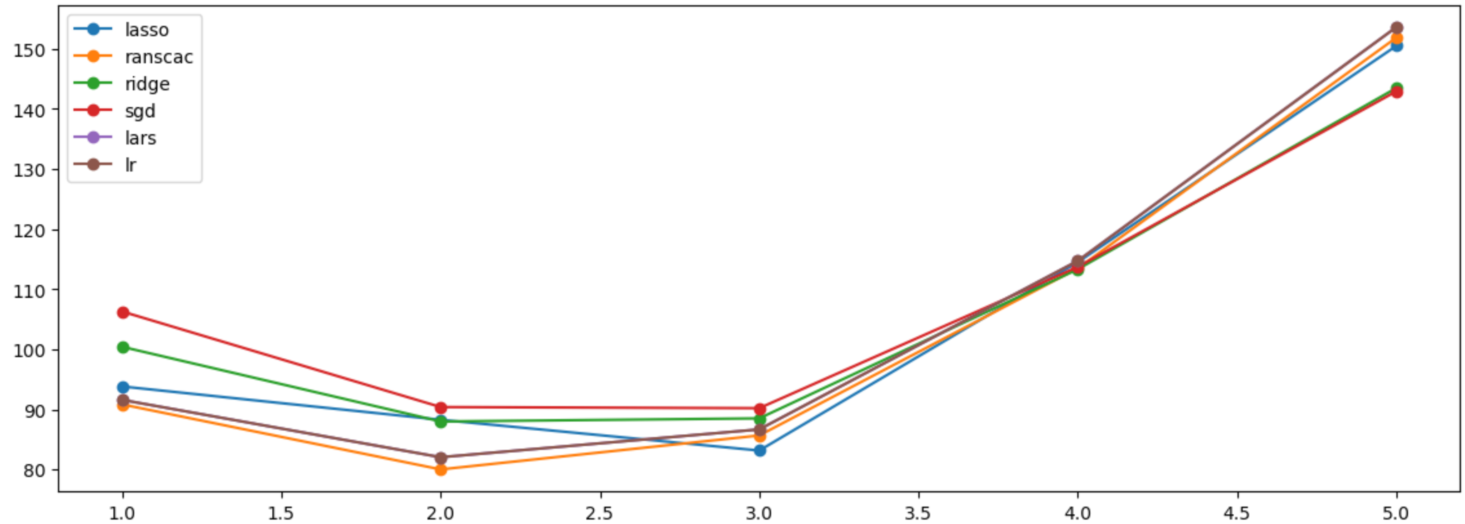
Actual vs Predicted Prices



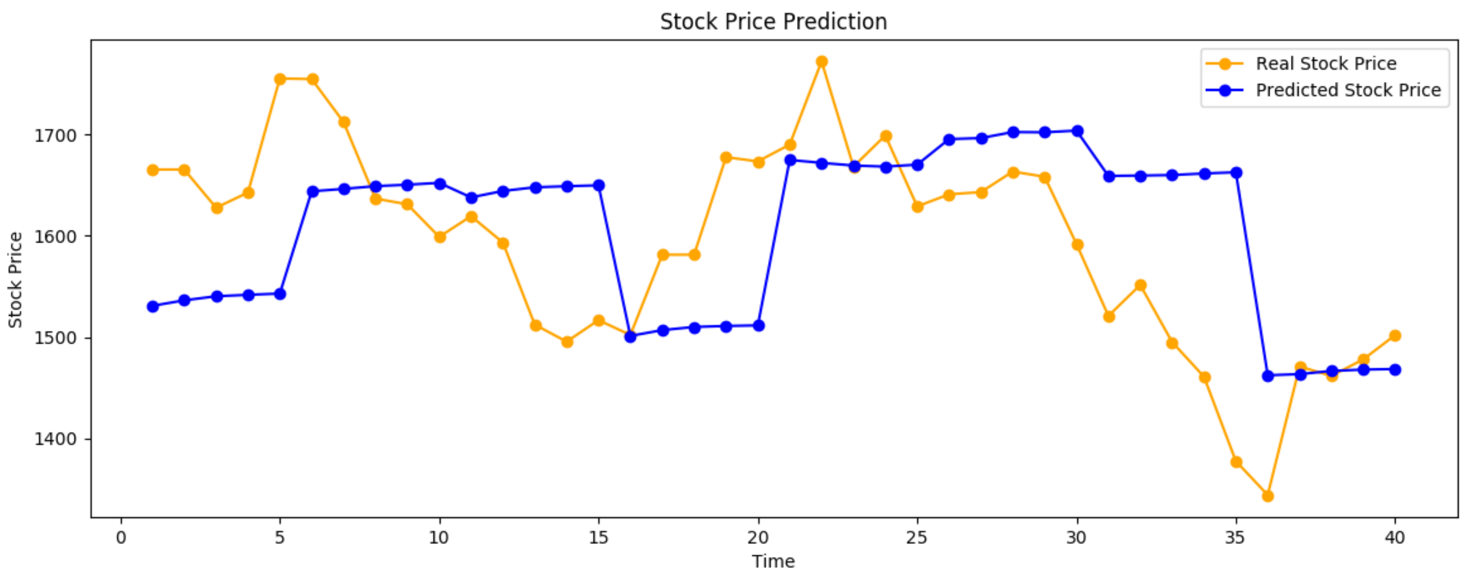

```
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 62.5% of the time.
```

Amazon (AMZN)

Loss comparison of models



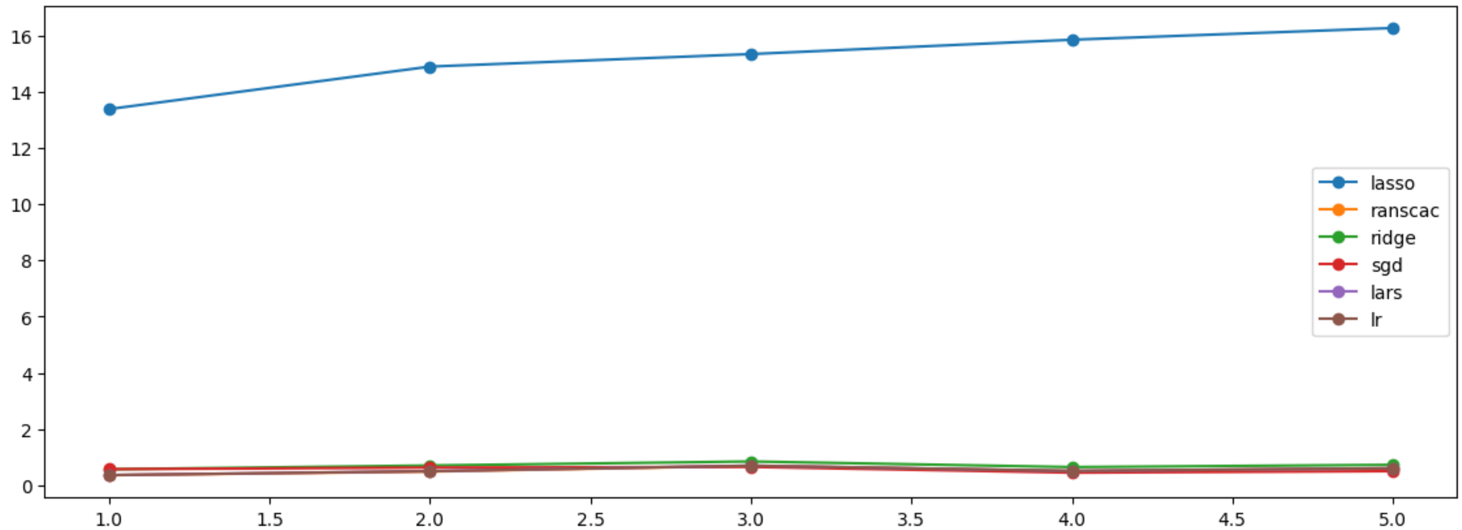
Actual vs Predicted Prices



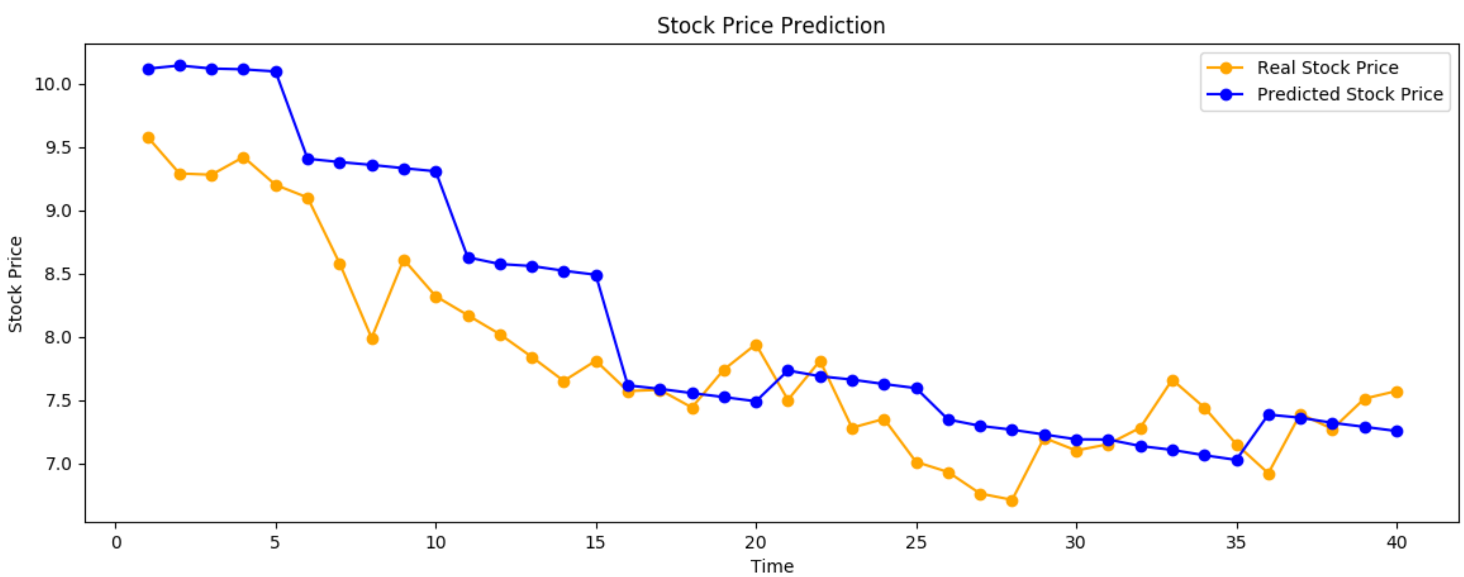

```
Prediction Direction [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 0, 0, 0, 0, 0]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 62.5% of the time.
```

General Electric (GE)

Loss comparison of models



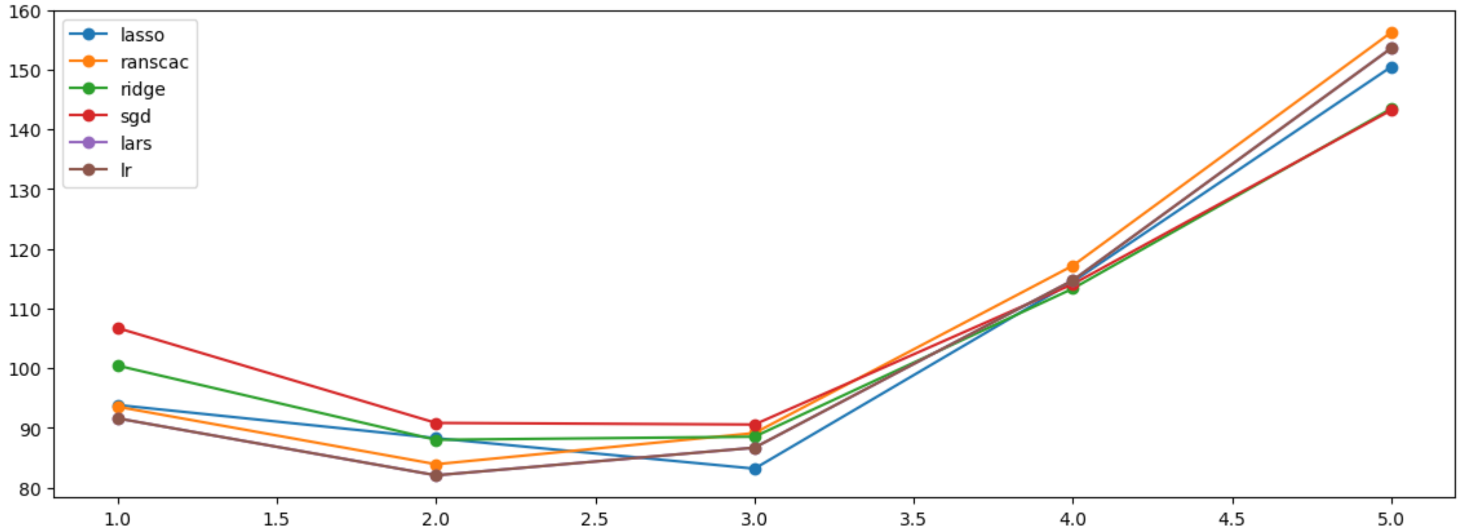
Actual vs Predicted Prices



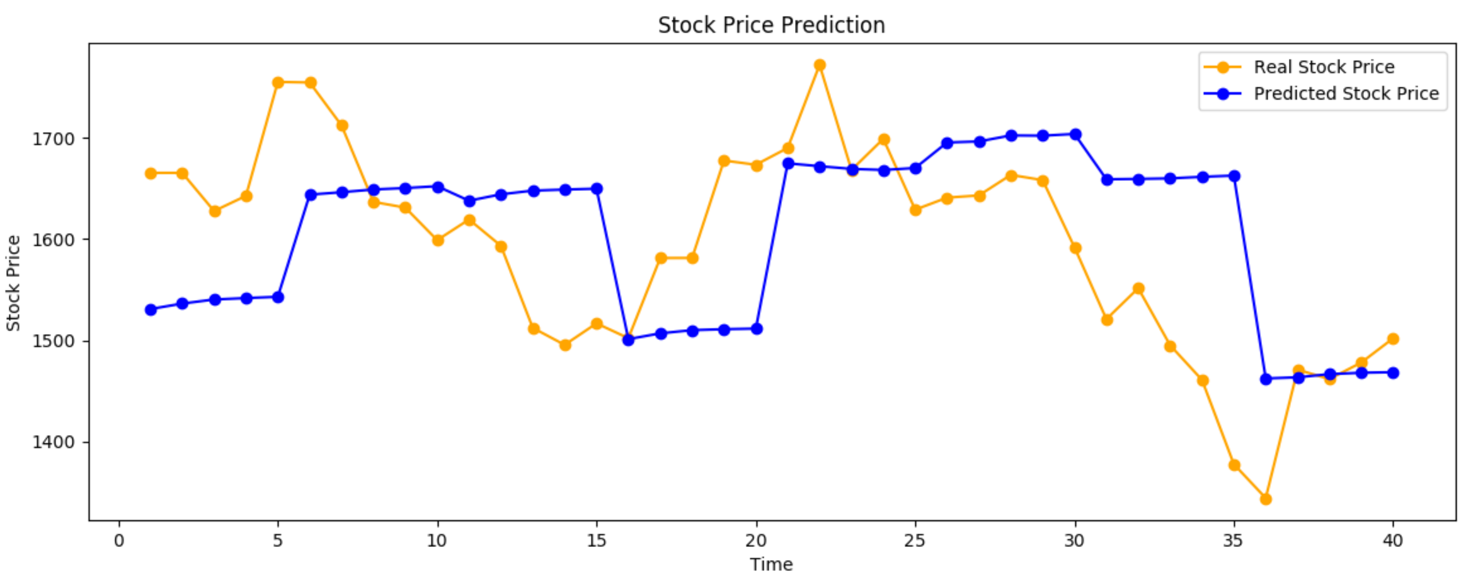

```
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 100.0% of the time.
```

Google (GOOG)

Loss comparison of models



Actual vs Predicted Prices

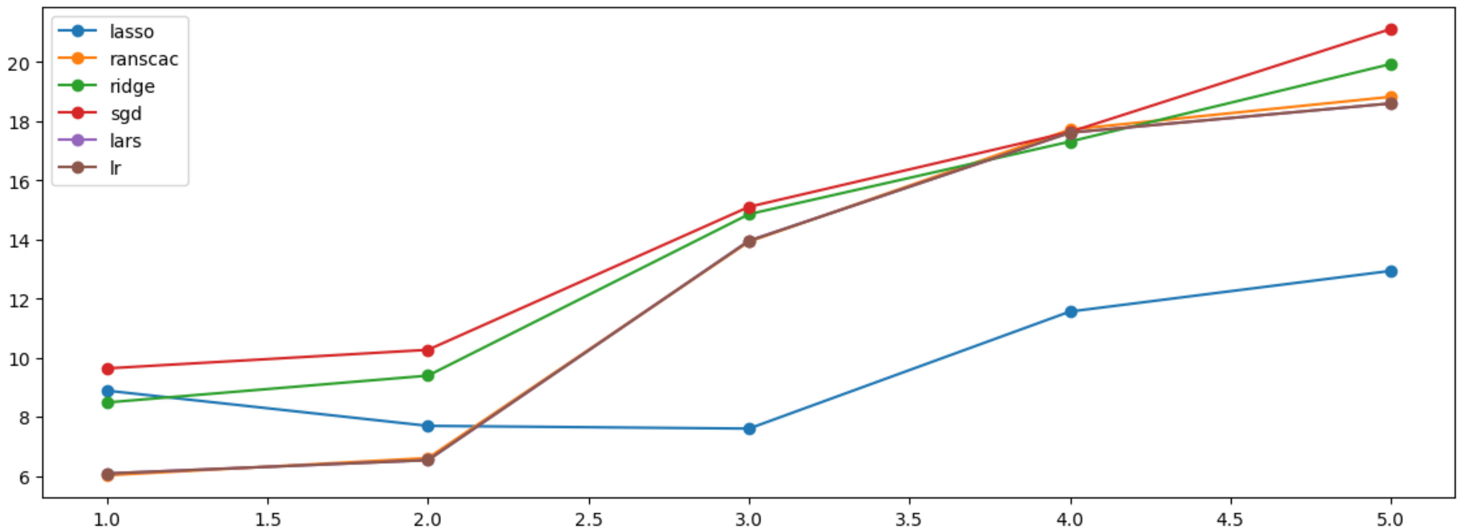



```
Prediction Direction [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 0, 0, 0, 0, 0]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 62.5% of the time.
```

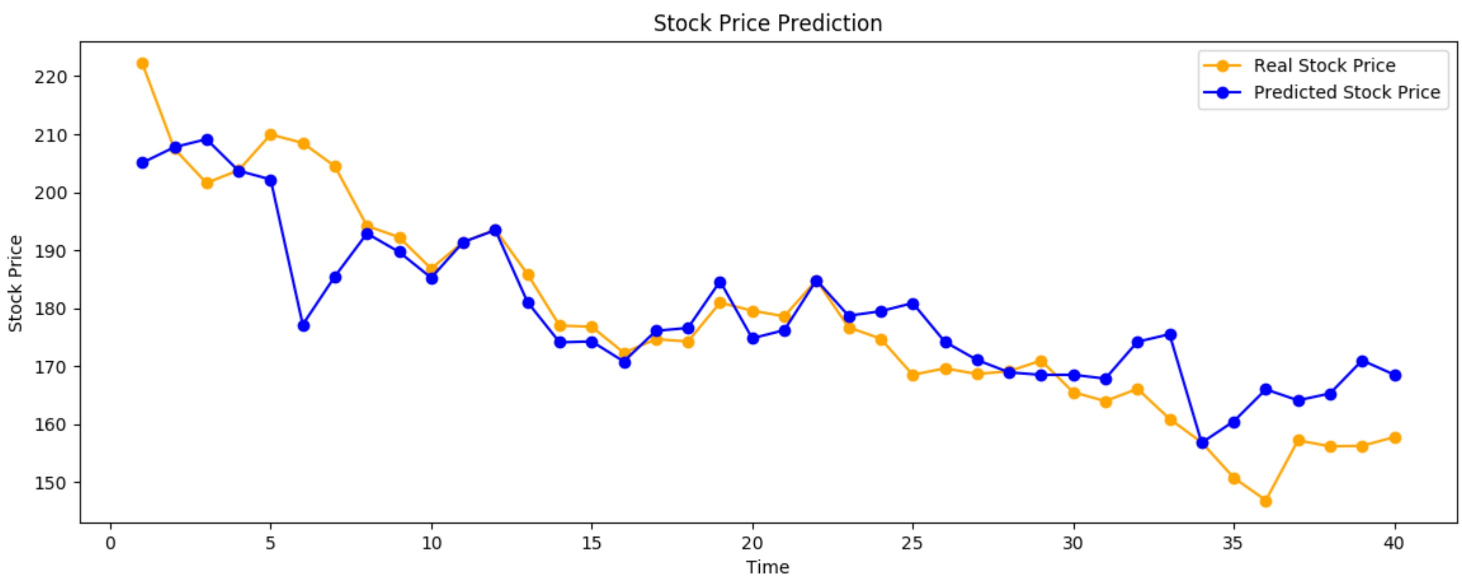
Direct Strategy

Apple (AAPL)

Loss comparison of models



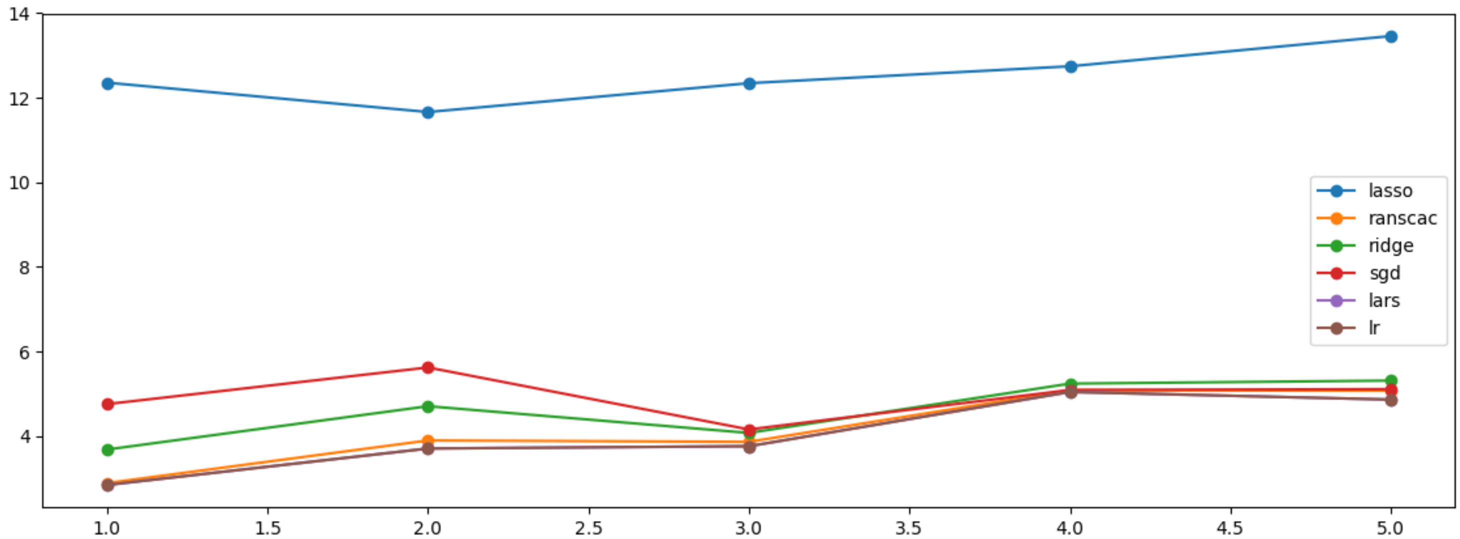
Actual vs Predicted Prices



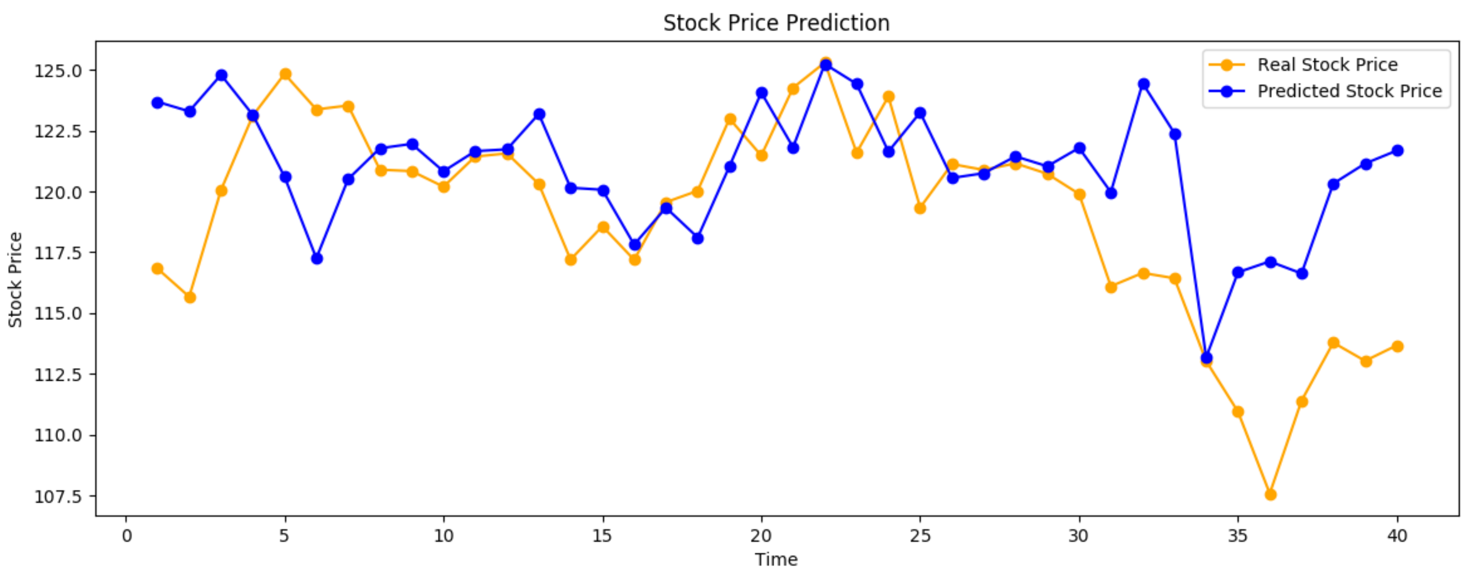

```
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 55.00000000000001% of the time.
```

IBM

Loss comparison of models



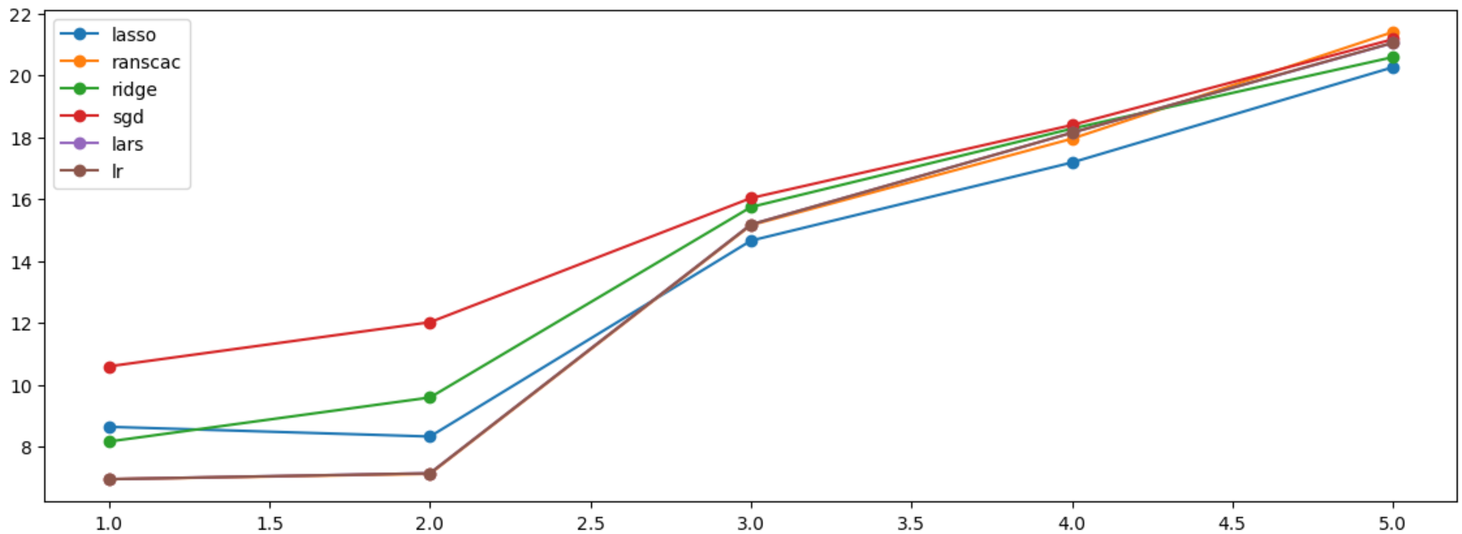
Actual vs Predicted Prices



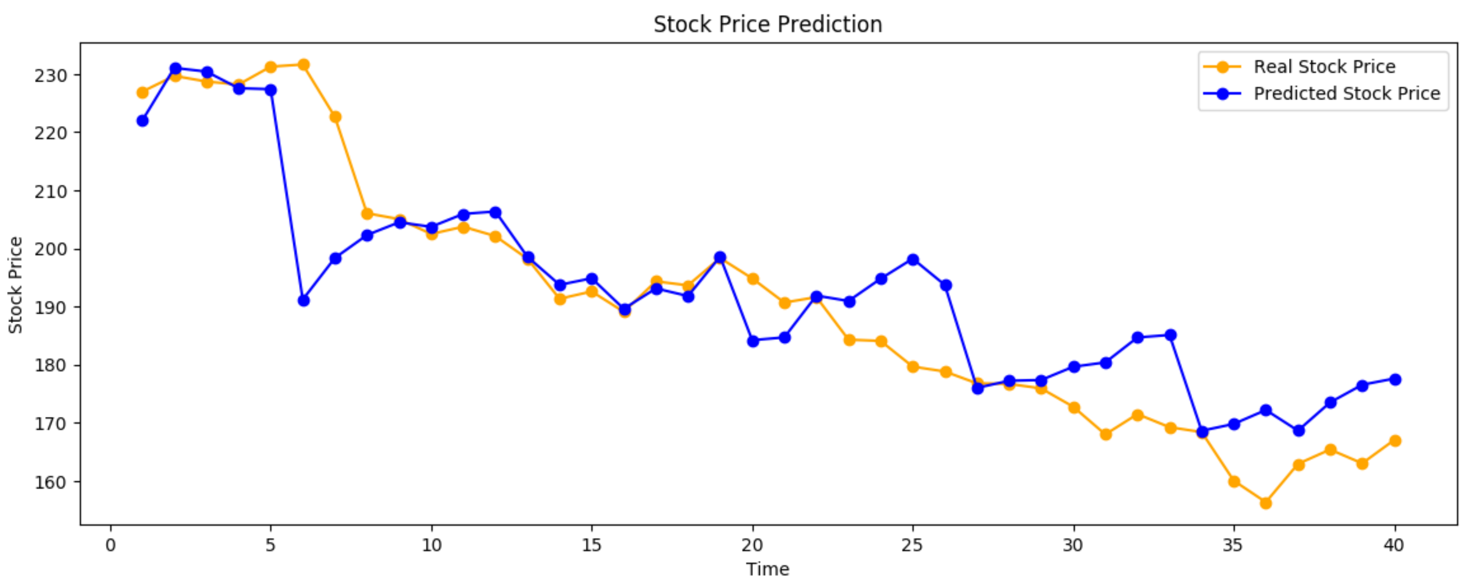

```
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 0, 1, 1, 1, 1, 1, 1]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 97.5% of the time.
```

Goldman Sachs (GS)

Loss comparison of models



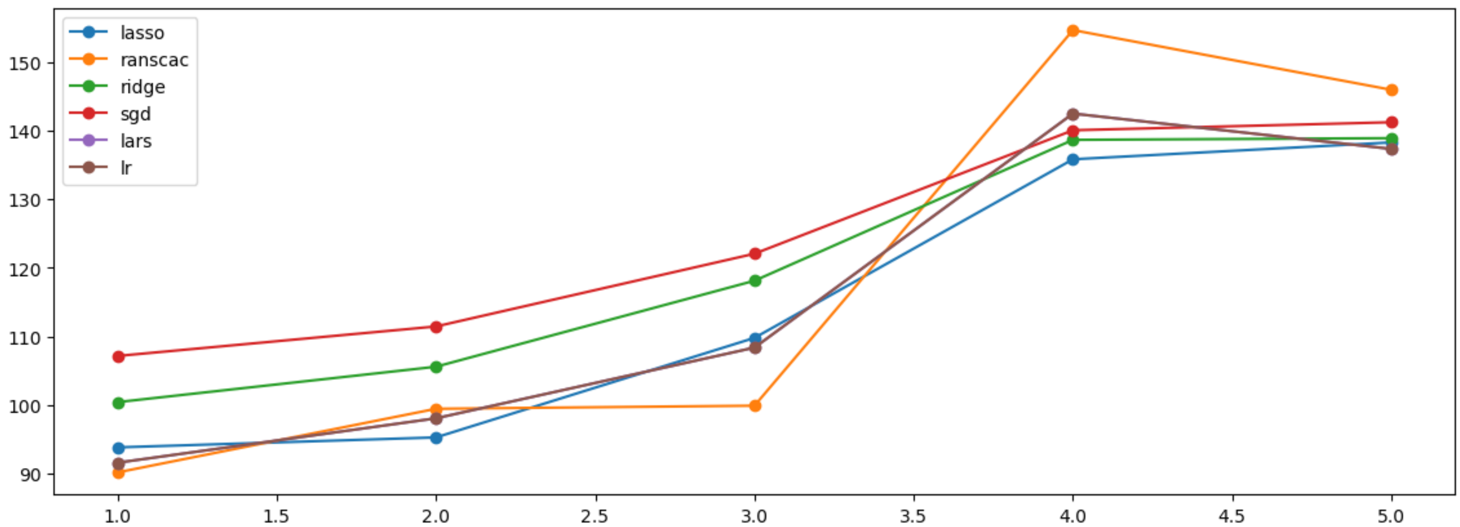
Actual vs Predicted Prices



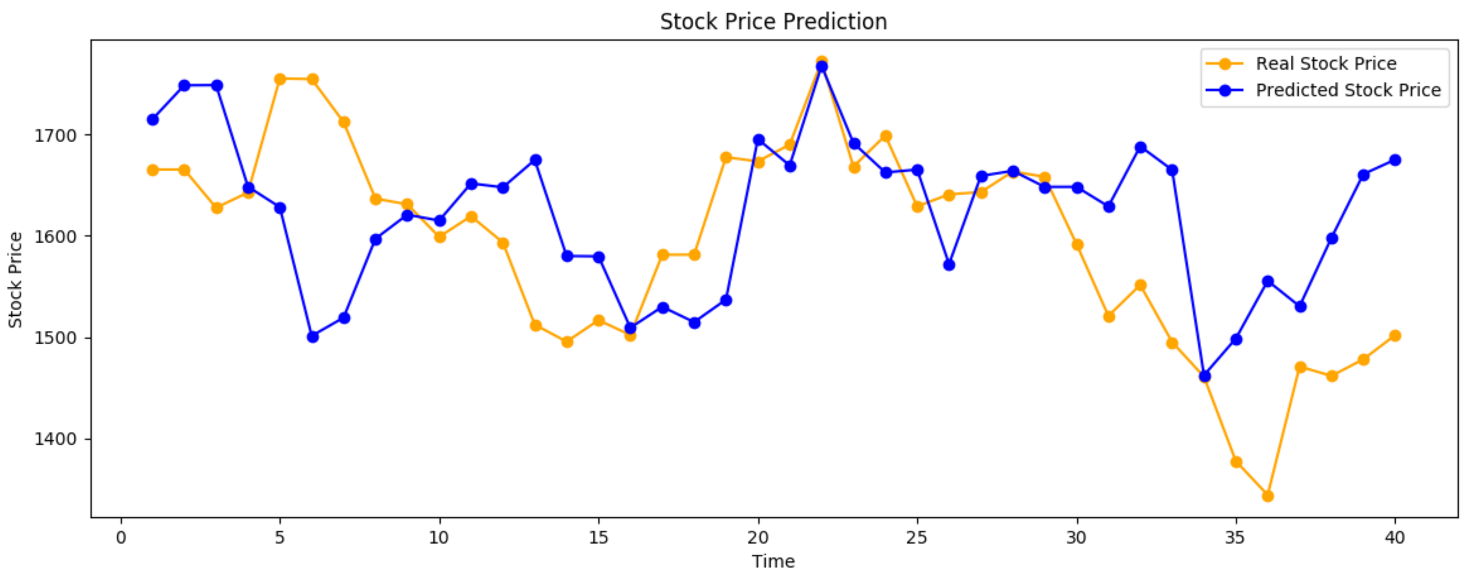

```
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 62.5% of the time.
```

Amazon (AMZN)

Loss comparison of models



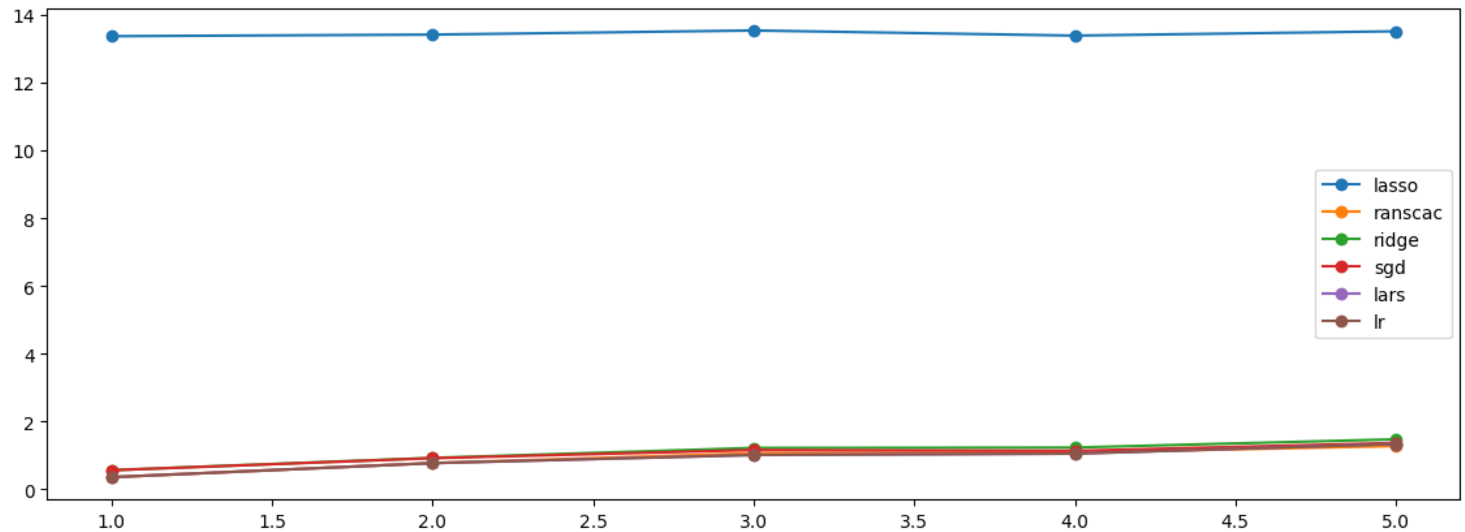
Actual vs Predicted Prices



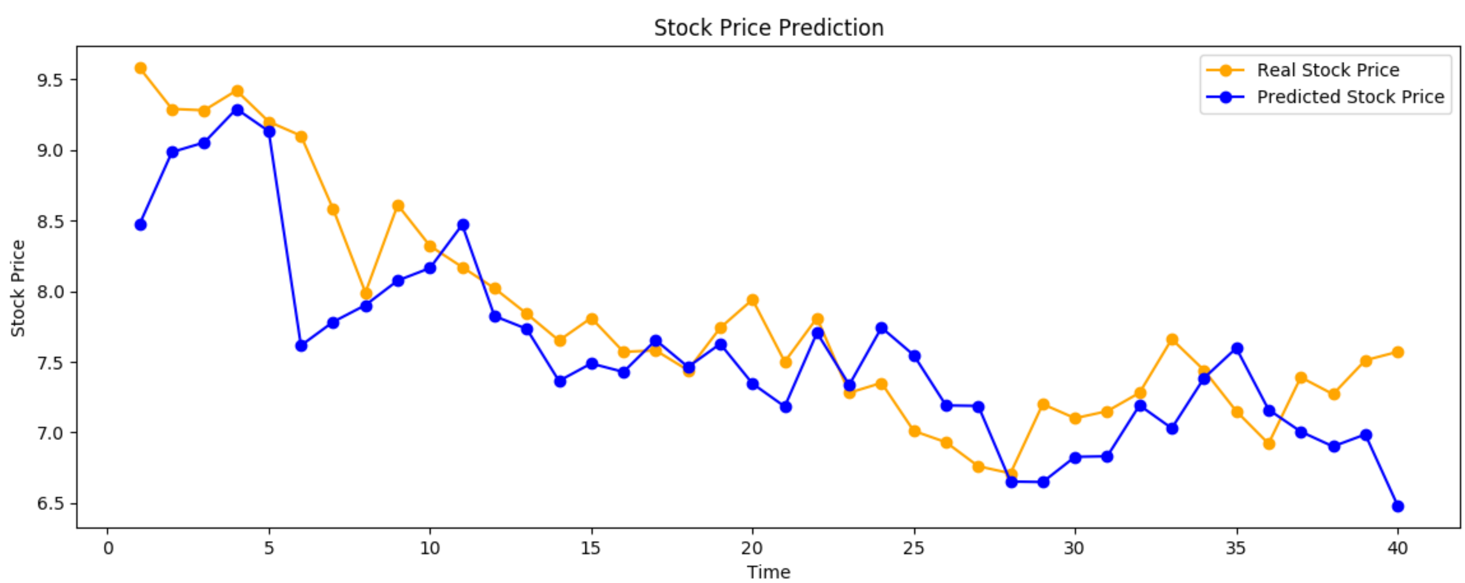

```
Prediction Direction [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
1, 0, 0, 0, 0, 1, 1, 1]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 67.5% of the time.
```

General Electric (GE)

Loss comparison of models



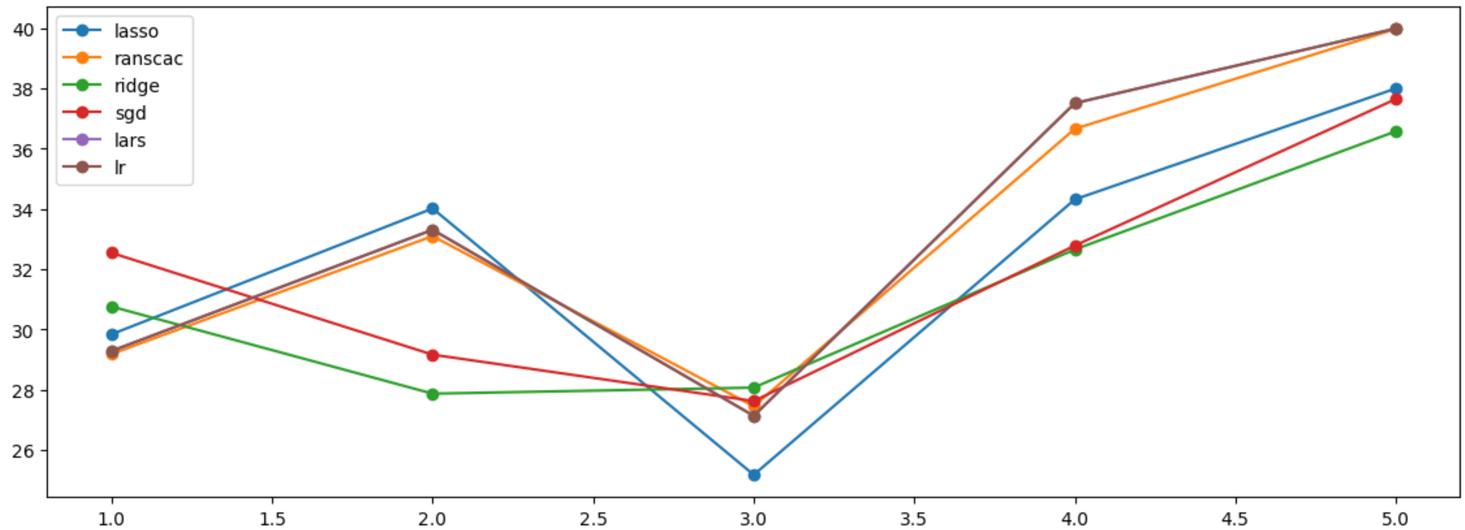
Actual vs Predicted Prices



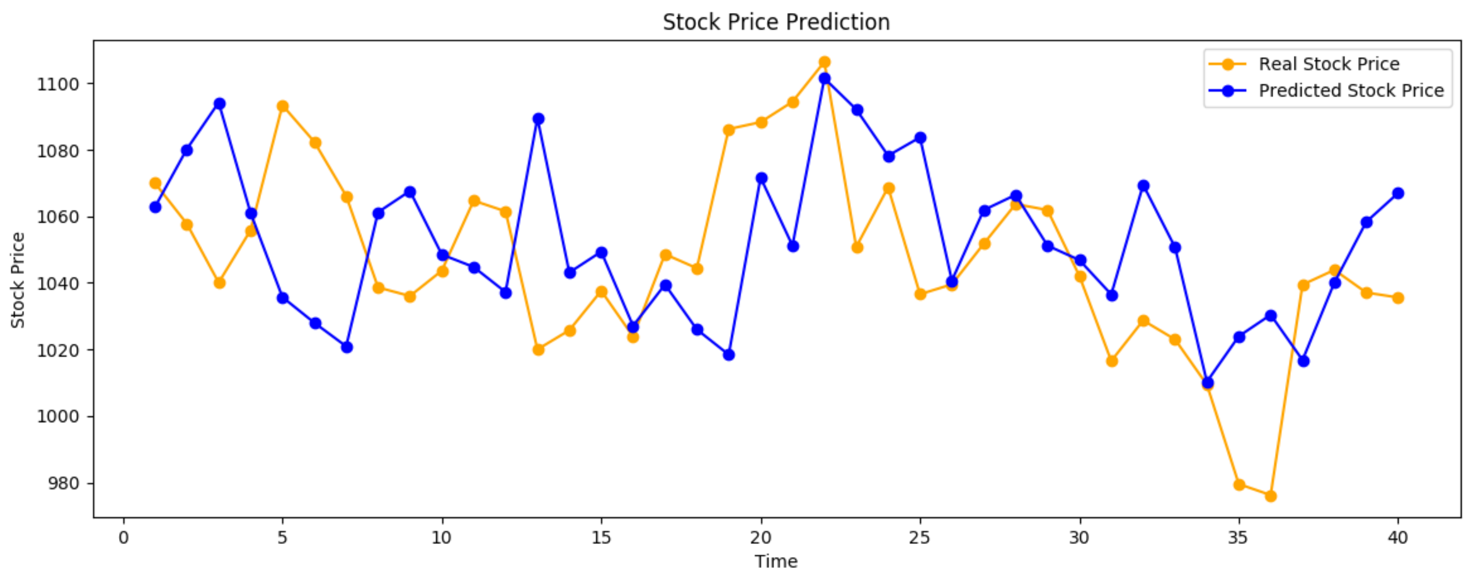

```
Prediction Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 100.0% of the time.
```

Google (GOOG)

Loss comparison of models



Actual vs Predicted Prices




```
Prediction Direction [1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1,
1, 0, 0, 0, 0, 0, 1, 1]

Real Direction [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1]

Predicted values matched the actual direction 57.49999999999999% of the time.
```

Interpretation of Recursive vs Direct :

We notice that the Direct strategy is performing slightly better than the Recursive Strategy because it does not allow the accumulation of errors.

EXTENSIONS TO THE PROJECT

- I would use ARIMA predictions as a feature in the Technical Analysis
- I would also derive feature through Autoencoders
- I would use a more sophisticated model with Generators - Discriminators in GANs
- Although, I did train some models over the GPU, I would like to train on GPUs for longer times with more sophisticated architectures.
- I would generate more visualizations that help us understand the data and results better.
- I would use the multi-step algorithms over neural networks and other non-linear models.
- I would like to spend more time on research and hyperparameter tuning

LESSONS I LEARNED

- How to delve into uncharted territory such as the Stock Market and come out having a good understanding of its complex price fluctuations and the whole range of factors that affect it in a very short period of time.

- How to quickly grasp new algorithms and implement them on own datasets.

- How to learn and use new tools such as Paperspace/Floydhub GPUs , new Machine Learning/ Deep Learning libraries, APIs, and other tools required to complete an extensive project.

- How solving such an interesting problem to solve is extremely engaging and satisfying.

- How data Collection and Preparation can be one of the most challenging tasks.

- How investigating to find out why the model is going terribly wrong can be extremely challenging, brain wrecking and really interesting.

- How it’s almost impossible to anticipate how solving little portions of the problem could take up hours together.

CONCLUSION

I learned so much more than I already knew about Deep Learning algorithms, Feature Selection methods, data preparation techniques and how to tie everything together.

I had a lot of fun learning about the Stock Market, its movement behaviors, factors that affect it and how to make sense of them.

REFERENCES

https://www.investopedia.com/trading/macd/

https://www.learndatasci.com/tutorials/python-finance-part-3-moving-average-trading-strategy/

https://machinelearningmastery.com/develop-word-embedding-model-predicting-movie-review-sentiment/

https://finviz.com/quote.ashx?t=aapl

https://github.com/borisbanushev/stockpredictionai

https://www.quantinsti.com/blog/quantitative-value-investing-strategy-python

http://digital-thinking.de/deep-learning-combining-numerical-and-text-features-in-deep-neural-networks/

https://medium.com/@Currie32/predicting-the-stock-market-with-the-news-and-deep-learning-7fc8f5f639bc

https://machinelearningmastery.com/multi-step-time-series-forecasting-with-machine-learning-models-for-household-electricity-consumption/

https://machinelearningmastery.com/multi-step-time-series-forecasting/

https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/

https://www.quantinsti.com/blog/asset-beta-market-beta-python

https://www.reuters.com/

https://machinelearningmastery.com/develop-word-embedding-model-predicting-movie-review-sentiment/