# STOCK MARKET ANALYSIS USING TIME SERIES ANALYSIS

Team:

Paul Sampson Ledala <pledala1@umbc.edu>
Sai Charita Thati <sthati1@umbc.edu>
Akanksha Narsin <anarsin1@umbc.edu>
Solomon Jayakar Durgam <sdurgam1@umbc.edu

## 1. Introduction:

Determining the stock trade information is a vital money-related subject that includes an assumption that the basic data freely accessible within the past has a few prescient connections to long-run stock returns. Stock market estimating involves revealing the showcase patterns, arranging speculation strategies, recognizing the finest time to buy the stocks and which stocks to buy. Time-series data analysis methods utilize irrefutable data as the premise for assessing future results. Time series information can be characterized as numerical information collected in a specific grouping over a period at customary intervals. The purpose is to discover in case there's an interface between the information collected so far and in what way does the information change.

In this project, we performed stock market forecasting using time series analysis with the help of recurrent neural network models(RNNs) and a classical model to compare and analyze the accuracy of various models. We implemented long short-term memory(LSTM), Gated recurrent units(GRU) and the classical model Autoregressive integrated moving average(ARIMA) to predict stock prices. We also aimed to perform risk analysis on each of the stocks we used in the project.

## 2. Description of the dataset

For this project, we collected the stock dataset of four companies from yahoo finance. YFinance is a free library available in python which offers a wide range of market data on stocks of many companies. Here, we used stock data of 4 companies namely, JP Morgan Chase & Co., Bank of America Corporation, Wells Fargo & Company and Morgan Stanley.

We collected past one year stock data of the above-mentioned companies where the dataset contains information about Open, High, Low, Close, Adj_Close, Volume, Company_name. The below table explains about the variables in the dataset.

| Variable | Description |
|---|---|

| | |
|---|---|
| Open | This refers to the price at which the price of stock has began on that day |
| High | This is the highest amount of price the stock has reached on that day |
| Low | This is the lowest amount of price the stock has reached on that day |
| Close | This is the price of the stock when the trading has ended |
| Adj Close | This value is the adjusted stock price after taking corporate actions into account such as stock splits, which affect the stock price.It is useful when examining historical returns. |
| Volume | It refers to the number of shares of security traded between its daily open and close. |
| Company_name | This is the name of the company which the stock price details belongs to. |

| Date | Open | High | Low | Close | Adj Close | Volume | company_name |
|---|---|---|---|---|---|---|---|
| **2022-04-26** | 84.290001 | 85.220001 | 82.279999 | 82.370003 | 81.668640 | 10062000 | Morgan Stanley |
| **2022-04-27** | 82.379997 | 83.669998 | 81.809998 | 82.209999 | 81.510002 | 9522900 | Morgan Stanley |
| **2022-04-28** | 82.699997 | 83.849998 | 81.610001 | 83.400002 | 83.400002 | 6974600 | Morgan Stanley |
| **2022-04-29** | 83.059998 | 83.739998 | 80.309998 | 80.589996 | 80.589996 | 8093900 | Morgan Stanley |
| **2022-05-02** | 80.589996 | 82.139999 | 80.029999 | 81.959999 | 81.959999 | 8737100 | Morgan Stanley |
| **2022-05-03** | 82.489998 | 84.809998 | 82.250000 | 83.750000 | 83.750000 | 9430500 | Morgan Stanley |
| **2022-05-04** | 84.070000 | 87.529999 | 83.940002 | 87.220001 | 87.220001 | 10495000 | Morgan Stanley |
| **2022-05-05** | 85.750000 | 86.160004 | 83.410004 | 84.940002 | 84.940002 | 8556500 | Morgan Stanley |
| **2022-05-06** | 84.139999 | 84.570000 | 82.309998 | 84.230003 | 84.230003 | 7331700 | Morgan Stanley |
| **2022-05-09** | 82.540001 | 83.239998 | 80.720001 | 81.150002 | 81.150002 | 11158800 | Morgan Stanley |

Sample Data

## 3. Exploratory Data Analysis

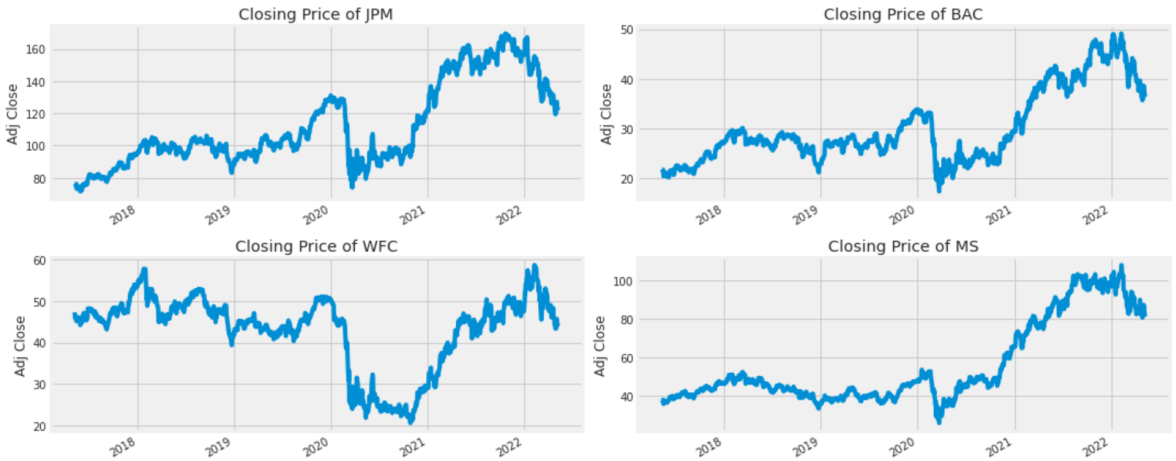
We have delved into the statistical analysis of each stock in the stock list. We have used the summary statistics which consists of count, mean standard deviation, minimum, maximum and the 3 quartile range units. This gives us an initial understanding of the performance of the stocks and what kind of returns/risk we can expect from it.

Summary Stats of JPM stock

| | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| **count** | 1259.000000 | 1259.000000 | 1259.000000 | 1259.000000 | 1259.000000 | 1.259000e+03 |
| **mean** | 119.546966 | 120.717117 | 118.381184 | 119.524638 | 111.954364 | 1.466007e+07 |
| **std** | 23.387696 | 23.510901 | 23.237334 | 23.368868 | 25.867376 | 6.635330e+06 |
| **min** | 81.559998 | 82.989998 | 76.910004 | 79.029999 | 71.521790 | 3.324300e+06 |
| **25%** | 101.710003 | 102.875000 | 100.415001 | 101.549999 | 93.983807 | 1.032390e+07 |
| **50%** | 112.910004 | 114.120003 | 112.150002 | 113.029999 | 101.558365 | 1.305440e+07 |
| **75%** | 137.225006 | 138.190002 | 136.184998 | 137.220001 | 129.398453 | 1.674555e+07 |
| **max** | 172.710007 | 172.960007 | 170.539993 | 171.779999 | 169.500061 | 5.441880e+07 |

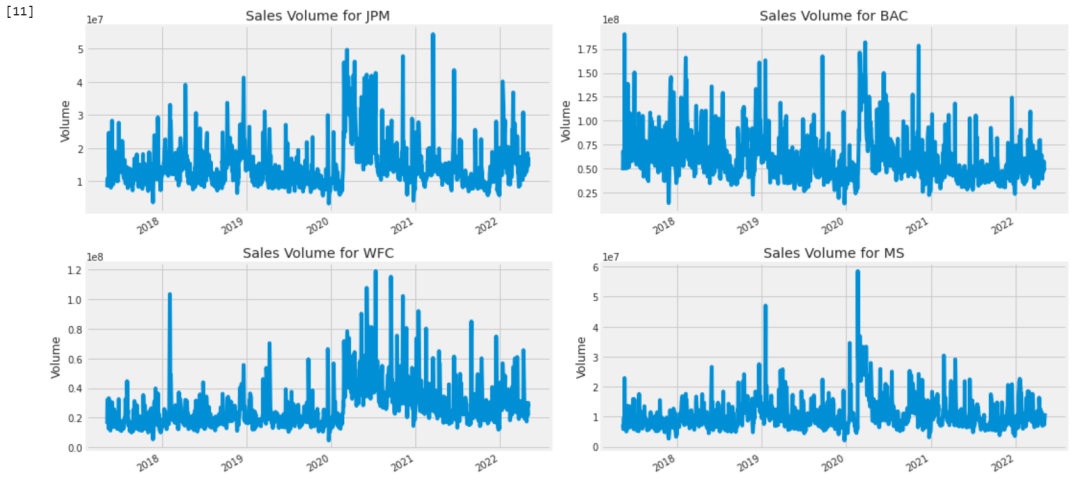
Summary Stats of JPM Stock

We have used some of the techniques of data visualization to view the closing prices stock graph. This gives us a picture of the long-term trend of the stock over a time period that we can choose. In our project that time period is 5 years from the current date.



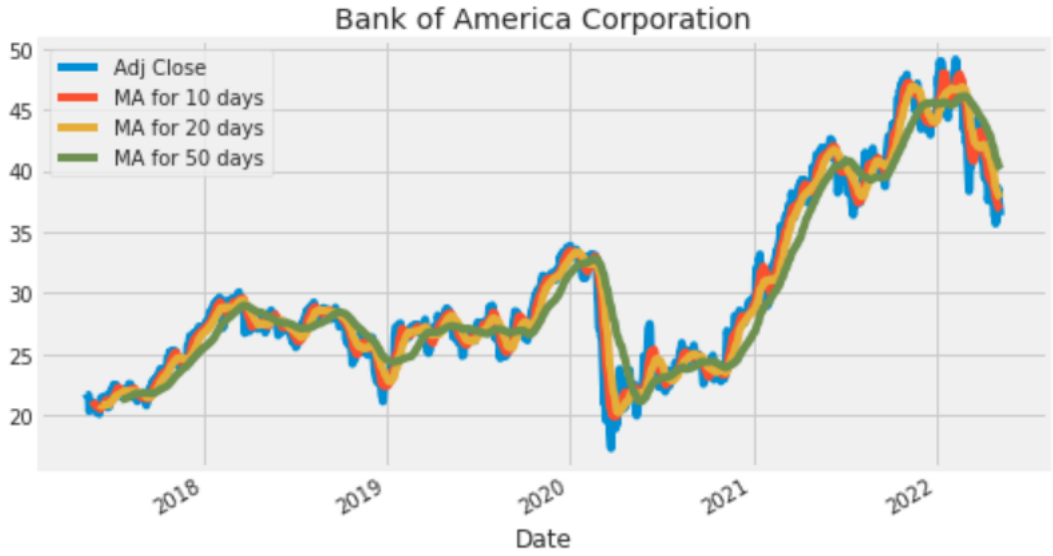

Closing Price trends of all the bank stocks

To get the sense of volatility of the stock, we've used the volumes of stock sales to get an understanding of how much the stocks are bought and sold and the interest generated by these stocks. We can also try to find correlations between the sales volume and price variations and risk (as measured by standard deviation).



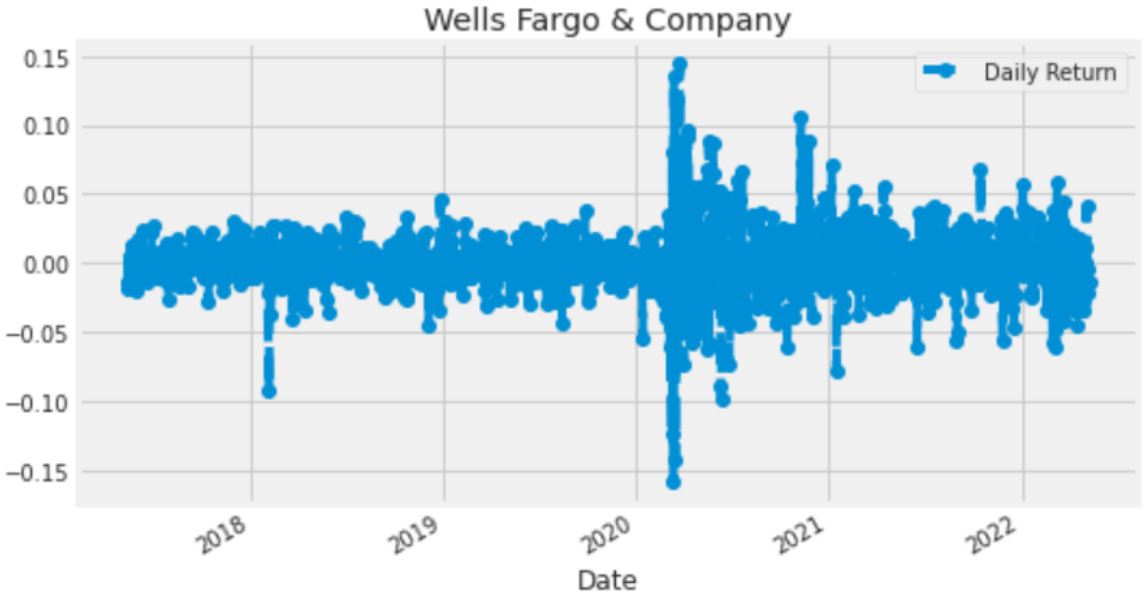

Daily Sales Trades of all the bank stocks

Moving Average of an individual stock gives the smoothed trend lines of a company's stock performance. We can clearly see whether there is an upward or downward trend. This helps the user in gauging the long-term performance of each stock removing the local minimums and maximums. We can choose to calculate the moving average for any number of consecutive days. We took the samples for 10, 20 and 50 days.

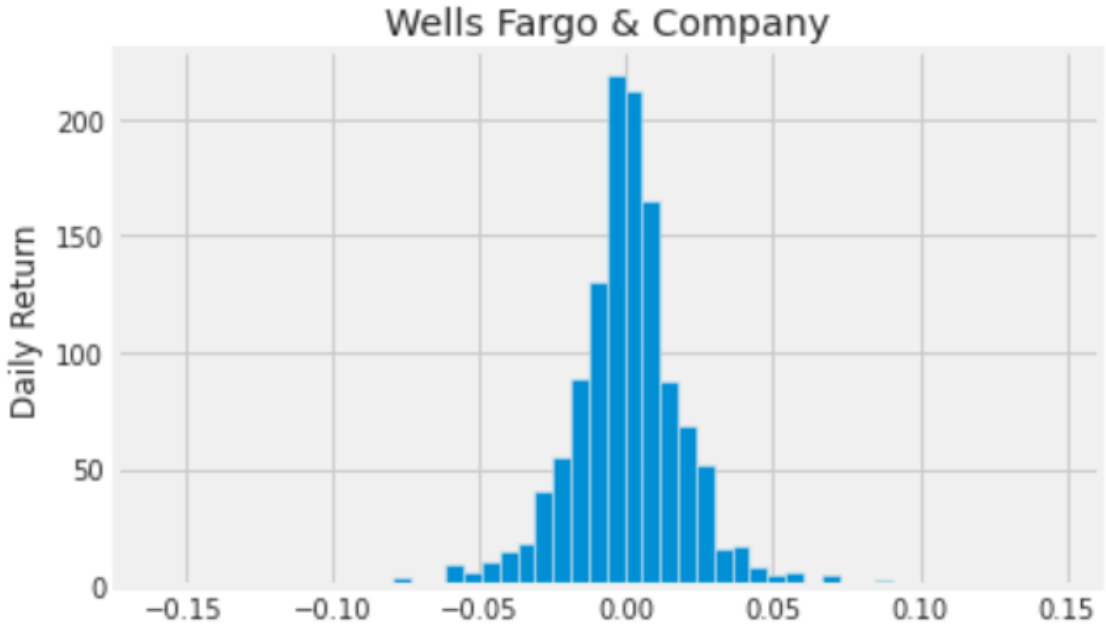



Moving Averages of BAC

The next graph is self-explanatory giving the daily returns of an individual stock for a set time period. This is especially useful in giving an initial idea of how much returns we can expect from a single day trading session. We can see that a company can have a wide range of returns in a small interval of time when the market is highly volatile (or crashes). We can call this time a stock market bubble followed by the crash. Aggregation of returns follows a normal distribution as shown.

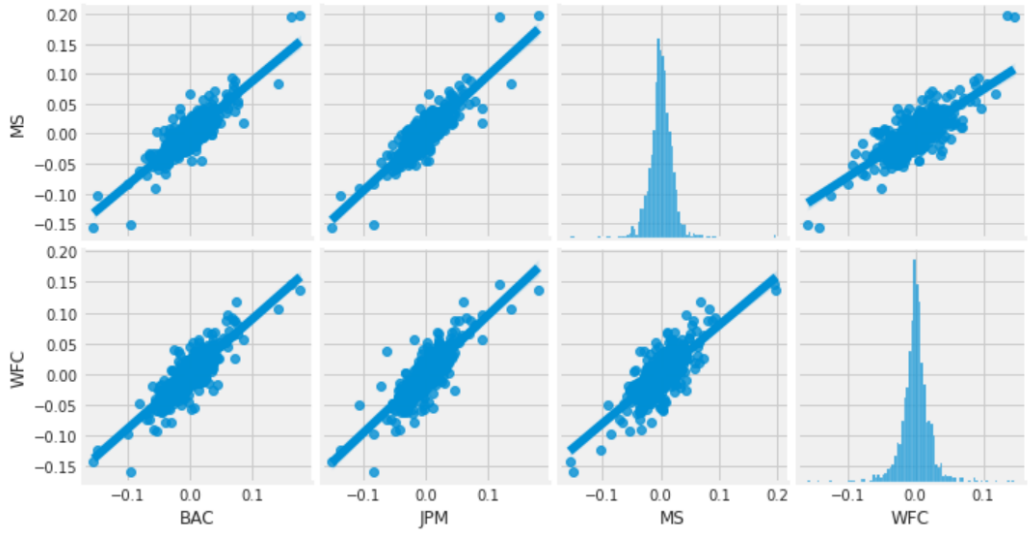



Daily Return Percentage of WFC

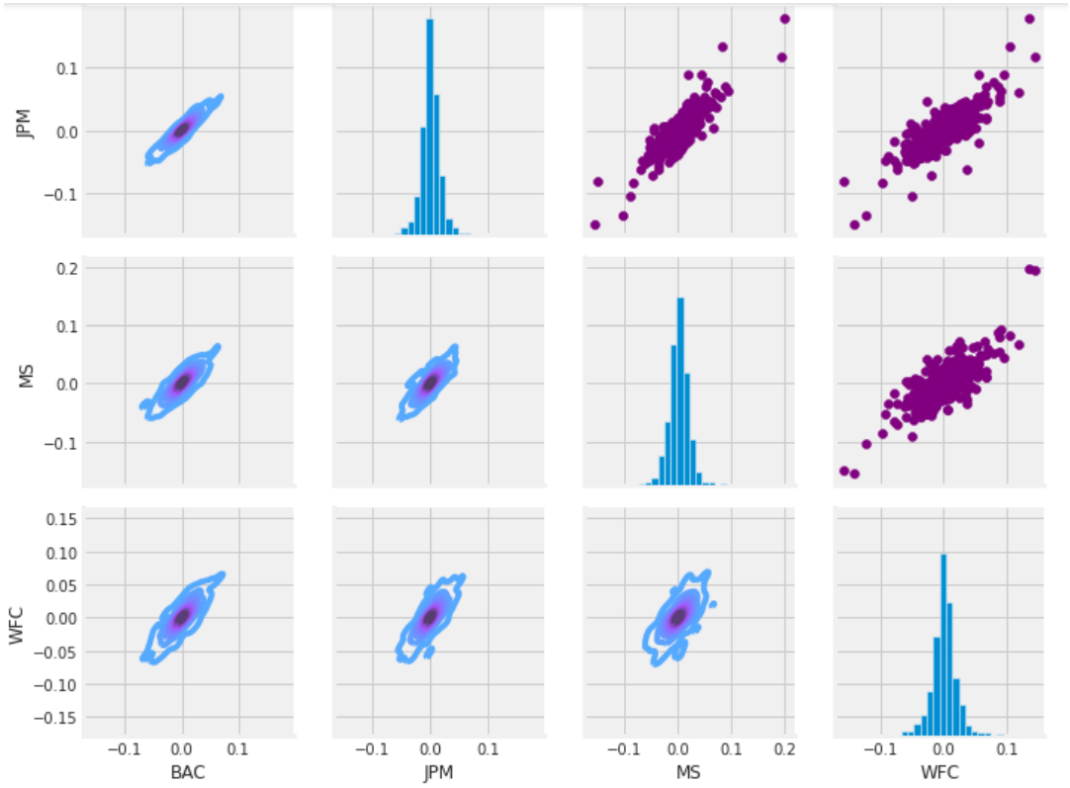



Daily Return Aggregation of WFC

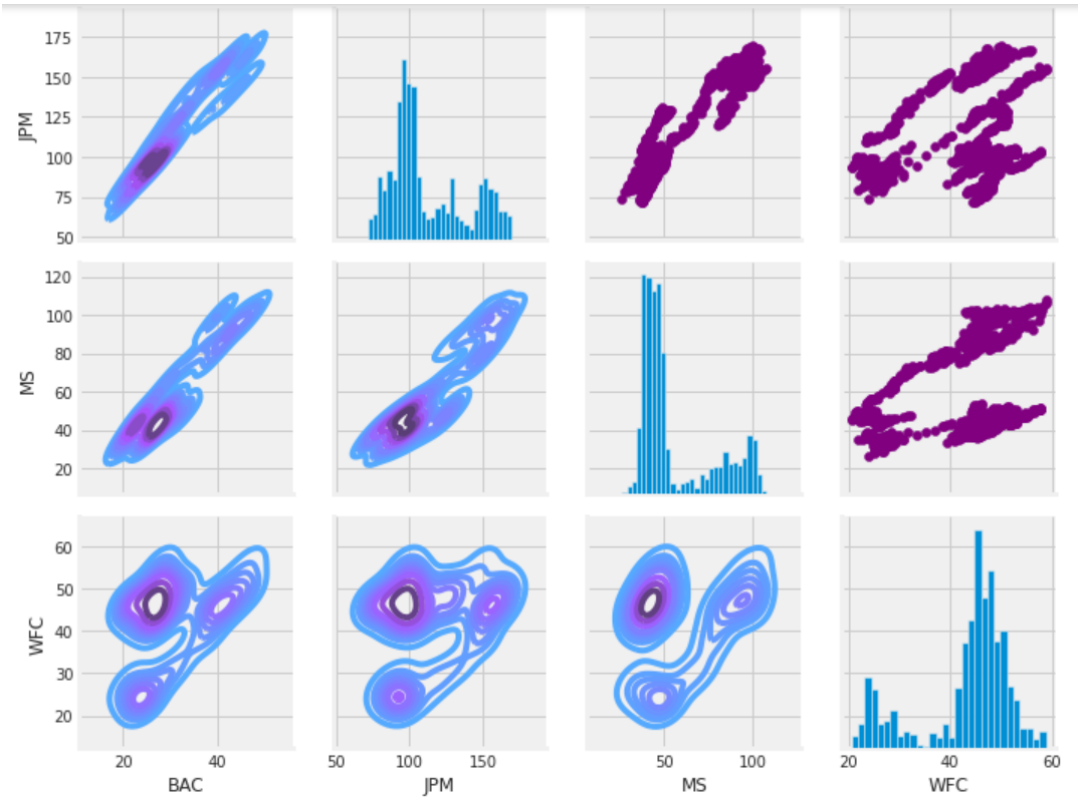
Correlation between the stocks can be measured and visualized by a number of techniques including pair-plots, scatterplot kdeplot, heatmaps, etc. We have implemented these using all the banking stocks we extracted from yfinance. These correlations usually tell us which stocks are closely correlated with each other to help in the analysis of returns of each stock. If one of the highly correlated stocks falls, we can generally expect the other to dive as well. That can be good information for the model to run and get its information regarding the trend.



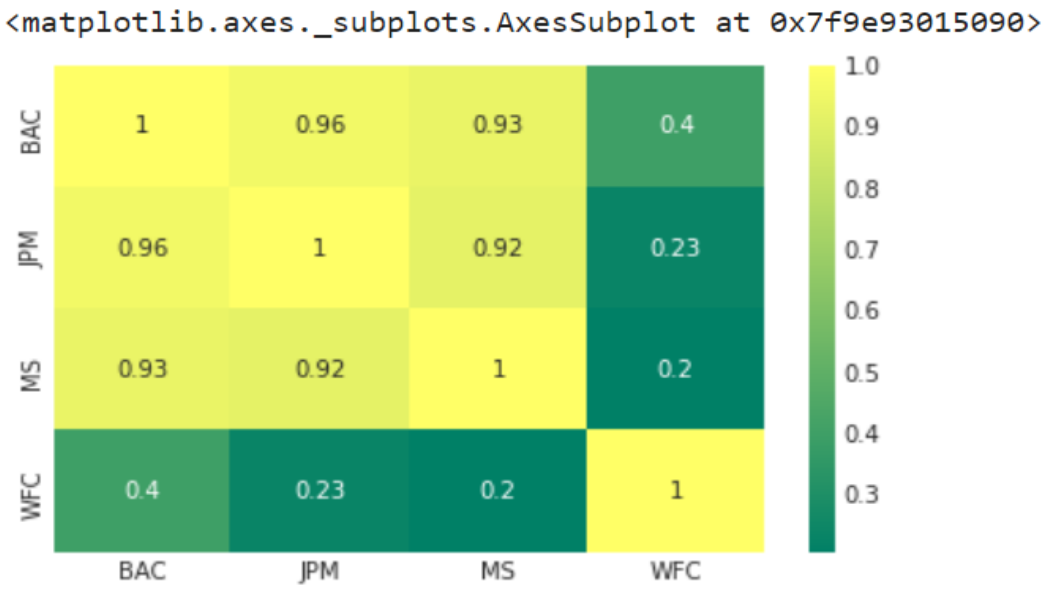

Pair-plots between all the stocks




Scatter plots and KDE plot based on the daily percentage returns

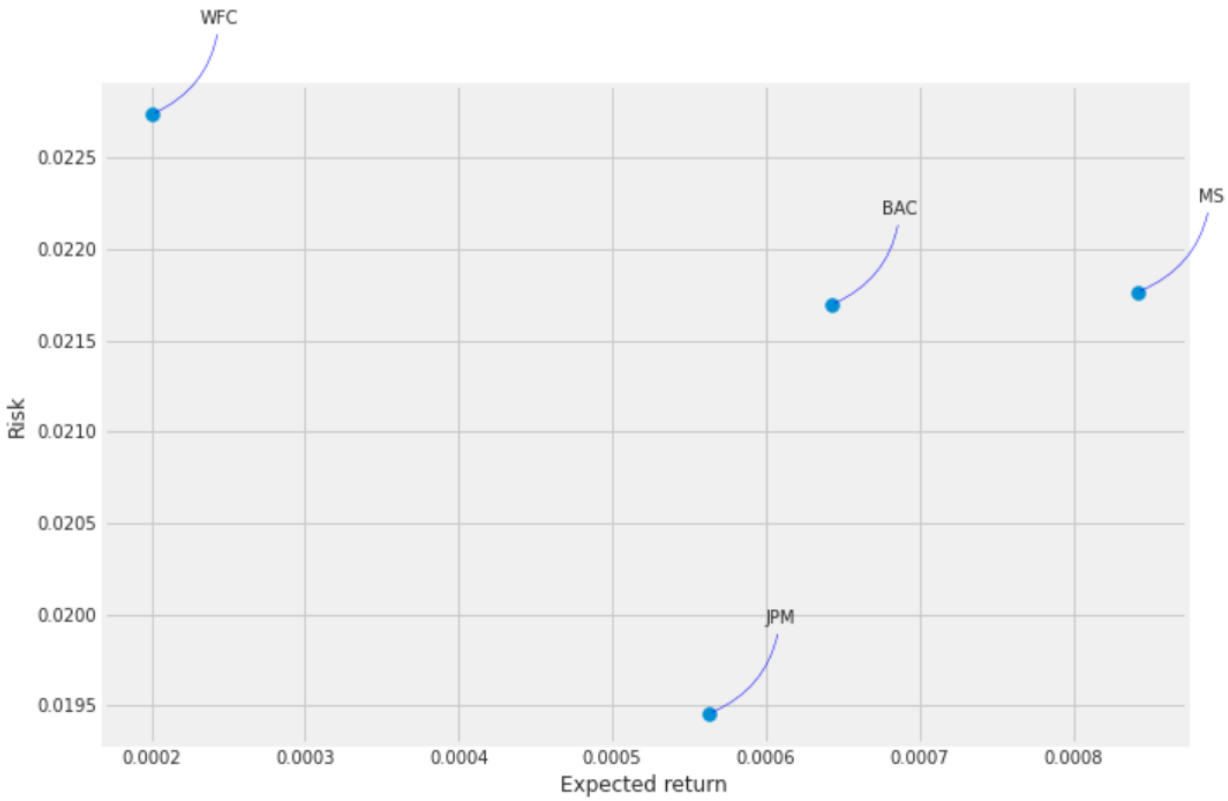



Scatter plots and KDE plot based on the daily closing price




Heatmap of correlation between the banking stocks

Risk is calculated as the standard deviation of the individual stock from its expected return. Expected return gives the multiple of increase/decrease of the individual stock. This can be shown taken in a graph plotting all the stocks. Such a comparison gives us an insight into the most investible company among those selected.




Risk vs Return of banking stocks

## 4. Modeling and Results:

Our problem statement was to build a model that is capable of time-series forecasting the stock returns of an individual stock. These models are only built on individual stocks and have to be run for each of the individual stocks that needs to be analyzed. The output returned is the expected return. The input taken is the closing price of the stock.

As this is a time series of historical data, we took about a 5 year period of closing prices for the input to the model. The test-train split was 95%-5%. This is due to the fact that the number of days of data that can be used for training is massie, but we only need to prove the model on a relatively smaller amount of test data. Such a model can be used for predicting only a few days into the future- which is ideal for us.

The evaluation of the models needed to have a common metric for them to compare the models. These metrics need to punish huge deviations of expected returns from the actual returns. Thus we considered Mean Squared Error, Mean Absolute Error and the Root Mean Squared Error. These are ideal metrics to calculate the deviations from the predicted value, by penalizing large deviations.

# Modeling and Results

**Model Input**: Closing price (individual stock)

**Model Output**: Prediction of closing price

**Test-Train Split**: 95 - 5

**Evaluation**:
Mean Squared error (MSE)
Mean Absolute error (MAE)
Root Mean Squared Error (RMSE)

The three models which we considered are:

- LSTM: It is an abbreviation for Long short-term memory. It is a deep learning model. It is a special kind of RNN capable of learning long term sequences by remembering. We used this model to give us the time series of the stocks over a short term-long term trend.

- GRU: Gated Recurrent Units are a gating mechanism in recurrent neural networks. It is similar to LSTM. GRU does not possess internal memory. The GRU model is to calculate the long-term trend lines. So we wanted to use this for our stocks to take the long-term consideration of the returns.

- ARIMA: Autoregressive Integrated Moving Average. It predicts a given time series based on its own past values. It can be used for any nonseasonal series of numbers that exhibits patterns and is not a series of random events. For example, sales data from a clothing store would be a time series because it was collected over a period of time. One Key characteristic is the data collected over a series of constant regular intervals. For these reasons, we used this as the ideal classical method to predict the stock returns.

**ARIMA:**

ARIMA Model Results

| | | | |
|---|---|---|---|
| Dep. Variable: | D.Close | No. Observations: | 2340 |
| Model: | ARIMA(2, 1, 0) | Log Likelihood | 6230.293 |
| Method: | css-mle | S.D. of innovations | 0.017 |
| Date: | Tue, 10 May 2022 | AIC | -12452.587 |
| Time: | 00:18:52 | BIC | -12429.555 |
| Sample: | 1 | HQIC | -12444.197 |

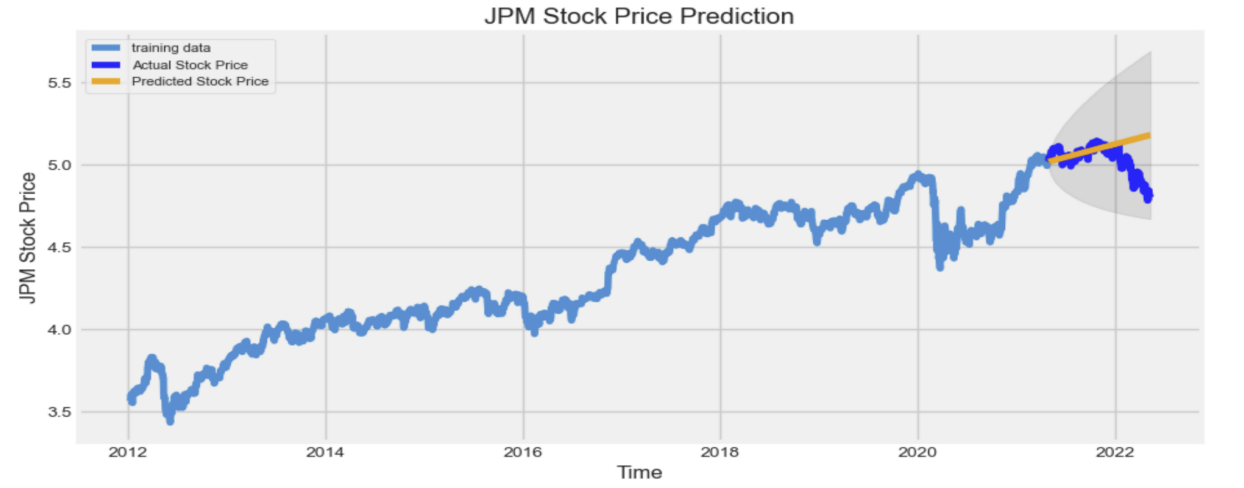
| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 0.0006 | 0.000 | 1.868 | 0.062 | -3.06e-05 | 0.001 |
| ar.L1.D.Close | -0.1126 | 0.021 | -5.459 | 0.000 | -0.153 | -0.072 |
| ar.L2.D.Close | 0.0638 | 0.021 | 3.095 | 0.002 | 0.023 | 0.104 |

Roots

| | Real | Imaginary | Modulus | Frequency |
|---|---|---|---|---|
| AR.1 | -3.1731 | +0.0000j | 3.1731 | 0.5000 |
| AR.2 | 4.9367 | +0.0000j | 4.9367 | 0.0000 |

In the ARIMA model, we need to specify three parameters, which are p, d and q. Here p is for number of lagged observations in the mode, d is for degree of differencing(in order to make the data stationary) and q is for size of the moving average window. So there are two ways of specifying these parameters. One is by trial and error method and the other is auto arima where this returns a fitted model by giving us the best p, d, q values. So here we used auto arima and it gave us the values as p: 2, d: 1, q: 0. Here 0 specifies that we don't use q in the model, this can be seen in the above image.

Arima Model Result Statistics




Arima Predictions

The results given by ARIMA are evaluated as follows.

MSE: 0.0203
MAE: 0.0938
RMSE: 0.1427

**LSTM:**

We can see that in the LSTM model, we have two LSTM layers and two dense layers. The input shape is taken as 60*128, the output prediction is only a single valued output from the final dense layer. The number of trainable parameters is over 117k.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 lstm_2 (LSTM)                (None, 60, 128)           66560

 lstm_3 (LSTM)                (None, 64)                49408

 dense_2 (Dense)              (None, 25)                1625

 dense_3 (Dense)              (None, 1)                 26

=================================================================
Total params: 117,619
Trainable params: 117,619
Non-trainable params: 0
_________________________________________________________________
```

LSTM Model Layers


LSTM Prediction

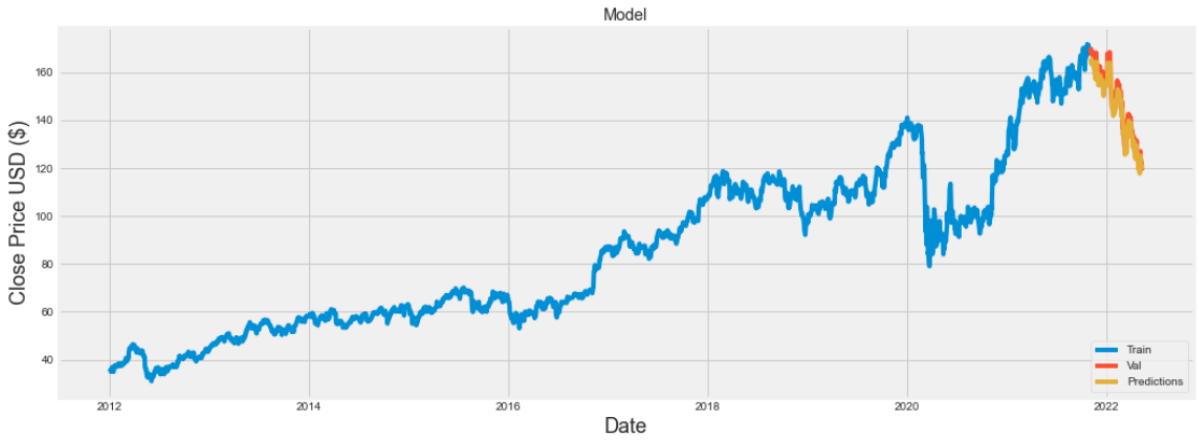
We can see through the predicted values graph that LST does a good job in getting the direction of the stocks right. However we see from the result, that the drop in a stock price due to market conditions, has a detrimental effect on the performance of the model; as the RMSE gets higher with every sudden drop in the price.

The results given by LSTM are evaluated as follows.

MSE: 8.260
MAE: 2.206
RMSE: 2.874

GRU:

We also took GRU as our second deep neural network to predict the stock returns. GRU was a useful control method which sees the long term patterns and predicts using them. In this graph we can see that the predicted value is pretty close to the LSTM model. We can only differentiate the two models by the evaluation metrics which give us a clear picture about the better prediction model.




GRU Prediction

The results given by GRU are evaluated as follows.

MSE: 15.090
MAE: 3.307
RMSE: 3.884

This gives a clear picture that the LSTM prediction has a slight edge over GRU for our problem.

Our hypothesis was that the neural networks models LSTM and GRU will provide more accurate results when compared to classical model ARIMA. But suprisingly, according to the root mean squared error we calculated for all the models considering the actual closing price and the predicted closing price, the LSTM and GRU models showed high error when compared with the traditional ARIMA model.

Our ARIMA model returned a root mean square error of 0.142 whereas both the LSTM and GRU models returned an error of 2.87 and 3.88 respectively. Now this raises a new question as to why would a classical model be more accurate than a neural network model? Here our input to the models is closing price and we predict closing price as the output. When looking at the price prediction graphs, we could observe that there are deviations in the prediction from the actual price when there is a steep drop. There are a few instances, there is a sudden drop or raise in the actual closing price. This leads to huge deviation from the predicted trend line. This leads to the huge difference in the errors between the models.

## 5. Discussion and future work

In this project, we learnt the different types of models used in time-series analysis both in the classical statistical models and the newer deep-learning models. We learnt different architectures which retain the long-term memory (LSTM) while other architectures have a mechanism to only predict using the short term factors (RNN). We also learnt about GRU's where we have the reset gate, which gives the control of how much to forget in terms of short term data while persisting the long-term memory.

We were intrigued by the exploratory analysis that could be done even before implementing the model. There are several data visualizations which we found to be very useful in the analysis of stocks, their returns, volumes traded, correlation and return vs risk.

We also found implementing the deep learning models to satisfy our theoretical knowledge. By practically checking the layers of the model and the trainable parameters, we understood why the deep learning models make such accurate predictions based on strong foundations of mathematical theory.

We initially wanted to use a classical model like ARIMA or SARIMA to see how better a deep learning model would perform over it. We found that ARIMA was a straight line prediction and not a typical stock trend line. However, this gave better results than the closely resembling trend curve predictions of LSTM and GRU.

In future work, we would like to further investigate on why the classical model is predicting more accurately when compared to the neural network models. We intend on identifying the factors contributing to these predictions and further work on increasing the

accuracy of the neural network models including (but not limited to) feeding it market data, or data of a correlated stock.

## 6. References

1. Y. Gao, R. Wang, and E. Zhou, “Stock prediction based on optimized LSTM and GRU models,” *Scientific Programming*, 29-Sep-2021. [Online]. Available: https://www.hindawi.com/journals/sp/2021/4055281/#abstract.[Accessed: 20-Mar-2022].

2. Fischer, T. and Krauss, C., 2022. *Deep learning with long short-term memory networks for financial market predictions*. [online] Available at: <https://www.sciencedirect.com/science/article/abs/pii/S0377221717310652?via%3Dihub> [Accessed 20 March 2022].

3. https://pypi.org/project/yfinance/

4.https://www.analyticsvidhya.com/blog/2021/07/stock-market-forecasting-using-time-series-analysis-with-arima-model/

## 7. Description of the code:

We intend to include a jupyter notebook- ‘Stock_prediction.ipynb’ along with the current document which is the source code of our program and shows the various functions that we have put together and the models that we have built to perform the stock market predictions. The comments can be referred to, to understand how the code functions.