Introduction

A **stock** (also known as equity) is a security that represents the ownership of a fraction of a corporation. This entitles the owner of the stock to a proportion of the corporation's assets and profits equal to how much stock they own. Units of stock are called "shares." The stock market refers to the collection of markets and exchanges where regular activities of buying, selling, and issuance of shares of publicly-held companies take place.

A **stock market prediction** is an attempt to forecast the future trend of an individual stock, a particular sector or the market, or the market as a whole. These forecasts generally use fundamental analysis of a company or economy, or technical analysis of charts, or a combination of the two.

In the past few years a lot of models based on deep learning have been gaining popularity for predicting volatility of the stock market prices. In this project, we try to implement several **classical deep learning models** like RNNs, CNNs and their many variants by implementing them using Tensorflow framework in Python. Then we perform a comparative study of all the aforementioned models with our aim being to find out which model is the best fit for volatility prediction.

In an attempt to further increase the accuracy of our classical models we try to incorporate **sentiment analysis** into our models. Another motivation

to use sentiment analysis is that, according to the Efficient Market Hypothesis (EMH), stock volatility cannot be predicted by historical prices alone in the long term as investors are guided by fear and greed.
Thus begins the second part of our project. For a given company's stock we scrape news articles for it on each day and run sentiment analysis on the articles to get a sentiment score which is an input to the model. We try to find out whether adding sentiment analysis gives us a reasonable increase in the accuracy.

Brief Discussion & Shortcomings of Related Work

It is widely acknowledged that stock price prediction is a job full of challenges due to the highly unpredictable existence of financial markets. Many market participants or analysts, however, attempt to predict stock prices using different mathematical, econometric or even neural network models in order to make money or understand the nature of the equity market.

In the paper *Neural networks for stock price prediction (2018)* by Ren-Jie Han, Yu-Long Zhou, Yue-Gang Song , five neural network models, namely, back propagation (BP) neural network, radial base function (RBF) neural network, general regression neural network (GRNN), support vector machine regression (SVMR), least square support vector machine regression, are surveyed and compared with predictive capacity (LS-SVMR). They conclude that the BP neural network reliably and robustly outperforms the other four models by following mean square error and average absolute percentage error as parameters.

Another paper *Stock Prices Prediction using Deep Learning Models (2019)* by Jialin Liu, Fei Chao, Yu-Chen Lin, and Chih-Min Lin, talks about the challenges of using deep learning models for predicting stock prices. This is a challenge, since there is a lot of noise and confusion in stock price-related details. In order to denoise the data, this work utilizes sparse autoencoders with one-dimensional (1-D) residual convolutionary networks. In order to forecast the stock price, long-short term memory (LSTM) is then used. Prices, indexes and macroeconomic factors in the past are the attributes used to estimate the expense of the next day.

Stock Trend Prediction using News Sentiment Analysis by Kalyani Joshi, Prof. Bharathi H.N., Prof. Jyothi Rao focuses on the famous theory of stock prediction i.e the Efficient Market Hypothesis. This project is about taking non-quantifiable data such as a company's financial news articles and forecasting the future market trend with the classification of news sentiment. This is an attempt to research the relationship between news and stock trends, believing that news stories have an impact on the stock market. They developed three distinct classification models to explain this which indicate that the polarity of news articles is positive or negative. Observations show that in all forms of testing, RF and SVM perform well. Naive Bayes performs well, but not in contrast to the other two. The accuracy of the forecast model is over 80 percent and 50 percent accuracy relative to news random labeling; the model has improved accuracy by 30 percent.

Sentiment Analysis of Twitter Data for Predicting Stock Market Movements (2016) by Venkata Sasank Pagolu highlights that social media now-a-days are a true reflection of public perception and opinion on current affairs. An fascinating area of research has been stock market forecasting based on public opinions shared on Twitter. Previous studies have shown that Twitter's aggregate public mood could well be associated with the Dow Jones Industrial Average Index (DJIA).Two separate textual representations, Word2vec and N-gram, have been used in the present paper to examine public feelings in tweets.

Problem addressed in the BTP

We try to contrast and compare the effectiveness of Classical Deep Learning based models to predict the stock prices for stocks of a specific company.Here we use the following models and their common variations in our thesis.

1. Vanilla RNN
2. Long Short Term Memory(LSTM) Model
3. Gated Recurrent Unit(GRU)
4. Convolutional Neural networks(CNN)

We apply relative accuracy as the measure to differentiate among the above models and observe that the accuracies were homogenous and more importantly insignificant .

One of the core reasons for the poor results in classical deep learning methods is because the investors don't behave in a rational and logical manner but instead tend to be dominated by greed and fear as proven in Efficient market hypothesis. According to the EMH, stocks always trade at their fair value on exchanges, making it impossible for investors to purchase undervalued stocks or sell stocks for inflated prices.

Therefore, it should be impossible to outperform the overall market through expert stock selection or market timing, and the only way an investor can obtain higher returns is by purchasing riskier investments.Therefore, in order to capture the investor's mood, we are building a RNN(Recurrent neural network) which uses sentiment analysis along with historical stock price data to observe if it provides any improvement over the classical approach.

Classical Model

Approach

1. **Historical Data Collection**: We use Yahoo Finance for collecting the historical stock price data which will be used in all of the models as a means to record and observe the previous trends.

2. **Historical Stock Price Data Processing:** Normalization: The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

3. **Model Creation based on only Historical Stock Data**
    - **Vanilla RNN**: A Recurrent Neural Network is a type of neural network that contains loops, allowing information to be stored within the network.
    - **Long Short Term Memory(LSTM) Model**: LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior.
    - **Bi-LSTM**: Train two LSTMs instead of one LSTMs on the input sequence. Can be trained using all available input info in the past and future of a particular time-step.
    - **Sequence to Sequence LSTM**: Trained to map an input sequence to an output sequence not necessarily of the same length.
    - **Gated Recurrent Unit (GRU)**: GRU is like a long short-term memory (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate. GRUs have been shown to exhibit better performance on certain smaller and less frequent datasets.
    - **Bi-GRU**
    - **Sequence to Sequence GRU**
    - **Convolutional Neural Network(CNN)**: CNNs are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and

descriptive tasks, often using machine vison that includes image and video recognition, along with recommender systems and natural language processing

Data sets and experimental setup needed

Historical Stock Prices:- Yahoo Finance
Google: https://finance.yahoo.com/quote/GOOG/history?p=GOOG

Software Tools:-

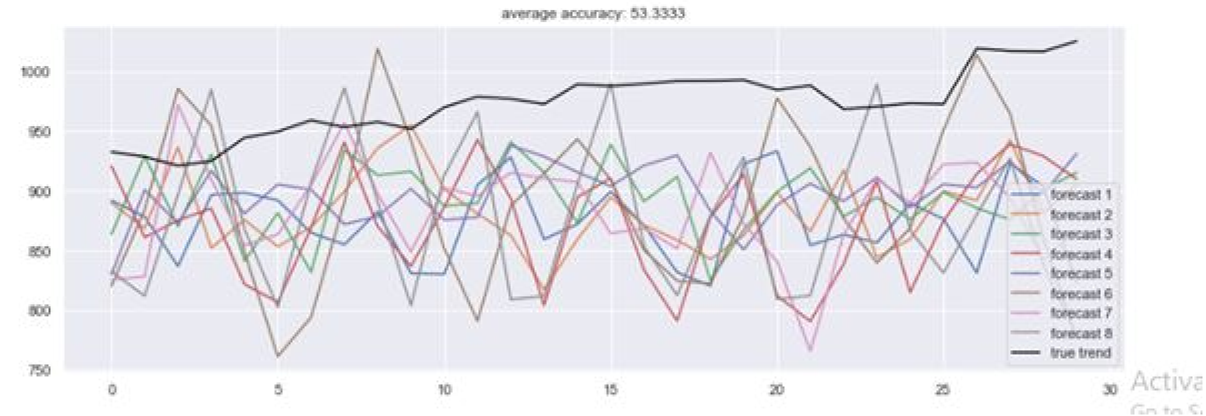
- PyCharm IDE
- Jupyter Notebook
- Anaconda Terminal
- Ubuntu and Windows 10

Experimental Steps

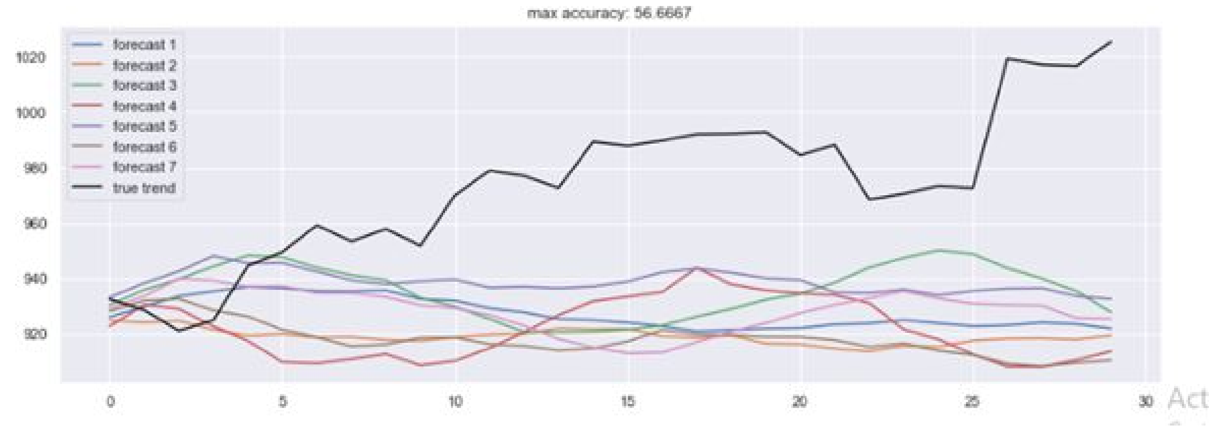
- Model Construction
  Parameters:-
  num_layers = 1,size_layer = 128, timestamp = 5, epoch = 10, dropout_rate = 0.8,future_day = test_size, learning_rate = 0.01, test_size=30
- Model Training
  - The 4the column of the dataset i.e. “Close” indicating the closing stock price data for each specific date used.
  - The 222 data points are divided into batch size of 5 stored in the variable batch_X and sent as a placeholder for X in the model contructed using the feed_dict object.batch_Y is also formed of 5 data points but the data starts from the position 1 ahead of starting position of that of batch_X.These are also sent to the model object and is used in the calculation of loss function.
  - The model is run on the batch_X and batch_Y values and other inputs given and it gives 4 outputs variables/vectors: logits(last layer output),last_state,loss(cost of error func)
  - Prev lbatch utput is sent to the model as initial state for next batch.

- Model testing
  Similar algorithm for training but rather here we use the last 30 days data for predicting the results on the pre Trained Model
- Running Simulations
- Graph Plots

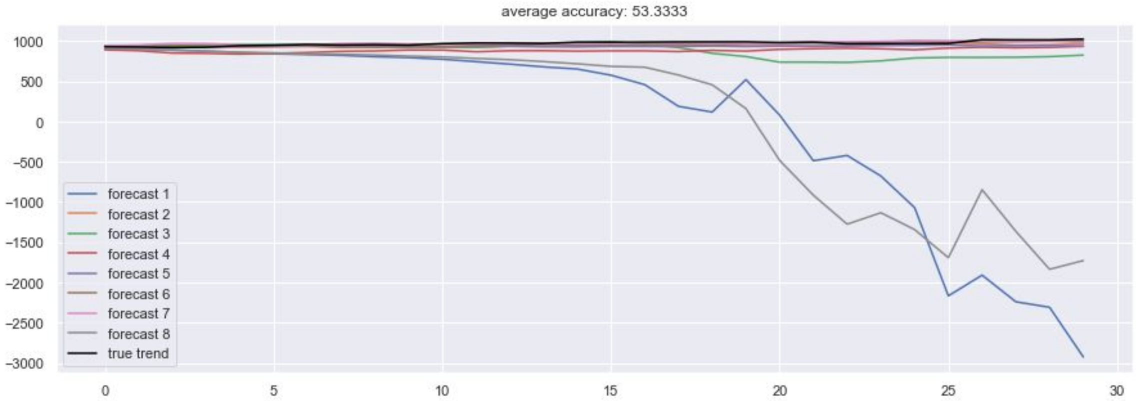
Vanilla R



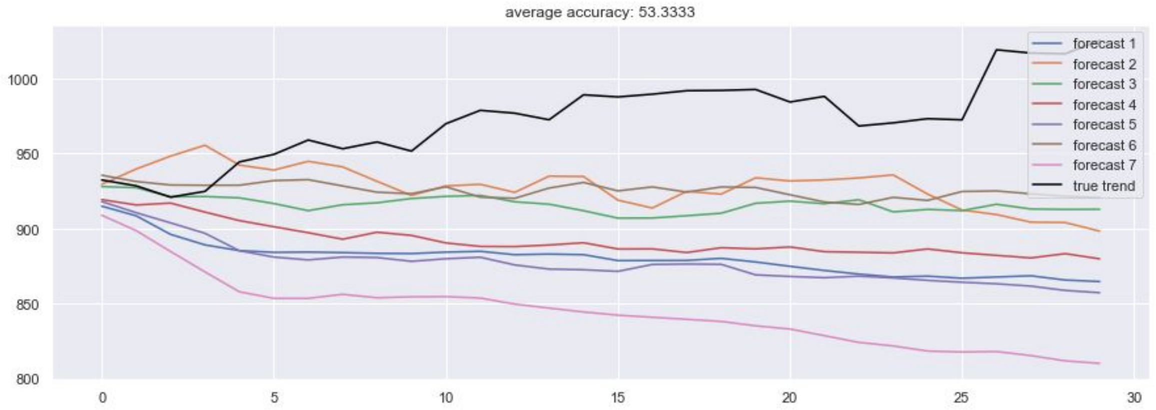
LSTM



GRU



CNN



Stock volatility prediction using Sentiment Analysis

Approach

A. **News Article Generation:** Web Scraper made in Python is used to get the list of links of news articles.This uses the python module “scrapy”.We use the website “Reuters.com” for this purpose.The role of this file is to scrap out a list of news articles from the website about a specific company mentioned by us date wise and output it in a csv file with 2 columns date and url of news article.
B. **Sentiment Score Generation**

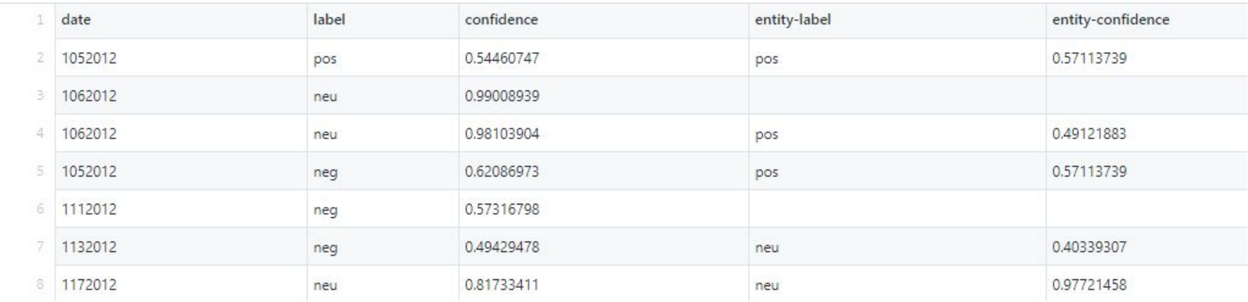
The output csv file from Step A is used as an input in this step. Here we use “Rossete” API for text analysis for the contents for each article. This file outputs a csv file which contains the columns date, label,confidence,entity-label and entity-confidence. The column label can take 3 values:-

1. pos:-expressing the article is positive for company and its share
2. neg:- expressing the article is negative for company and its share
3. neu:- expressing the article is neutral for company and its share

The confidence colomn presents the probability/confidence with which the label is decided for that specific date

> label defines the attitude/emotion of the whole article,while entity_label defines the emotion towards a specific entity here in this case is the company name.Similarly confidence is the probability for the label while entity_confidence is probability for entity_label

Below is a subset of sentiment scores generated for Apple company news articles:-



date	label	confidence	entity-label	entity-confidence
1052012	pos	0.54460747	pos	0.57113739
1062012	neu	0.99008939		
1062012	neu	0.98103904	pos	0.49121883
1052012	neg	0.62086973	pos	0.57113739
1112012	neg	0.57316798		
1132012	neg	0.49429478	neu	0.40339307
1172012	neu	0.81733411	neu	0.97721458

C. Calculating Bullishness and Sorting it in Chronological Order

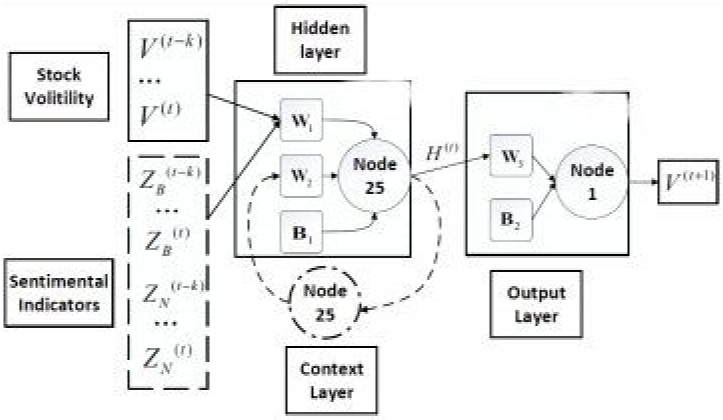
Using the sentiment score calculated for each article in the previous step, We calculate the bullishness score of the selected stock on a per day basis. Bullishness for each day is calculated as:
Bullishness score = (Sum of positive confidence – sum of Negative confidence)/(total articles)
In the next file, we arrange these scores in a chronological order.

D. Combining sentiment Analysis with Stock Prices

For each day, combine the sentiment analysis of that day with the corresponding stock price. In case, the sentiment score of a day is not available, mark the sentiment score as the score of the latest previous day available. The output file after this step contains sentiment score and stock price for each day.

E. RNN Model

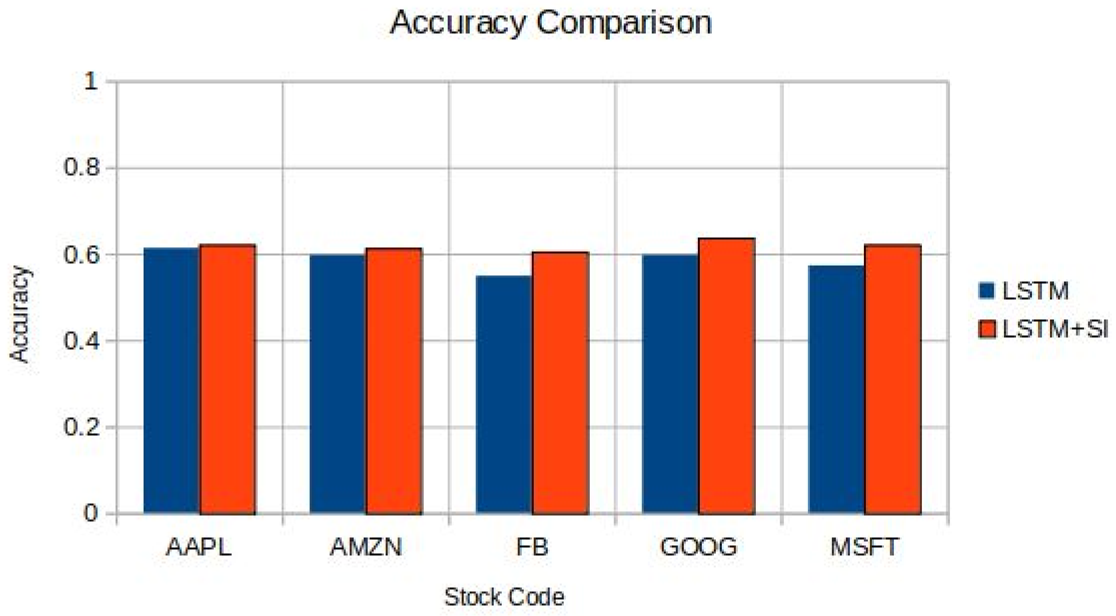
For training our model, the sentiment score goes z score normalization followed by min-max normalization. Then, we construct a 2 layer LSTM followed by 2 Dense layers. Input of which will be previous k days stock price and it's corresponding sentiment score where k is the window size. The output is the prediction of stock price. Hyperparameters of this model are as follows batch_size=128, epochs=200, validation_split=0, verbose=0).



F. Results

Accuracy and the best k for RNN+EMM and RNN

	AAPL	AMZN	FB	GOOG	MSFT
LSTM	0.6129	0.5968	0.5484	0.5968	0.5726
LSTM+SI	0.621	0.6129	0.6048	0.6371	0.621
k(LSTM)	4	7	6	7	6
k(LSTM+SI)	4	9	7	7	5



G. Conclusion

After completing the prediction, we can clearly observe that the results using sentiment analysis offer marginally better results as compared to predicting without them. Hence, we conclude that analyzing sentiments in order to predict the volatility in the stock market is not worth the effort.

Datasets Used

- News articles from Reuters.com on particular stock ranging from 2012 - 2018
- Historical Stock Prices from Yahoofinance.com

List of papers referred

S. No	Research Paper	Author(s)
1.	*Stock Prices Prediction using Deep Learning Models (2019)*	*Jialin Liu, Fei Chao, Yu-Chen Lin and Chih-Min Lin*
2.	*Neural networks for stock price prediction (2018)*	*Ren-Jie Han, Yu-Long Zhou and Yue-Gang Song*
3.	*Stock Trend Prediction using News Sentiment Analysis*	*Kalyani Joshi , Prof. Bharathi H. N. and Prof. Jyothi Rao*
4.	*Sentiment Analysis of Twitter Data for Predicting Stock Market Movements (2016)*	*Venkata Sasank Pagolu*
5.	*Stock Volatility Prediction Using Recurrent Neural Networks with Sentiment Analysis*	*Yifan Liu, Zengchang Qin, Pengyu Li and Tao Wan*
6.	*Market Trend Prediction using Sentiment Analysis: Lessons Learned and Paths Forward*	*Andrius Mudinas and Dell Zhang*
7.	*Stock Forecasting using M-Band Wavelet-Based SVR and RNN-LSTMs Models*	*Xiaodi Wang and Abdul Hasib Rahimyar*
8.	*A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing*	*Sidra Mehtab and Jaydip Sen*

Timeline

Task	**Achieved by Date**
Familiarising with stock market	*14th Aug.*
Literature survey of stock forecasting using deep learning techniques	*31st Aug.*
Collecting Dataset of historical share prices	*5th Sept.*
Training LSTM Models	*14th Sept.*
Training GRU Models	*22nd Sept.*
Training Vanilla and CNN Model	*30th Sept.*
Literature survey on Stock Volatility prediction using Sentiment Analysis	*8th Nov.*
Dataset collection for sentiment Analysis	*12th Nov.*
Training RNN Model for LSTM+SA	*26th Nov.*
Competing Project Report and Presentation	*1st Dec.*