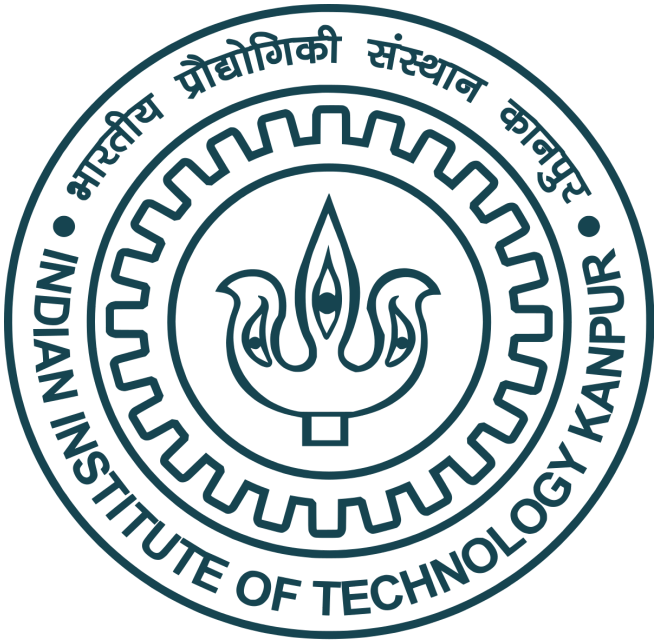
# ECO399A Project

## Prediction of the stock market using News Headline




Aditya Jindal | 17817048

**Supervisor: Prof. Wasim Ahmad**

# Abstract

From the age of Indutrialization, stock market prediction is an area of extreme importance to an entire economy. The behavior of human investors determines the stock price, and a smart investor can determine stock prices by using openly available information to predict how the market will act by the change in human bhevaior.  Financial news articles can thus play a significant role in influencing the stock movement as humans respond to the surrounding information through social media, news, and other social interactions.

Previous research has suggested a relationship between news report and stock price movement, as there is some time lag between when the news article is released and when the market has moved to reflect this information. Also, they are inherently volatile and reflect diverse macroeconomic and microeconomic trends, which makes them difficult to predict.

In this undergraduate research project, we use natural language processing in conjunction with a deep neural network to predict the daily change in the Dow Jones Industrial Average price. Our model was slightly better than random guessing for indicating whether the stock market would rise and fall on a given day. In a nutshell, we built a text classification system to categorize news articles containing references to world news and predict the impact of world news on the stock price.

# Introduction

Much of the world's wealth can be found in large financial markets, which allow individuals and big/small organizations to purchase shares of companies on a public exchange. The price of a company's shares is subject to a wide range of variables–some intuitive, i.e., systematic risk, and some not, e.g., non-systematic risk. The task of using these inputs to predict a stock's price movement is a trillion-dollar industry.

For decades, only large financial corporations and expert analysts have had access to the data required for understanding financial markets. In the past decade, however, publicly-provided market information has enabled independent investors(freelancers) to make much more intelligent, informed investment decisions.

For most of the stock market history, investment decisions have been made by educated and informed humans. Though the stock market has displayed a consistent overall trend for its existence, the results of attempts to 'beat' the stock market have been mixed. It is not clear which variables influence stock prices and by how much. Microeconomic trends and consumer preferences often determine the price of a single stock, but these factors can be challenging to predict; even experts sometimes perform this task very poorly.

For this research project, we focus on predicting the price of the Dow Jones Industrial Average (DJIA), a price-weighted index of thirty large American companies representing the state of the domestic economy as a whole. Since significant changes in the DJIA index are caused by factors that affect each of its constituent stocks, its price reflects macroeconomic trends. Previous research has shown that macroeconomic trends can be predicted based on daily news, and further research shows that news headlines alone can be used to achieve the same results.

We have modelled a neural network that predicts the float amount by which the DJIA index increases or decrease based on a textual input consisting of twenty-five news headlines concatenated into one string per day.

# Related work

Past research work had used text drawn from various sources to generate predictions about stock market movement. Another research work used Twitter to extract public opinion and was able to use a neural network to transform these opinions into relatively accurate predictions about the DJIA closing price. Other leading researchers have experimented with event detection algorithms and other natural language processing techniques to create rich input for neural networks to learn their relationship to stock prices.

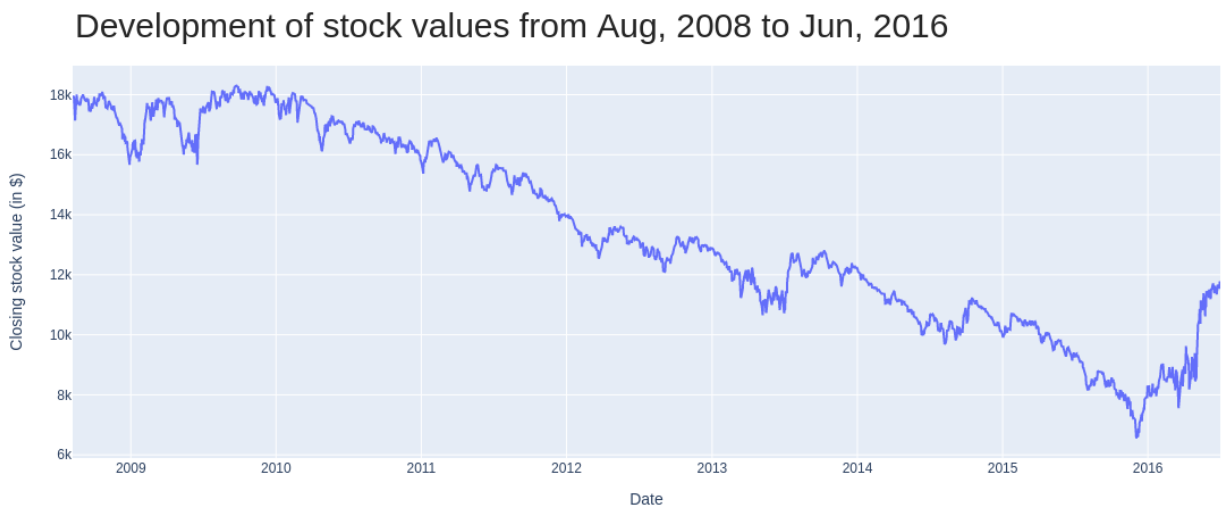
Others have focused on developing a neural network architecture that can convert these inputs into accurate predictions. Since macroeconomic trends are closely tied to public opinion and sentiment, effective sentiment analysis is critical for our objective. One promising model uses a convolutional neural network in conjunction with a recurrent neural network to conduct sentiment analysis of short texts. This work applies to our objective because it extracts sentiment from too brief documents to contain many contexts, like news headlines.

# Dataset and Features

We used a publicly available dataset of Dow Jones Industrial Average (DJIA) prices and daily news articles available from Kaggle. The DJIA data included the opening and closing price of the DJIA, plus other metrics such as its daily trading volume, daily high, and daily low, for every market day from August 8, 2008 through July 1, 2016. This dataset can also be downloaded from Yahoo Finance.

| | Date | Open | High | Low | Close | Volume | Adj Close |
|---|---|---|---|---|---|---|---|
| 0 | 2016-07-01 | 17924.240234 | 18002.380859 | 17916.910156 | 17949.369141 | 82160000 | 17949.369141 |
| 1 | 2016-06-30 | 17712.759766 | 17930.609375 | 17711.800781 | 17929.990234 | 133030000 | 17929.990234 |
| 2 | 2016-06-29 | 17456.019531 | 17704.509766 | 17456.019531 | 17694.679688 | 106380000 | 17694.679688 |

**Fig: An example of all the metrics included for one day in our dataset.**




The news data included the 25 most popular news headlines for every day (including weekends) in the same range, sourced from Reddit News. The 25 news headlines for each day are those that received the most votes from Reddit users.

For example, corresponding to the price change on 1st July, 2016, there are 25 global news headlines ranging from "The president of France says if Brexit won, so can Donald Trump" to "Spain arrests three Pakistanis accused of promoting militancy".

Our data included 10% test set of our total data, dev set included 10% of our total data, and finally train data included 80% of our total data.

Another key feature to each day in our dataset is the previous 5 days' stock price changes. The intuition behind this feature is that stock prices depend not only on the daily news headlines, but also recent previous stock price movements. The number of previous stock price changes is a hyperparameter that we tuned appropriately.‘ Thus, we implemented an LSTM to create a sequential model in addition to newsheadlines over time. Our workflow and architecutre is explained in the Methods section of this paper.

For our news input data, we clean the data before training our model. To do so, we convert the headlines to lower case, replace contractions, remove unwanted characters, and remove stop words that are irrelevant for this natural-language

processing task. Next, we create embeddings, which are low-dimensional feature vectors representing discrete data like strings or, in this case, news headlines. We use pre-trained, 300-dimensional GloVe embeddings for all words with corresponding GloVe embeddings and use random embeddings of the same size for words not found in GloVe's vocabulary. The GloVe vocabulary is used to create a vector space representation of words to capture semantic and syntactic meaning. For more information on GloVe, see references.

To structure our previous 5 days' stock price changes, we create a two-dimensional matrix with rows that represent each day in our corpus. Every row then contains the last 5 days' stock price changes. Thus, we end up with a (1988,5) NumPy matrix. For the first 5 days, we zero-pad the non-existent entries. For instance, on July 1, 2016, the first row of this input data would be like this in the given figure.

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| -1.47363277 | 0 | 0 | 0 | 0 |
| -1.47363277 | -1.79395714 | 0 | 0 | 0 |
| -1.47363277 | -1.79395714 | -1.85602318 | 0 | 0 |
| -1.47363277 | -1.79395714 | -1.85602318 | 1.18877929 | 0 |
| -1.47363277 | -1.79395714 | -1.85602318 | 1.18877929 | 4.208866 |

**Fig: This is a sample of the first 5 rows of our two-dimensional array. The first row corresponds to the previous stock price changes on August 8, 2008 which is all zeros since this is the beginning of our dataset. The sixth row corresponds to August 13, 2008 which contains the five previous stock price changes.**

# Methods

In this study, supervised methods and econometrics methods have been explored: Linear Regression, Time Series such as Autoregressive Integrated Moving Average (ARIMA), machine learning methods like SVM, Random forest and advance deep learning methods like sequence models including LSTM, etc.

## I. A basic Linear regression model :

$$Y = X\beta + \in$$

We set this traditional linear regression model to set up baseline results for our bilateral trade model, we further induce some dummy variables for the same to see the better $R^2$ results.

## II. ARIMA model:

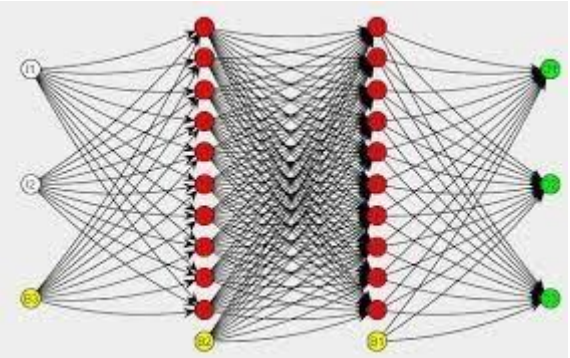
$$ln\left(Y_t\right) = \alpha + \beta\, ln(Y_{t-1}) + \epsilon$$

To investigate further we introduce more lags terms in this model, to achieve better relationships between time agnostic variables.

## III. Random forest:

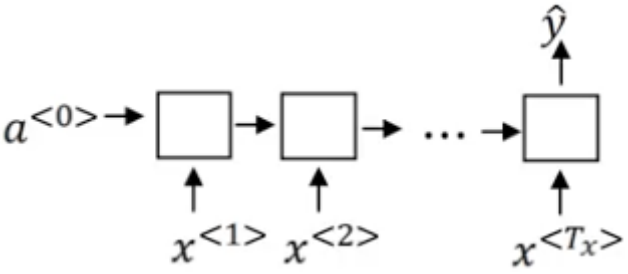
Random forest is another ensemble machine learning model builds upon multiple decision trees and joins them together to get a more accurate and consistent prediction. They assumed to be a good choice for predictive modeling because they are easy to understand due to their tree type structure and are also very robust. The basic goal of a decision tree is to split a population of data into smaller sections.

## IV. Fully Connected, Feedforward Neural Networks using Logarithmic Features:



Since we assume no a priori knowledge on interactions between features (except for Gravity Model), we utilized a fully connected neural network. To find an optimal architecture of the neural network, we will compare architectures with different layer and node-setups

## V. Fully Connected, sequence LSTM models:

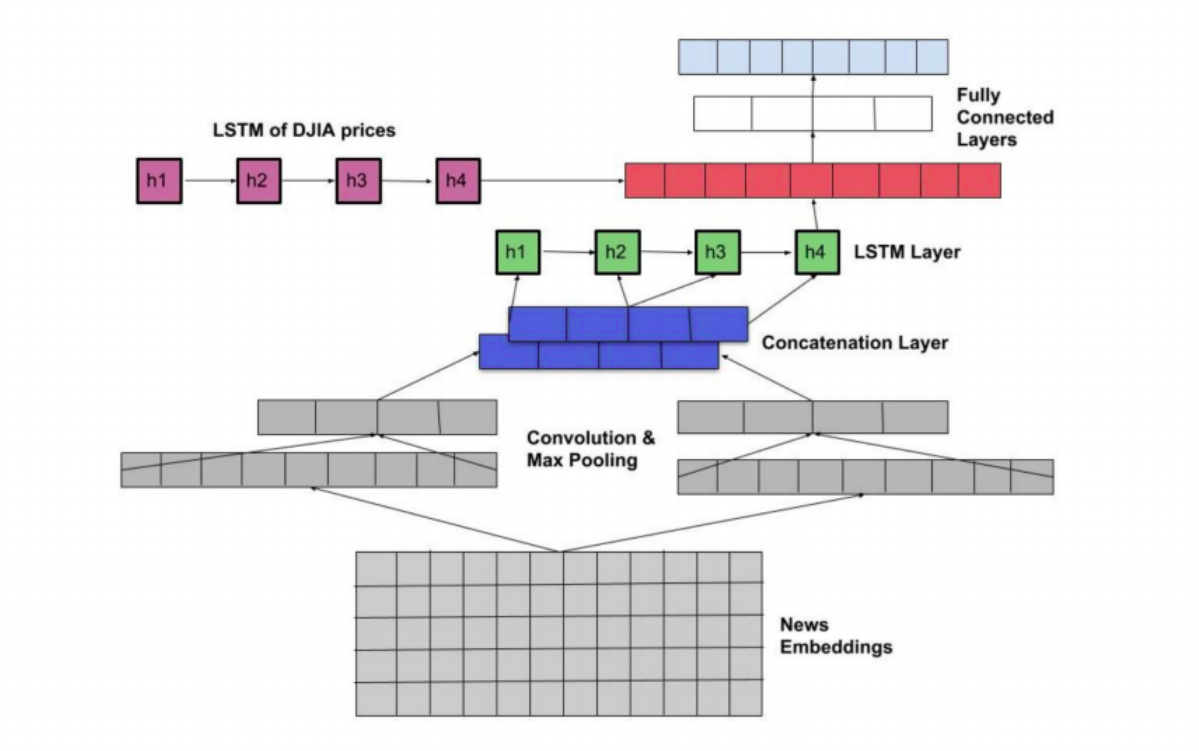



We use this LSTM model of deep learning to capture the time effect, this is similar to the ARIMA model, but it's quite robust and can account for non-linear relationships too unlike the basic time series model.

We are using stacked lstm , which is we take 2 or more layers of LSTM one onto another , which captures the relationship and captures non-linear behavior more accurately.

The inputs to our neural network are sequential, as we discussed in the above section. More specifically, every day from August 8, 2008, to July 1, 2016, has 25 news headlines. Additionally, each input vector also contains the difference between the DJIA stock price at time t and the DJIA stock price at t− 1. In this model, we are not predicting the actual stock price; instead, we predict that stock price will rise or fall for each day, i.e., the change in its price (a positive or negative difference). To reduce the effect of Inflation, we also normalize the data by subtracting the mean and dividing by the standard deviation of the stock price changes.

As our baseline method, we simplified this problem into a pretty straightforward sentiment analysis task. Since we are working with brief news headlines, we used Wang et al's(refer below) approach to sentiment analysis, feeding sentence embeddings into a CNN that takes the local features and inputs these results into an RNN for sentiment analysis of short texts or news headlines. The output of the RNN is fed into a fully connected network, which aims to learn its relationship to sentiment. The model architecture is shown in the below figure. We used the mean squared error as our loss function and Adam optimization algorithm. This network is fully implemented using Tensorflow-based Keras.




**Model architecture**

In this architecture, we apply windows and convolutional operations with different matrices in the CNN while maintaining sequential information in the texts. Then, the RNN takes in these encoded features and learns long-term dependencies.

More specifically, for our RNN, we use a Long Short-Term Memory (LSTM) that learns long-term dependencies. Rather than using a traditional feed-forward network, an LSTM detects features from an input sequence and carries the information over a long distance. Our model is built on the Keras and scikit-learn machine learning frameworks.

In addition to the sentiment analysis on the textual inputs, we also included the change in the price of the stock market for the previous twenty days. We feed the previous twenty days' price changes into an LSTM, which then becomes another input to the fully connected network. This should help our model recognize long-term trends in the DJIA's index price.

# Experiments

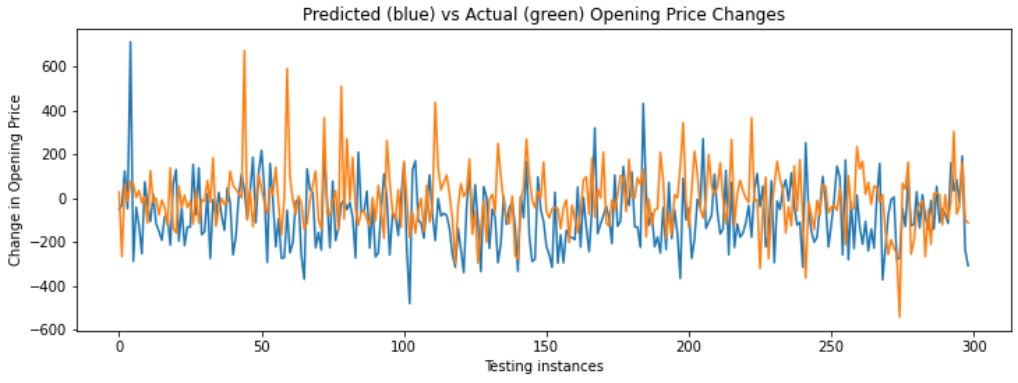
Our baseline model relied on news data as the sole source of input. And to keep it simple, we apply ridge regression and then an auto-regressive model. With this model and the optimal hyperparameters found, we could correctly predict whether the stock market would rise/fall 54% of the time. This was our starting point. From this baseline, we iterated through several experiments to develop our final model. Our hyperparameters include:

- Different learning rates.
- The value of p in dropout.
- The number of layers to include in the fully connected layer.
    - The number of elements in the LSTM.
    - The number of convolution layers on the textual data.

We used both grid search as well as random search to find the best possible values of these hyperparameters.

To put the performance of our model in perspective with other datasets, we created a simple logistic regression model that uses NYC weather data to predict whether the price of the DJIA rises or falls. With this simple (and logically unrelated) data, we were able to predict positive/negative price movements in the DJIA with 52% accuracy–slightly better than flipping a coin.

We tried six different models, varying the input data (weather, news, and previous stock prices) and the model architecture. Our best accuracy results came from the most profound and most comprehensive networks, which used more convolutional and fully connected layers.



# Results

Using the previous stock price changes and global news headlines, we achieved 55.28% accuracy with a mean absolute error of 71.40, meaning that this model more accurately predicted price changes than our baseline.

Further tuning our hyperparameters, we increased the depth and width of our network by using 3 Fully Connected(FC) layers, 2 convolutional layers, and 128 hidden dimensions in the fully connected network to further increase our accuracy, reaching 58.28%.

Our best model used 5 fully connected layers, 5 1D convolutions for the textual news data , and 256 hidden dimensions in the fully connected layer, achieving 61.31% accuracy in predicting a positive or negative change in the DJIA, a 7.03% increase of our baseline model. Our model was well-fit to the data, since the training error for our our best model was around 58% while our test accuracy was 60.31%.

One notable trend throughout our experiments is the reasonably high mean absolute error, which provides insight on the average magnitude of error in our predictions regardless of direction. Our model is best suited for a binary classification task of predicting a rise/fall and struggles with determining the magnitude of stock price changes.

| Model | Accuracy | RMSE |
|---|---|---|
| **Logistic regression** | 52.21% | 1.113 |
| **Ridgre regression** | 53.30% | 1.094 |
| **AR model*(Baseline)** | 54.1 % | 1.001 |
| **CNN and LSTM(no lag)** | 56.5% | 0.983 |
| **CNN and GRU(with lag)** | 58.5% | 0.952 |
| **CNN and LSTM(with lag)** | **61.4%** | 0.931 |

**Table: Different models and accuracy I tried along the project**

# Conclusion

Advance machine learning models outperform traditional economics models. Since our data has both news headline and time dimensions, using the LSTM model extended to a panel setting may be a better approach that we can take in the future.

Since the neural network approach expects to show good results with capturing nonlinear interaction effects among other economic indicators, including many more economic features may also be effective in further improving our prediction.

Also, our model was able to outperform our baseline at the task of predicting whether the DJIA would rise or fall each day. We started at 54% accuracy and eventually got to 61.31%–a marginal difference and still barely better than a coin flip. This model didn't completely succeed at its task of predicting the overall stock price movements. However, it did far better than even many human analysts can do.

It was promising that our NN model reacted positively to the addition of historical market price data, as that shows that the basic model with a textual news input can be augmented and bolstered by building in relevant alternate input sources. This makes sense because a typical stock market is subject to many factors, any of which can cause the market's price to swing wildly. Historical price data is just one, and there are many others we could include in our model. Some exciting ideas include tweets from the presidential Twitter account and price information from foreign markets. It could also be interesting to apply this model on a company-specific basis, feeding it news headlines focused on a single company. We suspect that this data would be even more relevant because it is directly relevant to the company's health (which should be reflected in its stock price).

# Tools Used

1) Pandas for data management
2) Scikit-learn for linear regression/ kernels
3) Matplotlib for plotting
4) Keras for neural networks
5) Numpy for data transformation

6) Tensorflow for sequence models based stacked LSTM

7) Jupyter Notebook for coding and plotting graphs

# References


1. Xiaojun Zeng Johan Bollen, Huina Mao. Twitter mood predicts the stock market.Journal of Computational Science, 2(1):1–8, 2011.
2. Ting Liu Junwen Duan Xiao Ding, Yue Zhang. Deep learning for event-driven stock prediction. 2014.
3. Zhiyoung Luo Xingyou Wang, Weijie Jiang. Combination of convolutional and recurrent neural network for sentiment analysis of short texts.COLING, 2016.
4. Aaron7sun. Daily news for stock market prediction, 2016.
5. Christopher D. Manning Jeffrey Pennington, Richard Socher. Glove: Global vectors for word representation, 2014.
6. David Currie. Predicting-the-dow-jones-with-headlines.https://github.com/Currie32/Predicting-the-Dow-Jones-with-Headlines.
7. Open Source Francois Challet. Keras deep learning framework.
8. Open Source. scikit-learn machine learning framework.