DS5220 – Supervised Machine Learning

Prediction of Stock Movement using News Headlines

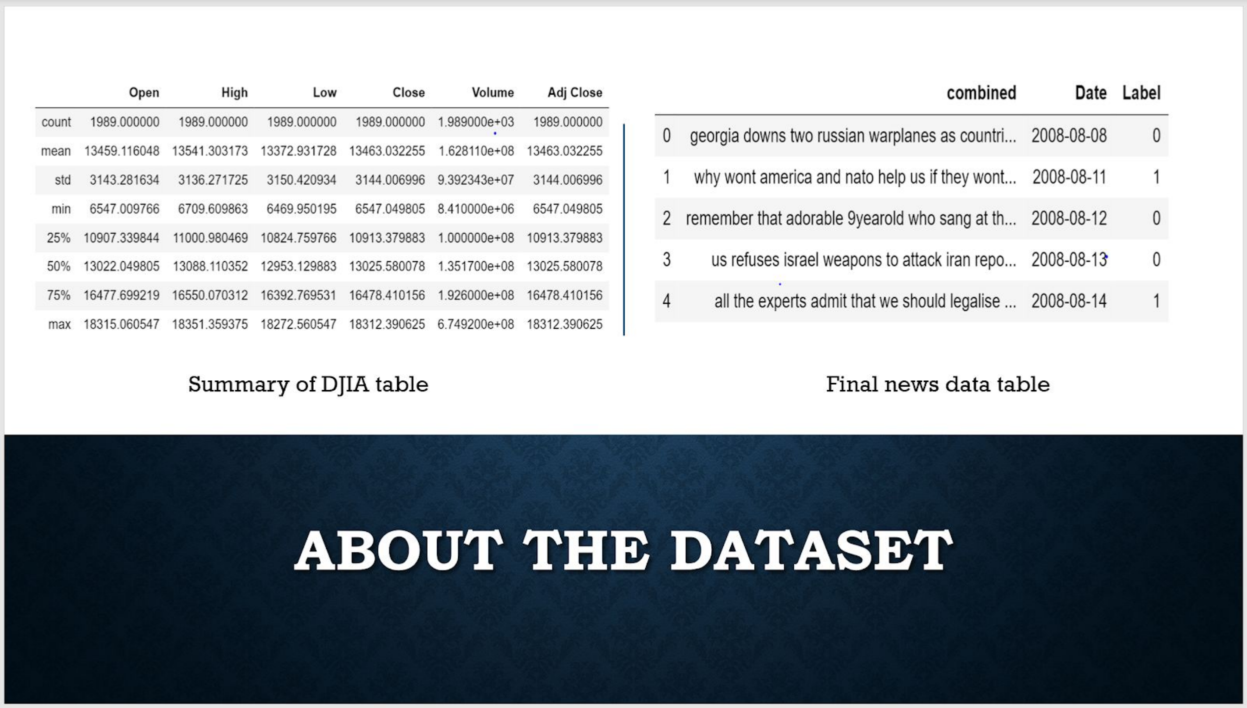
Pranshu Diwan
001347331

Ruthvik Ravindra
001377866

Objective:

We plan to predict the movement of the stock price using the news headlines and past DJIA index data. Through the course of the project, we have explored various techniques, analyzed and formulated them to build the best model for our case. The business case was to help investors make more informed decisions based on our model which utilizes headlines.

The stock market going up or down can be represented by the Dow Jones Industrial Average (DJIA). It is a stock market index that measures the stock performance of 30 large companies listed on stock exchanges in the United States. Apart from that, the data contains some other stock market measurements like low, high, close, and volume. We aim to predict the stock movement using the top 25 news headlines. We started off by performing data cleaning and wrangling techniques.



	Open	High	Low	Close	Volume	Adj Close
count	1989.000000	1989.000000	1989.000000	1989.000000	1.989000e+03	1989.000000
mean	13459.116048	13541.303173	13372.931728	13463.032255	1.628110e+08	13463.032255
std	3143.281634	3136.271725	3150.420934	3144.006996	9.392343e+07	3144.006996
min	6547.009766	6709.609863	6469.950195	6547.049805	8.410000e+06	6547.049805
25%	10907.339844	11000.980469	10824.759766	10913.379883	1.000000e+08	10913.379883
50%	13022.049805	13088.110352	12953.129883	13025.580078	1.351700e+08	13025.580078
75%	16477.699219	16550.070312	16392.769531	16478.410156	1.926000e+08	16478.410156
max	18315.060547	18351.359375	18272.560547	18312.390625	6.749200e+08	18312.390625

Summary of DJIA table

	combined	Date	Label
0	georgia downs two russian warplanes as countri...	2008-08-08	0
1	why wont america and nato help us if they wont...	2008-08-11	1
2	remember that adorable 9yearold who sang at th...	2008-08-12	0
3	us refuses israel weapons to attack iran repo...	2008-08-13	0
4	all the experts admit that we should legalise ...	2008-08-14	1

Final news data table

ABOUT THE DATASET

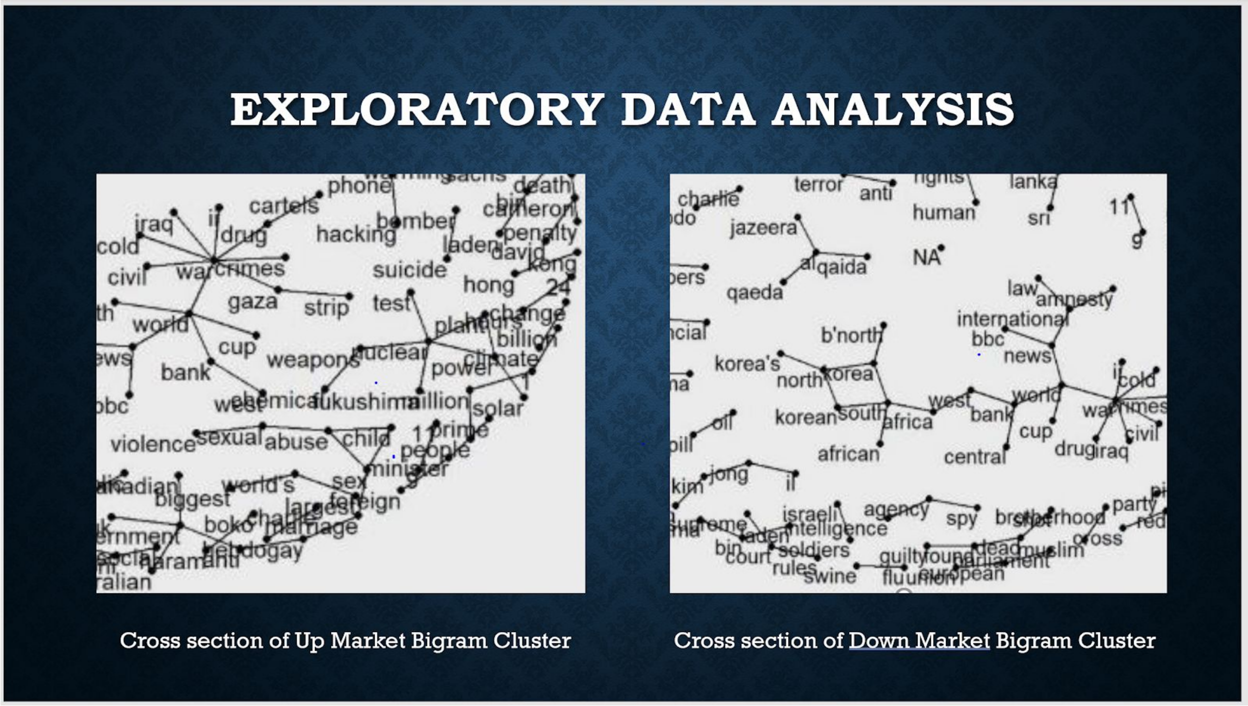
Dataset:

We had three tables: The upload_djia having the DJIA values like Close, Open, High, Low and Volume (1989x7), and the Reddit news data having the headlines (73608x2) from January 2008 to July 2016. We merged this into one data frame for our project, called the combined (1989x27).

Data Wrangling:

The data contained 15 missing values that were removed. The text data was cleaned using basic text preprocessing techniques like lower casing, punctuation removal and removing the bold tags. After getting the clean text, we combined all 25 columns into one column containing the text corpus for the day. For feature engineering, we further Stemmed and Lemmatized the same corpus to get a rich set of vectors that were fed into the machine learning models.

Stemming and Lemmatization was used to get the base form of the word from a set of related forms of the word. For example, ‘eating’ and ‘eats’ will be converted to ‘eat’ when the word is stemmed. And ‘better’ is lemmatized to good. We then continued to extract information and features by performing Exploratory Data Analysis on the clean text.



EXPLORATORY DATA ANALYSIS

Cross section of Up Market Bigram Cluster

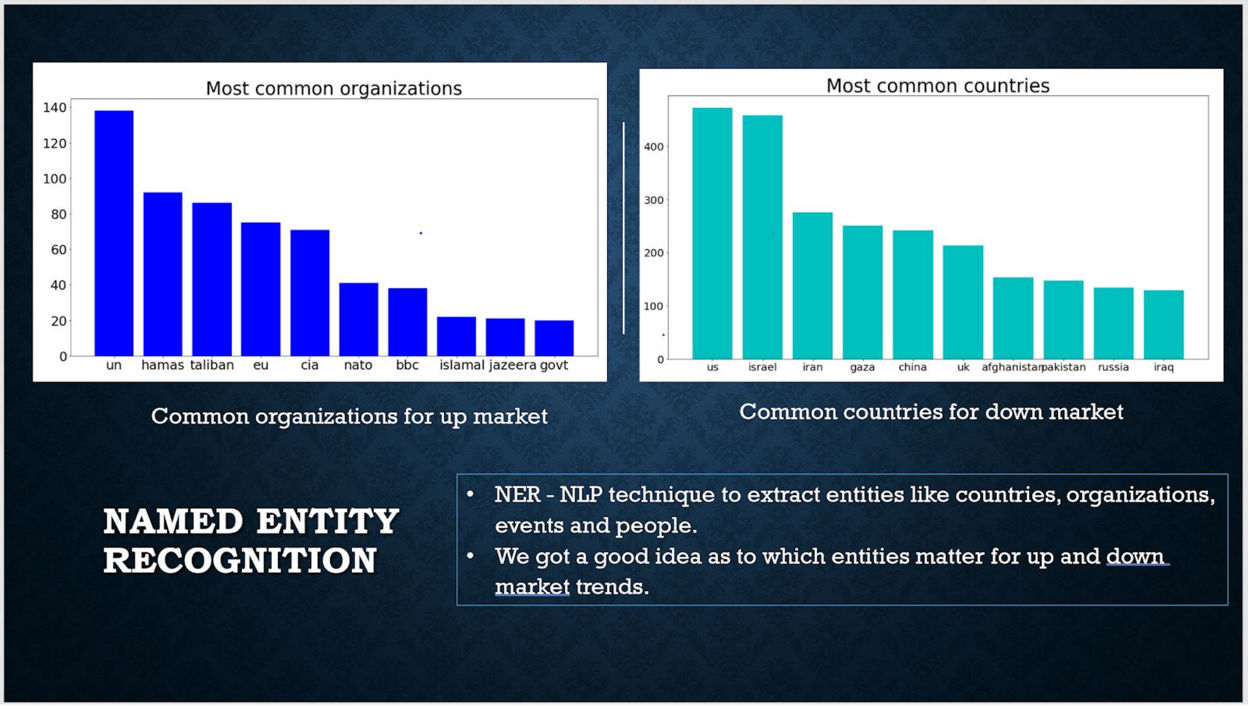
Cross section of Down Market Bigram Cluster

Exploratory Data Analysis:

We created separate word dictionaries for the up and down market to see the influence on stock movement for each corpus. We then found both dictionaries had similar words like ‘US’, ‘Government’, ‘War’, ‘World’ with varying frequencies. These results did not reveal any new insights due to similarities. So we thought of building bigrams clusters for each market direction to find differences in the clusters.

Bigram clusters for the up market revealed clusters such as (Iraq, Gaza, civil war) and (solar, power, climate). Whereas for the down market, the bigram clusters revealed close connections to (North Korea, South Korea, Kim jong) and (Israel, intelligence, soldiers, agency and spy).

This does not mean that words which are commonly thought of in negative sentiment such as words in the (Gaza, civil war) cluster influences the stock market in a negative way.



Common organizations for up market

Common countries for down market

NAMED ENTITY RECOGNITION

- NER - NLP technique to extract entities like countries, organizations, events and people.
- We got a good idea as to which entities matter for up and down market trends.

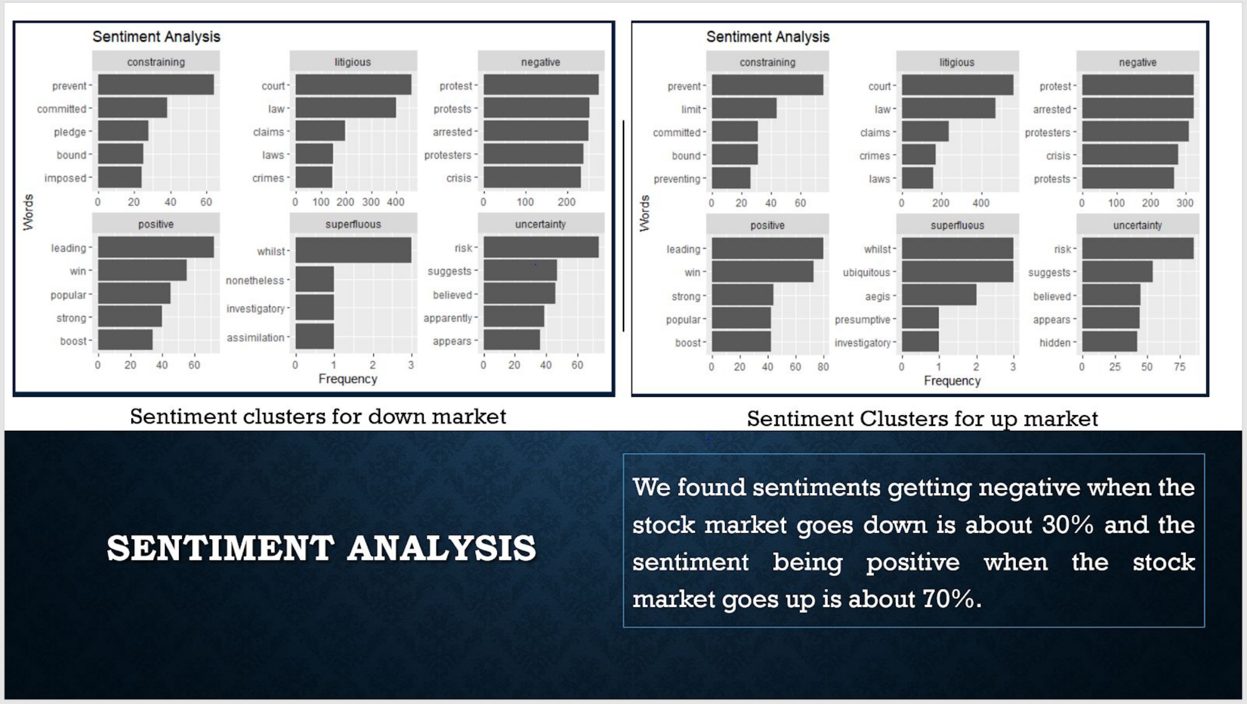
Named Entity Recognition:

In Natural Language Processing, Named Entity Recognition (NER) is a technique to identify and extract named entities from the text corpus into predefined categories such as Countries, Organizations, People and Events.

To gain more details from the observed bigram clusters, we performed NER for both up and down market clusters. Using NER, we extracted the most important entities contributing to both directional trends.

The most common organizations for the up market are Hamas, Taliban, CIA, NATO. On the other hand, the most important countries for down market are Israel, Iran, Gaza and China.

We used the weighted frequency of the entities as a predictor of our model. Since word frequencies approaches do not give a complete picture. We then conducted sentiment analysis and clustering.



Sentiment clusters for down market

Sentiment Clusters for up market

SENTIMENT ANALYSIS

We found sentiments getting negative when the stock market goes down is about 30% and the sentiment being positive when the stock market goes up is about 70%.

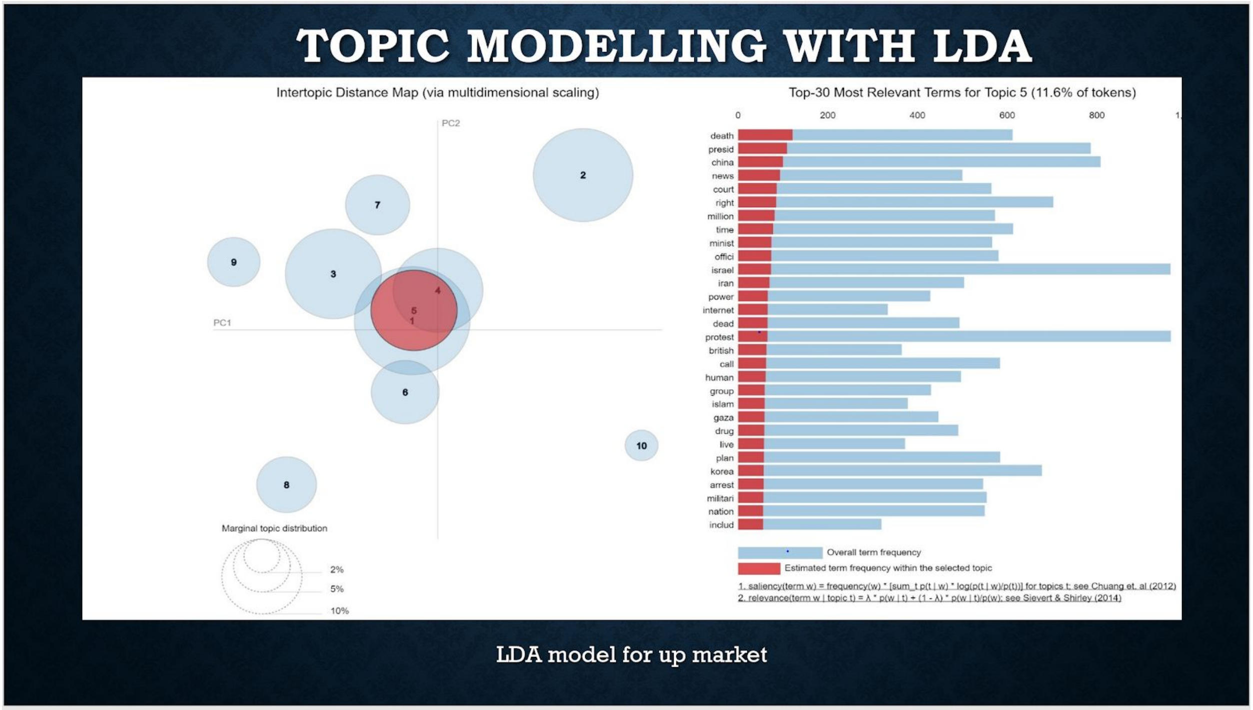
Sentiment Analysis:

We analyzed the sentiments – positive, neutral and negative that were prevalent in the text corpus. We found sentiments getting negative when the stock market goes down is about 30% and the sentiment being positive when the stock market goes up is about 70%.

This indicates a positive sentiment has a high probability that the stock market is going to increase. We then clustered the sentiments to find the frequency of words for 6 sentiments in the Loughran lexicon.

From the graph, we can see that even though the words appearing in the sentiments are similar, the frequency with which they appear is significant. For example, risk and believe appear on both sides but with varying frequencies.

We then performed topic modeling using Latent Discriminant Allocation (LDA) on the text corpus to gain the most important topics.



TOPIC MODELLING WITH LDA

LDA model for up market

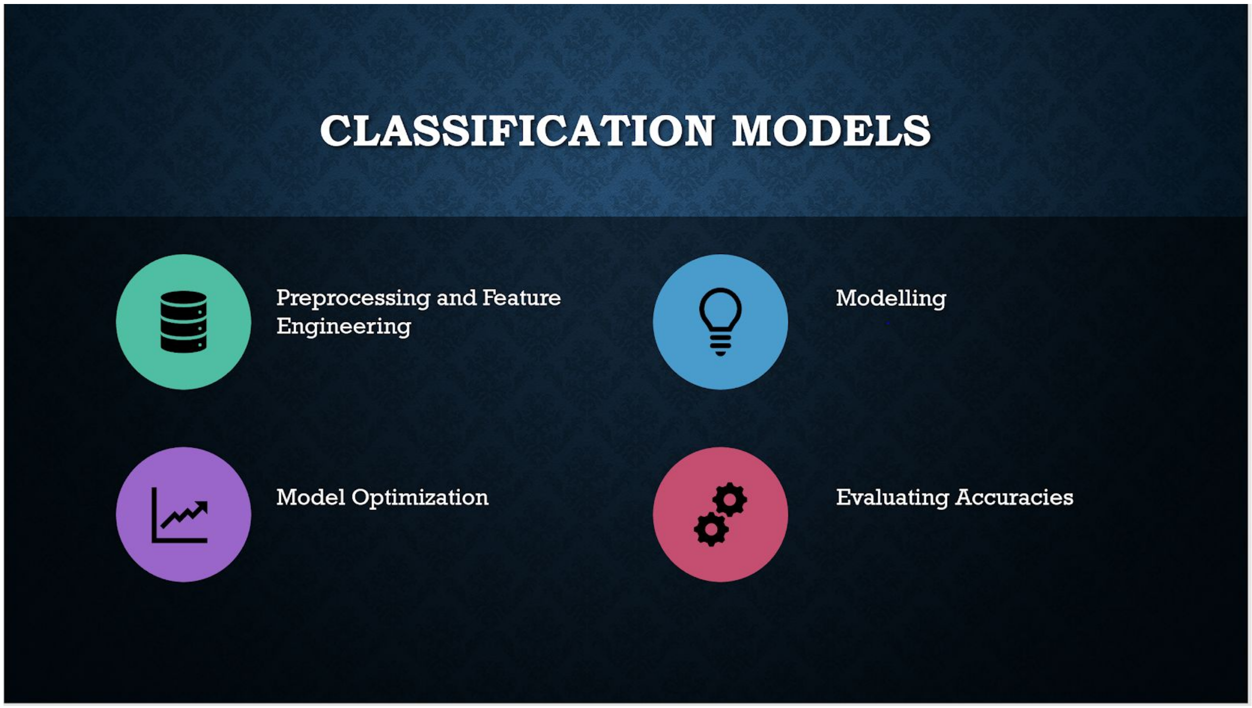
Topic Modeling with Latent Discriminant Allocation:

We used the up and down market text corpus to form Bag of Words and Term Frequency - Inverse Document Frequency(Tf-Idf) vectors. The Bag of Words gives the terms with the frequencies whereas Tf-Idf gives us the most relevant frequencies for each document. These were used to generate the top 10 topics for up and down market movements.

We aimed to get the most relevant terms for each topic using topic modeling with LDA. These topics were then transformed into probability vectors that were used as a predictor to classify the market movement.

Latent Dirichlet Allocation is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.

For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. Since we obtained all the significant predictors, we proceed to build our classification models.



CLASSIFICATION MODELS

Preprocessing and Feature Engineering

Modelling

Model Optimization

Evaluating Accuracies

Preprocessing and Feature Engineering:

We converted the text corpus into Bag of words(Count) and Tf-Idf vectors. These vectors along with the LDA probability vector was used to predict stock movement using various classification models.

Modeling:

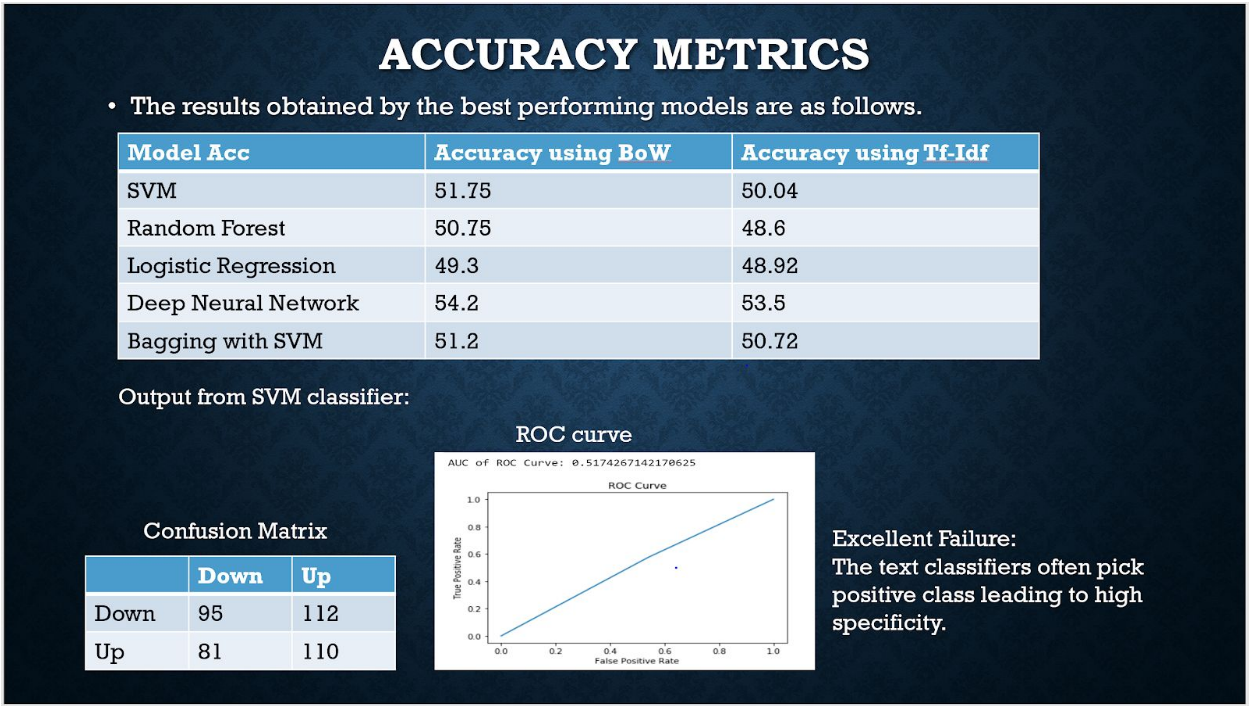
The classification models used were Logistic Regression, Naïve Bayes, Support Vector Machines, KNN, Deep Neural Networks, Random Forest, Gradient Boosting and Bagging with SVM Methods.

Model Optimization:

We improved the feature set by using bigrams and trigrams instead of single words. We also tried to improve the accuracy by optimizing the models using hyperparameter tuning methods like GridSearch and cross-validation.

Evaluating Accuracies:

We used accuracy metrics such as Score, Confusion Matrix and ROC curve to compare the various models that were used for classification.



ACCURACY METRICS

- The results obtained by the best performing models are as follows.

Model Acc	Accuracy using BoW	Accuracy using Tf-Idf
SVM	51.75	50.04
Random Forest	50.75	48.6
Logistic Regression	49.3	48.92
Deep Neural Network	54.2	53.5
Bagging with SVM	51.2	50.72

Output from SVM classifier:

Confusion Matrix

	Down	Up
Down	95	112
Up	81	110

ROC curve

```
AUC of ROC Curve: 0.5174267142170625
```

Excellent Failure:
The text classifiers often pick positive class leading to high specificity.

Accuray Metrics:

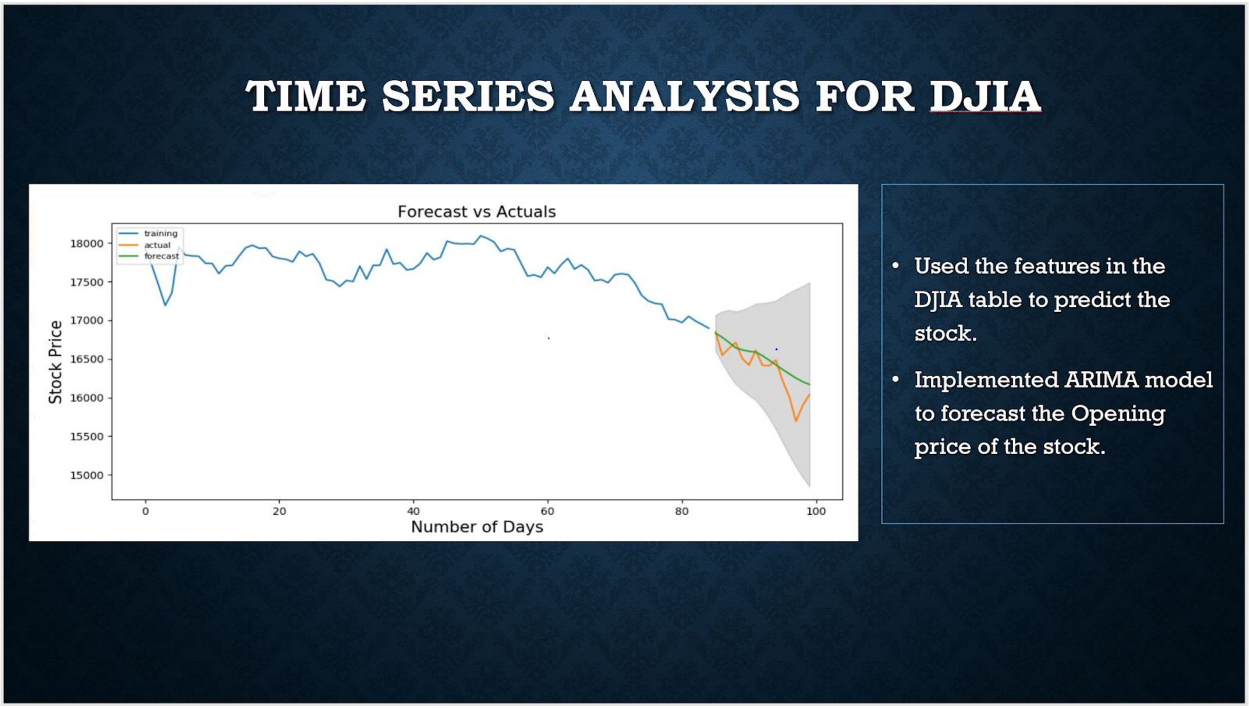
We built and trained our models using libraries like Scikit-Learn, Pandas, NumPy, Keras. The above table summarizes the top-performing models used for classification.

Accuracy scores for all the models are largely the same. The Deep Neural network gave us the best score. However, none of these models had a competitive performance that could be considered to be put into production.

Excellent Failure:

As we see from the confusion matrix, the classifiers are not able to pick up on the nuances of the text data using the BoW and Tfidf vectors. So, they end up classifying the news articles as spam or ham instead. This leads to the positive class being picked more often and high specificity in the model.

Due to these results, we decided to instead use the DJIA data to forecast the values using time series analysis.



TIME SERIES ANALYSIS FOR DJIA

- Used the features in the DJIA table to predict the stock.
- Implemented ARIMA model to forecast the Opening price of the stock.

Time Series Analysis:

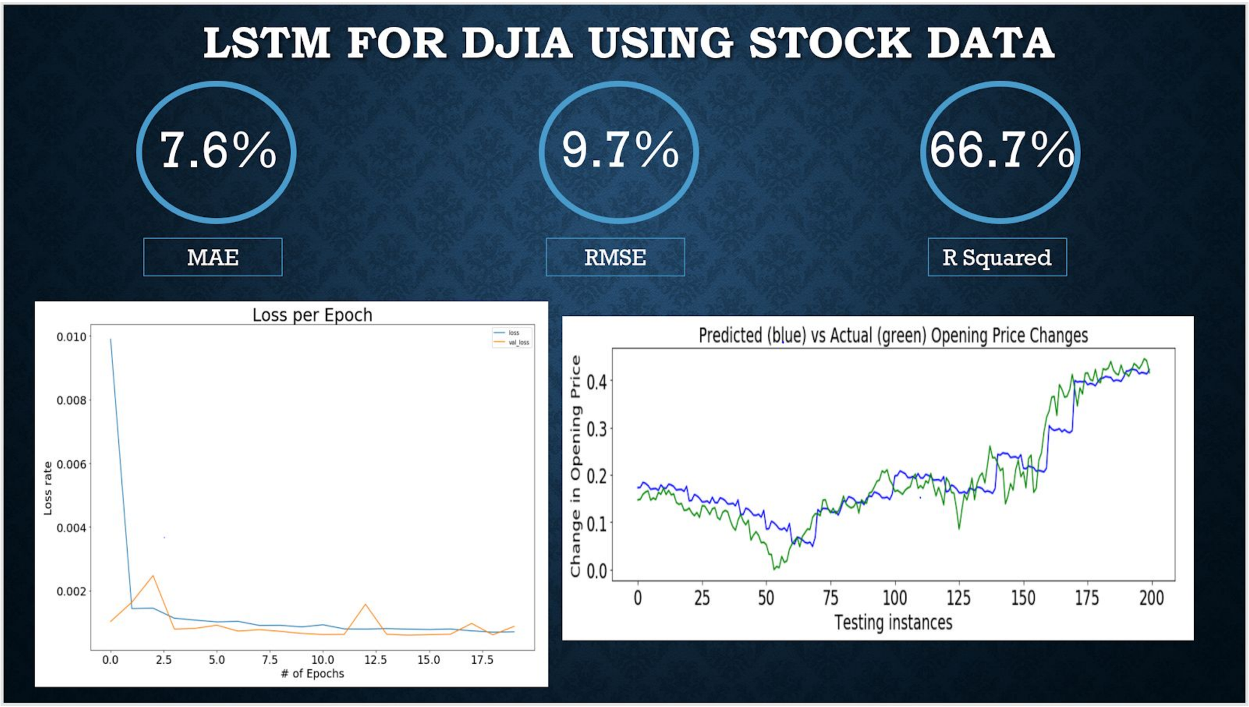
Using the columns such as Close, Volume, High and Low, we forecasted the Opening price of the data. We performed the Dicky-Fuller test on the Stocks data to test the stationarity of the data.

It returned a p-value of 0.6 which is significant and we accepted the null hypothesis which meant we needed to use the Auto-Regressive Integrated Moving Average algorithm to model the stocks data.

We built an ARIMA model that forecasted the Opening price of the stock with a (2,2,1) setting. This indicates that we used 2 Autoregressive terms, 2nd order differential for obtaining non-stationarity and 1 lag term. We can improve the model by optimizing it to account for seasonality and using better order selection methods.

As we can see from the above graph, the forecasted values follow the trends shown by the actual values but it does not give a very accurate prediction.

To get better results, we employed the LSTM model on the stock data.



LSTM FOR DJIA USING STOCK DATA

7.6%

MAE

9.7%

RMSE

66.7%

R Squared

LSTM on stock data:

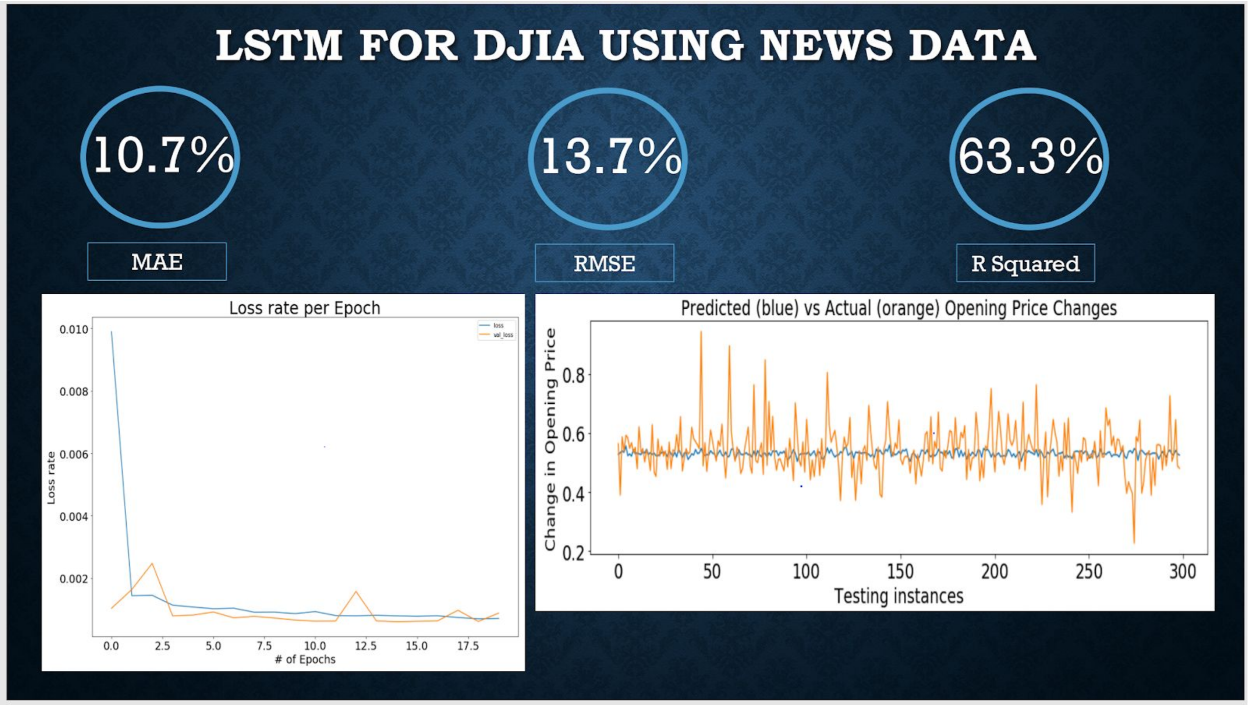
We used the stock data such as Close, High, Low, Volume to predict the Opening price of the DJIA index using a Long Short Term Memory(LSTM) model.

LSTM is a special case of a Recurrent Neural Networks capable of learning order dependence in sequence prediction problems. We used LSTMs as compared to RNNs due to the size of the data which allows the LSTMs to work faster and gives more accurate results.

The results achieved are documented as above. The Mean Absolute Error is 7.6%, Root Mean Squared Error is 9.7% and R Squared is 66.7%.

The graph on the left shows the change in loss rate for the training samples and the validation samples with the increase in the number of epochs that are used to train the model.

As we can see from the output graphs, the model is able to predict and keep up with the actual changes that occur on the dataset. Seeing favorable results on the stock data, we decided to employ the LSTM model on our text corpus as well.



LSTM FOR DJIA USING NEWS DATA

10.7%

MAE

13.7%

RMSE

63.3%

R Squared

LSTM on news data:

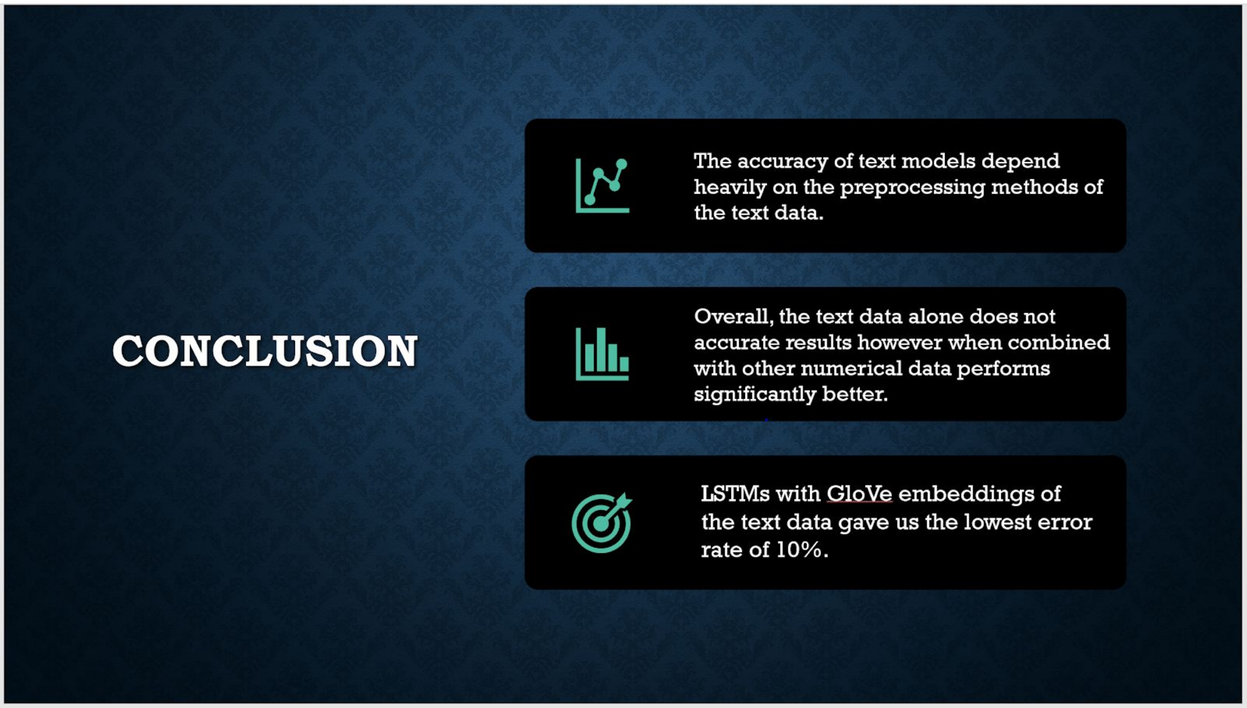
We used the Stanford NLP GloVe method to create embeddings of the text corpus that were used as predictors for the LSTM model.

The results achieved are documented as above. The Mean Absolute Error is 10.7%, Root Mean Squared Error is 13.7% and R Squared is 63.3%.

The graph on the left shows the change in loss rate for the training samples and the validation samples with the increase in the number of epochs that are used to train the model.

As we can see from the output graphs, on the test data the model is able to predict the mean change of the actual values. However, there is a great change in the actual test values compared to the predicted values.

Hence, this model does not provide a very accurate prediction as compared to one which uses the stock data.



CONCLUSION

The accuracy of text models depend heavily on the preprocessing methods of the text data.

Overall, the text data alone does not accurate results however when combined with other numerical data performs significantly better.

LSTMs with GloVe embeddings of the text data gave us the lowest error rate of 10%.

Conclusion:

We can postulate that before modeling, it is necessary to clean and preprocess the data by our employed methods to avoid generating noise in the predictions
The word frequencies generated using bag of words, bigrams, NERs do not always give a complete picture of the data. They give only words which have a market-moving attribute.

As seen from the above results, we saw that models that used only the text data as features did not perform as well compared to the time series models which used the stock data. This can be further emphasized by the performance of the LSTM models. We can further say that models that use text data combined with other past numerical data give better results.

The best results were obtained when we used LSTM model on stock data with GloVe embeddings giving an accuracy of 87%.