Analysis of the impact of news on Stock Market

Aditya Lakshman Prabhu (prabhu.a@husky.neu.edu)

Deeksha Doddahonnaiah (doddahonnaiah.d@husky.neu.edu)

Varsha Muroor Rao (rao.va@husky.neu.edu)

Abstract - Stock market prediction is considered to be a very complex task owing to its highly unstable and dynamic nature. World events have a significant impact on the movement of stock rates. We investigate the role of news in stock market prediction by finding out the top categories of news that affect the stock price. We also evaluate the accuracy of this model by building a stock price change prediction model using the topic distributions in the news headlines in combination with news sentiment polarity. This will help us learn how to improve the topic modelling features used in the prediction model.

Keywords: Topic modeling, LDA, Stock market prediction, sentiment analysis, neural networks

I. INTRODUCTION

Stock market prediction has been a very interesting field of research. In order for the stock prediction algorithms to effectively predict the stock prices, the right set of data would have to be used as input. The knowledge of the topics and the kind of effect they have on the movement of stock prices becomes a key factor in narrowing down the number of articles that the stock market prediction algorithms use. In this project, we aim to find the top categories of news that affect the stock market and the kind of effect they have; positive or negative. We use a combination of topic modelling and sentiment analysis to build features for a supervised learning model that would then predict the change that a headline can bring in the stock prices. For our experiments, we use news headlines instead of full articles as usually the headlines have the needed summary of the articles, and articles can have a lot of noise that can have a skewed impact on the analysis of important topics. Also, processing full sized articles for months/years will take a lot of processing power and time.

II. RELATED WORK

A number of approaches have been implemented in the field of stock price prediction. An experiment conducted by Heeyoung et al. used 8-K financial reports to forecast stock price changes and showed that using text boosts the prediction accuracy [1]. Another research conducted by Kalyani et al. used non-quantifiable data such as financial news articles about a company to predict future stock trend using news sentiment classification [2]. Ayman et. al used financial news and historical stock market prices combined with sentiment analysis to predict future stock prices [3]. Another paper which was written by Adriano et.al used LSTM networks to predict the future trends of stock prices based on the price history [4]. Based on these papers, we came to the conclusion that text data/news articles have a significant role in predicting the stock prices. Since we wanted to find out if the current world events impact the stock market, we decided to use news articles, without filtering them specifically for financial news. In addition, we also used NASDAQ historical stock prices to distinguish a relationship between world news and its effect on NASDAQ prices.

Since our aim was to find out the top categories of news that affect the stock market, we had to use the concept of topic modelling. We found out that several experiments were done on modeling the topics in journalistic texts. Carina et al. used the news from New York Times to perform content analysis using LDA on nuclear technology related articles [5]. Liwei et al. performed topic modeling using LDA on financial research reports to predict the correlated industries of financial news [6]. Based on these research experiments, we decided to use Latent Dirichlet Allocation (LDA) model to classify the topics. But what we wanted to work and study is not only the effect of topics on the change in stock markets but if this knowledge in combination with sentiment analysis of the news headline can help build a better stock price change prediction model.

.

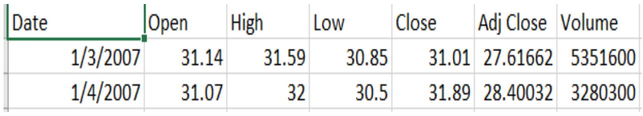
III. DATASETS

By referring to platforms like Kaggle and UCI Machine Learning Repository, we have concluded that the Reuters dataset is one of the most reliable and widely used dataset in Machine Learning experiments. For the historical stock prices, we learned from Investopedia and several data science related blogs like ‘towardsdatascience.com’ that Yahoo Finance provides accurate historical stock prices that can be used for studies like ours. Summing it up, below are the two input datasets that we used for performing the experiments:

(1) The Reuters dataset consists of 8,551,441 news titles, links and timestamps for the news from Jan 2007 to Aug 2016. (~2GB)[7]. Each day’s headlines is stored in a separate file, where the file name indicates the date.

(2) Since we need the historical stock price data for analysis and prediction for the same time period as the news articles, we use NASDAQ stock price data for the period from Jan 2007 to December 2016, taken from Yahoo Finance website (~200 KB)[8]. This ‘.csv’ (comma separated values) file contains the open, close, high and low prices of NASDAQ corresponding to each date entry available in the Yahoo finance database for NDAQ.

Fig 3.1 and Fig 3.2 show a snapshot of the sample input dataset that we use.



Date	Open	High	Low	Close	Adj Close	Volume
1/3/2007	31.14	31.59	30.85	31.01	27.61662	5351600
1/4/2007	31.07	32	30.5	31.89	28.40032	3280300

Figure 3.1: Historical NASDAQ stock price data collected from Yahoo Finance

```
(S'http://www.reuters.com/article/filmNews/idUSMAN15969020070220’
S'ts'p4S'20070219\xc2\xa011:49 PM EST’
S'title'p6S'Philippine group opposes U.S. film on Abu Sayyaf’ )
```

Figure 3.2: Link, title, timestamp of one of the headlines taken from Reuter dataset

The final goal of the experiments is to find topics (from the news headlines) that are relevant and have an effect on the stock market prices. This analysis is done by finding the topics and their correlation with the change in stock prices as well as sentiment from the sentiment analysis of the topics. We work on getting a correlation matrix which contains the correlation coefficients for the topics and the kind of impact they have on the stock market along with the correlation with the sentiments scores for positive, negative, neutral and compound sentiments.

IV. METHODOLOGY

Data preprocessing:

The raw Reuters news dataset had HTML tags in documents and other unnecessary data like timestamps, headings and markers. We cleaned the individual documents to fetch only the file name and all the headlines for that day in a dictionary. The date formed the key in the dictionary and the headlines corresponding to that day was stored in an array and used as the value for that key. The stock price dataset is of a CSV format that was read and put into a data frame for further use.

Preprocessing is done on the dictionary values before feeding the data to the topic modeling algorithms. The following steps were performed:

1. Tokenization and case normalization: All the headlines were tokenized using the Gensim library and also converted to lowercase.
2. Stopwords removal: Tokens like ‘I’, ‘for’, ‘the’, etc., are removed from the document using Gensim as they can affect the co-occurrence of tokens in the headlines. Tokens less than 3 characters were also removed as they tend to be either an acronym or have no meaning.
3. Lemmatization: Using WordNet lemmatizer from NLTK, the tokens that are verbs are all changed to the first person and also to present tense.
4. Stemming: Using NLTK’s Snowball Stemmer, All the tokens are reduced to their root forms such that different usages of the same words shouldn’t be accounted for as two separate tokens.

For the stock price data, we calculate the change in stock prices as the difference of “close” and “open”. Each day now has a change value and based on our modeling attempts described later, we broadcast the price change of each day to all the headlines of the same day. Also, for the Reuters dataset, we filter to include headlines only for the days for which the stock data is available. This would remove headlines for days when the market is closed.

Also, since this is a really huge data set, and our systems have limited processing power, it would take a lot of time for our study and some of the feature building, so for some of the experiments, we decided to use only the data for the period from Jan 2014 to December 2016 (~ 500 MB). This reduced the number of news titles to 1,737,490. Further in our analysis we subset this data to include days, for which the absolute change is above a certain threshold $\boldsymbol{\alpha}$. Based on the distribution of change we chose $\boldsymbol{\alpha}$= 0.5.

Once the data is cleaned and preprocessed it can be used as input to topic modeling algorithms. For our experiments, we use LDA for topic modeling and create a classification model using Neural Networks for extrinsic evaluation.

Latent Dirichlet Allocation (**LDA**) 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 [9,10]. The underlying data, i.e., news headlines is considered to originate from a generative process that contains hidden variables. LDA assumes that every document is made up of a set of topics and that a topic can be associated with a document with a certain probability. The output given by the LDA model is a set of topics and the corresponding words along with the probability with which the topic can generate those words. The input to the LDA is the TF-IDF model and dictionary of words obtained from the headlines. We have used the LDA model provided by the Gensim library [11].

1. Generate bag-of-words model from the dictionary of preprocessed headlines.
2. Generate TF-IDF model from the bag-of-words obtained from step 1
3. Get the dictionary of all words using Dictionary module from Gensim
4. Run LDA multicore to generate the document-topic matrix.

We performed the topic modeling experiments for different number of topics and compared the evaluation scores, and what was the optimal number of topics that we chose is discussed in the evaluation section.

Neural networks: We use state-of-the-art Neural networks, they are known to best work with non-linear time-series data such as stock prices. We use the headline topic distribution probabilities to create feature vectors and build a classification model to determine the accuracy of the model to compare our baseline model and our improved model. We optimize the NN hyperparameters for both models and split the dataset into 75% for the training set and 25% for the test set to determine the accuracy.

To find out which topics affect the stock price, we followed two approaches:

1)**The Baseline model**: For our baseline model we use LDA for topic modelling with the same preprocessing as mentioned in the methodology but without filtering data according to the dates or any specific change value. We get a set of topics with words and word distribution probabilities. Next, for every headline corresponding to a day, we find the topic distribution probabilities. Then we calculate the average of these probabilities for every topic across all the headlines for a day. We then multiply this average topic probability for the given topic with the change value for the same date. This gives the probability of the effect that topic has on the stock price change for that particular date. Averaging this value for a particular topic among all the dates gives the average probabilistic change the topic has on the stock prices. The average of these numbers across all the dates sampled from the corpus is used to score the topics to know the general effect of a topic on the change in stock price.

Using the average topic distribution for each day as the input vector, and the corresponding change in stock for the day as the output value, we trained a neural network to get the accuracy of our topic model. The baseline model without any filtering on input data performed poorly, with around 21.23% accuracy. This can be explained by the large amount of volatile stock data which has insignificant changes in prices. Also, since the topic distribution has really a small correlation with stock change, averaging the topic distribution further impacts the correlation. This data being highly inconsistent and small caused the neural network to make inaccurate predictions in spite of using the best model hyperparameters, which leads us to our next attempt of refining the data to create a subset which can help the NN to better correlate the topics and stock movements.

We used Tensorflow’s Keras to model the neural network with the following hyperparameters [12]:

Number of nodes in hidden layer 1 - 200
Number of nodes in hidden layer 2 - 10
Number of nodes in output layer - 1
Optimizer - Stochastic Gradient Descent(sdg)
Loss - accuracy
Metric - accuracy
Activations – Relu(hidden) and sigmoid(output)

2) Using Topic Distribution per headline in combination with sentiment scores:

To further improve our predictions, we incorporated sentiment analysis of headlines. We can say that this is a good additional feature to support our modeling as it can to a fair degree determine if a particular topic seems to have positive polarity or a negative polarity.

Our approach was to find the correlation values between the topics, sentiments and the response variables using correlation matrix. To build correlation matrix, we would need the feature vectors of 19 features which include topic distribution and sentiment score (15 topics + 4 sentiment scores) across all headlines and the response variable values. The response variable here is the change in stock prices for the day, which indicates positive and negative movements of stock.

There were two steps involved in building the feature vectors. First, a vector containing the topic probability distribution was found for each headline, and later, the sentiment polarity scores were appended to this vector. To find the sentiment scores, we used the SentimentIntensityAnalyzer module from the ‘nltk.sentiment.vader’ package. Given a sentence, the Vader algorithm outputs sentiment scores to 4 classes of sentiments; positive, negative, neutral, and compound. These 4 values obtained for each headline was appended at the end of the previously found feature vector of the topic probability distributions. A correlation matrix was derived from this data in combination with the stock price change, which was then used to get the topics that affect the stock price change.

Our first attempt with predictions was to use NN regression to predict actual change in prices of the stock data. This model although gave a significantly low Mean Squared Error, looking closely we found out that the data in itself is volatile with very small fluctuations in price everyday. This caused the NN to fit a line, which was almost always around 0 to minimize error, for most of the data. Although regression is a good technique for prediction of change, it would not work well with the data we had in hand with very moderate to little market fluctuations.

As our focus was to predict the impact of topics on stock market movement, we first classified the response variable to 0 or 1, based on the direction of change, 0 would correspond to stock market dropping, and 1 corresponds to the market rising. With this as our output variable we tuned the model hyper parameters to best predict the direction of change.

On sub-setting the data as described previously, we work on further tuning the NN hyperparameters to obtain better accuracy. We found that sub-setting the data increased the accuracy by about 36%, which will be explained further in the results. The following hyperparameters have worked best for the model:

V. Evaluation

Evaluation for topic modelling is a tricky task since there is no labelled “ground truth” data to compare the results. However, topic modeling requires defining parameters beforehand like the number of topics to divide the data into. So, model evaluation is important in order to find an optimal set of parameters for the given data. In a good model, there should be enough topics to be able to distinguish between overarching themes in the text but not so many topics that they lose their interpretability and such that all topics overlap each other.

We use two types of evaluation metrics to evaluate our topic model; intrinsic and extrinsic. Intrinsic evaluation metrics computes the quality of the model independent of an external application. Perplexity and ‘u_mass’ topic coherence measure are the two measures that we are using to intrinsically evaluate our topic model [13, 14]. Perplexity describes how well a model predicts a sample, i.e. how much it is “perplexed” by a sample from the observed data. The closer the perplexity value to 0 the better the model. Perplexity is calculated by taking the log likelihood of unseen text documents given the topics defined by a topic model. But a lower perplexity doesn’t really mean the topics have good human interpretability. But topic coherence has better human interpretability than perplexity. Topic coherence helps to know whether the words which are clustered under a particular topic tend to co-occur together. It is basically rating the interpretability of the topics generated by the model. U_mass measure compares preceding and succeeding words from ordered set to determine coherence values. Higher the coherence values, the better it is. We performed topic modeling for different number of topics, and then chose that model which gives a balance between perplexity and coherence values, to derive the results from the prediction model. Table 5.1. shows the perplexity and coherence values for different models.

Topics	Perplexity	U_mass Coherence	Accuracy of prediction models	Average topic coherence
15	-10.12541	-6.16215	0.56821671	-9.243
20	-10.43214	-7.083038	0.55070861	-11.0830
30	-10.79817	-8.139258	0.55311685	-24.4178
40	-11.24406	-8.9419	0.55748698	-36.2354

Table 5.1: Evaluation measures for different models

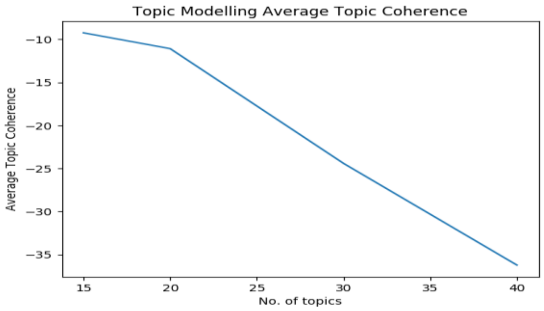


Figure 5.1: Number of topics versus average topic coherence

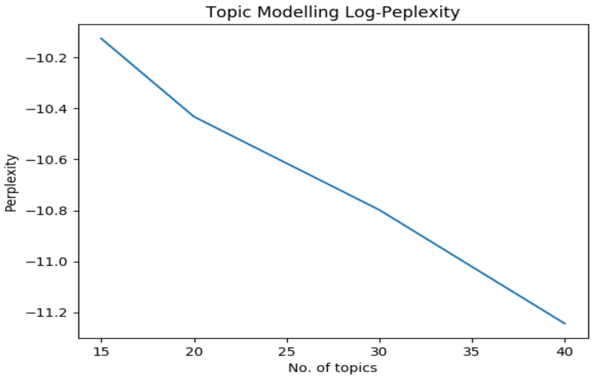


Figure 5.2: Number of topics versus perplexity

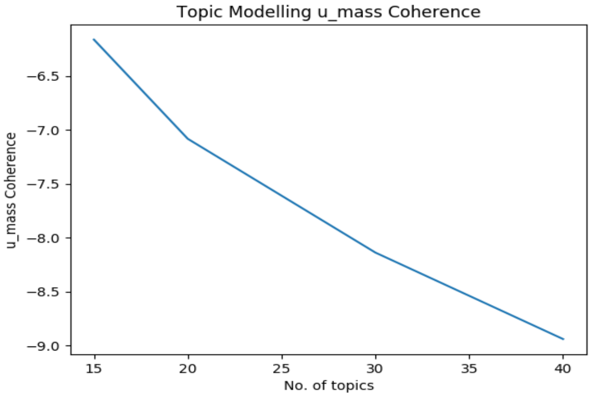


Figure 5.3: Number of topics versus u_mass coherence values

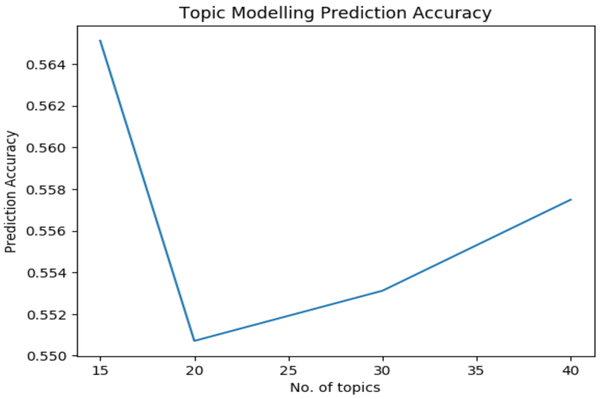


Figure 5.4: Number of topics versus prediction accuracy

We also use ‘pyLDAvis’ tool to visualize the fit of our LDA model across topics and their top salient words as shown in Fig. 5.5 and 5.6 [15]. If the top salient words for each topic are not coherent enough then the topic model can be improved upon. Also, if the average intertopic distance measure is low and many topics are overlapping that means the model can be improved.

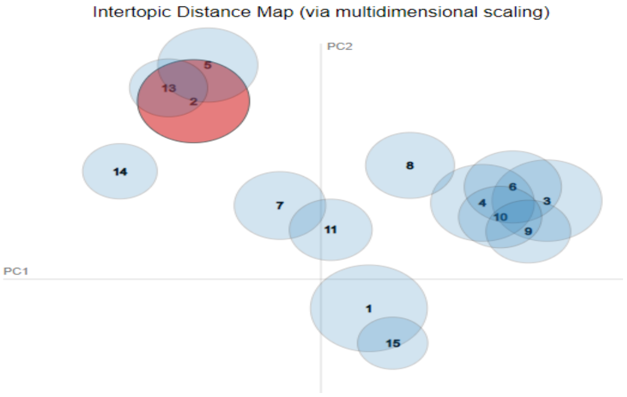


Figure 5.5: Intertopic Distance Map for 15 topics

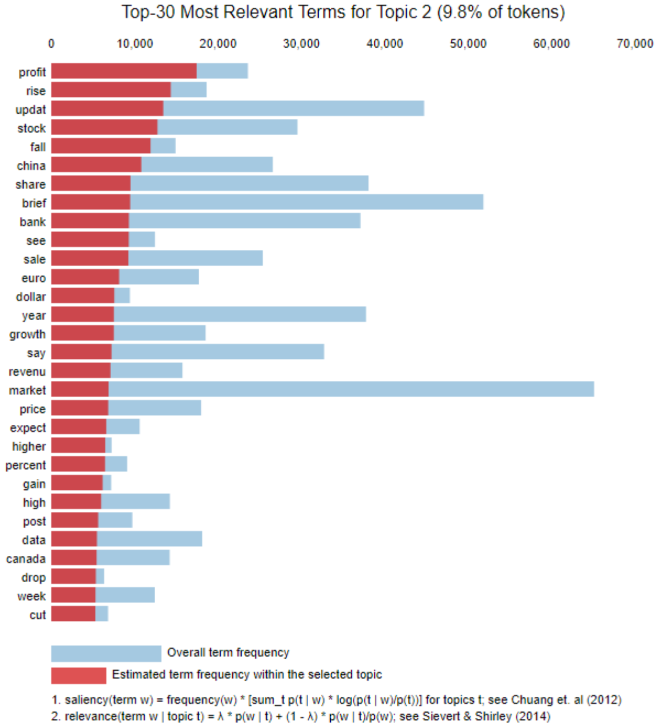


Figure 5.6: Sample visualization of LDA results using pyLDAvis

Extrinsic evaluation computes the quality of the model based on its performance in an external application. The approach that we used to perform extrinsic evaluation was to use the topic distribution probabilities in the news headlines to train a neural network which predicts the change in per day’s stock price for a test set. In addition to using topic distribution probabilities, we also used the sentiment polarity scores while creating the feature vectors. The best accuracy we get is 56.82% and it is achieved from the model with 15 topics and the parameters mentioned below.

The neural network built using Keras Tensorflow model had the following parameters:

1) 2 hidden layers, where the first layer had 500 nodes and the second layer had 10 nodes.

2) ‘relu’ activation function was used at the hidden layers, and ‘sigmoid’ activation function was used at the output layer.

3) Binary cross entropy was chosen as the loss function and accuracy as the metric function, in addition to ‘Adam’ optimizer.

By taking a closer look at the evaluation scores (both intrinsic as well as extrinsic), we see that the model with 15 topics gives a good trade-off between coherence and prediction accuracy values. Hence, we decided to choose this model to compute the correlation matrix, which contains the pairwise correlation of topics. This matrix will be helpful in determining how much and what kind of effect the topics had on the stock prices.

We also generated a heat map [Fig. 5.7] to depict the correlation values through a system of color coding, using the seaborn data visualization library.

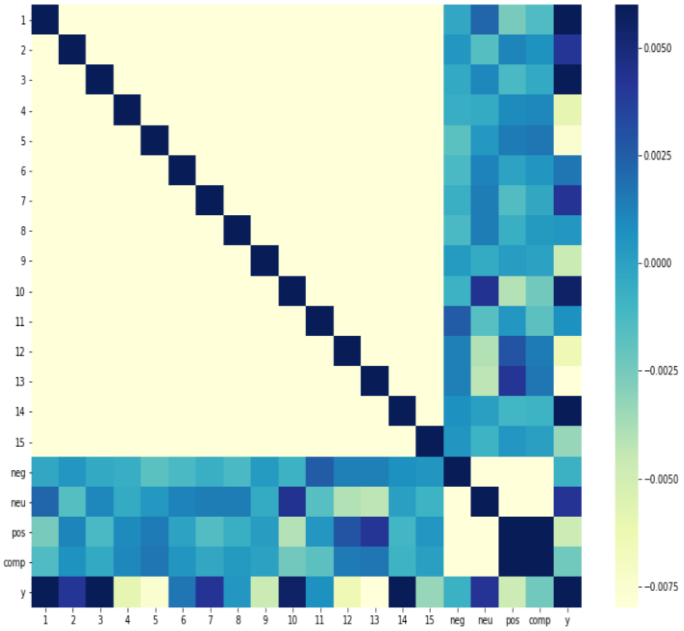
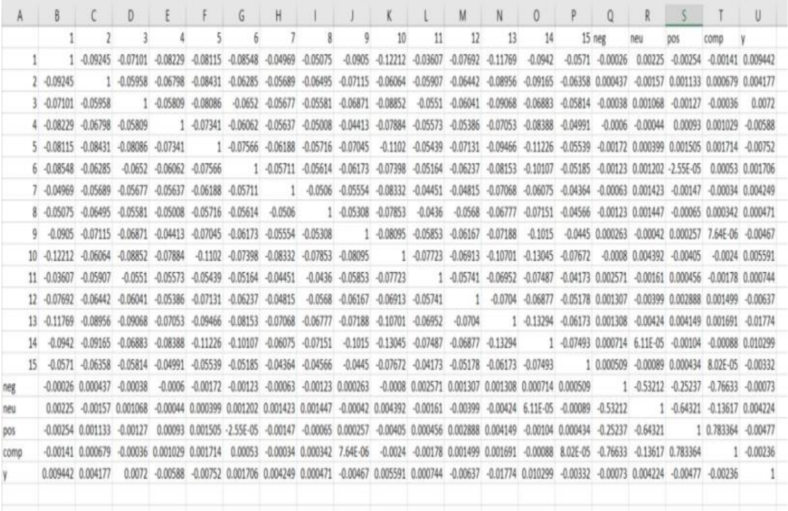


Figure 5.7*:* Heat map for correlation matrix with 15 topics

From figure 5.7, we see that the topics with darker colored boxes have higher positive correlation and the lighter colored boxes have higher negative correlation. Though heat maps are a good visualization to see and analyze the correlation data. The correlation matrix provides more precise results. The correlation matrix for 15 topics is as seen in figure 5.8.



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	neg	neu	pos	comp	y
1	1	-0.09245	-0.07101	-0.08229	-0.08115	-0.08548	-0.04969	-0.05075	-0.0905	-0.12212	-0.03607	-0.07692	-0.11769	-0.0942	-0.0571	-0.00026	0.00225	-0.00254	-0.00141	0.009442
2	-0.09245	1	-0.05958	-0.06798	-0.08431	-0.06285	-0.05689	-0.06495	-0.07115	-0.06064	-0.05907	-0.06442	-0.08956	-0.09165	-0.06358	0.000437	-0.00157	0.001133	0.000679	0.004177
3	-0.07101	-0.05958	1	-0.05809	-0.08086	-0.0652	-0.05677	-0.05581	-0.06871	-0.08852	-0.0551	-0.06041	-0.09068	-0.06883	-0.05814	-0.00038	0.001068	-0.00127	-0.00036	0.0072
4	-0.08229	-0.06798	-0.05809	1	-0.07341	-0.06062	-0.05637	-0.05008	-0.04413	-0.07884	-0.05573	-0.05386	-0.07053	-0.08388	-0.04991	-0.0006	-0.00044	0.00093	0.001029	-0.00588
5	-0.08115	-0.08431	-0.08086	-0.07341	1	-0.07566	-0.06188	-0.05716	-0.07045	-0.1102	-0.05439	-0.07131	-0.09466	-0.11226	-0.05539	-0.00172	0.000399	0.001505	0.001714	-0.00752
6	-0.08548	-0.06285	-0.0652	-0.06062	-0.07566	1	-0.05711	-0.05614	-0.06173	-0.07398	-0.05164	-0.06237	-0.08153	-0.10107	-0.05185	-0.00123	0.001202	-2.55E-05	0.00053	0.001706
7	-0.04969	-0.05689	-0.05677	-0.05637	-0.06188	-0.05711	1	-0.0506	-0.05554	-0.08332	-0.04451	-0.04815	-0.07068	-0.06075	-0.04364	-0.00063	0.001423	-0.00147	-0.00034	0.004249
8	-0.05075	-0.06495	-0.05581	-0.05008	-0.05716	-0.05614	-0.0506	1	-0.05308	-0.07853	-0.0436	-0.0568	-0.06777	-0.07151	-0.04566	-0.00123	0.001447	-0.00065	0.000342	0.000471
9	-0.0905	-0.07115	-0.06871	-0.04413	-0.07045	-0.06173	-0.05554	-0.05308	1	-0.08095	-0.05853	-0.06167	-0.07188	-0.1015	-0.0445	0.000263	-0.00042	0.000257	7.64E-06	-0.00467
10	-0.12212	-0.06064	-0.08852	-0.07884	-0.1102	-0.07398	-0.08332	-0.07853	-0.08095	1	-0.07723	-0.06913	-0.10701	-0.13045	-0.07672	-0.0008	0.004392	-0.00405	-0.0024	0.005591
11	-0.03607	-0.05907	-0.0551	-0.05573	-0.05439	-0.05164	-0.04451	-0.0436	-0.05853	-0.07723	1	-0.05741	-0.06952	-0.07487	-0.04173	0.002571	-0.00161	0.000456	-0.00178	0.000744
12	-0.07692	-0.06442	-0.06041	-0.05386	-0.07131	-0.06237	-0.04815	-0.0568	-0.06167	-0.06913	-0.05741	1	-0.0704	-0.06877	-0.05178	0.001307	-0.00399	0.002888	0.001499	-0.00637
13	-0.11769	-0.08956	-0.09068	-0.07053	-0.09466	-0.08153	-0.07068	-0.06777	-0.07188	-0.10701	-0.06952	-0.0704	1	-0.13294	-0.06173	0.001308	-0.00424	0.004149	0.001691	-0.01774
14	-0.0942	-0.09165	-0.06883	-0.08388	-0.11226	-0.10107	-0.06075	-0.07151	-0.1015	-0.13045	-0.07487	-0.06877	-0.13294	1	-0.07493	0.000714	6.11E-05	-0.00104	-0.00088	0.010299
15	-0.0571	-0.06358	-0.05814	-0.04991	-0.05539	-0.05185	-0.04364	-0.04566	-0.0445	-0.07672	-0.04173	-0.05178	-0.06173	-0.07493	1	0.000509	-0.00089	0.000434	8.02E-05	-0.00332
neg	-0.00026	0.000437	-0.00038	-0.0006	-0.00172	-0.00123	-0.00063	-0.00123	0.000263	-0.0008	0.002571	0.001307	0.001308	0.000714	0.000509	1	-0.53212	-0.25237	-0.76633	-0.00073
neu	0.00225	-0.00157	0.001068	-0.00044	0.000399	0.001202	0.001423	0.001447	-0.00042	0.004392	-0.00161	-0.00399	-0.00424	6.11E-05	-0.00089	-0.53212	1	-0.64321	-0.13617	0.004224
pos	-0.00254	0.001133	-0.00127	0.00093	0.001505	-2.55E-05	-0.00147	-0.00065	0.000257	-0.00405	0.000456	0.002888	0.004149	-0.00104	0.000434	-0.25237	-0.64321	1	0.783364	-0.00477
comp	-0.00141	0.000679	-0.00036	0.001029	0.001714	0.00053	-0.00034	0.000342	7.64E-06	-0.0024	-0.00178	0.001499	0.001691	-0.00088	8.02E-05	-0.76633	-0.13617	0.783364	1	-0.00236
y	0.009442	0.004177	0.0072	-0.00588	-0.00752	0.001706	0.004249	0.000471	-0.00467	0.005591	0.000744	-0.00637	-0.01774	0.010299	-0.00332	-0.00073	0.004224	-0.00477	-0.00236	1

Figure 5.8: Correlation matrix for 15 topics

VI. RESULTS

The baseline model evaluation results are:

Number of topics	Perplexity	U_mass Coherence	Accuracy of prediction models	Average topic coherence
15	-8.50439	-7.24818	0.212351	-7.2482

Table 5.1: Evaluation measures for baseline model

The final score for the below two topic clusters were found to be +0.046569 and -0.034711 respectively, thus indicating that these topics found in the headlines tend to have the most effect on the change of the stock price, Topic 10 had the most positive effect and Topic 11 had the most negative effect.

Topic: 10 Words: 0.017*"updat" + 0.016*"stock" + 0.015*"profit" + 0.013*"rise" + 0.012*"fall" + 0.010*"share" + 0.010*"see" + 0.008*"market" + 0.008*"euro" + 0.007*"sale"
Topic: 11 Word: 0.006*"technolog" + 0.006*"award" + 0.006*"launch" + 0.005*"servic" + 0.005*"announc" + 0.005*"name" + 0.005*"solut" + 0.005*"mobil" + 0.005*"manag" + 0.005*"busi"

Taking a closer look, we see that these results are justifiable. The topics that contain words such as “gain”, “share”, “market”, “profit”, “rise”, have the maximum impact on stock prices.

The evaluation results for model using topic distribution and sentiment scores are:

Number of topics	Perplexity	U_mass Coherence	Accuracy of prediction models	Average topic coherence
15	-10.1254	-6.16215	0.5682167	-9.243

Table 5.2: Evaluation measures for model using topic distribution and sentiment scores

From the correlation table we have generated for 15 topics as seen in Fig. 5.8, we see that topic 14 has the most positive correlation with the change in stock prices and topic 13 has the most negative correlation with the change in stock prices. Below are the clusters 13 and 14. The scores for topics 13 and 14 are -0.0177 and 0.0103, indicating negative and positive effect.

Topic: 13 Word: 0.006*"hong" + 0.006*"kong" + 0.005*"output" + 0.004*"updat" + 0.004*"econom" + 0.004*"china" + 0.004*"diari" + 0.004*"event" + 0.004*"notif" + 0.004*"brief"
Topic: 14 Word: 0.039*"rate" + 0.039*"variabl" + 0.024*"bank" + 0.016*"citibank" + 0.015*"deutsch" + 0.015*"mellon" + 0.013*"york" + 0.010*"trade" + 0.010*"iiroc" + 0.009*"ocrcvm"

Although, this information can just be used only for human judgement, the methodology to get these results help us in creating useful feature vectors as input to create better classification models to use supervised learning method to predict the future stock prices.
We also notice that creating feature vectors for each headline with the combination of topic probability distribution and sentiment polarity score improves the accuracy of the classification model used in our extrinsic evaluation. This informs us that finding out the important topics and the effect it has on the stock prices helps in better prediction of effect the headlines have on stock market prices.

Also, we see that sub-setting data by filtering out headlines with change < 0.5 helps eliminate volatility of data to certain extent and improves accuracy of the model to 56.82% from 21.23%. This also is a direct effect of optimizing the hyperparameters of the Neural Networks model used for classification.

VII. Future work

The experiments that we performed use the news headlines directly from the Reuters data set. One possible improvement that can be done during the preprocessing step is to filter out the noisy headlines. Since we had a huge dataset, the LDA topic modeling was very time consuming. So, we decided to implement LDA with 3 passes. The results obtained from topic modelling could possibly be improved upon by using more number of passes. Also, LDA topic model can be modified to capture how the structure of a document/news changes over time, in addition to the structure of the text.

References

[1] Heeyoung L, Mihai S, Bill M, Dan J, “On the Importance of Text Analysis for Stock Price Prediction”, 2014

[2] Kalyani J, Bharathi H. N, Jyothi R, “Stock Trend Prediction using news sentiment analysis”

[3] Ayman E. K, S. E. Salama, Nagwa Y, “Predicting Stock Market Behavior using Data Mining Technique and News Sentiment Analysis”, 2017.

[4] David M. Q, Adriano C. M, Renato A. O, “Stock Market’s Price Movement Prediction With LSTM Neural Networks”, 2017

[5] Carina Jacobi, Wouter van Atteveldt, Kasper Welbers, “Quantitative analysis of large amounts of journalistic texts using topic modelling”, 2015

[6] Liwei Yan, Bo Bai, “Correlated industries mining for Chinese financial news based on LDA trained with research reports”, 2016

[7] https://in.finance.yahoo.com/quote/%5EIXIC/history/

[8] https://github.com/philipperemy/Reuters-full-data-set

[9]https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24

[10] https://radimrehurek.com/gensim/models/ldamodel.html

[11]https://markroxor.github.io/gensim/static/notebooks/lda_training_tips.html

[12] https://www.tensorflow.org/guide/keras

[13] http://mallet.cs.umass.edu/diagnostics.php

[14]https://datascienceplus.com/evaluation-of-topic-modeling-topic-coherence

[15] https://pypi.org/project/pyLDAvis/

[16] https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation