Impact of Daily News on the Stock Market Movement

Team: We R

Kokila Dular

Chan Wu

Suvir Gupta

Raghavendra Singh Raghav

Vishal Page

Milind Jagre

Date: 4/26/2017

Table of Contents

Contents

Executive Summary

The main objective of this project is to perform an analytical study on the daily news for the last 10 years to come up with a relation between the impact of the daily news on the Dow Jones Industrial Average (DJIA) Index. As it is the general case that whenever something extraordinary thing happens, a lot of factors get affected because of it. Through this project, we are trying to prove the same, whether the daily news has any impact on the stock market or not.

For carrying out this analytical and predictive study, our team has used following major packages.

- tm
- SnowballC
- text2vec
- LDAvis
- topicmodels
- xts
- tseries

These five packages are applied on the three input CSV files provided on the Kaggle website. You can click here to land on the homepage of this project and can download the dataset by clicking here.

The objective of extracting this solid evidence from the above-mentioned packages are derived because each of the package mentioned above has so much to offer in its capacity.

Let us look at the description of each package in brief.

- **tm**: This package is used for doing the Text Mining. All the operations required for performing Text Mining operations are provided by this package.
- **SnowballC**: This package is used for stemming purposes in our project. The concept of stemming resembles to that of synonymous words. More will be explained about this package when we will talk about the Algorithms sections in this report.
- **text2vec**: This R package provides a concise API for doing the Text Analysis and Natural Language Processing (NLP)
- **LDAvis**: This R package is used for creating an interactive web-based visualization of a topic model that has been fit to a corpus of text data using Latent Dirichlet Allocation (LDA).
- **topicmodels**: This package is used for fitting the topic models.
- **xts:** This R package gives more functionality to the user by providing more customization and extension, while preserving the original cross-class interoperability.
- **tseries:** This R package is used for performing the time series analysis and computational finance.

- **glmnet:** This R package is used for performing the regression operations such as linear regression, logistic regression, and Cox models.

Since we are clear now about the role of each package in this project, let us explore other aspects of this projects. In the further sections, we will see what are all the operations we performed to over the constraints in this project.

Methodology

In this section, we will see the methodology we used for executing this project. This section is divided into following categories.

- Data Description
- Data Preprocessing
- Data Exploration
- Sentiment Analysis of the Daily Stock News

Let us dive into each subsection one by one.

Data Description

As already mentioned, the dataset for this project has been taken from Kaggle website. You can click here to land on this project's homepage and can click here to download this dataset.

If you look at this dataset, you will see that the dataset contains three CSV files.

We will look at the data in each file as follows.

- **Combined_News_DJIA.csv**
  This file contains a total of 27 columns, which are described as follows.

Column Name	Description
Date	The date
Label	Sentiment
Top1 … Top 25	The top 25 news of the day (corresponds to each date entry)

- **DJIA_table.csv**
  This file contains a total of 7 columns, which are described as follows.

Column Name	Description
Date	The date
Open	DJIA index at the opening
High	Highest value of the DJIA index
Low	Lowest value of the DJIA index

Close	Closing value of the DJIA index
Volume	Number of stocks involved in the day's transactions
Adj Close	Adjusted closing value of the DJIA index

- **RedditNews.csv**
  This file contains a total of 2 columns, which are described as follows.

Column Name	Description
Date	The date
News	News of the day

Data Preprocessing

This section deals with all the operations we performed as a part of Data Preprocessing. This includes the following parts.

- Missing Values
- Data Transformation

Let us look at each of these sections individually.

Missing Values

When it comes to imputing missing values, we generally follow the following techniques.

- Removing the records containing missing values
- Imputing the mean of the integer column in place of a missing value
- Imputing the mean of the category in case of multiple categories available for the data
- Imputing some defined term in place of a missing value in the categorical variable

Each of the above process has its advantages and disadvantages.

The given dataset on the Kaggle website does not contain any missing values, therefore we did not use any of the above-mentioned technique for minimizing the effect of missing values on the analysis.

Data Transformation

This sub-section deals with the transformation operations we performed for carrying out a task while doing the analysis.

The time series analysis contains the major portion of the data transformation part. Apart from this, the text mining part contains a small portion which is responsible for doing the preprocessing of the data.

Transformation using xts

As already mentioned, the package called **xts** is used for doing the customization to the variable in time series analysis. In this project, we are using the method called **xts()** in the **xts** package to convert the index values to not only unique and sequential, but also in a time-based format. This xts objects extends a class called **zoo** which is a part of the **xts** package.

The below snippet might come handy to understand the syntax of this data transformation.

```
13 # it is used for transforming the already existing DJIA_table data frame
14 # into a time-based class to use for the time series analysis
15 DJIA_table <- xts(DJIA_table[,-1],order.by = as.Date(DJIA_table$Date,"%Y-%m-%d"))
16
```

Date Format Conversion

The sentiment analysis part of the project required us to do some transformation over the date field. The input date field had the format **MM/DD/YYYY**, whereas we wanted to convert it into **YYYY-MM-DD** format**.** This was done with the help of strptime() method which is a part of the base R package.

The following snippet gives us a peek inside this method's implementation.

```
24 # converting the input DATE format into expected DATE format
25 #input DATE format = MM/DD/YYYY
26 #outout DATE format = YYYY-MM-DD
27 Combined_News_DJIA$Date_f<- as.Date(strptime(
28   Combined_News_DJIA$Date,'%m/%d/%Y'))
29
```

Stop Words Removal

This is one of the most important operations in the Text Mining operations for doing the sentiment analysis of the daily news for the stock market prediction. This operation removes the common used English words from the news columns to further improve the sentiment labeling.

The below snippet explains this stop words removal phenomenon in greater detail.

```r
library(tm)
# rm_words() function is used for removing the stop words
# below is the rm_words() function definition
rm_words <- function(string, words)
{
  stopifnot(is.character(string), is.character(words))
  splitted <- strsplit(string, " ", fixed = TRUE) # fixed = TRUE for speedup
  vapply(splitted, function(x) paste(x[!tolower(x) %in% words],
                                     collapse = " "), character(1))
}
# we call rm_words() function with input document and list of stop words
# these stop words are given in the tm package for English language
data$document = rm_words(data$document, tm::stopwords("en"))

```

Stemming the keywords

Apart from the stop words mentioned above, the news entries contain a lot of repetitive words in different forms. For example, the news may contain the words like “class” and “classes”, which in English language hold the same meaning, but when it comes to programming, those two words are considered differently and are counted as two individual words. This should not be the case in text mining, therefore we perform this stemming operation to equate these kinds of words. With the help of stemming, the program will become more intelligent to conclude that “class” and “classes” are not two different words, but two different forms of the same words. Therefore, with the help of stemming, the word “class” will be counted twice eliminating the presence of the word “classes”.

The following snippet provides an insight into this amazing concept.

```r
# creating a user defined function for stemming words
# passing x as an input and returning the refined
# list of words as an output
stem_tokenizer = function(x)
{
  token = word_tokenizer(x)  ## generate words documentwize
  return(lapply(token,SnowballC::wordStem,language="porter"))
}
# calling this stem_tokenizer() function
data$document =stem_tokenizer(data$document)

```

Package Information

This section focuses on explaining each package in details involved in this project.

Let us look at these packages one by one.

tm

As mentioned, this package is used for performing the text mining operations in R.

Various operations like data import, corpus handling, preprocessing, metadata management, and creation of term-document matrices are performed with the help of this tm package.

The main structure for managing documents in this package is called Corpus. A corpus is an abstract concept and it represents a collection of text documents.

There are two ways in which this Corpus can be implemented. The first one is VCorpus and PCorpus.

- VCorpus (Volatile Corpus): It is an acronym for Volatile Corpus. We create VCorpus for temporary purpose. If we are not bothered about accidental deletion of an object, then it makes sense to create VCorpus object. If you delete an VCorpus object, then the data is also deleted along with it.
- PCorpus (Permanent Corpus): PCorpus works differently as compared to VCorpus. It is an acronym of Permanent Corpus. There is a huge difference between VCorpus and PCorpus. The difference lies in the fact that PCorpus does not store the data in any object, instead of that, it does create a link to the database and whenever you access the PCorpus object, that database is accessed with the help of this pointer link. If you delete this PCorpus object, then only that object is deleted and your data in the database remains intact.

SnowballC

This package implements Porter's word stemming algorithm for collapsing words to a common root. This aid in a comparative vocabulary. Currently following are the languages supported by this package.

- Danish
- Dutch
- English
- Finnish
- French
- German
- Hungarian
- Italian,
- Norwegian
- Portuguese
- Romanian
- Russian
- Spanish

- Swedish
- Turkish

Let us look at the functionality of Porter's word stemming algorithm.

This principle works on the concepts of compiler design. Let me dig deep into this.

Consider the following representation.
A vowel -> v
A consonant -> c
Multiple vowels -> VVV
Multiple consonants -> CCC
Therefore, we can say that any word has the following four formats.
V…V
V…C
C…V
C…C
It can be transformed into the following.
[C]V…C…[V]
Above expression can be generalized into following format.
$[C](VC)^m[V]$

In above expression, the superscript m denotes the measure or number of occurrences of the collection of letters in a word.
This m can have any value greater than or equal to zero (>=0)
Now, actual stemming occurs with the help of some rules. You can consider these rules to be equivalent to the association rules in the Market Basket Analysis Algorithm and they do react in the same way. Let us take the following example to explain this.

The syntax of these rules is as follows.
(CONDITION) S1 -> S2
It translates into – If the CONDITION is satisfied, then the string S1 should be translated to S2.
This CONDITION may have any form and acts like a similar condition statement in an if..else block. This CONDITION can have the regular expressions, the OR condition, the AND condition and similar statements.
The CONDITION can be NULL as well. In that case, it gets transformed to
S1 -> S2
Let me give you an example.
Suppose S1=CARESSES and the rules are as follows.
SSES -> SS
IES -> I
SS -> SS
S ->
Then in the above case, the precedence is given to that rule which matches the most number of characters for the S1 word. In this case, S1 matches the most

with rule 1, therefore the first rule is applied and we get the partially stemmed word like follows.

CARESSES -> CARESS (SSES got replaced by SS)

Now, it is time for the third rule to apply, which translates CARESS to CARESS (SS got replaced by SS)

Once this rule is exhausted, we move on to the final rule, i.e. rule number 4. It translates CARESS to CARES and furthermore, CARES to CARE. (replaces S by NULL string twice)

Ultimately, CARESSES got stemmed into a new word – "CARE".

This is how the Porter's word stemming algorithm works in SnowballC package.

text2vec

This package provides a concise API for implementing text analysis and Natural Language Processing (NLP).

Following goals are achieved with the help of text2vec API.

- Concise: explore as few functions as possible
- Consistent: explore unified interfaces, no need to explore different package/interface for each task
- Flexible: some of the complex tasks can be easily solved with the help of text2vec package
- Fast: quite faster within a single thread. Multithreading is also supported
- Memory efficient: uses streams and iterators, therefore RAM usage is optimized

Some of the distinguishing factors about this package are as follows

- The author has written this package in C++, therefore it is quite memory friendly.
- Some of the parts, which were possible, have been implemented parallel with the help of RcppParallel package.
- It provides a streaming API, which means user does not need to load the entire data into RAM.

LDAvis

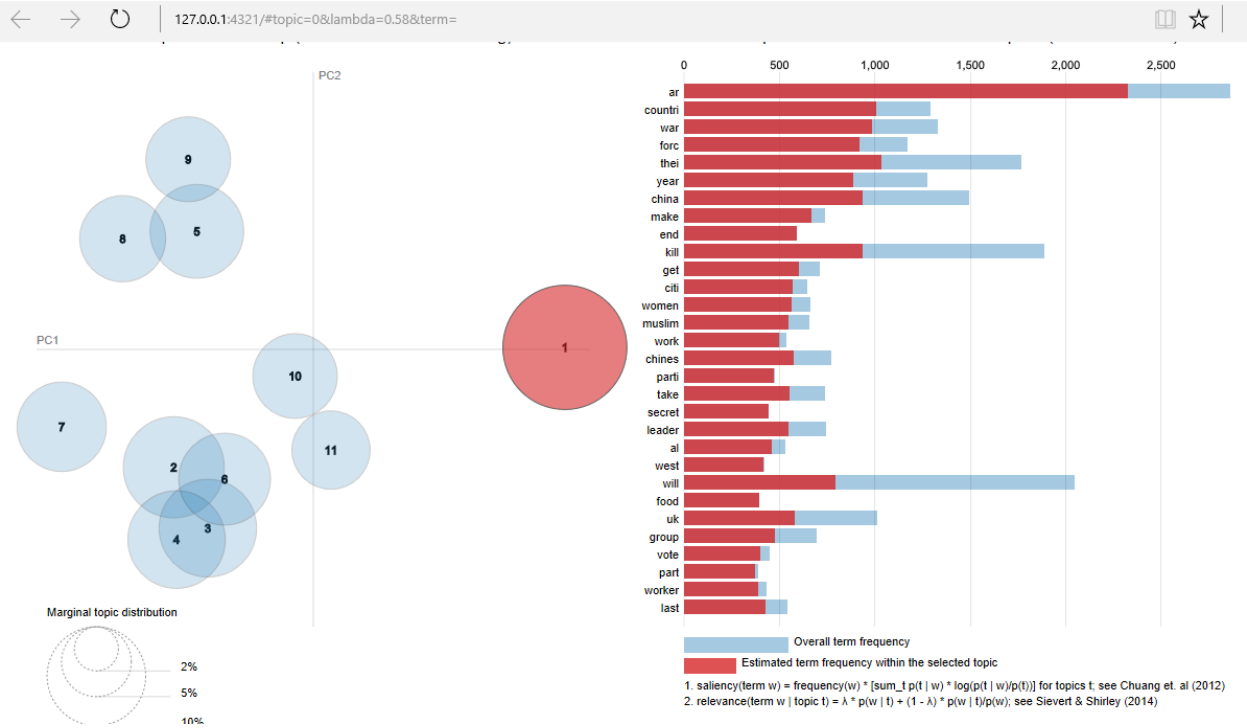
This R package is used for interactive model visualization.

This is a user-created package and can be found here.

It is used for showing the clustered information of the fitted data of the corpus in the web-based interactive application window.

The interactive dashboard is built with the help of a visualization tool d3.js. This dashboard can be accessed via a browser window. The goal is to help users interpret the topics in their LDA topic model.

A sample visualization dashboard is shown as in the following screenshot.



topicmodels

This R package provides an interface to the Latent Dirichlet Allocation (LDA) models and Correlated Topic Models (CTM).

Latent Dirichlet Allocation estimates a model using the VEM Algorithm or Gibbs Sampling.

Variation Expectation Maximization Algorithm is an iterative method to find the maximum likelihood estimates of parameters in statistical models. In this case, the model depends upon the unobserved latent variables. It means that the errors in this model is because of the missing values and the unobserved latent variables, causing the shift in the regression.

xts

This package is related to the Date format in R. It can be used for doing the conversion of the Date format. It gives the flexibility to the user to do the date formatting according to one's convenience. It extends another package called zoo.

tseries

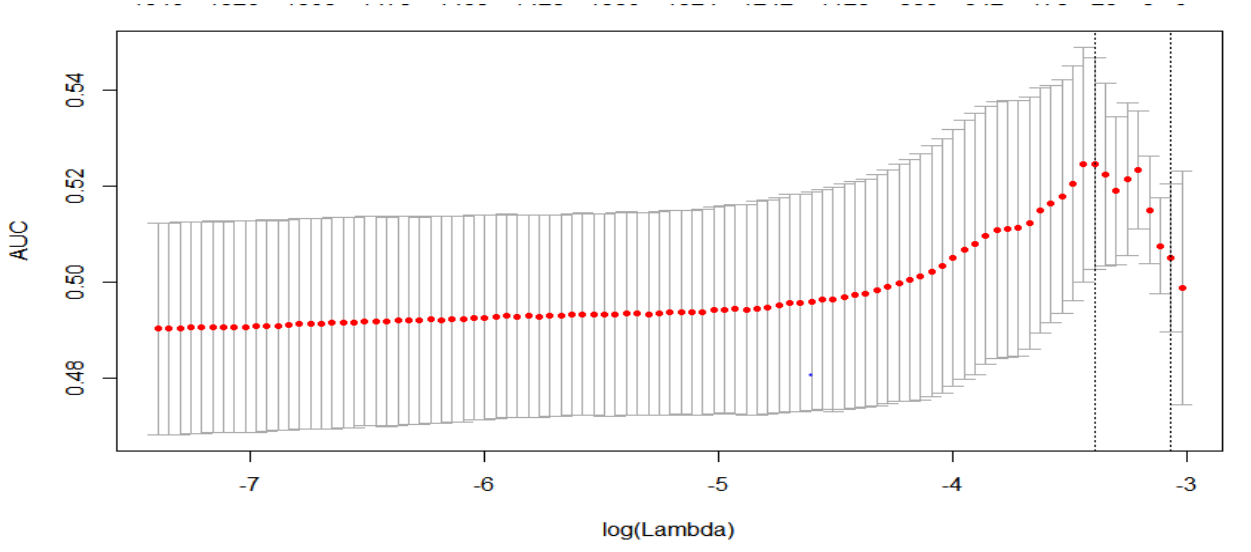
This R package is used for doing the time series analysis and computational finance. The plots related to the time series analysis in this project are done with the help of this package. The different types of models like ARMA and GARCH models. This package internally imports some of the other packages provided by R like graphics, stats, utils, quadprog, and zoo.

This completes the description of all the packages that we have used for this project to implement.

Modeling

After vectorization of the words, we have utilized Glmnet algorithm for model building. Glmnet fits a generalized linear model via penalized maximum likelihood. The regularization path is set up for lasso or elasticnet penalty at a grid of values for regulation parameter lamda. This algorithm is significantly fast and can be used to exploit sparsity in input matrix.

Cross validation feature is used and model is fitted, however the results are not promising as we able to achieve AUC of 0.53 which is little above the baseline.

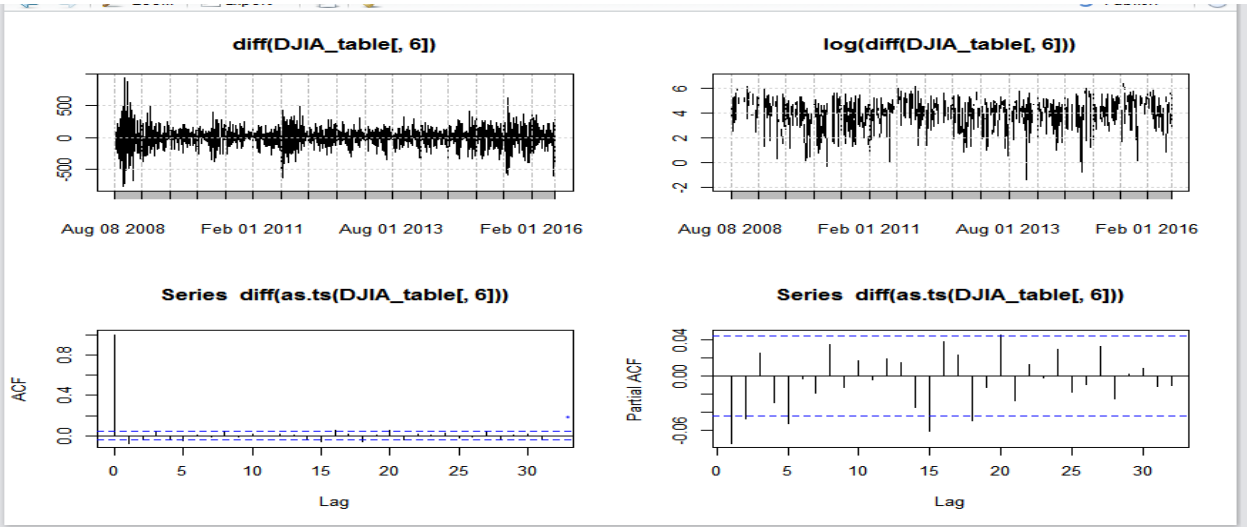


Glmnet Model AUC

We have utilized the time series to gain the insight about the stock market time line and try to intercept effect of news on the behavior of stock market.

We have built the ARIMA model , by first converting the daily index information into monthly.

Below are the model results and the forecasted pattern:



Forecasted Pattern

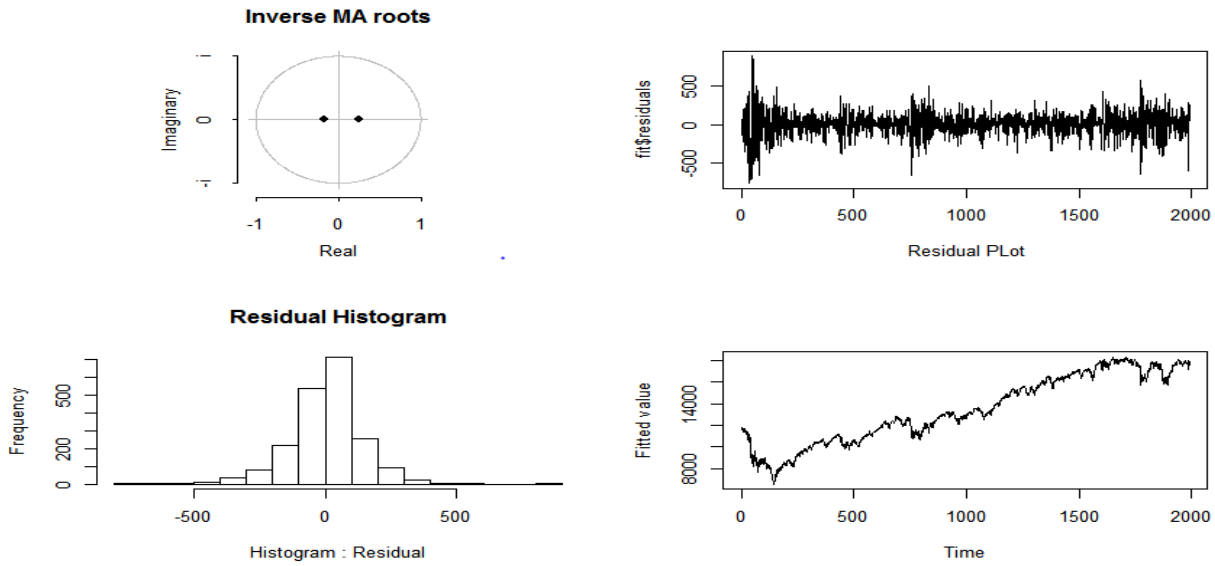
```
Series: DJIA_table[, 6]
ARIMA(0,1,2)

Coefficients:
          ma1      ma2
      -0.0758  -0.0406
s.e.   0.0225   0.0235

sigma^2 estimated as 20420:  log likelihood=-12684.58
AIC=25375.15   AICc=25375.16   BIC=25391.94

Training set error measures:
                    ME     RMSE      MAE         MPE      MAPE      MASE         ACF1
Training set 3.532459 142.7908 100.2725 0.01539275 0.8125963 0.9996305 -0.001576348
```

Error Measures

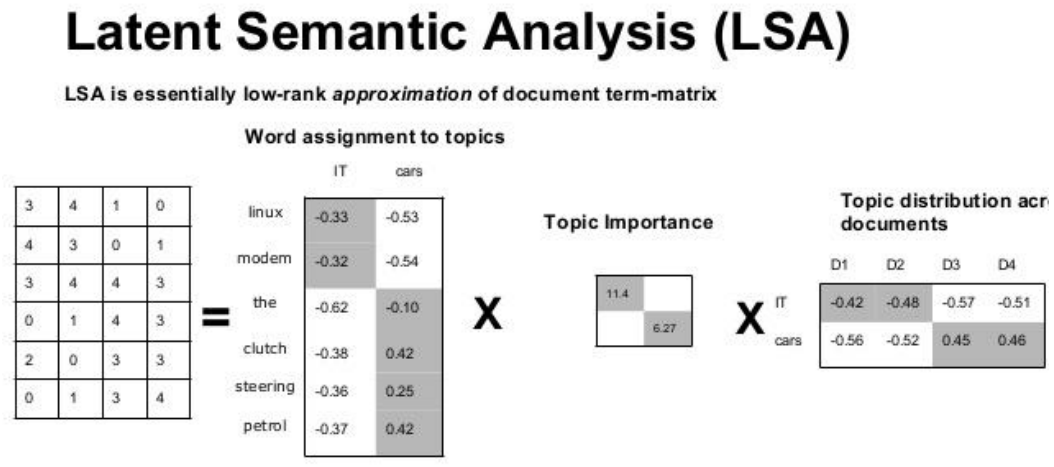


Modeling Graphs

Clustering

Latent semantic analysis(LSA): Count based Matrix model

It is basically division of high dimension sparse Document Term Matrix(DTM) to lower dimension dense vector matrix using SVD (single value decomposition).



Latent Semantic Analysis

1. As shown in the diagram we have matrix with topic distribution across document.
2. Words distribution across topics

Hence used for topic clustering an well as indexing of words to document and vice versa without need to store the high dimension DTM

```
### LSA MODEL

lsa = LatentSemanticAnalysis$new(15)
d1 = lsa$fit_transform(lda_dtm_train)
d2 =  lsa$fit_transform(t(as.matrix(lda_dtm_train)))
## finding index with the known words and getting weights for each topic
d1["2008-09",] # Document topic matrix
d2["financial_crisis",]  # Words topic matrix
```

Latent Dirichlet allocation(LDA): Probabilistic model

To learn the topic representation of each document and word associated with each topic.

Gibbs algorithm(implementaion)

1. Go throw each document and randomly assign words in the document to k topics
2. Random assignment already gives both topic representations of all the documents and word distributions of all the topics but needs improvement
3. To improve go through each word w in document and compute
   a. p(topic t | document d) = the proportion of words in document d that are currently assigned to topic t
   b. p(word w | topic t) = the proportion of assignments to topic t over all documents that come from this word w
   c. Reassign w a new topic, where we choose topic t with probability p(topic t | document d) * p(word w | topic t)
   d. In other words, in this step, we're assuming that all topic assignments except for the current word in question are correct, and then updating the assignment of the current word using our model of how documents are generated.
4. After repeating previous steps, we reach steady state where allocation are good.

```
## LDA MODEL
## Train LDA model to fit the words in corpus to set no of topics defined
lda_model =
  LatentDirichletAllocation$new(n_topics = 15, vocabulary = lda_vocab,
          doc_topic_prior = 0.1, topic_word_prior = 0.01)
## train document term matrix to allocate optimum no of topic percentage to documents
doc_topic_distr =
  lda_model$fit_transform(lda_dtm_train, n_iter = 1000, convergence_tol = 0.01,
                          check_convergence_every_n = 10)


install.packages("LDAvis")
library(LDAvis)     # vizualise the LDA model
x = lda_model$plot()
lda_model$get_word_vectors()

range(doc_topic_distr[,3])
doc_topic_distr[doc_topic_distr[,3]>950,3]
```

After generation of different Clusters from the corpus, we are able to identify the impacted event date based on the key word or sentiment.

Below are some of the results to verify the event date detection based on feature word:

```
> DTM_frame["financial_crisis",]
                X2008.08 X2008.09 X2008.10 X2008.11 X2008.12 X2009.01 X2009.02 X2009.03 X2009.04 X2009.05 X2009.06
financial_crisis        0        1        5        1        1        2        1        0        0        0        2
                X2009.07 X2009.08 X2009.09 X2009.10 X2009.11 X2009.12 X2010.01 X2010.02 X2010.03 X2010.04 X2010.05
financial_crisis        0        0        0        1        0        0        0        0        0        0        3
                X2010.06 X2010.07 X2010.08 X2010.09 X2010.10 X2010.11 X2010.12 X2011.01 X2011.02 X2011.03 X2011.04
financial_crisis        2        0        0        1        1        0        0        0        0        2        0
                X2011.05 X2011.06 X2011.07 X2011.08 X2011.09 X2011.10 X2011.11 X2011.12 X2012.01 X2012.02 X2012.03
financial_crisis        0        1        0        0        0        0        1        0        1        0        1
                X2012.04 X2012.05 X2012.06 X2012.07 X2012.08 X2012.09 X2012.10 X2012.11 X2012.12 X2013.01 X2013.02
financial_crisis        0        3        0        1        0        0        1        2        0        0        0
                X2013.03 X2013.04 X2013.05 X2013.06 X2013.07 X2013.08 X2013.09 X2013.10 X2013.11 X2013.12 X2014.01
financial_crisis        0        2        0        2        0        0        0        1        0        0        0
                X2014.02 X2014.03 X2014.04 X2014.05 X2014.06 X2014.07 X2014.08 X2014.09 X2014.10 X2014.11 X2014.12
financial_crisis        0        0        1        1        1        0        0        0        1        0        1
                X2015.01 X2015.02 X2015.03 X2015.04 X2015.05 X2015.06 X2015.07 X2015.08 X2015.09 X2015.10 X2015.11
financial_crisis        0        1        0        0        1        1        0        0        1        2        0
                X2015.12 X2016.01 X2016.02 X2016.03 X2016.04 X2016.05 X2016.06 X2016.07
financial_crisis        0        0        0        0        1        0        0        0
```

Financial crisis happened in 2008 , is successfully detected by algorithm.

```
> DTM_frame["referendum",]
           X2008.08 X2008.09 X2008.10 X2008.11 X2008.12 X2009.01 X2009.02 X2009.03 X2009.04 X2009.05 X2009.06 X2009.07
referendum        0        0        0        1        0        0        0        0        0        0        0        3
           X2009.08 X2009.09 X2009.10 X2009.11 X2009.12 X2010.01 X2010.02 X2010.03 X2010.04 X2010.05 X2010.06 X2010.07
referendum        2        0        1        1        1        0        0        5        0        0        0        0
           X2010.08 X2010.09 X2010.10 X2010.11 X2010.12 X2011.01 X2011.02 X2011.03 X2011.04 X2011.05 X2011.06 X2011.07
referendum        1        0        0        1        1        3        0        1        1        3        1        2
           X2011.08 X2011.09 X2011.10 X2011.11 X2011.12 X2012.01 X2012.02 X2012.03 X2012.04 X2012.05 X2012.06 X2012.07
referendum        1        0        1        7        0        1        2        0        0        0        1        1
           X2012.08 X2012.09 X2012.10 X2012.11 X2012.12 X2013.01 X2013.02 X2013.03 X2013.04 X2013.05 X2013.06 X2013.07
referendum        1        0        1        2        0        0        0        1        0        0        0        0
           X2013.08 X2013.09 X2013.10 X2013.11 X2013.12 X2014.01 X2014.02 X2014.03 X2014.04 X2014.05 X2014.06 X2014.07
referendum        1        0        1        0        1        0        0        5        4        3        2        1
           X2014.08 X2014.09 X2014.10 X2014.11 X2014.12 X2015.01 X2015.02 X2015.03 X2015.04 X2015.05 X2015.06 X2015.07
referendum        0        7        0        1        1        1        0        0        0        1        1        1
           X2015.08 X2015.09 X2015.10 X2015.11 X2015.12 X2016.01 X2016.02 X2016.03 X2016.04 X2016.05 X2016.06 X2016.07
referendum        0        0        0        1        1        0        2        2        1        0       11        0
```

Referendum word came into surface during splitting of Britain from European union in mid – 2016 is also

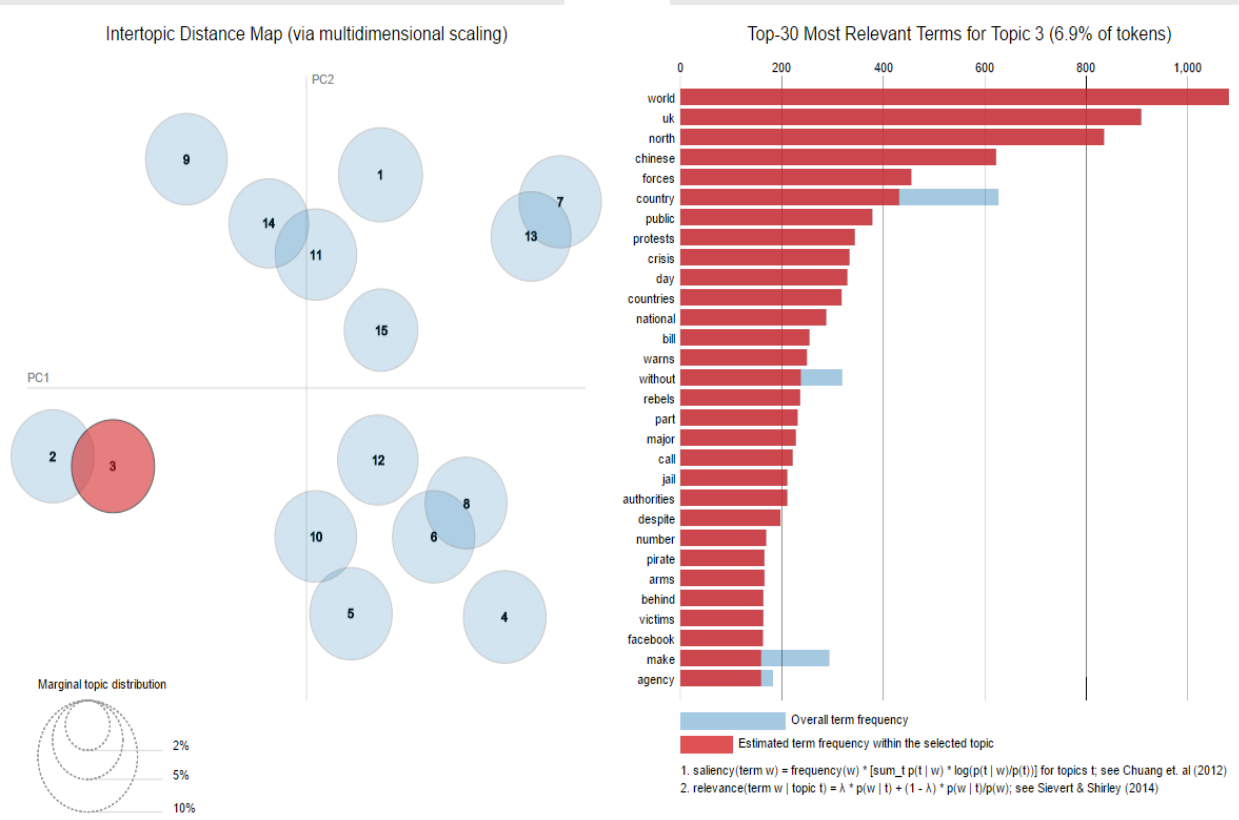
is detected.

```
> (DTM_frame["brexit",])
       X2008.08 X2008.09 X2008.10 X2008.11 X2008.12 X2009.01 X2009.02 X2009.03 X2009.04 X2009.05 X2009.06 X2009.07  7
brexit        0        0        0        0        0        0        0        0        0        0        0        0  3
       X2009.08 X2009.09 X2009.10 X2009.11 X2009.12 X2010.01 X2010.02 X2010.03 X2010.04 X2010.05 X2010.06 X2010.07  7
brexit        0        0        0        0        0        0        0        0        0        0        0        0  0
       X2010.08 X2010.09 X2010.10 X2010.11 X2010.12 X2011.01 X2011.02 X2011.03 X2011.04 X2011.05 X2011.06 X2011.07  7
brexit        0        0        0        0        0        0        0        0        0        0        0        0  2
       X2011.08 X2011.09 X2011.10 X2011.11 X2011.12 X2012.01 X2012.02 X2012.03 X2012.04 X2012.05 X2012.06 X2012.07  7
brexit        0        0        0        0        0        0        0        0        0        0        0        0  1
       X2012.08 X2012.09 X2012.10 X2012.11 X2012.12 X2013.01 X2013.02 X2013.03 X2013.04 X2013.05 X2013.06 X2013.07  7
brexit        0        0        0        0        0        0        0        0        0        0        0        0  0
       X2013.08 X2013.09 X2013.10 X2013.11 X2013.12 X2014.01 X2014.02 X2014.03 X2014.04 X2014.05 X2014.06 X2014.07  7
brexit        0        0        0        0        0        0        0        0        0        0        0        0  1
       X2014.08 X2014.09 X2014.10 X2014.11 X2014.12 X2015.01 X2015.02 X2015.03 X2015.04 X2015.05 X2015.06 X2015.07  7
brexit        0        0        0        0        0        0        0        0        0        0        0        0  1
       X2015.08 X2015.09 X2015.10 X2015.11 X2015.12 X2016.01 X2016.02 X2016.03 X2016.04 X2016.05 X2016.06 X2016.07  7
brexit        0        0        0        0        0        0        0        1        0        0       18        1  0
```

Similarly, Brexit new word came into surface during splitting of Britain from European union in mid – 2016 is also is detected, help us associate event with the time peroid.

```
> (DTM_frame["great_depression",])
                X2008.08 X2008.09 X2008.10 X2008.11 X2008.12 X2009.01 X2009.02 X2009.03 X2009.04 X2009.05 X2009.06 07
great_depression        0        0        0        0        0        0        0        0        1        0        0  0
                X2009.07 X2009.08 X2009.09 X2009.10 X2009.11 X2009.12 X2010.01 X2010.02 X2010.03 X2010.04 X2010.05 07
great_depression        0        0        0        0        0        0        0        0        0        0        0  0
                X2010.06 X2010.07 X2010.08 X2010.09 X2010.10 X2010.11 X2010.12 X2011.01 X2011.02 X2011.03 X2011.04 07
great_depression        0        1        0        0        0        0        0        0        0        0        0  0
                X2011.05 X2011.06 X2011.07 X2011.08 X2011.09 X2011.10 X2011.11 X2011.12 X2012.01 X2012.02 X2012.03 07
great_depression        0        1        0        0        0        0        0        0        0        0        0  0
                X2012.04 X2012.05 X2012.06 X2012.07 X2012.08 X2012.09 X2012.10 X2012.11 X2012.12 X2013.01 X2013.02 07
great_depression        0        0        0        0        0        0        0        0        0        0        0  1
                X2013.03 X2013.04 X2013.05 X2013.06 X2013.07 X2013.08 X2013.09 X2013.10 X2013.11 X2013.12 X2014.01 07
great_depression        0        0        0        0        0        0        0        0        0        0        0  0
                X2014.02 X2014.03 X2014.04 X2014.05 X2014.06 X2014.07 X2014.08 X2014.09 X2014.10 X2014.11 X2014.12 07
great_depression        0        0        0        0        0        0        0        0        0        0        0  1
                X2015.01 X2015.02 X2015.03 X2015.04 X2015.05 X2015.06 X2015.07 X2015.08 X2015.09 X2015.10 X2015.11 07
great_depression        0        0        0        0        0        0        0        0        0        0        0  0
                X2015.12 X2016.01 X2016.02 X2016.03 X2016.04 X2016.05 X2016.06 X2016.07
great_depression        0        0        0        0        0        0        0        0
```

Great depression word search, help us to identify period during which one of the biggest fall out of Dow jones index in the history occurred.



Key words associated with clusters , graph show one of such cluster and its associated words and their importance / weight.

Performance Improvements

The time taken to execute some of the operations was optimized using different techniques. While performing analytics, saving at least few seconds for a simple operation scales up to few minutes of the server time, which makes a huge difference when it comes to performance.

One of the techniques that is used for improving the performance of statistical operations is the usage of data.table package. This package contains a function called setKey() which is used for fast indexing. Therefore, the results generated by this function are extracted in reduced time as compared to the other operations.

setKey() function works in the same way as the Map which contains the data in key and value pair. As soon as the keys are involved in any object, the execution time for that object is reduced multifold giving room for further improvement.

The following screenshot will give you an idea about the implementation of this command.

```
94  # use data.table fuction for fast indexing hence faster
95  #results compared to traditional subseting
96  data<- setkey(data,Date_f)
```

This is how we improved the performance to some extent during the project execution.

Conclusion

In conclusion, we would like to mention the following things.

- The data was subjected to exploratory analysis and necessary modifications were made based on the exploratory analysis.
- Lot of cleaning of the data is being performed to derive clear meaning full words by removing the stop words, spaces, punctuations etc.
- Text2vec algorithm found to be promising algorithm to perform text mining as make prediction/inferences based the data. We have performed Time series analysis as well to gain insight about the trend of the stock market based on the news.
- The model not able to predict with good accuracy, this is the corpus and irrelevant information in the news in relation with stock market. As all news not have similar effect impact on stock market and sometime not even have an impact.
- To gain more depth, we also performed topic modeling, idea is to figure cluster of words which have direct effect on stock market trend. So that we have utilize these information for the better prediction accuracy.

  Finally, we can conclude that use of topic modeling with the tet2vec algorithm, we able to predict behavior of market in much better way.

Appendix

Packages used

- tm
- SnowballC
- text2vec
- LDAvis
- topicmodels
- xts
- tseries
- glmnet

Project files included

- STEX-PRED code version 3.0.0.0.R
- DJIA_table.csv
- Combined_News_DJIA.csv

Project Usage Guidelines

- Copy ***DJIA_table.csv*** and ***Combined_News_DJIA.csv*** into one folder and note down the location of that folder.
- Do corresponding directory path changes in ***STEX-PRED code version 3.0.0.0.R***
- Run ***STEX-PRED code version 3.0.0.0.R***
- Observe the output

References

- Kaggle competition landing page
- Kaggle dataset download link
- text2vec package documentation
- LDAvis package documentation
- SnowballC package documentation
- tm package documentation
- topicmodels package documentation
- Porter's word stemming algorithm
- LDAvis package at GitHub
- VEM Algorithm
- VEM Algorithm Wikipedia