Mini Project with Seminar

On

STOCK MARKET SENTIMENT ANALYSIS

Submitted in partial fulfillment of the requirements for the award of the

Bachelor of Technology

In

Department of Computer Science and Engineering

By

R. Sai Ramana	**19241A05G2**
R.Revanth	**19241A05G3**
P. Pranay Prasad	**19241A05F6**
B. Harinath	**19241A05C5**

Under the Esteemed guidance of

D. Swathi

Assistant Professor

Department of Computer Science and Engineering

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Bachupally, Kukatpally, Hyderabad-500090

Department of Computer Science and Engineering

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Bachupally, Kukatpally, Hyderabad-500090

CERTIFICATE

This is to certify that the mini project entitled "**STOCK MARKET SENTIMENT ANALYSIS**" is submitted by **R. Sai Ramana (19241A05G2), R. Revanth (19241A05G3), P. Pranay Prasad (19241A05F6),B. Harinath(19241A05C5)** in partial fulfillment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering during academic year 2021-2022.

INTERNAL GUIDE
D. Swathi
Assistant Professor

HEAD OF THE DEPARTMENT
Dr. K. MADHAVI

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We take the immense pleasure in expressing gratitude to our Internal guide**,** **D.Swathi,** Assistant Professor**,** dept of CSE, GRIET. We express our sincere thanks for her encouragement, suggestions and support, which provided the impetus and paved the way for the successful completion of the project work.

We wish to express our gratitude to **Dr.K.Madhavi**, Head of the Department**,** our Project Co-coordinator **Rubeena Rustum**, Assistant Professor for their constant support during the project.

We express our sincere thanks to **Dr. Jandhyala N Murthy,** Director, GRIET**,** and **Dr. J. Praveen,** Principal, GRIET**,** for providing us the conductive environment for carrying through our academic schedules and project with ease. We also take this opportunity to convey our sincere thanks to the teaching and non
teaching staff of GRIET College, Hyderabad.

R. Sai Ramana (19241A05G2)

R. Revanth (19241A05G3)

P. Pranay Prasad (19241A05F6)

B. Harinath(19241A05C5)

DECLARATION

We hereby declare that the industrial mini project entitled **"Stock Market Sentiment Analysis"** is the work done during the period from **17th January 2022 to 1st May 2022** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad).The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

R. Sai Ramana (19241A05G2)

R. Revanth (19241A05G3)

P. Pranay Prasad (19241A05F6)

B. Harinath(19241A05C5)

ABSTRACT

Stock prices and financial markets are frequently influenced by public mood, leading to research efforts to forecast stock market trends based on popular sentiment and media conveyed on social media platforms such as Facebook and Twitter.

Here we look into the effect of sentiment on the stock market and use Machine Learning to make predictions. Specifically, we use a set of text featurization and machine learning techniques to analyze the Top 25 News Headlines and extract financial sentiment. The relationship between daily emotion and stock price change is then investigated.
Various news headlines are included in the data set. Using news headlines, we'll train a model for stock sentiment analysis. Using news headlines, this model evaluates whether the stock price has increased or decreased. This is a project based on Natural Language Processing and Machine Learning. Sentiment analysis has now become the most popular method for obtaining sentiment and ratings from web sources. To assign sentiment scores to the categories within a sentence, the Sentiment Analysis system for text analysis combines NLP and ML techniques.

The project's purpose is to forecast the expected price of a stock by combining historical stock data and sentiment analysis of news headlines. Lemmatization and Count vectorization are used. Words are reduced to a normalized form via lemmatization. Textual data is utilized in deep learning and machine learning models such as textual classification thanks to the Count vectorizer. Various classifiers are used for classification purpose.

TABLE OF CONTENTS

Chapter-1

INTRODUCTION

1.1 Rationale

- Several factors influence market sentiment in the stock market, including news of various genres and social media. Trading volume, firm income, and stock market volatility are all affected by these factors.
- This can be a great tool for investors with which they can utilize sentiment analysis and machine learning to to figure out the period where the market is driven by the emotion not by rational decision-making. They can analyze and can observe the shifts in sentiment if there is news to explain stock price movements.
- We look into the impact of sentiment on the stock market and use Machine Learning to make predictions.

1.2 Goal

- The project's goal is to forecast the expected price of a stock by combining existing stock data and news headlines with the help of sentiment analysis.
- We can predict the trend of the stock market.
- Likewise, we can use the same process to predict the future performance of a company which it can help them to formulate new policies according to their performance.

1.3 Existing Systems

- In the Earlier Methods, the Feature Extraction was done using basic text preprocessing and random news Labelling.

-Existing Methods were able to achieve 50-60 % accuracy using those Methods.

SVM

A supervised learning approach is a support vector machine (SVM), which is sometimes referred to as a help vector network (SVN). Supervised learning is an artificial intelligence method in which a computer set of rules is trained on data that has been classified for output. It inputs a known set of input statistics (the learning set) and familiar reactions to the statistics (the output), and constructs a model to make sensible predictions for reaction to the new input statistics. When you have current statistics for the outcome you're trying to anticipate, use supervised learning.

-This SVM has the disadvantage of not being suitable for very big data sets. When data sets contain more noise, SVM does not produce accurate findings. There is no statistical justification for the classification because the support vector classifier works by positioning facts points above and below the classifying hyperplane.

- In the Proposed Model, NLP Techniques such as Lemmatization,Count Vectorization are applied, which are giving better results comoared to existing results.

Dataset

- The data set under consideration is a mix of global news and stock price fluctuations.
- The data was scraped from Yahoo Finance and covers the years 2008 through 2016.
- The data frame contains 25 columns of top news headlines for each day.

Class 0- the stock price stayed the same or decreased.
Class 1- the stock price increased.

	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	...	Top16	Top17	Top18	Top19
0	2000-01-03	0	A 'hindrance to operations': extracts from the...	Scorecard	Hughes' instant hit buoys Blues	Jack gets his skates on at ice-cold Alex	Chaos as Maracana builds up for United	Depleted Leicester prevail as Elliott spoils E...	Hungry Spurs sense rich pickings	Gunners so wide of an easy target	...	Flintoff injury piles on woe for England	Hunters threaten Jospin with new battle of the...	Kohl's successor drawn into scandal	The difference between men and women
1	2000-01-04	0	Scorecard	The best lake scene	Leader: German sleaze inquiry	Cheerio, boyo	The main recommendations	Has Cubie killed fees?	Has Cubie killed fees?	Has Cubie killed fees?	...	On the critical list	The timing of their lives	Dear doctor	Irish court halts IRA man's extradition to Nor...
2	2000-01-05	0	Coventry caught on counter by Flo	United's rivals on the road to Rio	Thatcher issues defence before trial by video	Police help Smith lay down the law at Everton	Tale of Trautmann bears two more retellings	England on the rack	Pakistan retaliate with call for video of Walsh	Cullinan continues his Cape monopoly	...	South Melbourne (Australia)	Necaxa (Mexico)	Real Madrid (Spain)	Raja Casablanca (Morocco)
3	2000-01-06	1	Pilgrim knows how to progress	Thatcher facing ban	McIlroy calls for Irish fighting spirit	Leicester bin stadium blueprint	United braced for Mexican wave	Auntie back in fashion, even if the dress look...	Shoaib appeal goes to the top	Hussain hurt by 'shambles' but lays blame on e...	...	Putin admits Yeltsin quit to give him a head s...	BBC worst hit as digital TV begins to bite	How much can you pay for...	Christmas glitches
4	2000-01-07	1	Hitches and Horlocks	Beckham off but United survive	Breast cancer screening	Alan Parker	Guardian readers: are you all whingers?	Hollywood Beyond	Ashes and diamonds	Whingers - a formidable minority	...	Most everywhere: UDIs	Most wanted: Chloe lunettes	Return of the cane 'completely off the agenda'	From Sleepy Hollow to Greeneland

1.4 Methodology

Count Vectorization

- It is a NLP Technique which is used to convert textual data into numerical vectors based on the number of occurances of each word in text data.
- Textual data can be utilized in deep learning applcations and machine learning models such as textual classification thanks to the Count vectorizer.
- The data will be tokenized and split into n-gram chunks, the length of which can be specified by giving a tuple to the n gram range parameter.

Lemmatization

- Lemmatization is one of the most used text pre-processing techniques.
- Words are reduced to a normalised form using lemmatization. This process uses a dictionary to map various versions of words to its root form.
- Search engines and chatbots employ lemmatization to figure out what a term means. The context in which the term is used is used in lemmatization.
- Lemmatization is the process of extracting a word's meaning from a source such as a dictionary.

Random Forest Classifier

- Random Forest is a simple and flexible machine learning algorithm.
- It is used for both classification and regression problems. Forest means many trees or group of many trees.
- In this approach, multiple trees, i.e, decision trees are created and the outputs of respective decision trees are used to reach a single result.

Naive Bayes Classifiers

- Another Bayes Theorem-based approach for estimating probabilities and conditional probabilities is Naive Bayes. In comparison to other classification techniques, it can be extremely fast.

Logistic Regression

- Logistic regression is a supervised learning algorithm. It can be used for classification purpose. The prediction of the probability of target variable is done by this classifier. The target or the dependent variable has only two classes. These variables are bifurcated.

1.5 Contribution

Innovativeness

The idea behind this work is that to develop a machine learning model that forecasts the stock Price based on the top 25 news headlines in our dataset. Here we are extracting the features from the News and training the model using Classification algorithms

Usefulness

- It is used in forecasting the future Stock Price.
- It can be used by companies to analyze their company's performance and formulate policies accordingly.
- We can investigate the impact of sentiment on stock Market and make predictions using Machine Learning.

1.6 Report Organization

The report's remaining sections are organised as follows:

- This project's technical requirements, analysis, and design are discussed in Chapter2.
- This project's construction and execution details are detailed in Chapter 3.
- The conclusion and scope of this project are presented in Chapter 4.

Chapter-2

SYSTEM ANALYSIS

2.1 Problem Statement

- In this work, we look into the effect of sentiment on Stock Market and make predictions using Machine Learning.
- We will classify whether the stocks of the company will go up or go down on the basis of the top 25 headlines about the company.
- The project's goal is to forecast the expected price of a stock by combining existing stock data and news headlines with the help of sentiment analysis and machine learning.

2.2 Functional Requirements

The functional requirement define the model's essential functionality. The functional requirements fundamentally state what a system is intended to do. They define the system's purpose and capabilities. These are gathered from users based on their needs.

These are gathered as functional requirements documents from clients, and developers work on implementing all of the features.

The following are the project's functional requirements:

- This model requires news headlines in textual format. There should be top 25 new Headlines in order to train the model and extract features from the data

2.3 Non Functional Requirements

Non-functional requirements are those that are not directly related to the system's specific functionality. These are mostly concerned with project functionalities that are not included as fundamental functionalities. They could be linked to emergent traits like dependability and usability.

- Ease of use.
- Availability
- Reliability
- Maintainability

2.4 Software Details

- Operating system

  - Windows 10,11 can be used

- Coding language

  - Python 3.8 is used in this project.

  - Many useful analytics libraries are already installed in the Python 3 environment.

- Coding environment -

  - Anaconda is a distribution which supports Python and R programming language.

  -It is used in many machine learning applications, large-scale data processing. Majority of data Science projects are done in this distribution(spyder and Jupyter notebooks)

  -It is used in predictive analytics. Anaconda distribution consists of Jupyter Notebooks application.

  - Jupyter Notebook is an open source web tool for creating and sharing documents with live code, equations, visualizations, and text.

  - Data visualisation, machine learning (ML), and statistical modelling are all part of the platform.

- Libraries

  - nltk is to be downloaded in jupyter notebooks.

2.5 Hardware Details

Minimum Requirements

- Processor

  - Intel i3 6^{th} gen and above

- Speed

  - 1.1 GHz and above

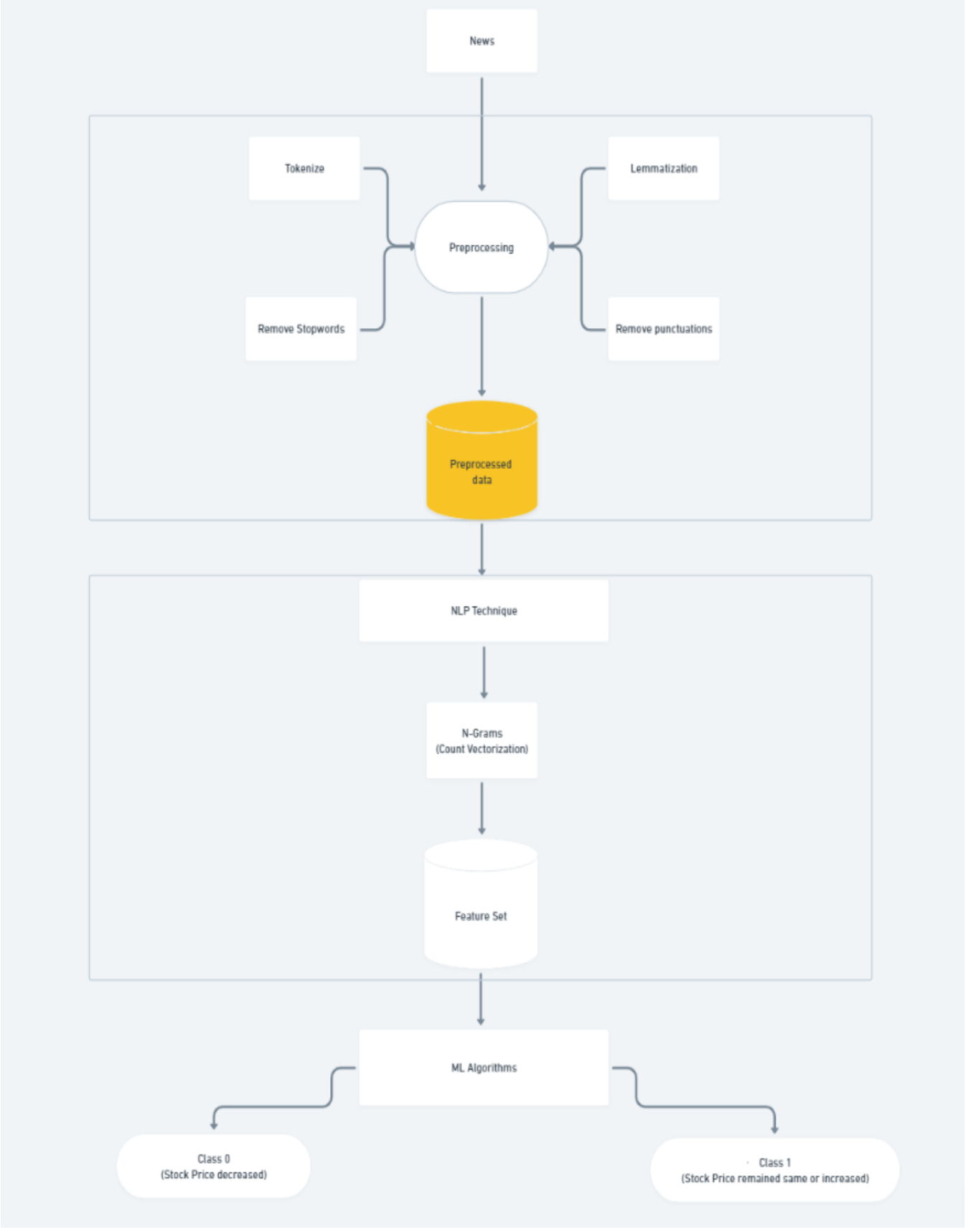
- RAM

  - 4GB(minimum),8GB(recommended)

- Display

  – 1280*720 pixels

2.6 Architecture



Chapter 3
Implementation

3.1 Importing the required Libraries

The following Libraries are used in order to implement the model

- Numpy
- Pandas
- WordNetLemmatizer (nltk.stem)
- stopwords (nltk.corpus)

Sklearn Libraries

- Count vectorizer(feature_extraction.text)
- Random Forest Classifier
- Multinomial NB(Naive Bayes)
- Logistic Regression
- classification report
- confusion matrix

Import all the required libraries

In [1]:

```
import numpy as np
import pandas as pd
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

3.2 Reading the dataset and splitting the data

- Using pandas function, read_csv dataset(which is in CSV format) can be read and data frame can be created.
- Encoding “ISO-8859-1” is used and passed as a parameter in read_csv() function.
- After reading the data, the data can be split into training and testing data based on the dates.

Read dataset

```
df=pd.read_csv('Data.csv', encoding = "ISO-8859-1")
```

Split the dataset into training and test data

```
df_train = df[df['Date'] < '20150101']

df_test = df[df['Date'] > '20141231']
```

3.3 Text Preprocessing and Feature Extraction

- **Check for Null Values**

  using isnull(), null fields can be identified and can be replaced by empty string.

```
#check for null values
df.isnull().sum()
```

```
df = df.replace(np.nan, ' ', regex=True)

#sanity check
df.isnull().sum().sum()
```

```
0
```

- **Removing Punctuations**

  From reviewing the data, we noticed that there are a lot of punctuations which would not contribute to interpreting the sentiment in the message.

  To replace everything except a-z and A-Z with blank space and put in-place in the "data" variable, we may use " replace("[a-z A-Z]," ",regex=True, inplace=True) ".

```
# function for cleaning the data
def clean_data(dataset):
    data = dataset.iloc[:,2:27]
    data.replace("[^a-zA-Z]", " ", regex=True, inplace=True)
    return data
```

- **Combining all the headlines to form a paragrapgh**

  Next, we'll combine the news headlines to a paragraph so that we can transform it to a vector and use it in NLP (NLP models usually involve working with vectors).

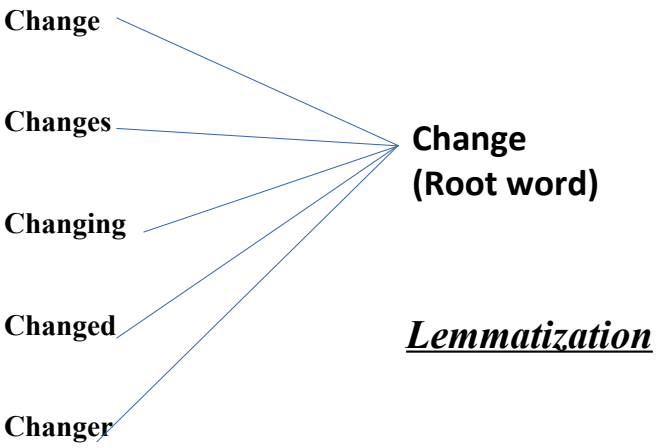
```
# function for combining the headlines of all the columns into single column
def combine_data(data):
    headlines = []
    for i in range(len(data.index)):
        headlines.append(' '.join(str(x) for x in data.iloc[i, :]))
    return headlines
```

- **Converting Upper Case to Lower Case**

  We can change all the characters to lower case because the case of letters in words has no effect on the sentiment.

- **Lemmatization**

- Lemmatization is mostly used in NLP(Natural Language Processing) and Machine Leaning.
- It is a text pre-processing technique which is frequently used.
- Words are reduced to a normalised form using lemmatization. This process uses a dictionary to map various various versions of words to its root form.
- Search engines and chatbots employ lemmatization to figure out what a term means. The context in which the term is used is used in lemmatization.
- Lemmatization is the process of extracting a word's meaning from a source such as a dictionary.
- When compared to their stemming counterparts, most lemmatization methods are slower.
- Lemmatization has a computing overhead as well.
- A lemma (plural lemmas or lemmata) is a group of words in their dictionary or canonical form.
- For example, because walks, walking, and walked are all versions of the word walk.
- Walk is the lemma of all these words,



- The only difference between lemmatization and stemming is, the final word in the lemmatization process is a meaningful. Stemming doesn't use kind of dictionary.
- In this context, news headlines consists of combination of many words which can be normalized and can be altered to its root form
- This helps in the vectorization of data where the frequency of words plays an important role

```python
# function to perform lemmatization of the word
def lemmatize_data(data, lemmatizer):
    cleaned_dataset = []
    for i in range(len(data)):
        clean_text = data[i].lower()
        clean_text = clean_text.split()
        clean_text = [lemmatizer.lemmatize(word) for word in clean_text if word not in stopwords.words('english')]
        cleaned_dataset.append(' '.join(clean_text))
    return cleaned_dataset
```

3.4 Count Vectorization

- It is a NLP Technique which is used to convert textual data into numerical vectors based on the number of occurances of each word in text data.
- CountVectorizer is a sophisticated tool for extracting and analyzing features from text data.

Data=[‘The ‘,’quick’,’black’,’rat’,’jumps’,’over’,’the’,’lazy’,’cat’]

The	Quick	black	rat	Jumps	Over	Lazy	cat
2	1	1	1	1	1	1	1

- Above picture is used to visualize the representation of words but actually in practice, these words are converted to numbers, which reflect the sparse matrix's positional index.
- CountVectorizer creates a sparse matrix (representation of words)

Create CountVectorizer object

```python
countvector = CountVectorizer(ngram_range=(2,2))
```

- The data will be tokenized and split into n-gram chunks, the length of which can be specified by giving a tuple to the n gram range parameter.
- For example, 1,1 gives us unigrams or 1-grams like "whey" and "protein," whereas 2,2 gives us bigrams or 2-grams like "whey protein."
- n gram range: An n-gram is simply a sequence of n words. The 2-grams 'I am' and 'am Hulk', for example, are found in the sentence 'I am Hulk.' The sentence is a three-gram unit in and of itself.
- n gram range=(a,b), a is the least size of n grams you wish to include in your features and b is the maximum size.
- The n gram is set to 2,2 in order to use pairs of words that appear together as features (columns) If we set (1, 1) to only unigrams (single words), (1, 2) to unigrams and bigrams (2 words), and (2, 2) to only bigrams, we get the following results.

3.5 Vectorize the data

- A step in feature extraction is vectorization. By transforming text data to numerical form specifically to numerical vectors, we can extract some observable features from the text and the model can learn from that accordingly
- On our training data, we use the fit transform() method, and on our test data, we use the transform() method.

  fit_transform ()

  - This is similar to fitting and then transforming, however it is more efficient.

  - It is used to the training data in order to scale the data and learn the scaling parameters.

  - Our test data is then scaled using the parameters we've learned.

  transform ()

  - Returns changed training data as output using the initial calculated values.

  - This method can be used to change a dataset using the same settings.

  - Pre-processing is done before modelling.

```
# function to vectorize the data
def vectorize_data(data, cv):
    vectorized_dataset = cv.fit_transform(data)
    return vectorized_dataset
```

Vectorize the data

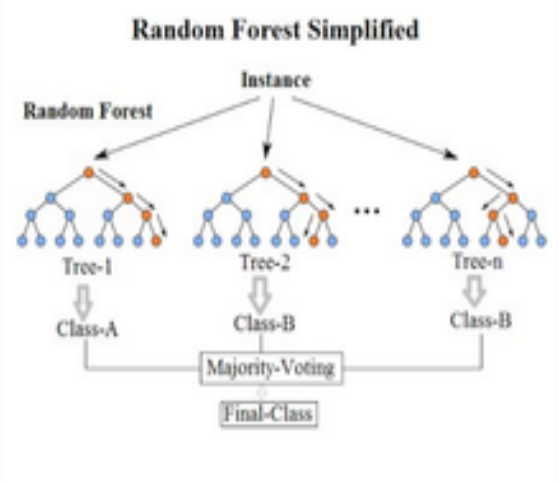
```
vec_train_data = vectorize_data(train_data, countvector)
vec_test_data = countvector.transform(test_data)
```

3.5

Classification

- **Random Forest Classifier**

Random Forest is a simple and flexible machine learning algorithm. It is used for both classification and regression problems. Forest means many trees or group of many trees. In this approach, multiple trees, i.e, decision trees are created and the outputs of respective decision trees are used to reach a single result. These trees are generated from training dataset. Majority Voting of predictions is done and final output is predicted. It is very consistent and generates good results. It can be used on larger volume of data with high dimensionality.



@image credits:google

```
# create classifier
rf_clf = RandomForestClassifier(n_estimators=200, criterion='entropy')
rf_clf.fit(vec_train_data, df_train['Label'])
```

n estimators: This represents the number of trees.The higher number of trees, the higher the performance, prevents the overfitting problem but computation is very heavy and execution time is more.

criterion: This is how the tree's nodes are chosen. There are two choices:
gini: the outcomes are marginally better.

entropy is significantly faster

We utilise the Random Forest Classifier because we want both speed and accuracy.

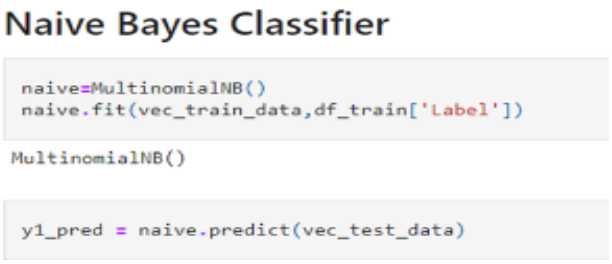
- **Naive Bayes Classifiers**

Another Bayes Theorem-based approach for estimating probabilities and conditional probabilities is Naive Bayes. In comparison to other classification techniques, it can be extremely fast.

- The Naive Bayes is a supervised learning technique for solving problems of classification which are based on Bayes theorem.
- It can be used when the datasets is of large-scale.

-It is widely used for solving the problems of classification.

-Face Recognition, Medical Diagnosis, Weather Prediction are some of the examples of Naive Bayes Algorithm.

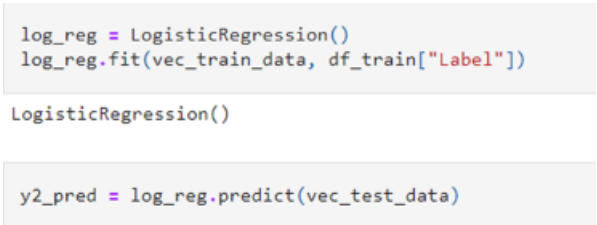


- **Logistic Regression**

Logistic regression is a supervised machine learning algorithm. It can be used for classification purpose. The prediction of the probability of target variable is done by this classifier. The target or the dependent variable has only two classes. These variables are bifurcated.

Logistic Regression is a fantastic machine learning technique since it may offer opportunities and categorize fresh data using non-stop datasets. Logistic Regression is helpful in categorizing data using specific data types and can evaluate variables to use for the classification.

It gives us the probabalistic values that is between 0 and 1.It does not give the exact values. Logistic Regression is same as the Linear regression. Logistic Regression is widely used for solving the problems of classification. Problems of Regression are done by the Linear Regression.



3.6 Evaluation Metrics

Accuracy

- The percentage of test samples that are successfully categorised. This indicator measures how close the model prediction is to the actual data.
- Random Forest Classifier found to be performing better than others.

Models	Accuracy
Random Forest Classifier	0.854497
Logistic Regression	0.846560
Naive bayes Classifier	0.846560

Random Forest Classifier

In [19]:
```
# create classifier
rf_clf = RandomForestClassifier(n_estimators=200, criterion='entropy')
rf_clf.fit(vec_train_data, df_train['Label'])
```

Out[19]:
```
RandomForestClassifier(criterion='entropy', n_estimators=200)
```

In [20]:
```
# run precictions on test data
y_pred = rf_clf.predict(vec_test_data)
```

In [21]:
```
matrix=confusion_matrix(df_test['Label'], y_pred)
print(matrix)
print("Random Forest Classifier: ",accuracy_score(df_test['Label'], y_pred))
print(classification_report(df_test['Label'], y_pred))
```

```
[[131  55]
 [  0 192]]
Random Forest Classifier:  0.8544973544973545
              precision    recall  f1-score   support

           0       1.00      0.70      0.83       186
           1       0.78      1.00      0.87       192

    accuracy                           0.85       378
   macro avg       0.89      0.85      0.85       378
weighted avg       0.89      0.85      0.85       378
```

Naive Bayes Classifier

In [22]:
```
naive=MultinomialNB()
naive.fit(vec_train_data,df_train['Label'])
```

Out[22]:
```
MultinomialNB()
```

In [23]:
```
y1_pred = naive.predict(vec_test_data)
```

In [24]:
```
matrix=confusion_matrix(df_test['Label'], y1_pred)
print(matrix)
print("Naive Bayes Classifier: ",accuracy_score(df_test['Label'], y1_pred))
print(classification_report(df_test['Label'], y1_pred))
```

```
[[148  38]
 [ 20 172]]
Naive Bayes Classifier:  0.8465608465608465
              precision    recall  f1-score   support

           0       0.88      0.80      0.84       186
           1       0.82      0.90      0.86       192

    accuracy                           0.85       378
   macro avg       0.85      0.85      0.85       378
weighted avg       0.85      0.85      0.85       378
```

Logistic Regression

In [25]:
```
log_reg = LogisticRegression()
log_reg.fit(vec_train_data, df_train["Label"])
```

Out[25]:
```
LogisticRegression()
```

In [26]:
```
y2_pred = log_reg.predict(vec_test_data)
```

In [27]:
```
matrix=confusion_matrix(df_test['Label'], y2_pred)
print(matrix)
print("Logistic Regression: ",accuracy_score(df_test['Label'], y2_pred))
print(classification_report(df_test['Label'], y2_pred))
```

```
[[144  42]
 [ 16 176]]
Logistic Regression:  0.8465608465608465
              precision    recall  f1-score   support

           0       0.90      0.77      0.83       186
           1       0.81      0.92      0.86       192

    accuracy                           0.85       378
   macro avg       0.85      0.85      0.85       378
weighted avg       0.85      0.85      0.85       378
```

Chapter 4
Conclusion and Scope

4.1 Conclusion

- Using Machine Learning, we can comprehend and analyse the impact of sentiment on the stock market and create predictions.
- We used a set of text featurization and machine learning techniques to analyse the Top 25 News Headlines and extract financial sentiment. The relationship between daily sentiment and daily stock price fluctuation is investigated.
- We got better results with the Random Forest Classifier.
- We are able to extract the features from the dataset using NLP techniques and able to train a model which can predict the stock price.

4.2 Scope

- We can use the same method to forecast a company's stock future performance and help them design new policies based on that performance of the stock.
- In the future, this research could be expanded to include more types of sentiment analysis as well as more text data.
- Developing a stock market price prediction model based on the superior algorithms discussed in this paper would reveal whether or not such predictive models are more accurate.
- Conducting research with a broader range of data sources and over a prolonged interval of time may indicate whether the results of this study are repeated in other studies.

References

[1] Sun, J. (2016, August). Daily News for Stock Market Prediction, Version 1. Retrieved [Date You Retrieved This Data] from https://www.kaggle.com/aaron7sun/stocknews.

[2] Image Credits: Google

BATCH C9_2

ORIGINALITY REPORT

4%	3%	0%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijsrd.com Internet Source	1%
2	Submitted to Birla Institute of Technology and Science Pilani Student Paper	1%
3	Submitted to The British College Student Paper	1%
4	Submitted to University of Ibadan Student Paper	1%
5	etd.astu.edu.et Internet Source	<1%
6	Submitted to University of Westminster Student Paper	<1%
7	www.coursehero.com Internet Source	<1%
8	mafiadoc.com Internet Source	<1%

APPENDIX

Screenshots

	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	...	Top16	Top17	Top18	Top19
0	2000-01-03	0	A 'hindrance to operations': extracts from the...	Scorecard	Hughes' instant hit buoys Blues	Jack gets his skates on at ice-cold Alex	Chaos as Maracana builds up for United	Depleted Leicester prevail as Elliott spoils E...	Hungry Spurs sense rich pickings	Gunners so wide of an easy target	...	Flintoff injury piles on woe for England	Hunters threaten Jospin with new battle of the...	Kohl's successor drawn into scandal	The difference between men and women
1	2000-01-04	0	Scorecard	The best lake scene	Leader: German sleaze inquiry	Cheerio, boyo	The main recommendations	Has Cubie killed fees?	Has Cubie killed fees?	Has Cubie killed fees?	...	On the critical list	The timing of their lives	Dear doctor	Irish court halts IRA man's extradition to Nor...
2	2000-01-05	0	Coventry caught on counter by Flo	United's rivals on the road to Rio	Thatcher issues defence before trial by video	Police help Smith lay down the law at Everton	Tale of Trautmann bears two more retellings	England on the rack	Pakistan retaliate with call for video of Walsh	Cullinan continues his Cape monopoly	...	South Melbourne (Australia)	Necaxa (Mexico)	Real Madrid (Spain)	Raja Casablanca (Morocco)
3	2000-01-06	1	Pilgrim knows how to progress	Thatcher facing ban	McIlroy calls for Irish fighting spirit	Leicester bin stadium blueprint	United braced for Mexican wave	Auntie back in fashion, even if the dress look...	Shoaib appeal goes to the top	Hussain hurt by 'shambles' but lays blame on e...	...	Putin admits Yeltsin quit to give him a head s...	BBC worst hit as digital TV begins to bite	How much can you pay for...	Christmas glitches
4	2000-01-07	1	Hitches and Horlocks	Beckham off but United survive	Breast cancer screening	Alan Parker	Guardian readers: are you all whingers?	Hollywood Beyond	Ashes and diamonds	Whingers - a formidable minority	...	Most everywhere: UDIs	Most wanted: Chloe lunettes	Return of the cane 'completely off the agenda'	From Sleepy Hollow to Greeneland

Import all the required libraries

In [1]:
```
import numpy as np
import pandas as pd
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

Read dataset

In [2]:
```
df=pd.read_csv('Data.csv', encoding = "ISO-8859-1")
```

In [3]:
```
df.head()
```

Out[3]:

	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	...	Top16	Top17	Top18	Top19	Top20	Top21	Top22	Top23	Top24	Top25
0	2000-01-03	0	A 'hindrance to operations': extracts from the...	Scorecard	Hughes' instant hit buoys Blues	Jack gets his skates on at ice-cold Alex	Chaos as Maracana builds up for United	Depleted Leicester prevail as Elliott spoils E...	Hungry Spurs sense rich pickings	Gunners so wide of an easy target	...	Flintoff injury piles on woe for England	Hunters threaten Jospin with new battle of the...	Kohl's successor drawn into scandal	The difference between men and women	Sara Denver, nurse turned solicitor	Diana's landmine crusade put Tories in a panic	Yeltsin's resignation caught opposition flat-f...	Russian roulette	Sold out	Recovering a title
1	2000-01-04	0	Scorecard	The best lake scene	Leader: German sleaze inquiry	Cheerio, boyo	The main recommendations	Has Cubie killed fees?	Has Cubie killed fees?	Has Cubie killed fees?	...	On the critical list	The timing of their lives	Dear doctor	Irish court halts IRA man's extradition to Nor...	Burundi peace initiative fades after rebels re...	PE points the way forward to the ECB	Campaigners keep up pressure on Nazi war crime...	Jane Ratcliffe	Yet more things you wouldn't know without the ...	Millennium bug fails to bite
2	2000-01-05	0	Coventry caught on counter by Flo	United's rivals on the road to Rio	Thatcher issues defence before trial by video	Police help Smith lay down the law at Everton	Tale of Trautmann bears two more retellings	England on the rack	Pakistan retaliate with call for video of Walsh	Cullinan continues his Cape monopoly	...	South Melbourne (Australia)	Necaxa (Mexico)	Real Madrid (Spain)	Raja Casablanca (Morocco)	Corinthians (Brazil)	Tony's pet project	Al Nassr (Saudi Arabia)	Ideal Holmes show	Pinochet leaves hospital after tests	Useful links
3	2000-01-06	1	Pilgrim knows how to progress	Thatcher facing ban	McIlroy calls for Irish fighting spirit	Leicester bin stadium blueprint	United braced for Mexican wave	Auntie back in fashion, even if the dress look...	Shoaib appeal goes to the top	Hussain hurt by 'shambles' but lays blame on e...	...	Putin admits Yeltsin quit to give him a head s...	BBC worst hit as digital TV begins to bite	How much can you pay for...	Christmas glitches	Upending a table, Chopping a line and Scoring ...	Scientific evidence 'unreliable', defence claims	Fusco wins judicial review in extradition case	Rebels thwart Russian advance	Blair orders shake-up of failing NHS	Lessons of law's hard heart
4	2000-01-07	1	Hitches and Horlocks	Beckham off but United survive	Breast cancer screening	Alan Parker	Guardian readers: are you all whingers?	Hollywood Beyond	Ashes and diamonds	Whingers - a formidable minority	...	Most everywhere: UDIs	Most wanted: Chloe lunettes	Return of the cane 'completely off the agenda'	From Sleepy Hollow to Greeneland	Blunkett outlines vision for over 11s	Embattled Dobson attacks 'play now, pay later'...	Doom and the Dome	What is the north-south divide?	Aitken released from jail	Gone aloft

5 rows × 27 columns

In [4]:
```
#check for null values
df.isnull().sum()
```

Out[4]:
```
Date     0
Label    0
Top1     0
Top2     0
Top3     0
Top4     0
Top5     0
Top6     0
Top7     0
Top8     0
Top9     0
Top10    0
Top11    0
Top12    0
Top13    0
Top14    0
Top15    0
Top16    0
Top17    0
Top18    0
Top19    0
Top20    0
Top21    0
Top22    0
Top23    1
Top24    3
Top25    3
dtype: int64
```

In [5]:
```
df.shape
```

Out[5]:
```
(4101, 27)
```

In [6]:
```
df = df.replace(np.nan, ' ', regex=True)

#sanity check
df.isnull().sum().sum()
```

Out[6]:
```
0
```

Split the dataset into training and test data

In [7]:
```
df_train = df[df['Date'] < '20150101']

df_test = df[df['Date'] > '20141231']
```

Feature Engineering

In [8]:
```
# function for cleaning the data
def clean_data(dataset):
    data = dataset.iloc[:,2:27]
    data.replace("[^a-zA-Z]", " ", regex=True, inplace=True)
    return data

# function for combining the headlines of all the columns into single column
def combine_data(data):
    headlines = []
    for i in range(len(data.index)):
        headlines.append(' '.join(str(x) for x in data.iloc[i, :]))
    return headlines

# function to perform lemmatization of the word
def lemmatize_data(data, lemmatizer):
    cleaned_dataset = []
    for i in range(len(data)):
        clean_text = data[i].lower()
        clean_text = clean_text.split()
        clean_text = [lemmatizer.lemmatize(word) for word in clean_text if word not in stopwords.words('english')]
        cleaned_dataset.append(' '.join(clean_text))
    return cleaned_dataset

# function to vectorize the data
def vectorize_data(data, cv):
    vectorized_dataset = cv.fit_transform(data)
    return vectorized_dataset
```

Clean train and test data

In [9]:
```
# clean train and test data
clean_train_data = clean_data(df_train)
clean_test_data = clean_data(df_test)
```

In [10]:
```
clean_train_data.head(1)
```

Out[10]:

	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	Top10	...	Top16	Top17	Top18	Top19	Top20	Top21	Top22	Top23	Top24	Top25
0	A hindrance to operations extracts from the...	Scorecard	Hughes instant hit buoys Blues	Jack gets his skates on at ice cold Alex	Chaos as Maracana builds up for United	Depleted Leicester prevail as Elliott spoils E...	Hungry Spurs sense rich pickings	Gunners so wide of an easy target	Derby raise a glass to Strupar s debut double	Southgate strikes Leeds pay the penalty	...	Flintoff injury piles on woe for England	Hunters threaten Jospin with new battle of the...	Kohl s successor drawn into scandal	The difference between men and women	Sara Denver nurse turned solicitor	Diana s landmine crusade put Tories in a panic	Yeltsin s resignation caught opposition flat f...	Russian roulette	Sold out	Recovering a title

1 rows × 25 columns

In [11]:
```
clean_test_data.head(1)
```

Out[11]:

	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	Top10	...	Top16	Top17	Top18	Top19	Top20	Top21	Top22	Top23	Top24	Top25
3723	Most cases of cancer are the result of sheer b...	Iran dismissed United States efforts to fight ...	Poll One in Germans would join anti Muslim ...	UK royal family s Prince Andrew named in US la...	Some asylum seekers refused to leave the bu...	Pakistani boat blows self up after India navy ...	Sweden hit by third mosque arson attack in a week	cars set alight during French New Year	Salaries for top CEOs rose twice as fast as av...	Norway violated equal pay law judge says Jud...	...	Ukrainian minister threatens TV channel with c...	Palestinian President Mahmoud Abbas has entere...	Israeli security center publishes names of ...	The year was the deadliest year yet in Sy...	A Secret underground complex built by the Nazi...	Restrictions on Web Freedom a Major Global Iss...	Austrian journalist Erich Mchel delivered a pr...	Thousands of Ukraine nationalists march in Kiev	Chinas New Years Resolution No More Harvestin...	Authorities Pull Plug on Russia s Last Politic...

1 rows × 25 columns

Combine headlines

In [12]:
```
# combine the headlines in single column
comb_train_data = combine_data(clean_train_data)
comb_test_data = combine_data(clean_test_data)
```

In [13]:
```
comb_train_data[0]
```

Out[13]:
```
'A  hindrance to operations  extracts from the leaked reports Scorecard Hughes  instant hit buoys Blues Jack gets his skates on at ice cold Alex Chaos as Maracana builds up for United Depleted Leicester prevail as Elliott spoils Everton s party H
ungry Spurs sense rich pickings Gunners so wide of an easy target Derby raise a glass to Strupar s debut double Southgate strikes  Leeds pay the penalty Hammers hand Robson a youthful lesson Saints party like it s      Wear wolves have turned into
lambs Stump mike catches testy Gough s taunt Langer escapes to hit     Flintoff injury piles on woe for England Hunters threaten Jospin with new battle of the Somme Kohl s successor drawn into scandal The difference between men and women Sara Denv
er  nurse turned solicitor Diana s landmine crusade put Tories in a panic Yeltsin s resignation caught opposition flat footed Russian roulette Sold out Recovering a title'
```

Creating Lemmatizer Object

```
lemmatizer = WordNetLemmatizer()
```

Lemmatize the data

```
# Lemmatize data
train_data = lemmatize_data(comb_train_data, lemmatizer)
test_data = lemmatize_data(comb_test_data, lemmatizer)
```

```
train_data[0]
```

```
'hindrance operation extract leaked report scorecard hughes instant hit buoy blue jack get skate ice cold alex chaos maracana build united depleted leicester prevail elliott spoil everton party hungry spur sense rich picking gunner wide easy targe
t derby raise glass strupar debut double southgate strike leeds pay penalty hammer hand robson youthful lesson saint party like wear wolf turned lamb stump mike catch testy gough taunt langer escape hit flintoff injury pile woe england hunter thre
aten jospin new battle somme kohl successor drawn scandal difference men woman sara denver nurse turned solicitor diana landmine crusade put tory panic yeltsin resignation caught opposition flat footed russian roulette sold recovering title'
```

Create CountVectorizer object

```
countvector = CountVectorizer(ngram_range=(2,2))
```

Vectorize the data

```
vec_train_data = vectorize_data(train_data, countvector)
vec_test_data = countvector.transform(test_data)
```

Random Forest Classifier

In [19]:
```
# create classifier
rf_clf = RandomForestClassifier(n_estimators=200, criterion='entropy')
rf_clf.fit(vec_train_data, df_train['Label'])
```

Out[19]:
```
RandomForestClassifier(criterion='entropy', n_estimators=200)
```

In [20]:
```
# run precictions on test data
y_pred = rf_clf.predict(vec_test_data)
```

In [21]:
```
matrix=confusion_matrix(df_test['Label'], y_pred)
print(matrix)
print("Random Forest Classifier: ",accuracy_score(df_test['Label'], y_pred))
print(classification_report(df_test['Label'], y_pred))
```

```
[[131  55]
 [  0 192]]
Random Forest Classifier:  0.8544973544973545
              precision    recall  f1-score   support

           0       1.00      0.70      0.83       186
           1       0.78      1.00      0.87       192

    accuracy                           0.85       378
   macro avg       0.89      0.85      0.85       378
weighted avg       0.89      0.85      0.85       378
```

Naive Bayes Classifier

In [22]:
```
naive=MultinomialNB()
naive.fit(vec_train_data,df_train['Label'])
```

Out[22]:
```
MultinomialNB()
```

In [23]:
```
y1_pred = naive.predict(vec_test_data)
```

In [24]:
```
matrix=confusion_matrix(df_test['Label'], y1_pred)
print(matrix)
print("Naive Bayes Classifier: ",accuracy_score(df_test['Label'], y1_pred))
print(classification_report(df_test['Label'], y1_pred))
```

```
[[148  38]
 [ 20 172]]
Naive Bayes Classifier:  0.8465608465608465
              precision    recall  f1-score   support

           0       0.88      0.80      0.84       186
           1       0.82      0.90      0.86       192

    accuracy                           0.85       378
   macro avg       0.85      0.85      0.85       378
weighted avg       0.85      0.85      0.85       378
```

Logistic Regression

In [25]:
```
log_reg = LogisticRegression()
log_reg.fit(vec_train_data, df_train["Label"])
```

Out[25]:
```
LogisticRegression()
```

In [26]:
```
y2_pred = log_reg.predict(vec_test_data)
```

In [27]:
```
matrix=confusion_matrix(df_test['Label'], y2_pred)
print(matrix)
print("Logistic Regression: ",accuracy_score(df_test['Label'], y2_pred))
print(classification_report(df_test['Label'], y2_pred))
```

```
[[144  42]
 [ 16 176]]
Logistic Regression:  0.8465608465608465
              precision    recall  f1-score   support

           0       0.90      0.77      0.83       186
           1       0.81      0.92      0.86       192

    accuracy                           0.85       378
   macro avg       0.85      0.85      0.85       378
weighted avg       0.85      0.85      0.85       378
```

CODE

Python code

```
#Import all the required libraries
import numpy as np
import pandas as pd
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score

#Read dataset

df=pd.read_csv('Data.csv', encoding = "ISO-8859-1")

df.head()

df.shape

#check for null values

df.isnull().sum()

df = df.replace(np.nan, ' ', regex=True)

df.isnull().sum().sum()

# Split the dataset into training and test data

df_train = df[df['Date'] < '20150101']

df_test = df[df['Date'] > '20141231']

# Feature Engineering
# function for cleaning the data

def clean_data(dataset):
    data = dataset.iloc[:,2:27]
    data.replace("[^a-zA-Z]", " ", regex=True, inplace=True)
    return data

# function for combining the headlines of all the columns into single column
def combine_data(data):
    headlines = []
    for i in range(len(data.index)):
        headlines.append(' '.join(str(x) for x in data.iloc[i, :]))
    return headlines
```

```
# function to perform lemmatization of the word

def lemmatize_data(data, lemmatizer):
    cleaned_dataset = []
    for i in range(len(data)):
        clean_text = data[i].lower()
        clean_text = clean_text.split()
        clean_text = [lemmatizer.lemmatize(word) for word in clean_text if word not in
stopwords.words('english')]
        cleaned_dataset.append(' '.join(clean_text))
    return cleaned_dataset

# function to vectorize the data

def vectorize_data(data, cv):
    vectorized_dataset = cv.fit_transform(data)
    return vectorized_dataset

# clean train and test data

clean_train_data = clean_data(df_train)
clean_test_data = clean_data(df_test)

`   clean_train_data.head(1)

clean_test_data.head(1)

# combine the headlines in single column

comb_train_data = combine_data(clean_train_data)
comb_test_data = combine_data(clean_test_data)

comb_train_data[0]

# creating Lemmatizer Object

lemmatizer = WordNetLemmatizer()

# Lemmatize the data

# lemmatize data
train_data = lemmatize_data(comb_train_data, lemmatizer)
test_data = lemmatize_data(comb_test_data, lemmatizer)

train_data[0]


# create CountVectorizer object

Countvector = CountVectorizer(ngram_range=(2,2))
```

```
# vectorize the data

vec_train_data = vectorize_data(train_data, countvector)
vec_test_data = countvector.transform(test_data)

# Random Forest Classifier

#  create classifier

rf_clf = RandomForestClassifier(n_estimators=200, criterion='entropy')
rf_clf.fit(vec_train_data, df_train['Label'])

RandomForestClassifier(criterion='entropy', n_estimators=200)

#  run precictions on test data

y_pred = rf_clf.predict(vec_test_data)

matrix=confusion_matrix(df_test['Label'], y_pred)
print(matrix)
print("Random Forest Classifier: ",accuracy_score(df_test['Label'], y_pred))
print(classification_report(df_test['Label'], y_pred))

# Naive Bayes Classifier

naive=MultinomialNB()
naive.fit(vec_train_data,df_train['Label'])

MultinomialNB()

y1_pred = naive.predict(vec_test_data)
matrix=confusion_matrix(df_test['Label'], y1_pred)
print(matrix)
print("Naive Bayes Classifier: ",accuracy_score(df_test['Label'], y1_pred))
print(classification_report(df_test['Label'], y1_pred))

# Logistic Regression

log_reg = LogisticRegression()
log_reg.fit(vec_train_data, df_train["Label"])

LogisticRegression()

y2_pred = log_reg.predict(vec_test_data)
matrix=confusion_matrix(df_test['Label'], y2_pred)
print(matrix)
print("Logistic Regression: ",accuracy_score(df_test['Label'], y2_pred))
print(classification_report(df_test['Label'], y2_pred
```