STOCK SENTIMENT ANALYSIS USING NEWS HEADLINES

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering

By

Ayush Soni 181224

UNDER THE SUPERVISION OF

Dr. Pankaj Dhiman

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234,

Himachal Pradesh, INDIA

TABLE OF CONTENT

LIST OF FIGURES

I

DECLARATION

I hereby declare that this project has been done by me under the supervision of **(**Dr. Pankaj Dhiman, Assistant Professor (Grade-II)), the Jaypee University of Information Technology. I also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:
Dr. Pankaj Dhiman
Assistant Professor (Grade-II)
Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology

Submitted by:

Ayush Soni 181224 Ayush Soni

Computer Science & Engineering Department
Jaypee University of Information Technology

II

CERTIFICATE

This is to certify that the work which is being presented in the project report titled "Stock Sentiment Analysis using News Headlines" is in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Ayush Soni(181224) during the period from January 2022 to May 2021 under the supervision of Dr. Pankaj Dhiman, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Ayush Soni 181224

The above statement made is correct to the best of my knowledge.

Dr. Pankaj Dhiman
Assistant Professor (Grade-II)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat

ACKNOWLEDGEMENT

First and foremost, I offer my heartfelt gratitude to Almighty God for his divine grace, which has enabled me to successfully complete the project work.

I am grateful to my supervisor, Dr. Pankaj Dhiman, Assistant Professor (Grade-II), Department of CSE Jaypee University of Information Technology, Wakhnaghat, and express my deep gratitude. My supervisor's extensive knowledge and deep interest in the fields of "Visualization, Data Science, and Deep Learning" aided me in completing my research. His never-ending patience, intellectual direction, persistent encouragement, constant and energetic supervision, constructive criticism, helpful suggestions, and reading numerous poor versions and revising them at all stages allowed this project to be completed.

I'd like to thank Dr. Pankaj Dhiman, Department of CSE, for his invaluable assistance in completing my project.

I'd also like to express my gratitude to everyone who has directly or indirectly assisted me in making this project a success. In this unique situation, I'd want to express my gratitude to the different staff members, both teaching and non-teaching, who have provided me with valuable assistance and assisted my project.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Ayush Soni

ABSTRACT

Data science refers to the process of cleansing, aggregating, and modifying data in order to undertake advanced data analysis. The broad consensus about a stock or the stock market as a whole is referred to as market sentiment. When prices are rising, market sentiment is positive. When prices are declining, market sentiment is pessimistic. This study involves gathering non-quantifiable data, such as financial news stories about a company, and using news sentiment classification to forecast its future stock trend. This is an attempt to investigate the relationship between news and stock movement, assuming that news stories have an impact on the stock market.

We can determine the moment of the change in stock values by comparing these results to the movement of stock market values over the same time periods using sentiment analysis of economic news headlines.

Chapter 01:- INTRODUCTION

1.1 Introduction

Cleaning, aggregating, and modifying data to do sophisticated data analysis are all part of data science. The stock market, as well as its tendencies, are particularly volatile in nature in the realm of finance. It draws scholars who are interested in capturing the volatility and forecasting its next actions. Market behaviour is studied by investors and market analysts, who then plan their purchase and sell tactics accordingly.

Because the stock market generates a significant amount of data every day, it is impossible for an individual to examine all current and historical data in order to forecast a stock's future trajectory. Forecasting market trends can be done in a couple of ways. There are two types of analysis: the technical and fundamental. In technical analysis, previous price and volume are used to forecast future trends, but in fundamental analysis, the past price and volume are used to forecast future trends. a basic examination Fundamental analysis, on the other hand, entails examining a company's financial data in order to get insight.The efficacy of both technical and fundamental analysis is disputed by the efficient-market hypothesis which states that stock market prices are essentially unpredictable.

This study uses the fundamental analysis technique to predict future stock trends by using news items about a firm as primary data and attempting to categorise the news as good (positive) or poor (negative) (negative). If the news sentiment is favourable, the stock price is more likely to rise, whereas if it is negative, the stock price is more likely to fall. This study aims to develop a model that can forecast news polarity, which could influence stock market patterns. Check the effect of news items on stock prices, in other words. To determine news polarity, we use machine learning algorithms and other text mining approaches. Also, be able to classify unclassified news that isn't used in the classification process.

1.2 Problem Statement

Stock exchange is a subject that is highly affected by economic, social, and political factors. There are several factors e.g. external factors or internal factors which can affect and move the stock market. Stock prices rise and fall every second due to variations in supply and demand. Various Data mining techniques are frequently involved to solve this problem. But technique using machine learning will give a more accurate, precise and simple way to solve such issues related to stock and market prices. “Stock Sentiment Analysis using News Headlines” a method for predicting stock prices is developed using news articles. The changes in stock prices of a company, the rises and falls, are correlated with the public opinions being expressed in news about that company. Understanding author’s opinion from a piece of text is the objective of sentiment analysis. Positive news and news in social media about a company would definitely encourage people to invest in the stocks of that company and as a result the stock price of that company would increase. A prediction model for finding and analyzingthe correlation between contents of news headlines and stock prices and then making predictions for future can be developed by using data science and machine learning algorithm.

1.3 Objectives

One of the most essential topics to be addressed in academic and financial study is stock price prediction. The research usually use a variety of data mining approaches. To resolve this issue. However, applying machine learning/deep learning techniques will provide a more accurate, precise, and straightforward method of resolving such difficulties as stock and market prices. The information gleaned from the news is quite valuable for forecasting. The task of judging opinion in a piece of text as good, negative, or neutral is known as sentiment analysis of news headlines.

Using news headlines, a system for forecasting and classifying stock sentiments is developed in this study. Sentiment analysis of the collected news data is used to construct a prediction model for detecting and analysing correlations between news article contents and stock prices, and then making forecasts for future prices using data science and machine learning techniques. Stock price volatility is determined by the gains or losses of specific corporations. One of the most important variables influencing the stock market is news stories. This is an attempt to investigate the relationship between news and stock market trends, as well as assess feelings.

1.4 Methodology

- Data Collection: - We have taken a dataset which consist of news headlines. There are 25 columns of top news headlines for each day in data frame. Data ranges from 2000 to 2016. There are labels in dataset Class 1 for the increase in stock price and Class 0 if the stock price decreases or stay same.
- Data Pre-Processing: - News Headlines consists of many acronyms, emoticons and unnecessary data like pictures and URL's. So, news headlines are pre-processed to represent correct emotions of public. For pre-processing of news, we employed three stages of filtering: Tokenization, stop words removal and regex matching for removing special characters.
- Sentiment Analysis: - Sentiment analysis task is very much field specific. News Headlines are classified as positive, negative and neutral based on the sentiment present. Out of the total news headlines are examined by humans and annotated as 1 for Positive, 0 for Neutral and Negative emotions. For classification of nonhuman annotated news, a machine learning model is trained whose features are extracted from the human annotated news.
- Feature Extraction: - Textual representations can be done using n-grams. N-gram Representation: N-gram representation is known for its specificity to match the corpus of text being studied. In these techniques a full corpus of related text is parsed which are news in the present work, and every appearing word sequence of length n is extracted from the news to form a dictionary of words and phrases. In this representation, headlines is split into N-grams and the features to the model are a string of 1s and 0s where 1 represents the presence of that N-gram of the news in the corpus and a 0 indicates the absence.

- Model Training: - The features extracted using the above methods for the news are fed to the classifier and trained using classification methods like Random Forest Classifier, Naïve Bayes to estimate the movement of the change in stock market price vs the volume as well as sentiment of news articles.

1.5 Organization

- Chapter I, Contains the Introduction, Problem statement, scope, Objectives of the System or Project.
- Chapter 2, the literature survey discusses an abstract survey of the published papers and if any disadvantages are identified in the paper.
- Chapter 3 discusses the detailed requirement of the problem identified for the major project, system architecture and implementation details
- Chapter 4 Discusses the Performance Analysis of the model.
- Chapter 5 Concludes the Report
- Chapter 6 Discusses any Future Scope

Chapter 02:- LITERATURE SURVEY

2.1 Survey

Stock price trend prediction is an active research area, as more accurate predictions are directly related to more returns We have used this sentiment detection approach based on this research. Our main motive behind this is to learn NLP techniques and to analyze that how the news can change the stocks.

in stocks. Therefore, in recent years, significant efforts have been put into developing models that can predict for future trend of a specific stock or overall market. Following is discussion on previous research on sentiment analysis of text data and different classification techniques: -

Nagar and Hahsler in their research presented an automated text mining based approach to aggregate news stories from various sources and create a News Corpus. The Corpus is filtered down to relevant sentences and analyzed using Natural Language Processing (NLP) techniques. They also state that the time variation of News Sentiment shows a very strong correlation with actual stock movement

Many algorithms ofdata mining have been proposed to predict stock price. The opinion summary is explored based on the opinion sentences. Kalyani Joshi, Prof. Bharathi H. N., Prof. Jyothi Rao also worked on Stock price prediction using sentiment analysis on news heaadlines.

2.2 Existing System

Different chance variables, as well as numerous algorithms, consisting of databases, programmes, and much more, are used in traditional divination procedures and patterns. The high-chance and low-chance patient types are determined mostly based on group assessments. According to the stock price prediction method, which was intended to assist investors in making financial decisions. Most studies concentrate on "lowest price buy" and "highest selling price." When buying stocks at the lowest price and selling them at the highest price, this is known as the "lowest purchase" and "highest selling" approach. The dataset was pre-processed and adjusted up for real analysis using the ANN approach with back propagation for stock price forecasts. As a result, our administrator may upload stock price history, including the open, highest, lowest, and closing prices for the day. Data preparation will also be discussed in the paper. After preparing the data, the system scans stock price history and feeds it into the Back propagation algorithm. Furthermore, the proposed article investigates the use of the prediction system in real-world scenarios, as well as difficulties related to the correctness of the overall values provided. The output of back propagation is the ultimate projected rate. The suggested system may generate a stock price prediction list and a graph of the prediction table so that the user can see the final projected result. Stock market institutions will benefit greatly from good stock prediction since it will bring real-world answers to the challenges that stock investors encounter.

2.3 Proposed System

The proposed Data Science & Machine Learning (ML) Based Ailment Divination system will use a number of techniques, algorithms, and other tools to create a Stock Sentiment Analysis based on News Headlines and compare it to previously collected system data. In machine learning, classification is a fascinating topic. In our situation, we need to build a classifier using our tagged news corpus that is "positive," "negative," or "neutral." We've categorized news as "positive," "negative," or "neutral" using the implemented classifier, which is based on taught news attributes. The Random Forest Classifier and the Naive Bayes models are two common text classification classifiers investigated in this study.

Furthermore, we used a bag of words to create the Random Forest Classifier. It is used by the algorithm to forecast stock prices. We've categorised news headlines and labels by day. The train classifier will be used to forecast stock sentiment and whether it will rise or fall in the future. The confusion matrix will be used to assess correctness.

2.4 Feasibility Study

A feasibility study is a preliminary investigation into the information of potential users in order to estimate the system's resource requirements, costs, benefits, and viability. A feasibility study considers the system's implementation and operation in light of numerous restrictions. The resources required for implementation, such as computer equipment, labour, and expenditures, are estimated at this stage. The anticipated costs are compared to available resources, and a system cost-benefit analysis is performed. The feasibility study is an activity that entails analysing the problem and gathering all essential project information. The feasibility study's major goal is to see if the project is viable in terms of economics, technical feasibility, operational feasibility, and scheduling feasibility.

2.4.1 Economic Feasibility

The difference between the benefits or outcomes we obtain from a product and the entire cost we spend to improve it is called economic recovery. In order to evaluate economic feasibility, the costs of creating and implementing a new system are compared against the benefits of having the new system in place. This feasibility report provides top management with the financial foundation for the new system. A plain economic analysis that offers a real cost-benefit comparison is significantly more effective in this situation. Furthermore, as the project progresses, this provides as a useful benchmark for evaluating actual costs. Automation may provide a number of intangible advantages. These might contribute to higher product quality, better decision-making, and more timely information, as well as enhanced operational accuracy, better documentation, and record-keeping, and faster information retrieval.

2.4.2 Schedule Feasibility

A project will fail if it takes too long to finish before it is helpful. This usually comprises evaluating how long it will take to build the system and deciding if it can be finished in a particular length of time using methods like payback period. The feasibility of a timetable is a statistic for assessing the timeliness of a project. Given our technological skills, are the project timeframes reasonable? Some undertakings are launched with the goal of meeting a deadline. It's critical to determine if the deadlines are necessary or desired.

2.4.3 Technical Feasibility

Determining technical feasibility is the most challenging part of a feasibility study. This is owing to the fact that the system has yet to be designed, making it impossible to access problems such as performance, prices (due to the sort of technology to be installed), and so on. A number of things must be considered while doing a technical analysis, including a grasp of the different technologies involved in the proposed system. Before we begin the task, we need to be completely certain of a few things. What technologies will be required for the creation of the new system?

2.4.4 Operational Feasibility

Only if the proposed project can be transformed into information systems that meet the operating criteria will it be beneficial. Simply put, this feasibility test determines whether or not the system will function after it has been developed and implemented. Are there any significant obstacles to implementation?

The idea was to create a more user-friendly web application. It's easier to use and may be utilized on any website. It is both free and inexpensive to use.

Chapter 03:- SYSTEM DEVELOPMENT

3.1 Requirements

After the extensive analysis of the problems in the system, we are familiarized with the requirement that the current system needs. The requirement that the system needs is categorized the functional and non-functional requirements. These requirements are listed below:

3.1.1 Functional Requirements

Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users.Based on this, the functional requirements that the system must require are as follows:-

- System should be able to process news headlines stored in dataset after retrieval.
- System should be able to analyze data and classify each headlinessentiment.

3.1.2 Non-Functional Requirements

Non-functional requirements is a description of features, characteristics and attribute of the system as well as any constraints that may limit the boundaries ofthe proposed system.The non-functional requirements are essentially based on the performance, information, economy, control and security efficiency and services.Based on these the non-functional requirements are as follows:

- User friendly
- System should provide better accuracy
- To perform with efficient throughput and response time

3.1.3 Requirements for Hardware and Software

- Pentium 4. Intel Core i3, 5, i7. and 2 GHz processor RAM must be at 512MB.
- Input Keyboard and Mouse are the devices that are used.
- Monitor or PC as an output device.
- Versions of Windows 7, 10, and above are supported.
- Jupyter Notebook or Google colaboratory as a platform.
- Python is the programming language.

3.1 3.2 Tools & Technologies Used

3.2.1 Python

Python is a robust programming language with a wide range of capabilities. It has several capabilities that make working with specialised programmes (such as meta-programming and meta-objects) a joy, and it fully supports object-oriented programming. Many more paradigms are envisioned as extensions, such as contract generation and logic programming. Python incorporates power typing, as well as reference calculations and trash collection waste management. Advanced word processing (late binding), which binds the way words change during the process, is also available.

Python sentiment analysis is a way for examining a piece of text and determining the hidden sentiment. This is achieved through the use of a combination of machine learning and natural language processing (NLP). Sentiment analysis is a technique for analysing the emotions represented in a piece of writing.

3.2.2 Why Python?

We're using Python because it works on a wide range of platforms, including Windows, Mac OS X, Linux, Raspberry Pi, and more. Python is a language with no stages. Python is a language that is comparable to and as simple as English. Python supports a large number of libraries and has a simple linguistic structure similar to English, whereas Java and C++ have complex codes. Python programmes are shorter than those written in other programming languages. That is why we use Python for artificial intelligence, artificial consciousness, and dealing with massive amounts of data. Python is an article-oriented programming language. Python's ability to swiftly generate and manage data structures is one of its most widespread uses - Pandas, for example, provides a variety of tools for manipulating, analyzing, and even representing data structures and complex datasets.

3.2.3 Scikit-Learn

Scikit-learn is the most useful machine learning package in Python. Several useful machine learning and mathematical modelling techniques, including as division, slowdown, merging, and size reduction, are included in the learning library. It's one of the most used Scikit-learn APIs. Because it provides a uniform interface and a wide range of ML applications, the Estimator API is utilised with all of Scikit-machine Learn's learning algorithms. The scale refers to the object that the data reads (which contains data).

The TensorFlow library is a more sophisticated version of the standard library. Scikit-Learn is a cutting-edge package that lets you to create learning algorithms for a wide range of machines. For example, you may build an object in one or a few lines of code and use it to measure an Iron point or forecast a result.

3.2.4 Natural Language Processing

Natural language processing (NLP) is a subject of computer science specifically, a branch of artificial intelligence (AI) concerning the ability of computers to understand text and spoken words in the same manner that humans can. Computational linguistics rule-based human language modeling is combined with statistical, machine learning, and deep learning models in NLP. These technologies, when used together, allow computers to process human language in the form of text or speech data and 'understand' it’s full meaning, including the speaker’s or writer's intent and sentiment.

Some uses of NLP in project are:-

- Sentence Segmentation: The text material is divided into individual sentences as part of the NLP process.
- Tokenization: After the document has been broken down into sentences, the sentences are further broken down into individual words. Tokenization gets its name from the fact that each word is called a token.
- Parts-of-Speech-Tagging: We feed each token, along with a few words surrounding it, through a pre-trained part-of-speech classification model, which outputs the token's part-of-speech.
- Lemmatization: When referring to the same item or action, words might take several different forms. We perform lemmatization, which is the process of grouping together numerous inflections of a word to examine them as a single item, designated by the word's lemma, to avoid the computer from thinking of distinct forms of a word as different words.

3.2.5 **Sentiment Analysis**

Sentiment analysis is a technique for assessing if a piece of text is good, negative, or neutral. A sentiment analysis system for text analysis uses natural language processing (NLP) and machine learning techniques to give weighted sentiment ratings to entities, topics, themes, and categories inside a sentence or phrase.

Sentiment analysis aids data analysts in major organizations in gauging public sentiment, doing detailed market research, monitoring brand and product reputation, and comprehending customer experiences. In order to provide meaningful insights to their own customers, data analytics organizations frequently incorporate third-party sentiment analysis APIs into their own customer experience management, social media monitoring, or workforce analytics platforms.

3.3 Machine Learning Algorithms

As per the research Random Forest and Naive Bayes algorithms performs good in text classification and Random Forest Classifies is best for its accuracy.

3.3.1 Random Forest

Random forest is a self-administered learning algorithm that can be used for classification and regression. Regardless, it is primarily used for classification. We know that a tree is made up of trees, and that having more trees means having more powerful woods. Similarly, a random backwoods calculation computation resolves on decision trees based on data testing, receives the assumption from all of them, and then chooses the best course of action by projecting a polling form. It is a better technique than a single option tree since it reduces overfitting by averaging the results.

Working of Random Forest Algorithm

With the help of the stages below, we can learn how the Random Forest algorithm works.

- Begin by determining irregular cases from a given dataset.
- Following that, for each case, this calculation will generate a choice tree. Then it will get the desired outcome from each decision tree at that point.
- For each expected outcome, a polling form will be projected in this movement. Finally, as the last forecast outcome, choose the most casted ballot expectation result.

Applications of Random Forest Algorithm

- Banking Industry
- Healthcare and Medicine
- Stock Market
- E-Commerce

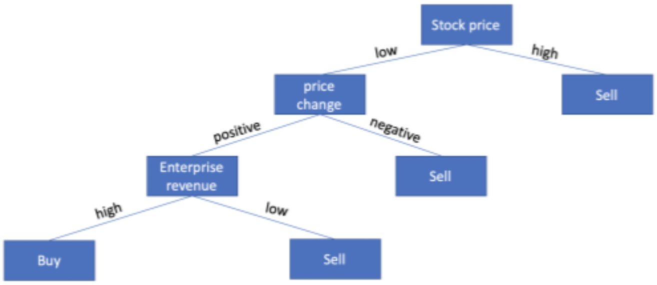
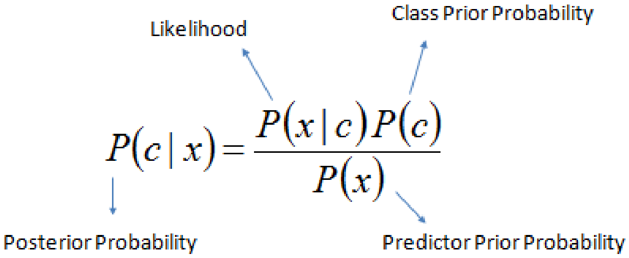


Fig 1 Decision Tress

3.3.2 Naive Bayes

Naive Bayes classifier is one of the oldest approaches for classification problems which is based on Bayes Theorem. The formula is:



$$P(c|x)=\frac{P(x|c)P(c)}{P(x)}$$

$$P(c|\mathrm{X}) = P(x_1|c)\times P(x_2|c)\times\cdots\times P(x_n|c)\times P(c)$$

Fig 2 Naive Bayes

The purpose is to determine the likelihood of an event. An occurrence with the letter B occurs. The naive Bayes classifier combines Bayes' model with decision rules such as the most likely result hypothesis. Naive Bayes methods are a collection of directed learning computations that rely on Bayes' hypothesis and the "naive" assumption of contingent freedom between each pair of highlights given the class variable's worth. It was created as a benchmark for text categorization tasks and is still used today.

Applications of Naive Bayes Algorithms

- **Real-time Prediction**: Naive Bayes is a fast and eager learning classifier. As a result, it can be utilized to make real-time forecasts.
- **Multi-class Prediction:** This algorithm is also well-known for its ability to predict multiple classes.
- **Message classification/Spam filtering/Sentiment analysis:** Naive Bayes classifiers are commonly utilized in message arrangement (due to increased prompts, multi-class concerns, and freedom rules)]. As a result, it's commonly used in Spam sifting (identifying spam email) and Sentiment Analysis (in web-based media examination, to spot positive and negative client feelings).

Building a basic model using Naive Bayes in Python

- Building a Naive Bayes model in Python is made easier with Scikit Learn (a Python module). The sci-kit learn package has three versions of the Naive Bayes model:
- Gaussian: it's a classification method that assumes features have a conventional distribution.
- Multinomial: It is a type of count that is used for discrete counts. Let's imagine we're dealing with a text classification issue. We can now examine Bernoulli trials, which are a step further than "word occurring within the record," and instead of "number of times result in number x_i is seen over the n preliminaries," you can think of it as "number of times result in number x_i is seen over the n preliminaries."

3.4 Working

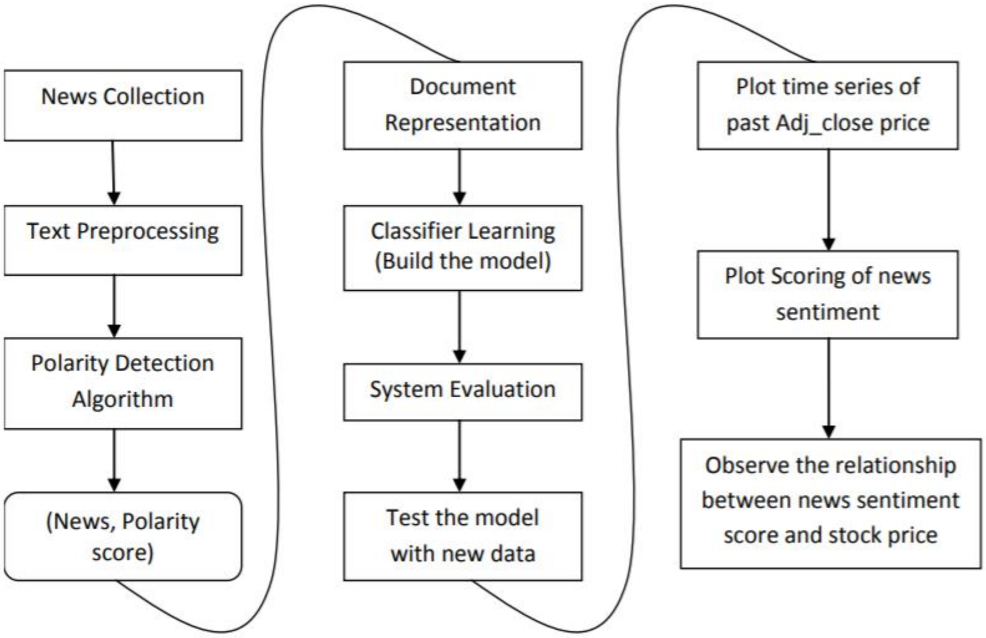

News Collection
Text Preprocessing
Polarity Detection Algorithm
(News, Polarity score)
Document Representation
Classifier Learning (Build the model)
System Evaluation
Test the model with new data
Plot time series of past Adj_close price
Plot Scoring of news sentiment
Observe the relationship between news sentiment score and stock price

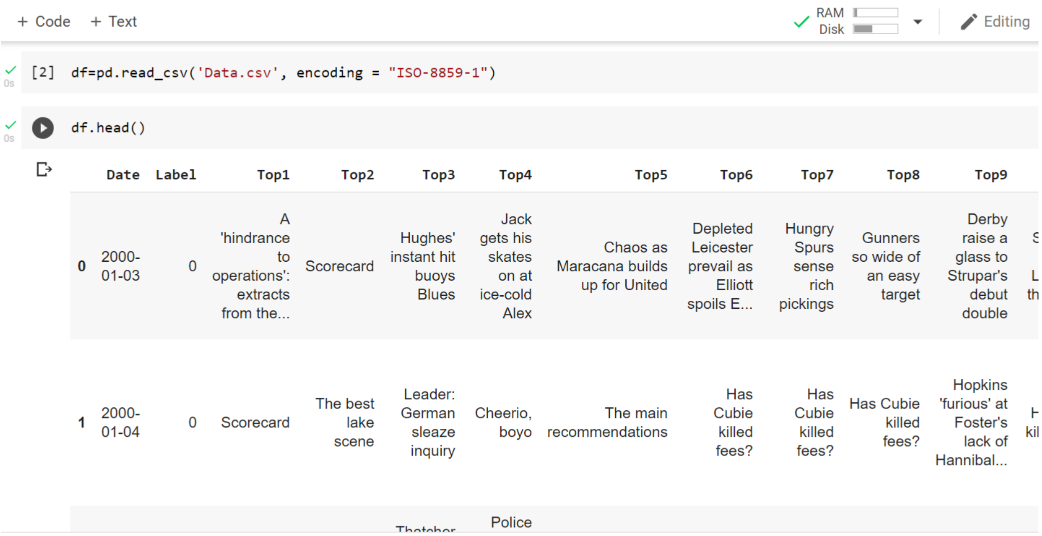
Fig 3 System Design

This design can be divided into three parts, with the first column of blocks representing phase 1, the second column representing phase 2, and the third column representing phase 3. Phase 1 produces news stories with polarity scores. Phase 2 receives this result as an input. Text is transformed to tf-idf vector space in phase 2 so that it may be fed to the classifier. Then, with the same data, Random forest classifiers are used to compare results. We analyse the findings provided by Random forest classifiers at the end of phase 2 and also test for classifier performance on new news articles. In the third phase, we look for a link between news stories and stock price data. We use a confusion matrix to plot both sets of data and keep track of the findings.

3.4.1 Data Preprocessing

We have taken a dataset which consist of news headlines. There are 25 columns of top news headlines for each day in data frame. Data ranges from 2000 to 2016. It involves cleaning the data collected. We create pickled data which includes the news headlines, label (0,1) and date. Data Preprocessing seems the most difficult task.

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

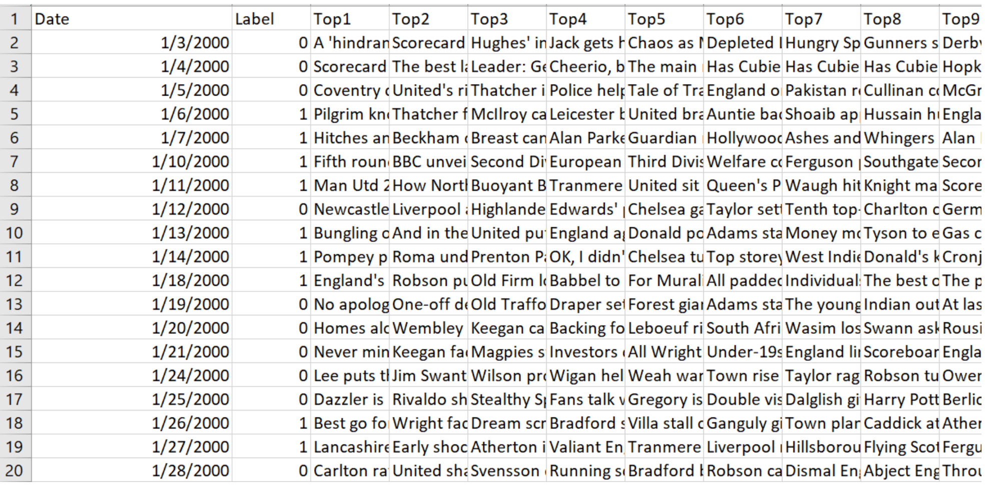


```
[2] df=pd.read_csv('Data.csv', encoding = "ISO-8859-1")
```

```
df.head()
```

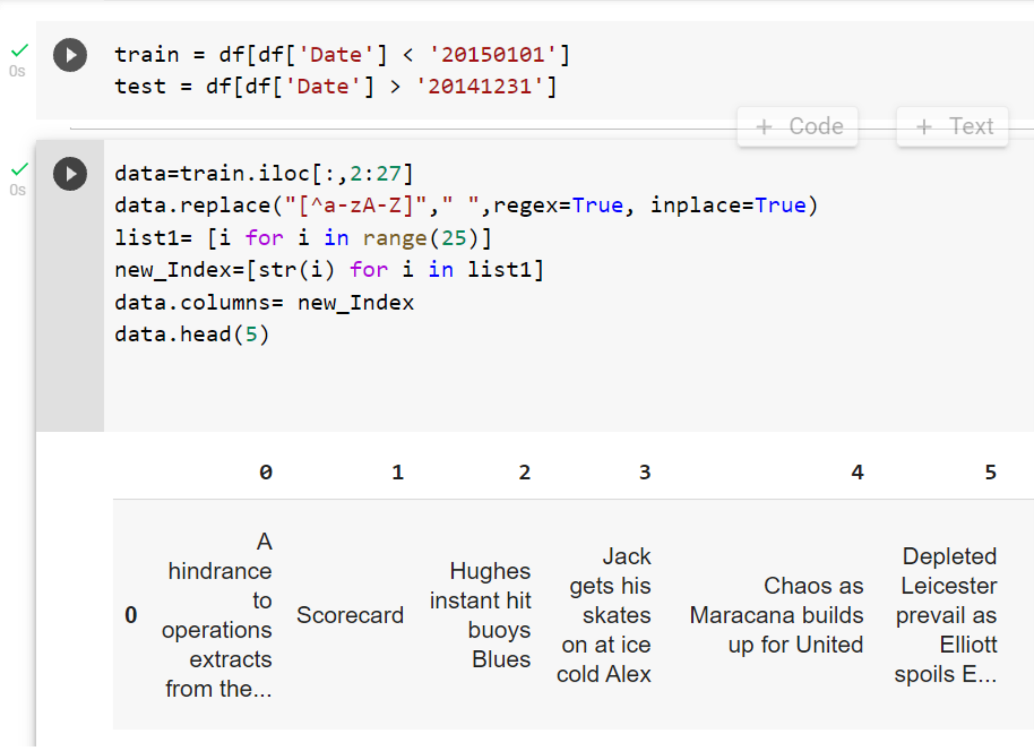
	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	
0	2000-01-03	0	A 'hindrance to operations': extracts from the...	Scorecard	Hughes' instant hit buoys Blues	Jack gets his skates on at ice-cold Alex	Chaos as Maracana builds up for United	Depleted Leicester prevail as Elliott spoils E...	Hungry Spurs sense rich pickings	Gunners so wide of an easy target	Derby raise a glass to Strupar's debut double	S L th
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?	Hopkins 'furious' at Foster's lack of Hannibal...	H kil
					Thatcher	Police						

Fig 4Dataset



1	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9
2	1/3/2000	0	A 'hindran	Scorecard	Hughes' in	Jack gets h	Chaos as M	Depleted L	Hungry Sp	Gunners s	Derby
3	1/4/2000	0	Scorecard	The best la	Leader: Ge	Cheerio, b	The main	Has Cubie	Has Cubie	Has Cubie	Hopk
4	1/5/2000	0	Coventry c	United's ri	Thatcher i	Police help	Tale of Tra	England o	Pakistan re	Cullinan co	McGr
5	1/6/2000	1	Pilgrim kn	Thatcher f	McIlroy ca	Leicester b	United bra	Auntie bac	Shoaib ap	Hussain hu	Engla
6	1/7/2000	1	Hitches an	Beckham c	Breast can	Alan Parke	Guardian	Hollywood	Ashes and	Whingers	Alan
7	1/10/2000	1	Fifth roun	BBC unvei	Second Di	European	Third Divis	Welfare co	Ferguson p	Southgate	Secor
8	1/11/2000	1	Man Utd 2	How North	Buoyant B	Tranmere	United sit	Queen's P	Waugh hit	Knight ma	Score
9	1/12/2000	0	Newcastle	Liverpool a	Highlande	Edwards' p	Chelsea ga	Taylor sett	Tenth top-	Charlton c	Germ
10	1/13/2000	1	Bungling c	And in the	United pu	England ag	Donald po	Adams sta	Money mo	Tyson to e	Gas c
11	1/14/2000	1	Pompey p	Roma und	Prenton Pa	OK, I didn'	Chelsea tu	Top storey	West Indie	Donald's k	Cronj
12	1/18/2000	1	England's	Robson pu	Old Firm lo	Babbel to	For Murali	All padded	Individuals	The best o	The p
13	1/19/2000	0	No apolog	One-off de	Old Traffo	Draper set	Forest gian	Adams sta	The young	Indian out	At las
14	1/20/2000	0	Homes alc	Wembley	Keegan ca	Backing fo	Leboeuf ri	South Afri	Wasim los	Swann ask	Rousi
15	1/21/2000	0	Never min	Keegan fac	Magpies s	Investors c	All Wright	Under-19s	England lir	Scoreboar	Engla
16	1/24/2000	0	Lee puts th	Jim Swant	Wilson pro	Wigan hel	Weah war	Town rise	Taylor rag	Robson tu	Ower
17	1/25/2000	0	Dazzler is	Rivaldo sh	Stealthy Sp	Fans talk v	Gregory is	Double vis	Dalglish gi	Harry Pott	Berlic
18	1/26/2000	1	Best go fo	Wright fac	Dream scr	Bradford s	Villa stall c	Ganguly gi	Town plar	Caddick at	Ather
19	1/27/2000	1	Lancashire	Early shoc	Atherton i	Valiant En	Tranmere	Liverpool r	Hillsborou	Flying Scot	Fergu
20	1/28/2000	0	Carlton ra	United sha	Svensson c	Running sc	Bradford b	Robson ca	Dismal En	Abject Eng	Throu

Fig 5 Dataset containing different headlines



```
train = df[df['Date'] < '20150101']
test = df[df['Date'] > '20141231']
```

```
data=train.iloc[:,2:27]
data.replace("[^a-zA-Z]"," ",regex=True, inplace=True)
list1= [i for i in range(25)]
new_Index=[str(i) for i in list1]
data.columns= new_Index
data.head(5)
```

	0	1	2	3	4	5
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...

Fig 6 Data Preprocessing

3.4.2 Sentiment Detection Algorithm

To identify automated sentiment in news stories, we use a Dictionary-based system that leverages the Bag of Word technique for text mining. We'll need two types of word collections to build the polarity dictionary: positive and negative words. Then we can compare the words in the article to both of these word lists, count the number of terms in each dictionary, and get the document's score. The polarity terms dictionary was created using the labels 0 and 1. In addition, we examine the string that comprises both the title and the news body for the news piece.

Algorithm

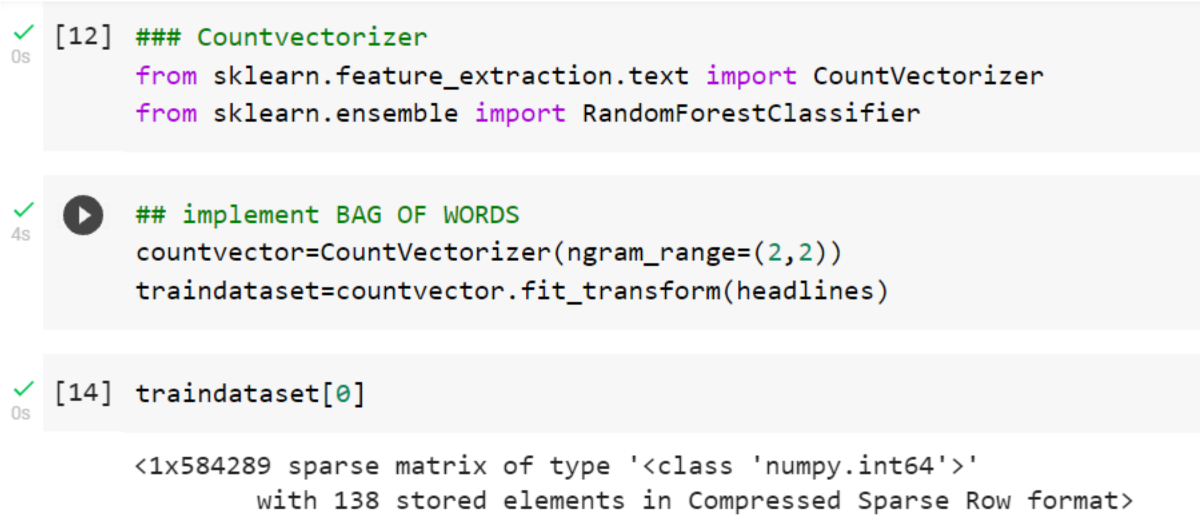
1. Tokenize the document into word vector.

2. Make a dictionary with terms that have the same polarity as the polarity (positive or negative)

3. Examine each word to see whether it fits one of the terms in either the positive or negative word dictionaries.

4. Count the number of positive and negative polarity terms.

5. As a consequence, we have a news collection with a sentiment score and positive or negative polarity.



```
### Countvectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
```

```
## implement BAG OF WORDS
countvector=CountVectorizer(ngram_range=(2,2))
traindataset=countvector.fit_transform(headlines)
```

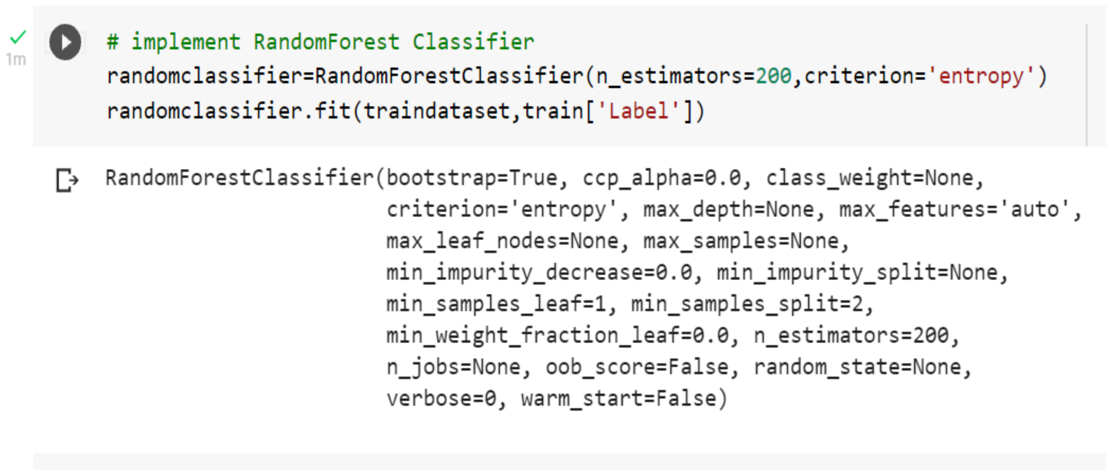
```
traindataset[0]
```

```
<1x584289 sparse matrix of type '<class 'numpy.int64'>'
	with 138 stored elements in Compressed Sparse Row format>
```

Fig 7 Vectorization

3.4.3 Classifier Learning

Random Forest and Nave Bayes classification algorithms perform well in text classification, according to the majority of research. As a result, we're using Random Forest to classify the text and assess the accuracy. All of the outcomes, such as accuracy, precision, recall, and other model evaluation methods, can be compared.The data was split into two groups: training and testing. We assessed the performance of classifier by examining their accuracy, precision, and recall.



```
# implement RandomForest Classifier
randomclassifier=RandomForestClassifier(n_estimators=200,criterion='entropy')
randomclassifier.fit(traindataset,train['Label'])
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=200,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Fig 8 Implementation of Classifier

3.5 Flow Chain of Model

A flowchart is a diagram that shows how a workflow or process works. A flowchart is a diagrammatic depiction of an algorithm, or a step-by-step procedure for completing a job. The flowchart depicts the stages as various types of boxes, with arrows linking the boxes in a logical order.

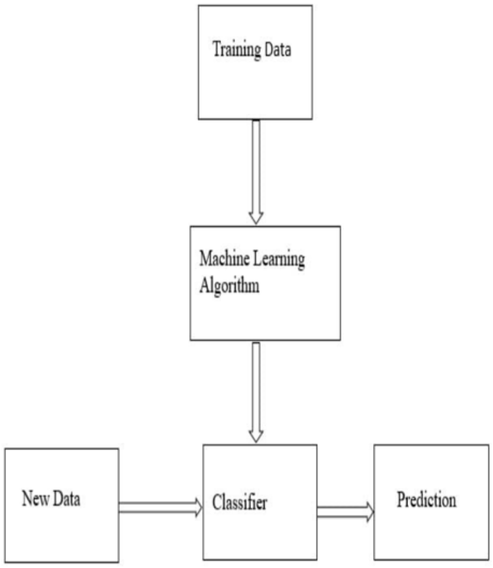


Fig 9 Flow Chart

3.6 Component Diagram

Component diagrams are used to depict the physical features of object-oriented systems. They are used for visualising, describing, and documenting component-based systems, as well as for forward and reverse engineering to create executable systems. Component diagrams are basically class diagrams that focus on the components of a system and are frequently used to describe a system's static implementation perspective.

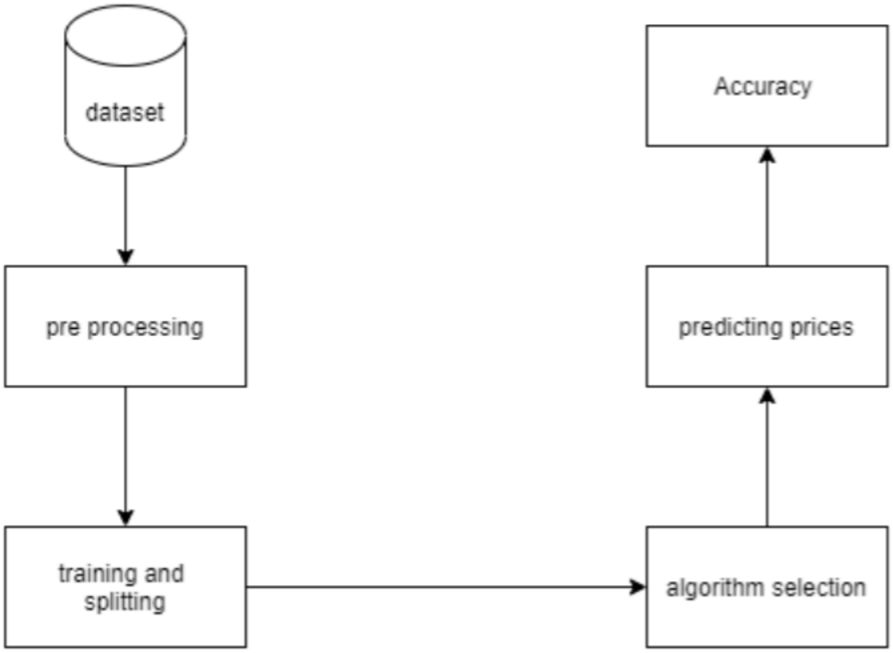


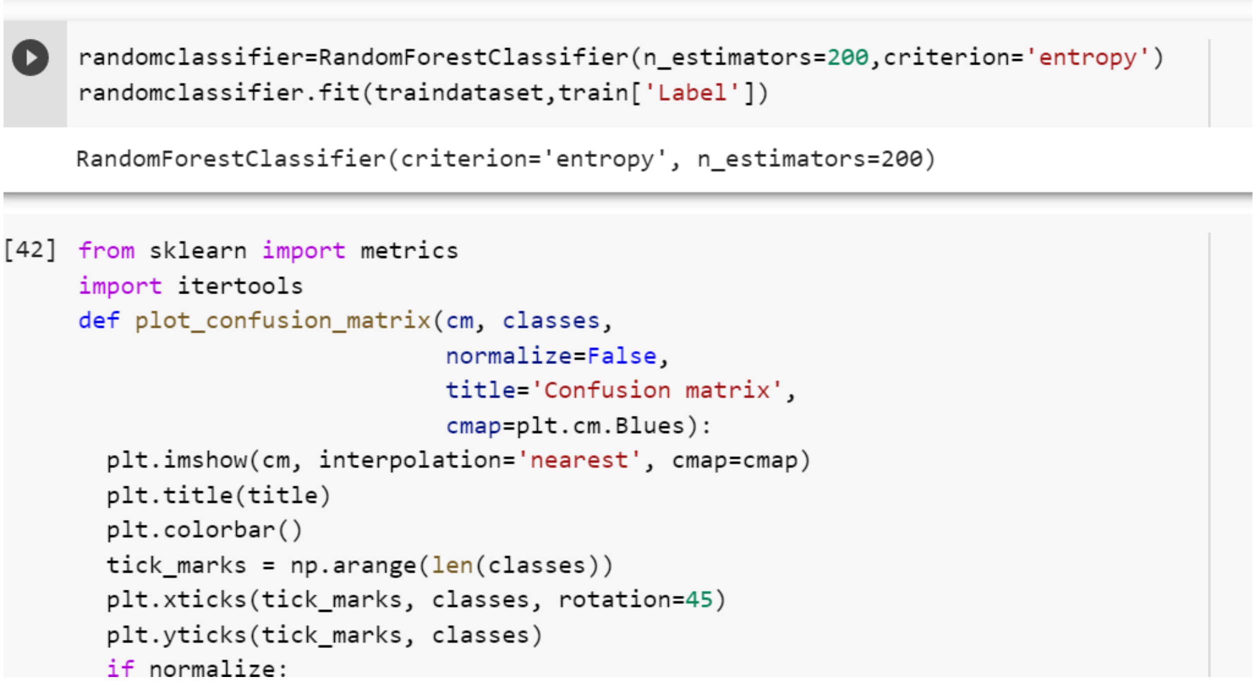
Fig 10 Components Present in System

Chapter 04:- PERFORMANCE ANALYSIS

4.1 Dataset

We have taken a dataset and data was collected from Kaggle.com which consist of news headlines. There are 25 columns of top news headlines for each day in data frame. Data ranges from 2000 to 2016.

4.2 Performance Evaluation



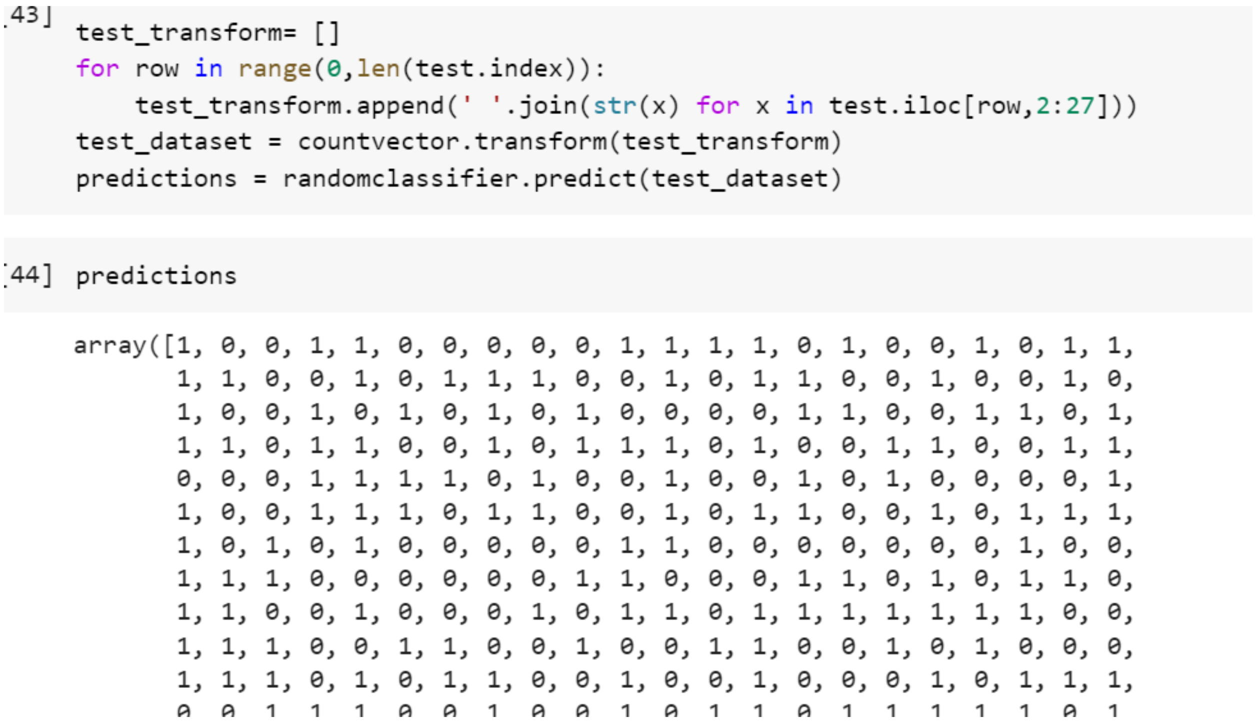
```
randomclassifier=RandomForestClassifier(n_estimators=200,criterion='entropy')
randomclassifier.fit(traindataset,train['Label'])
```

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

```
[42] from sklearn import metrics
     import itertools
     def plot_confusion_matrix(cm, classes,
                               normalize=False,
                               title='Confusion matrix',
                               cmap=plt.cm.Blues):
       plt.imshow(cm, interpolation='nearest', cmap=cmap)
       plt.title(title)
       plt.colorbar()
       tick_marks = np.arange(len(classes))
       plt.xticks(tick_marks, classes, rotation=45)
       plt.yticks(tick_marks, classes)
       if normalize:
```

Fig 11 Normalization

In fig 11 we have implemented normalization to evaluate the results.Normalization is a data preparation technique that is frequently used in machine learning. Normalization is the process of converting the values of numeric columns in a dataset to a similar scale without distorting the ranges of values or losing information.

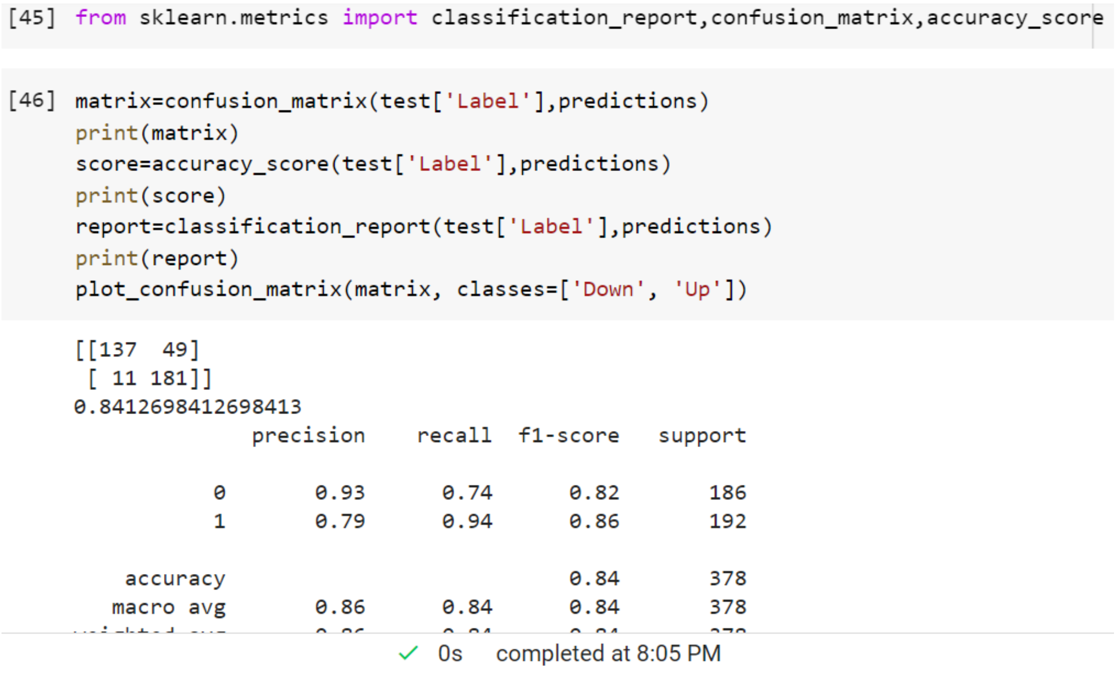


```
43]  test_transform= []
     for row in range(0,len(test.index)):
         test_transform.append(' '.join(str(x) for x in test.iloc[row,2:27]))
     test_dataset = countvector.transform(test_transform)
     predictions = randomclassifier.predict(test_dataset)
```

```
44]  predictions
```

```
array([1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1,
       1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1,
       1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1,
       0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1,
       1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0,
       1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0,
       1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1,
```

Fig 12 Predictions for test and train data set



```
[45]  from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

```
[46]  matrix=confusion_matrix(test['Label'],predictions)
      print(matrix)
      score=accuracy_score(test['Label'],predictions)
      print(score)
      report=classification_report(test['Label'],predictions)
      print(report)
      plot_confusion_matrix(matrix, classes=['Down', 'Up'])
```

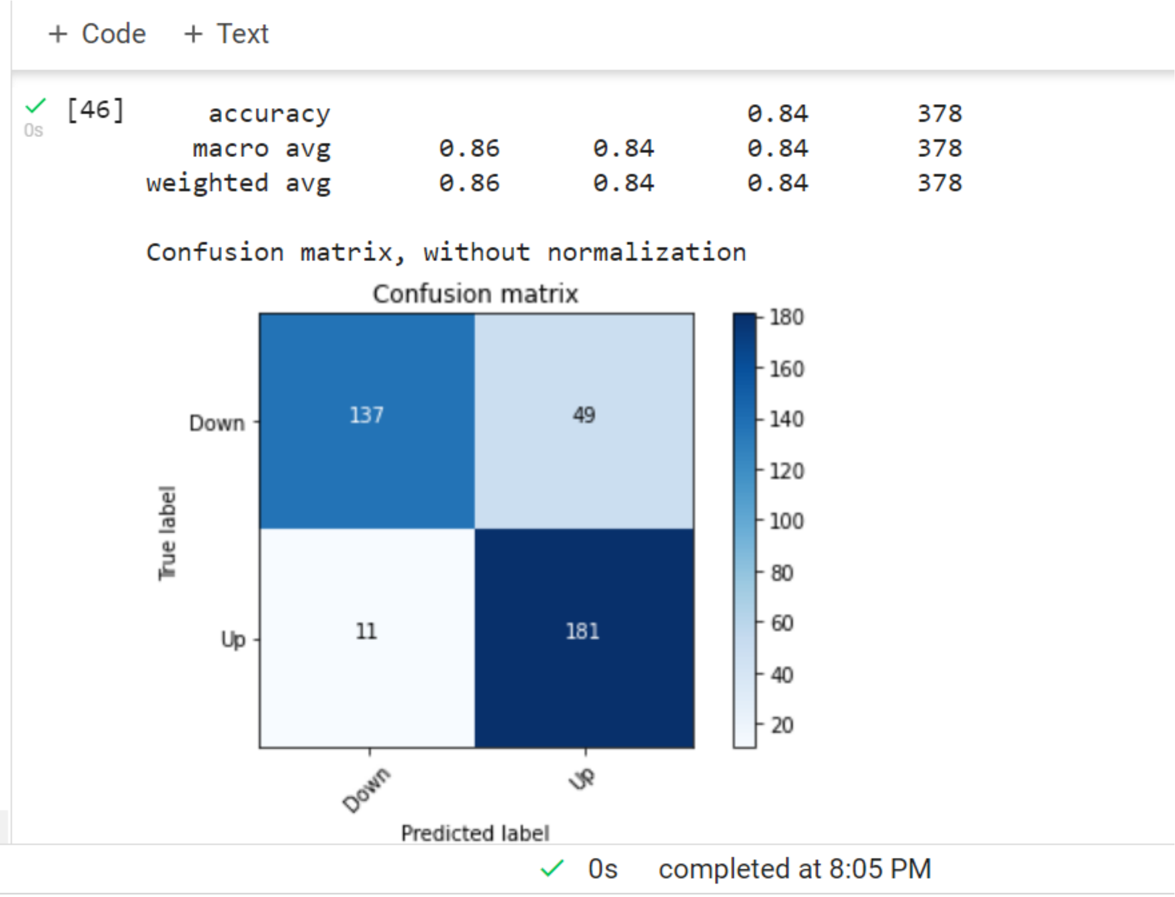
```
[[137  49]
 [ 11 181]]
0.8412698412698413
              precision    recall  f1-score   support

           0       0.93      0.74      0.82       186
           1       0.79      0.94      0.86       192

    accuracy                           0.84       378
   macro avg       0.86      0.84      0.84       378
weighted avg       0.86      0.84      0.84       378
```

✓ 0s completed at 8:05 PM

Fig 13Calculating Results



```
[46]     accuracy                           0.84       378
        macro avg       0.86      0.84      0.84       378
     weighted avg       0.86      0.84      0.84       378

     Confusion matrix, without normalization
```

Fig 14 Confusion Matrix

In figure 12, figure 13 and figure14 we completely evaluate the results and got result up to mark and finally plotted the confusion matrix to show the accuracy.

Chapter 05:- CONCLUSION

5.1 Conclusion

Finally, we'll remark that this project as stock market plays important role in today’s world. Because the stock market is random and tightly associated with real-time occurrences, some academics feel it is difficult to predict the stock market return. No one can predict the future. This paper intends to highlight and put into practice a method for attempting to forecast and using it as a supplement to human decision-making. We can tell from the tests in this report that the forecasts don't have a lot of precision, but from a long-term perspective, the results point in the right direction. We may utilize these findings in conjunction with human experience to make financial decisions.By using sentimental as new feature with relatively high weight in Random Forests model, it may improve the accuracy significantly.

5.2 Future Scope

In today's digital environment, the vast bulk of data is spread and abused. By understanding the existing data, we can apply it to previously unknown patterns. To assess and predict, machine learning techniques such as logistic regression, Naive-Bayes, and Random Forest Classifier can be employed. Future enhancements will include making the stock prediction and prediction results application more user-friendly in order to boost user involvement and the adoption of enhanced approaches and algorithms in order to improve the outcomes and their accuracy.

REFERENCES

1 Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, Giridhar Maji, Narayan C. Debnath, Soumya Sen

2 Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, “An innovative neural network approach for stock market prediction”, 2018.

3 Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari SahooDepartment of Computer Science & Information Systems, BITS Pilani K.K.Birla Goa campus, India bDivision of Mathematical Sciences, Nanyang Technological University, Singapore cDepartment of Mathematics, BITS Pilani K.K.Birla Goa campus, India - Forecasting directional movements of stock prices for intraday trading using LSTM and random forests.

4 Lavanya Ra SRM Institute of Science and Technology | SRM · Department of Computer Science - Stock Market Prediction.

5 Sentiment Analysis on Twitter with Stock Price and Significant Keyword Correlation http://apps.cs.utexas.edu/tech_reports/reports/tr/TR-2124.pdf

6 Pang and L.Lee.2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In ACL-2004.

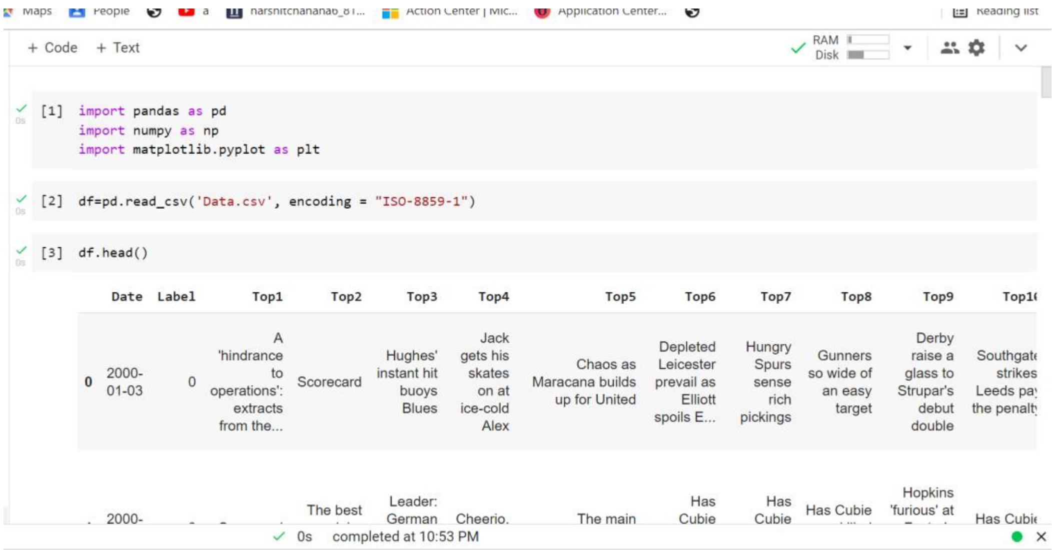
7 IEEE “Machine Learning Methods Used in Ailment” by www.Wikipedia.com.

8 R. Ahuja, H. Rastogi, A. Choudhuri and B. Garg, Stock Market Forecast Using Sentiment Analysis, In 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1008–1010, March (2015).

9 Anurag Nagar, Michael Hahsler, Using Text and Data Mining Techniques to extract Stock Market Sentiment from Live News Streams, IPCSIT vol. XX (2012) IACSIT Press, Singapore.

10 https://en.wikipedia.org/wiki/Random-Forest-Classifier

APPENDICES



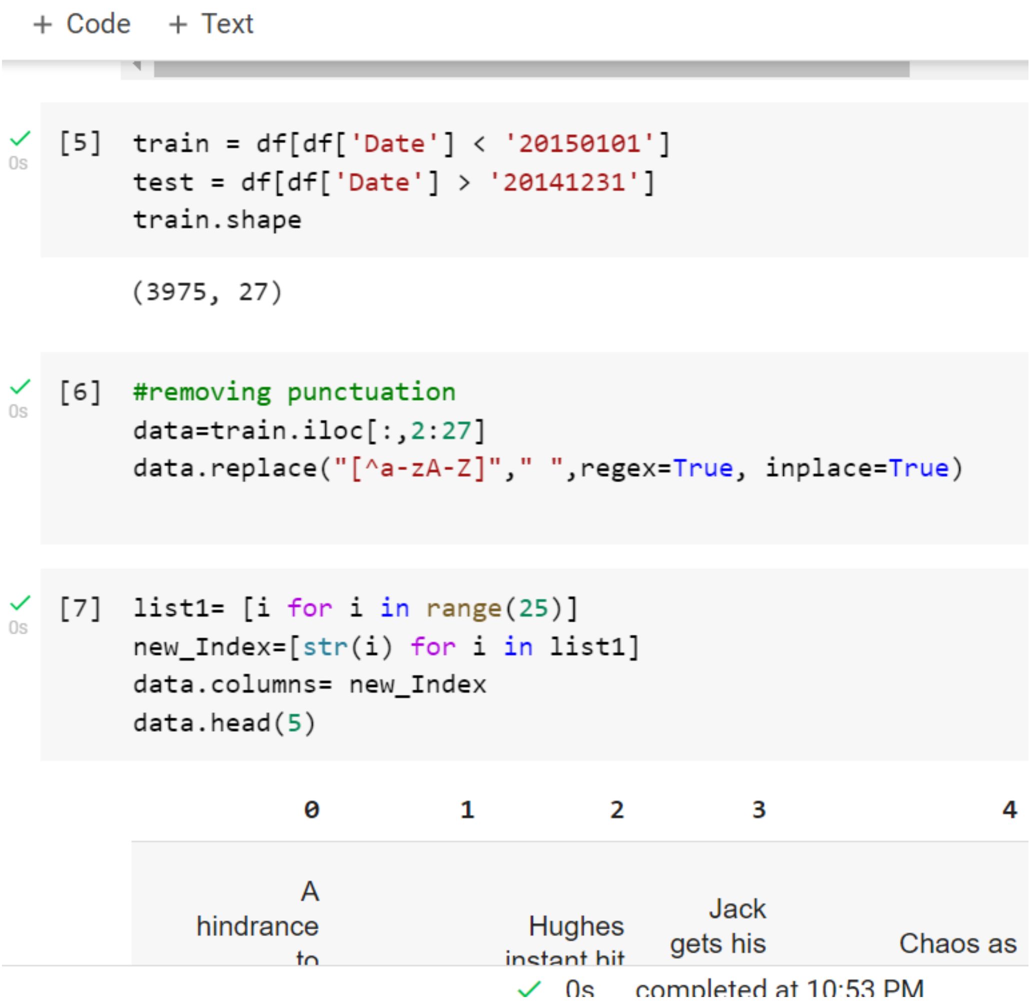
```
[1] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
```

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

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

	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	Top1(
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	Derby raise a glass to Strupar's debut double	Southgate strikes Leeds pay the penalty
	2000-			The best	Leader: German	Cheerio.	The main	Has Cubie	Has Cubie	Has Cubie	Hopkins 'furious' at	Has Cubie

✓ 0s completed at 10:53 PM



```
[5] train = df[df['Date'] < '20150101']
    test = df[df['Date'] > '20141231']
    train.shape
```

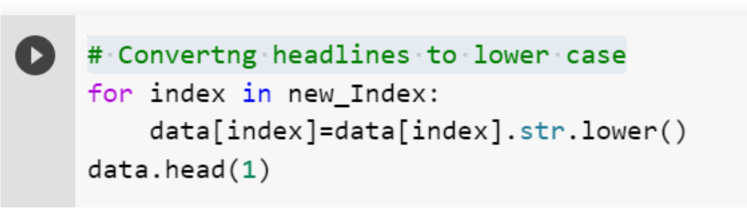
```
(3975, 27)
```

```
[6] #removing punctuation
    data=train.iloc[:,2:27]
    data.replace("[^a-zA-Z]"," ",regex=True, inplace=True)
```

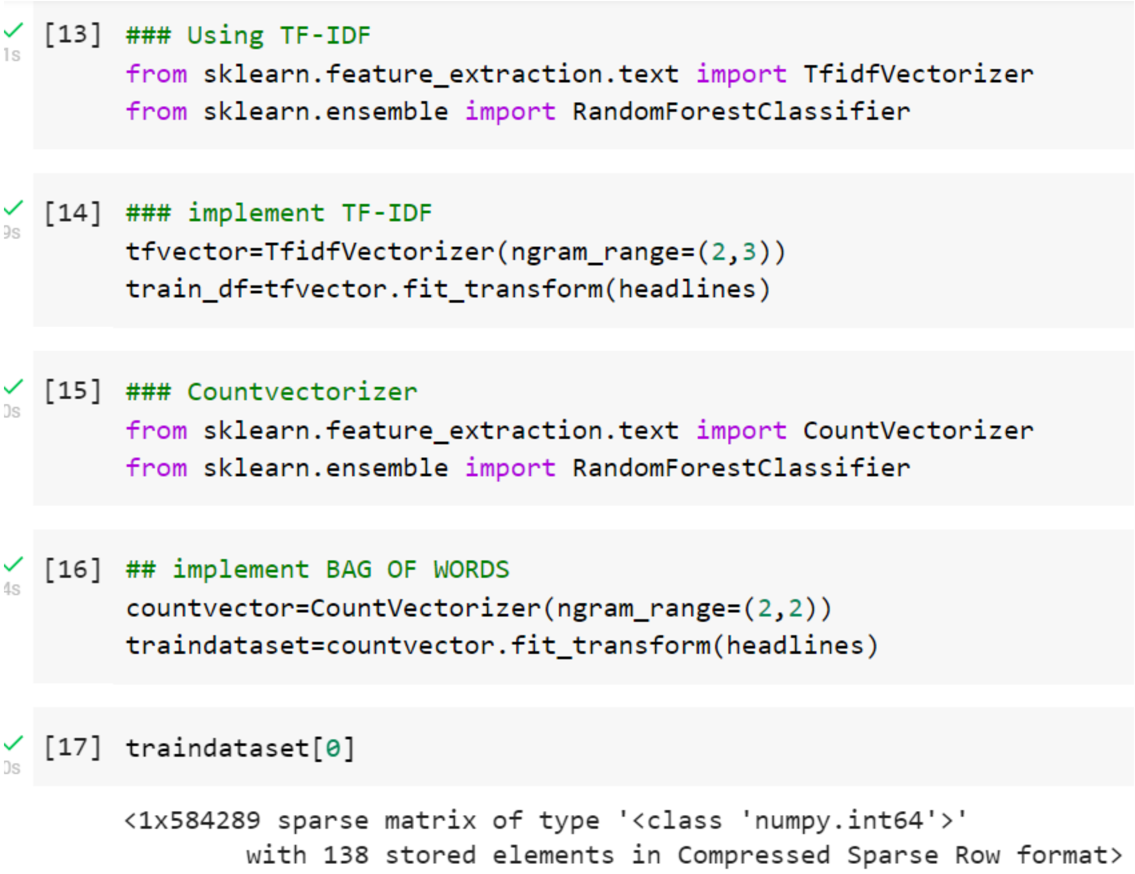
```
[7] list1= [i for i in range(25)]
    new_Index=[str(i) for i in list1]
    data.columns= new_Index
    data.head(5)
```

0	1	2	3	4
A hindrance to		Hughes instant hit	Jack gets his	Chaos as

✓ 0s completed at 10:53 PM



```
# Convertng headlines to lower case
for index in new_Index:
    data[index]=data[index].str.lower()
data.head(1)
```



```
[13] ### Using TF-IDF
     from sklearn.feature_extraction.text import TfidfVectorizer
     from sklearn.ensemble import RandomForestClassifier
```

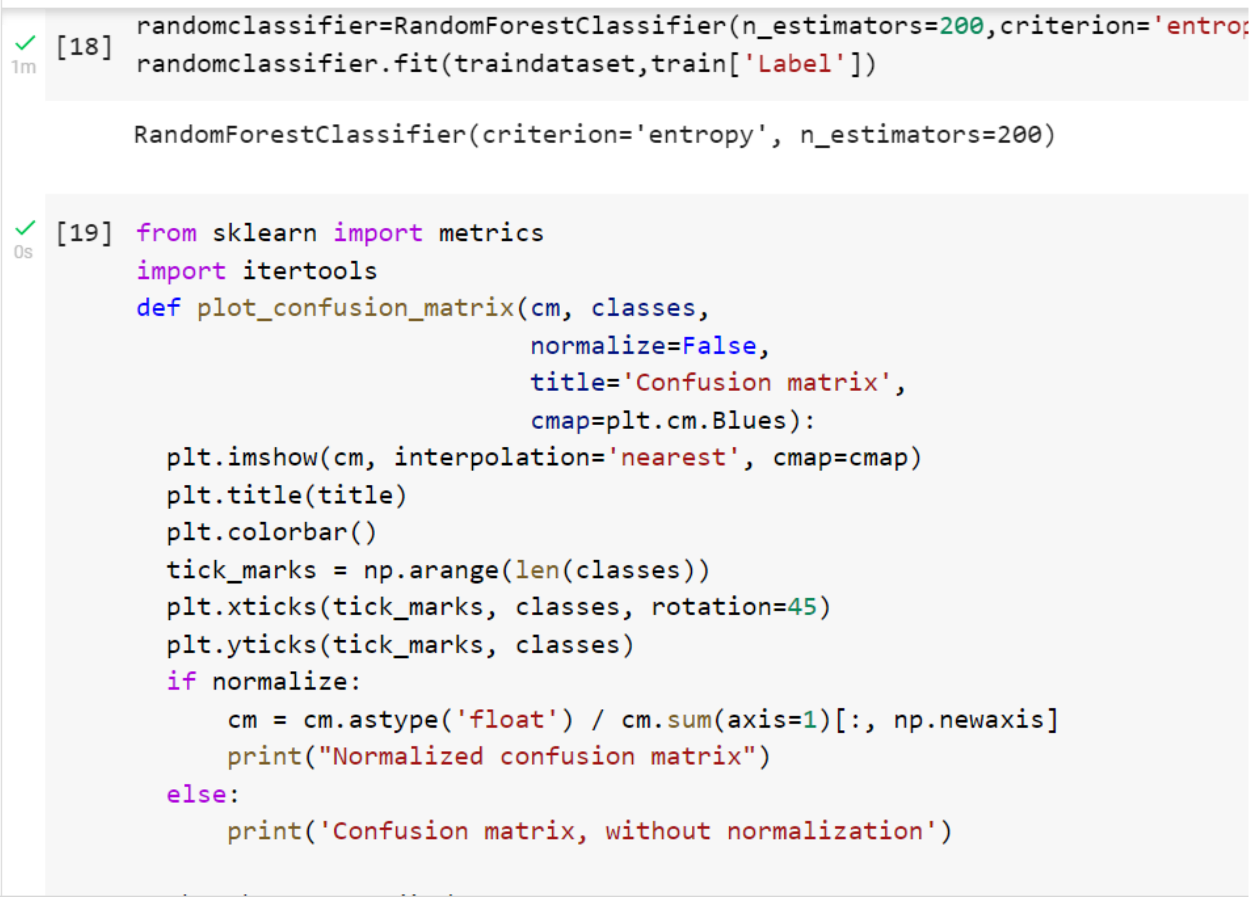
```
[14] ### implement TF-IDF
     tfvector=TfidfVectorizer(ngram_range=(2,3))
     train_df=tfvector.fit_transform(headlines)
```

```
[15] ### Countvectorizer
     from sklearn.feature_extraction.text import CountVectorizer
     from sklearn.ensemble import RandomForestClassifier
```

```
[16] ## implement BAG OF WORDS
     countvector=CountVectorizer(ngram_range=(2,2))
     traindataset=countvector.fit_transform(headlines)
```

```
[17] traindataset[0]
```

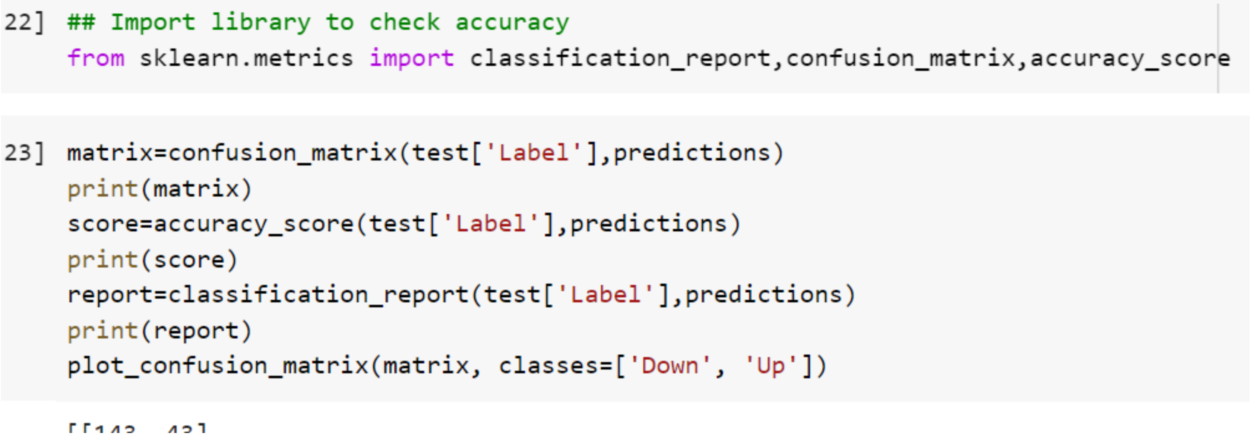
```
<1x584289 sparse matrix of type '<class 'numpy.int64'>'
	with 138 stored elements in Compressed Sparse Row format>
```



```
[18] randomclassifier=RandomForestClassifier(n_estimators=200,criterion='entrop
     randomclassifier.fit(traindataset,train['Label'])

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

```
[19] from sklearn import metrics
     import itertools
     def plot_confusion_matrix(cm, classes,
                               normalize=False,
                               title='Confusion matrix',
                               cmap=plt.cm.Blues):
       plt.imshow(cm, interpolation='nearest', cmap=cmap)
       plt.title(title)
       plt.colorbar()
       tick_marks = np.arange(len(classes))
       plt.xticks(tick_marks, classes, rotation=45)
       plt.yticks(tick_marks, classes)
       if normalize:
           cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
           print("Normalized confusion matrix")
       else:
           print('Confusion matrix, without normalization')
```



```
22] ## Import library to check accuracy
    from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

```
23] matrix=confusion_matrix(test['Label'],predictions)
    print(matrix)
    score=accuracy_score(test['Label'],predictions)
    print(score)
    report=classification_report(test['Label'],predictions)
    print(report)
    plot_confusion_matrix(matrix, classes=['Down', 'Up'])

    [[143  43]
```