Szkoła Główna Gospodarstwa Wiejskiego
w Warszawie

Wydział Zastosowań Informatyki i Matematyki

Bhargav Kasundra
195606

Analiza nastrojów giełdowych za pomocą uczenia maszynowego nagłówków wiadomości

Stock Sentiment Analysis using News Headlines Machine Learning

Praca dyplomowa magisterska

na kierunku – Informatyka i Ekonometria
Big Data Analytics

Praca wykonana pod kierunkiem
Dr LUIZY OCHNIO
Instytut Ekonomii i Finansów
Katedra Ekonometrii i Statystyki

Warszawa, 2021

Oświadczenie promotora pracy

Oświadczam że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia warunki do przedstawienia tej pracy w postępowaniu o nadanie tytułu zawodowego.

Data .... Podpis promotora pracy ....

Oświadczenie autora pracy

Świadom odpowiedzialności prawnej, w tym odpowiedzialności karnej za złożenie fałszywego oświadczenia, oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami prawa, w szczególności ustawą z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. Nr 90 poz. 631 z późn. zm.)

Oświadczam, że przedstawiona praca nie była wcześniej podstawą żadnej procedury związanej z nadaniem dyplomu lub uzyskaniem tytułu zawodowego.

Oświadczam że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Przyjmuję do wiadomości, że praca dyplomowa poddana zostanie procedurze antyplagiatowej.

Data .... Podpis autora pracy...

Abstract

Stock sentiment analysis using news headlines machine learning

News headlines are one of the most important factors in stock market that influence the most. News headlines and stock market price have a direct relation. News headline plays a major role in the stock price fluctuation. In this research, sentiment analysis has done on more than 100000 news headlines and predict the whether the stock price will rise up or down using machine learning. 7 Machine learning algorithms and 2 natural language processing techniques at their default parameter are used to get the predictions. A high level of programming techniques and 14 different machine learning algorithm's combinations were experimented to get best results. Seven powerful machine learning algorithm with 2 natural language processing technique, in total 14 combination were used in this study. In this thesis there were analyzed news headlines data and made predication whether the stock price would increase or decrease using machine learning techniques. The stock price goes up and down based on daily news headlines. So there is direct relation between stock prices a news headlines. Using machine learning algorithms and Python programming language it was made 14 models, which were able to predict whether stock price would go up or down. This study shows the effect of emotion sentiment of financial news to the stock market prices.

Keywords: Machine Learning, Sentiment Analysis, Python, Natural language processing, classification algorithms, sklearn

Abstrakt

Analiza nastrojów giełdowych z wykorzystaniem uczenia maszynowego nagłówków wiadomości

Nagłówki wiadomości to jeden z najważniejszych czynników na giełdzie, który ma największy wpływ. Nagłówki wiadomości i cena giełdowa mają bezpośredni związek. Nagłówki wiadomości odgrywają główną rolę w wahaniach cen akcji. W tym badaniu analiza nastrojów obejmowała ponad 100000 nagłówków wiadomości i przewidywała, czy cena akcji wzrośnie lub spadnie za pomocą uczenia maszynowego. 7 Algorytmy uczenia maszynowego i dwie techniki przetwarzania języka naturalnego z domyślnymi parametrami są wykorzystywane do uzyskiwania predykcji. Aby uzyskać najlepsze wyniki, przetestowano wysoki poziom technik programowania i 14 różnych kombinacji algorytmów uczenia maszynowego. W badaniach wykorzystano 7 algorytmów uczenia maszynowego z 2 technikami przetwarzania języka naturalnego, łącznie 14 kombinacji. W tej pracy przeanalizowano dane z nagłówków wiadomości i przewidywano, czy kurs akcji wzrośnie, czy spadnie, korzystając z technik uczenia maszynowego. Cena akcji rośnie i spada na podstawie codziennych nagłówków wiadomości. Jest więc bezpośredni związek ceny akcji z nagłówkami wiadomości. Korzystając z algorytmów uczenia maszynowego i Pythona stworzono 14 modeli, które są w stanie przewidzieć, czy cena akcji wzrośnie, czy spadnie. Badanie to zasadniczo pokazuje wpływ nastrojów wiadomości finansowych na ceny giełdowe.

Słowa kluczowe: uczenie maszynowe, analiza nastrojów, Python, przetwarzanie języka naturalnego, algorytmy klasyfikacji, sklearn

Acknowledgement

I thank God for giving me protection and ability to finish my work. Furthermore, I am very grateful to the Warsaw University of Life Sciences and the Faculty of Mathematics and Informatics for making it possible for me to study here. I give deep thanks to the Professors and lecturers at the Big Data Analytics program for their assistance so I am able to complete this study. Moreover, I want to express gratitude to Dr. M. Wierzbicki, my lecturer in Master Seminar, and Prof. Dr. P. J. Friedrich who had guided me to finish my study. My special and hearty thanks are for my supervisor, Dr LUIZA OCHNIO who encouraged and directed me. Her challenges brought this work toward a completion. It is with her supervision that this work came into existence. For any faults, I take full responsibility.

Table of Contents

Chapter 1 Conceptual Framework

1.1 Introduction

To trade in a financial market it is important to keep an eye on news headlines. There is a direct relation between news headlines and stock price. News headline plays an important role in stock market price fluctuations. Economic, political, or industrial news affects a lot of people's sentiment towards a particular company's stock price. News headlines can create a positive or negative sentiment on people's minds based on the people decide to buy or not to buy a particular stock. There is a strong and complicated relationship between the market and the information available in the form of news. News at any moment can change the perception particular company. News of some change in a company's policy or change in company's strategy can affect a lot in the company's stock price. These days because of the blessing of the internet the investors and traders have a constant eye on updated news. The news constantly molds their sentiment and influences them to invest in a particular company. News from standard authentic sources influences them a lot to invest or not to invest in a particular stock. When traders having a positive sentiment then they find that a company is worth to invest, so the company's growth also gets boosted up. On the other hand, if the traders having a negative perception about a particular company then they don't feel safe investing in a company and they withdraw their investments to avoid loss. So because of which the stock price of the company goes down drastically. News headlines have always been an important and reliable source of information to build a perception of stock market investments.

1.2 Emerge and Justification of the problem

As the volumes of news are increasing rapidly, it’s becoming very difficult or nearly impossible for an investor or even a group of investors to find out relevant news for them and find some meaningful information from it. To solve 21 century’s problem we need some latest 21st century's technologies also. A single human cannot process and implement these hectic processes, that's why machine learning comes in to picture. With the help of high computing algorithms and powerful programs, this task can be possible. This thesis presents some machine learning techniques to find sentiment behind the news headline and it will predict whether it is positive or it's negative. Based on available news headline data learning models predict the sentiment which they made.

1.3 Research Questions

In machine learning, there are 7 types of classification algorithms, Logistic Regression,

Naïve Bayes, Stochastic Gradient Descent, K-Nearest Neighbours, Decision Tree, Random Forest, and Support Vector Machine. Based on that this research some question can be raised:

1. Identify the best machine learning algorithm for stock sentiment analysis

There are many natural language processing techniques available to convert text data into the numerical format. Here in this research 2 of them - Bag of Words and TF-IDF were used, which are quite popular in the data science industry, so based on that this research question can be answered.

2. Identify the best technique among Bag of Words and TF-IDF for convert text news data into a vector?

Based on the 2 mentioned methods of convert text data into a vector and 7 machine learning classifiers 14 unique combinations can be applied. Based on that this research question can be answered.

3. How to achieve the best accuracy?

By identifying the best combination of natural language processing techniques with a machine learning classifier algorithm we can achieve the best accuracy.

1.4 Definition of terms used in the study

Machine learning. Is a study of mathematics and computer algorithms that improves itself by experience and data (Wikipedia, 2021).

Sentiment Analysis. A process of identifying or cauterizing opinions present in the text, basically to identify the attitude towards a particular topic either it is positive, negative, neutral (Zhan, 2015).

Data Visualization. It is a graphical representation of data. To understand data in a simple way we make a chart graph map of the data (Gubarev, 2013).

Data pre-processing. It allows to clean unusual or missing data or modify data before doing the final process on data (Wu, 2016).

Python is a high-level programming language, it is a general-purpose programming language that uses an interpreter (Owino, 2019).

IDE. It is an integrated development environment means a software application that provides facilities to computer programmers for software development. (Wikipedia, 2020)

Jupyter Notebook. Is an open-source web application or IDE that allows you to create and share documents that contain live code (Anon., 2020).

CSV Data. A comma-separated values file is a delimited text file that uses a comma to separate the values. In this file, each line is a data record. Each record has one or more fields and it's separated by commas (Anon., 2020).

Python Libraries. Are a set of useful functions. With the help of that, we can eliminate the need for writing codes from scratch. There are over 137,000 python libraries present today. Python libraries play a huge role in developing machine learning, data science, data visualization, and more (Advani, 2020).

Machine learning algorithms. Are the engine of machine learning, it means it turns data into models (Heller, 2019).

Machine learning models. A machine learning model is a file trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data (Cowley, 2019).

Classification machine learning problem. In machine learning classification is refers to a predictive modeling problem in which a class label is predicted for a given input data. For example classify if it is spam or not (Brownlee, 2020).

Dataset. “A collection of data that are treated as a single unit by a computer”. This means that a dataset contains several separate pieces of data but can be used to train an algorithm with the goal of finding predictable patterns inside the whole dataset (Sydorenko, 2021).

Countvectorizer. It is a function that is used to transform a given text into a vector, On the basis of their frequency of each word that occurs in the entire text or document. It is a great tool provided by the scikit-learn library in Python (Sharma, 2020).

1.5 The objective of the study

In this study 7, machine learning algorithms with 2 text pre-processing techniques in total 14 combinations were experimented at their default parameter to identify which combination performs the best. In this study, machine learning models were built to classify the sentiment of news headlines. Basically, a machine learning algorithm learns from the data which is given to the model, and based on data it will predict results in 0 and 1 .o means positive and 1 means negative. Such that 14 different machine learning models were built based on various mathematical algorithms. In the end, all of them compared with the actual result and identified which one is the best. This study is to find the best machine learning model for sentiment analysis of the news headlines. Identify whether the stock price will increase or decrease based on the positive or negative sentiment of news headlines. If the headline has positive sentiment then the stock price will increase and if the news headline have negative sentiment then the stock price will decrease. Word news dataset is used in the study in which it has news headlines of a particular day and labels for a particular day. Label says it is positive or negative sentiment. This data has been given to 14 different machine learning algorithms to find the best result. The main goal is to identify the best machine learning algorithm and text pre-processing technique combination for the sentimental analysis of world news data.

For doing sentimental analysis 7 machine learning algorithms and 2 text processing techniques were experimented with to achieve the best results. Based on the study and experiment which are done during research these were specific or sub-objectives.

1. In this study, it was implemented 7 machine learning techniques based on that it could be identified which was the best machine learning technique among those.

2. Two techniques for converting text data into numeric format were used during this study so based on the result it could be identified which one was the best.

3. 14 different combination of machine learning algorithms and text to vector technique was used during the study, so based on that it is important to identify which was the best combination.

1.6 Hypothesis

According to the Anita Kumari journal’s article TF-IDF is giving more accurate results compared with Bag of Words (Anita Kumari Singh, 2019). TF-IDF is a more advanced technique than Bag of Words. So the null hypothesis of the study is as follow

H0:- TF-IDF will give more accurate results than Bag of Words.

H1:- Bag of Words will give more accurate results than TF-IDF.

The magazine of analytics India mag’s article shows a comparison of machine learning algorithms (Garg, 2018). The purpose of his research was to put together the 7 most common types of classification algorithms along with the Python code: logistic regression, naïve Bayes, stochastic gradient descent, K-nearest Neighbors, decision tree, random forest, and support vector machine. According to this study we get that SVM, Gradient Descent is a high-performance algorithm and the rest are low-performance algorithms, so according to that SVM and Gradient Boosting should give higher accuracy than others. Based on this study following hypotheses were created.

H0: SVM and Gradient Boosting will give higher accuracy than other classification algorithms

H1: Logistic Regression, Naïve Bayes, K-Nearest Neighbors, Decision Tree, Random Forest will give higher result than SVM and Gradient Boosting

1.7 Delimitations of the study

In this study word news dataset used. This research was limited to this dataset only. The GitHub link with data used in this research can be found with following link.

https://github.com/bhargavkasundra/stock-sentiment/blob/main/Data.csv

Dataset have three type of data. First date, second label and third news headlines. Date column basically contains date of particular news headline. Date column contains data from 03-01-2000 to 01-07-201. In total 4101 days of records. Label column contains value of 0 and 1. Label 0 means positive and label 1 means negative. 0 indicates price of the stock will increase because sentiment of news his positive and 1 indicates stock price will decrease because news headline have negative sentiment. News headlines are in columns name top1 to top 25, it means for every day there are 25 headlines. This data set is combination of World news and stock price shifts which is available on Kaggle. Kaggle is an online community of Data Scientists and Machine Learning Practitioners. Kaggle allows user to find and publish datasets, and build Machine learning models on web based environment. It allows to work with other data scientists and data engineers, Kaggle makes competitions for solving data science challenges. This data set is one of the data set which Kaggle publish in one of their competition (wikipedia, 2010).

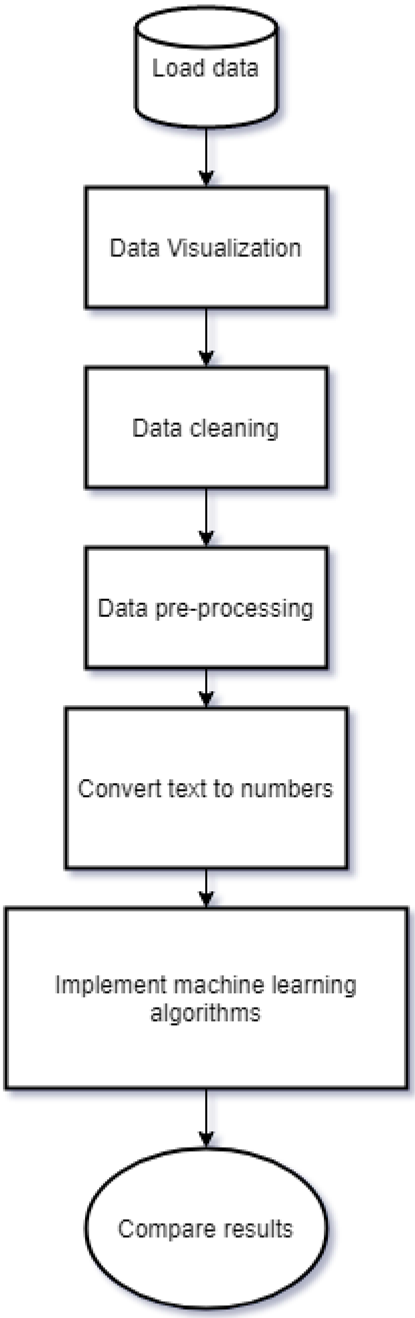
1.8 Methodology of the study

Machine learning is focused on the application that learns from experience and improves their decision-making power or predicts accurately over time. There are 3 types of Machine learning algorithms, first supervised machine learning algorithm, second unsupervised machine learning, and third reinforcement machine learning. The problem which is discussed in this study falls into supervised machine learning because it has ladled data. In the data with labels 0 and 1. 0 indicates stock price decrease and 1 indicates stock price increase. To solve the supervised machine learning problem supervised machine learning algorithms were used. The research is focused on sentiment analysis of news data. The news data have 2 classes positive and negative. This is a classification problem, so classification machine learning algorithms were chosen to solve this problem. There are basically 7 types of classification algorithms available. Data.csv file was given as an input to machine learning models, based on that data model were get trained. 25 headlines of a particular day were combined and then 2 natural language processing techniques were applied to them. Data pre-processing has been done and convert this data into a vector. After converting into the numeric format, data has been given to machine learning algorithms. Based on the algorithms it gives the prediction. At the end of the study, all of the algorithms' results were compared.

1.8.1 Structure of the study

The flow chart of the whole study is given as bellow

Figure 1 Process flowchart of the research



Source: - own preparation

1.8.2 Variables used in study

In the data.csv file, there were 2 types of variables independent and dependent. News headlines were independent variables and labels were dependent variables. There are 25 news headlines for a particular day and for each day there is one label. The label is either 0 or 1. 0 indicates negative sentiment and 1 indicated positive sentiment.

1.8.3 Text to vector techniques in machine learning

Two techniques are used in this study to convert text data into the numeric format.

- Bag of Words

  Bag of Words is a natural language processing technique. It is a natural language processing vectoring technique for creating text models. Text data cannot be fed to the machine learning model because machines do not recognize text data it only understands numeric data. Bag of Words can be used to converting text data into numeric vectors (Tavva, 2021)**.**

- TF-IDF

  The abbreviation stands for term frequency-inverse document frequency. This is also a technique to convert text data into the numeric format. This technique is used to quantify a word in the document**.** TF-IDF is a way to judge the topic of a document or article by the words it contains (Nicholson, 2020).

1.8.4 Classification algorithms in machine learning

Classification is a technique to categorize the data into class or category (Waseem, 2020). The main goal of the classification problem is to identify the class or category for a given data (Garg, 2018). Classification algorithms are the algorithms that map the input data into a specific category. The classification model is a model which tries to draw some conclusion from input values given as training data and predict class or category for the new data (Upasana, 2020). Binary classification is a classification task with 2 possible outcomes. Example :(true, false), (fake, real), (positive, negative).

To solve the classification problem we have 7 classification algorithms in machine learning.

1. Random forest
2. Naïve Bayes
3. Decision Tree
4. Gradient Boosting
5. K nearest neighbor
6. Logistic Regression
7. SVM

1.8.5 Tools and Techniques used in study

Table 1 Tools and Technique used during research

Pandas	Is a Python library for data manipulation and analysis
Sklearn	A Python library which features classification, regression, clustering algorithms and many more
encoding = "ISO-8859-1"	Encoding technique for read data
Info() function	Is used to print a summary of a Data.
Iloc operation	Is used for select integer location by location or position
Lower() function	This method returns the lowercased string from the given string
Append() function	Add a single item to existing list (Ramos, 2021)
CountVectorizer	It provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words
Type() function	It returns class type of the argument () passed as parameter.
confusion matrix	is a tabular summary of the number of correct and incorrect predictions (Brownlee, 2020)
predict()function	It enables us to predict the labels of the data values on the basis of the trained model.

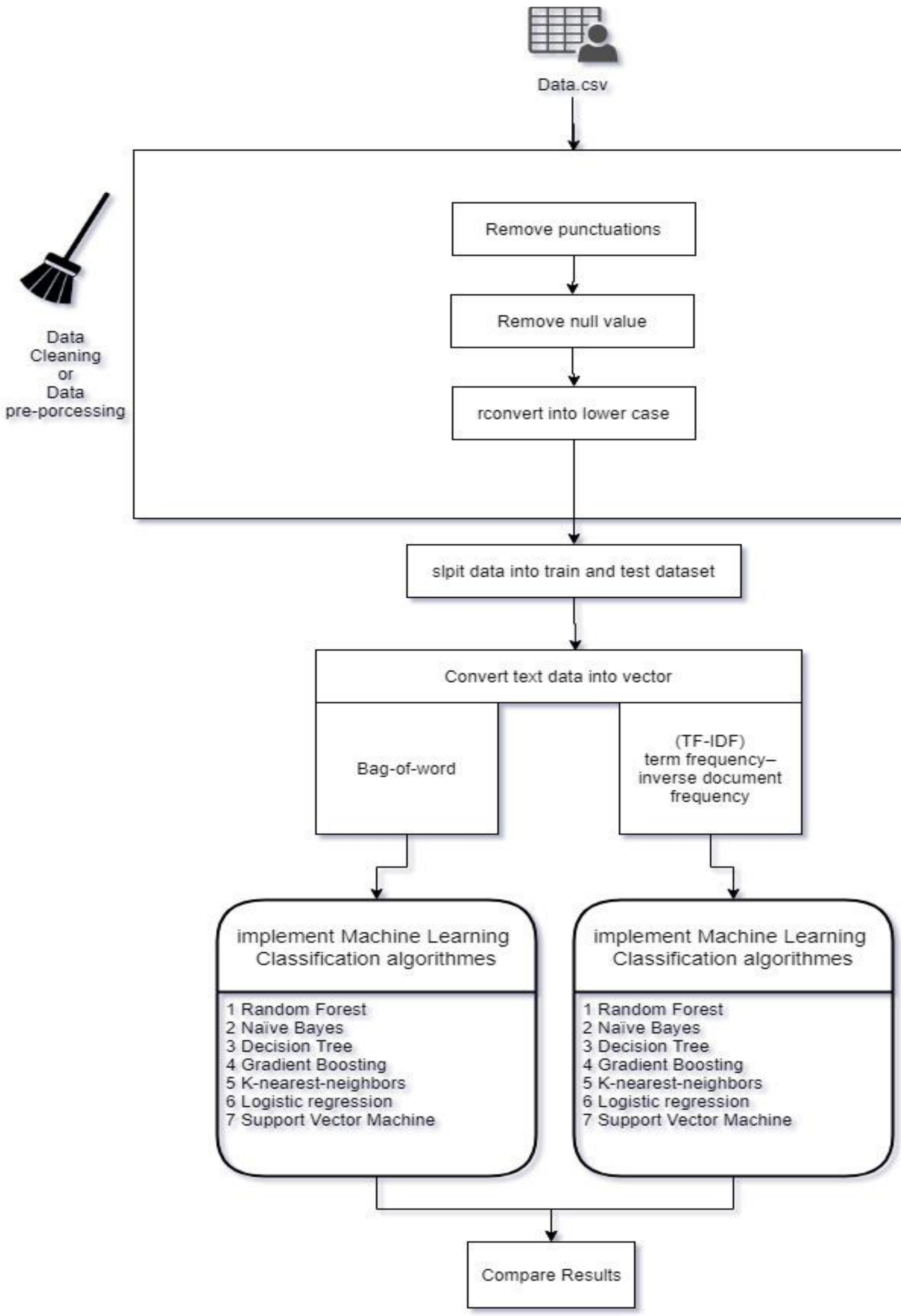
MultinomialNB	is a machine learning algorithm for classification
GradientBoostingClassifier	is a machine learning algorithm for classification
LogisticRegression	is a machine learning algorithm for classification
TfidfVectorizer	is a method which gives us a numerical weightage of words which reflects how important the particular word in document
Seaborn	Seaborn is a Python data visualization library. It provides a high-level interface for drawing attractive and informative statistical graphics (Waskom, 2020).
Read_csv() function	Pandas read_csv () function use to imports a CSV file to Data Frame format.
Heatmap	A heat map is a two-dimensional graphical representation of data where the individual values that are contained in a matrix are represented as colours (wikipedia, 2021)
Dropna() function	used to remove missing values
Replace() function	Returns a copy of the string where the old substring is replaced with the new substring.
Join() function	Returns a string in which the elements of sequence have been joined by str separator (tutorialspoint, 2021).
Length() function	To get the length of the given string, array, list, tuple, dictionary, etc (guru, 2021).
fit_transform() function	Is used on the training data so that we can scale the training data and also learn the scaling parameters of that data.
classification report	Visualizer displays the precision, recall, F1, and support scores for the model (Kohli, 2019).
accuracy score	is the fraction of predictions our model got right or percentage of correct predictions
RandomForestClassifier	is a machine learning algorithm for classification

DecisionTreeClassifier	is a machine learning algorithm for classification
KNeighborsClassifier	is a machine learning algorithm for classification
SVC	is a machine learning algorithm for classification

Source: - own preparation

1.9 Structural diagram of program

Figure 2 Flowchart of the program



Source: - own preparation

Chapter 2 Literature and review

2.1 Introduction

In this chapter includes a critical recaps of already have been researched to similar topic. Relevant literature which is similar to this research is listed in this section. Approach to structure literature review is chronological.

2.2 Review of related literature

An alternative text representation to TF-IDF and Bag-of-Words (Weinberger, 2013) in this research two techniques was used to transform text to number. Bag of Words and TF-IDF techniques were used in this research. They found that Bag of Words was simple but intuitive technique. In some simple use causes bag of word was performing very well, while in complex dataset it was not performing well. IF-IDF was performing well in complex dataset problems whereas in simple data set use case it was performing less accurate then Bag of Words. This two data pre-processing technique were popular since long time so this is chosen for this research. Bag of Word was an older but an effective technique where as TF-IDF was new and very effective in mostly modern use cases.

A study of News Analytics and Sentiment Analysis to Predict Stock Price Trends (Spandan Ghose Chowdhury, 2014) research by Spandan Ghose Chowdhury, Soham Routh , Satyajit Chakrabarti in 2014 talks about business news relation with stock price. In research they made predictive models to predict sentiment around stock price. They filtered real-time news headlines and press release data from large set of business news data and based on that he analysed the sentiment of particular company's stock price. They predict the sentiment of business news particular company. For make efficient prediction he plotted 15 company's sentiment over a period of 4 weeks. There was 67% co-relation between stock price trend and positive sentiment curve which was showing existence of semi strong hypothesis.

According to this study of Ananthi Sheshasaayee (2017) machine learning is a study which learn from data. It learns from train data and predict for the test data. Train data set is data where completely predicted data is available and in test data the output is going to be predicted.

This article used three machine learning algorithms. Naive Bay, Random Forest, and Support Vector machine. Machine get train with 500 records. Accuracy and classification errors from three algorithms are compared. Results shows that the support vector machine is 97.40% accurate which is better than the other two algorithms. Only three algorithms were used in this study, but 7 algorithms were compared in my study. According to this study, SVM is the best among two other algorithms.

An up-to-date comparison of state-of-the-art classification algorithms (Chongsheng Zhanga, 2017) in this study they compare the accuracy of 11 classification algorithms. Classification algorithms usually focus on common classifiers and their variations. It did not include many algorithms which are used in recent years. In addition, important properties such as class and feature count was not considered. In this article, they conduct a comparative study of the long established and recently proposed classifiers of 71 datasets. Stochastic Gradient Trees are compatible or higher than the performance of Support Vector Machines (SVM) and Random Forests (RF) and are the fastest algorithm in terms of predicted efficiency.

2.3 Literature used in the study

2.3.1 Text to vector techniques

- Bag of Words

In this section Use of the Beg of Words technique demonstrated with help of an example. It is used only for text data. To solving the sentimental analysis problem, in that case, can't just give text data to a machine learning model it needs to convert into the numerical format and it called vectors. Here taken an example and explained how beg of words works for converting these 3 sentences to numeric format.

Sentence 1:- He is a good boy

Sentence 2:- She is a good girl

Sentence 3:-Boy and girl are good

In-text pre-processing firstly sentences had been converted into the lower case because small and capital letters donated as a different word. In the next step, the stop word was removed. After that operation sentence becomes as follows.

Sentence 1:- good boy

Sentence 2:- good girl

Sentence 3:-boy girl good

Step-1

Created a histogram based on sentences. Histograms basically say the frequency of words in all of the sentences. This was shorted in descending order

Table 2 Histogram table based on frequency

Words	Frequency
good	3
boy	2
girl	2

Source: - own preparation

Step-2

here performed an actual bag of words to convert this histogram table into vectors.

So here good, boy and girl renamed, like an F1, F2, and F3. So for applying the machine learning model, this will be independent features F1, F2and F3 and output will be a dependent feature. So this how we can convert sentences into vectors.

For creating beg of the word very powerful library sklearn was used.

Step-1 and 2 can be implemented using the count vectorizer which is present in sklearn.

Create histogram, short histogram, creating matrix all these step we can do with countvectorizer function.

- TF-IDF

TF-IDF abbreviation stands for term frequency and inverse document frequency (Ganesh, 2021). This is also a technique for convert text into a numeric format (Ganesh, 2021). This is advanced technology compared to bad of words .in this technique every single word have their semantic meaning which is represented in decimal number. In Bag of Words, it just converts text into 0 and 1. In TF-IDF every word converts into some decimal number, so every word gets its own meaning full importance. Here in this section, TF-IDF terminology is explained with help of the same example which was used in Bag of Words.

Sentence 1:- good boy

Sentence 2:- good girl

Sentence 3:-Boy girl good

Step-1

Create a histogram based on these sentences. Histograms basically say the frequency of words in all of the sentences. This should be short in descending order

Table 3 Term frequency table generated based on frequency of words

Words	Frequency

good	3
boy	2
Girl	2

Source: - own preparation

Step2 Count Term frequency

Equation 1term frequency equation

$$\boldsymbol{TF = \frac{No\ of\ repetation\ of\ words\ in\ sentence}{No\ of\ words\ in\ sentence}}\ [1]$$

Source: - https://towardsdatascience.com/tf-term-frequency-idf-inverse-document-frequency-from-scratch-in-Python-6c2b61b78558

Based on this TF formula this table formed

Table 4 Term frequency table based on TF formula

	good	boy	Girl
Sentence1	1/2	1/2	0
Sentence2	1/2	0	1/2
Sentence3	1/3	1/3	1/3

Source: - own preparation

Step3

Equation 2 inverse document frequency

$$\boldsymbol{IDF = log\left(\frac{No.of\ Sentence}{No.of\ sentence\ containing\ words}\right)}\ [2]$$

Source: - https://towardsdatascience.com/tf-term-frequency-idf-inverse-document-frequency-from-scratch-in-Python-6c2b61b78558

Based on this IDF formula table no.5 was prepared

Table 5 inverse document frequency table based in IDF formula

good	boy	Girl
Log(3/3)=0	Log(3/2)	Log(3/2)

Source: - own preparation

Step4

TF-IDF=TF*IDF Based on this formula table no 6 was prepared

Table 6 Term frequency - inverse document frequency table based on TF-IDF formula

	F1	F2	F3
	good	boy	Girl
Sentence1	0	1/2*log(3/2)	0
Sentence2	0	0	1/2*log(3/2)
Sentence3	0	1/3*log(3/2)	1/2*log(3/2)

Source: - own preparation

So here it can see that every word converted into a number and got some decimal value based on their importance in the document. This numeric table can be used as an input to the machine learning model.

2.3.2 Classification algorithms

- Logistic regression

Logistic regression algorithm used to solve classification problem (Gupta, 2017). Both binary and multi-classification problems can be solved by using logistic regression. Logistic regression is an advanced version of linear regression. It can give a better result than linear regression. In linear regression line is fitted in data using linear line but in logistic regression s shape line is fitted to data. This s shape line can perfectly handle outlier. Logistic regression uses sigmoid and logit functions for making s shape line. It uses the concept of maximum likelihood for fitting lines into data. Logistic regression also can be used to identify what variables or parameters are useful for classifying samples. Logistic regression machine learning algorithm uses to predict a binary outcome For example yes or no, true or false. This algorithm analyzes independent variables to classify the result. It can be modeled using a logistic function. The advantages of Logistic regression are it was developed for this purpose of classification. It is particularly useful for understanding the influence of several independent variables on the outcome variable. The disadvantage of this algorithm is it only works if the predictor is binary. It assumes all predictors are independent of each other. It presumes that the data is free of missing values (Garg, 2018).

- Naive Bayes

Naïve Bayes is a collection of classification algorithm based on naïve theorem (Sambhav_Khurana, 2019). Naïve Bayes assumes that all features are independent of each other and importance of all features is equal. Naïve Bayes works on Bayes theorem. Bayes theorem finds the probability of events accruing based on past other events that occurred in past. For example, if event A occurs then event B will also occur. Generally for solving natural language processing problems multinomial Naïve Bayes is used. In this research also multinomial naïve Bayes is used. Naïve Bayes algorithm works well in some real-world scenarios such as document classification and spam filtering it works well. Naïve based algorithm can be trained base on small train data set also that is an advantage of the algorithm. It is quite fast as compared to other machine learning algorithms. Naive Bayes is a poor estimator that is a disadvantage of this algorithm. (Garg, 2018)

- Gradient boosting

Gradient boosting can be used to solve a classification problem. For creating predictive models it is one of the most powerful techniques. This algorithm can solve regression and classification problems. In the form of a set of weak prediction models, it makes a prediction model (Sameeruddin, 2020) is usually called a decision tree. Gradient boosting is a machine learning technique (mike, 2020). The advantage of this algorithm is it can benefit from regularization methods that penalize different parts of the algorithm and generally improve the performance of the algorithm by reducing overfitting. Gradient boosting is it is a greedy algorithm. This algorithm can quickly overfit a training dataset (Garg, 2018)

- K nearest neighbor

K nearest neighbor is a simple way to classify data. It uses to classify the data. This algorithm can classify data falls in which category. This algorithm cluster data and identify new data based on the distance between new data and cluster. If 2 cluster is at the same distance then it considers that cluster which has more population. Neighborhood-based classification is a type of lazy learning. From a simple majority of the k-nearest Neighbours of each point the classification it identifies the classification (Garg, 2018). This algorithm is easy to implement, robust against noisy training data, and it is effective when the training data is large. This is the advantage of this algorithm. You have to determine the value of K and the consumption cost is higher because you have to calculate the distance from each instance to all training patterns is a disadvantage of this algorithm. (Garg, 2018)

- Decision tree

Definition: Given attribute data and its classes, a decision tree generate a sequence of rules with which the data can be classified (Gupta, 2017). Advantages of The decision tree are easy to understand and visualize, It requires little data preparation and it can process both numerical and categorical data (Sosnovshchenko, 2020). Disadvantages of The decision tree can create complex trees that are not easy to generalize, and decision trees can be unstable because small variations in the data can create a completely different tree (Garg, 2018).

- Random forest

Definition: The Random Forest Classifier is a meta-estimator that can fits a series of decision trees across different subsamples of data sets and uses the mean to improve the predictive accuracy of the model and controls for overfitting (stackoverflow, 2016). The size of the sub-sample always corresponds to the original sample size of the input, but the samples are taken with replacement. The advantage of this algorithm is in most cases, the reduction in overfitting, and the random forest classifier is more accurate than with decision trees. Disadvantages of this algorithm slow forecast in real-time, difficulty to implement, and complex algorithm (Garg, 2018).

- Support Vector Machine

Definition: The Support Vector Machine is a representation of training data as points in space, which are divided into categories by as large a free space as possible. When the New data comes it mapped to the same space. Then after it is assumed to fall into a category based on which side of the gap they are on. The advantages of SVM is very efficient in large-dimensional spaces and it uses a subset of learning points in the decision-making function (Garg, 2018). Which also makes memory efficient. The disadvantage is, the algorithm does not provide direct probability estimates, these are calculated using an expensive five-pass cross-validation (Garg, 2018).

Chapter 3 Data visualization

3.1 Introduction

In this chapter, data visualization steps were included. Data visualization is a very important step in a data science project.it is the first step in this cycle. Before doing any procedure on data it is important to learn about data that's why we do data visualization first. Data visualization techniques are used based on the type and the condition of the data. This technique varies according to need.

3.2 load data

Jupyter notebook was used as a development tool. In this program data can upload to the folder where the source file is located, if it is not in the same folder or directory then it will not read data, so it is very important to keep the data file in the same folder. After that, we can give the command to read the file. In this case, it was a CSV data file. In python, it has a read function to read CSV file. With the help of the read function data is loaded in memory. So at this stage, the data is stored in data frame name df.

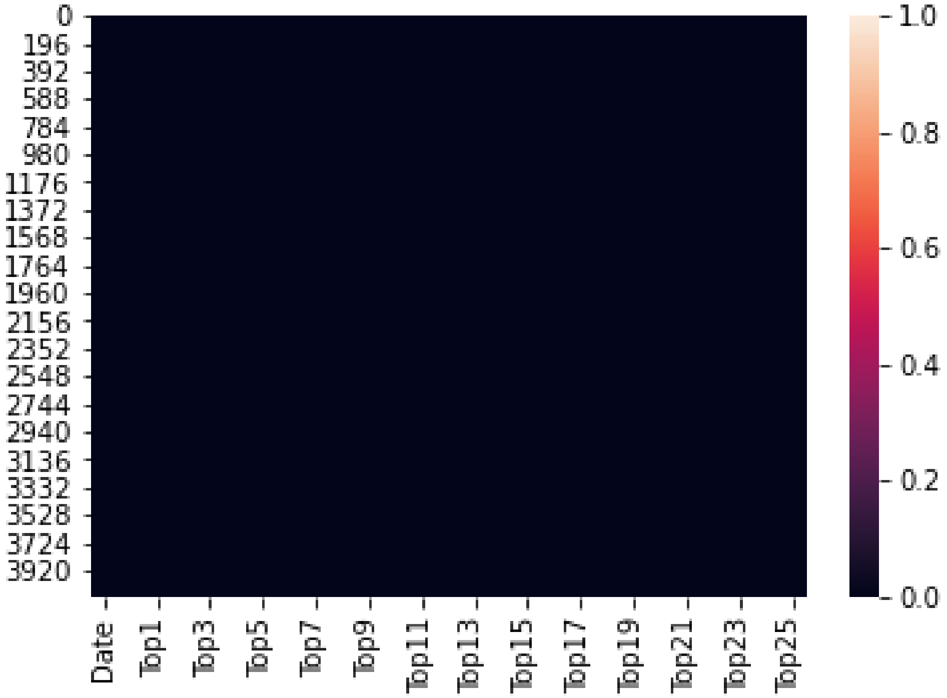
3.3 Identify and remove the null value

In data, it is possible that some value is missing or not available. The reason for that could be anything like corrupted data, failure to load data, or incomplete extraction of data from the source. Handle a missing value is a big task. Making the decision of how to handle missing values can effect on machine learning model. In this research case, it needs to check first if it has any missing value in data or not. For that heat, the map function is used from the seaborn library and df.info function to see the information of the data.

3.3.1 Heat map

A heat map is a technique to present data into a two-dimensional color graph.it is basically used to visualize a summary of the data. Heat map function used to find the major null value in data. The isnull function gives the null value present in data. The isnull function is given as an input to the heatmap function. So this will display a graph of a null value.

Figure 3 Heatmap of whole data



Source: - own preparation

This graph shows that there is no major null value available in data, if major null value was present then it will reflect as a white line or dot in the graph.

3.3.2 Df.info () function

Based on the heatmap it is confirmed that there is no major missing value but it doesn't mean there is not null value in data, so to find where and how much null value is available in data DF.info function was used. Which gives not a null count of columns.

Figure 4 Information report of the whole data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4101 entries, 0 to 4100
Data columns (total 27 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Date    4101 non-null   object
 1   Label   4101 non-null   int64 
 2   Top1    4101 non-null   object
 3   Top2    4101 non-null   object
 4   Top3    4101 non-null   object
 5   Top4    4101 non-null   object
 6   Top5    4101 non-null   object
 7   Top6    4101 non-null   object
 8   Top7    4101 non-null   object
 9   Top8    4101 non-null   object
 10  Top9    4101 non-null   object
 11  Top10   4101 non-null   object
 12  Top11   4101 non-null   object
 13  Top12   4101 non-null   object
 14  Top13   4101 non-null   object
 15  Top14   4101 non-null   object
 16  Top15   4101 non-null   object
 17  Top16   4101 non-null   object
 18  Top17   4101 non-null   object
 19  Top18   4101 non-null   object
 20  Top19   4101 non-null   object
 21  Top20   4101 non-null   object
 22  Top21   4101 non-null   object
 23  Top22   4101 non-null   object
 24  Top23   4100 non-null   object
 25  Top24   4098 non-null   object
 26  Top25   4098 non-null   object
dtypes: int64(1), object(26)
memory usage: 865.2+ KB
```

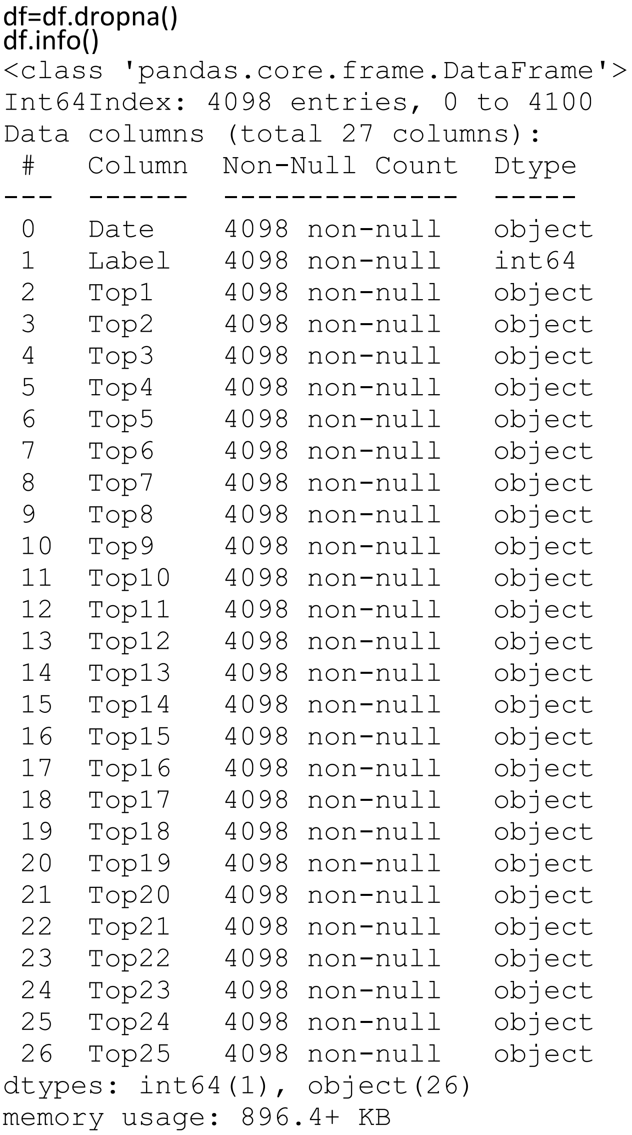
Source: - own preparation

Based on the information it is clear that data have 26 columns in the dataset. Column number 2 to 26 contains headline data named Top1 to Top25. All of the columns have 4101 numbers of value but only the top 24 and top 25 have 4098 numbers of value. It means in whole data have just 4 missing value. Data have only just 4 missing records it was convenient to delete those records because that is a very small amount of data and it would not make much effect on the machine learning model.

3.3.3 Remove null value

Python has a dropna() function to remove missing values. It removed all missing values so the total will be 4098 records for all 25 columns.

Figure 5 Not null report of the data



```
df=df.dropna()
df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4098 entries, 0 to 4100
Data columns (total 27 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Date    4098 non-null   object
 1   Label   4098 non-null   int64 
 2   Top1    4098 non-null   object
 3   Top2    4098 non-null   object
 4   Top3    4098 non-null   object
 5   Top4    4098 non-null   object
 6   Top5    4098 non-null   object
 7   Top6    4098 non-null   object
 8   Top7    4098 non-null   object
 9   Top8    4098 non-null   object
 10  Top9    4098 non-null   object
 11  Top10   4098 non-null   object
 12  Top11   4098 non-null   object
 13  Top12   4098 non-null   object
 14  Top13   4098 non-null   object
 15  Top14   4098 non-null   object
 16  Top15   4098 non-null   object
 17  Top16   4098 non-null   object
 18  Top17   4098 non-null   object
 19  Top18   4098 non-null   object
 20  Top19   4098 non-null   object
 21  Top20   4098 non-null   object
 22  Top21   4098 non-null   object
 23  Top22   4098 non-null   object
 24  Top23   4098 non-null   object
 25  Top24   4098 non-null   object
 26  Top25   4098 non-null   object
dtypes: int64(1), object(26)
memory usage: 896.4+ KB
```

Source: - own preparation

Chapter 4 Data Pre-Processing

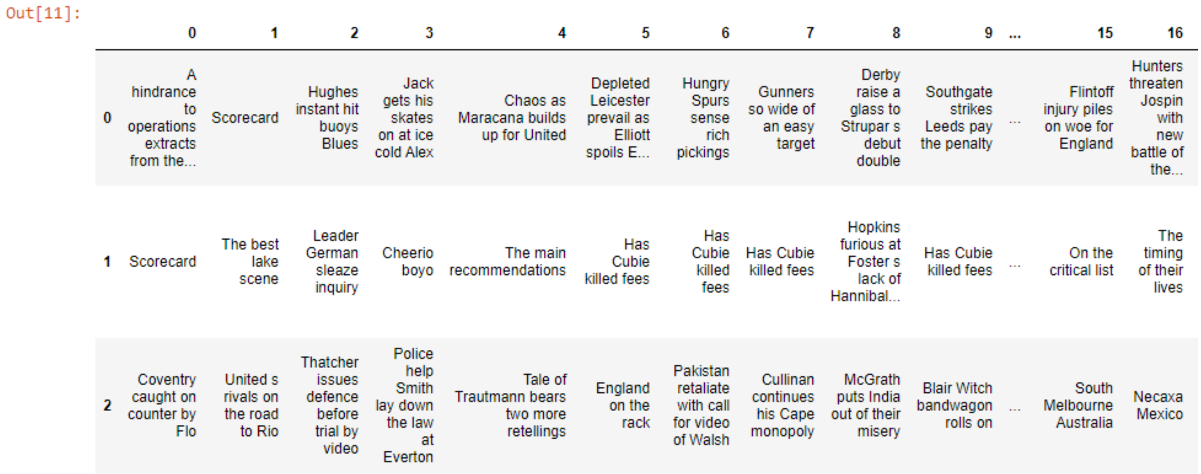
4.1 Introduction

In this chapter data pre-processing steps are included. Data pre-processing steps are very important in the machine learning life cycle because before giving data to a machine learning model if it is in the simpler form then the machine learning model can easily process that data. In this part simplification of data, splitting data into 2 data set, and transform data into numeric format is mentioned.

4.2 Rename column

For the ease of understanding and for the ease of access column name converted into a simpler form. In this data, it has column names like top1 to top25. For ease of access, it converted into 1 to 25.

Figure 6 column rename for ease of access



Out[11]:

	0	1	2	3	4	5	6	7	8	9	...	15	16
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...
1	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...	Has Cubie killed fees	...	On the critical list	The timing of their lives
2	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	McGrath puts India out of their misery	Blair Witch bandwagon rolls on	...	South Melbourne Australia	Necaxa Mexico

Source: - own preparation

4.3 Removing punctuations

This data set has a lot of punctuation marks and special characters, which is not useful so it is better to remove them before giving data to the model so that it will predict better results.

With help of Python code, apart from small a to z and capital A to Z rest of the other values replaced with blank space. With help of this Python code "data=train.iloc[:,2:27]" selected all features of the training dataset and put them into the list. A regular expression specifies a set of strings that matches it. The purpose of this let it check if a particular string is the same as a given regular expression or not. inplace=True means it will replace on its original place.

4.4 Converting headlines to lower case

Python is a case-sensitive language. Machine learning models also treat capital and small letters as different. So it is important that all test data which was given to the machine learning model should be in one format. In this step, headlines were got converted into lower case.

4.5 Join Headlines

Here for loop is iterating through each and every sentence and join them into one sentence. Here separate every sentence with blank space. Join operation was performed for the whole train dataset. Combined all sentences of all records. Using for loop it had iterate through all records and then created a list named headlines where it appends all the sentences. The headline was a list, which contains all of the headlines of the training dataset.

4.6 Split data

In machine learning data splits into train and test datasets because the training dataset will be given to the machine learning model and the machine learning model will make predictions. After that predictions can be compared with the actual result and could get the accuracy of the machine learning model. So basically to measure the performance of the machine learning model generally split the data into train and test dataset. Generally, the train and test splitting ratio is 70:30 or 80:20 (showmethebell, 2017). However, it is always better to have more data in the training dataset because if the Machine learning model gets more data them it predicts a better result. In this research case data spat into approx. 80:20 ratio. So approximately 80 percent of the data is in the test dataset which will be going to train in the machine learning model. The dataset, it has data from the year 2000 to 2016. From the year 2000 to 2014 news headlines are in the training dataset. 2015 to 2016 news headlines are in the test dataset.

4.7 Transformation of text data into a numerical vector

To transform the text data into the numeric format, 2 techniques were used in this research (Kalsi, 2018). Bag of Words and TF-IDF are pretty powerful techniques using now a day in the data science field. However, TF-IDF is a more advanced technique than Bag of Words but in some circumstances Bag of Words also performs well.

- Bag of Words

Countvectorizer provides a simple way to tokenize the text and convert them into a vector. Countvectorizer does all necessary steps for Bag of Words. It creates a histogram table, features a table, and creates vectors from them. All of the steps which are necessary for Beg of words were performed by countvectorizer. Countvectorizer is present in the sklearn library of python. This function and library are so powerful that with the help of just one or 2 lines of python code whole data can be converted into numeric form.

Here to import the CountVectorizer from python's sklearn library Feture_extraction.text module of sklearn library was used. Countvectorizer was performed on the training dataset. CountVectorizer has one feature called Ingram. Ngram basically justifies how many words should be treated together in one sentence.

Here an example is taken to explain it in a simpler way. For example "I am Bhargav" is the sentence. This sentence has 3-grams. If n-gram is (2, 2) then it will treat all 2 combinations of words "I am" and "am bhargav" will be treated as a combination. The parameter also can be set for simplification n-gram (a, b) where "a" is the minimum size of n-gram and "b" is the maximum size of n-gram. By default, it is (1, 1) if the parameter is not set. This research (1, 1) was used because it was giving more accuracy compared to other combinations.

- TF-IDF

TF-IDF is the second technique that was used in this research to convert text into a numeric format. However, this is an advanced technique than Bag of Words. Bag of Words is just a simple technique to convert text into vector but TF-IDF as its name suggests term frequency and inverse document frequency, It can give more meaning to a particular word which is beneficial for sentimental analysis. TfidfVectorizer is used to perform IF-IDF and it is present in the feture_extraction.text module of the sklearn library. TF-IDF can be performed with TfidfVectorizer. It basically helps to tokenize the document, count document frequency, and inverse document frequency, and build vocabulary (Cakır, 2020). Here in this study default parameter n-gram used because it was giving good accuracy.

Chapter 5 Implementation of ML Algorithms

5.1 Introduction

This chapter includes the main key part of the research which is training the machine learning model. This part is like the engine of the car. Machine learning algorithms play the main role in the whole program. In this research 7 machine learning algorithms implemented with 2 data pre-processing technique which makes 14 combinations. This means 14 times train data had been given to different combinations. The algorithm predicted the value for the test dataset. In this chapter, This 14 combination is discussed in detail.

5.2 Implementation using Bad of word

With the help of the Bag of Words train dataset already converted into a numeric form which is stored in the train dataset variable. This training dataset given as an input to all machine learning algorithms and based on the data models was got trained. After that test dataset given to all particular models and all of the models will predict some output.

1 Bag of Words with Random forest

Random forest classifier imported which is present in sklearn library's ensemble module. The random forest can be implemented with the help of a random forest classifier function. Train data set and train label is given as an input to the model. Here join function joins all test data set headlines into a test transform list and with the help of the countvectorizer it was converted into the numeric format and that values are stored in the test dataset variable. This test dataset variable was given as an input to the random forest algorithm and it predicts the output .that output was stored in a prediction array.

2 Bag of Words with Naïve Bayes

Naïve Bayes algorithm can be implemented with the help of the multinomialNB function, which is available in the Naïve Bayes module inside the sklearn library. Here miltinomialNB function was implemented and the training dataset and train label were given as an input to the algorithm. This had built a model and learn from train dataset. The same steps for the test data are performed like join headlines of the test dataset, transform them into the numeric format and give as an input to naïve Bayes algorithm to predict. The prediction is done by naïve Bayes and was stored in a prediction array.

3 Bag of Words with Decision Tree

The decision tree algorithm can be implemented with the DicisionTreeClassifier function which is present in the tree module of the sklearn library. Import keyword was used to import that library. Train dataset and train label was given as an input to decision tree algorithm and based on train data model were got trained. All headlines of test data are combined and converted into a vector with the help of a countvectorizer and it was given as an input to the decision tree algorithm. The decision tree algorithm predicts the output based on trained data and predictions were stored in a prediction array.

4 Bag of Words with Gradient Boosting

Gradient boosting algorithm is a very powerful algorithm that is present in the ensemble module of the sklearn library. Gradient boosting classifier algorithm can be performed using the GredientBoostinClassifier function. Train data set was given as an input to the algorithm. With join function, all headlines of the test dataset are combined and converted into a vector with the help of Bag of Words and it had given as an input to gradient boosting algorithm. Based on the test data gradient boosting algorithm predict the results for the test data set and these results were stored in a prediction array.

5 Bag of Words with K nearest neighbor

K nearest neighbor is a classification algorithm that can be performed with the KNeighboursClassifier function which is present in the neighbor module of the sklearn library. Two lines of Python code train dataset and train label were given as an input to K nearest neighbor algorithm. All headlines of the test dataset were combined and converted into a vector with help of Bag of Words and given as an input to the K nearest neighbor's algorithm. Based on that model got trained.

6 Bag of Words with Logistic Regression

Logistic regression algorithm can be implemented using LogisticregRession function which is present inside liner_model module of sklearn library. Here train dataset and train label was given as an input to the logistic regression algorithm, based on the train data model got trained.

Headlines of the test data set are combined and converted into a vector with the help of Bag of Words and given as an input to the logistic regression algorithm.

7 Bag of Words with SVM

Support vector machine is a very powerful and advanced algorithm in machine learning.it can be implemented with help of the SVC function which is present in the svm module of the sklearn library.

The train dataset and train label was given as an input to the SVM algorithm. Based on train data SVM model gets trained. All headlines of test data are combined and converted into a numeric format with help of Bag of Words and it was given as an input to the SVM algorithm to predict results. The results were stored in a prediction array.

5.3 Implementation using TF-IDF

In this part, TF-IDF is used to convert the text data into a vector. So TF-IDF with the same 7 classification algorithms is implemented in this section of research. Already import all algorithms earlier from sklearn library while implementing with Bag of Words so no need to import it again. The algorithm can use this directly because it had already imported into the program. TfidfVectorizer function was used to implement TF-IDF. TfidfVectorizer is present in the feture_extraction.text module of the sklearn library. Here headlines were given as an input to TF-IDF and it will convert into vector.

1 TF-IDF with Random forest

Train data set is converted into the vector using TF-IDF. Random forest classifier is trained with help of train data set. The model got trained and then test data set was given for prediction. The prediction result was stored in a prediction array.

2 TF-IDF with Naïve Bayes

Train dataset and train label were given as an input to naïve Bayes algorithm. Based on input data model got trained. Test data set's headlines joins together and converted into the vector using TF-IDF and given as an input to naïve Bayes algorithm to predict results. Naïve Bayes algorithm makes the predictions and it was stored in a prediction array.

3 TF-IDF with Decision Tree

Firstly Decision tree classifier algorithm was got trained whit a training dataset and a training label secondly all headlines of the test dataset were combined and converted into the vector using TF-IDF in the final test dataset it was given as an input to the decision tree classifier and it predicts output which was stored in prediction array.

4 TF-IDF with Gradient Boosting

The training dataset was given as an input to the gradient boosting algorithm so the model can be trained. Test dataset’s all headlines got combined and converted into a vector and given as an input to gradient boosting algorithm and algorithms predicts the output which was stored in prediction array.

5 TF-IDF with K nearest neighbor

Firstly train dataset and train labels were given as an input to K nearest neighbor algorithm to get trained .secondly all headlines of the test dataset are combined and converted into a vector with help of TF-IDF. At the last test, the dataset is given as input and based on the train dataset value algorithm made a prediction.

6 TF-IDF with Logistic Regression

Logistic regression model gets trained with train dataset and train label. The model got trained and learn from data then new data as a form of test data given for making predictions. The model predicts output and it was stored values in the prediction array.

7 TF-IDF with SVM

Support vector machine gets trained with train dataset. Test dataset which was converted into a vector with help of TF-IDF given as input to support vector machine. Support vector machine predicts output and that prediction was stored in a prediction array.

Chapter 6 Findings, Conclusions, and Suggestions

6.1 Introduction

This chapter includes the main results and performance of all machine learning algorithms which were used in this study. It also includes a discussion based on results and future works that could be possible in this study. For measure, the accuracy of the algorithm's accuracy score, confusion matrix, and classification report was used. These all 3 measures are available in the sklearn library of Python.

6.2 Findings of the study

6.2.1 Accuracy Score

The below table indicates the accuracy score of 7 machine learning models with respect to Bag of Words and TF-IDF.

Table 7 Accuracy Score of machine learning algorithms

algorithms	Accuracy with Bag of Words	Accuracy with TF-IDF
Random Forest	85.97	81.74
Naïve Bayes	85.18	50.79
Decision Tree	83.86	83.06
Gradient Boosting	71.95	73.28
K nearest Neighbours	58.73	65.87
Logistic Regression	83.06	80.15
SVM	79.62	84.65

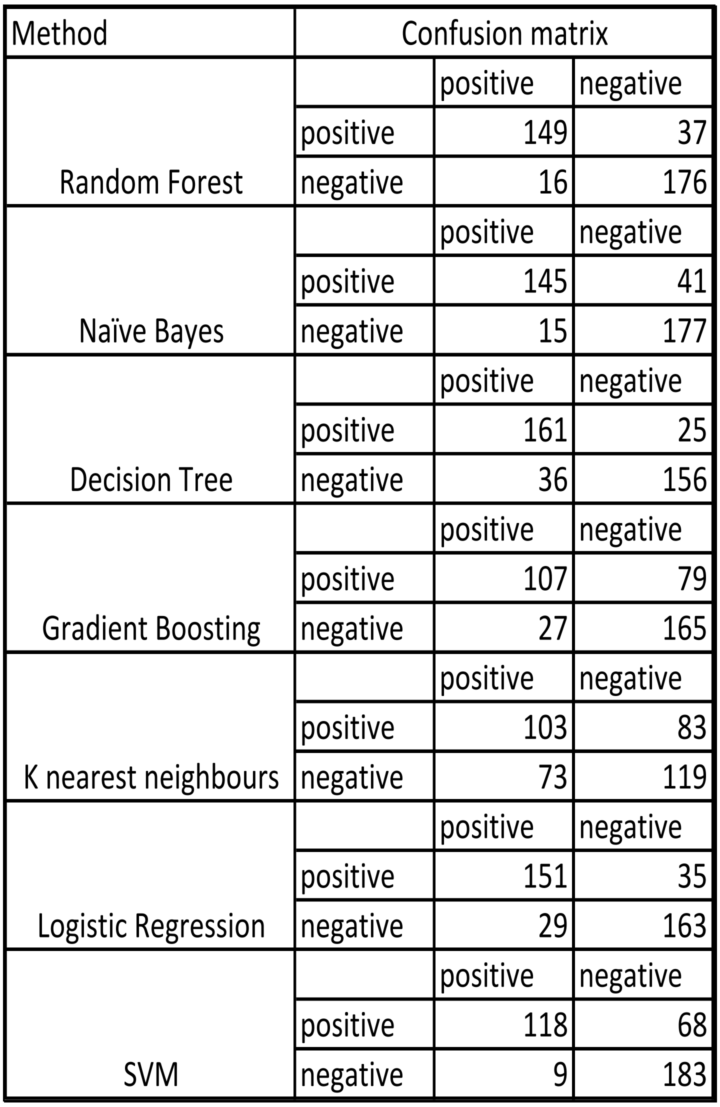
Source: - own preparation

In machines, learning accuracy is a measure for comparing different algorithms. It indicates how much correct prediction was made. Here data set was balanced so the accuracy score can be considered as a perfect measure for machine learning models. In the data set a total number of samples was around 4000 among 2000 samples was labeled 0 and another 2000 sample was labeled 1 so data was perfectly balanced. Because of balanced data set machine learning models didn't get biased to a particular label. That's why accuracy scores can be considered as the best parameter to measure the results.

6.2.2 Confusion matrix

The Bellow table shows the confusion matrix of the 7 machine learning models with respect to Bag of Words.

Table 8 Confusion matrix of machine learning algorithms with respect to Bag of Words

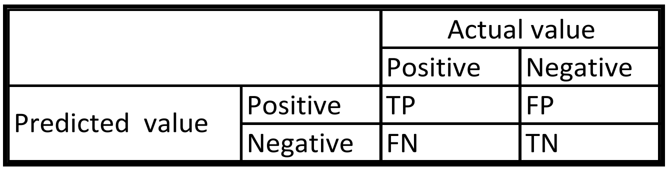


Method	Confusion matrix		
Random Forest		positive	negative
	positive	149	37
	negative	16	176
Naïve Bayes		positive	negative
	positive	145	41
	negative	15	177
Decision Tree		positive	negative
	positive	161	25
	negative	36	156
Gradient Boosting		positive	negative
	positive	107	79
	negative	27	165
K nearest neighbours		positive	negative
	positive	103	83
	negative	73	119
Logistic Regression		positive	negative
	positive	151	35
	negative	29	163
SVM		positive	negative
	positive	118	68
	negative	9	183

Source: - own preparation

Confusion matrix n*n matrix which used to measure the performance of the machine learning algorithm. Where n is a number of labels in data. This matrix is used to compare the algorithm's predicted value with the actual value.

Table 9 Confusion matrix Actual value*Predicted value



		Actual value	
		Positive	Negative
Predicted value	Positive	TP	FP
	Negative	FN	TN

Source: - https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

True Positive (TP):- Actual value was positive and the machine learning model predicted it positive. In this case, the model predicts that stock price will increase and actually indeed stock price was increased.

True negative (TN):-True Negative Actual value was negative and the machine learning model predicted it as a negative. In this case, the model predicts that stock price will decrease and actually indeed stock price decreased.

False Positive (FP) or Type 1 error: - Actual value was negative but the machine learning model predicted as a positive. . In this case, the model predicts that stock price will increase and actually indeed stock price also decreased.

False Negative (FN) or Type 2 error: - Actual value was positive but the machine learning model predicted as negative. . In this case, the model predicts that stock price will decrease and actually indeed stock price also increased.

Table nr.10 shows the results of the confusion matrix of the machine learning model which was used in the study with respect to TF-IDF.

Table 10 Confusion matrix of machine learning algorithms with respect to TF-IDF

Method	Confusion matrix		
Random Forest		positive	negative
	positive	142	44
	negative	25	167
Naïve Bayes		positive	negative
	positive	0	186
	negative	0	198
Decision Tree		positive	negative
	positive	160	26
	negative	38	154
Gradient Boosting		positive	negative
	positive	108	83
	negative	18	74
K nearest neighbours		positive	negative
	positive	106	80
	negative	49	143
Logistic Regression		positive	negative
	positive	125	61
	negative	14	178
SVM		positive	negative
	positive	131	55
	negative	3	189

Source: - own preparation

6.2.3 Classification Report

A classification report is used to measure the quality of prediction made by classification machine learning algorithms (MUTHUKRISHNAN, 2018).it indicates the value of precision, recall, and F1 score.

1 **Precision**

In general, tells how many of correctly predicted value actually turned into positive. Precision is a very good measure when a false positive is more important than a false negative, like in this research scenario. In this research, if the model predicts that stock price will increase and actually stock price decrease then it's not a big problem compared to if the model predicts stock price decreased but actually stock price increased. Precision can be calculated with the following formula

Equation 3 Precision accuracy measure

$$\boldsymbol{Precision = \frac{TP}{TP+FP}}\ [3]$$

Source: - https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

1 Recall

Recall tells how many positive cases were actually predicted positive correctly. The recall is preferred when false negative is a higher priority than false positive. For example especially in medical science cases. Recall can be calculated with help of the following formula.

Equation 4 Recall accuracy measure

$$\boldsymbol{Recall = \frac{TP}{TP+FN}}\ [4]$$

Source: - https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

2 F1 Score

In some situations, it is not decidable where precision or recall is more important in that scenario F1 score used to measure the model performance. When the precision value increases then the recall value decrease and if the recall value increases then the precision value decreases. So to measure the trend between precision and recall F1 score used. F1 score can be calculated with the following formula.

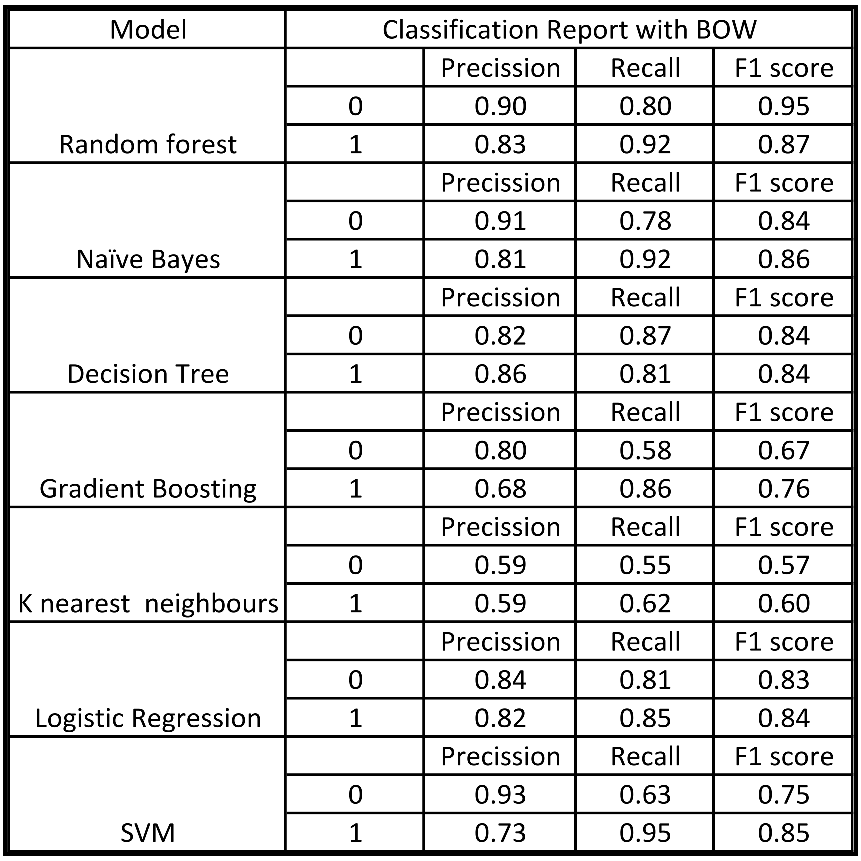
Equation 5 F1 Score accuracy measure

$$\boldsymbol{F1\ Score = \frac{2}{1/Recall+1/Precision}}\ [5]$$

Source: - https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Bellow table gives classification report of all machine learning model which perfume in this study with respect to Bag of Words. This report gives the value of precision, recall, and f1 score value for a particular modal. Based on the classification report with respect to the Bag of Words support vector machine performed the best and k-nearest neighbor performs the worst. Unexpectedly gradient boosting should perform well according to (mike, 2020) but it didn't perform well. Random forest and support vector machine was expected to perform well according to research (stackoverflow, 2016) and it performs well.

Table 11 Classification Report with Bag of Word



Model	Classification Report with BOW			
Random forest		Precission	Recall	F1 score
	0	0.90	0.80	0.95
	1	0.83	0.92	0.87
Naïve Bayes		Precission	Recall	F1 score
	0	0.91	0.78	0.84
	1	0.81	0.92	0.86
Decision Tree		Precission	Recall	F1 score
	0	0.82	0.87	0.84
	1	0.86	0.81	0.84
Gradient Boosting		Precission	Recall	F1 score
	0	0.80	0.58	0.67
	1	0.68	0.86	0.76
K nearest neighbours		Precission	Recall	F1 score
	0	0.59	0.55	0.57
	1	0.59	0.62	0.60
Logistic Regression		Precission	Recall	F1 score
	0	0.84	0.81	0.83
	1	0.82	0.85	0.84
SVM		Precission	Recall	F1 score
	0	0.93	0.63	0.75
	1	0.73	0.95	0.85

Source: - own preparation

Table no.12 gives a classification report of a particular model with respect to TF-IDF. According to this report, the support vector machine performs the best. Naive Bayes performed the worst. Support vector machine, random forest, and gradient boosting are powerful algorithm which was expected to perform well according to (Chongsheng Zhanga, 2017) and they work well. Unexpectedly decision tree performs well with TF-IDF.

Table 12 Classification Report with TF-IDF

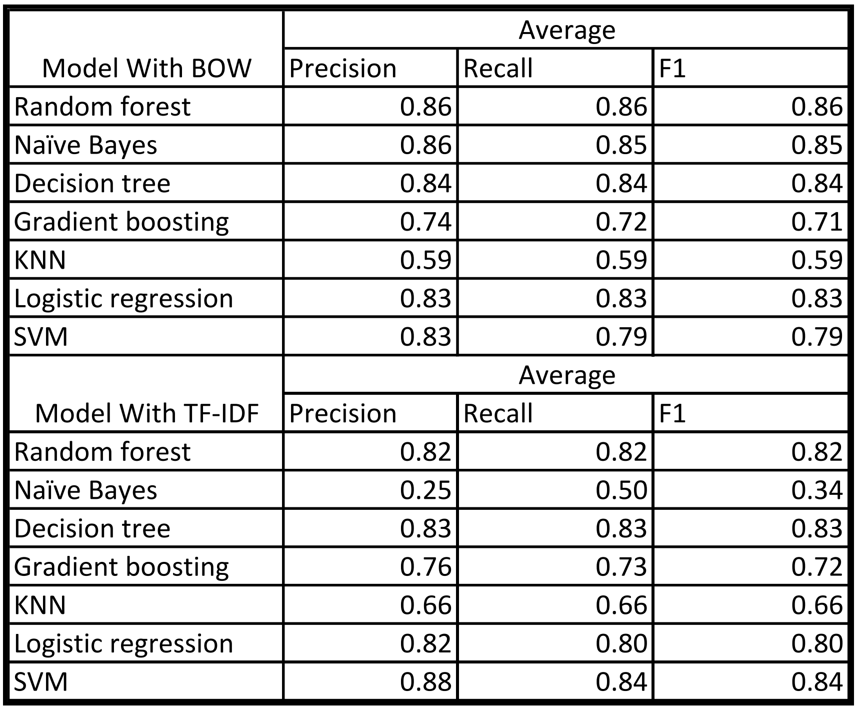
Model	Classification Report with TF-IDF			
Random forest		Precission	Recall	F1 score
	0	0.85	0.76	0.80
	1	0.79	0.87	0.83
Naïve Bayes		Precission	Recall	F1 score
	0	0.00	0.00	0.00
	1	0.51	1.00	0.67
Decision Tree		Precission	Recall	F1 score
	0	0.81	0.86	0.83
	1	0.86	0.80	0.83
Gradient Boosting		Precission	Recall	F1 score
	0	0.85	0.55	0.67
	1	0.68	0.91	0.78
K nearest neighbours		Precission	Recall	F1 score
	0	0.68	0.57	0.62
	1	0.64	0.74	0.69
Logistic Regression		Precission	Recall	F1 score
	0	0.90	0.76	0.77
	1	0.74	0.93	0.83
SVM		Precission	Recall	F1 score
	0	0.98	0.70	0.82
	1	0.77	0.98	0.87

Source: - own preparation

4 Macro average

It is the average precision, recall, and F1 score with respect to positive and negative class. Macro average is basically used to compare models. Table no.13 presents the value of the macro average of a particular model. According to the macro average random forest with Bag of Words and support vector machine performs best. K nearest neighbor with Bag of Words performs the worst. Unexpectedly naïve Bayes didn't perform well according to research (Ananthi Sheshasaayee, 2017) it was expected to perform well.

Table 13 Macro average of machine learning algorithms



Model With BOW	Average		
	Precision	Recall	F1
Random forest	0.86	0.86	0.86
Naïve Bayes	0.86	0.85	0.85
Decision tree	0.84	0.84	0.84
Gradient boosting	0.74	0.72	0.71
KNN	0.59	0.59	0.59
Logistic regression	0.83	0.83	0.83
SVM	0.83	0.79	0.79
Model With TF-IDF	Average		
	Precision	Recall	F1
Random forest	0.82	0.82	0.82
Naïve Bayes	0.25	0.50	0.34
Decision tree	0.83	0.83	0.83
Gradient boosting	0.76	0.73	0.72
KNN	0.66	0.66	0.66
Logistic regression	0.82	0.80	0.80
SVM	0.88	0.84	0.84

Source: - own preparation

6.3 Conclusion of the study

This research can lead to few conclusions, which are as follows:

1. Accuracy score is the best measure to measure the accuracy of machine learning model,
2. Precision is the second-best priority as per this research's type,
3. F1 score and at least recall are important measures for this research.

Based on accuracy measure top 3 models were chosen:

1 Random forest with Bag of Words

2 Naïve Bayes with Bag of Words

3 SVM with TF-IDF

Based on precision top 3 models were chosen:

1 SVM with TF-IDF

2 Random forest with Bag of Words

3 Naïve Bayes with Bag of Words

Based on the F1 score vale top 3 models

1 Random forest with Bag of Words

2 Naïve Bayes with Bag of Words

3 SVM with TF-IDF

Based on Recall vale top 3 models

1 Random forest with Bag of Words

2 Naïve Bayes with Bag of Words

3 SVM with TF-IDF

Based on the results of this research it can be concluded that Random forest with Bag of Words was the best performing model, the second-best model used SVM with TF-IDF, and the third was Naïve Bayes with Bag of Words. So in this study where 7 machine learning algorithms with their default parameter with 2 texts to vector technique, in total from 14 combinations of models, Random forest with Bag of Words found the best combination for solving this research problem.

6.4 Limitations and further suggestion of the study

In machine learning, there are many more algorithms which can be used in classification problem. In this thesis, 7 of them were used because of their proficiency and their popularity. Word2vec which is a very famous library nowadays in data science, research on this is continuing by Google to improve its performance of it.Word2vec also can be used to convert text data into a numeric vector were in this study Bag of Words and TF-IDF were used. In this research, all algorithm parameters were not set manually, but it was the default setting but to increase the accuracy of algorithms this parameter can be adjusted. For adjusting different parameters first, different parameters should experiment with a respective model which takes a lot of time. To make this work ease the sklearn library has functionality like random search CV and gridsearchcv. This functionality can be used to perform hyper parameter tuning, which can increase the model's accuracy.

Bibliography

Advani, V., 2020. 34 Open-Source Python Libraries You Should Know About. [Online] Available:https://www.mygreatlearning.com/blog/open-source-python-libraries[Accessed 05 11 2020].

Ananthi Sheshasaayee, G. T., 2017. Comparison of Classification Algorithms in Text Mining. SEMANTIC SCHOLAR, 10(Classification Algorithms), p. 10.

Anita Kumari Singh, M. S., 2019. Vectorization of Text Documents for Identifying. semanticscholar, 6(Vectorization), p. 6.

Brownlee, J., 2020. 4 Types of Classification Tasks in Machine Learning. [Online] Available:https://machinelearningmastery.com/types-of-classification-in-machine-learning.[Accessed 15 01 2021].

Brownlee, J., 2020. What is a Confusion Matrix in Machine Learning. [Online] Available:https://machinelearningmastery.com/confusion-matrix-machine-learning/ [Accessed 09 12 2020].

Cakır,A.,2020.nlp-classification-recommendation. [Online] Available:https://towardsdatascience.com/nlp-classification-recommendation-project-cae5623ccaae[Accessed 09 01 2021].

Chongsheng Zhanga, C. L. X. Z. G. A., 2017. An up-to-date comparison of state-of-the-art classification algorithms. elsevier, 23(classification algorithms), p. 23.

Cowley, E., 2019. What is a machine learning model?. [Online] Available:https://docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-.[Accessed 06 11 2020].

Ganesh,M.,2021.what-is-tf-idf-and-how-does-it-work. [Online] Available:https://blog.expertrec.com/what-is-tf-idf-and-how-does-it-work/

[Accessed 06 01 2021].

Garg,R.,2018.7-Types-of-Classification-Algorithms. [Online] Available:https://analyticsindiamag.com/7-types-classification-algorithms/ [Accessed 25 08 2020].

Gubarev, E. Y. G. a. V. V., 2013. Analytical Review of Data Visualization Methods in Application to Big Data. Journal of Electrical and Computer Engineering, 30(Data Visualization Methods), p. 20.

Gupta, A., 2017. 30 Questions to test your understanding of Logistic Regression. [Online] Available at: https://www.analyticsvidhya.com/blog/2017/08/skilltest-logistic-regression/ [Accessed 06 11 2020].

Gupta,P.,2017.decision-trees-in-machine-learning. [Online] Available:https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 [Accessed 08 01 2021].

guru,2021.Python-string-length-len()method [Online] Available:https://www.guru99.com/python-string-length-len.html
[Accessed 23 02 2021].

Heller, M., 2019. Machine learning algorithms explained. INfoWorld, 10(How machine learning works), p. 10.

jupyter,2020.jupyter.org. [Online] Available:https://jupyter.org/ [Accessed 01 12 2020].

Kalsi,S.,2018.Converting-Text-To-Numeric-Vector. [Online] Available at: https://sachinkalsi.github.io/blog/category/ml/2018/05/26/converting-text-to-numeric-vector.html
[Accessed 15 12 2020].

Kohli, S., 2019. Understanding a Classification Report For Your Machine Learning Model. [Online]
Available at: https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397
[Accessed 14 12 2020].

Mike,2020.gradient-boosting. [Online] Available:https://www.machinelearningmike.com/post/gradient-boosting
[Accessed 25 09 2020].

MUTHUKRISHNAN, 2018. understanding-the-classification-report-in-sklearn. [Online] Available:https://muthu.co/understanding-the-classification-report-in-sklearn/
[Accessed 11 12 2020].

Nicholson, C., 2020. Term Frequency-Inverse Document Frequency (TF-IDF). [Online] Available:https://wiki.pathmind.com/bagofwords-tf-idf.
[Accessed 05 12 2020].

Owino,S.,2019.Python-Programming-Language. [Online] Available:https://medium.datadriveninvestor.com/python-programming-language-ac762a3b5977
[Accessed 05 01 2021].

Ramos, L. P., 2021. Python's .append(): Add Items to Your Lists in Place. [Online] Available:https://realpython.com/python-append/
[Accessed 25 01 2021].

Sambhav_Khurana,2019.geeksforgeeks. [Online] Available:https://www.geeksforgeeks.org/naive-bayes-classifiers/
[Accessed 25 10 2020].

Sameeruddin, S., 2020. HOW GRADIENT BOOSTING ALGORITHM WORKS. [Online] Available:https://dataaspirant.com/gradient-boosting-algorithm/
[Accessed 12 01 2021].

Sharma, Y., 2020. Understanding Count Vectorizer. medium, 8(Understanding Count Vectorizer), p. 8.

showmethebell,2017.split-of-train-and-test-data. [Online] Available at: https://dataandbeyond.wordpress.com/2017/08/24/split-of-train-and-test-data/
[Accessed 03 12 2020].

Sosnovshchenko, A., 2020. Machine Learning with Swift by Alexander Sosnovshchenko. [Online]
Available:https://www.oreilly.com/library/view/machine-learning
[Accessed 02 01 2021].

Spandan Ghose Chowdhury, S. R. ,. S. C., 2014. News Analytics and Sentiment Analysis to Predict. International Journal of Computer Science and Information Technologies, 10(Predict Stock Price Trends), p. 10.

stackoverflow, 2016. Random Forest with bootstrap = False in scikit-learn python. [Online] Available at: https://stackoverflow.com/questions/40131893/random-forest-with-bootstrap-false-in-scikit-learn-python
[Accessed 04 08 2020].

Sydorenko, I., 2021. What Is a Dataset in Machine Learning: Sources, Features, Analysis. What Is a Dataset in Machine Learning: Sources, Features, Analysis, 21 04.

Tavva, R., 2021. Bag of Words: Convert text into vectors. data-science-blog.com, 11 04, p. 10.

tutorialspoint,2021.Python-String-join()Method. [Online]
Available:https://www.tutorialspoint.com/python/string_join.htm
[Accessed 26 01 2021].

Upasana,2020.Introduction-to-Classification-Algorithms. [Online]
Available:https://www.edureka.co/blog/classification-algorithms/
[Accessed 05 02 2021].

Waseem, M., 2020. How To Implement Classification In Machine Learning?. [Online]
Available:https://www.edureka.co/blog/classification-in-machine-learning/
[Accessed 05 02 2021].

Waskom,M.,2020.seaborn:statistical-data-visualization. [Online]
Available:https://seaborn.pydata.org/
[Accessed 06 09 2020].

Weinberger, K., 2013. An alternative text representation to TF-IDF and Bag-of-Words. 08,Journal of ResearchGate, p-7.

wikipedia,2020. [Online]
Available:https://en.wikipedia.org.
[Accessed 02 03 2021].

Wu, B. M. R. T.-y., 2016. Data Pre-processing. SpringerLink, 435(Data Analysis), p. 435.

Zhan, X. F. &. J., 2015. Sentiment analysis using product review data. Journal of Big Data, 10(Sentiment Analysis), p. 10.

Appendix

List of tables

List of figures

List of equation

Python code

```
import pandas as pd

import numpy as np

import seaborn as sb

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

type(df)

df
```

```
sb.heatmap(df.isnull())

df.info()

df=df.dropna()

df.info()

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

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

# Renaming column names for ease of access
list1= [i for i in range(25)]
new_Index=[str(i) for i in list1]
data.columns= new_Index
data.head(5)

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

' '.join(str(x) for x in data.iloc[1,0:25])

headlines = []
for row in range(0,len(data.index)):
    headlines.append(' '.join(str(x) for x in data.iloc[row,0:25]))

len(headlines)

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier

## implement BAG OF WORDS
countvector=CountVectorizer()
traindataset=countvector.fit_transform(headlines)

type(traindataset)

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

# implement RandomForest Classifier
randomclassifier=RandomForestClassifier()
randomclassifier.fit(traindataset,train['Label'])

## Predict for the Test Dataset
test_transform= []
for row in range(0,len(test.index)):
    test_transform.append(' '.join(str(x) for x in test.iloc[row,2:27])) #join all
test data headlines
test_dataset = countvector.transform(test_transform) #perform countvectrozer for
test data
predictions = randomclassifier.predict(test_dataset) #give test data to random
forest,did prediction and store into prediction
```

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

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

from sklearn.naive_bayes import MultinomialNB


naive=MultinomialNB()
naive.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = naive.predict(test_dataset)

type(predictions)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

from sklearn.tree import DecisionTreeClassifier

test_decisiontree_classifier=DecisionTreeClassifier()

test_decisiontree_classifier.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = test_decisiontree_classifier.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

#gradient boosting classifier
from sklearn import ensemble

gd_clf=ensemble.GradientBoostingClassifier()
gd_clf.fit(traindataset,train['Label'])
```

```
## Predict for the Test Dataset
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 = gd_clf.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

from sklearn.neighbors import KNeighborsClassifier

knn_clf=KNeighborsClassifier()
knn_clf.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = knn_clf.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

from sklearn.linear_model import LogisticRegression

lr_clf = LogisticRegression()
lr_clf.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = lr_clf.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

from sklearn.svm import SVC

sv_clf= SVC()
sv_clf.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = sv_clf.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)


#TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier

## implement TF-IDF
tfidfVectorizer=TfidfVectorizer()
traindataset=tfidfVectorizer.fit_transform(headlines)

# implement RandomForest Classifier for TF-IDF
randomclassifier=RandomForestClassifier()
randomclassifier.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = tfidfVectorizer.transform(test_transform)
predictions = randomclassifier.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

naive=MultinomialNB()
naive.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = tfidfVectorizer.transform(test_transform)
predictions = naive.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

test_decisiontree_classifier=DecisionTreeClassifier()

test_decisiontree_classifier.fit(traindataset,train['Label'])
```

```
## Predict for the Test Dataset
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 = tfidfVectorizer.transform(test_transform)
predictions = test_decisiontree_classifier.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

gd_clf=ensemble.GradientBoostingClassifier()
gd_clf.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = tfidfVectorizer.transform(test_transform)
predictions = gd_clf.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

knn_clf=KNeighborsClassifier()
knn_clf.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = tfidfVectorizer.transform(test_transform)
predictions = knn_clf.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

lr_clf = LogisticRegression()
lr_clf.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = tfidfVectorizer.transform(test_transform)
predictions = lr_clf.predict(test_dataset)
```

```
matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)

sv_clf= SVC()
sv_clf.fit(traindataset,train['Label'])

## Predict for the Test Dataset
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 = tfidfVectorizer.transform(test_transform)
predictions = sv_clf.predict(test_dataset)

matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)
```

Wyrażam zgodę na udostępnienie mojej pracy w czytelniach Biblioteki SGGW w tym Archiwum Prac Dyplomowych SGGW

……………………………………….
(*czytelny podpis autora pracy*)