**Predicting Club Football Player Transfer Possibility**

**based on**

**Machine Learning and Ensemble Learning**

By

**Rifat Talukder C151112**

**Mahmudul Hasan C151118**

A thesis submitted in partial fulfillment of the requirements for the degree of

*Bachelor of Science in Computer Science and Engineering*

Approved By

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mohammed Mahmudur Rahman** **Supervisor**

Professor

Department of Computer Science and Engineering

International Islamic University Chittagong

\_\_\_\_\_\_\_\_\_\_\_\_\_

**Tanveer Ahsan Chairman**

Associate Professor

Department of Computer Science and Engineering

International Islamic University Chittagong

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. Mohammad Shamsul Arefin External**

Professor

Department of Computer Science and Engineering

Chittagong University of Engineering Technology

**Declaration**

We hereby declare that, in this article, except some specific reference, all the information, knowledge and datasets are used by following the academic rules and ethical code and conduct which includes collection, process and present. We are announcing and promising about not sharing any portions of this research for any other qualification of this university or institution as well as others.

Md. Rifat Talukder

ID: C151112

Mahmudul Hasan

ID: C151118

**ACKNOWLEDGMENTS**

We are grateful to some persons and institutions, for their immense supports to continue our thesis research. At first, we would like to praise of Almighty Allah (SWT) for all the opportunities have been given to us. It is our pleasure to show appreciation to the respected teachers, senior and the Department of CSE,IIUC to provide us with a platform and support to do such intellectual work.

We would like to express our deepest gratitude to our Honorable supervisor Mohammed Mahmudur Rahman, Assistant Professor, Department of CSE,IIUC for his patient guidance, enthusiastic encouragement and useful critiques of this research work. His willingness to give his time so generously, inspiration and encouragement have been very much appreciated without which this work could not have been possible.

we want to thank our research methodology course instructor Dr. Abdul Kadar Muhammad Masum for motivate us to express our self in a correct way in research. Besides, it is our sincere gratitude to the Honorable teachers of the thesis committee, Department of CSE,IIUC for their valuable feedback.

**TABLE OF CONTENTS**

1. **Introduction 9**

A. Motivation 9

B. Research Question 10

C. Report Outline 11

1. **Background 12**

A. Machine Learning 12

Linear Classification

Support Vector Machine

Naïve Bayes

Logistic Regression

B. Ensemble learning 15

Random Forest

AdaBoost

C. Features 19

Bag-of-word Model

TF-IDF Scoring

Word2vec

GloVe

D. Evaluation Matric 23

Confusion Matrix

Precision and Recall

F1

ROC

E. Summary 26

1. **Related Work 27**

A. Text Classification using a Bag-of-Words Approach 27

B. Naive Bayes in Text Classification 28

C. Support Vector Machine in Text Classification 28

D. Logistic Regression in Text Classification 28

E. AdaBoost in Text Classification 29

F. Random Forest in Text Classification 30

1. **Proposed model 31**

A. System Workflow 31

B. Dataset 32

C. Pre-processing 33

Tokenization

Lemmatization

Remove stop words

1. **Experiments & Evaluation 37**

A. Feature Extraction 37

Bag-of-Word

TF\*IDF

Word2vec

B. Classifier 40

Support Vector Machine

Naïve Bayes

Logistic Regression

Random Forest

AdaBoost

C. Comparison of the Proposed Model 48

D. Summary 49

**Conclusion 50**

A. Discussion 50

B. Future Research 51

**Reference 52**

**LIST OF TABLE**

Table 4.1: Ucto tokenization 34

Table 5.1: Result of machine learning approach (Bag-of-Word) 37

Table 5.2: Result of machine learning approach (TF\*IDF) 38

Table 5.3: Result of ensemble learning approach (GoogleNews) 38

Table 5.4: Result of ensemble learning approach (GloVe) 39

Table 5.5: Result of the Accuracies, Precision, Recall and F1 for SVM 40

Table 5.6: Result of the Accuracies, Precision, Recall and F1 for Naive Bayes 42

Table 5.7: Result of the Accuracies, Precision, Recall and F1 for Logistic Regression 43

Table 5.8: Result of the Accuracies, Precision, Recall and F1 for Random Forest 45

Table 5.9: Result of the Accuracy, Precision, Recall and F1 for AdaBoost 46

**LIST OF FIGURE**

Fig 2.1: Overview of Ensemble Classifier 16

Fig 2.2: Overview of Random Forest Trees 17

Fig 2.3: Values ofConfusion matrix 24

Fig 4.1: Work flow of the system 31

Fig 4.2: Club Football Player Transfer News Dataset 32

Fig 4.3: Stop word removing 36

Fig 5.1 : Comparison Chart of SVM, Naïve Bayes, Logistic Regression (BOW) 37

Fig 5.2: Comparison Chart of SVM, Naïve Bayes, Logistic Regression (TF-IDF) 38

Fig 5.3: Comparison Chart of Random Forest and AdaBoost (GoogleNews) 39

Fig 5.4: Comparison Chart of Random Forest and AdaBoost (Glove) 39

Fig 5.5: Result of the Accuracies, Precision, Recall and F1 for SVM 40

Fig 5.6: ROC Curve of BOW for Logistic Regression, SVM, Naïve Bayes 41

Fig 5.8: Result of the Accuracies, Precision, Recall and F1 for Naive Bayes 42

Fig 5.9:Precision, Recall and F1 Curve of BOW feature for Logistic Regression, SVM, Naïve Bayes 43

Fig 5.10: Result of the Accuracies, Precision, Recall and F1 for Logistic Regression 44

Fig 5.11: Precision, Recall and F1 Curve of TF-IDF feature for Logistic Regression, SVM, Naïve Bayes 44

Fig 5.12: Result of the Accuracies, Precision, Recall and F1 for Random Forest 45

Fig 5.13: ROC Curve of TF-IDF feature for Naïve Bayes Logistic Regression, SVM 45

Fig 5.14: Positive and Negative sentence feature scoring Using Glove 46

Fig 5.15: ROC Curve of GloVe for Random Forest and AdaBoost 46

Fig 5.16 : Result of the Accuracy, Precision, Recall and F1 score for AdaBoost 47

Fig 5.17: ROC Curve of GoogleNews in Random Forest and AdaBoost 47

Fig 5.18: Precision, Recall and F1 Curve of GoogleNews in RF and AdaBoost 48

**Chapter 1**

**Introduction**

Sports news site like [www.newsnow.co.uk](http://www.newsnow.co.uk), [www.90min.com](http://www.90min.com), [www.goal.com](http://www.goal.com), [www.express.co.uk](http://www.express.co.uk), [www.foxsports.com.au](http://www.foxsports.com.au), [www.espnfc.com](http://www.espnfc.com), [www.skysports.com](http://www.skysports.com), www.euronews.com have become extremely popular since their appearance and news. Today hundred of sports news portals site share their news about the club football player transfer.

In professional football, a transfer is the action taken whenever a player under contract moves between clubs. It refers to the transferring of a player's registration from one club to another. In general, the players can only be transferred during transfer window and according to the rules set by a governing body. In transfer window, before transfer of a player in a club we got many news from sports news agency about his transfer. So we collect these news as a data and we have to make a decision about his transfer possibility rate in a club using machine learning.

In this thesis, We researched the feasibility of using machine learning (ML), collection of training algorithms for classifying the chance of a transfer rate for football players. Research encompasses the whole process: data collection and note-up, extraction and engineering of the features, evaluation of the performance of the various algorithms.

**A.** **Motivation**

Sports news websites share their news at any time in today's connected world. The sports news site does not only serve as a cheap tool for communicating news on the game scores or next game analysis; it also uses it to share rumors and news from a football club transfer and the players ' movement who want to be more open to the public. The growing spectrum of sports news sites on the internet has led to the formation of these resources as a new source for the player transfer movement. The tracking of news media transmission in crisis has been attractive to the research community increasingly[1].

Researcher note that billions of articles that people leave each month can not be controlled by surveys. This highlights the need for automated methods in intellectual text analysis, allowing large amounts of information to be processed and the meaning of user messages to be understood in a short period of time. The most important and complex element of automated processing is this understanding of the meaning of message. The use of state-of - the art big data technologies and methods has already helped researchers to automate, in particular, data collection, data preparation, management and visualiation of content. These innovations enable extensive research to be carried out and the sports news in real time to be monitored.

The fields of natural languagery, computer linguistics, text mining, and a range of methods range from the master learning to rules based methods for text classification. Machine learning methods include model training on specific document collections. Many researchers have recently been working on determining people's feelings in various social media data collected. Their classification and grouping of data has been using well-known machine learning techniques. The comparison of the methods for the classification of feelings has not been carried out before and we therefore need to compare existing techniques used in the classification of social media texts in this thesis.

They should consider existing methods used to characterize data collected from the sports news portal, as the same approach can be used various datasets according to the relevant research. This article discusses the challenges stemming from the analysis and classification of sentiments. This thesis also includes the analysis of experiments and research to come to terms with similar issues. We already discovered one major trend through the concurrent research of which methods are utilized and which are successful: the output of the particular process depends mainly on the dataset. If there have been positive, negative, neutral data in the datasets, it depends on the sophistication of the data.

**B. Research Question**

Based on the observation discussed above we have arrived at the following primary research question.

**RQ 1.**How do standard machine learning and ensemble learning techniques applied to text classification compare to sports news site data?

Some techniques mentioned in Chapter 2 are not typically used to use text data. Data should be modified to apply these models to preprocess text. The second question of research leads to this:

**RQ 2.**Which preprocessing techniques are available for converting natural language text into a suitable format?

 To respond to these questions, we need experiments through various machine learning techniques, ensemble learning techniques and processing techniques of natural languages.

**C. Report Outline**

The rest of the thesis is organized as follows.

Chapter 2 contains background and text classification information on the news site of sports.

First of all, we define the classification of text, provide some background necessary for further analysis with machine learning and Ensemble methods. In segment two, Naive Bayes, Logistic Regression, SVM and ensemble learning systems: Random Forest and AdaBoost describe briefly the following machine learning methods.

The related literature and case studies are described in the 3rd chapter. What was the same solution proposed? Which challenges were there?

Chapter 4 describes the experimental set-up. Furthermore, the fourth chapter describes, analyzed and preprocessed data.

Several experimental sessions and their results are outlined in Chapter 5.

The conclusion and future work on the results obtained are provided in Chapter 6.

**Chapter 2**

**Background**

**A. Machine Learning**

The functions in machine learning are usually divided into two categories: supervised or unsupervised learning. The formula is given a correlated output degree in supervised learning, and therefore the goal is to look for the mapping between the two, which is a good way for new inputs to be generalized. In unsupervised learning, the formula is provided exclusively with associated degree input, so the aim is to find some structure inside the knowledge. We are eager to observe a supervised learning downside throughout that report. The goal is to make your mind knowledgeable of this type of brand new analysis. Despite the obvious methodology for the machine learning, one has to specify what the synthesis should consider as knowledge. The insights within training expertise were converted into the alternatives known as empirical and subjective assets. Numerous options are merged into a matrix that is used to enter the learning algorithm for the computer. Training information is translated to operational vectors, which are injected with their respective roles into the structure. The different supervised training algorithms there are outsized and are therefore tested during this study. The following is defined in Section 2.1.1: logistic regression, naive bayes and support vector machine.

**Linear Classification**

In this thesis, three classifiers have been evaluated which are all linear algorithms. We first explain momentarily what is the linear classification. Given a number of datasets, each of which actually applies to one of the two classes, the goal is to determine the class of each new data element. On the background of a value of a linear set, a vector classifier makes a decision. Linear classifications can be interpreted as the function y= f (w, x) which, by a set of weights w, maps a collection of inputs x towards an output y. The algorithm attempts to find a hyperplane that maximizes separation among the two categories (in the 2dimensional case, a line). The data points can be viewed as p-dimensional vectors in which the hyperplane (p 1) attempts to isolate the data points. The n data points with size p are:(x allow1, y1),. (2.1) when every x / tuned vector is p-dimensional and each yi is either 0 or 1 indicating the class of the x / tuned vector. Then we would like to consider the hyperplane which divides the points x= 1, which are yi= 1 from those with the yi=0. The convention must use 1 or 0 for both binary classes for logistic regression and neural network, but the convention must use 1 or −1 for SVM.

**Support Vector Machine**

Original delineations of the vector support machines (SVMs) were given for non-probabilistic binary classification by Cortes and Vapnik[1995]. A high-dimensional house is formed in order to make SVMs initial. On this feature house victimisation, a non-linear mapping or kernel works the feature vectors from the coaching area unit. An optimum separation hyperplane is provided during this feature house to separate the joy points from 2 categories. This optimum hyperplane maximizes the space to the coaching information of each categories, thus it's placed in such the simplest way that every information is as so much as potential from the hyperplane separating the 2 areas. The goal of this maximization is to reduce noise. In the event that a nonlinear mapping is used in an feature house that has sufficiently high Space Properties, two categories may invariably be separated by a Hyperplane[ Han and al. 2011]. This implies additional dimensions to separate the 2 category instances. An example of this, where a two-dimensional house is transformed into a three-dimensional house to accomoate the separating hyperplane. The specific kernel process choice selected in keeping with Han dynasty et al.[2011] sometimes had a limited effect on the reliability of the corresponding template.

The classification of substitute information shall be concluded by mapping it to a constant, non-linear mapping victimization home. The new feature vector will be classified in one in all 2 categories, supported by its position within the feature house diverse to the Hyperplane. An important advantage of SVMs, which allows feature selection less important, is that they function well in large field areas[ Hotho et al., 2005]. Because text data typically has huge numbers of choices, SVMs are terribly informed of text identification activities. The rule conjointly has its downsides: SVM coaching is extremely memory-intensive, and it will be troublesome to see what kernel to use for the mapping method.

**Naïve Bayes**

Naive Thomas Bayes area classification unit a sort of probabilistic classifications of applied mathematics. These classified options supported the coaching knowledge feature values and also the classified vector. The Bayes theorem7 of Naive Thomas Bayes classifier is mixed with the idea that each of its options is mutually independent. Bayes ' theorem may be used to reflect the category possibilities of the feature values by presuming the interdependence of options. The estimation would be advanced, although not the expectation, because dependency between options might be appropriate. Such dependencies unit can be computationally dearly obtained, and with massive quantities of choices. The shared autonomy principle is generally quite incorrect in text mining because the system for the field of language communication is clearly combined, allowing the individual terms to be related and not autonomous. Nonetheless, although the opinion is wrong, it is comparatively economic and productive to identify the Naive Thomas Bayes area unit because it's plain, and therefore area unit normal in text categorization employment[ Nigam et al., 2000, Tang et al., 2016]. The output is usually best with large sets of information but can be vulnerable.

The models of Naive Bayes are multiplied, each with its own strengths. These are mainly different in calculating the probabilities of features. For example, the Nairobian Naive bays of Bernoulli, Nairobi Bay and Gairobi Bay of Gaussian Naive. Multi-installation classifiers in Naive Bayes have been showt to perform usually better for scanty sets like text data[ Tang et al. 2016] than for other variations. These classification agents assume that the term frequency satisfies a multinomial distribution within a document. The probability of a term given a class in a Multinomial Naive Bayes model is defined as follows by [Manning et al. [2008]](#_bookmark91), with term x, class C, |X| as the vocabulary size of C and |FxC | as the number of occurrences of x in C.

P (X C) = |FxC |

|X| |FiC |

That means that in Multinomial Naive Bayes, the probability of a given term is the number of occurrences within the class divided by the total number of occurrences in this class of terms. It is clear to see that this likelihood calculation method is appropriate for the identification of text: If a word is often present in a file, it will likely be relevant to the subject of t. Smoothing can be paired with Naive Bayes systems. Smoothing gives each word frequency a certain percentage or value in order to avoid introducing value zero into a probabilities product words that do not appear in the training set. This would lead in reports not defined properly and terminology not included in training data as the chances for these words for each category were set to null. The question was overcome by adding an additional value to a chance that means that the possibility of an event can never be null and that identification can therefore happen.

**Logistic Regression**

Linear regression is one way to think of two variables ' partnership. Logistic regression is also a statistical model; it implies that the variable is a fixed, exponential relationship and a category log chances. The link between a classification and a predictor has the following symbolic expression with a single predictor and intercept term:

P(yi=tragedy)=logit−1(β0+β1xi)

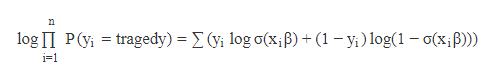
=eβ0+βxi1+eβ0+β1xi

=11+e−(β0+β1xi)

=σ(β0+β1xi)

Typically we have more than one observation.

Letting σ(xiβ)σ(xiβ) stand in for 11+e−(β0+β1xi)11+e−(β0+β1xi) the maximum likelihood estimate  for ββ is the value of ββ which maximizes the log likelihood of the observations:



While there is a closed-form solution for a linear regression frequently, logistic regression lacks a smooth solution. The solution (there actually is a single maximum) is typically found with less quadrants which are iteratively weighted.

**B. Ensemble learning**

The integration of research helps to improve the performance of machine learning with several models This approach allows greater predictive output than a single model. See Figure 2.1, how to identify the function of ensemble.

Tools together are meta-algorithms that combine multiple machine learning approaches into a single predictive model in order to minimize uncertainty (bagging), bias (boosting), and adjust predictions (stacking).

Ensemble methods can be divided into two groups:

* Sequential ensemble approaches where specific students (e.g. AdaBoost) are created sequentially. Sequential method's simple incentive is that the dependence among the base students is abused. Weighing previously incorrect cases of higher weight may improve the overall performance.
* Dynamic system approaches of dynamic development (e.g. Random Forest) for simple learners. Concurrent approaches have the basic motivation to take advantage of autonomy between the different students as a consequence of the average error reduction.

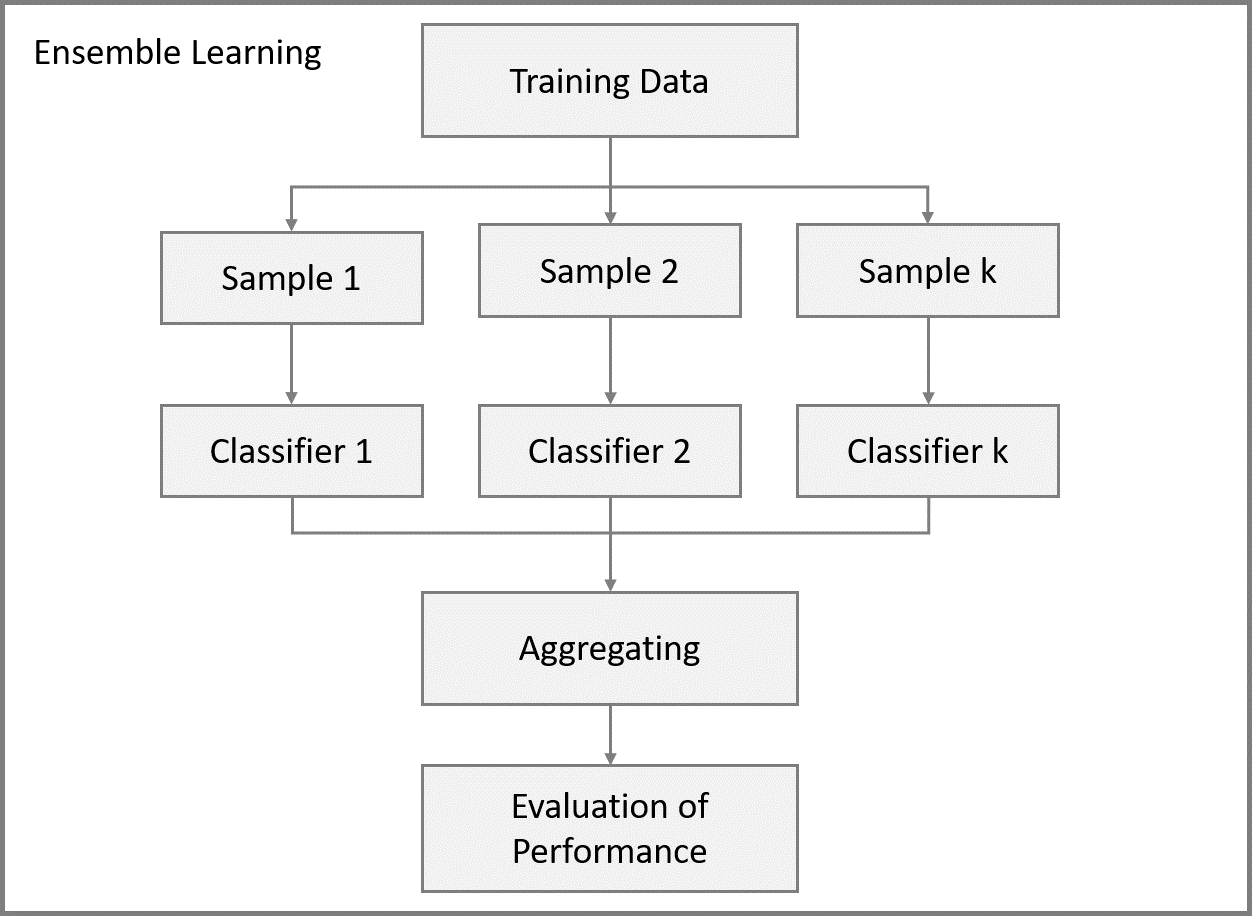


Fig 2.1 : Overview of Ensemble Classifier

Many ensemble approaches are based on a single base learning algorithm to generate standardized simple learners, i.e. students of the same kind, and contribute to standardized ensembles.

Some methods, i.e. learners of various kinds, that lead to heterogeneous ensembles, are also trained. The basic teachers need to be as accurate as possible and diverse as possible so that their combination methods are more accurate than any of their members.

**Random Forest**

Forest Random is another machine algorithm which follows the technique of bagging. The estimator algorithm is an extension. The random forest base estimators are decision-making trees. Unlike the meta-evaluation luggage, random forests use random features to determine the best splitting in each tree node.

In spite of it progressively, a random model of random forests has it:

1. Random subsets are created from the original dataset (bootstrapping).
2. At each node in the decision tree, only a random set of features are considered to decide the best split.
3. A decision tree model is fitted on each of the subsets.
4. The final prediction is calculated by averaging the predictions from all decision trees.

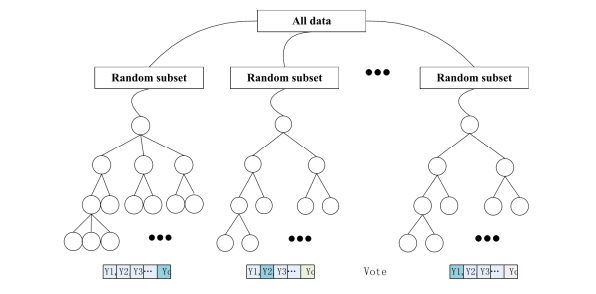


Fig 2.2: Overview of Random Forest Trees

That tree of the ensemble comprises of random forests of a specimen taken from the training site with a substitution (i.e. a bootstrap). Furthermore, a random subset of features is choosen, which further randomizes the tree in figure 2.2 rather than using all the features.

As a consequence, the forest differences are growing marginally, but their variability reduces due to the median of the less clustered species, resulting in a better overall template. Randomness goes a step further: The division limits are randomized in a highly randomized trees algorithm. Instead of choosing the most unequal threshold for every individual, the thresholds are drawn arbitrarily and the highest of these rands is selected as the law of segregation. It usually helps the variability of the design to be further minimized at the cost of a slightly higher partiality.

### AdaBoost

Adaptive boosting is one of the most basic boosting algorithms. AdaBoost For simulation, decision-making trees are typically used. Many sequential models are produced, each of which corrects the last model errors. AdaBoost assigns weights to the wrongly predicted observations and the following model works to correctly forecast these values. Adaboost is a meta-algorithm to improve classification findings, short to "Adaptive Boosting"[ Freund and Schapire, 1997]. This trains multiple identification instruments, each based on a sample of the entire training set. The AdaBoost algorithm begins with the complete training set by assigning weights to all instances. Every case gets the same weights initially. To train each model, a Si sample is taken on the basis of the weight of the individual instances from all training data. The increasing the weight, the more probable it is that the specimen may appear. All operational vectors are given the same weight and therefore parity of presentation during the initial phase. The test is used to train a simple classifier as training data11. The result is a Ci grade. This identification is checked using Si's own training data. The weights of the instances in the sample are adjusted to show the classification; if the instance was properly classified, its weight is reduced, and the weight of the instance is improperly classified. New classification techniques were conditioned on a new experiment in the following steps [ Schapire, 2013 ]. It places the classifier's emphasis on difficult situations.

In the last class, the outcomes of the entire range of classifications are merged. Nonetheless, the output weights the quality of each classifier. High-precision classifiers have higher weights, thus having a greater influence on the final rating. For classification, it implies that in the final classification of the AdaBoost classifier[ Han et al., 2011], weights for all classifier that allocate a certain class to an example and the class with the highest total weight is allocated. AdaBoost also improves efficiency when the simple classifiers used in the ensemble are already working fairly well. The algorithm[ Wang et al., 2009] is however susceptible to noise, as it can also incorporate outliers and atypical instances into the sample and is highly likely to have greater weights. In these cases, which causes the template to be overfit.

Below are the steps for performing the AdaBoost algorithm:

Initially, all observations in the dataset are given equal weights.

1. A model is built on a subset of data.
2. Using this model, predictions are made on the whole dataset.
3. Errors are calculated by comparing the predictions and actual values.
4. While creating the next model, higher weights are given to the data points which were predicted incorrectly.
5. Weights can be determined using the error value. For instance, higher the error more is the weight assigned to the observation.
6. This process is repeated until the error function does not change, or the maximum limit of the number of estimators is reached.

The predictions in the AdaBoost algorithm are then combined to produce the final prediction by means of a weighted majority (classification) or a weighted sum (Restriction). A difference is that the simple students are trained sequentialy on a stable version of the results, between boosting and committee approaches, such as labeling.

**C**. **Features**

One of the most important aspects of machine learning is to decide which features to use. For various types of text classification problems similar to the one in this thesis, the most common baseline approach is the bag-of-words model, TF-IDF and word2vec.

**Bag-of-word Model**

For the machine learning algorithm, we need a way to represent text information and the wordpack template allows us to accomplish the goal. It's easy to understand and apply the Word Bag model It is a way to extract features from the text for use in algorithms for machine learning.

In this method we use the tokenized terms to evaluate the intensity of each token for each finding.

Let’s take an example to understand this concept in depth:

“It was the best of times”  
“It was the worst of times”  
“It was the time of transfer”  
“It was the foolish decision”

We treat each sentence as a separate document and we make a list of all words from all the four documents excluding the punctuation. We get,

‘It’, ‘was’, ‘the’, ‘best’, ‘of’, ‘times’, ‘worst’, ‘time’, ‘transfer’, ‘foolish’, ‘decision’

The next step is the create vectors. Vectors convert text that can be used by the machine learning algorithm.

In this approach, each word or token is called a “gram”. Creating a vocabulary of two-word pairs is called a bigram model.

We take the first document — “It was the best of times” and we check the frequency of words from the 10 unique words.  
“it” = 1  
“was” = 1  
“the” = 1  
“best” = 1  
“of” = 1  
“times” = 1  
“worst” = 0  
“decision” = 0  
“transfer” = 0  
“foolish” = 0

Rest of the documents will be:  
“It was the best of times” = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]  
“It was the worst of times” = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]  
“It was the time of transfer” = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]  
“It was the foolish of decision” = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

In this approach, each word or token is called a “gram”. Creating a vocabulary of two-word pairs is called a bigram model.

For example, the bigrams in the first document: “It was the best of times” are as follows:  
“it was”  
“was the”  
“the best”  
“best of”  
“of times”

The process of converting NLP text into numbers is called **vectorization** in ML. Different ways to convert text into vectors are:

* Counting the number of times each word appears in a document.
* Calculating the frequency that each word appears in a document out of all the words in the document.

**TF\*IDF Scoring**

The word frequency-inverse file rate stands for TF-IDF. The weight of TF-IDF is a statistical measure used to assess the importance of a word for a document or corpus in a collection. By proportion to the number of times the term is shown in the text, the value is offset by the volume of the word in the corpus.

**Term Frequency (TF)**: It is an evaluation in the current document of the frequency of the word. Since each document's length is different, it is possible that long documents would show a term much more often than shorter documents. The frequency of the term is often divided by the length of the document.

**Inverse Document Frequency (IDF)**: is a scoring of how rare the word is across documents. IDF is a measure of how rare a term is. Rarer the term, more is the IDF score.

Thus,



While word frequency is a useful means of representing documents, it does not provide information on the use of one word in its entirety. It is often useful to know for the purposes of text mining whether a specific word is a common word or is used relatively much in a particular document. Such data is not given in Word frequencies. The use of term weighting can help to reflect these data by, say, having a lower weight to commonly used words in the body. TF-IDF, the word frequency-reverse data frequency Feldman and Sangar is a common term weighting method in text mining. The TF-IDF meaning for a term in a document shows how often the word is in that report throughout the whole corpus. A strong TF-IDF value suggests that the term ap-pear is relatively common in a text and can therefore be considered a characteristic of the report (Feldman & Sanger, 2007). A low value means that the term also occurs in a corpus or in a specific text only once. If the document does not contain a word, the TF-IDF is not defined. This can be viewed as an extremely small TF-IDF performance and is commonly known as 0. As in the case of words frequency vectors, binary counts can be taken when TF-IDF is calculated rather than term frequency. In this case, if the word occurs

in the document, the TF-IDF will be replaced by one or 0.

**Word2vec**

Mikolov et al. (2013) have introduced Word2vec as a word embedding technique that captures degree of similarity between languages. Word embedding is a mathematical method used in natural language processing to represent words or phrases in real-time number vectors, and word2vec is a software that generates such vectors, if words of similar meaning or background are located closely related in the space within the vector. Word2vec may render the Continuous Word Bag (CBOW) and Skip-gram, which each contain different features, with two different models. CBOW uses the words, but the forecasts are not based on the sequence. Based on the current term, skip-gram forecasts the corresponding terms. According to Mikolov et al.(2013), the overall Skip-gram model enhances the quality of words vectors, particularly in semantics, and also increases computational time to approx. one factor of 3 for less occurring words. The resultant term vectors have somewhat unexpected and curious properties. Looking at the Word2vec template conditioned by Swedish Twitter Messages2, Stockholm, Gothenburg, Malmö and Uppsala are the 5 closest Vector words, indicating that in vector space the gap between cities (wide cities in Sweden) catches a certain amount of similarity. The five nearest vectors are: hound, kanin, liked, kissed and kitten. Another instance is the term. Words that often happen in sentences near each other will have vectors in the vector space close together.

**GloVe: Global Vectors for Word Representation**

GloVe is an unregulated computing method for the processing of term vectors. The training is done in aggregate global word-word co-occurrence corpus statistics and shows interesting linear substructures in the word vector area in the resulting representations. A common spirit quantifies the relation between two terms with the resemblance measures used in closest neighbor assessments. This simplicity may be problem-free, as two words are almost always more complex than a single number could capture. Of example, men can be seen to be similar to women by both words, which describe human beings; on the other hand, the two terms are often seen as opposites as they indicate a primary orientation that people differ. A template should equate more than one number with a word pair in order to capture the ambiguity that is needed to differentiate between man and woman in a numerical manner. The vector difference between the two word vectors is a natural and simple candidate for an enlarged series of discriminatory numbers. GloVe is designed in order that such vector differences capture as much as possible the meaning specified by the juxtaposition of two words. The GloVe model is trained on the non-zero entries of a global word-word matrix, which shows how frequently words co-occur in a given body. To order to fill this array, the data need to be obtained through the whole corpus. This pass can be expensive for large companies, but it is a one-off cost in advance. The learning cycles are much quicker, as there is usually a much lower number of non-zero matrix entrants than the actual number of words in the corpus. The resources in this program simplify the data-co-occurrence compilation and preparation for design output.

**D. Evaluation Matric**

**Confusion Matrix**

The effects of a classification method can be visualized using an uncertainty matrix. The algorithm may be used to determine whether the test specimen is either 0 or 1 for binary cases with 1 and 0 as the two possible outcomes. We can count four different metrics to calculate how well the algorithm performs: one is defined as positive and 0 is defined as negative:

1. True positive (TP), the algorithm classifies 1 where the correct class is 1.
2. False positive (FP), the algorithm classifies 1 where the correct class is 0.
3. True negative (TN), the algorithm classifies 0 where the correct class is 0.
4. False negative (FN), the algorithm classifies 0 where the correct class is 1.

The confusion matrix is simply these four values visualized in one table, see Fig 2.3

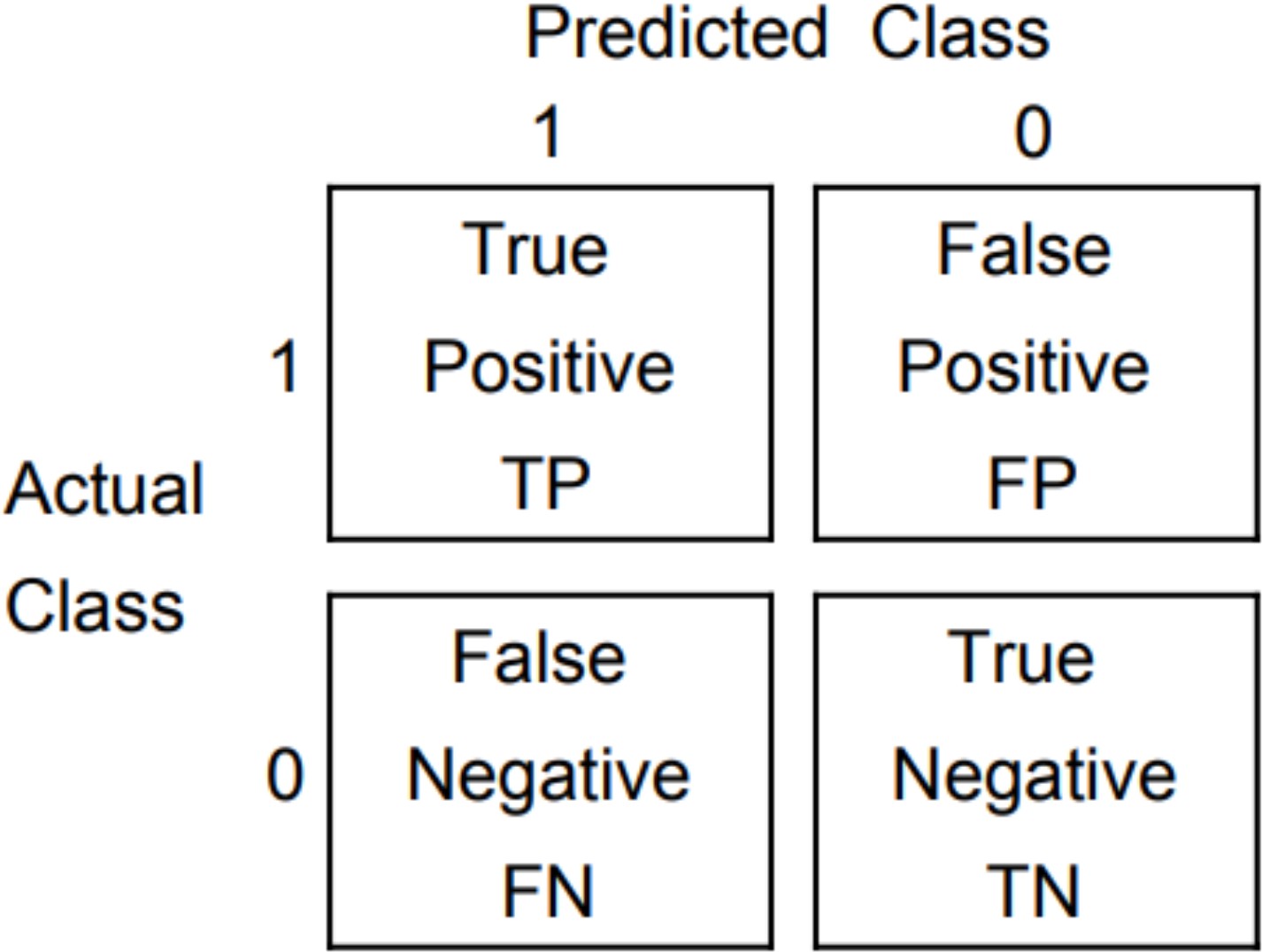


Figure 2.3: Values ofConfusion matrix

**Precision and Recall**

As a way to evaluate the performance of an machine learning algorithm, one can use precision and recall. Precision is defined as:

And Recall defined as:

A very high precision means that the algorithm classifies almost no inputs as positive unless they are positive. A high recall would mean that the algorithm misses almost no positive values.

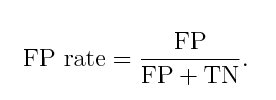
**F1**

The f-score (or F1-Score) is the harmonic mean of precision and recall. The f-score is defined as:

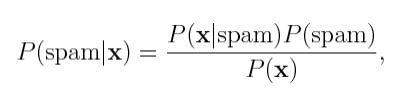
This means that an algorithm with a precision of 1 and recall 0 would still get a f-score of 0.

**ROC Curve**

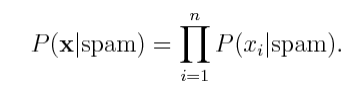
The receiver operating characteristic (ROC) curve is a more visual way of measuring the performance of a binary classifier. It is produced by comparison with the false positive rate (FPR) that has yet to be explicitly defined with the true positive rate (TPR) (or a reminder).



The question it answers is the following:  
“When it is actually the negative result, how often does it predict incorrectly?”  
we can obtain this curve. First, note that our Naive Bayes algorithm isn’t only able to predict if each email is spam or not, but it can also give us the **predicted probability** for such an event. Probability of an email being spam, for example, given its features, is given by Bayes’ theorem:



And we assume “naively” that the features are conditional independent of the class:



**E. Summary**

Sports News Site Text Classification is a huge domain which requires a background necessary to cover in this chapter. We began with the introduction to social media and the analysis of feelings. A variety of machine learning techniques have been listed in the second part of this article. The assessment measures are then represented. And we defined text representation methods as well. In the case of machine learning methods from all different fields only the methods used in the experimental session are described.

**Chapter 3**

**Related Work**

In this chapter, we present our research findings and related work on sentiment analysis. This study of the related works explains why we have chosen special methods in this work and makes a few comments about methods for improving their understanding of the work and results.

### A. Text Classification using a Bag-of-Words Approach

Previously it was intended to classify the subject of a given document by the bag of words approach. In a dataset of different news articles, Joachims [2] used this approach. To order to reduce the number of terms in these papers, a preprocessing method was carried out to delete the words to prevent them. Your tests showed that the reliability and memory of many topics speculated that it was more than ninety. On the other hand the exactness and reminder can come through the greatest score of 77, once this approach was applied on the data set of medical abstracts. The performance of a word bag approach will therefore vary with respect to the type of text to be classified. Over the years, numerous different types of text processing challenges are subject to a bag of words strategy. In an overly spam finding mission, Sahami et al. [3] used this strategy. The goal was to create a server that could isolate spam messages. Randomly, the data set consisted of a junk or legal e-mail array wherever the distribution was overwhelmingly beneficial to the junk class. The results showed that a category of Naive Bayes qualified mistreatment only in unigrammy forms alleged 97.1% reliability and ninety-four.3% of police mistreatment were identified through fake emails. The scoring was enhanced by including hand-made word phrases assisted by choices as well as domain-specific information. Nonetheless, the training classification mistreatment solutions alone will also reach a high performance in police work with fake e-mails. In a data set containing film reviews, Pang et al.[4] used the bag-of-word approach for classification of sentiments. A motion picture evaluation is graded by their study as positive or negative. The results have shown that an accuracy approach of up to 82.9 percent can occur. A human-a selected uniquote basis, which only rumored the greatest score of 69, was considerably superior in this score. An eighty two,9 percent score was significantly lower than the highest score in multiple categories, largely categorization task based on topics. We also helped this finding by the need to speedily translate emotions through user-generated content by additional technical means. Even in excessively large questions[6] the answering tasks that contribute to the simple use of an approach to a bag of words when the response region has the unit length of paragraph, wherever the aim is to find the text that contains the solution to a given question. However, Sta-matatus et al.[6] used such a method to categorize the kind of a particular document mechanically. According to the alternatives, their best performance obtained an error rate of two.5% in four classes by mishandling the amplitude of the unigrams. From these works it is rather fascinating, in an extremely new text classification task or dataset, to find the performance of a bag of words approach.

**B. Naive Bayes in Text Classification**

Pak and Paroubek are responsible for some of the major work in sentiment analytics using the Naive Bayes[7]. The learning data was obtained using the presumption that the emoticons in the document reflect the general feeling of the language. A large number of training data are obtained automatically using this hypothesis. This study employed an ensemble of two classifications from Naive Bayes; one was trained to use unigrams while the other used vocal tagging. The combined accuracy of the two classifiers was 74 percent. To achieve similar results as the previous study, Pang et al.[ 8] used a single Naive Bayes classification in a movie review corpus. Many Naive Bay models have been trained with a variety of features like voice taging, unigrams and bigrams. They achieved a 77,3 per cent classification precision considered a high level of achievement in the Naive Bayes classification in this field.

**C. Support Vector Machine in Text Classification**

Twitter data have been used to establish public feelings and to inform the model markets by researchers Ritterman et al.[8]. We often apply an SVM-based algorithm to evaluate microblog messages surrounding a particular topic in order to predict community feelings. The method was used in microblogs about an influenza pandemic, and prediction market figures from an independent source were compared with the results. Their research suggests that information from social media can be used as a collective proxy. Java[ 9] has created a BlogVox software to elicit views on a given topic from the blogosphere. BlogVox uses SVM to determine whether a blog post expresses an opinion after the pre-processing to remove spam and superfluous information. This differs from the detection of the topic because the data miner wants to know how people feel about a certain topic versus the topic.

**D. Logistic Regression in Text Classification**

Logistic regression (LR) is the application of linear techniques for regression for circumstances in which the output is categorical. Logistic Regression is commonly used in a number of data mining and machine learning topics in which LR represents one or more predictor variables with response variables[ 19]. Logistic regression algorithm (LR) was introduced to solve study problems in recent work, for example. Cheng and Eyke (2009), Kötziantis et al. (2009), Kötznik and Koziantis et al. (2004). Al (2003), Mittal (2009) and Felix (2014) respectively. Cheng and Eyke (2009) proposed to complete the Multilabel Classification in conjunction with Instance-based Learning and Logistic Regression[20]. In order to find the best fitting of the transfer template with student-learning results, Freyberger et al. (2004) suggested the Logistical Regression (LR) algorithm[21]. Rus et coll. (2009) tried to compare data processing outcomes using several machine learning approaches for the cognitive template identification of learners, for instance. Logistic Regression and Decision Trees[22] Naïve Bayes, network of Bayes, System Vector Machines (SVM). In the construction model of the transfer to predict students, the logistic regression proposed by Feng and Back (2009) may represent students of their knowledge[23]. Kotsiantiset. Kotsiantiset. Al (2003), with the help of a Neural Network, Decision Tree, the Naive Bayes, Instance-based Learning, Logistic Recovery and Vector Machine Support (SVM)[ 24] tried to classify student dropout predictions. Mittal (2009) has conducted space prediction work utilizing Twitter information using Linear Regression, Logistic Regression, SVM[25].

**E. AdaBoost in Text Classification**

Al-Radaideh et al. (2006)[10], by applying three different classification methods (ID3, C4.5 and NaifBayes), studied the performance in the decision tree model to predict the final grade of students studying at Yarmouk University in Jordan in 2005. They noticed that the estimation of the equations for the decision tree was the best model used. Kotsiantis et al. (2003)[11]compared six student retirement algorithms. There were 350 cases of statistical or categorical information in the database. We found the best performed algorithms were Naïve Bayes and Neural Network. Five classification methods for predicting course results have been compared with Wilhelmiina and Vinni (2006)[12] using very small datasets (125 and 88 rows). Multiple linear regression and vector machine classifiers for numerical data have been used whereas Naïve Bayes classifier variations have been used for category data. The results are that the grouping of Naïve Bayes was the best method for classification. Dekker et al. (2009)[13] provided a case study to forecast the failure of students showing the feasibility of various classifications and the cost-effective approach to training in more than 500 statistical and marginal datasets. The findings are helpful in contrast to other algorithms, including Bayes Net and JRip, when using basic cassifiers (J48 and CART). Kalles and Pierrakeas(2004)[14] researched various techniques for computer training: (decision trees, neural networks, Naive Bayes, instance-driven learning, regression of the mechanics, aid of vector machines). They also compared it to genetic algorithms based on decision-making trees. They analyze the academic performance of students based on the tasks of the students and derive short rules that explain and predict the achievement / failure of the final examinations.

**F. Random Forest in Text Classification**

Castillo et al.[ 15] conducted a study in the field of sentiment analysis using the decision tree algorithm. The primary focus of the study was on receiving recognition for Twitter tweets, but also the sentiment analysis was secondary. The J48 algorithm was used to characterize emotions in the Twitter database. The algorithm generated a precision of 70 percent by training the algorithm with manually noted instances. The weighted voting random forest in a movie review database was carried out by Tsutsumi et al[ 16] study. For each random tree in the forest the scoring criteria was used to assign a weighted ballot. Using this approach, 83.4 percent in a dataset of 1400 reviews have provided the algorithm precision. In order to analyze the mood of society through a piece of special news on Twitter posts, Kanakaraj and Guddeti[ 17] proposed extracting data. To increase the classification's accuracy, they have chosen to include, in particular, semantic and text-sensory disambiguation techniques (NLP). Different algorithms for machine learning are commonly used to solve problem classification. "Ensemble methods" in machine learning, combines the effect of multiple machine learning algorithms on the given problem set to obtain a better predictive power than its constituent algorithms by separately. The findings of the Decision Tree, the Random Forest, Extremely Randomized Trees and the Decision Tree regression on the study of Twitter emotions are studied in Kanakaraj and Guddeti. Tests to equate Ensemble's output to other machine learning algorithms such as SVM, Baseline, MaxEntropy and Naive Bayes are performed. Figure 3.1 indicates the typical findings of their analysis. In this situation, \hope "or \fear," the term Zhang et al.[18] reflects on psychological principles and coincides with a set of business measures.

**Chapter 4**

**Proposed model**

**A. System Workflow**

In this thesis, we proposed Machine Learning and Ensemble Learning approach text classification.

Figure 4.1 below represents workflow of the system proposed in this thesis.

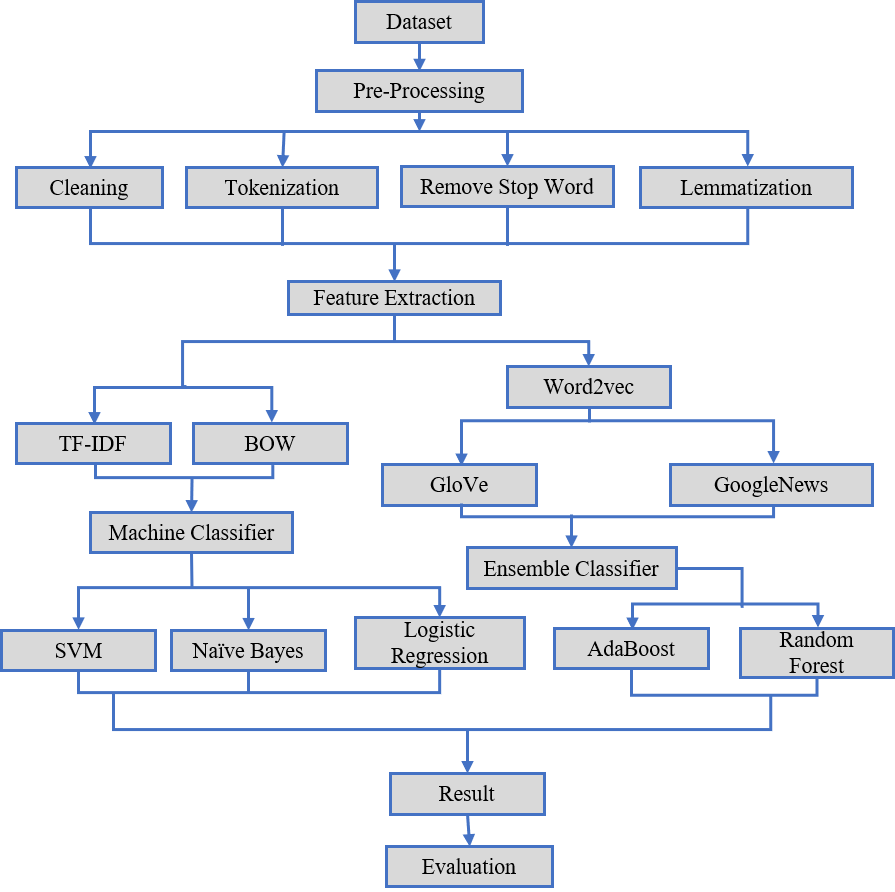


Figure 4.1: Work flow of the system

At the very beginning, we collect the club football player transfer news data from sports news sit by web scrapping. In the stage of preprocessing, the system splits the sentences into words then tokenize all sentences according to word by word. After that all punctuations are removed and words are stopped. The system also transfers word to its root version if a multi-illustrative event is occurring in different forms, so that a word is not misinterpreted under circumstances as different words.

when the preprocessing is successfully done the system moves to the feature extraction part. In this part, initially, we use three types of features extraction techniques. In machine learning approach we use BOW and TF-IDF mechanism for sentence scoring. And in ensemble learning approach we use Word2vec feature extraction techniques. In Word2vect feature extraction, we two pretrain model. One is GloVe and another is GoogleNews.

After feature extraction we measure the accuracy, precision, recall and f1 score using machine learning and ensemble learning classifier. We use Naive Bayes, SVM, Logistic Regression algorithm in machine learning approach. And we also use Random Forest and AdaBoost algorithm in ensemble learning approach.

**B. Dataset**

The dataset used for the experiments contains sentences sampled from player transfer news found on different sports news agency website like; [www.goal.com,www.skysports.com](http://www.goal.com,www.skysports.com) etc. On their website, they publish the all rumors and real news about to related a player transfer which can be read by other people who are visiting the site. We collect all these player transfer related data by scrapping these website. In first phase we collect 3000 data from sports news agency website and we label these data by manually. In future we will collect more data for make a real world application about measuring player transfer possibility rate.

|  |
| --- |
| **News** |
| Rooney didn’t want to leave Man Utd for City but I had to fight to keep him, admits Moyes. |
| There are certainly clubs interested in Kane. |
| Solskjaer confirms Man Utd are looking to sign a striker. |
| Maddison urged to snub Man Utd move by former Leicester boss Adams. |
| Messi could return for Barcelona's Champions League clash with Inter. |
| Japan legend Honda offers to play for Man Utd. |
| Barca want Chelsea's Willian on free transfer. |
| Liverpool boss Klopp confirms new contract talks with Milner. |
| Perez signs new Barcelona contract after first-team breakthrough. |
| Tobi Olanrewaju: Equatorial Guinea defender joins Sporting de Huelva from Rivers Angels. |
| Letsholonyane: Former Kaizer Chiefs and SuperSport United midfielder joins Highlands Park. |
| Chelsea youngster Anjorin to sign new deal as Lampard prepares to give him first-team chance. |
| Spurs have found a coach that is highly regarded, and one that guided them to the Champions League final in 2018-19. |

Fig 4.2: Club Football Player Transfer News Dataset

An annotator was used to label the sentences as an experience or not. The Experience category may also be considered a neutral class in this mission, while the non-experience group represents a negative class. During that process, the data set manually removed phrases which were not sensitive due to the abundance of dotting tokens and gibberish and phrases containing only one word. We used python library to retrieve player transfer news from sports news agency website was created using Python and the Beautiful Soup library. This library is specifically designed to create software that can scrape websites.

**C. Pre-processing**

In order to train a text data model, some preprocessing steps are required to make the data appropriate for model learning. Machine learning algorithms typically take statistical information to convert the texts into usable vectors. The first measures are primarily linguistically and can improve the classifier by removing redundant information or by structuring the data elsewhere. Most preprocessing is optional and depends on tasks and algorithms, so several options which are used or considered for this thesis are discussed below. Tokenization, lemmatization, removal of stop-words, etc.

**Tokenization**

Tokenization is one of the initial steps of preprocessing. The method of splitting a text into a particular part is tokenization, often letters, symbols and phrases. While it may appear an easy task to only cut the text into pieces in white spaces and dot marks, it can be very difficult. In most vocabulary limits for terms and phrases, white space and punctuation are not clearly defined. Punctuation in English can happen at the end of or contraction of a sentence(' it's','' Mr.',' Solskjaer'). There is no white space around terms within a sentence in other languages such as Chinese. Sentences must then be parsed so individual terms can be broken up. Such difficulties suggest that the information should be structured as consistently as feasible by tokenization. Next, the texts were separated into space-based parts. Such pieces are then contrasted to standard language words. If a match is found, the matching section of the fragment is marked with the corresponding regular expression as a word token. The remaining parts were known to be independent components which then fit.

An example of Ucto’s tokenization of the sentence Manchester United's Marcus Rashford could be out for a little while, manager Ole Gunnar Solskjaer has said. is given in table [4.1](#_bookmark19)

|  |  |  |
| --- | --- | --- |
| Word | Word-level type | Sentence-level type |
| Manchester  United  's  Marcus  Rashford  could  be  out  for  a  little  while  ,  manager  Ole  Gunnar  Solskjaer  has  said  . | WORD  WORD  SUFFIX  WORD  WORD  WORD  WORD  WORD  WORD  WORD  WORD  WORD  PUNCTUATION  WORD  WORD  WORD  WORD  WORD  WORD  PUNCTUATION | BEGINOFSENTENCE NEWPARAGRAPH  NOSPACE ENDOFSENTENCE |

Table 4.1: Ucto tokenization

**Lemmatization**

In text mining, it is important to build the feature-vectors used to train a template that a term appears in a file. A word often found in a single document will probably say something about the subject of the text. Nevertheless, it is not as easy to list word occurrences in text documents as it often seems. Words that shift the Word from its basic form or lemma may be inflected in many languages–identified as tense, grammatical or other data. One example is' running–moving–heading' in English. Since the inflected terms are typically formed differently than their lemmas, two lemma-like words are sometimes regarded as occurrences of various words. If ' running' and ' driving' all contribute against ' moving,' the term ratios are intuitively clear. Lemmatization is the mechanism by which inflexions can be eliminated and dictionary terms minimized, which can boost statistic counting [ Bonatti et al., 2016]. For example, the "good" lemma is "Better," "find" lemma is "found." This indicated that after lemmatization, the level of "poor" in the sentence I felt that the danger was great, but fans liked neymar more than 2 because both "nice" and "better" were listed in terms of "good" intensity. Lemmatization is identical to stemming (a process that eliminates a term from inflection), but it is based on the meaning, unlike stemmed. Cutting unclear terms to the right lemma is typically used, although stoppings these words may contribute to counting the wrong words. Both a noun and a (form of a) verb can be the name "meeting." Both stems are reduced to the same form ("meet"), whereas the lemmatizer chooses the appropriate context-based basic form ("meet"(verb) or" meeting"(noun)).

**Remove stop words**

“Stop words” are the most common words in a language like “the”, “a”, “on”, “is”, “all”. These words do not carry important meaning and are usually removed from texts as shown in Fig 4.2. It is possible to remove stop words using NLTK, a suite of libraries and programs for symbolic and statistical natural language processing. Like;

**Input**= “Liverpool manager Jurgen Klopp has described the signing of Joel Matip on a free transfer from Schalke 04 as the club has done in recent years.”

**Output=**[‘ Liverpool’,’ manager’,’ Jurgen’,’ Klopp’,’ described’,’ signing’,’ Joel Matip’,’ free’,’ transfer’,’ from’,’ Schalke’,’ club’,’ done’,’ recent’,’ years’,’.’]

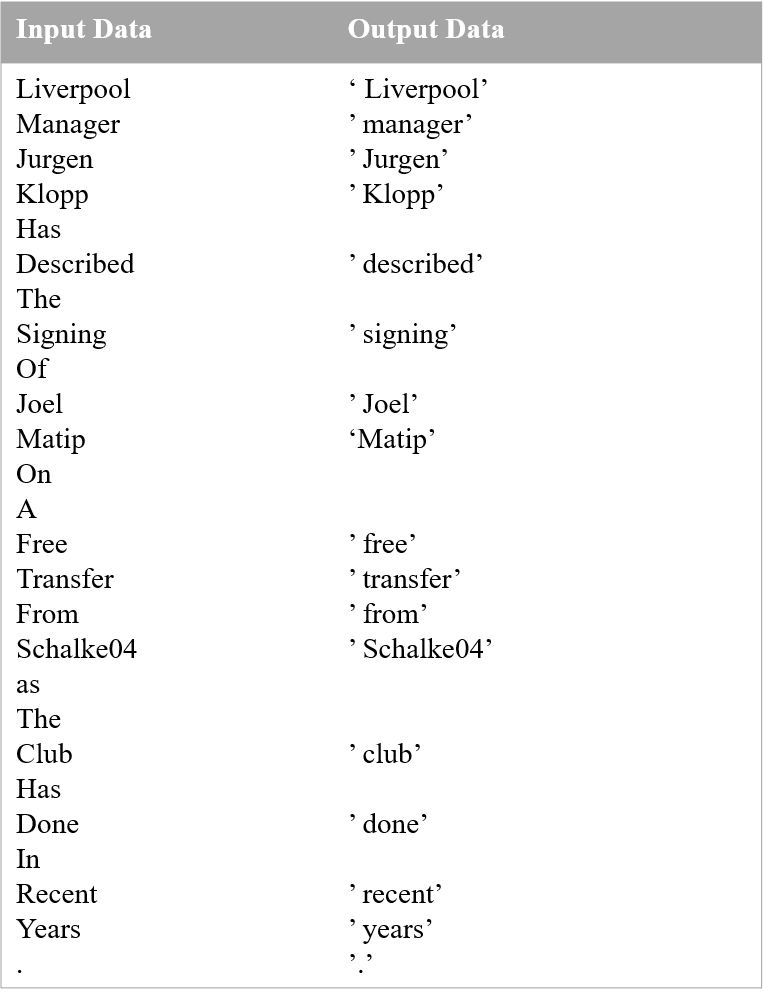


Fig 4.3: Stop word removin

**Chapter 5**

**Experiments & Evaluation**

This section discusses some testing sessions of Section 4 data sets. Within Chapter 2 they have used many machine learning methods. Chapter 4 describes all main experimental settings. Chapters 5.1 and 5.2 show how the SVM and Naive Bayes methods have been applied to data sets of club football players for text classification transfers. The following structure applies in this chapter. Instead, these data sets are accompanied by the following machine learning methods, Logistic regression (5.3), Random Forest (5.4) and AdaBoost (5.5). A section outlines the findings and improvement made during each study session.

**A. Feature Extraction**

One of the most important aspects of machine learning is to decide which features to use. For various types of text classification problems similar to the one in this thesis, the most common baseline approach is the bag-of-words model, tf-idf and word2vec.

**Bag-of-word:** We apply BOW method in over SVM, Naïve Bayes and Logistic Regression algorithm. Our machine learning model give better performance with Bag-of-Word approach as shown in Table 5.1 and Fig 5.1.

|  |  |  |  |
| --- | --- | --- | --- |
| Bag-of-word | | | |
|  | SVM | Naïve Bayes | Logistic Regression |
| Accuracies | 92.16 | 87.16 | 93.33 |
| Precision | 97.21 | 92.55 | 95.19 |
| Recall | 92.16 | 90.3 | 96.03 |
| F1 Score | 94.68 | 91.41 | 95.61 |

Table 5.1 : Result of machine learning approach (bag-of-word)

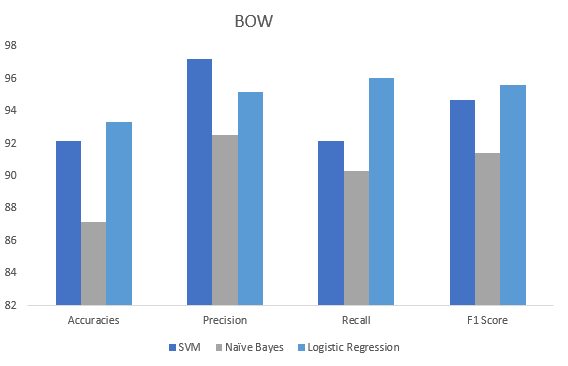


Fig 5.1 : Comparison Chart of SVM, Naïve Bayes, Logistic Regression (BOW)

**TF-IDF**

When we perform Bag-of-word model then there is some problem occurs. All the words consider as the same importance. So it’s a problem in Bag-of-word model. So overcome this problem we used TF-IDF. The result of TF-IDF feature extraction over SVM, Naïve Bayes and Logistic Regression algorithm as shown in Table 5.2 and Fig 5.2.

|  |  |  |  |
| --- | --- | --- | --- |
| **TF-IDF** | | | |
|  | SVM | Naïve Bayes | Logistic Regression |
| Accuracies | 92 | 81 | 81.83 |
| Precision | 96.78 | 80.15 | 82.92 |
| Recall | 92.51 | 99.56 | 97.35 |
| F1 Score | 94.59 | 88.81 | 89.56 |

Table 5.2: Result of machine learning approach (TF\*IDF)

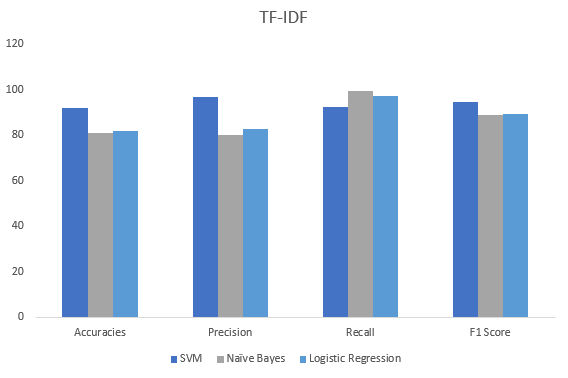


Fig 5.2: Comparison Chart of SVM, Naïve Bayes, Logistic Regression (TF-IDF)

**Word2Vec**

In this thesis, for ensemble learning approach we use Random Forest and AdaBoost algorithm. Also in feature extraction, we use Word2vec word embedding technique. For word embedding purpose we used two pretrained model one is GoogleNews and another is Glove. The result of GoogleNews Word2vec as shown in Table 5.3 and Fig 5.3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GoogleNews | | | | |
| **Column1** | **Accuracy** | **Precision** | **Recall** | **F1** |
| Random Forest | 93.66 | 93.82 | 98 | 95.86 |
| AdaBoost | 94.16 | 95.4 | 96.88 | 96.14 |

Table 5.3: Result of ensemble learning approach (googlenews)

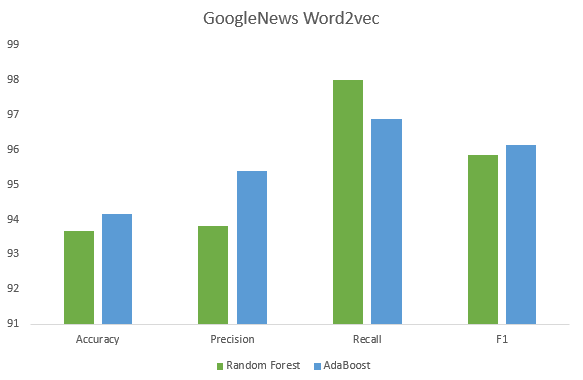


Fig 5.3: Comparison Chart of Random Forest and AdaBoost (GoogleNews)

The result of Glove as shown in Table 5.4 and Fig 5.4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Glove | | | | |
| **Column1** | **Accuracy** | **Precision** | **Recall** | **F1** |
| Random Forest | 93 | 93.96 | 96.88 | 95.4 |
| AdaBoost | 92.83 | 94.72 | 95.77 | 95.24 |

Table 5.4: Result of ensemble learning approach (glove)

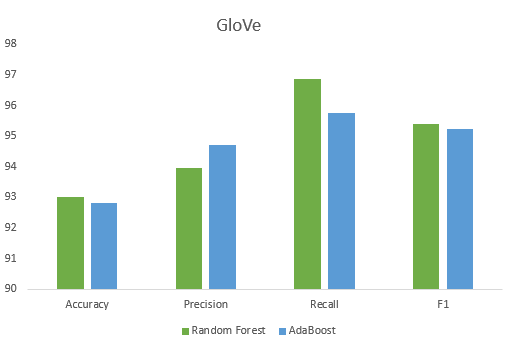


Fig 5.4: Comparison Chart of Random Forest and AdaBoost (Glove)

**B. Classifier**

**Support Vector Machine**

This experimentation deals with Support Vector Machines and its implementation in Jupyter notebook was chosen as a tool. However as it was described in theory section SVM has parameters and they affect models performance. The main disadvantage of SVM models is that it is very slow: time to build a model depends on the complexity of this model and with the growing parameters values can increase exponentially.

The accuracy, precision, recall and F1 score for the testing data using TF-IDF Vectorizer and Bag-of-Word over SVM are presented in Table 5.5:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Support Vector Machine** | | | | |
|  | Accuracy | Precision | Recall | F1 |
| TF-IDF | 92.0 | 96.78 | 92.51 | 94.59 |
| BOW | 92.16 | 97.21 | 92.29 | 94.68 |

Table 5.5: Result of the Accuracies, Precision, Recall and F1 for SVM

We run the SVM method for club football player transfer news dataset and find out the accuracy and all features. In TF-IDF and BOW over SVM, the BOW model gives better performance then TF-IDF. An only recall is 92.51% in TF-IDF feature extraction, which is better than the BOW model. Also, the accuracy of TF-IDF is 92% and in BOW 92.16% as shown in Fig 5.5.

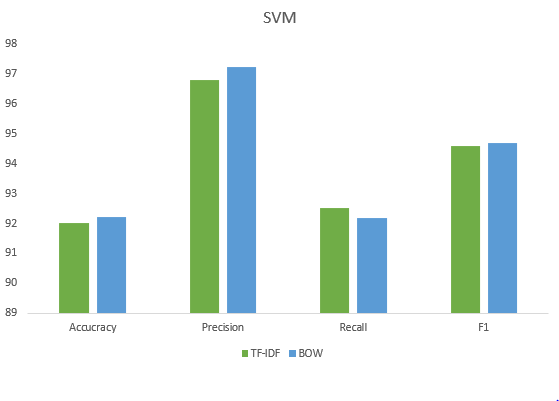


Fig 5.5: Result of the Accuracies, Precision, Recall and F1 for SVM

If we take a look at the Fig 5.6 and Fig 5.7, in both ROC curve and Precision,Recall,F1 for BOW model, SVM gave better performance than Naïve Bayes and Logistic Regression algorithm.

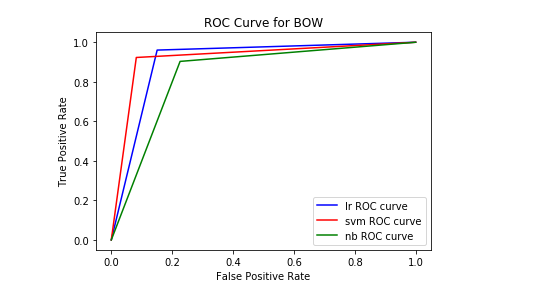


Fig 5.6: ROC Curve of BOW for Logistic Regression, SVM, Naïve Bayes

As a result, the SVM model using BOW feature extraction achieved better performance than TF-IDF Vectorizer which we saw in Table 5.5 .

If we take a look at the Fig 5.6 and Fig 5.7, in both ROC curve and Precision,Recall,F1 for TF-IDF feature extraction, SVM gave better performance than Naïve Bayes and Logistic Regression algorithm.

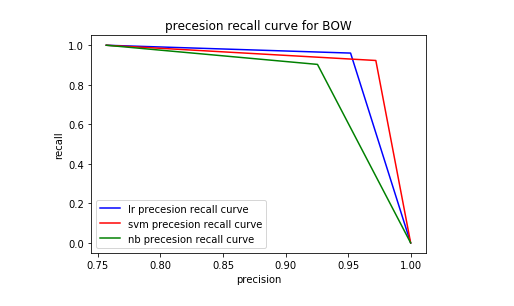


Fig 5.7: Precision, Recall and F1 Curve of BOW feature for Logistic Regression, SVM, Naïve Bayes

**Naïve Bayes**

In the first experiment probabilistic method namely Naive Bayes Classifier are used. For experiments, Jupyter notebook implementation was chosen since Jupyter is open-source software, has no limits for the number of instances and contains tools for data preprocessing, visualization, classification, regression, rules, and others. Accuracy, precision, recall, and F-measure were chosen as evaluation measures. The last three measures are measures to evaluate the quality of algorithms only concerning one specific class (positive or negative). The accuracy the metric is convenient for multiclass classification tasks to account for imbalanced test data.

The accuracy, precision, recall and F1 score for the testing data using TF-IDF Vectorizer and Bag-of-Word over Naïve Bayes are presented in Table 5.6 and Fig 5.8:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Naïve Bayes** | | | | |
|  | Accuracy | Precision | Recall | F1 |
| TF-IDF | 81.0 | 80.15 | 99.56 | 88.81 |
| BOW | 87.16 | 92.55 | 90.30 | 91.41 |

Table 5.6: Result of the Accuracies, Precision, Recall and F1

for Naive Bayes

We run the Naïve Bayes method for club football player transfer news dataset and find out the accuracy and all features. In TF-IDF and BOW over Naïve Bayes, the BOW model gives better performance then TF-IDF. The only recall is 99.56% in TF-IDF feature extraction, which is better than the BOW model. Also accuracy of TF-IDF is 81% and in BOW 87.16%, precision of TF-IDF is 80.15% and in BOW 92.55%, F1 score of TF-IDF is 88.81% and in BOW 91.41%.

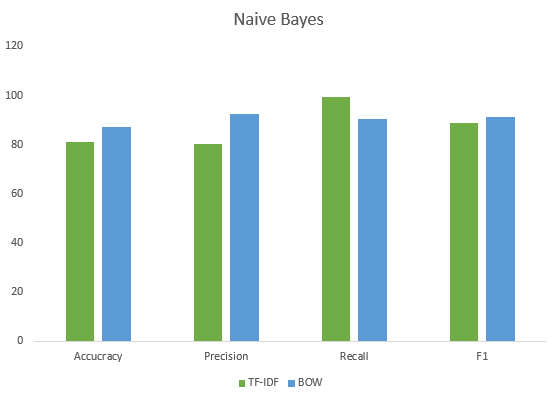


Fig 5.8: Result of the Accuracies, Precision, Recall and F1

for Naive Bayes

As a result, the Naïve Bayes model using BOW feature extraction achieved better performance than TF-IDF Vectorizer which we saw in Table 5.6.

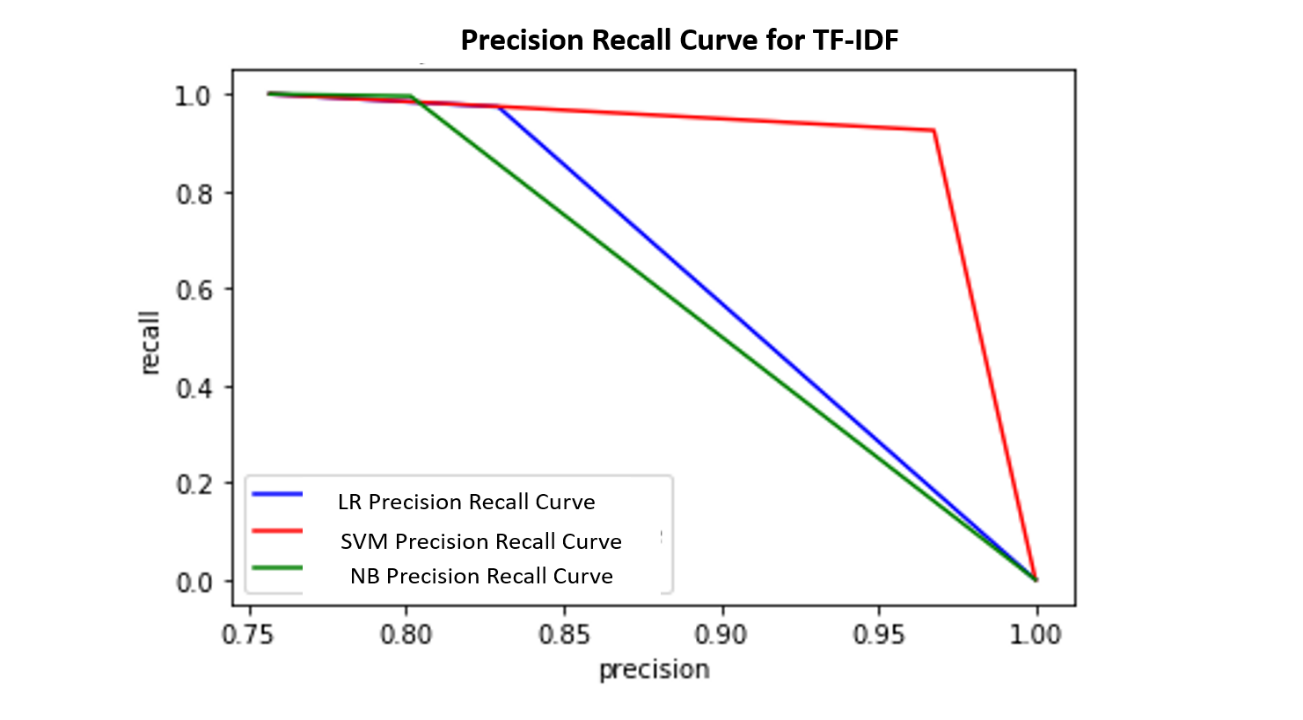


Fig 5.9: Precision, Recall and F1 Curve of TF-IDF feature for Logistic Regression, SVM, Naïve Bayes

**Logistic Regression**

This experimentation deals with the Logistic Regression method and its implementation in Jupyter was chosen as a tool.

The accuracy, precision, recall and F1 score for the testing data using TF-IDF Vectorizer and Bag-of-Word over Logistic Regression are presented in Table 5.7 and Fig 5.10:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logistic Regression** | | | | |
|  | Accuracy | Precision | Recall | F1 |
| TF-IDF | 82.83 | 82.92 | 97.35 | 89.56 |
| BOW | 93.33 | 95.19 | 96.03 | 95.61 |

Table 5.7: Result of the Accuracies, Precision, Recall and F1 for Logistic Regression

We run the Logistic Regression method for club football player transfer news dataset and find out the accuracy and all features. The accuracy of TF-IDF is 82.83% and in BOW 93.33%, precision of TF-IDF is 82.92% and in BOW 95.19%,recall of TF-IDF is 97.35% and in BOW 96.03%, F1 score of TF-IDF is 88.81% and in BOW 95.61%.

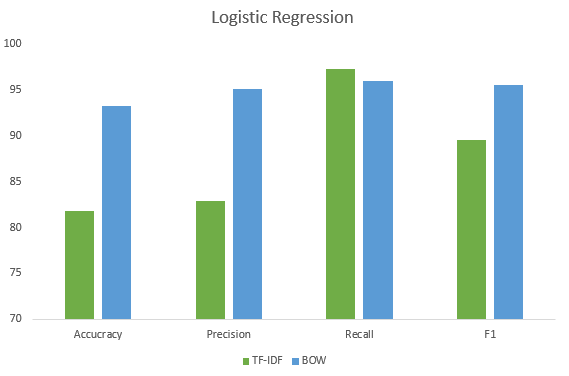


Fig 5.10: Result of the Accuracies, Precision, Recall and F1 for Logistic Regression

As a result, the Logistic Regression model using BOW feature extraction achieved better performance than TF-IDF Vectorizer.

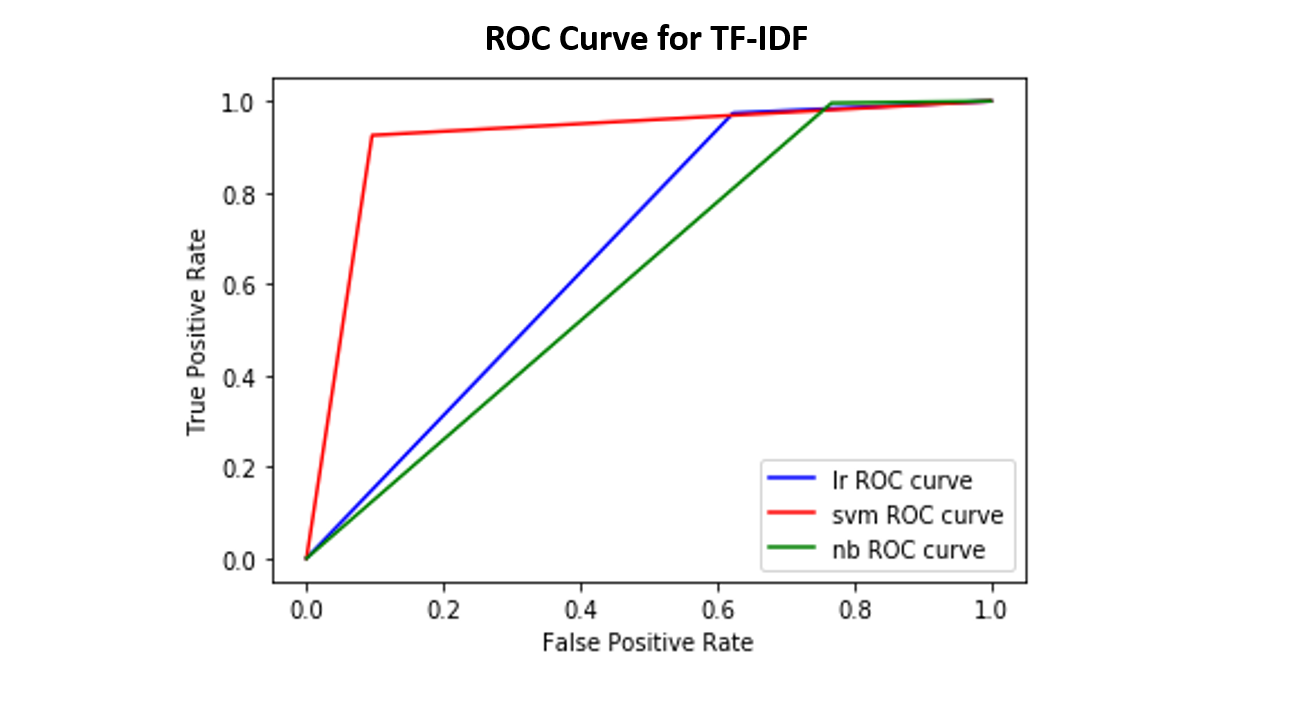


Fig 5.11: ROC Curve of TF-IDF feature for Logistic Regression, SVM, Naïve Bayes

**Random Forest**

This experimentation deals with the Random Forest method and its implementation in Jupyter was chosen as a tool. Random Forest method for classification is implemented in Jupyter as RandomForest. We run the Random Forest classifier with 100 random trees. As we have seen in the literature (Chapter 3) the Random Forest algorithm can produce a high performance for text-based classification. By combining multiple simple random trees the Random Forest algorithm can produce significantly higher performance than each tree individually. For such a simple algorithm the accuracy is high.

The accuracy, precision, recall and F1 score for the testing data using GoogleNews Word2vec and Glove over Random Forest are presented in Table 5.8 and 5.12:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Random Forest** | | | | |
|  | Accuracy | Precision | Recall | F1 |
| GoogleNewsWord2vec | 93.66 | 93.82 | 98.0 | 95.86 |
| Glove | 93 | 93.96 | 96.88 | 95.4 |

Table 5.8: Result of the Accuracies, Precision, Recall and F1 for Random Forest

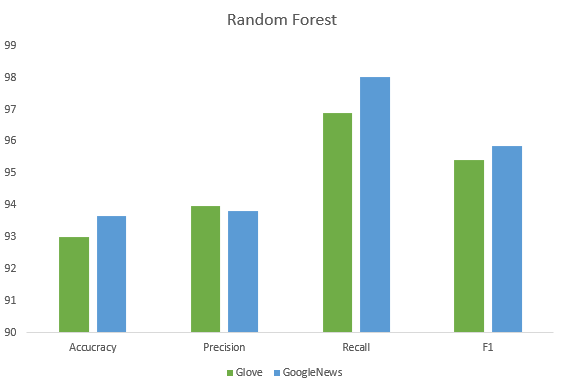


Fig 5.12: Result of the Accuracies, Precision, Recall and F1 for Random Forest

We use the word embedded system in this dataset and we apply GoogleNews and GloVe over random forest algorithm. In Glove the accuracy is 93% and also when we apply GoogleNews Word2vec in this dataset, in there also accuracy is 93.66%.

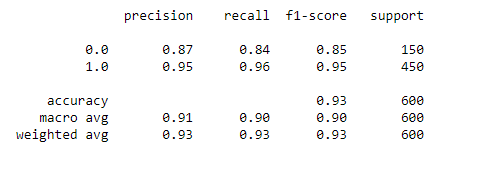


Fig 5.13: Positive and Negative sentence feature scoring Using Glove

As a result, GoogleNews Word2vec give better performance over GloVe. We saw in the Fig 5.13,where positive sentence feature scoring is greater then negative sentence in Glove.

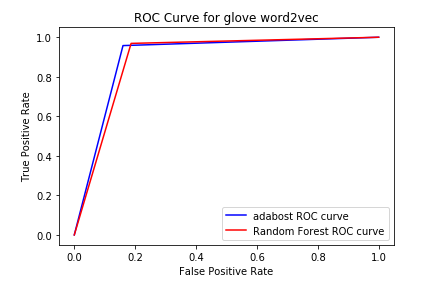


Fig 5.14: ROC Curve of GloVe for Random Forest and AdaBoost

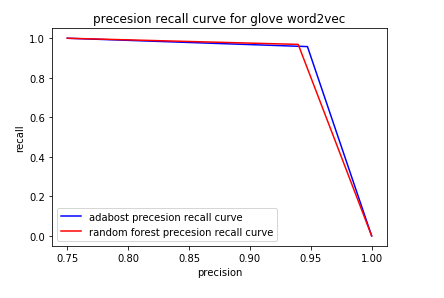


Fig 5.15: Precision, Recall and F1 Curve of GloVe for Random Forest and AdaBoost

If we take a look at the Fig 5.14 and Fig 5.15, it will be clear that Random Forest classifier also good for text classification.

**AdaBoost**

This experimentation deals with the AdaBoost method and its implementation in Jupyter was chosen as a tool. AdaBoost method for classification is implemented in Jupyter as AdaBoost.

The accuracy, precision, recall and F1 score for the testing data using GoogleNews Word2vec and Glove over AdaBoost are presented in Table 5.9 and Fig 5.16:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AdaBoost** | | | | |
|  | Accuracy | Precision | Recall | F1 |
| GoogleNewsWord2vec | 94.16 | 95.40 | 96.88 | 96.14 |
| Glove | 92.83 | 94.72 | 95.77 | 95.24 |

Table 5.9: Result of the Accuracy, Precision, Recall and F1 score for AdaBoost

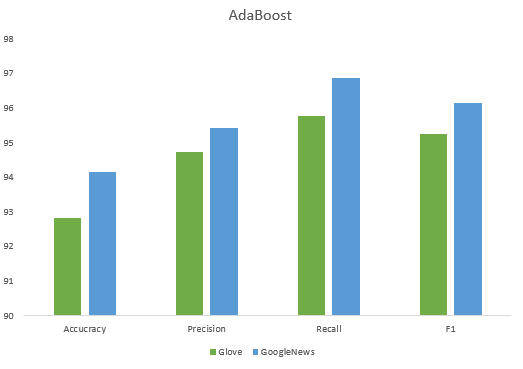


Fig 5.16 : Result of the Accuracy, Precision, Recall and F1 score for AdaBoost

Maximum accuracy is 94.16%% for the dataset with GoogleNews Word2vec features which is split in the range 80% for training and 20% for testing. Also in Glove the accuracy is 92.16%. The values of the accuracy of the Multilayer Perceptron method decreases with increasing dataset sizes.

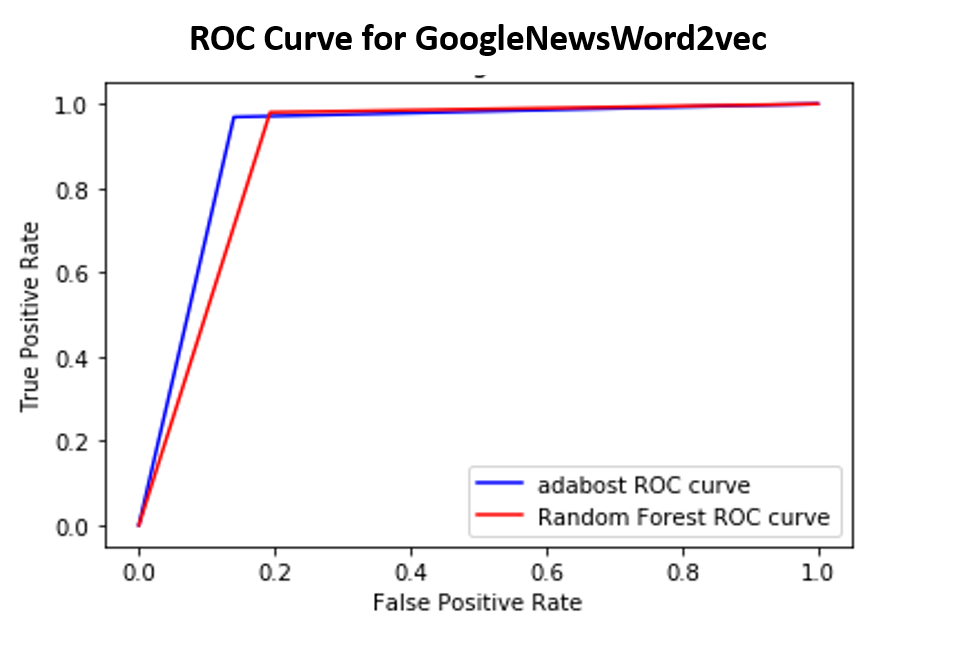


Fig 5.17: ROC Curve of GoogleNews Word2vec feature in Random Forest and AdaBoost

In this experiment, AdaBoost is perform better than Random Forest classifier, which in see in Fig 5.17. True and False positive rate are increasing in AdaBoost method. AdaBoost are performed better than all five algorithm. As a result, the finding of this research is AdaBoost algorithm is best for text classification.

If we take a look at the Fig 5.18,we saw that Precision, Recall and F1 score are also increased in GoogleNews Word2vec feature for AdaBoost.

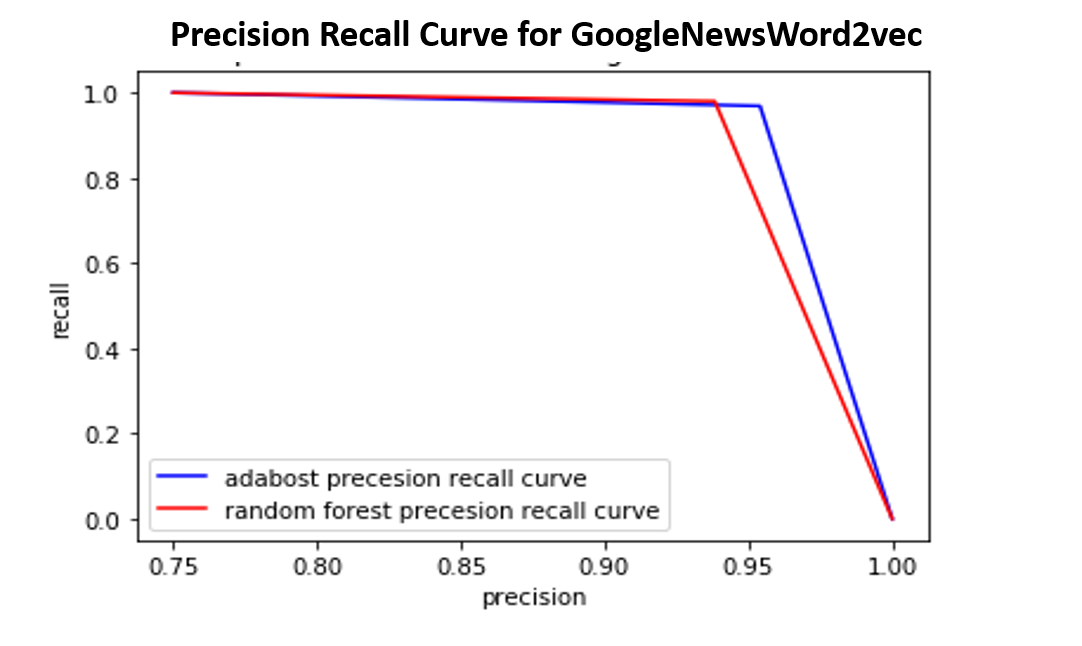


Fig 5.18: Precision, Recall and F1 Curve of GoogleNews Word2vec in Random Forest and AdaBoost

**C. Comparison of Proposed Model**

In this thesis, we used machine learning and ensemble learning both approach for analyzing the text classification algorithm. It’s a algorithmic text classification analysis over the club football player transfer news corpus data. We apply five different algorithm over this dataset. We use SVM, Naïve Bayes, Logistic Regression algorithm in machine learning model and we use Random Forest and Adaboost algorithm in ensemble learning model. For the feature extraction process, we use BOW, TF-IDF and Word2vec method. For word embedding purpose we used two pretrained model one is GoogleNews and another is Glove.

After the experiment over five classifier, we find that in machine learning approach SVM and Logistic Regression algorithm give the better performance which we shown in Fig 5.1 and Fig 5.2. In BOW feature extraction Logistic Regression give the better performance, which we saw in Fig 5.1. Also in TF-IDF feature extraction SVM give the better performance than Logistic Regression, which we saw in Fig 5.2. If we compare BOW and TF-IDF both feature extraction technique SVM is better than Logistic Regression.

In ensemble learning approach we use Word2vec feature extraction technique. We use two pretrain model, those are GoogleNews and Glove. In GoogleNews AdaBoost algorithm give the better performance than Random Forest as shown in Fig 5.3. But when we apply glove the performance of Random Forest is increase as shown Fig 5.4. If we compare both pretrain model, AdaBoost gives better performance than Random Forest. The final discussion of this thesis, in machine learning and ensemble learning approach, the ensemble classifier give the better performance than machine learning classifier in our club football player transfer news corpus dataset.

**D. Summary**

This section demonstrates the tests and test results performed in the sports news site for the data sets. The first study of the results of the tests (see Chapter 4) was conducted in chapters 5.1 and 5.2 of the Naive Bayes, SVM and Logistic regression approaches on our databases, which comprise of information of the sports news page. The most reliable way of classification of these data sets was identified. The data should be expressed in vector type as described in Chapter 2.5 before machine learning techniques were applied to text data. In purpose to find the best data representation, experiments were conducted and the Bag-of-Words model and TF-IDF model representation of the text were chosen. The Bag-of-Words model of representation of the text showed the best result for all methods. From Machine learning approach SVM and Logistic Regression methods performed best over Naïve Bayes method. The accuracy by a TF-IDF model of Dataset is 92.0% for SVM, 81% for Naïve Bayes. 82.83% for Logistic Regression and also the accuracy by a BOW model of Dataset is 92.16% for SVM,87.16% for Naïve Bayes,93.33% for Logistic Regression. Comparison between TF-IDF and BOW feature extraction, the BOW model is best for text classification using a machine learning approach. The performance of Random Forest and AdaBoost are not significantly different. The Random Forest approach has the best performance for the sample, with 100 random forests. The Random Forest method is slightly more efficient than every single tree by adding many simple random trees. The accuracy by the GoogleNews Word2vec model is 93.66% for Random Forest and 94.16% for AdaBoost. Also using GloVe the accuracy is 93% for Random forest and 92.83% in AdaBoost. Overall comparison in Word2vec feature extraction GoogleNews Word2vec is better than GloVe and overall ensemble learning approach, AdaBoost algorithm gives better performance than others. Comparing the performance of different methods we can say that Multinomial AdaBoost and Support Vector Machines classifiers represented high results.

**Chapter 5**

**Conclusion**

## **A. Discussion**

In this thesis, I tried to find an answer to the question how text mining can be used to classify club football player transfer news text . To make this question concrete, the goal was to train a model based on text mining for correct classification. Preprocessing steps and model options included linguistic preprocessing, multiple methods of feature construction and selection, boosting and resampling. Four different base algorithms were compared: Naive Bayes, Random Forest, SVM, Logistic Regression and AdaBoost.

In this thesis, AdaBoost, SVM, Random Forest and Logistic Regression algorithm seemed to have very limited effect on the performance of the different classifiers. This might be due to conditions that can be improved, such as the amount of data, but this cannot be stated with certainty. Other factors, such as filtering out terms of low relevance, had a large positive impact on the model performances. However, more research on different data sets is needed to make any general claims.

The results of this research are highly promising, and we show that text mining is a very practicable option for classifying player transfer data. The data set used in this research, however, is small and class imbalances are very great. It poses identification issues, provided that there are no positive examples that become difficult to generalize. The experiment would give a more robust and reliable result by repeating with a larger training set. If the training set is to be increased, several experts should note the data, reduce potential ambiguity and ensure that the annotation is clear and correct.

The effect on system quality of tuning the systems was not a priority in this investigation. However, hyperparameters that are appropriate for the data are likely to improve classifier performance. Ideally, the test covered less algorithms with a further analysis, on which parameter values a model most suitable for the data set was created. This relates in general to AdaBoost models, which are considered to rely heavily on performance tuning. In this sense, the AdaBoost models ' average performance is already quite good in this study, but it might be enhanced.

### B. Future Research

As mentioned above, one of the main disadvantages of this research was the training set size. Since football transfer news are headline when player transfer from one club to other club or at the beginning of the new season. That’s why it’s pretty much difficult for us to collect football transfer news corpus. This thesis did not have enough data to experiment with these algorithms, but with the large amount of labeled data in this set, chances of improvement are reasonably high.

SVM algorithms usually increase performance. In this study, SVM results were not very good: many results actually got worse when SVM was applied. A worthwhile option would be to include more text-specific SVM algorithms, to see whether these would perform better than Logistic Regression for this specific data set. More tuning on the (boosting) algorithms might also improve results. In this research, there was not enough focus on finding the optimal parameter values for each algorithm, which could have a large impact on the results.

Term normalization is yet another step not done but that may have a big effect on performance. Written text includes a number of synonyms, mistakes in pronunciation and alternatives. All these characteristics have been listed as different characteristics in this analysis, indicating that the average word incidence is smaller. A template could be made easier to distinguish specific features by incorporating a sort of tool to incorporate various spellings and synonyms within a common vocabulary similar to the lemmatization process. Throughout fact, the amount of features will greatly decrease, while improving memory and computing needs, rendering design learning easier and quicker.

**Reference**

[1] Vargas, Saúl, Richard McCreadie, Craig Macdonald, and Iadh Ounis. "Comparing overall and targeted sentiments in social media during crises." In *Tenth international AAAI conference on web and social media*. 2016.

[2] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." In *European conference on machine learning*, pp. 137-142. Springer, Berlin, Heidelberg, 1998.

[3] Sahami, Mehran, Susan Dumais, David Heckerman, and Eric Horvitz. "A Bayesian approach to filtering junk e-mail." In *Learning for Text Categorization: Papers from the 1998 workshop*, vol. 62, pp. 98-105. 1998.

[4] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79-86. Association for Computational Linguistics, 2002.

[5] Zakzouk, Tarik S., and Hassan I. Mathkour. "Comparing text classifiers for sports news." *Procedia Technology* 1 (2012): 474-480.

[6] Stamatatos, Efstathios, Nikos Fakotakis, and George Kokkinakis. "Text genre detection using common word frequencies." In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pp. 808-814. Association for Computational Linguistics, 2000.

[7] Pak, Alexander, and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." In *LREc*, vol. 10, no. 2010, pp. 1320-1326. 2010.

[8] Ritterman, Joshua, Miles Osborne, and Ewan Klein. "Using prediction markets and Twitter to predict a swine flu pandemic." In *1st international workshop on mining social media*, vol. 9, pp. 9-17. ac. uk/miles/papers/swine09. pdf (accessed 26 August 2015), 2009.

[9] Java, Akshay. "Mining social media communities and content." (2008).

[10] Al-Radaideh, Qasem A., Emad M. Al-Shawakfa, and Mustafa I. Al-Najjar. "Mining student data using decision trees." In *International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan*. 2006.

[11] Kotsiantis, Sotiris B., C. J. Pierrakeas, and Panayiotis E. Pintelas. "Preventing student dropout in distance learning using machine learning techniques." In *International conference on knowledge-based and intelligent information and engineering systems*, pp. 267-274. Springer, Berlin, Heidelberg, 2003.

[12] Hämäläinen, Wilhelmiina, and Mikko Vinni. "Comparison of machine learning methods for intelligent tutoring systems." In *International Conference on Intelligent Tutoring Systems*, pp. 525-534. Springer, Berlin, Heidelberg, 2006.

[13] Dekker, Gerben W., Mykola Pechenizkiy, and Jan M. Vleeshouwers. "Predicting Students Drop Out: A Case Study." *International Working Group on Educational Data Mining* (2009).

[14] Kalles, Dimitris, and Christos Pierrakeas. "Analyzing student performance in distance learning with genetic algorithms and decision trees." *Applied Artificial Intelligence* 20, no. 8 (2006): 655-674.

[15] Castillo, Carlos, Marcelo Mendoza, and Barbara Poblete. "Information credibility on twitter." In *Proceedings of the 20th international conference on World wide web*, pp. 675-684. ACM, 2011.

[16] Tsutsumi, Kimitaka, Kazutaka Shimada, and Tsutomu Endo. "Movie review classification based on a multiple classifier." In *Proceedings of the 21st pacific Asia conference on language, information and computation*, pp. 481-488. 2007.

[17] Kanakaraj, Monisha, and Ram Mohana Reddy Guddeti. "Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques." In *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pp. 169-170. IEEE, 2015.

[18] Zhang, Xue, Hauke Fuehres, and Peter A. Gloor. "Predicting stock market indicators through twitter “I hope it is not as bad as I fear”." *Procedia-Social and Behavioral Sciences* 26 (2011): 55-62.

[19] Menard, Scott. *Applied logistic regression analysis*. Vol. 106. Sage, 2002.

[20] Cheng, Weiwei, and Eyke Hüllermeier. "Combining instance-based learning and logistic regression for multilabel classification." *Machine Learning* 76, no. 2-3 (2009): 211-225.

[21] Freyberger, Jonathan E. "Using association rules to guide a search for best fitting transfer models of student learning." (2004).

[22] Rus, Vasile, Mihai Lintean, and Roger Azevedo. "Automatic Detection of Student Mental Models during Prior Knowledge Activation in MetaTutor." *International working group on educational data mining* (2009).

[23] Feng, Mingyu, and Joseph Beck. "Back to the Future: A Non-Automated Method of Constructing Transfer Models." *International Working Group on Educational Data Mining* (2009).

[24] Kotsiantis, Sotiris B., C. J. Pierrakeas, and Panayiotis E. Pintelas. "Preventing student dropout in distance learning using machine learning techniques." In *International conference on knowledge-based and intelligent information and engineering systems*, pp. 267-274. Springer, Berlin, Heidelberg, 2003.

[25] Mittal, Anshul, and Arpit Goel. "Stock prediction using twitter sentiment analysis." *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)* 15 (2012).