CLASSIFICATION OF YOUTUBE COMMENT SPAM USING TF-IDF AND MULTINOMIAL NAÏVE BAYES CLASSIFIER

NANG MYA OO

M.C.Sc. **JUNE 2022**

CLASSIFICATION OF YOUTUBE COMMENT SPAM USING TF-IDF AND MULTINOMIAL NAÏVE BAYES CLASSIFIER

By

NANG MYA OO

B.C.Sc(Hons:)

A dissertation submitted in partial fulfillment of the requirement for the degree of

Master of Computer Science

M.C.Sc.

University of Computer Studies, Yangon
JUNE 2022

ACKNOWLEDGEMENTS

Firstly, I would like to express my appreciation and special thanks to **Dr. Mie Mie Khin**, Rector of the University of Computer Studies, Yangon, for her kind permission to submit this dissertation.

I also would like to extend my sincere thanks to course coordinator, **Dr. Nan Saw Kalayar,** Professor and Head of Faculty of Information Science, University of Computer Studies, Taunggyi for her close supervision, proper guidance, valuable suggestions, advice and encouragement during the course of this work.

I would like to thank course coordinators, **Dr. Si Si Mar Win** and **Dr. Tin Zar Thaw,** Professor of Faculty of Computer Science**,** University of Computer Studies, Yangon for their superior suggestions and administrative supports during my academic study.

I am very grateful to **Daw Myat Myat Moe,** Associate Professor and Head of Language Department (English), University of Computer Studies, Taunggyi, and **U Myo Myat Naing**, Assistant Lecturer, English Department, University of Computer Studies, Yangon, for their advice, editing and suggestions from the language point of view.

I also like to acknowledge **all my teachers** who taught me throughout the master's degree course and **my friends** for their corporation and I wish to express my gratitude to **my beloved parents** for their invaluable support and encouragement to fulfill my wish.

Finally, I wish to express my sincere thanks to all of them who gave me valuable supports to accomplish my seminar as well as my thesis.

ABSTRACT

Nowadays, social networking such as Facebook, YouTube, Telegram, Instagram, etc. are very popular among people as IT technologies are developing more and more. Facebook and YouTube are the most popular social media platforms between young and old people, especially young people. The users can subscribe and give their opinion as the comments on YouTube. This system is developed with YouTube comment spam classification framework by using term frequency- inverse document frequency (TF-IDF) and Multinomial Naïve Bayes. TF-IDF is a statistical method to measure the weight or score of each word in each document to the whole corpus. This system is implemented using ASP.Net programming language on Microsoft Visual Studio 2015 IDE and Microsoft SQL Server 2017 Express Version as Database Engine. In this system, 1965 comments typed for five music videos of five singers (PSY, Katy Perry, LMFAO, Eminem, and Shakira) uploaded on YouTube are collected as data set. The purpose of this system is to categorize the YouTube comments into the suitable categories by using Multinomial Naïve Bayes Classifier and classify the comments as spam or legitimate (ham) depending on the contents in comment. Finally, the system evaluates the results with the accuracy (precision, recall and F-measure).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

YouTube is a video-sharing and video watching website that was first launched in 2005. It was later purchased by Google in 2006 and is currently one of the company's subsidiaries. Since then, YouTube has established itself as a major player in the video-sharing market. YouTube users can upload, watch, save, rate, add to favorites, report, download, subscribe movies, films, music videos, other video files on YouTube, and then can give opinions as comments on the interested videos. YouTube viewers consume about 1 billion hours of content per day and generally over 400 hours of content per minute.

YouTube's commenting system, which allows viewers or users to leave comments on videos uploaded to other channels, is one of the most popular features. It not only allows users to interact with one another while watching a video, but also exchange and share their opinions, feelings, etc.

The YouTube users are attacked by the spammers via comments written on the uploaded videos. There are several studies for detecting YouTube Spam to classify the comments as spam or ham by using machine learning techniques such as Support Vector Machine (SVM), k-nearest neighbor, Naïve Bayes, etc. When the system is implemented to classify the input textual comments as spam or ham by using Naïve Bayes classifier, the users are available to be aware of the spam to reduce. Finally, this system evaluates the accuracy with precision, recall and f-measure.

1.1 Objectives of the Study

The main purposes of this system are:

- To study the videos uploaded on YouTube , one of the most popular social websites
- To learn how to apply on YouTube comments and how to classify spam or ham using Multinomial Naïve Bayes approach.
- To learn feature selection (TF-IDF) method for YouTube comment spam classification
- To evaluate how the system accurate.

1.2 Related Works

In sentiment analysis system of review datasets using Naïve Bayes' and K-NN classifier, Lopamudra Dey said that the system performed movie and hotel reviews by using two classifiers: Naïve Bayes' and K-NN classifier. This system is used to evaluate the performance for sentiment classification in terms of accuracy, precision and recall. Finally, the system made decision that Naïve Bayes classifier produced better results than K-NN classifier and gave above 80% accuracy [11].

In the performance of analysis system on mail classification with Multilayer Perceptron (MLP), the e-mails sent from others are classified by using MLP approach, feature selection methods and term frequency and inverse document frequency (TF-IDF) as spam or ham. Finally, the system produced the good accuracy and reduced the complexity [15].

According to Paras Seth, the system classified the SMS messages as spam or ham by using three algorithms: Naive Bayes, Random forest and Logistic Regression in SMS spam detection and comparison system of various machine learning algorithms. Finally, the system made decision that Naïve Bayes is the highest accuracy with 98.445% [17].

1.3 Motivation of the Study

YouTube is popular day by day among the people using the social media. But the people who uploaded videos on YouTube face some problems. These problems are:

- Some comments on videos uploaded are not relevant
- Attackers can distribute virus via the comments.
- The viewers' devices such as mobile phone, tablet, notebook, and laptop are harmful and other software and applications in this devices are destroyed.

Due to these problems, the system is developed to classify the spam and ham comments.

1.4 Overview Spam Comments

Spam comments are unimportant messages in the comment page that appears repeatedly. The spam comments can contain inappropriate messages. YouTube spam can sometimes be used to send unsolicited bulk message to users. YouTube spammer targets comments including the promotion of their videos to distribute virus, to launch malware, to advertise technology, business and others, to promote channels, to introduce links to other websites.

1.5 Contributions of the Study

The system is developed domain textual comments for 5 music movies uploaded on YouTube and feature selection methods and Multinomial Naïve Bayes classifier. The proposed classification system is very useful in YouTube domain area. The contributions of the Study are as follows:

(i) Domain comment on YouTube is collected.

(ii) YouTube comment text classification model is constructed.

(iii)Multinomial Naïve Bayes classifier is used to analyze the text document.

1.6 Organization of the Study

This Study is organized into five chapters, Introduction, Background Theory, Probabilistic Models for Classification, System Design and Implementation and Conclusion.

In chapter one, classification of YouTube comment spam system is introduced. This chapter also describes the motivation, contribution, aim, and objectives of the work.

In chapter two, the background theory of the system is described. How to handle various data types are explained. And then, the variations of data classification are described. The common techniques in data classification, features selection methods such as probabilistic method, handing different data types (large scale data, big data, and big data framework and data streams), text classification, multimedia classification, and time series and sequence classification, are presented in this chapter. Furthermore, various types and models of data classification, and (SVM) are briefly explained.

In chapter three, probabilistic method, Bayes' theorem and Naïve Bayes classifier used for the system are definitely described with formulae.

Chapter four presents system design, implementations with figures, term weighing schemes (TF-IDF method) calculations and Multinomial Naïve Bayes calculation are described.

In chapter five, the conclusion of the research work is described. In this chapter, further extensions propose implementations that could be made are presented. In addition, the limitations and advantages of the system are described in this chapter.

CHAPTER 2
BACKGROUND THEORY

This chapter not only explains the details of theory background about classification of YouTube comment spam but also presents classification problems, data classification techniques, handling different data types, and variations of data classification. Firstly, this chapter describes the classification problems based on domain areas. Secondly, this chapter describes the common techniques used in classification of data. Thirdly, it explains handing different data types: data streams, big data. Finally, it presents various types of data classification and variations briefly.

To solve the problem of data classification, there are numerous applications concerned with mining applications. There are many attempts for learning and solving the relationship, association and practical problems between a set of feature variables and a target variable of domain. The problem for classification may be expressed as follows: Determine the class label for an unlabeled test instance by using a collection of training data points and respective training labels. There are numerous variations of this problem that can be defined in various circumstances. Classification algorithm typically contains two phases:

- Training Phase: In this phase, the model is built from the training instances.
- Testing Phase: In this phase, the model built in the training phase is applied to assign a label to an unlabeled test instance.

In certain cases, such as lazy learning case, the training step is completely skipped, and the classification is done only based on the association between the training and test instances. This scenario is shown by instance-based approaches like nearest neighbor classifiers. In this case, to run efficiency during the testing, a pre-processing phase, the building of a nearest neighbor index, may be constructed. The result of the classification algorithm can be displayed with one of two ways for a test instance:

1. Discrete Label: A label is returned for the test instance in the case.
2. Numerical Score: For each class label and test instance combination, a numerical score is returned. Note that the numerical score for a test instance can be transformed to a discrete label by selecting the class with the greatest score for that test instance. The benefit of a numerical score is allowing to compare and rank the relative tendency of different test instances contained a

given and important class. These methods which frequently utilized in rare class identification issues where the original class distribution is highly split from others and the discovery of some classes is more valuable than the discovery of others.

As a result, the classification problem is based on the class label and it will segmented new test instance. Although data categorization is particularly relevant to concepts of similarity, **l**arge departures from similarity-based segmentation, like in clustering, can be attained in practice. As a result, supervised learning is used a categorization problem, unsupervised learning is used for clustering problem. As the class labels is an important of the supervision process. Some of the most commonly use applications are listed below:

- Customer Target Marketing: The problem of customer target marketing is the popular one for that the classification problem emphasizes to connect feature variables to target classes. In such case, customer-specific feature variables can be utilized to predict their purchasing preferences based on previous training samples. The customer's buying interest could be encoded in the target variable.
- Medical Disease Diagnosis: The application is used with data mining technologies in medical technology. That has become popular in last few years. The users can retrieve features from the health histories, and the class designations also indicate whether a patient is likely to get a sickness in the coming. In these circumstances, it is appropriate to develop a disease prediction utilizing the medical disease diagnosis data.
- Supervised Event Detection: In many temporary circumstances, the user may link class labels to time stamps referring to notable events. For instance, the time-series classification method is very useful in such situations when an invasion occurs can be represented as a class label.
- Multimedia Data Analysis: The classification of huge information from multimedia data, for example music, movies, photographs or other complex multimedia data and the complexity of the feature space and feature values are commonly desired, multimedia data analysis can often be challenge for the programmers and the users [19].
- Biological Data Analysis: Biological data is usual intended as discrete sequences and it is predicted the characteristics of particular sequence.

Biological data is sometimes called in the form of networks. As a result, there are many alternative ways to employ categorization methods in this scenario.

- Document Categorization and Filtering: Many applications need to categorize large amounts of data over time, like newswire services. This application is an important application in the field of and is in its private right.
- Collective classification and other techniques of social network analysis correlate labels underlined with nodes. These are then used for prediction the labels of other nodes. Such applications are extremely useful for predicting useful properties of actors in a social network.

The diversity of problems addressed by classification algorithms is significant, and covers many domains. As a result, this book will divide the classification field into important areas of interest. The work in the data classification field is usually divided into a few major areas;

- Technique-centered: To solve data classification, there are many strategies. They are decision trees, probabilistic methods, rule-based methods, neural networks, SVM methods, probabilistic methods, closest neighbor methods, and so on.
- Data-Type Centered: There are different applications that generate a variety of data types. Some different types of data include unreliable text, time series, and multimedia and network data. Each of these data kinds requires the creation of unique procedures, which can vary greatly.
- Variations on Classification Analysis: There are many variations on the standard categorization problem that address more challenging situations, including rare class learning, supervised learning, transfer learning, semi-supervised learning, unsupervised learning . To improve classification algorithms, ensemble analysis can be used. These problems concern the model evaluation process. These problems concern the model evaluation processes.

2.1 Common Techniques in Data Classification

The many approaches for data classification that are regularly used will be explained in this section. The most frequent techniques used for data classification, and it would be different the solving ways on data of system. The most common methods are used in data classification. They are decision trees, role-based methods,

rule-based methods, probabilistic methods, neural networks, SVM methods, and instance-based methods [12]. This chapter will go over each of these strategies briefly.

2.1.1 Feature Selection Methods

The first step in almost all classification algorithms is feature selection. Individuals who are not domain specialists typically collect a wide variety of features in most data mining scenarios. Clearly, because irrelevant features are not properly related to the class label, they can lead to poor modeling. When the training data set is small and features are applied as a part of the training model, such features will typically worsen the classification accuracy because of over fitting. For example, consideration a medical example in which features from different patients' blood tests are utilized to predict a specific disease. Clearly, a characteristic like cholesterol level predicts heart disease, whereas a feature1 like PSA level does not. However, due of random changes, the PSA level may have freak correlations with heart disease if a short training data set is used. While a single variable may have a minor effect, the combined effect of numerous irrelevant features might be significant. As a result, the training model will generalize poorly to unknown test instances. Using the correct features during the training process is crucial.

There are two kinds of feature selection methods:

1. Filter Models: In these cases, a single feature, or a group of features, is evaluated for classification appropriateness using a crisp criterion. This method is not dependent of the algorithm that is being utilized.
2. Wrapper Models: In these cases, the feature selection process is embedded into a classification algorithm. This algorithm works well with a variety of features.

In order to perform filter models, one of the feature selection methods, a number of different measures are used for quantifying the relevant feature to the classification process. These metrics compute the feature value imbalance over different attribute ranges, which can be discrete or numerical. Sample computing are

• Let $P_{1,\dots},P_k$ be the proportion of classes that correlated to a discrete attribute value of a given value. The gini-index of the discrete attribute value is then given by:

$$G = 1 - \sum_{i=1}^{k} P_i^2 \tag{2.1}$$

The value of *G* series between 0 and *1 − 1/k*. A small value indicates a class inequality. This indicates that the feature value is more unequal for classification. A weighted average over the discrete attribute's various values or the maximum gini-index over any of the several discrete values can be used to get the attribute's overall gini-index. Although the weighted average is the most common, different tactics could be preferred in various situations.

• Entropy: The entropy of a discrete attribute value can be calculated as follows:

$$E = -\sum_{i=1}^{k} P_i \log(P_i) \tag{2.2}$$

The same symbols as in the gini-index example above are used. An entropy value with a small difference between 0 and *log (k) indicates* class skew.

- Fisher's Index: The Fisher's index is a comparison between class and class distribution. Therefore, if p_j is a percentage of training instances related to class *j*, u_j is the meaning of a separate feature for *class j*, μ is the feature, and σ_j is the feature for class *j*, then the Fisher score F can be computed as follows:

$$F = \frac{\sum_{j=1}^{k} p_j(\mu_j - \mu)^2}{\sum_{j=1}^{k} p_j \sigma_j^2} \tag{2.3}$$

In order to assess the discriminative power of attributes, a range of different measures such as the χ2-statistic and mutual information are available. In order to integrate the many features into directions in the data that are extremely relevant to classification, a method known as Fisher's discriminant is also used. Such methods are of course feature transformation methods, which are connected to feature selection methods, just as unsupervised dimensionality reduction methods are related to unsupervised feature selection methods.

More broadly, multiple features are frequently highly connected with one another, and the added value of an attribute once a given set of features has already been selected differs from its standalone utility. The Minimum Redundancy Maximum Relevance strategy was presented to address this problem, in which features are incrementally picked based on their incremental gain when added to the feature collection. Because the evaluation is on a subset of features and a crisp criterion is used to evaluate the subset, this method is also a filter model.

In wrapper models, feature selection phase is included in an iterative approach with a classification algorithm. The classification algorithm assesses a certain collection of features in each iteration. This set of features is then enhanced using a

specific technique, and evaluated to upgrade the quality of the classification. Because the classification algorithm is used for evaluation, this approach will usually provide a feature set, which is sensitive to the classification algorithm. Because of the large variety of data classification models, this strategy has been found to be useful in practice. For example, in an SVM, the two classes are separated out by using a linear model, in a nearest neighbor classifier would prefer features, different classes are clustered into spherical regions.

2.1.2 Probabilistic Methods

Probabilistic methods are the most basic methods among all data classification methods. Probabilistic classification algorithms use the statistical inference to choose the best and most suitable class. These algorithms will produce a interacting posterior probability of the test instance being a member of each of the available classes, in addition to assign the best class like other classification algorithms. The posterior probability is defined as the probability after observing the specific characteristics of the test instance. The prior probability, on the other hand, is just the proportion of training records that correspond to each specific class, without knowledge of the test case. The posterior probability can then be utilized to identify class membership for each new instance using decision theory.

2.2 Handing Different Data Types

Different data types are used and are solved by using different techniques for data classification. This is different because data type chosen for the system often qualifies the kind of problem and the problem solved by the classification approach. In this section, we will discuss the various data types studied in classification problems and these data types may be required by an accurate level of special handling.

2.2.1 Large Scale Data: Big Data and Data Streams

The collecting different large scale of data have become a challenge in the classification process with the increasing ability. The larger data set make clearly the more accurate and mare effective sophisticated models. However, if one is limited computationally by scale issues, this is not always helpful. The issues of data streams

and big data analysis are distinct. In the first situation, real-time processing raises difficulties, but in the second, the problem arises from the inefficiency of computation and data access over exceptionally huge amounts of data. Summary statistics from huge volumes are sometimes difficult to compute since access must be spread and it is too expensive to execute amount of data. This section will go over each of these difficulties.

2.2.2 Data Streams

The ability to continuously collect and handle enormous amounts of data has led to the popularity of data streams. Two major issues occur in the building of training models in the streaming scenario.

- One-pass Constraint: Due to the huge number of data streams, all processing algorithms must complete their calculations in a single pass over the data. This is a serious difficulty because many iterative algorithms that perform reliably over static data sets are not applicable. As a result, it is critical to create effective training models.
- Concept Drift: A generating process, which may change over time, is generally used to create data streams. Concept drift occurs as a result of this, and it correlates to changes in the underlying stream patterns over time. Because models get stale over time, the presence of idea drift can be damaging to classification algorithm. As a result, it's critical to change the model gradually such that it obtains high accuracy over current test cases.
- Massive Domain Constraint: Various stream contains discrete attribute .The streaming scenario discrete attributes take on millions of potential values. This is because streaming items is related to discrete identifier. Email addresses in an email, IP addresses in a network packet stream, and URLs in a click stream collected from proxy Web logs are just a few examples. In streaming applications, the enormous domain problem is common. In reality, this issue has been taken into consideration when designing a number of synoptic data structures, like the count-min sketch and the Flajolet-Martin data structure. While this topic hasn't received much attention in the stream mining literature, new research has made some progress in this regard.

In order to handle the aforementioned issues, traditional classification algorithms must be appropriately adjusted. Special cases, such as those in which the

stream data domain is huge or the classes are uncommon, present unique issues. For easily updatable models, most well-known streaming classification techniques use space-efficient data structures. Furthermore, by making the models temporally flexible or by applying various models over different portions of the data stream, these methods are specifically designed to tackle concept drift. In the streaming situation, special scenarios or data categories necessitate particular methods. For example, in the massive-domain case, the count-min data structure might be used as a summary structure within the training model. Exceptional class learning, in which rare class instances may be combined with occurrences of totally new classes, is a particularly difficult situation. This problem falls in between classification and outlier prediction. Nonetheless, in applications such as intrusion detection, this is the most typical scenario in the streaming domain. Nonetheless, in applications like intrusion detection, it is the most typical scenario in the streaming domain.

2.2.3 Big Data and Framework

While streaming algorithms assume that the data is too massive to be kept explicitly, the big data framework develops the storage technology to actually store and process the data. Even if data can be explicitly saved, processing and extracting insights from it is difficult. Saving the data on disk is the most basic example, and scaling up the technique with disk-efficient algorithms is desired. Nearest neighbor and associative classifiers may be rated by using more efficient subroutines, and decision trees and SVMs require dedicated scaling methods.

SLIQ, BOAT, and RainForest are examples of scalable decision tree methods. The SPRINT method was one of the first parallel implementations of decision trees. Scalable decision tree methods are usually implemented in one of two ways. By maintaining attribute-wise summaries of the training data, methods like RainForest boost scalability. These summaries are sufficient for efficiently splitting single attributes. Methods like BOAT use bootstrapped samples to produce a decision tree that is extremely near to the accuracy that would be produced if the entire data set was used.

2.3 Text Classification

Text data is one of the most commonly used data formats when it comes to classification. Text data is everywhere, thanks to its prevalence on the Web and in social networks. While a text document can be regarded as a string of words, it is most typically used as a bag-of-words, which does not apply the ordering information between words. This text form is much more like multidimensional data. Standard multidimensional classification methods, on the other hand, frequently need to be updated for text [6].

The data is dimensional and sparse, which makes text classification difficult. A normal text lexicon can be hundreds of thousands of words long, yet a document typically contains. As a result, the majority of attribute values are zero, and the frequencies are low. Many frequent terms are likely to be noisy and non-discriminative in classification. As a result, feature selection and representation problems are especially important in text categorization.

For text data, not all classification methods are equally popular. Rule-based methods, the Bayes method, and SVM classifiers, for example, are more popular than others. Some rule-based classifiers, such as RIPPER, were originally designed for text classification. In some cases, neural methods and instance-based methods are used. Rocchio's method is a common instance-based text classification method. Instance-based methods are also commonly employed with centroid-based classification, where the original documents are replaced by frequency-truncated centroids of class-specific clusters. Because the centroid of a small closely related set of documents is often a more stable representation of that data localization than any single document, this generally delivers higher accuracy. This is true given because of the sparse nature of text data, in which two related documents may often have only a small number of words in common.

Many classifiers, such as decision trees, are popular in other data domains but aren't as popular with text data. Because decision trees use a strong hierarchical segmentation of the data, this is the situation. As a result, the features at the higher levels of the tree are implicitly given greater importance than other features. A single word in a text collection with hundreds of thousands of characteristics (words) usually reveals relatively little about the class label. A decision tree will also often partition the data space with a limited number of splits. When this value is orders of magnitude

less than the underlying data dimensionality, this is a problem. Of course, decision trees in text are unbalanced as well, because a given word is only found in a small percentage of the texts. Consider the instance where the presence or absence of a word corresponds to a split. Because of the tree's imbalance, most paths from the root to the leaves will correspond to word-absence judgments, with only a few (less than 5 to 10) to word-presence decisions. Clearly, this will result in incorrect classification, particularly in situations where word absence does not convey much information and just a small number of word presence judgments are necessary. Due to the disproportionate relevance of some features and the resulting inability to efficiently exploit all available features, univariate decision trees do not operate well for very high dimensional data sets. Multivariate splits can be used to improve the effectiveness of decision trees for text classification, but this can be quite costly.

The regular classification methods for the text domain must also be appropriately changed. This is due to the text domain's high dimensionality and sparseness. The multinomial Bayes model, for example, is a specific model for text that differs from the standard Bernoulli model. The Bernoulli model symmetrically treats the presence and absence of a word in a text document. However, only a small percentage of the word size is present in any particular text document. A word's absence is frequently significantly less informative than a word's presence. In the text domain, the symmetric treatment of word presence and absence might be detrimental to the performance of a Bayes classifier. The multinomial Bayes model is applied to attain this goal, which utilizes the frequency of word presence in a document but ignores non-occurrence.

Scalability is significant in the context of SVM classifiers because they scale poorly with the number of training documents and data dimensionality (lexicon size). Additionally, text sparsity (i.e., a small number of non-zero feature values) should be used to improve training efficiency. This is because an SVM classifier's training model is built using a limited quadratic optimization problem with the same number of constraints as the number of data points. This is quite huge, and it directly causes the related Lagrangian relaxation to grow in size. The space requirements for the kernel matrix in kernel SVM could also increase quadratically with the number of data points.

2.3.1 Multimedia Classification

Multimedia data has become increasingly common as social media sites have grown in popularity. Users can publish their photographs or movies to sites such as Flickr or YouTube. In such situations, classification of portions or all of a photograph or video is desirable. Rich meta-data may be accessible in some situations, providing for more successful data classification. The problem of data representation is especially important for multimedia data since weak representations have a huge semantic gap, which makes classification problematic. The combination of text with multimedia data in order to establish more effective classification models has been discussed in Many methods, such as semi-supervised learning and transfer learning, can be utilized to increase the data classification process' performance. In terms of data representation and information fusion, multimedia data presents distinct issues.

2.3.2 Time Series and Sequence Data Classification

Both of these data types are temporal data types with two different types of properties. The first is the contextual attribute (time), and the second is the behavioral attribute, which correlates to the time series value. The major distinction between time series and sequence data is that the former is continuous while the latter is discrete. This distinction is important, though, because it transforms the character of the usually utilized models in the two situations [23].

Time series data is widely used in a variety of applications, including sensor networks and medical informatics, where huge volumes of streaming time series data are required for classification [20]. With time-series data, two types of classification are possible:

- Classifying specific time-instants: These relate to specific events inferred at certain points in the data stream. In some circumstances, the labels are linked to specific moments in time, and the behavior of one or more time series is used to categorize these instances. In this context, for example, detecting major events in real-time apps can be a crucial application. Classifying portions or the entire series: Class labels are connected with portions or the entire series in these situations, and these are utilized for classification. An ECG time-series, for example, will show distinct forms for various diagnostic criteria for diseases.

Both of these situations are equally relevant in terms of drawing analytical conclusions in a range of scenarios. These situations are also applicable in the case of sequencing data. In biological, Web log mining, and system analysis applications, sequence data is regularly encountered. Because of the discrete character of the underlying data, methods that are not relevant to continuous time series data must be used. For example, the nature of distance functions and modeling methodologies in discrete sequences varies significantly from those in time series data.

2.4 Variations on Data Classification

The variations problem of data classification changes the conventional classification problem or classification enhancements with the addition of more data. Rare-class learning and distance function learning are two important versions of the classification problem. More data in methods, meta-algorithms, such as human intervention in visual learning, active learning, transfer learning and co-training, can be used to improve the data classification problem. Furthermore, evaluation model is important in the context of data classification. Because of the importance of evaluation model, there are effectiveness of classification meta-algorithms.

2.4.1 Rare Class Learning

Rare class learning is a variant of the classification problem that is strongly linked to outlier analysis. It's actually a supervised version of the outlier detection problem. In uncommon class learning, the distribution of classes in the data is significantly imbalanced, and accurately determining the positive class is often more important. Consider the situation where it is desirable to categorize patients into malignant and non-malignant groups. The majority of patients may be normal in such instances, but misclassifying a truly cancerous patient is usually significantly more costly (false negative). As a result, false negatives cost more than false positives. Because of the misclassification of different classes, the problem is intimately related to cost-sensitive learning [12].

The main distinction from the ordinary classification problem is that the problem's objective function must be changed with costs. This introduces a number of possibilities for efficiently resolving the problem:

- Example Weighting: In this situation, the weighting of the examples is determined by the cost of misclassification. This leads to minor changes the objective function in an SVM classifier, for example, must be correctly weighted with costs, whereas the quantification of the split criterion in a decision tree must weigh the data with costs. The k nearest neighbors are correctly weighted while determining the class with the largest presence in a nearest neighbor classifier.
- Example Re-sampling: The examples (rare classes are over-sampled and the normal classes are under-sampled) are appropriately re-sampled. The re-sampled data is subjected to a conventional classifier with no modifications. This approach is similar to example weighting from a technical standpoint. However, from a computational standpoint, such an approach has the advantage of resulting in significantly smaller re-sampled data. This is because the vast majority of the examples in the data belong to the normal class, which is severely under-sampled, whereas the rare class is often just slightly over-sampled.

There are numerous versions of the rare class detection problem, whether or not single-class (accessible) and the regular class (contaminating with rare class).

2.4.2 Distance Function Learning

Data classification is directly related to distance function learning, which is an essential problem. It is desirable to associate pairs of data instances to a distance value using either supervised or unsupervised approaches in this problem. Consider the case of an image collection in which the similarity is determined using a user-centered semantic criterion. Standard distance functions, such as the Euclidian metric, may not accurately express the semantic similarities between two images in this situation, as they are based on human perception and may even differ from collection to collection. As a result, the best way to deal with this problem is to explicitly include human feedback in the learning process [2].

This feedback is usually expressed as pairs of images with specific distance values, or as rankings of distinct images relative to a given target image. This method can be applied to a wide range of data domains. This is the training data that is used for learning purposes.

2.4.3 Ensemble Learning for Data Classification

A meta-algorithm, a classification method, reuses one or more currently existing classification algorithms by mixing the same algorithms' result with different parts of the data, or by using several models for robustness. The overall purpose of the algorithm is to produce more accurate results by incorporating the outcomes of various training models, independently or sequentially. The total error of a classification model is determined by the bias and variance of the data, as well as the intrinsic noise in the data. No corresponding the decision boundary of a specific model with the actual decision border affects a classifier's bias [24].

For instance, without having a linear decision boundary, an SVM classifier will assume that the training data may have a linear decision boundary. The variance is determined by the random fluctuations found in the training data set. Variance will be higher in smaller training data sets. Various types of ensemble analysis are used to reduce bias and variance. The reader is referred to for an excellent discussion on bias and variance. In order to acquire more accurate results from various data mining problems, meta-algorithms are often utilized in clustering and outlier analysis data mining problems.

Because of its specific evaluation criteria and easy operation of incorporating the results of several algorithms, classification is the richest domain in terms of meta-algorithms. The meta-algorithms' popular examples are:

- Boosting: Boosting is a common classification technique. The goal is to focus on more challenging regions of the data set to develop models that can more properly categorize the data points in these sections, and then use the ensemble scores across all components. To determine the wrongly classified instances for each component of the data set, a hold-out strategy is applied. The goal is to find superior classifiers for more challenging sections of the data one at a time, then combine the results to create a meta-classifier that works well on all parts of the data [8].
- Bagging: Bagging is a technique that uses random data samples and combines the results of models built with different samples. Each classifier's training examples are chosen by sampling with replacement. These are referred to as bootstrap samples. In many cases, this strategy has been proved to produce better results, however this is not always the case. Because of the special

random features of the training data, this technique is not useful for decreasing bias, although it can reduce variation.

- Random Forests: Random forests are a method that uses sets of decision trees on randomly generated vectors or random selections of the training data to compute the score as a function of these varied components. The random vectors are typically created using a predefined probability distribution. As a result, either random split selection or random input selection can be used to construct random forests. In terms of how the sample is chosen, random forests are closely connected to bagging, and bagging with decision trees can be considered a specific case of random forests (bootstrapping). It is also feasible to generate trees in a lazy approach in the case of random forests, which is suited to the specific test instance at hand [1].
- Model Averaging and Combination: one of the most common ensemble analysis models is model averaging and combination and the random forest method mentioned earlier is a subset of this concept. Many Bayesian methods exist for the model combining process in the context of the classification problem. The usage of multiple models ensures that the inaccuracy caused by a classifier's bias does not dominate the classification results.
- Stacking: Stacking methods combine various models with various ways, including utilizing a second-level classifier. A new feature representation for the second level classifier is created using the output of several first-level classifiers. Alternative bagged classifiers or different training models can be used to select these first level classifiers. To avoid over fitting, the training data for the first and second level classifiers must be separated into two subsets [14].
- Bucket of Models: In this approach, data set’s "hold-out" portion is used to determine the best suitable model that achieves the highest accuracy. This strategy can be considered as a competition or bake-off between the various models.

The field of classification meta-algorithms is enormous, and different variations may perform better in different situations.

CHAPTER 3
PROBABILISTIC MODELS FOR CLASSIFICATION

Classification, a supervised method in machine learning, infers a function from class label in training data. The training data is made up of a set of sample data associated domain, each contains a vector $x = x_1, x_2, \ldots, x_d$ and a target label $y \in \{C_1, C_2, \ldots, C_k\}$ [6]. A classification algorithm's objective, given a collection of training data, is to examine the data and produce an inferred function for classifying new (unknown) samples by assigning a corresponding and most suitable class label to each of them.

Probabilistic classification is a common class of classification, and some probabilistic classification methods will be included in this chapter. Statistical inference is used by probabilistic classification algorithms to determine the best class for a given example. Probabilistic classification algorithms will return a corresponding probability of relevant classes, like other classification methods, providing the best class. The best class is usually regarded as the highest probability class. Generally, probabilistic classification algorithms has a few improvement over non-probabilistic classifiers. Firstly, it can output a confidence value (like likelihood) related with the class label chosen, and so abstain if the output's confidence is too low. Secondly, probabilistic classifiers can be effectively integrated to be larger machine learning tasks, avoiding the problem of error propagation in part or whole.

The fundamental principle of probabilistic classification in a probabilistic framework is to estimate the posterior class probability (C_k|x). After computing this probabilities, decision theory can be used to determine class labels for every new input x. There are typically two methods for estimating these probability [5].

It focuses on calculating the class-conditional probabilities for each class C_k separately, $p\ (x|C_k)$. as well as inferring the prior class $p\ (C_k)$ separately. The posterior class probabilities $p\ (C_k|x)$ can then be calculated using Bayes' theorem. Alternatively, the joint distribution $p(x,C_k)$ can be explicitly modelled and then normalize the posterior probabilities. The generative models: explicitly or implicitly models, distribute inputs and outputs, because the class-conditional probabilities determine the features generated by the statistical process. In generative model, fitting the parameters is a common method, the measured data is actually a sample from this model. Two common

examples of probabilistic generative models in classification approach are the Naive Bayes classifier and the Hidden Markov model.

By learning a discriminative function $f(x) = p(C_k|x)$, where input x directly assigns a class label C_k, the posterior probabilities $p(C_k|x)$ are directly modelled in another class of models. The focus of this strategy, known as the discriminative model, is on establishing the overall discriminative function while taking into account class-conditional probabilities.

In probabilistic classification, various basic models and algorithms are described.

- Naive Bayes Classifier: Using the Bayes theorem and naive independence assumptions, it is a straightforward probabilistic classifier. Document classification is an excellent application of the Naive Bayes classifier.
- Logistic Regression: Based on observable variables, it is a method for forecasting a categorical dependent variable's outcome. A logistic function is used to represent the probability describing the various outcomes as a function of the observed variables [7].
- Hidden Markov Model (HMM): is a simple dynamic case. It produces the hidden states (a chain) in a Bayesian network, and only some possible values from each state. One of HMM's goals is to infer hidden cases from inspected values and associated dependent relationships. A very important application of part-of-speech tagging in the field of NLP is important in using HMM [3].
- Conditional Random Fields: A Conditional Random Field (CRF) is a type of Markov random field in which each node's state is dependent on some observed values. Because they do not model the distribution of observations, CRFs can be called discriminative classifiers. One of CRF's applications is name entity recognition in information extraction [16].

3.1 Naïve Bayes Classification

The Naive Bayes classifier derived from the Bayes theorem and is especially useful when the inputs have a high dimensionality. Although it is simple, it outperforms more advanced classification methods like decision tree and chosen neural network classifier. When applied to massive datasets, this classifiers demonstrates high

accuracy and speed. In this section, Bayes' theorem is briefly reviewed, which used use in machine learning techniques, and particularly document classification are described.

3.1.1 Bayes' Theorem and Preliminary

A basic probability theorem known as Bayes' theorem or Bayes' rule provides a widely used framework for classification [4]. Firstly, two fundamental rules of probability theory are described:

$$P(X) = \sum_{Y} P(X, Y) \tag{3.1}$$

$$P(X, Y) = P(Y|X)P(X) \tag{3.2}$$

Where the equation 3.1 is the sum rule and the equation 3.2 product rule. Here, a joint probability is $p(X, Y)$, a conditional probability is the quantity $p(Y|X)$, and a marginal probability is the quantity $p(X)$. All probability theory is built on these two fundamental rules. The following equation 3.3 is Bayes' theorem and it can be easily calculated by using the product rule and the symmetry property $p(X, Y) = p(Y, X)$,

$$P(Y|X) = \frac{P(X|Y)}{P(X)} \tag{3.3}$$

This is important in machine learning, particularly classification. The denominator in Bayes' theorem can be represented in terms of the quantities in the numerator using the sum rule.

The normalization constant is required to ensure that the sum of the conditional probability on the left-hand side of Equation (3.3) over all values of Y equals one is the denominator in Bayes' theorem.

To understand the probability theory's and the Bayes' theorem's fundamental concepts, consider a simple example. Two boxes, one red and one white, each containing two apples, four lemons, and six oranges [10]. The white box also has three apples, six lemons, and one orange. Let's assume one of the boxes is picked at random and then one item at random from that box, observing what kind of item it is. The item in the box arrived during the procedure is replaced, several times. When an item from a box is chosen, any of the things in the box are chosen, with the red box being chosen

40% of the time and the white box being chosen 60% of the time. Let's start by defining a random variable. *Y* to represent the box chose,

$$P(Y = r) = {}^{4}/_{10}$$

, and

$$P(Y = w) = {}^{6}/_{10}$$

Where *p(Y =r)* indicates the peripheral probability of choosing the red box and *p(Y =w)* choosing the white box. A box at random is picked, the likelihood of choosing an item is the fraction of which item selected in the box, and this can be stated as the following conditional probabilities

$$P(X = a|y = r) = {}^{2}/_{12} \tag{3.4}$$

$$P(X = l|Y = r) = {}^{4}/_{12} \tag{3.5}$$

$$P(X = o|Y = r) = {}^{6}/_{12} \tag{3.6}$$

$$P(X = a|Y = w) = {}^{3}/_{10} \tag{3.7}$$

$$P(X = l|Y = w) = {}^{6}/_{10} \tag{3.8}$$

$$P(X = o|Y = w) = {}^{1}/_{10} \tag{3.9}$$

These probabilities have been standardized so that,

$$P(X = a|Y = r) + P(X = l|Y = r) + P(X = o|Y = r) = 1$$

And

$$P(X = a|Y = w) + P(X = l|Y = w) + P(X = o|Y = w) = 1$$

The next instance, when an item has been chosen, it is an orange. This determines the probability distribution over boxes conditions to choose the item's identity. The probabilities from equation (3.4) to equation (3.9) showcase how the item is distributed on the identification of the box. By applying Bayes' theorem to reverse the conditional probability, we may determine the posterior probability.

$$P(Y = r|X = o) = \frac{P(X = o|Y = r)p(Y = r)}{P(X = o)} = \frac{{}^{6}/_{12} \times {}^{4}/_{10}}{{}^{13}/_{50}} = \frac{10}{13}$$

Using the sum and product rules, It is possible to determine the comprehensive possibility of selecting an orange*(X = o)*.

$$P = (X = o) = P(X = o|Y = r)P(Y = r) + P(X = o|Y = w)P(Y = w)$$

$$= \frac{6}{12} \times \frac{4}{10} + \frac{1}{10} \times \frac{6}{10} = \frac{13}{50}$$

The sum rule follows that $p(Y = w|X = o) = 1-10/13= 3/13$. In most situations, the classes' probabilities for the data samples are interested.

Suppose that random variable *Y is used* to designate the class label, and *X* the feature of data samples. $p(Y = C_k)$ can be explained as the prior probability for the class *Ck*, This indicates the likelihood that a data sample's class label will be true is C_k before we observe the data sample. The relevant posterior probability $p(Y|X)$ can be calculated using Bayes' theorem after observing the feature *X* of a data sample. The quantity $p(X|Y)$ can expressed the observed data and *X* is for different classes, *likelihood*.

Although its integral with respect to *Y* is often one, likelihood is not a probability distribution over *Y* [22]. It is possible to express Bayes' theorem as posterior $\propto$ *likelihood* × *prior*.

3.1.2 Naïve Bayes Classifier

Despite its strong independence assumption, the simplest Naive Bayes classifier has become an important probabilistic model and improves effectiveness. Among other applications, Text classification, medical diagnosis, and performance management have all exhibited the utility of naive Bayes. The Naive Bayes classifier model, maximum likelihood estimates, and applications are described in the following subsections [7].

A classifier that productions the posterior probability $p(Y|X)$ is tested for various *Y* values. According to Bayes' theorem, $p(Y = Ck\ |X = x)$ can be represented as

$$P(Y = C_k|X = x) = \frac{P(X = x|Y = C_k)}{P(X = x)}$$

$$= \frac{P(X_1 = x_1, X_2 = x_2, \dots, X_d = x_d|Y = C_k)P(Y = C_k)}{P(X_1 = x_1, X_2 = x_2, \dots, x_d = x_d)}$$

(3.10)

To *calculate p(Y|X), p(X|Y)* and *p*(*Y*) are evaluated by using training data. Then *p*(*Y*|*X* = *x*(*i*)) can be estimated any modern example, *x (i).*

Exact Bayesian classifiers are typically difficult to learn. If *Y* is Boolean and *X* is a vector of *d* Boolean features, 2^d parameters $p(X_1 = x_1, X_2 = x_2, \ldots, X_d = x_d \mid Y = C_k)$ is needed to estimated.[18] The reason for this is that there are 2^d possible values of *x* for each given value C_k, requiring the computation of $2^d - 1$ independent parameters. A total of $2(2^d - 1)$ such parameters given two possible values for *Y* are estimated. Each of these distinct instances with multiple times are observed to create better estimates of each of these parameters unachieved in practical classification domains. If *X* is a vector with 20 Boolean features.

The Naive Bayes classifier reduces the volume by assuming that the features $X_1, \ldots, X_d$ are all conditionally independent of one another, given *Y*. The number of parameters to be estimated for modeling *p(X|Y)* drops from the original $2(2^d - 1)$ to merely 2^d in the preceding example due to the conditional independence assumption. Think about the chance of Equation $p(X = x \mid Y = C_k)$ (3.10),

$$P(X_1 = x_1, X_2 = x_2, X_d = x_d | Y = C_k)$$

$$= \prod_{j=1}^{d} P\left(X_j = x_j \middle| X_1 = x_1, X_{2=} x_2, \ldots, X_{j-1} = x_{j-1}, Y = C_k\right)$$

$$= \prod_{j=1}^{d} P\left(X_j = x_j \middle| Y = C_k\right)$$

(3.11)

In the chain rule, the second line is a generically property of probabilities, and the third way is the rate for the random variable X_j distinct of all other feature values, $X_{j\_}$ for $j\_ = j$ concerning the source of the label *Y*., 2*d* parameters to declare $p(Xj|Y = Ck)$ are needed when *Y* and *Xj* are boolean variables.

Naive Bayes classifier is obtained by deriving equation (3.11) from equation (3.10).

$$P(Y = C_k | X_1, \ldots, X_d) = \frac{P(Y = C_k) \prod_j P\left(X_j \middle| Y = C_k\right)}{\sum_i P(Y = y_i) \prod_j P\left(X_j \middle| Y = Y_i\right)}$$

(3.12)

The value of *Y* can use the Naive Bayes classification rule:

$$Y \leftarrow arg \min_{C_k} \frac{P(Y = C_k) \prod_j P(X_j | Y = C_k)}{\sum_i p(Y = y_i) \prod_i P(X_j | Y = y_i)}$$

(3.13)

$$Y \leftarrow arg \min_{C_k} P(Y = C_k) \prod \prod_i P(X_j | Y = C_k)$$

(3.14)

Because the denominator does not depend on C_k, the above formulation can be simplified to the following

3.1.3 Maximum-Likelihood Estimates for Naïve Bayes Models

The method of maximum likelihood estimates is used to estimate parameters for Naive Bayes models. [13] Estimate of two different types of parameters is necessary for the Naive Bayes model. The very first is

$$\pi k = P(Y = C_k)$$

for any possible values C_k of Y. The constraints $\pi k \geq 0$ and $\Sigma^K_{k=1} \pi k = 1$, and one can interpret the parameter. as the likelihood of encountering the label C_k. Note there are K of these parameters, $(K-1)$ of which are independent.

Assume that d input features X_i can join J distinct values with $X_i = x_i j$. The below is

$$\theta ijk = p(X_i = x_i | Y = C_k)$$

X_i is each input feature, each of the possible values C_k of Y. The probability of feature X_i value $xi\ j$, trained on the underlying label being C_k, is represented by the value for $\theta i\ jk$. Each pair of i, k values must content $\Sigma j\ \theta i\ jk = 1$, and dJK is parameters, and $d(J-1)K$ are independent. Maximum likelihood estimates based on calculating the relative frequencies of the distinct events in the data can be used to estimate these parameters. Maximum likelihood estimates for $\theta i\ jk$ given a set of training examples are [9]

Where *count*(x) return the training set satisfied property x, e.g.,

$$count(X_i = j \wedge Y = C_k) = \sum_n^N \{X^{(n)} i = x_j j \wedge Y^{(n)} = C_k\},$$

and

$$count(Y = C_k) = \sum_n^N \{Y^{(n)} = C_k\}$$

$$\theta\hat{}\_ijk = p\hat{}(X\_i = x\_ij \mid Y = C\_k) = count(X_i = x_{i\wedge Y} = C\_k)/count(Y = C\_k)$$

(3.15)

The number of times label C_k perceived in connection with X_i taking value *xi j*, as well as the overall number of times label C_k seen are counted and these two terms' ratios are computed. The numerator condition is matched to prevent the scenario when there are no training examples in the data**,** it is typical to use an easy estimate that

$$\hat{\theta}_{ijk} = \hat{p}\left(X_i = x_{ij} \middle| Y = C_k\right)$$

(3.16)

$$= \frac{count\left(X_i = x_{ij} \wedge Y = C_k\right) + l}{count(Y = C_k) + lj}$$

Where *J* is the number of distinct values that X_i can take on, and *l* determines the strength of this smoothing. If *l* is set to 1, this approach is called Laplace smoothing.

$$\hat{\pi}_k = \hat{p}(Y = C_k) = \frac{count(Y = C_k)}{N}$$

(3.17)

Where *J* is the number of various values that X_i can cambat, and *l* is the smoothing strength. This method is known as Laplace smoothing when *l* is set to 1. The following are the maximum likelihood estimates for *πk*:

Where $N = \Sigma^{K}_{k=1}\, count(Y = C_k)$ is illustrations in the practice set. In the same way, a smoothed estimate can be obtained using the format below

$$\hat{\theta}_k = \hat{p}(Y = C_k) = \frac{count(Y = C_k) + l}{N + lK}$$

(3.18)

Where *K* is the number of possible values for *Y*, and The intensity of the prior hypotheses in relation to the data observed is measured by *l*.

3.2 Probabilistic and Naïve Bayes Classifier

In the Naive Bayes model, two distinct argument types must be estimated. The assumption in this variety model is that each class contributes to the variety [3].Each component of the mixture is effectively a generative model. that calculates the probability of sampling a specific term for that component or class. It uses a

probabilistic model with independent assumptions about the distributions of different words to model the distribution of documents in each class. For naive Bayes classification, two types of models are often used. Both models essentially compute the posterior probability of a class based on the distribution of words in the document. These models work with the "bag of words" assumption, ignoring the actual position of the words in the document. The assumption in terms of taking (or not taking) word frequencies into consideration, as well as the corresponding approach for samples, are the key differences between these two models:

- **Multivariate Bernoulli Model:** The presence or absence of words in a text document is used as a feature in this model as a document. As a result, word frequencies are not utilized in document modeling, and word features in text are assumed as binary, with two values denoting the present or not present (absent) word in text. Binary features must be modeled. Each class's model is a multivariate Bernoulli model.
- **Multinomial Model:** A document with a bag of words in this model is represented to capture the frequency of words in a document. Each class's documents represented a multinomial word distribution as examples. As a reaction, the conditional probability of a document in a given class is elementary the creation of the probabilities of each regard word in that class.

The component class models can be used in conjunction with the Bayes rule and the posterior probability can be computed and the class with the highest posterior probability can be allocated to the document, regardless of how documents in each class are modelled. The following will describe these two models in more detail.

3.2 1 Bernoulli Multivariate Model

A document is treated as a collection of distinct words with no frequency information, where a term can be present or absent [5]. Assume that by $V = \{t_1 \ldots t_n\}$ denotes the lexicon from which the terms are derived. The text or bag of words in question contains the terms $Q = \{ti1 \ldots tim\}$ and the class is chosen from $\{1 \ldots k\}$. Then, assuming the document (which is supposed to have been created from one of the classes' term distributions) contains the terms $Q = \{ti1 \ldots tim\}$ the intention is to estimate the posterior probability that it sets to class i. Understanding the Bayes method as a productive process from the basic mixture model of classes is the best approach to

understand it. The Bayes probability of class *i* can be modeled by sampling a set of terms *T* from the term distribution of the classes:

The class of the sampled set *T* by C^T and the corresponding posterior probability by $P\ (C^T = i|T = Q)$ are denoted for this system. The important fact is no allowing replacement, picking a subset of terms from *V* with no frequencies attached to the picked terms so the set *Q* may not to contain duplicate elements. It is also important to note that this model has no restriction on the number of terms picked. The concept of Naive Bayes with independence between terms is essentially equivalent to selecting or not selecting each term with a probability based on term distribution. These concepts are the key differences with the multinomial Bayes model. The Bayes approach classifies a given set *Q (sample* data distribution of class *i)* based on the posterior probability, i.e., $P\ (C^T = i|T = Q)$.

More mathematical description of Bayes modeling is $P\ (C^T = i|Q)$. This conditional probability, the Bayes rule can be used to *estimate* more easily from the underlying corpus. To put it another way, it can be summarized as follows:

$$P(C^T = i|T = Q) = \frac{P\left(C^T = i\middle|p(T = Q|C^T = i)\right)}{P(T = Q)}$$

$$= \frac{P(C^T = i)\prod_{t_j \in Q} P\left(t_j \in T\middle|C^T = i\right)\prod_{t_j \notin Q}\left(1 - P\left(t_j \in T\middle|C^T = i\right)\right)}{P(T = Q)}$$

The naive independence assumption is used in the last condition of the above sequence because the probability of occurrence of the different terms are independent of one another. This is required to convert the probability equations into a form that can be approximated from the underlying data.

The class with the highest posterior probability is the one allocated to *Q*. This decision is not affected by the denominator, which is the marginal probability of observing *Q*. That is, the following class will be assigned to *Q*:

$$\hat{\imath} = arg\min_i P(C^T = i|T = Q)$$

$$= arg\min_i P(C^T = i)$$

$$\prod_{t_j \in Q} P\left(t_j \in T\middle|C^T = i\right)\prod_{t_j \notin Q}\left(1 - P\left(t_j \in T\middle|V^T = i\right)\right)$$

It's important to remember that the training corpus can be used to estimate all terms in the above equation. The value of $P(C^T = i)$ is calculated as the percentage of

documents in class *i* globally, or the total number of documents. is represented by P (t j ∈ T|CT = i) in the *i*th class that contain The aforementioned are all maximum likelihood estimates for term t jof the corresponding probabilities. Laplacian smoothing is applied to prevent the occurrence of words with 0 probability by adding small values to the frequencies of terms. In Bayes classifier, the class with the highest probability value is identified rather than the actual probability value associated with it, which is why we do not need to compute the normalizer $P\ (T = Q)$. In the case of binary classes, the Bayes algorithm are used and a number of terms that do not impact are eliminated in order to simplify class probabilities with the computation of these Bayes "probability" values.

Some applications necessitate the exact computation of the posterior probability $P\ (C^T = i|T = Q)$. One way to achieve this is simply to take a sum over all the classes:

$$P(T = Q) = \sum_i P(T = Q|C^T = i)P(C^T = i)$$

This is predicated on each class's conditional independence of features. Data sparsity may be encountered since the parameter values are predicted separately for each class. Another way to compute it, which may help with the data sparseness problem, is to include the assumption of (global) term independence and compute it as:

$$P(T = Q) = \prod_{j \in Q} P\big(t_j \in T\big) \prod_{t_j \notin Q} \Big(1 - P\big(t_j \in T\big)\Big)$$

The term probabilities in all classes are based on global term distributions. A natural question is whether it is possible to create a Bayes classifier that does not depend on the naive assumption and instead models the terms' dependencies during the classification process. Because of the increasing computational costs and difficulty to estimate the parameters accurately and robustly in the presence of limited data, methods that generalize the naive Bayes classifier without utilizing the assumption of independence poor performance. On the one hand, assuming perfect dependency results to a Bayesian network model that is computationally extremely expensive. Allowing minimal amounts of dependency, on the other hand, has been found to give excellent tradeoffs between computational accuracy results and costs. Despite being a practical approximation, the independence assumption, it has shown that the approach is theoretical. Extensive experimental testing has shown that the naive classifier performs well in practice.

By introducing fresh features for each of these elements, the Bayes method provides a natural way to include such additional information into the classification process. For classification, the standard Bayes technique is utilized in conjunction with this augmented representation. Other types of domain knowledge, such as hyperlink information, have been incorporated into the classification process using the Bayes technique.

When the training data is organized in a taxonomy of categories, the Bayes method is also suitable for hierarchical classification. The Open Directory Project (ODP), Yahoo! Taxonomy, and a number of news sites, for example, all have huge collection of documents organized into hierarchical groups. Context-sensitive feature selection has been observed to produce more relevant classification results. A Bayes classifier is generated at each node in hierarchical classification, which then supplies us with the next branch to follow for classification purposes. Two such methods are proposed, both of which depend on node-specific features for classification.

3.2.2 Multinomial Distribution

A document is treated as a collection of words with related frequencies in this technique. As a result, duplicate elements are permitted in the set of words. We assume that the set of words in the document is designated by *Q* and is selected from the vocabulary set *V*, as in the previous case. The set Q includes the different terms *{ti*1 . . *.tim}* m and their corresponding frequencies *F = {Fi*1 . . *.Fim}*. We denote the terms and their frequencies is denoted by [*Q*,*F*]. The total number of terms in the document (or document length) is denoted by $L = \Sigma mj = 1\ F(i\ j)$. Then, assuming that the document *T* contains the terms in *Q* with the associated frequencies *F*, the goal is to estimate the posterior probability that it belongs to class *i*. The following sampling process can be used to model the Bayes probability of class *i* [16]:

The probability mentioned is denoted by $P(C^T = i|T = [Q,F])$. The document's length is assumed as independence of class label, which is a common used in these models. The document length is used as a priori information to generalize the approach. Two values in order to compute the Bayes posterior are needed to estimate.

$$P(C^T = i|T = [Q,F]) = \frac{P(C^T = i)P(T = [Q,F]|C^T = i)}{P(T = [Q,F])}$$

$$\propto P(C^T = i)P(T = [Q,F]|C^T = i) \quad (3.19)$$

For the purpose of determining the class label for *Q*, it is not essential to compute the denominator, *P (T = [Q, F])*, as in the previous case. The fraction of documents belonging to class *i* can be used to determine the value of the probability *P (CT = I).*The computation of *P* ([*Q, F*] $|C^T = i$) is more complicated. The number of possible ways to sample the distinct terms so s to result outcome [*Q, F*] is provided by *L*! Π*mi* =1 *Fi*! When considering the sequential order of the *L* different samples. Using the naive independence assumption, the probability of each of these sequences is given by Π*t j*∈*QP*(*t j* ∈ *T*)^{Fj},

$$P(T = [Q, F]|C^T = i) = \frac{L!}{\prod_{i=1}^{m} F_i!} \prod_{t_j} P\left(t_j \in T \middle| C^T = i\right)^{F_j}$$

(3.20)

Substitute Equation 3.20 for Equation 3.19 to get the class with the highest Bayes posterior probability, where the class priors are calculated as before, and the probabilities $P(t\ j \in T|C^T = i)$ may be simply estimated with Laplacian smoothing as before. Note that for the purpose of choosing the class with the highest posterior probability, we do not really have to compute *L*! Π*mi* =1 *Fi*! as it is a constant not depending on the class label (i.e., the same for all the classes). We also note that the probabilities of class absence are not present in the above equations because of the way in which the sampling is performed.

A number of different multinomial model variations have been proposed. The research shows how a category hierarchy may be utilized to improve the estimation of multinomial parameters in the naive Bayes classifier and dramatically improve classification accuracy. The fundamental concept is to smooth the parameters for data-sparse child categories and their parent nodes using shrinkage techniques. As a result, the training data for related categories is effectively "shared" in a weighted manner, improving the robustness and accuracy of parameter estimate when there is insufficient training data for each child category. The Bernoulli and multinomial models were compared extensively on different corpora in this study, and the following conclusions were resented:

- The multi-variate Bernoulli model can sometimes outperform the multinomial model with small vocabulary sizes chosen optimally for both vice visa when the ideal vocabulary size is selected for both.
- On average, this model decreases the inaccuracy by 27%.

The afore-mentioned results seem to suggest that the two models may have different strengths, and may therefore be useful in different scenarios.

3.3 Preprocessing of the System

Tokenization

- Tokenization is the process of breaking down the text corpus in to individual elements.

Hello! Dear, I am on leave today.

hello	dear	i	am	on	leave	today

Removing Stop Words

- Stop words are unimportant words that frequently appear in texts.
- Stop words are removed first, for example, so, and, or, the, and so on. The stop words in the diagram below are: you, are, that, have, and the.
- Which are removed by using this technique.

You are lucky, that you have won the cash prize

lucky	won	cash	prize

3.4 Term Weighting Schemes (TF-IDF method)

Term Frequency (TF) is term weight or score based on the words frequency appeared in a document. If the word's TF value in a document is high, the term's effect on the document is also high. Inverse Document Frequency (IDF) is the weighting method based on number of words appeared on all the documents. TF-IDF, one of the simplest weighting method, a statistical method, is to measure the important of a word in the document to the whole corpus. TF-IDF and its algorithm version are good performance on a number of various data sets. The formulation of this method is as following;

$$TF - IDF = tf_n(t, d).idf(t) \tag{3.21}$$

Where:

- w(t,d) = term weight in document d
- tf (t,d) = term frequency in document
- N = the total number of document
- n_t = number of documents that have term t

3.5 Multinomial Naïve Bayes Method

Naive Bayes is a probability statistical method based on Bayes Theorem with a strong independent assumption to predict the class of a document based on probability. In this system, the MNB model utilized as an algorithm in the classification. Multinomial Naive Bayes is one of the definite methods of Naïve Bayes, analyzed text documents using word counts. (TF-IDF) approach is used to indicate text documents rather of including binary values in Naïve Bayes Classifier. A probabilistic classifier which counts the probabilities of every class by employing Naïve Bayes theorem. It will calculate the conditional probability of each comment in the individual category. The maximum-likelihood estimate use to calculate the term frequencies based on the training data to approximate the class-conditional probabilities in the multinomial model:

Bayes Theorem is started the formula,

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

$$Posterior\ Probability = \frac{Likelihood \times Class\ Prior\ Probability}{Predictior\ prior\ Probability}$$

, (3.22)

Where:

- $(c|x)$ is the posterior probability of class (*c* , *target*) given predictor (*x* , *attributes*)
- $P(c)$ is the prior probability of class
- $P(x|c)$ is the likelihood which is the predictor probability given class
- $P(x)$ is the prior of predictor

Where the probability of a document *d* for class *c* can be calculated by the following formula:

$$P(c|term\ document\ d) = P(x_1|c) \times P(x_2|c) \times, \dots, \times P(x_i|c) \times P(c)$$

(3.23)

Where,

- *P(c)* is the prior probability of class *c*
- x_i is the word *i* in document *d*
- $P(c|term\ document\ d)$ is the probability of a document including class *c*
- $P(x_i|c)$ is the probability of the word *i* is known class *c*.

Prior probability in class *c* is determined by the formula:

$$P(c)\frac{N_c}{N}$$

(3.24)

Where,

- N_c is the number of class in all documents
- *N* is the number of all documents.

The probability of the word *i* is determined using the following equation:

$$P(x_i|C_j) = \frac{\sum tf(x_j, d \in c_j) idf(x_i) + \propto}{\sum N_{d \in c_j} + \propto . V}$$

(3.25)

Where,

- x_i: A word from the feature *x* of a particular samples.
- $\sum tf(x_i, d \epsilon c_j).\ idf(x_i)$: The sum of TF-IDF all words xi from all documents in the training samples that belong to class c_j.
- $\sum N_{d \epsilon c_j}$: The sum of all term frequencies in the training dataset for class c_j.
- α: An additive smoothing parameter (α= 1 for Laplace smoothing).
- *V*: The size of the vocabulary (number of different words in the training set).

CHAPTER 4
SYSTEM DESIGN AND IMPLEMENTATION

Nowadays, informal online communities like Face book and YouTube have become dramatically as a common digital platforms to society. People use social media for different reasons to stay in touch with their friends and family and to also share their thoughts and ideas in blogs as a virtual community platform. As long as technology is developing, these platforms attract a great number of clients. Besides, it is easy to meet spammers' targets. YouTube are well-known as an informal community among the youngsters. For example, many makeup tutorials are intended to teenage girls by bloggers or beauty influences because only most of them are interested in beauty. These days, most clients create their new YouTube content (videos) in every day. That why spammers can get an opportunity to make irrelevant content directed to the users. These unnecessary or unwanted messages are planned to disturb the users by sending out invitation them into clicking links to view unacceptable sites containing phishing, spyware, malware, and scams. There is the highlighted feature in YouTube that is the comments section. This feature allows users to share their opinions and suggestions. The user can post their every video and give a comment on the other' posted videos.

In this system, the comments section of YouTube videos are presented the prediction whether the spam or not by using the concept called machine learning or artificial intelligence. The proposed classification algorithm (Multinomial Naïve Bayes) is used to surmise and analyze the spam comment. The purpose of this system is to submit briefly the techniques of machine learning and to outline technique how to predict. The conventional data analysis techniques are much more superior to the other. A new opportunity can be opened to search and expand the prediction accuracy by machine learning.

This system can develop a YouTube comment spam classification framework by using term frequency – inverse document frequency (TF-IDF) and Multinomial Naive Bayes. TF-IDF is a statistical method to measure the important of a word in the document

to the whole corpus. Multinomial Naïve Bayes classifier was used to categorize the YouTube comments into the suitable category.

4.1 The System Flow

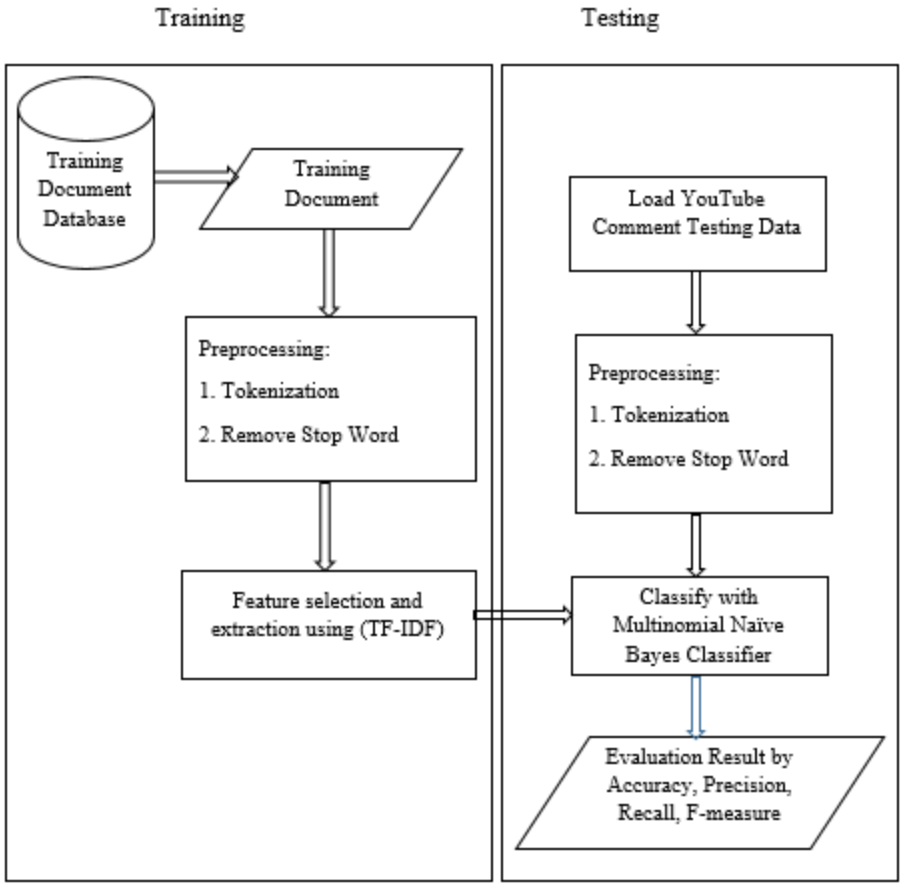


Figure 4.1: YouTube Comment Text Classification Model

This figure 4.1 shows the concise interaction stream of the proposed framework. The handling periods of the framework can be isolated as Training Phase and Testing Phase. In both stages, information pre-handling (Tokenization and Removing Stop Words) will occur prior to computing weight or characterization. The concise clarification of pre-handling steps are shown as follow and the excess advances which are made sense of in section 4.4 are expressed in detail.

4.2. Case Study

Training Data Set (Sample Data Set):

No.	Content	Class
1	Nice songï»¿	0
2	I love song ï»¿	0
3	Fuck it was the best ever 0687119038 nummber of patrik kluivert his son share !ï»¿	1
4	The best world cup song ever!!!!ï»¿	0
5	I like shakira..ï»¿	0
6	SEE SOME MORE SONG OPEN GOOGLE AND TYPE Shakira GuruOfMovieï»¿	1
7	Waka best oneï»¿	0
8	Check out this playlist on YouTube:ï»¿	1
9	i remember this song!	0
10	Check out this playlist on YouTube:Central ï»¿	1

Step1: Tokenization for Training Data Set

No.	Content	Class
1	nice / song	0
2	i / love / song	0
3	fuck / it / was / the / best / ever / 0687119038 / nummber / of / patrik / kluivert / his / son / share	1
4	the / best / world / cup / song / ever	0
5	i / like / shakira	0
6	see / some / more / song / open / google / and / type / shakira / guruofmovie	1
7	waka / best / one	0
8	check / out / this / playlist / on / youtube	1
9	i / remember / this / song	0
10	check / out / this / playlist / on / youtube / central	1

Step2: Stop Words Removing for Training Data Set

No.	Content	Class
1	nice/ song	0
2	love/ song	0
3	fuck/ best/ nummber/ patrik/ kluivert/ son/ share	1
4	best/ world/ cup/ song	0
5	like/ Shakira	0
6	song/ open/ google/ type/ shakira/ guruofmovie	1
7	waka/ best	0
8	check/ playlist/ youtube	1
9	remember/ song	0
10	check/ playlist/ youtube/ central	1

Step3: Calculate TF-IDF for Training DataSet

$$TF - IDF = tf_n(t,d).idf(t)$$

TF.IDF for Comment 1
nice = 1/song = 1/

Total term for Comment 1 ==> 2

TF.IDF (nice) = (1/ 2) * Log(10 / 1) = 1.1513

TF.IDF (song) = (1/ 2) * Log(10 / 5) = 0.3466

TF.IDF for Comment 2

love = 1/song = 1/

Total term for Comment 2 ==> 3

TF.IDF (love) = (1/ 3) * Log(10 / 1) = 0.7675

TF.IDF (song) = (1/ 3) * Log(10 / 5) = 0.2310

TF.IDF for Comment 3
fuck = 1/best = 1/nummber = 1/patrik = 1/kluivert = 1/son = 1/share = 1/

Total term for Comment 3 ==> 14

TF.IDF (fuck) = (1/ 14) * Log(10 / 1) = 0.1645

TF.IDF (best) = (1/ 14) * Log(10 / 3) = 0.0785

TF.IDF (nummber) = (1/ 14) * Log(10 / 1) = 0.1645

TF.IDF (patrik) = (1/ 14) * Log(10 / 1) = 0.1645

TF.IDF (kluivert) = (1/ 14) * Log(10 / 1) = 0.1645

TF.IDF (son) = (1/ 14) * Log(10 / 1) = 0.1645

TF.IDF (share) = (1/ 14) * Log(10 / 1) = 0.1645

TF.IDF for Comment 4

best = 1/world = 1/cup = 1/song = 1/

Total term for Comment 4 ==> 6

TF.IDF (best) = (1/ 6) * Log(10 / 3) = 0.1831

TF.IDF (world) = (1/ 6) * Log(10 / 1) = 0.3838

TF.IDF (cup) = (1/ 6) * Log(10 / 1) = 0.3838

TF.IDF (song) = (1/ 6) * Log(10 / 5) = 0.1155

TF.IDF for Comment 5

like = 1/shakira = 1/

Total term for Comment 5 ==> 3

TF.IDF (like) = (1/ 3) * Log(10 / 1) = 0.7675

TF.IDF (shakira) = (1/ 3) * Log(10 / 2) = 0.5365

TF.IDF for Comment 6

song = 1/open = 1/google = 1/type = 1/shakira = 1/guruofmovie = 1/

Total term for Comment 6 ==> 10

TF.IDF (song) = (1/ 10) * Log(10 / 5) = 0.0693

TF.IDF (open) = (1/ 10) * Log(10 / 1) = 0.2303

TF.IDF (google) = (1/ 10) * Log(10 / 1) = 0.2303

TF.IDF (type) = (1/ 10) * Log(10 / 1) = 0.2303

TF.IDF (shakira) = (1/ 10) * Log(10 / 2) = 0.1609

TF.IDF (guruofmovie) = (1/ 10) * Log(10 / 1) = 0.2303

TF.IDF for Comment 7

waka = 1/best = 1/

Total term for Comment 7 ==> 3

TF.IDF (waka) = (1/ 3) * Log(10 / 1) = 0.7675

TF.IDF (best) = (1/ 3) * Log(10 / 3) = 0.3662

TF.IDF for Comment 8

check = 1/playlist = 1/youtube = 1/

Total term for Comment 8 ==> 6

TF.IDF (check) = (1/ 6) * Log(10 / 2) = 0.2682

TF.IDF (playlist) = (1/ 6) * Log(10 / 2) = 0.2682

TF.IDF (youtube) = (1/ 6) * Log(10 / 2) = 0.2682

TF.IDF for Comment 9

remember = 1/song = 1/

Total term for Comment 9 ==> 4

TF.IDF (remember) = (1/ 4) * Log(10 / 1) = 0.5756

TF.IDF (song) = (1/ 4) * Log(10 / 5) = 0.1733

TF.IDF for Comment 10

check = 1/playlist = 1/youtube = 1/central = 1/

Total term for Comment 10 ==> 7

TF.IDF (check) = (1/ 7) * Log(10 / 2) = 0.2299

TF.IDF (playlist) = (1/ 7) * Log(10 / 2) = 0.2299

TF.IDF (youtube) = (1/ 7) * Log(10 / 2) = 0.2299

TF.IDF (central) = (1/ 7) * Log(10 / 1) = 0.3289

Table 4.1: TF-IDF Calculate Results of Training Data

Term(words)	TF-IDF									
	Comment 1	Comment 2	Comment 3	Comment 4	Comment 5	Comment 6	Comment 7	Comment 8	Comment 9	Comment 10
nice	1.1513									
song	0.3466	0.2310		0.1155		0.0693			0.1733	
love		0.7675								
fuck			0.1645							
best			0.0785	0.1831			0.3662			
number			0.1645							
patrik			0.1645							
kluivert			0.1645							

son			0.1645							
share			0.1645							
world				0.3838						
cup				0.3838						
like					0.7675					
shakira					0.5365	0.1609				
open						0.2303				
google						0.2303				
type						0.2303				
Guruofmo ive						0.2303				
waka							0.7675			
check								0.2682		0.2299
playlist								0.2682		0.2299
youtube								0.2682		0.2299
remember									0.5756	
central										0.3289
Total	1.4979	0.9985	1.0655	1.0662	1.304	1.1541	1.1337	0.8046	0.7489	1.4612
Class	0	0	1	0	0	1	0	1	0	1

Step4: Testing Phase:

Testing Data	Check out our Channel for nice Beats!!ï»¿
Tokenization	check / out / our / channel / for / nice / beats
Stop words remove	check/ channel/ nice/ beats

Step -4.1

Compute the prior probability value of each class:

$$P(c)\frac{N_c}{N}$$

$$\boldsymbol{P(Ham) = \frac{6}{10} = 0.6, P(Spam) = \frac{4}{10}, 0.4}$$

Step -4.2

Compute the probability of the words *i* in the document:

$$P(x_i|C_j) = \frac{\sum tf(x_j, d \in c_j) idf(x_i) + \propto}{\sum N_{d \in c_j} + \propto . V}$$

The value of $\sum N_{d \in c_j}$ is obtained by the follow:

Ham = 1.4979+0.9985+1.0662+1.304+1.1337+0.7489 = 6.7492

Spam = 1.0655+1.1541+0.8046+1.4612 = 4.4854

P(check | Ham)= [0 + 1] / (6.7492+ 24) = 0.0325

P(check | Spam)= [(0.2682+0.2299) + 1] / (4.4854+ 24) = 0.0526

P(channel | Ham)= [0 + 1] / (6.7492+ 24) = 0.0325

P(channel | Spam)= [0 + 1] / (4.4854+ 24) = 0.0351

P(nice | Ham)= [1.1513+ 1] / (6.7492+ 24) = 0.0700

P(nice | Spam)= [0 + 1] / (4.4854+ 24) = 0.0351

P(beats | Ham)= [0 + 1] / (6.7492+ 24) = 0.0325

P(beats | Spam)= [0 + 1] / (4.4854+ 24) = 0.0351

Step 4.3,

Compute the probability of the test document to know the class using *Equation (3):*

$$P(c|term\ document\ d) = P(x_1|c) \times P(x_2|c) \times, \dots, \times P(x_i|c) \times P(c)$$

P(H| check, channel, nice, beats) = P(check | H) ×P(channel | H) ×P(nice | H) ×P(beats | H) × P(H)

= 0.0325× 0.0325× 0.0700× 0.0325× 0.6= 0.00000144178

P(S| check, channel, nice, beats) = P(check | S) ×P(channel | S) ×P(nice | S) ×P(beats | S) × P(S)

= 0.0526 × 0.0351× 0.0351 × 0.0351× 0.4= 0.00000090984

The highest probability values is 0.00000144178. The test document is Ham

4.3 Implementation of the System

The proposed system implements the YouTube Spam comments classification by using ASP.Net Language on Microsoft Visual Studio 2015 Version IDE and Microsoft SQL Server 2017 Express version for database engine. The classification analysis will be developed by using TF.IDF and Multinomial Naïve Bayes Approach. The system implementation of the proposed system are shown in the following figure

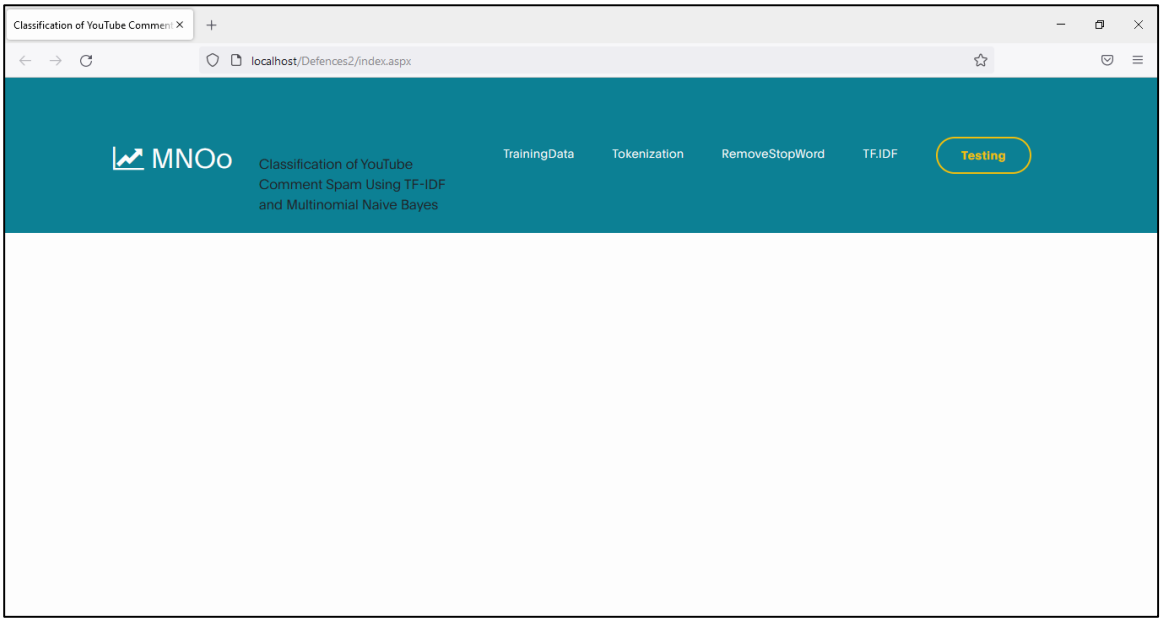


Figure 4.2: System Main Page

Preprocessing is important prior to remove the elements. Stop words, for example, action word to be, pronouns, relational words and conjunctions do not give significant data for opinion examination. Thus, the stop words are eliminated to save the handling time. The means of preprocessing are as following as figure 4.3 and figure 4.4.

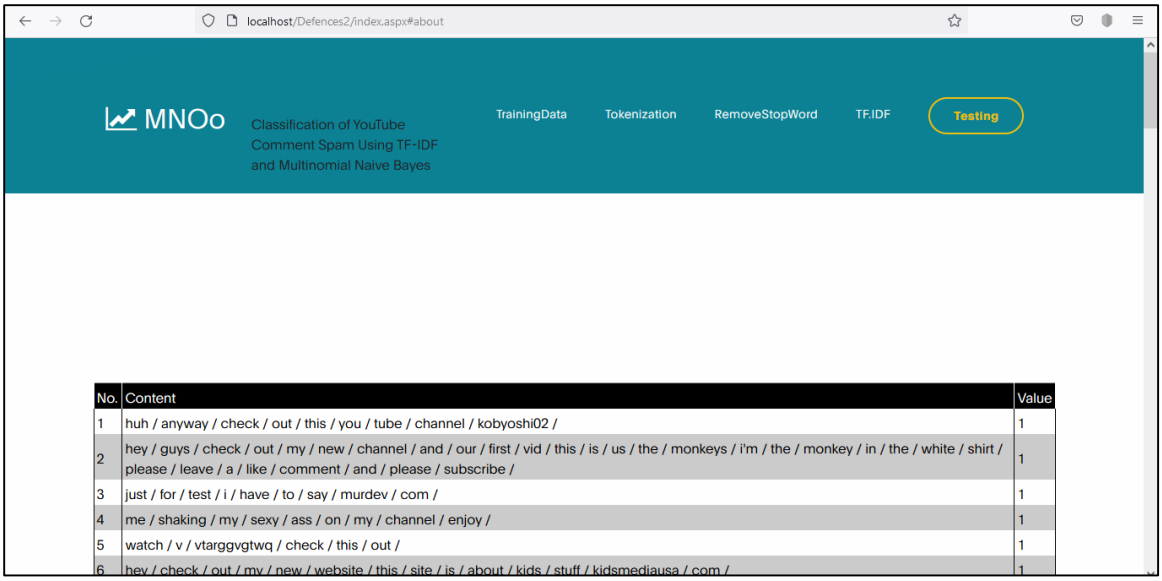


Figure 4.3: Training Data Tokenization

Figure 4.3 shows the "Tokenization" step of the training process. Detail processes of the tokenization are as follow:

- Individual words are formed from the comments.
- Tokenizing splits the words from the training comments.
- Tokenizing is done by the help of String Tokenizer.

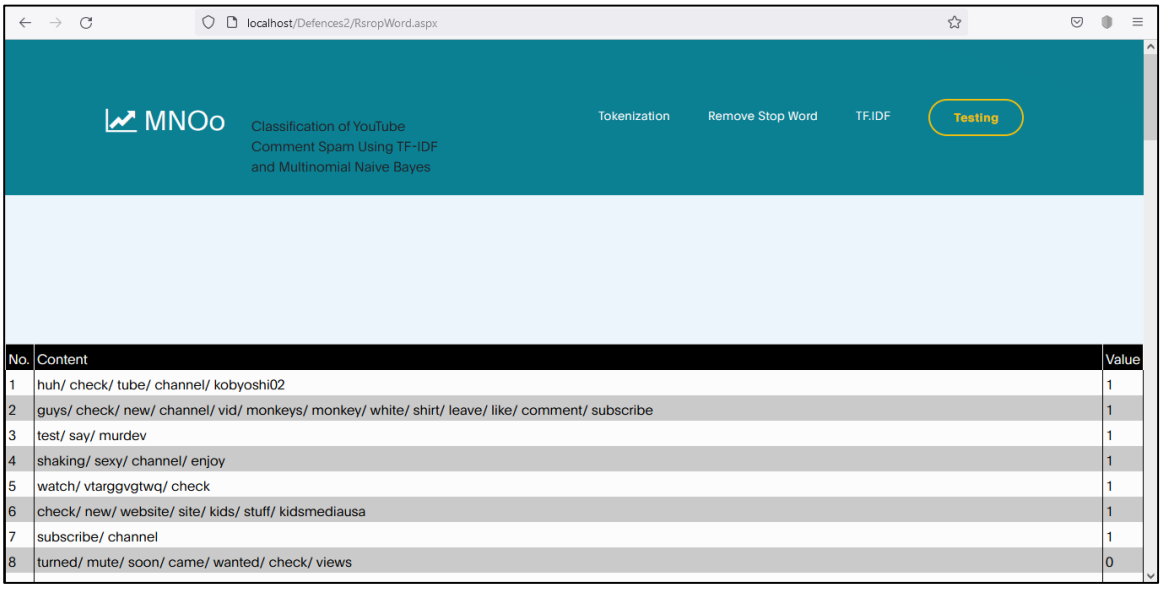


Figure 4.4: Removing Stop-word on Training Data

Figure 4.4 is the step of removing stop-words from training data.

- Words that do not have particular meaning such as “a”, “and”, “the” and some other common words should be removed.
- Stop-words lists are stored in the system database.

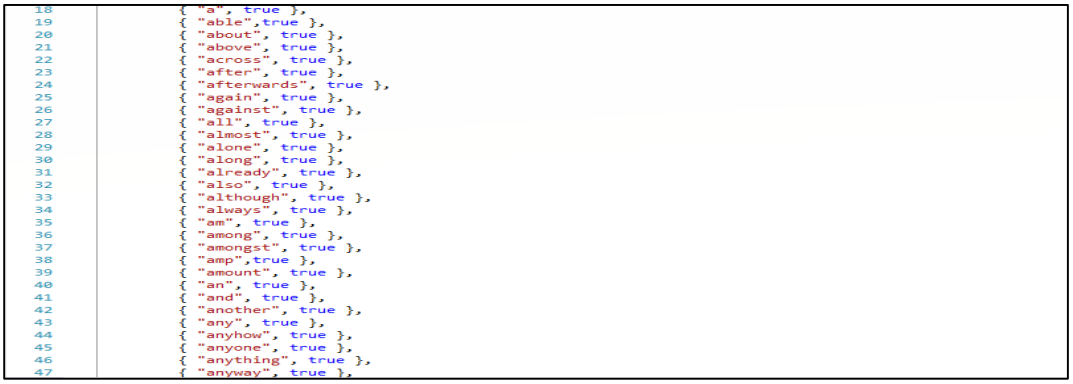


Figure 4.5: List of Stop Word

- The stop-words removes from this system after tokenizing

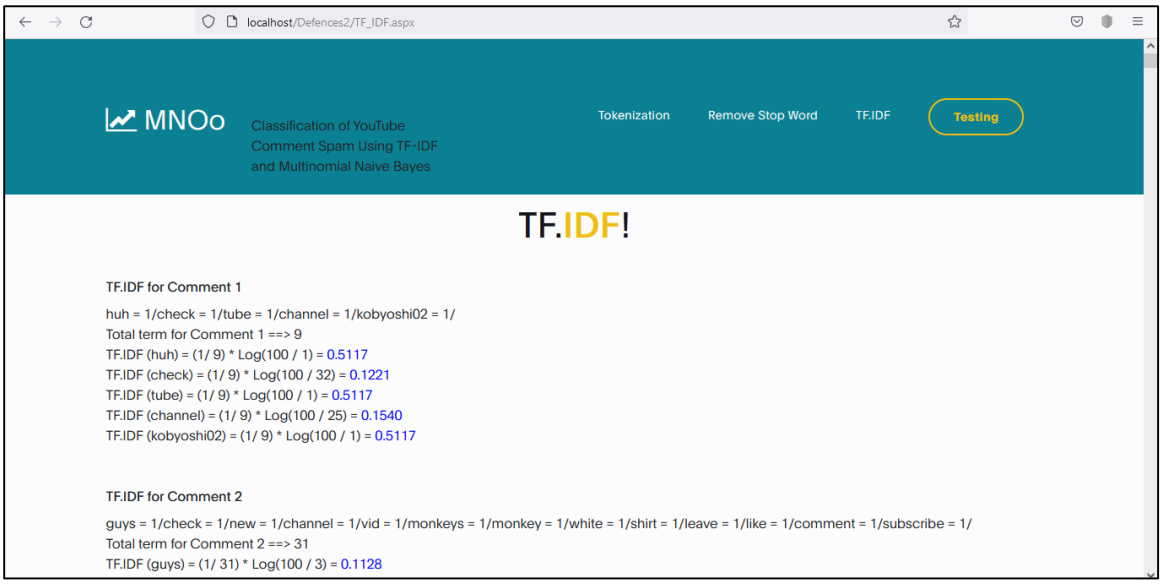


Figure 4.6: Weight Calculation on Training Data (TF.IDF)

IF.IDF is the element determination technique it is a factual measure that assesses how significant a word is to a report in an assortment of reports. This is finished by duplicating two measurements: how frequently a word shows up in a record, and the backwards report recurrence of the word across a bunch of records. After the assignment of component determination is done, the preparation interaction is finished. Then the

testing stage can be used to continue for the recognition of Spam Mail. The testing information stacking pages is displayed in the accompanying figure 4.6.

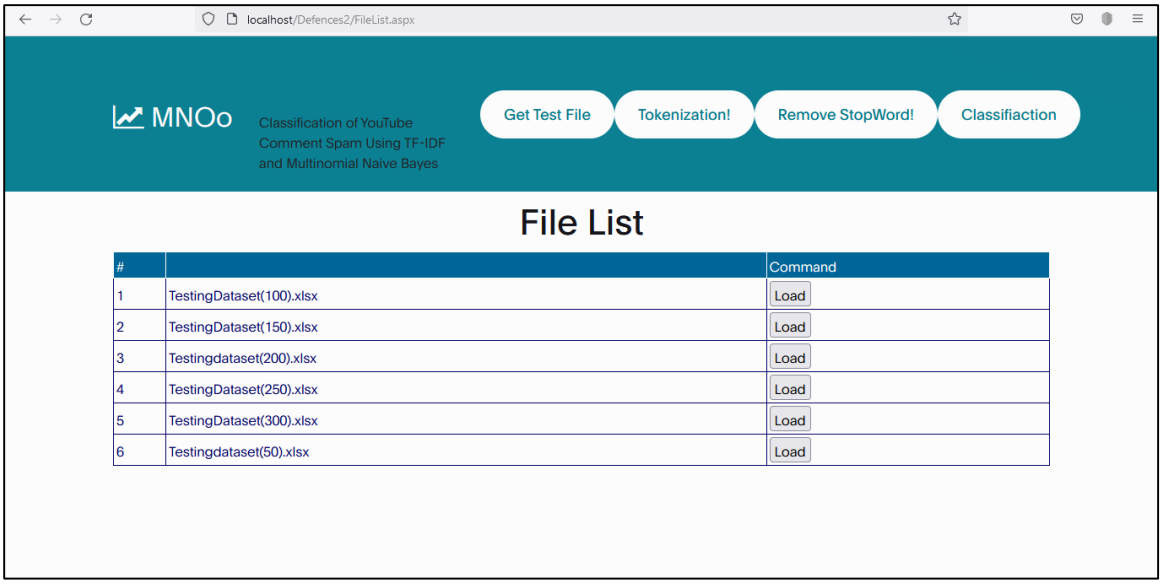
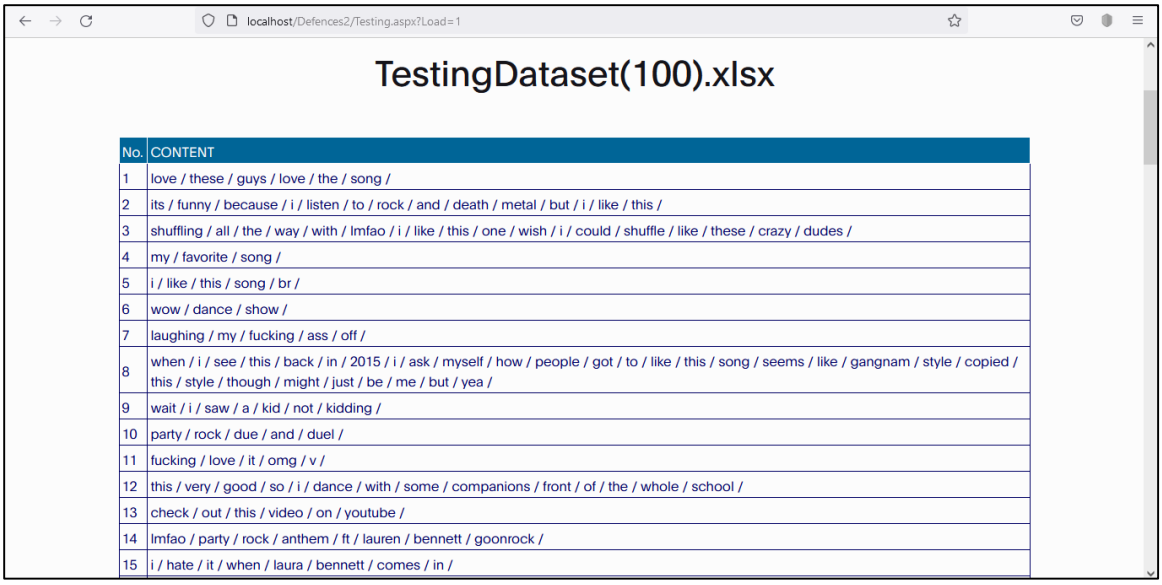


Figure 4.7: Testing Data Loading

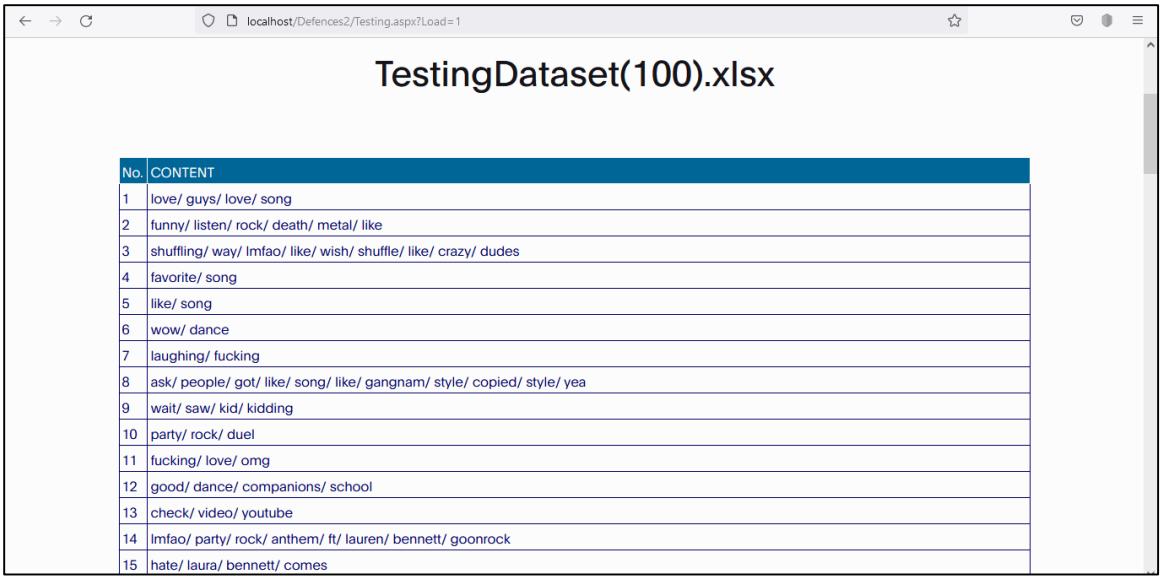
Stop words are accessible in overflow in any human language. By eliminating these words, the low level data from the eliminated text can give more clarity of mind to the significant data. The evacuation of all words denote state inappropriate results on the model that is trained for errand. Evacuation of stop words certainly diminishes the dataset size and the preparation time is decreased in this manner because of the less number of tokens associated with the preparation.



No.	CONTENT
1	love / these / guys / love / the / song /
2	its / funny / because / i / listen / to / rock / and / death / metal / but / i / like / this /
3	shuffling / all / the / way / with / lmfao / i / like / this / one / wish / i / could / shuffle / like / these / crazy / dudes /
4	my / favorite / song /
5	i / like / this / song / br /
6	wow / dance / show /
7	laughing / my / fucking / ass / off /
8	when / i / see / this / back / in / 2015 / i / ask / myself / how / people / got / to / like / this / song / seems / like / gangnam / style / copied / this / style / though / might / just / be / me / but / yea /
9	wait / i / saw / a / kid / not / kidding /
10	party / rock / due / and / duel /
11	fucking / love / it / omg / v /
12	this / very / good / so / i / dance / with / some / companions / front / of / the / whole / school /
13	check / out / this / video / on / youtube /
14	lmfao / party / rock / anthem / ft / lauren / bennett / goonrock /
15	i / hate / it / when / laura / bennett / comes / in /

Figure 4.8: Testing Data Tokenization

After the testing data is loaded, tokenization phase is needed to proceed as shown in figure 4.9. The words for the most part sifted through prior to handling a characteristic language that are called stop words. These are really the most widely recognized words in any language (like articles, relational words, pronouns, conjunctions, and so on) and do not add a lot of data to the text. Illustration of a few stop words in English are "the", "a", "an", "so", "what". Stop word eliminating in testing dataset is taken set as displayed in figure 4.8.



No.	CONTENT
1	love/ guys/ love/ song
2	funny/ listen/ rock/ death/ metal/ like
3	shuffling/ way/ lmfao/ like/ wish/ shuffle/ like/ crazy/ dudes
4	favorite/ song
5	like/ song
6	wow/ dance
7	laughing/ fucking
8	ask/ people/ got/ like/ song/ like/ gangnam/ style/ copied/ style/ yea
9	wait/ saw/ kid/ kidding
10	party/ rock/ duel
11	fucking/ love/ omg
12	good/ dance/ companions/ school
13	check/ video/ youtube
14	lmfao/ party/ rock/ anthem/ ft/ lauren/ bennett/ goonrock
15	hate/ laura/ bennett/ comes

Figure 4.9: Testing Data Stop-word Removing

After the pre-processing of the testing data, spam mail classification is ready to classify. In the classification phase, the detail calculation steps are visually supported in user interface as shown in figure 4.9. The performance evaluation of the testing data classification is discussed in section 5.

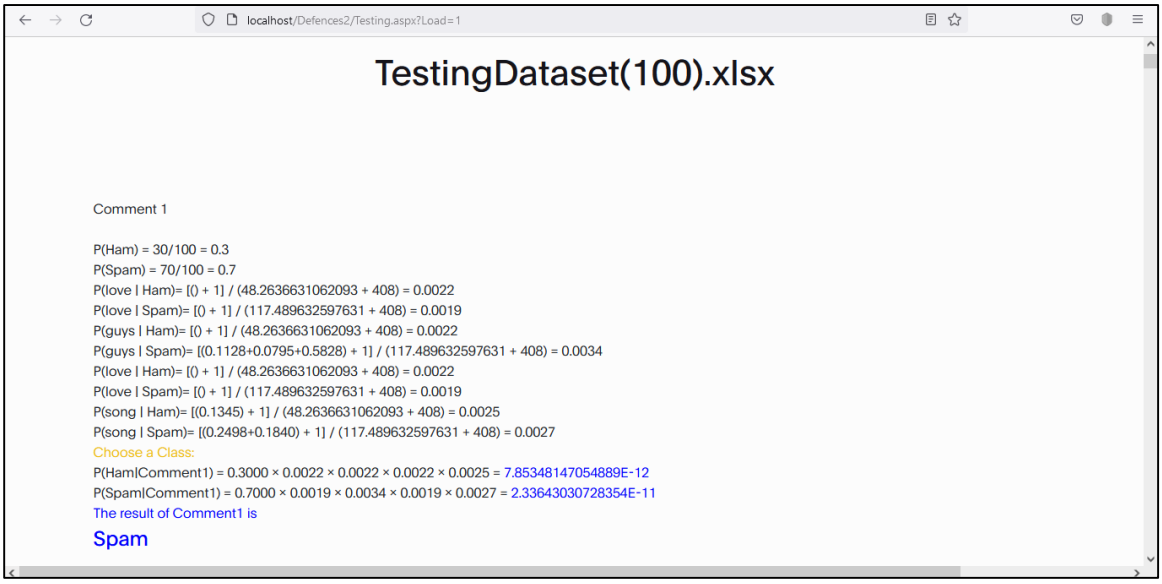


Figure4.10: Naïve Bayes Calculation

4.4 Evalulation Performance

Four evalulation performances which are precision, recall, F-measure and accuracy are used to estimate the efficiency of the system. These are calculated by using Equation (4.1), (4.2), (4.3), (4.4).

$$Accuracy = \frac{TP+tN}{TP+FN+FP+TN} \quad (4.1)$$

$$Recall = \frac{TP}{TP+FP} \quad (4.2)$$

$$Precison = \frac{TP}{TP+FN} \quad (4.3)$$

$$F-measure = \frac{2*Recall*Precision}{Recall+Precision} \quad (4.4)$$

Where:

1. True Positive (TP): the classifier predicted ‘spam ‘and the comment were actually spam.
2. True Negative (TN): the classifier predicted ‘not spam’ and the comment were actually real.
3. False Positive (FP): the classifier predicted ‘spam’ and the comment were actually real.
4. False Negative (FN): the classifier predicted the comment ‘non spam’ and the comment were actually spam.

In this YouTube Spam and Ham classification system, experiments are made for 7 times. In each analysis of 7 different training dataset and testing dataset pairs are used (Test1: Training Data 100 records and Testing Data 50 records; Test2: Training Data 150 records and Testing Data records 50; Test 3: Training Data 400 records and Testing Data 100 records; Test 4: Training Data 700 records and Testing Data 150 records; Test 5: Training Data 1000 records and Testing Data 200 records; Test 6: Training Data 1300 records and Testing Data 200 records; Test 7: Training Data 1655 records and Testing Data 300 records). This system makes the experiment result’s performance evaluation based on Accuracy, Precision, Recall and F-measure of each analysis. The analysis results of 7 different dataset are shown in table 4.1 and figure 4.10.

Table 4.2: Experimental Results of Analysis

Testing No	Training and Testing data Proportion	Accuracy Result	Precision Result	Recall Result	F-Measure
Test 1	100/50	52%	50%	83.33%	62.50%
Test 2	150/50	62%	57.14%	83.33%	67.80%
Test 3	400/100	71%	57.89%	62.86%	60.27%
Test 4	700/150	81.33%	93.88%	64.79%	76.67%
Test 5	1000/200	90%	91.84%	88.24%	90%
Test 6	1300/200	91%	91.05%	92.42%	91.73%
Test 7	1655/300	93.67%	93.67%	94.27%	93.97%

Based on the analysis, the system can give better classification result if the more trained data can feed to this system.

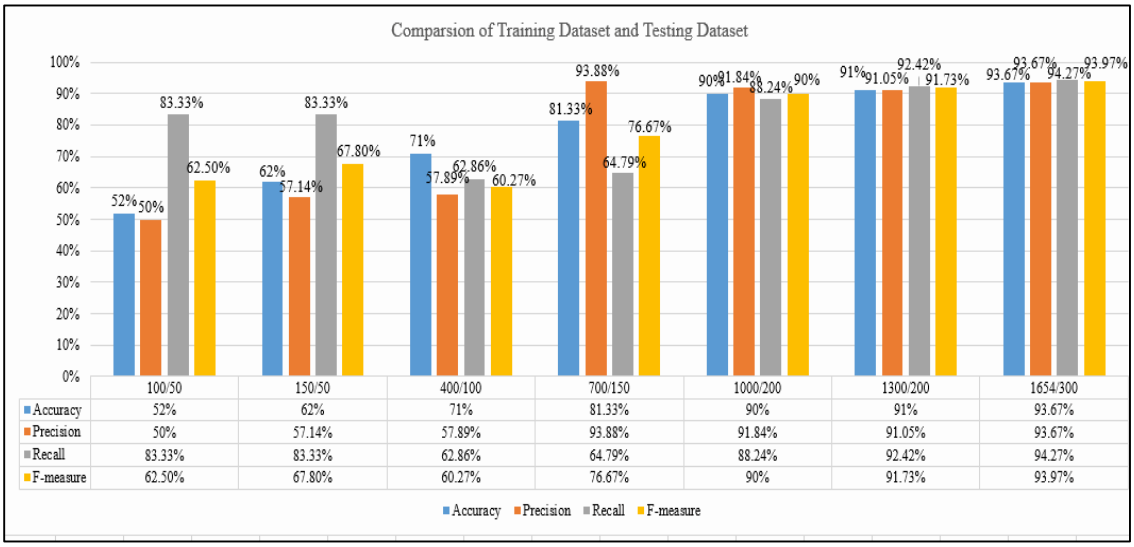


Figure 4.11: Experiment Results with Different Datasets

CHAPTER 5
CONCLUSION, LIMITATIONS AND FURTHER EXTENSIONS

This study intends to classify and develop the YouTube comments as spam or ham documents not only by using feature selection method also Multinomial Naïve Bayes classifier. The various components, concepts, and sample calculations of this system are investigated and their implementation to the overall performance of the system are analyzed. In this chapter, the main contents of the study are concluded, benefits and limitations of the system, and future work are suggested.

5.1 Conclusion

YouTube is considered as one of the most popular videos sharing websites that is growing very fast. Because of its popularity, it attracts different types of spammer, who publish unwanted spam comments. A system classifies the YouTube Comment Spam that is Ham or Spam. The system mainly focuses on implementing the classifier for YouTube comment spam classification using Multinomial Naïve Bayes approach with feature selection (TF-IDF) methods. Multinomial Nave Bayes is one of the most effective and widely used in text classification algorithms. This can be seen by the performance evaluation result which is sufficient. Multinomial Naïve Bayes requires a huge dataset to define the data accurately this system is used YouTube Comment Spam dataset, form UCI machine repository.

5.2 Advantages of the System

The proposed system serves high-performance with high accuracy for comments that are uploaded on YouTube. The Multinomial Naive Bayes Classifier cannot directly work with categorical data. For better results TF-IDF is typically utilized on the text before training the model. The use of stop words is removed before TF-IDF resulted in more accurate text classification.

The tf-idf feature selection methods can provide appropriate feature vectors to improve the performance of the classifier. According to the system's evaluation, after the training step is done, the system can perform effectively in classification in a short time.

5.3 Limitations and Further Extensions

This music file can be set up own dataset or future extension. In addition, future extension may be possible by using own dataset under such a popular music. Moreover such a music dataset can be applied along with other classifier by comparison.

REFERENCES

[1] Adele Custler, “Random Forests” Machine Learning January 2011.

[2] Aharon Bar-Hillel, Tomer Hertz, “Learning Distance Functions Equivalence Relations” Conference Paper, Julty 2003.

[3] Alperen Degirmenci, “Introduction to Hidden Markov Models” 2014.

[4] Daniel Berrar, “Bayes’ Theorem and Naïve Bayes Classifier” January 2018.

[5] David E.Losads, “Assessing Multi-variate Bernoulli models for Informaion Retrieval” ACM, 2008.

[6] Hiroshi Shimodair, “Text Classification using Naïve Bayes”, January-March 2020.

[7] https://www.ic.unicamp.br/~rocha/teaching/2011s2/mc906/aulas/naive-bayes-classifier.pdf, Naïve Bayes Classifier.

[8] Huma Lodhi “Boosting Strategy for Classification” in Intelligent Data Analysis, July 2002.

[9] Joseph K, Blitzstein, Joseph K, Blitzstein, “Probability and counting” 2014.

[10] Kristine Monteith, “Truning Bayesian Model Averaging Into Bayesian Model Combination”, July 2011.

[11] Lopamudrs Dey, “Sentiment Analysis of Review Datasets Using Naïve Bayes and K-NN Classifier”, Department of Computer Scienace and Engineering, Heritage Institute of Technology Kolkata, India, 2016.

[12] Luis Mena, Jesus A. Gonzalez, “Symbolic One-class Learning from Imbalanced Datasets: Applicaion in Medical Diagnosis” International Journal on Artificial Intelligence Tools, Vol. 18, No. 2 (2009).

[13] Michael Collins, “The Naïve Bayes Model, Maximum-Likelihood Estination and the EM Algorithm”.

[14] Mohd Khalid Awang, “Improving Customer Churn Classification with Ensemble Stacking Method” (IJACSA) International Journal of Advanced Computer Science and Application.

[15] Ner Ni Hlaing, M.C.Sc 2017, “Performance Analysis on Mail Classification with Multilayer Perceptron (MLP).

[16] Niemeyer, “Conditional Random Fields for Lidar Pont Could Classificaion”, ISPRS Annals of the Photogrammetry Remote Sensing and Spatial Information Sciences, Volume I-3, 2012.

[17] Paras Seth, “SMS Spam Detection and Comparison of Various Machine Learning Algorithm”, International Conference Computing and Communication Technologies for Smart Nation, 2017.

[18] Probability Calculation Using Logistic Regression, from https://docs.tibco.com/pub/sfire-dsc/6.5.0/doc/html/TIB.

[19] Rosaida Rosly, “Comprehensive study on ensemble classification for medical applications” International Journal of Engineering and Technology, April 2018.

[20] Rui Manuel Gameiro de Castro, Silvio Miguel Frgoso Rodrigues “An Overview of Deep Learning Strategies for Times Series”. Electrical and Computer Engineering June 2018.

[21] Taca Rosa, Adi Wijaya, “Comparsion of Distance Measurement Methods on K-Nearest Neighbor Algorithm for Classification” Sriwijava International Conference on Information Technology and Its Applications (2019).

[22] Thipireddy Rishith Reddy, “Difficulty Level Prediction of a Question Paper Using Naive BnaïveClassifier", Journal of Xi’an University of Architecture and Technology, ISSN No: 1006-7930, 2020.

[23] Wenzhao Lian, “Modeling Time Series and Sequences: Learning Representatins and Makinig Predictions”, 2015.

[24] Yiheng Li, “A Comparative Performance Assessment of Ensemble Learning for Credit Scoring”. 2020.