NLP based Model for Classification of Complaints: Autonomous and Intelligent System

Author
Qurat-ul-ain
NUST CEME 320367

Supervisor
Dr. Arslan Shaukat

DEPARTMENT OF COMPUTER AND SOFTWARE ENGINEERING
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
ISLAMABAD
MARCH, 2022

NLP based Model for Classification of Complaints: Autonomous and Intelligent System

Author

Qurat-ul-ain

NUST CEME 320367

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

MS Computer Engineering

Thesis Supervisor:

Dr. Arslan Shaukat

Thesis Supervisor's Signature:____

DEPARTMENT OF COMPUTER AND SOFTWARE ENGINEERING
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
ISLAMABAD

MARCH, 2022

Declaration

I certify that this research work titled "NLP based Model for Classification of Complaints: Autonomous and Intelligent System" is my work. The work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged/referred to.

Signature of Student
Qurat-ul-ain
NUST CEME 320367

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical, and spelling mistakes. The thesis is also according to the format given by the university.

Signature of Student
Qurat-ul-ain
NUST CEME 320367

Signature of Supervisor
Dr. Arslan Shaukat

Copyright Statement

Acknowledgements

I am extremely thankful to Allah Almighty for giving me patience, guidance, and the ability to carry out this work. Without HIS blessings I could not have completed my thesis. Indeed Allah is the most merciful and worthy of praise.

I would like to express my gratitude to my supervisor Dr. Arslan Shaukat for their guidance, valuable suggestions, and moral support for the entire duration. It’s a blessing to have such great people as a mentor for research.

I am also thankful to the entire thesis committee: Dr. Usman Akram and Dr. Farhan Hussain and Dr. Ali Hassan for their tremendous support and cooperation.

My acknowledgment would be incomplete without mentioning my beloved parents without whom I am nothing and who have supported me in every second of my life. I would also like to thank my sisters who have always supported me and especially during the research work.

Finally, I would like to express my gratitude to my friend and all the individuals who have encouraged and supported me throughout the entire duration.

Dedicated to my exceptional parents, adored siblings, and friends whose tremendous support and cooperation led me to this wonderful accomplishment

Abstract

These days, Artificial Intelligence is playing a key role in the progression of humanity as it helps to curtail human struggle in every aspect of life. An Immense amount of data is in structured and non-structured foam from numerous industrial platforms that are striving to get into the shape of useful information to be a part of scientific research. Although today's major concern is how to manage a huge amount of feedback data i.e., Text format citizen complaints. At this point, proposing a model that automatically classifies the textual complaints by analyzing the content with the help of NLP (Natural Language Processing) and different ML (machine learning) models can be beneficial. Primarily, data of complaints are collected from the concerned platforms as well as from the international Consumer Complaint Database (for validation). The methodology is comprised of four different stages i.e. (1) initial pre-processing (2) preprocessing (3) future extraction (a) count vectorizer (b) term frequency-inverse document frequency (TF-IDF) (4) ML models for categorical classification of the complaints. At the evaluation stage, 10 different classes are present in assembled complaint dataset and more than 70 % accuracy is achieved from all classifiers. Likewise, on Consumer Complaint Dataset, 86% accuracy has been achieved. This model is used to optimize the complaint division automatically and saves a lot of time.

Table of Contents

Declaration iii

Language Correctness Certificate iv

Copyright Statement v

Acknowledgements vi

Abstract viii

List of Figures xii

List of Tables xiv

1 Chapter 1: Introduction 1

1.1 Motivation 2

1.2 Aims and Objectives 3

1.3 Structure of Thesis 3

2 Chapter 2: Technical Background 4

2.1 Natural Language Processing (NLP) 4

2.1.1 Industrial applications on NLP 4

2.1.2 Corpus 6

2.1.3 Tokenization 7

2.1.4 N-gram 8

2.2 Feature Extraction Techniques 8

2.2.1 Word Vector 9

2.2.2 Bag-of-words 10

2.2.3 Count Vectorization 10

2.2.4 TF-IDF 11

2.2.5 Hashing Vector 12

2.3 Machine learning 12

2.3.1 Training and test data 12

2.3.2 Over and underfitting 12

2.4 Machine learning algorithms 13

2.4.1 Support Vector Machine 14

2.4.2 Multinomial Naive Bayes Algorithm 15

2.4.3 Random Forest 16

2.4.4 KNN 17

2.4.5 Logistic Regression 17

2.5 Evaluation 17
2.5.1 10-fold cross-validation 17
2.5.2 Classification report 18
2.5.3 Confusion matrix 19
3 Chapter 3: Literature Review 20
3.1 Text Categorization with Support Vector Machines 20
3.2 Sentiment analysis of IMDB movie reviews 20
3.3 Application of Text mining for classification of Community Complaints and Proposals 21
3.4 Bank Chatbot – An Intelligent Assistant System Using NLP and Machine Learning 21
3.5 Restaurant reviews classification using NLP Techniques 21
3.6 Complaint Analysis and Classification for Economic and Food Safety 22
3.7 Article Classification using Natural Language Processing and Machine Learning 22
3.8 Towards Explainable NLP: A Generative Explanation Framework for Text Classification 22
3.9 Fake News Detection with Different Models 22
3.10 Machine learning text classification model with NLP approach 23
3.11 A generalized approach to sentiment analysis of short text messages in natural language processing 23
4 Chapter 4: Proposed Methodology 28
4.1 Construct a conceptual framework 28
4.1.1 Problem tree 28
4.2 Dataset 28
4.2.1 Local Data Set I 29
4.2.2 Consumer Complaints Data Set II 29
4.2.3 Initial Pre-processing 30
4.2.4 Data Analysis 31
4.3 Pre-processing data 32
4.3.1 Tokenization 32
4.3.2 Conversion of text into lower case 33
4.3.3 Removal of stop words and punctuation marks 34
4.3.4 Lemmatization 35
4.4 Feature Extraction Techniques 36
4.4.1 Count Vector 36
4.4.2 TF-IDF 37

4.5 Classification 38
4.5.1 Split Data 38
4.5.2 Machine learning model 38
4.6 Evaluate model 40
5 Chapter 5: Experiment & Results 41
5.1 Local Dataset1 41
5.1.1 Count Vector 41
5.1.2 TF-IDF 46
5.1.3 Accuracies of local dataset I 51
5.2 Consumer Complaints Dataset 2 52
5.2.1 Count Vector 52
5.2.2 TF-IDF 57
5.2.3 Accuracies of Kaggle Dataset II 62
5.3 Error analysis 63
5.4 Imbalanced data 63
5.5 Comparing the Consumer Complaints results 64
7 Chapter 6: Conclusion and Future work 66
7.1 Future Work 66
7.2 Limitations 66
References 67

List of Figures

Figure 1.1: Demonstrate the type of complaints 2
Figure 2.1: The workflow of a machine learning process [13]. 12
(a) Right fitting by the model (b) Overfitting by the model Figure 2.2 Visual representation of overfitting[23] 13
Figure 2.3: Machine Learning Diagram 14
Figure 2.4 Representations of training texts [24] 15
Figure 2.5: The suggested decision boundary [24] 15
Figure 2.6: Random forest with 2 decision trees[26] 16
Figure 2.7 10-fold cross-validation procedure [32]. 18
Figure 2.8 Confusion Matrix [35] 19
Figure 4.1 Problem tree 28
Figure 4.2 Problem tree of the dataset branch 29
Figure 4.3: Showing the architecture of our dataset in Pandas 30
Figure 4.4: Dataset I has a total of n samples per class 31
Figure 4.5: Dataset II has a total of n samples per class. 31
Figure 4.6: Demonstrate stages of pre-processing 32
Figure 4.7: Before and after the tokenization of the data frame 33
Figure 4.8: Before and after transforming all letters to lowercase in the data frame. 33
Figure 4.9: After eliminating punctuation marks from the data 34
Figure 4.10: Removed stop words from the data frame 35
Figure 4.11: After Lemmatization on the dataset 35
Figure 4.12: Demonstrate stages of Feature Extraction Technique 36
Figure 4.13: Division of data samples in training and testing 38
Figure 5.1: Classification reports of Local dataset I from various ML models by Count Vector technique 42
Figure 5.2: Confusion matrix of Local dataset I from various ML models by Count Vector technique 44
Figure 5.3: Confusion matrix from Count Vector of 10-Cross Validation (Logistic Regression) 45
Figure 5.4: Accuracy graph on Local Dataset I from Count Vector Technique 46
Figure 5.5: Classification reports of Local dataset I from various ML models by TF-IDF technique 48
Figure 5.6: Confusion matrix of Local dataset I from various ML models by TF-IDF technique 49
Figure 5.7: Confusion matrix from TF-IDF of 10-Cross Validation (Logistic Regression) 50
Figure 5.8: Accuracy graph on Local Dataset I from TF-IDF Technique 51
Figure 5.9: Classification reports of Consumer Complaints Dataset II from various ML models by Count Vector technique 53
Figure 5.10: Confusion matrix of Consumer Complaints Dataset II from various ML models by Count Vector technique 55
Figure 5.11: Confusion matrix from Count Vector of 10-Cross Validation (logistic Regression) on Consumer Complaints Dataset 56
Figure 5.12: Accuracy graph on Consumer Complaints Dataset II from Count Vector Technique 57
Figure 5.13: Classification reports of Consumer Complaints Dataset II from various ML models by TF-IDF technique 58

Figure 5.14: Confusion matrix of Consumer Complaints Dataset II from various ML models by TF-IDF technique 60
Figure 5.15: Confusion matrix from TF-IDF of 10-Cross Validation (logistic Regression) on Consumer Complaints Dataset 61
Figure 5.16: Accuracy graph on Consumer Complaints Dataset II from TF-IDF Technique 62

List of Tables

Table 3.1: Breakdown of literature review 23
Table 4.1: Model of Count Vector 36
Table 4.2: Model of TF-IDF Vectorizer 37
Table 5.1: Cross-Validation of Local dataset I from various ML models by Count Vector technique 44
Table 5.2: Cross-Validation of Local dataset I from various ML models by TF-IDF technique 49
Table 5.3: Overall Accuracies of the local dataset I. 51
Table 5.4: Cross-Validation of Consumer Complaints Dataset II from various ML models by Count Vector technique 55
Table 5.5: Cross-Validation of Consumer Complaints Dataset II from various ML models by TF-IDF technique 60
Table 5.6: Overall Accuracies of Consumer Complaints Dataset II. 62
Table 5.7: Accuracies of Consumer Complaints Dataset [50] 64

Chapter 1: Introduction

Humans are currently using artificial intelligence (AI) in almost every aspect of their lives. The private industry is not the only one looking into methods to use artificial intelligence (AI) to solve its unique issues while also using its vast volumes of organized and unstructured data. Text from social networks and newswires and textual forms of direct communication between individuals and political institutions are being processed using natural language processing (NLP) methods [1].

It's critical and beneficial to differentiate between various sorts of text documents. This is applicable in a wide range of scenarios. To be valid, automating these jobs must be on par with or better than human performance. Automated Essay Scoring has received a lot of attention since it is a process that is both time-consuming and crucial. It's possible to automate the removal of content from social media sites that violate the rules of use or are unlawful (e.g., hate speech or threats of physical harm).

Many government agencies offer electronic services. These organizations process citizens' input (such as requests or complaints), commonly done by email or contact forms in so-called virtual counters. A country's or administrative region's population density may quickly overwhelm a person's ability to maintain meaningful personal connections. It is possible to apply NLP approaches to enhance public services based on this data [2].

These virtual counters categorized the various types of information they received. Despite these recent developments, classifying brief texts still provides significant issues Figure 1.1: Demonstrate the type of complaints. Textual context may be inferred from conventional texts since the language itself is extensive and "clear" enough to comprehend. Additionally, organized phrases that adhere to a particular language's syntactical principles may be found.

When the text is brief, however, this is not the case. It is necessary to solve the issues posed by the ambiguity and sparsity of brief messages and widespread misspellings. For example, the phrase "more resources for the plant" is a concise description of the procurement process of a business. Noun phrase (subject) + verb are not a typical English clause construction. In addition, the statement might imply that either the physical plant or the industrial plant requires more resources than the one being discussed. Traditional NLP approaches such as syntactic parsing cannot handle this problem because of the restricted vocabulary and absence of context and grammatical cues in these brief messages.

Figure 1.1: Demonstrate the type of complaints

1.1 Motivation

This paper focuses on the needs of the public sector complaint portal; a public administrative authority that worked with regard to education and scholarship, answerable for observing and authorizing administrative regulation. One of the principal contributions of this organization is including citizen complaints on the quality of education, with more than 20,000 grievances being gotten yearly. Normally, over 30% of these are viewed not as on the horizon of this authority; the remaining are sent to explicit functional units. The utilization of human work to dissect and appropriately handle these grievances is a bottleneck, carrying the need to mechanize this cycle to the extent possible. The quality of user-generated material (often referred to as UGC [3] is very variable, which makes it difficult to utilize contact forms to collect useful information.

This study will evaluate if supervised classifiers can effectively be used to automatically assign labels to different complaints. Different types of classifiers will be evaluated and compared against each other in different aspects. There is a large variety of algorithms that can be used, and many of them work very differently. Some published reports have, for instance, used tree-based, deep learning, or lazy learning, to perform multi-label classification [4]. Developers that use supervised learning for text classification must select a classifier that is both suitable for their data set and project requirements. These developers can also decide to incorporate a few or a very large number of unique labels. This report will examine how algorithms scale performance has given different designs and a different number of output labels

In this study, we examine two datasets one of 10,000 education sector complaints whereas the other has 1,20,000 complaints regarding the ranks. A machine learning-based classifier is used for classifying the complaints accurately in the English language to our knowledge.

1.2 Aims and Objectives

The goal of this research is to compare the performance of several classifiers. The following questions will be addressed:

- Can multi-label classification be consistently performed using supervised learning algorithms?
- Which classifier performs best when there are a variety of output labels to choose from?

The first goal is to collect a large amount of data that can be categorically classified. Next, the data needs to be cleaned and transformed into a suitable format. Classifiers with text input and a categorical output will be implemented and trained. These classifiers will be constructed that take multiple classes and must handle a different number of outputs variables. Finally, performance metrics have to be established and the models will have to be evaluated.

1.3 Structure of Thesis

The following is how this piece is organized:

Various characteristics recovered by researchers in the past for text categorization are described in Chapter 2, as well as technological background.

A comprehensive assessment of the literature and previous research on text categorization is presented in Chapter 3. It also outlines the qualities that researchers have gleaned and the databases that may be tapped into.

In Chapter 4, the suggested technique is described in great depth. Pre-processing stages, feature extractors, and algorithms are all included.

Figures, tables, and performance metrics for all of the experiments are included in the 5th chapter.

The thesis is concluded in Chapter 6, which also shows the research's future directions.

Chapter 2: Technical Background

This thesis mainly focuses on the machine learning algorithm of supervised learning for tackling complaints by two different feature extraction techniques in NLP. In this chapter, we first introduce Natural Language processing with a focus on how they deal with different feature extractions. We then describe the different algorithms of supervised machine learning and also the evaluation of those algorithms in detail.

2.1 Natural Language Processing (NLP)

It is known as Natural language processing which uses natural language data as an input or produces as an output source. Computers have a harder time interpreting unstructured human communication than organized communication. Natural language is very complex for example: If "I saw a lady with a binocular on the hill" is used, it might mean either that I observed her in her use of the binocular on the hill or that I saw her with the binocular in her possession on the hill. These ambiguities might lead to various problems in multiple domains such as in the medical book. Thus, the remedy must be discovered [5,6]. Only until computers are capable of translating the real meaning of individual words in a sentence if transcriptions are employed efficiently. Speech analysis and research focus mostly on the study of an audio signal, but NLP comes in handy when it comes to the semantic list of information i.e., words and sentences once they have been detected [7].

There is a variety in a language such it is counties as well as discrete. One cannot understand the meaning of a symbol from the symbol itself. Thus, the relationship between words may vary. To conclude, one must consider the meaning of the symbols in the context in which they are used. To compare two separate colors in the same image, you don't have to rely on complicated lookup tables or complicated procedures. There are letters, words, and phrases that make up the language, in addition to being created. Words are limited in number, but the ways words may be employed to convey meaning are almost infinite.

2.1.1 Industrial applications on NLP

NLP plans to overwhelm human-to-machine interaction to the place were conversing with a machine is essentially as simple as conversing with a human. NLP keeps on connecting unstructured information and making it significant to a machine. IDC as of late determined that the amount of examined information by mental frameworks will develop by a component of 100 to 1.4 ZB by 2025 affecting a large number of businesses and organizations all over the globe [8]. Mechanical technology, medical care, monetary administrations, associated auto, and smart homes are a portion of the areas that will keep on being progressed by NLP.

One of the beginning usages of NLP in the early years of 2000 was machine interpretation to function as an interpreter starting with one human language then onto the next. In any case, it quickly observed its acknowledgment in the customer service industry. The most well-known utilization of NLP in client support is called "Chatbots" or Virtual Collaborations.

Industrial applications of NLP can be mostly categorized into 3 categories: Conversational systems, Text Analytics, Machine translation

2.1.1.1 Conversational Systems

Using a speech or text-based interface, we may converse with an automated computer in natural language using a conversational system. A company firm helps to automate complex procedures with 24X7 assistance to its users. Virtual Assistants and Chatbots are the two most frequent types of conversational gadgets. Banks, e-commerce, social networking, and other self-provider factor-of-income systems all use these devices to serve their clients with a wide range of services.

2.1.1.2 Text Analytics

Text Analytics additionally called textual content mining pursuits to extract meaningful content from text, either in files, emails, or brief-form communications such as tweets and SMS texts [8]. Most commonplace use cases of textual content analytics on social media analytics.

2.1.1.3 Machine Translation

Device translation is the venture of automatically translating one natural language into another, retaining the means of the input text [9]. Maximum famous software for device translation is Google translator. Other machine translation software programs are also utilized in speech translation and teaching. Now, we will observe some industrial packages in the following area regions: Healthcare, car, Finance, manufacturing, retail, education, and customer service.

2.1.2 Corpus

A tremendous quantity of data is used in the development of NLP-related applications. The word "corpus" may be used to describe a big set of data. As a result, the corpus may be formalized and technically defined as follows:

Using a corpus is a way to study how language is utilized in written or spoken form, which is saved on the computer. In other words, a corpus is a digital collection of real language used for linguistic and corpus analytic purposes. Having over one corpus is termed a corpus.

A corpus of written or spoken language content is necessary for the development of NLP applications. To assist us to create NLP applications, we utilize this content, which includes all input data. A single corpus is sometimes used as input by NLP systems, while other times numerous corpora are used as input.

The following are just a few of the numerous benefits of leveraging corpora while creating NLP applications:

- Statistics like frequency distribution and word co-occurrence are all possible with corpus data. Rest assured, we'll cover some fundamental static analyses of corpora later on.
- It is possible to design and test linguistics rules for different NLP applications. A grammatical correction system uses the text corpus to look for grammatical errors, and then defines the grammatical rules that may be used to gather these errors.
- There are several linguistic rules that may be defined based on how the language is used. The rules-based method makes it possible to construct linguistic rules and then test those rules against a corpus of text.

Authentic and communicative contexts are the contexts in which language ideas may be studied in detail via the examination of the corpus. While updating some information, we'll be talking about the accessible corpus that can be accessed, retrieved, and analyzed by an organization.

For corpus analysis, there are four primary areas, which include statistically probing, altering, and generalizing the dataset. For corpus text data analysis, we tend to focus on the total number of words in the corpus, rather than the frequency of individual terms in the corpus. We look for any noise in the corpus and attempt to eliminate it. Basic corpus analysis is required for nearly every

NLP application, so we can better comprehend our data. Nltk comes with a built-in corpus. This pre-existing corpus is what we use for corpus analysis. To get the most out of nltk, it is essential to know what kind of corpora it contains.

There are four kinds of corpora in Nltk. Peruse the following list of topics in turn.

- **Isolated corpus:** A collection of literature or language processing is an isolated corpus. As a starting point, the corpus includes works such as Gutenberg and online content.
- **Categorized corpus:** These are writings that have been categorized according to a predetermined set of criteria.
  For example, the brown corpus comprises data for a wide range of topics, such as current events, hobbies, and so on.
- **Overlapping corpus**: These texts are classified, however, the categories intersect with each other in an overlapped corpus. The Reuters corpus is an instance of this type of corpus, which includes material that is classified yet whose categories overlap.
- **Reuters corpus** is an example that I wish to describe in further detail. For instance, if you group various varieties of coconuts, you'll have coconut oil subcategories and cotton oil as well. So, in the Reuters Corpus, there is a lot of overlap between the different data types.
- **Temporal corpus:** A collection of natural language used across a period.
  The inauguration speech corpus is an instance of this type of corpus. Let's say you wanted to document the use of a tongue in 1950 in one of India's cities. Afterward, you perform the same exercise to examine how the city's use of language has changed through the years. A variety of data elements about how individuals are using the language and also what changes occurred over time would have been documented.

2.1.3 Tokenization

When you break down a text into its tokens or words known as tokenization. Tokens are used as input for other processes. An example of the technique in action is shown here. You'll get the following tokenized output if you enter the text into the input box. The input is: [Friends, Romans, Countrymen, lend me your ears]; The output after tokenization will be: [’ Friends’, ’Romans’, ’Countrymen’, ’lend’, ’me’, ’your’, ’ears’] [10].

2.1.4 N-gram

An n-gram is a sequence of n consecutive words drawn from a predetermined set of words. A text must be modeled first before it can be used [11]. "I am Smith," for example, maybe condensed from three grams. After converting a two-gram version of the words "I am" and "am Smith."

2.2 Feature Extraction Techniques

Features extraction refers to the process of converting textual data into real-valued vectors that may be fed into machine learning models. Various approaches have been explored, and some textual illustrations have been researched and reported in this field, but further study is needed.

The letters are the primary source of information when the focal item is a word that has been taken out of context. For example, terms like "booking," "booked," and "books" all share the same lemma, which is a dictionary entry for the word. Lemma lexicons and morphological analyzers are often used to achieve this mapping. Language-defined processes may not perform effectively with forms that aren't included in the lexicon or that are misspelled in error. The tough process is stemming it maps multiple similar words into each other. Stemmed words, like "picture," "pictures," and even "pictured" may be used interchangeably and stemmed from ‘pictu’, but they aren't grammatically correct terms. Rather than being accessible to people, lexical resources are computerized dictionaries. For example, some lexicons map conjugated word forms to their potential morphological analyses, indicating that a certain word may be a single masculine noun or even a past perfect verb. There are lexicons like this.

The count, words, and the arrangement of the letters in the text are the characteristics when the focal entity is text, such as in sentences or paragraphs of documents. Feature extraction using a "bag of words" is quite popular. We examine the histogram of a text's words. We can compute quantities that are directly derived from the words and the letters, such as the length of the sentence. We can also integrate statistics based on external information. It is common to use a bag of words, Count Vectorization (CV), Term-Frequency Inverse-Document-Frequency (TF-IDF) & Hashing Vectorization weighting [12].

An important aspect of a word in a phrase or document is its context, which includes both the words and sentences surrounding it. As a rule of thumb, it is usual to concentrate on the immediate context of a word by examining the windows around it (with typical values of 2, 5, and 10 words to each side). These techniques are also interested in Absolute word positions, such as "the word comes in at number five in the phrase" or "the word occurs inside the first 10 sentences," which may also be of interest to us.

For example, we may also look at the distance between two words in a context, as well as the identities of those words that occur between them. The structure of sentences in natural language goes beyond the order of their words. Syntax refers to the underlying structure, which is not visible to the naked eye. Even if it isn't directly stated, the language implies it. There are specialized systems for the prediction of linguistic features such as parts of speech, syntactic trees, semantic roles, and discourse interactions. Classification issues may benefit from these predictions. It is also possible to mix several characteristics. As an alternative, we may offer a set of basic characteristics to an ML model and depend on the training approach to choose key combinations of them. It is said that the context in which a word is employed determines its meaning according to the distributional hypothesis. According to the co-occurrence patterns of terms in a large corpus of text, it is feasible to deduce that two words are related. Many algorithms have been developed to take advantage of this. Each word may be assigned to one of many clusters and represented by its membership in one of these groups [13]. Comparable words (with a similar distribution) have similar vectors and embedding-based approaches that encode words as vectors [14, 15].

2.2.1 Word Vector

Vector-space word representations are used in many recent techniques to NLP to address particular problems, such as retrieving, classification, named entity identification, or parsing. Complex ontologies like WordNet [16], which represent numerous sorts of semantic connections among words, may be eliminated by using word vectors (such as synonymy, hypernymy, meronymy, etc). It is not uncommon for analogies to be represented directly in the vector space of word vectors. According to the theory of vector space, the comparison "the king is to man as queen is to woman" should be true:

$$\mathrm{x}_{\mathrm{king}} - \mathrm{x}_{\mathrm{man}} + \mathrm{x}_{\mathrm{woman}} \approx \mathrm{x}_{\mathrm{queen}} \tag{2.1}$$

When it comes to oncology, it's possible to imagine the following:

$$\mathrm{x}_{\mathrm{glioma}} - \mathrm{x}_{\mathrm{glia}} + \mathrm{x}_{\mathrm{connective}} \approx \mathrm{x}_{\mathrm{fibroma}} \tag{2.2}$$

Co-occurrences in big text corpora are used as the basis for the majority of techniques for producing word vectors. Word-document co-occurrence may be assessed using semantic similarity analysis (e.g., using word embeddings [14] or Globally Vectors (GloVe) [15] or word-level (e.g., using word2vec [14]. Using precompiled word vector libraries trained on billions of tokens collected from Wikipedia, the British Gigaword 5, Frequent Crawl, or Twitter is a common approach. All of these libraries were developed with general-purpose software in mind, and they are exclusively accessible in English.

2.2.2 Bag-of-words

Bag-of-words representations of textual materials have been used in the past [12]. Using this method, you may describe a document using a collection of words. Multisets let you account for the number of times a word appears in a document. Bag-of-words representations of documents may readily be converted into vector representations.

Naive Bayes (NB) and SVM text classifiers may be applied using bag-of-words representations, such as those utilizing bigrams or trigrams [17]. They are, however, plagued by two major issues. It is now impossible to use the syntactic structure of sentences since the sequence in which words appear in documents has been altered. The orthogonal representation of separate words, even if they are semantically near, is also true. Furthermore, the unlabeled records, which make up the overwhelming bulk of the dataset, maybe ignored using this approach.

2.2.3 Count Vectorization

Characters and words are not understood by machines. If we want to communicate with a computer using text data, we must first convert it into numbers. It's one of the simplest methods to encode words numerically, and it's also known as count vectorizing.

A matrix is created by CountVectorizer, and every text sample from the document is represented by a row in the matrix. Each cell's value is just the word count.

2.2.4 TF-IDF

Another well-known technique in the field of natural language processing is TF-IDF [18], which stands for the term frequency-inverse document frequency. The frequency of words in the corpus is taken into account and balanced with the frequency of individual words in a given text in TF-IDF.

Some terms, such as "a," "an," "is," "this," and "the," are found in almost every English-language document collection. Most of these words are only "connectors," and as a result, they convey very little about the document's actual content. There are disadvantages to using these "connection words" since they obscure the frequency of more essential and intriguing concepts that may appear less often [19] when all the words on a page are given straight into the learning process. Thus, the term weighting of each token is accomplished using a technique known as the IDF. Inverse document frequency (IDF) is referred to as the inverse of term frequency (Tf). In a document, the frequency or tf is the number of times a word appears. This is how you get at the Inverse document frequency, If:

$$\mathrm{idf(t)} = \log\frac{1 + \mathrm{n_d}}{1 + \mathrm{df(d,t)}} + 1 \tag{2.3}$$

Where $\mathrm{n_d}$ is the total number of documents in the corpus, d denotes a document, and df (d,t) is the total number of documents that have the term t [17].

Tf-idf is computed by multiplying tf (t,d) by IDF (t) as shown below:

$$\mathrm{tf - idf(t,d) = tf(t,d)Xidf(t)} \tag{2.4}$$

The generated tf-idf vectors are then normalized using the following Euclidean norm:

$$\mathrm{v_{norm}} = \frac{\mathrm{v}}{||\mathrm{v}||_2} = \frac{\mathrm{v}}{\sqrt{\mathrm{v_1^2 + v_2^2 + \cdots v_n^2}}} \tag{2.5}$$

Vectors v1, v2, are all part of the feature space, and each is a vector in its own right. Classification and clustering have been proven to be particularly successful with the use of the van TF-IDF algorithm, which was previously used to rank sites in search engines.

2.2.5 Hashing Vector

Integer index mapping projection on the Euclidean unit is known as a hashing vectorizer method if the vectorizer technique employs the hashing trick to get the token string name.

2.3 Machine learning

When a computer learns from examples and past experiences, this is called "machine learning" [20]. Automated computers can make predictions based on data rather than building sophisticated algorithms for individual issues. Training and testing data must be provided for an algorithm to generate these kinds of predictions. Afterward, the method is put to use to develop a machine-learning algorithm.

2.3.1 Training and test data

Getting the right data to feed into a machine learning model is one of the biggest hurdles. Training and testing data sets are included in the data set. To improve the model's accuracy, it is fed with practice data. After the development phase, test data serves as a verification tool. Machine learning models are capable of identifying random test samples to some extent, and this model estimates their accuracy as a percentage. Another consideration is that the data presented should be in line with the situation at hand[21] . Figure 2.1 depicts the development process, beginning with the collection of data and ending with the construction of a final product.

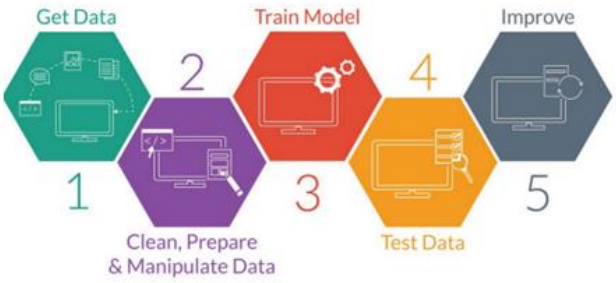


Figure 2.1: The workflow of a machine learning process [13]

2.3.2 Over and underfitting

Models that can generalize from training data and make predictions on fresh data in the same problem area are the objective of machine learning. There are two basic reasons for a machine learning algorithm's poor performance: overfitting and underfitting [22]. Overfitting is a term used to describe a model that detects the pattern in the training data rather than learning from it. Having a huge dataset that is too complicated to fit the model is the most common source of this problem. It is common for an over-fitted model to have relatively high training scores but poor validation scores. A model that is under-fitted, on the other hand, is incapable of generalizing to fresh data. A limited dataset is frequently to blame. While the validation score is reasonably high in an under-fitted model [23], the training score declines(a) *Right fitting by the model* *(b) Overfitting by the model* *Figure 2.2* provides a graphic depiction of the process.

(a) Right fitting by the model (b) Overfitting by the model

Figure 2.2 Visual representation of overfitting[23]

It's impossible to find a universal cure for overfitting since there are so many possible causes. A common solution is to gather additional information. If this isn't achievable, then the cause of the overfitting issue and the remedy to it should be identified. Cross-validation may be used to identify overfitting. Using a method known as cross-validation, you may separate your training data from your test data. Training and testing data are not simply set at predetermined percentages anymore. The k-fold technique is an example of this. K-fold cross-validation divides the dataset into k equal-sized chunks. These data are divided into two sets: a testing set and a training set. With each new testing and training set, this procedure is repeated k times over. A prediction is made for each set in the training data, and these predictions are merged. The model's ability to generalize to new data is shown by the findings. Models that have a high cross-validation score are overfitting

2.4 Machine learning algorithms

Depending on the task, a wide variety of machine learning methods may be selected. Different machine learning approaches are shown in Figure 2.3, along with how they are used in practice. According to the diagram, machine learning may be divided into two major categories: supervised learning and unsupervised learning. The five most prevalent machine learning methods for text categorization are discussed in this chapter.

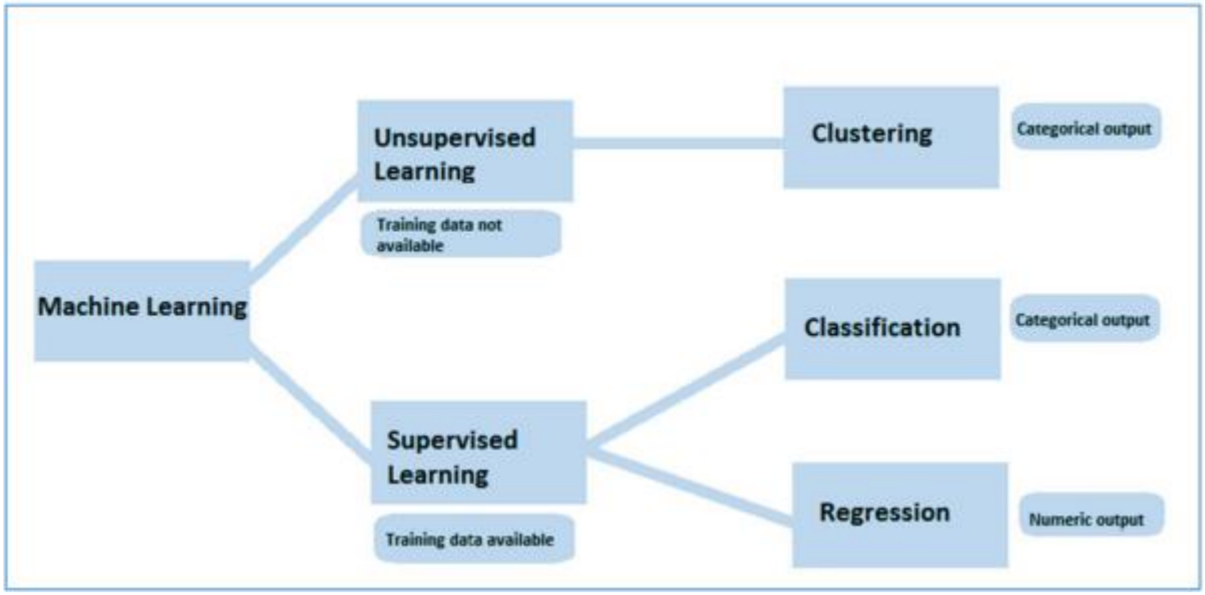


Figure 2.3: Machine Learning System

2.4.1 Support Vector Machine

It is possible to classify text using the SVM Support Vector Machine, a supervised machine learning technique. The algorithm's goal is to find the optimum decision boundary between two vectors that belong to distinct data types. Data structures known as vectors [24] hold information on the spatial coordinates in which they are stored. Deciding where to draw the optimal lines, SVM splits space into two subspaces while making its decision boundary. The category is another name for these subspaces. A product's price is shown in Figure 2.4Figure 2.4 by the circles, which are training data for pricing. The triangles reflect training data that does not define a product's price.

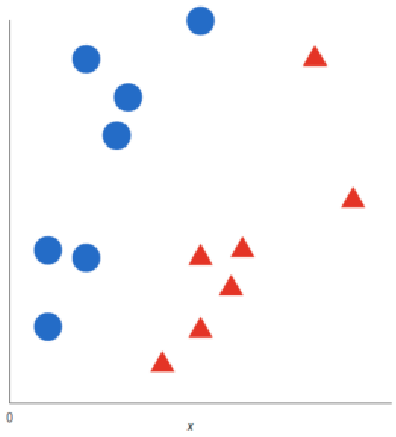


Figure 2.4 Representations of training texts [24]

Using the decision boundary shown in Figure 2.5, we can distinguish between data having product price and data that does not. The data is subdivided into many categories using the hyperplane.

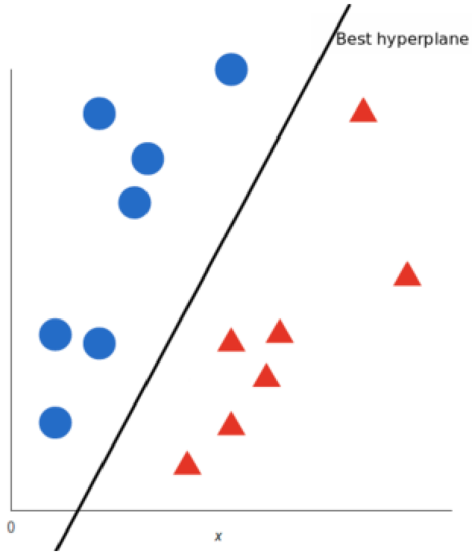


Figure 2.5: The suggested decision boundary [24]

2.4.2 Multinomial Naive Bayes Algorithm

A basic probability-based method, Naive Bayes, is commonly employed in text categorization because of its simplicity. It assumes that every characteristic of the dataset contributes separately to the likelihood of classification, even when there may be relationships between the features.

Naive Bayes is a method that estimates the conditional probability of just one token given a class as the relative incidence of t in all the documents belonging to the class as shown below Multinomial Naive Bayes

$$P(t|c) = \frac{T_{ct}}{\sum_{t\prime} T_{ct\prime}} \tag{2.6}$$

This formula is used to calculate Act [25], the total number of times the word "t" has been used in all texts from class C.

2.4.3 Random Forest

A well-known supervised learning method is the Random Forest algorithm, often known as the random decision forests algorithm. For classification and regression issues, this approach is excellent. For a more accurate and reliable forecast, the random forest method combines the predictions of many unrelated decision trees. Typically, the output class is the class that has occurred the most often as a decision result class [27]. Images of a random forest with two decision trees are shown in Figure 2.6Figure 2.6.

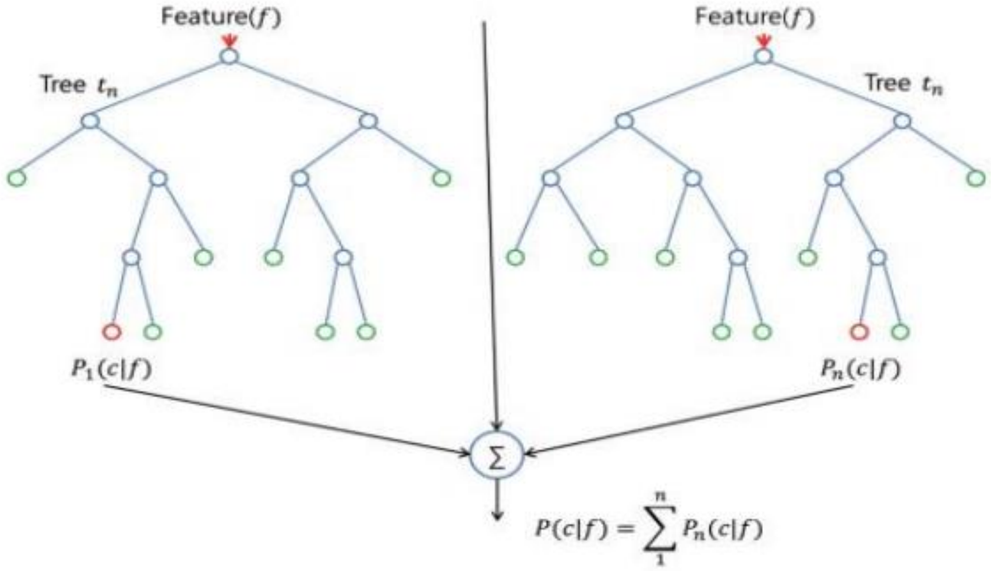


Figure 2.6: Random forest with 2 decision trees [26]

The random-forest method incorporates more randomization into the tree-growing process. "It uses a random subset of characteristics to seek for the best features instead of looking for the greatest feature when separating a node. There is a lot of variety and unpredictability in this process [28]. While splitting a node, only a random subset of characteristics is considered in this approach. Multiple decision trees are created and trained on the same dataset using random forests. To reduce the variation, these deep decision trees are averaged [27].

2.4.4 KNN

A machine-learning algorithm known as KNN is among the simplest of them all. Use of kNN for classification and regression in pattern recognition k closest training examples in sample space are used as input for both classification and regression, and the result is determined by whether classification or regression was used. According to [29], KNN has been utilized for statistical estimates and pattern recognition. Euclidean, Euclidean Squared, City-block, and Chebyshev distances may all be calculated using various methods. Euclidean geometry is the most often used method for determining the separation between two points [30].

Two points x and y in M dimensions are separated by the Euclidean distance [30] (d).

$$d(x, y) = \sqrt{\sum_{i=m}^{M} (x_i - y_i)^2} \tag{2.7}$$

2.4.5 Logistic Regression

Although it may be used as a multi-classifier, the logistic regression model is a binary classifier. It is similar to linear regression for two classes in that each item in the dataset must have a value assigned to it. Logistic regression produces a discrete, rather than a continuous, result, in contrast to linear regression. The likelihood that a value falls into a certain category may be calculated using a logistic function.

2.5 Evaluation

2.5.1 10-fold cross-validation

Cross-validation of ten times According to Figure 2.7, 10-fold cross-validation splits the dataset into 10 random folds. The model is trained on nine parts of the data and tested on six parts of the data. The learning procedure has been repeated a total of 10 times on the training data, and each portion is utilized only once for testing [31].

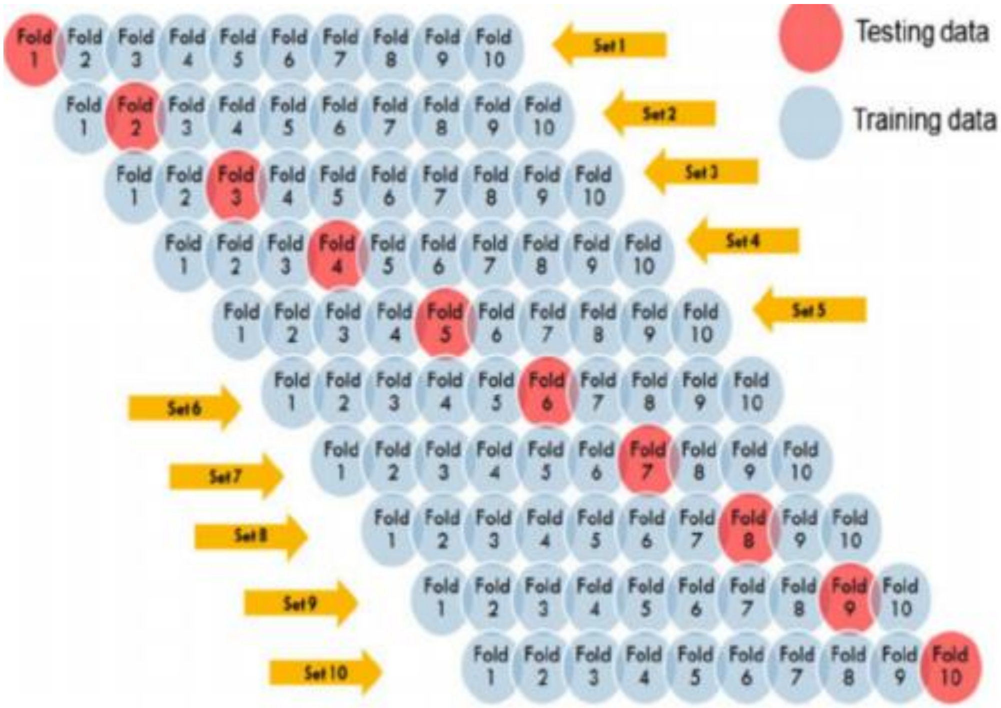

Testing data
Training data
Set 1
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10
Set 2
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10
Set 3
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10
Set 4
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10
Set 5
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10
Set 6
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10
Set 7
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10
Set 8
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10
Set 9
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10
Set 10
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Fold 6 Fold 7 Fold 8 Fold 9 Fold 10

Figure 2.7 10-fold cross-validation procedure [32]

2.5.2 Classification report

Model accuracy, recall, and F1-score are shown in a classification report [33]. The symbols below are explained before we go into the specifics of these phrases.

- Predicted values are negative and real values are also negative.
- The predicted value is negative, while the real value is positive.
- When the forecast is correct and the actual value is incorrect, the FP is true.
- A positive forecast and a good outcome.

True positives and false positives are divided by the total number of true positives and false positives to calculate the precision [34] , see give (2.8) equation below.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.8}$$

See equation 6 below for the formula for recall, which is the total number of properly identified true positives divided by the total of true positives and false negatives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (2.9)$$

F1-score is the balance between precision and recall [35], see equation 7 below.

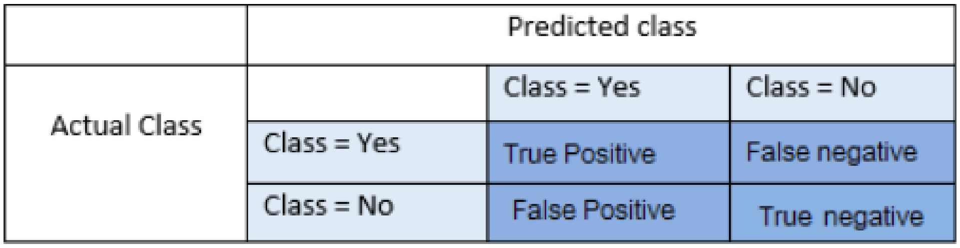
$$\text{F1} - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (2.10)$$

To measure accuracy, we must divide the total number of properly categorized data points by the total number of data points in the dataset. Equation 8 is used to compute it.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (2.11)$$

2.5.3 Confusion matrix

A classification model's prediction results are used to create a confusion matrix [35]. It demonstrates how much a classifier is unsure of itself while generating a prediction based on a given dataset. This is a useful way to assess the classifier's performance. A confusion matrix is shown in Figure 2.8.



	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False negative
	Class = No	False Positive	True negative

Figure 2.8 Confusion Matrix [35]

Chapter 3: Literature Review

In this section, we will look into the theories related to Natural language processing (NLP) with the computational techniques for the automatic analysis and representation of human language. The research on NLP is doing for many years (in which the analysis of a sentence could take up to 7 minutes) to the era of Google and the likes it (in which millions of webpages can be processed in less than a second). The following review papers draw on recent developments in NLP research to look at the past, present, and future of NLP technology in a new light. Studying the relevant classifier literature will help me understand more fully how other factors play a big role in the development of a good with different techniques.

3.1 Text Categorization with Support Vector Machines

With a Variety of Feature-rich Learning Opportunities, SVMs (Support Vector Machines) may be used to learn text categorization, according to Thorsten Joachims [36]. The goal of this essay is to examine the attributes of a text-based learning system and determine why SVM is an excellent choice for text categorization. The algorithm's capacity to generalize in feature spaces, prevent catastrophic failures, and have a long lifespan, according to the paper's author, makes it an effective classification approach. The author fails to include any papers that are relevant to the study's objective. It is clear from the findings that SVM is an effective technique for text categorization.

3.2 Sentiment analysis of IMDB movie reviews

Alejandro Pelaez and his colleagues [37] used sentiment analysis to categorize movies in an IMDB dataset. The study's goal is to see whether customer reviews can be divided into two categories: positive and negative. Support Vector Machine, Multinomial Bayes, Logistic regression, & Random forests are employed in this study's supervised classification techniques. Before running the algorithms, the writers purge the data using NLP, according to the research. Algorithms' precision may be improved by deleting irrelevant terms from the input. Remove capitalization, punctuation, and applying TF-IDF for vectorization are all examples of natural language processing (NLP) methods used. Cross-validation is the method used by the authors to test their system. The study's authors did not provide any references to other studies that could be relevant. The study demonstrates that without TF-IDF and pre-processing, the accuracy is 96 percent, whereas the accuracy after employing them is 98 percent.

3.3 Application of Text mining for classification of Community Complaints and Proposals

There is a way to categorize Jakarta residents' concerns and ideas, according to BN Sanditya Hardaya et al. [38]. An electronic participation tool was created by the Jakarta administration to increase the participation of Jakarta residents in community planning. The e-participant system received approximately 40000 complaints in 2013 and 2014. Using SVM, issues about floods, transportation, residential and land use, and education were categorized into separate subcategories. The paper explains how to use TF-IDF vectorization in conjunction with pre-processing big chunks of text. With the use of pre-processing, the accuracy of 91.37 percent was a few percentage points higher than without.

3.4 Bank Chatbot – An Intelligent Assistant System Using NLP and Machine Learning

Intelligent systems, more like virtual assistants, are what chatbots are. At first, they had a hard time answering all of the clients' questions. As a primary aim, the chatbot should allow customers to speak in English so that the Chabot can respond to their questions as quickly as possible [39]. Vectorization (BOG) and a variety of machine learning models are used to apply the article's ideas. They explain how accurate the testing is. One of the most accurate ML models is the Decision Tree classifier. Other high-accuracy models include the KNN, the Multinomial Naive Bayes, and the Random Forest classifier. The article's strongest aspect is the extensive usage of classifiers to verify the model's correctness. When it comes to time and space, this paper didn't reveal which classifier is the most expensive.

3.5 Restaurant reviews classification using NLP Techniques

Customer feedback on food service, food, and drink was analyzed using machine learning approaches in this article. The review contains information in the form of words, while machine learning relies on numerical data. Since NLP and preprocessing the data are necessary for this, it uses multiple vectorization techniques to produce numerical data [40]. Different classifiers are used, the model is trained, tested, and the accuracy is checked, with Logistic Regression outperforming TF-IDF with an accuracy of 88%. Mainly, this page provides a comparison table of accuracy using several classifiers and vectorization algorithms.

3.6 Complaint Analysis and Classification for Economic and Food Safety

Using a variety of methods, the author [41] demonstrates their findings on how to classify complaints and how to do error analysis. The author compares the accuracy rates of different classifiers. Pre-processing and TIFD feature extraction is used first, followed by classifiers. As a deep learning method, the author uses LSTM and checks the matrix to conduct error analysis on many classes.

3.7 Article Classification using Natural Language Processing and Machine Learning

NLP and machine learning were used to the (.doc) document data that the author [42] retrieved from the internet to categorize the article's subject, as well as to verify the author's information, title, and abstract. Various vectorization methods were employed to cover the whole material in this piece. Accuracy rates range from 76% to 91% when using KNN, Naive Bayes, and SVM as classifiers. The comparison of classifiers provided in this article is useful.

3.8 Towards Explainable NLP: A Generative Explanation Framework for Text Classification

For text categorization, this study [43] has divided the dataset into three parts: lengthy review, short text, and scoring number components. According to the author's theory, neural networks are used. The Combinatorial Explanation Framework (GEF) presented in this study is useful since it describes how to create fine-grained datasets, build the GEF, and apply the suggested framework's least risk training strategy. This paper's flaw is that it takes a long time and doesn't provide as much precision as the algorithm's complexity would suggest.

3.9 Fake News Detection with Different Models

Fake news is now the most troubling problem in our society. Count Vectorization, TF-IDF Vectorization, and numerous Machine Learning Algorithms [44] are used in this study to identify false news (fake news) from various news websites. The experimental presented in this work yields excellent findings, with an average of 15 precisions or accuracies reported.

3.10 Machine learning text classification model with NLP approach

To get over the problem of analyzing data from several chatbots, An NLP platform and machine process of learning are among the goals of Razno [45]. In this post, the author proposes a variety of solutions to the problem of low response rates from NLP Chatbots. Because it explained how to create an error-free chatbot, this article is useful. However, the paper's main weakness is that it does not provide any specific classifiers.

3.11 A generalized approach to sentiment analysis of short text messages in natural language processing

The author of this research [46] employs a variety of strategies to examine the attitudes expressed on various platforms. To do this, they used a variety of preprocessing techniques and examined each result independently. With each pre-processing stage, we use logistic regression as just a classifier to classify the data. The author uses a distinct library, TPOT, which aids in several parts of the process. Each pre-processing step is summarized in the document, making it convenient to use. To determine which steps are the most time are consuming, the article displays the time complexity for each one. Research demonstrates that the Logistic Regression technique has an accuracy of roughly 87%, which is the greatest among the other classifiers.

Table 3.1: Breakdown of literature review

S.No.	Author	Year	Journal /Conference	Dataset	Feature Extraction Technique/Classifier	Accuracy
1	Thorsten Joachims[36]	2019	European conference on machine learning	1. ModApte 2. Ohsumed corpus	1. Bayes 2. Rocchio 3. C4.5 4. KNN 5. SVM (poly) 6. SVM(RBF)	1. 72.0% 2. 79.9% 3. 79.4% 4. 82.35 5. 86.0% 6. 86.4%
2	Talal Ahmed Alejandro [37]	2015	MS Thesis	Github	TF-IDF 1. Frequency Vector 2. Binary Vector	1. 79% 2. 79%

3	I.B.N.S.Hardaya [38]	2017	3rd International Conference on Science in Information Technology (ICSITech)	Private Dataset	1. SVM(without stemming and synonym recognition) 2. SVM(with stemming and without synonym recognition) 3. SVM(without stemming and with synonym recognition) 4. SVM(with stemming and synonym recognition)	1. 89% 2. 90% 3. 90% 4. 91%
4	Chaitrali S. Kulkarni [39]	2017	International Research Journal of Engineering and Technology (IRJET)	Private Dataset	1. Decision Tree 2. Bernoulli Naive Bayes 3. Gaussian Naive Bayes 4. K-nearest neighbor 5. Multinomial Naive Bayes 6. Random Forest 7. Support vector machine	1. 98% 2. 92% 3. 82% 4. 98% 5. 98% 6. 98% 7. 95%
5	Anuradha Tutika [40]	2019	Journal of Information and Computational Science	Super data science	Feature Extraction techniques a) Count Vector b) TF-IDF c) Hashing Vector	Multiple Accuracies With Different techniques

					Classifiers 1. K-NN 2. Logistic Regression 3. SVM	a) Countvectorizer 1. 75% 2. 80% 3. 70% b) TFIDF 1. 78% 2. 88% 3. 68% c) Hashing vectorizer 1. 62% 2. 68% 3. 69%
6	Joao Filgueiras [41]	2019	International Conference on Statistical Language and Speech Processing	The Economic and Food Safety Authority (ASAE) Dataset	1. Random (stratified) 2. Bernoulli NB 3. Multinomial NB 4. Complement NB 5. K-Neighbors 6. SVM (linear) 7. SGD 8. Decision Tree 9. Extra Tree 10. Random Forests 11. Bagging	1. 0.5308 2. 0.6866 3. 0.6661 4. 0.6929 5. 0.5877 6. 0.7953 7. 0.7927 8. 0.7002 9. 0.6532 10. 0.7477 11. 0.7440
7	Tran Thanh Dien [42]	2019	International Conference on Advanced Computing and Applications (ACOMP)	Private Dataset	1. SVM 2. Naïve Bayes 3. KNN	1. 91.2% 2. 80.9% 3. 76.5%

8	Hui Liu [43]	2019	arXiv	1. Skytrax User Reviews Dataset. 2. PCMag Review Dataset.	1. LSTM 2. LSTM+GEF 3. CNN 4. CNN+GEF	1. 76.89 2. 77.96 3. 76.85 4. 79.07
9	Sairamvinay Vijayaraghavan [44]	2020	arXiv	Git Hub: Fake News Dataset	Feature Extraction techniques a) Count Vector b) TF-IDF c) Word2Vec Classifiers 1. SVM 2. ANNs 3. LSTMs 4. Logistic random forest	Multiple Accuracies With Different techniques a) Count vectorizer 1. 93% 2. 94 % 3. 94% 4. 94 % 5. 87% b) TFIDF 1. 94% 2. 93% 3. 93% 4. 94% 5. 87% c) Word2Vec 1. 91.1% 2. 93.0% 3. 92.2% 4. 91.3% 5. 88.6%
10	Razno [45]	2019	Computational Linguistics and	Yelp academic dataset review	A basic methodology and use of different classifiers do not give the specific name	No results reported

			Intelligent Systems			
11	E. V [46]	2020	Информационно-управляющие системы	1. Stanford Artificial Intelligence : Sentiment Analysis 2. Kaggle: Amazon Reviews for Sentiment Analysis	1. Logistic Regression 2. Count Vectorizer 3. Doc2Vec 4. Gradient Boosting	1. 87% 2. 86% 3. 70% 4. 72%

Chapter 4: Proposed Methodology

For the automatic identification of text in a given dataset, this study presents a categorization of complaints. We'll start with a look at natural language processing (NLP), and then move on to classifying data. To improve one's understanding of data categorization, one may do a literature review.

4.1 Construct a conceptual framework

4.1.1 Problem tree

A first problem tree was erected to have a clearer understanding of the issue at hand. Dataset, machine learning method, and assessment are all branches of the tree shown in Figure 4.1.

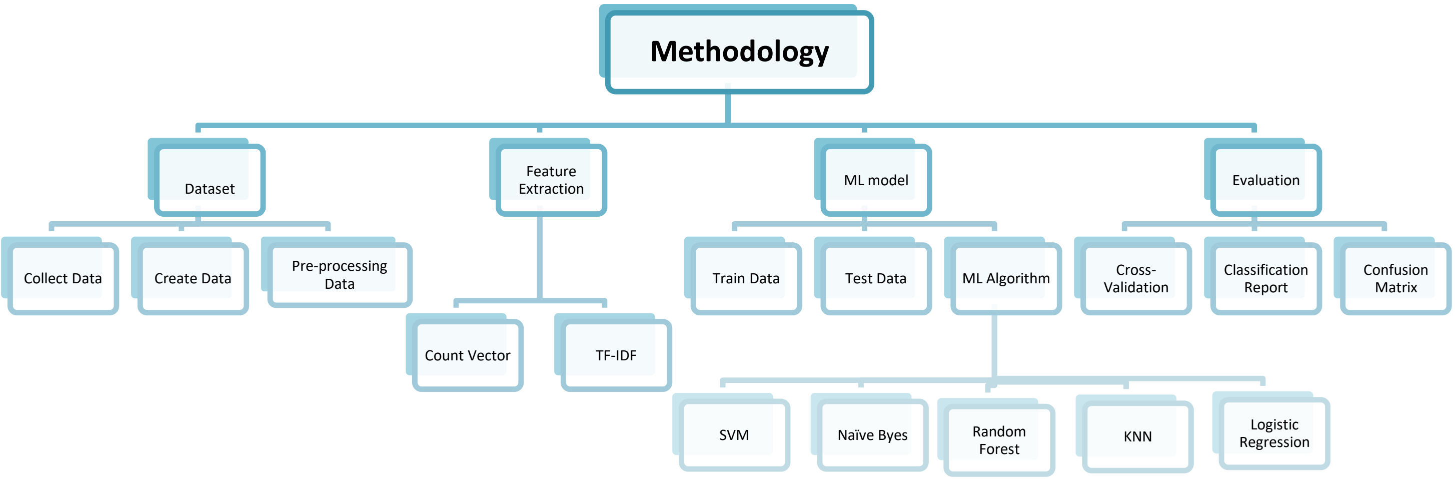


Figure 4.1 Problem tree

4.2 Dataset

Data collection, data creation, and data pre-processing are the three sub-branches of the dataset Figure 4.2. Collecting data is indeed the process of obtaining datasets that are publicly accessible. Created data is not publicly accessible, and is used for a specified reason alone; it's a private endeavor. Pre-processing data is used to improve the algorithm's understanding of the data samples.

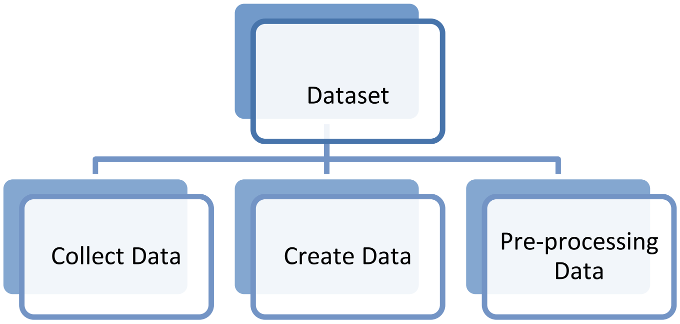


Figure 4.2 Problem tree of the dataset branch.

By integrating publicly accessible datasets with custom-generated data samples, the dataset was constructed from scratch. The dataset that is accessible to the public was compiled using data from two distinct systems. A multi-level classification dataset is created by labeling the text samples. Raw data is transformed into a comprehensible format via the use of data pre-processing. Raw data is typically incoherent and includes symbols or phrases that are prone to create mistakes. Section 4.3 outlines the steps involved in this approach.

4.2.1 Local Data Set I

Ten separate divisions/classes, each with a different domain to cater to in dataset I, are included in this dataset. These divisions have different kinds of complaints such as on Scholarships, academics, Accreditation, Information & Technology, Attestation, Sports, Research and Development, Equivalency of different degrees and domains, Quality Assurance Agency & Quality Assurance Division. Each sample includes the question about the aforementioned divisions. We received 10,002 closed complaints from these divisions, which were kept on the organization's website. There are several overlaps between these criticisms. There are a variety of ways to report a problem (samples). The amount of grievances filed by students in each group varies.

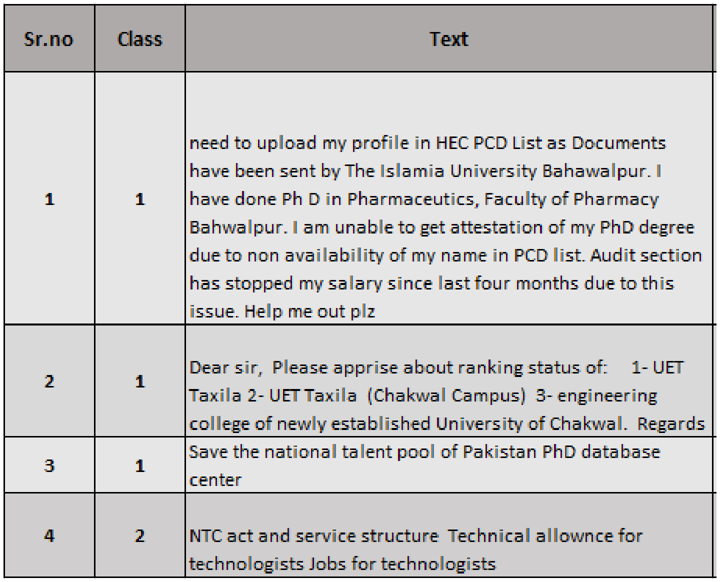
4.2.2 Consumer Complaints Data Set II

You may get the worldwide Consumer Complaints data set II from Kaggle as well as on Github. There are 1, 62,421 complaints in the dataset. There are five classifications in this dataset, and each class has a distinct amount of grievances. There are courses on retail banking, credit card, and credit reporting, mortgage, or debt collection in the bank department.

4.2.3 Initial Pre-processing

Data from various languages are carefully cleaned. On average, there are more than two times as many unique events in the original collection as there are unique ones. Recurring occurrences, such as repeated complaints, contribute to an excessive number of duplications. If the department and the complaint description are the same, but the date or time is different then the event has been deemed a duplicate. There are arguments in favor of and against deleting these reoccurring occurrences. Both types of argument’s points of view will be taken into account.

There is a strong case for removing duplicates because of the potential for overfitting. This is likely to be seen as a significant link during training if the training set includes many duplicate occurrences. For less frequent occurrences, the model can accurately reflect the occurrence of the same event several times. The set's high duplication count, on the other hand, is an accurate reflection of the situation in actual life. Complaints about the same things happen more often than if they were a one-time occurrence. Duplicates were eliminated to get the best possible performance. However, it isn't clear whether this is a superior strategy in reality. Considering this is a whole other area of study, it will not be addressed here. Figure 4.3Figure 4.3: Showing the architecture of our dataset in Pandas



Sr.no	Class	Text
1	1	need to upload my profile in HEC PCD List as Documents have been sent by The Islamia University Bahawalpur. I have done Ph D in Pharmaceutics, Faculty of Pharmacy Bahwalpur. I am unable to get attestation of my PhD degree due to non availability of my name in PCD list. Audit section has stopped my salary since last four months due to this issue. Help me out plz
2	1	Dear sir, Please apprise about ranking status of: 1- UET Taxila 2- UET Taxila (Chakwal Campus) 3- engineering college of newly established University of Chakwal. Regards
3	1	Save the national talent pool of Pakistan PhD database center
4	2	NTC act and service structure Technical allownce for technologists Jobs for technologists

Figure 4.3: Showing the architecture of our dataset in Pandas

4.2.4 Data Analysis

To proceed, we needed to examine the data. If the dataset is not balanced, then this step is necessary. There are unequal numbers of data samples in each class (as seen in Figure 4.4 & Figure 4.5), which means the dataset is unbalanced. These graphs are made after doing initial pre-processing.

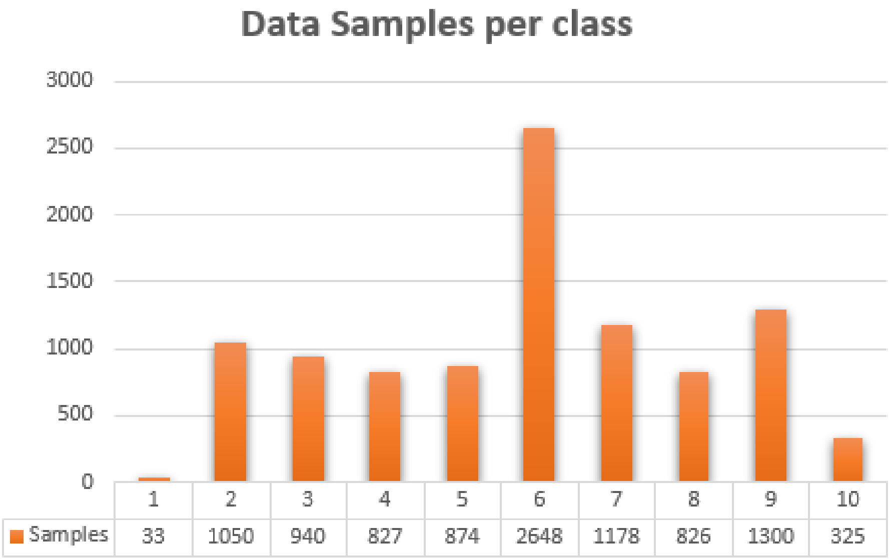


Figure 4.4: Dataset I has a total of n samples per class.

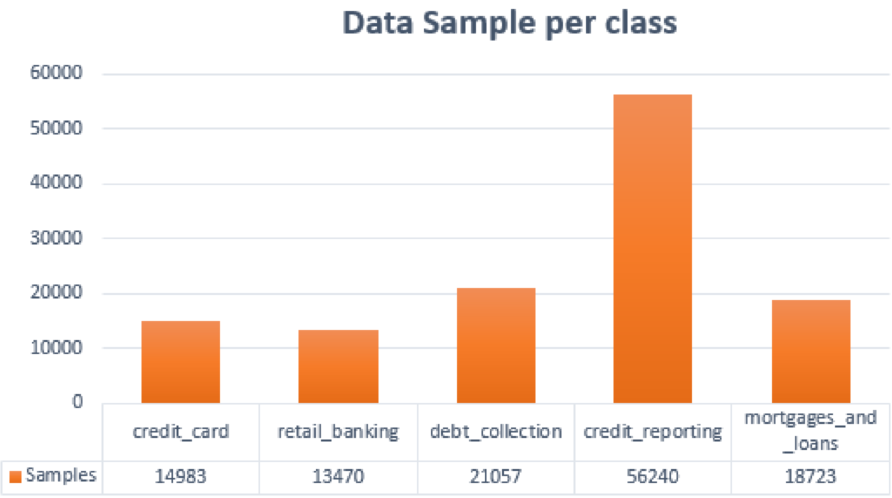


Figure 4.5: Dataset II has a total of n samples per class.

4.3 Pre-processing data

Before categorization, pre-processing is a necessary step. Algorithms can't make accurate predictions if they don't grasp what they're looking at. Cleaning text data is the first step in preprocessing. Following the block diagram Figure 4.6 are the stages of the pre-processing.

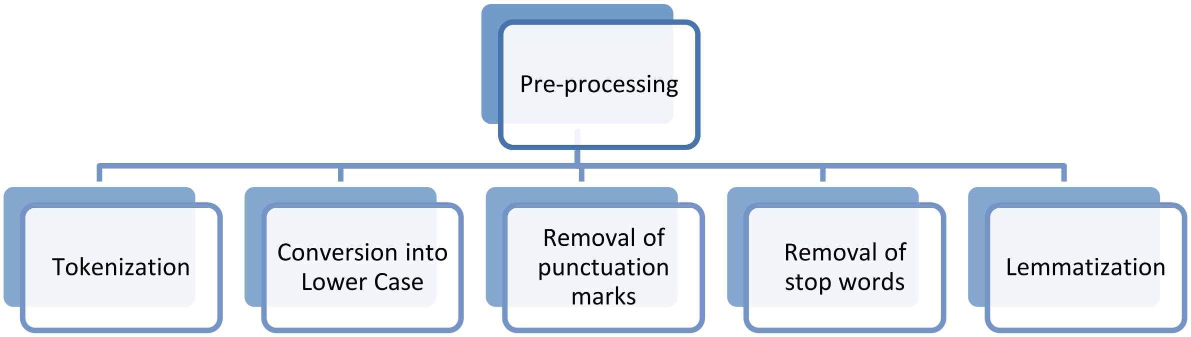


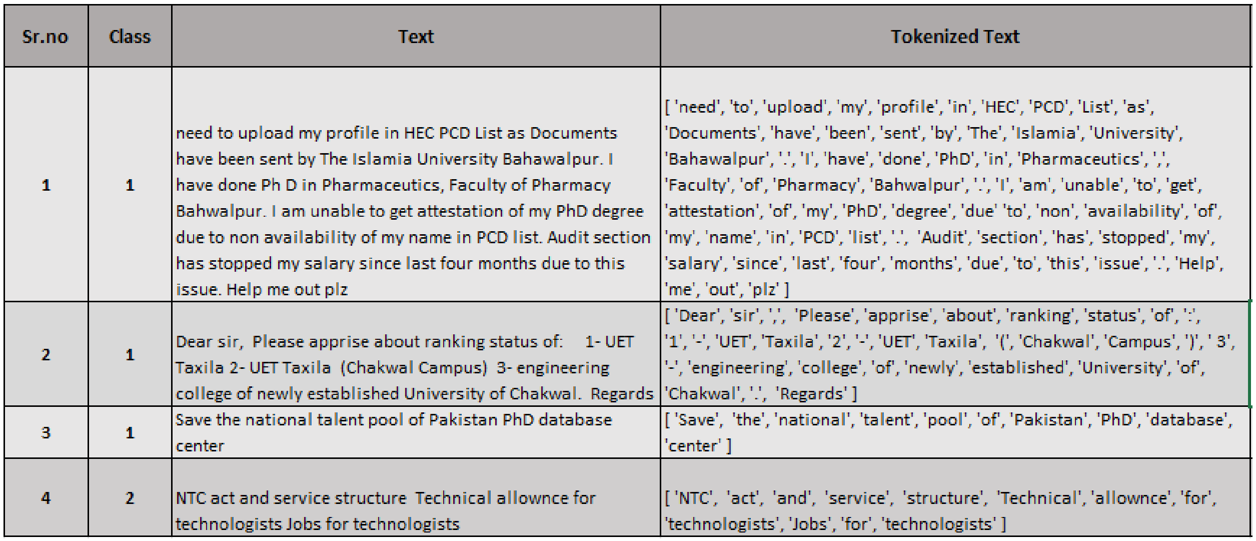
Figure 4.6: Demonstrate stages of pre-processing

Cleaning the data consists of:

- Tokenization.
- Converting text to lower case.
- Removing stop words.
- Removing punctuation and unwanted symbols and characters.
- Lemmatization

4.3.1 Tokenization

We tokenized the dataset using the NLTK package after eliminating all punctuation and unnecessary symbols from it. As mentioned in Section 2.3.1, we broke the statement down into individual words. As can be seen in Figure 4.7, the dataset has been tokenized.

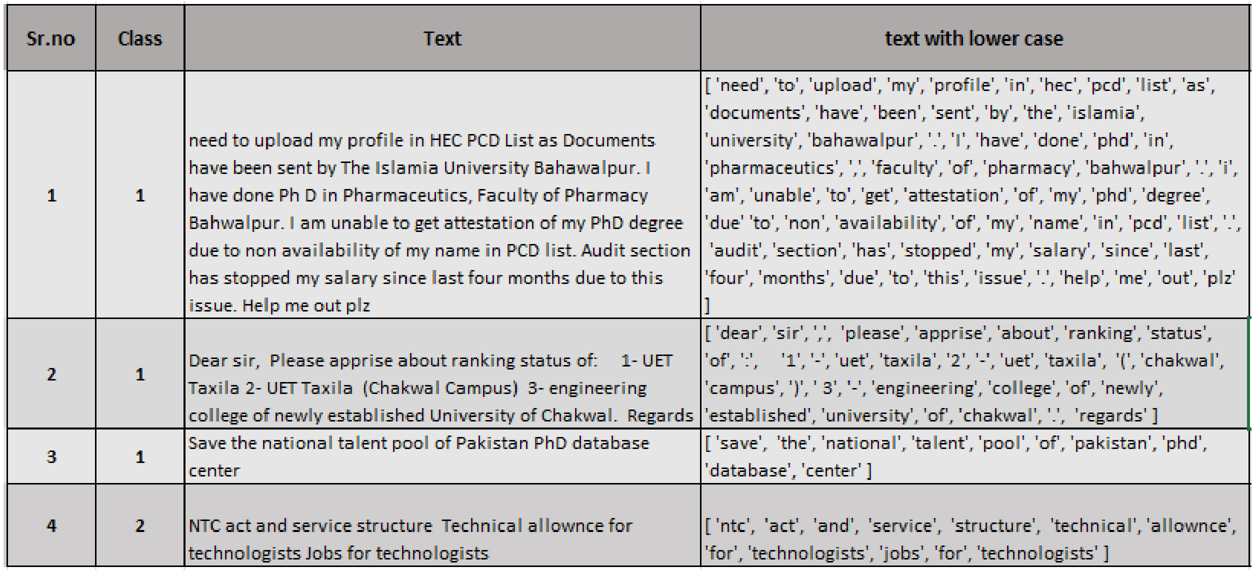


Sr.no	Class	Text	Tokenized Text
1	1	need to upload my profile in HEC PCD List as Documents have been sent by The Islamia University Bahawalpur. I have done Ph D in Pharmaceutics, Faculty of Pharmacy Bahwalpur. I am unable to get attestation of my PhD degree due to non availability of my name in PCD list. Audit section has stopped my salary since last four months due to this issue. Help me out plz	['need', 'to', 'upload', 'my', 'profile', 'in', 'HEC', 'PCD', 'List', 'as', 'Documents', 'have', 'been', 'sent', 'by', 'The', 'Islamia', 'University', 'Bahawalpur', '.', 'I', 'have', 'done', 'PhD', 'in', 'Pharmaceutics', ',', 'Faculty', 'of', 'Pharmacy', 'Bahwalpur', '.', 'I', 'am', 'unable', 'to', 'get', 'attestation', 'of', 'my', 'PhD', 'degree', 'due' 'to', 'non', 'availability', 'of', 'my', 'name', 'in', 'PCD', 'list', '.', 'Audit', 'section', 'has', 'stopped', 'my', 'salary', 'since', 'last', 'four', 'months', 'due', 'to', 'this', 'issue', '.', 'Help', 'me', 'out', 'plz']
2	1	Dear sir, Please apprise about ranking status of: 1- UET Taxila 2- UET Taxila (Chakwal Campus) 3- engineering college of newly established University of Chakwal. Regards	['Dear', 'sir', ',', 'Please', 'apprise', 'about', 'ranking', 'status', 'of', ':', '1', '-', 'UET', 'Taxila', '2', '-', 'UET', 'Taxila', '(', 'Chakwal', 'Campus', ')', ' 3', '-', 'engineering', 'college', 'of', 'newly', 'established', 'University', 'of', 'Chakwal', '.', 'Regards']
3	1	Save the national talent pool of Pakistan PhD database center	['Save', 'the', 'national', 'talent', 'pool', 'of', 'Pakistan', 'PhD', 'database', 'center']
4	2	NTC act and service structure Technical allownce for technologists Jobs for technologists	['NTC', 'act', 'and', 'service', 'structure', 'Technical', 'allownce', 'for', 'technologists', 'Jobs', 'for', 'technologists']

Figure 4.7: Before and after the tokenization of the data frame

4.3.2 Conversion of text into lower case

There are several ways to mark the beginning and conclusion of a phrase or stress a certain term in a text document. All letters must be changed to lower case before any further processing can begin. Samples of text before and after eliminating capital letters are shown in Figure 4.8. The textual data is easier for the computer to grasp if all letters are lowercase [25].

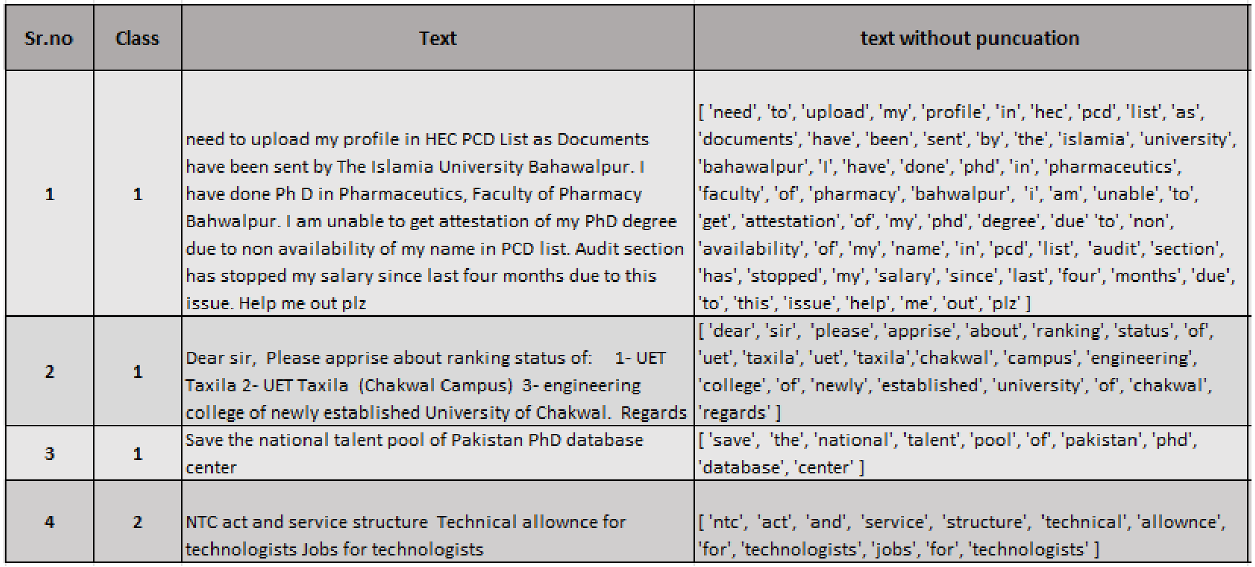


Sr.no	Class	Text	text with lower case
1	1	need to upload my profile in HEC PCD List as Documents have been sent by The Islamia University Bahawalpur. I have done Ph D in Pharmaceutics, Faculty of Pharmacy Bahwalpur. I am unable to get attestation of my PhD degree due to non availability of my name in PCD list. Audit section has stopped my salary since last four months due to this issue. Help me out plz	['need', 'to', 'upload', 'my', 'profile', 'in', 'hec', 'pcd', 'list', 'as', 'documents', 'have', 'been', 'sent', 'by', 'the', 'islamia', 'university', 'bahawalpur', '.', 'I', 'have', 'done', 'phd', 'in', 'pharmaceutics', ',', 'faculty', 'of', 'pharmacy', 'bahwalpur', '.', 'i', 'am', 'unable', 'to', 'get', 'attestation', 'of', 'my', 'phd', 'degree', 'due' 'to', 'non', 'availability', 'of', 'my', 'name', 'in', 'pcd', 'list', '.', 'audit', 'section', 'has', 'stopped', 'my', 'salary', 'since', 'last', 'four', 'months', 'due', 'to', 'this', 'issue', '.', 'help', 'me', 'out', 'plz']
2	1	Dear sir, Please apprise about ranking status of: 1- UET Taxila 2- UET Taxila (Chakwal Campus) 3- engineering college of newly established University of Chakwal. Regards	['dear', 'sir', ',', 'please', 'apprise', 'about', 'ranking', 'status', 'of', ':', '1', '-', 'uet', 'taxila', '2', '-', 'uet', 'taxila', '(', 'chakwal', 'campus', ')', ' 3', '-', 'engineering', 'college', 'of', 'newly', 'established', 'university', 'of', 'chakwal', '.', 'regards']
3	1	Save the national talent pool of Pakistan PhD database center	['save', 'the', 'national', 'talent', 'pool', 'of', 'pakistan', 'phd', 'database', 'center']
4	2	NTC act and service structure Technical allownce for technologists Jobs for technologists	['ntc', 'act', 'and', 'service', 'structure', 'technical', 'allownce', 'for', 'technologists', 'jobs', 'for', 'technologists']

Figure 4.8: Before and after transforming all letters to lowercase in the data frame.

4.3.3 Removal of stop words and punctuation marks

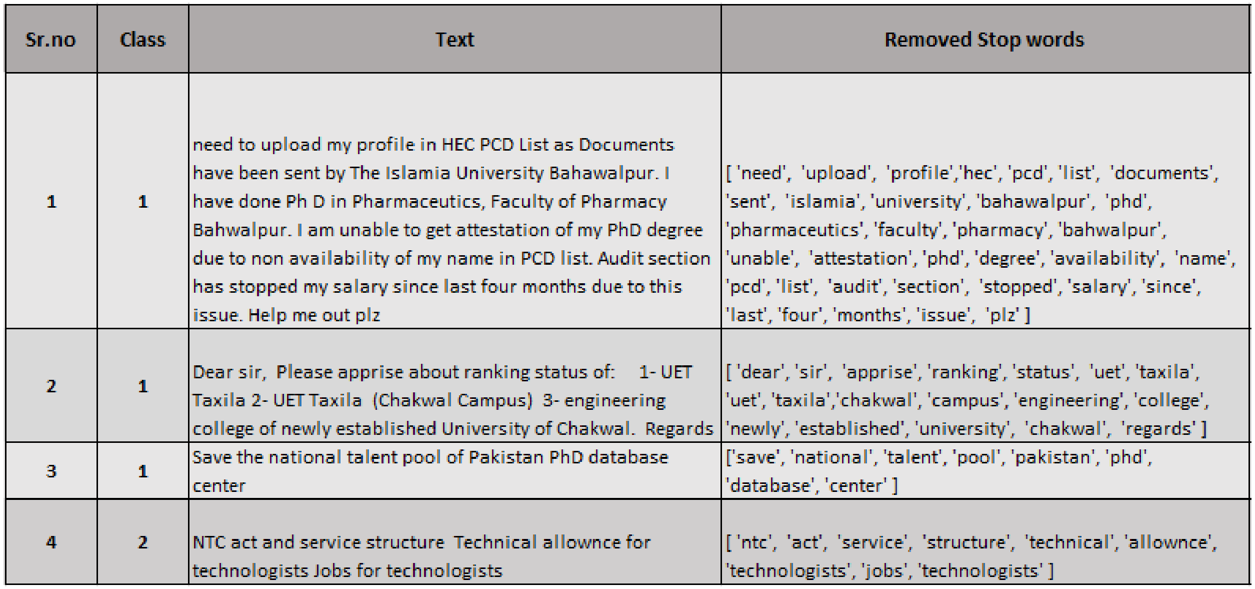
As a second benefit, deleting punctuation ensures that all words are processed equally. The dataset was thoroughly cleaned to eliminate any unnecessary punctuation and symbols. Because some of our data come from a publicly accessible dataset, we need to eliminate symbols such as [!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~] that is often seen in that dataset and may include human mistake Figure 4.9 illustrates the outcome of this stage.



Sr.no	Class	Text	text without puncuation
1	1	need to upload my profile in HEC PCD List as Documents have been sent by The Islamia University Bahawalpur. I have done Ph D in Pharmaceutics, Faculty of Pharmacy Bahwalpur. I am unable to get attestation of my PhD degree due to non availability of my name in PCD list. Audit section has stopped my salary since last four months due to this issue. Help me out plz	['need', 'to', 'upload', 'my', 'profile', 'in', 'hec', 'pcd', 'list', 'as', 'documents', 'have', 'been', 'sent', 'by', 'the', 'islamia', 'university', 'bahawalpur', 'I', 'have', 'done', 'phd', 'in', 'pharmaceutics', 'faculty', 'of', 'pharmacy', 'bahwalpur', 'i', 'am', 'unable', 'to', 'get', 'attestation', 'of', 'my', 'phd', 'degree', 'due' 'to', 'non', 'availability', 'of', 'my', 'name', 'in', 'pcd', 'list', 'audit', 'section', 'has', 'stopped', 'my', 'salary', 'since', 'last', 'four', 'months', 'due', 'to', 'this', 'issue', 'help', 'me', 'out', 'plz']
2	1	Dear sir, Please apprise about ranking status of: 1- UET Taxila 2- UET Taxila (Chakwal Campus) 3- engineering college of newly established University of Chakwal. Regards	['dear', 'sir', 'please', 'apprise', 'about', 'ranking', 'status', 'of', 'uet', 'taxila', 'uet', 'taxila','chakwal', 'campus', 'engineering', 'college', 'of', 'newly', 'established', 'university', 'of', 'chakwal', 'regards']
3	1	Save the national talent pool of Pakistan PhD database center	['save', 'the', 'national', 'talent', 'pool', 'of', 'pakistan', 'phd', 'database', 'center']
4	2	NTC act and service structure Technical allownce for technologists Jobs for technologists	['ntc', 'act', 'and', 'service', 'structure', 'technical', 'allownce', 'for', 'technologists', 'jobs', 'for', 'technologists']

Figure 4.9: After eliminating punctuation marks from the data

Stop words are terms that are often seen in written materials, such as: ("the," "is," "and," "in"). Generally speaking, these terms are of little use when it comes to categorizing text data because of their high frequency [47]. To create a place for less commonly used words, which are more relevant to the classification job, stop words may be filtered out Shown in Figure 4.10 is an example of a frame lacking stop words.

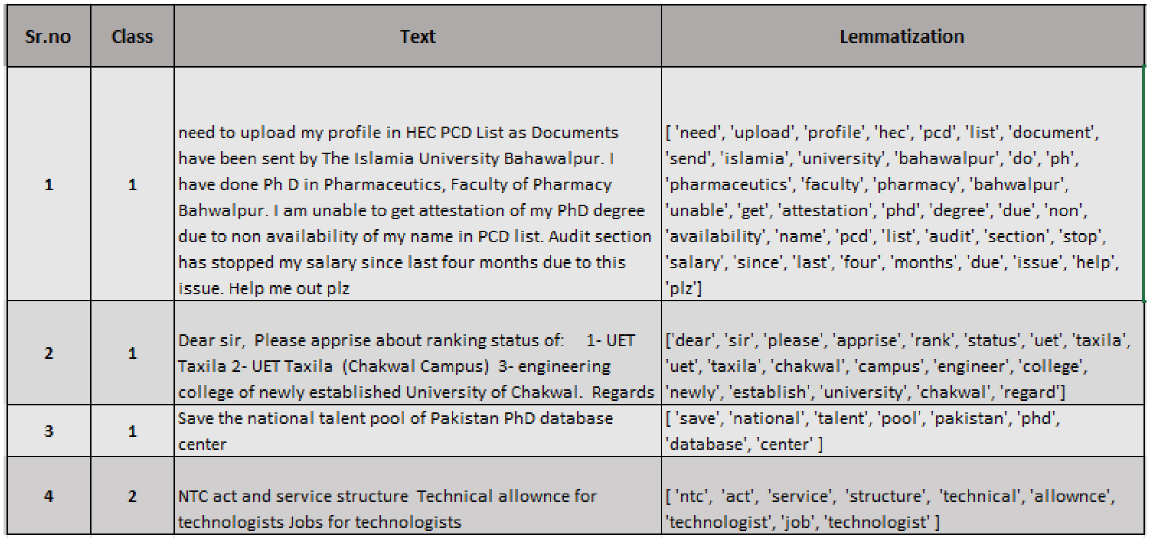


Sr.no	Class	Text	Removed Stop words
1	1	need to upload my profile in HEC PCD List as Documents have been sent by The Islamia University Bahawalpur. I have done Ph D in Pharmaceutics, Faculty of Pharmacy Bahwalpur. I am unable to get attestation of my PhD degree due to non availability of my name in PCD list. Audit section has stopped my salary since last four months due to this issue. Help me out plz	['need', 'upload', 'profile','hec', 'pcd', 'list', 'documents', 'sent', 'islamia', 'university', 'bahawalpur', 'phd', 'pharmaceutics', 'faculty', 'pharmacy', 'bahwalpur', 'unable', 'attestation', 'phd', 'degree', 'availability', 'name', 'pcd', 'list', 'audit', 'section', 'stopped', 'salary', 'since', 'last', 'four', 'months', 'issue', 'plz']
2	1	Dear sir, Please apprise about ranking status of: 1- UET Taxila 2- UET Taxila (Chakwal Campus) 3- engineering college of newly established University of Chakwal. Regards	['dear', 'sir', 'apprise', 'ranking', 'status', 'uet', 'taxila', 'uet', 'taxila','chakwal', 'campus', 'engineering', 'college', 'newly', 'established', 'university', 'chakwal', 'regards']
3	1	Save the national talent pool of Pakistan PhD database center	['save', 'national', 'talent', 'pool', 'pakistan', 'phd', 'database', 'center']
4	2	NTC act and service structure Technical allownce for technologists Jobs for technologists	['ntc', 'act', 'service', 'structure', 'technical', 'allownce', 'technologists', 'jobs', 'technologists']

Figure 4.10: Removed stop words from the data frame

4.3.4 Lemmatization

Lemmatization is an example of natural language processing. Stemming is the most widely used form of language processing, however, there are others. One way to stem words is to turn them into their standard form. For example, all plural terms are shortened to their single equivalents. Alternatively, all past tense words might be changed to present tense terms. Text reduction and adding synonyms to the token set are examples of alternative ways of language processing.



Sr.no	Class	Text	Lemmatization
1	1	need to upload my profile in HEC PCD List as Documents have been sent by The Islamia University Bahawalpur. I have done Ph D in Pharmaceutics, Faculty of Pharmacy Bahwalpur. I am unable to get attestation of my PhD degree due to non availability of my name in PCD list. Audit section has stopped my salary since last four months due to this issue. Help me out plz	['need', 'upload', 'profile', 'hec', 'pcd', 'list', 'document', 'send', 'islamia', 'university', 'bahawalpur', 'do', 'ph', 'pharmaceutics', 'faculty', 'pharmacy', 'bahwalpur', 'unable', 'get', 'attestation', 'phd', 'degree', 'due', 'non', 'availability', 'name', 'pcd', 'list', 'audit', 'section', 'stop', 'salary', 'since', 'last', 'four', 'months', 'due', 'issue', 'help', 'plz']
2	1	Dear sir, Please apprise about ranking status of: 1- UET Taxila 2- UET Taxila (Chakwal Campus) 3- engineering college of newly established University of Chakwal. Regards	['dear', 'sir', 'please', 'apprise', 'rank', 'status', 'uet', 'taxila', 'uet', 'taxila', 'chakwal', 'campus', 'engineer', 'college', 'newly', 'establish', 'university', 'chakwal', 'regard']
3	1	Save the national talent pool of Pakistan PhD database center	['save', 'national', 'talent', 'pool', 'pakistan', 'phd', 'database', 'center']
4	2	NTC act and service structure Technical allownce for technologists Jobs for technologists	['ntc', 'act', 'service', 'structure', 'technical', 'allownce', 'technologist', 'job', 'technologist']

Figure 4.11: After Lemmatization on the dataset.

4.4 Feature Extraction Techniques

Transforming tokens into words by assigning numbers to them is what it means to create a vector representation. The textual representation of the word cannot be used by a classifier since it can only deal with numbers. Count vectorization and TF-IDF vectorization are two of the many vectorization techniques available to you below in Figure 4.12. Because the hashing vector didn't provide excellent results, we utilize two vectorization methods.

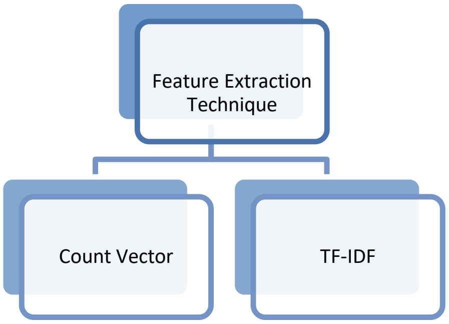


Figure 4.12: Feature Extraction Techniques

4.4.1 Count Vector

Token value is calculated using a count vectorizer, which takes into account the token's frequency of occurrence. This phenomenon is known as occurrence frequency. It maintains track of the number of times each token appears in a given piece of writing. A token's worth increases as it happens more often, as mentioned in subject 2.2.3. The count vector model is shown in Table 4.1.

Table 4.1: Model of Count Vector

	W1	W2	W3	W4	W5	W6	….
Document 1	1	1	2	0	1	2	….
Document 2	2	0	2	1	2	3	….
Document 3	1	0	1	2	2	0	….
Document 4	2	1	0	3	2	4	….

w1, w2, and w3 are the tokens in the document (sample), and the numbers denote the number of times they appear.

4.4.2 TF-IDF

TF-IDF goes a step farther than the others. While the word frequency is taken into account, it also takes into consideration the specificity of the token used. Word frequency - document term frequency is a term for this combination. For instance, the word 'the' is used frequently in all texts. This token will be given a low value by a TF-IDF vectorizer. A term with a high frequency in a small number of texts but a low frequency in other texts will have a greater value.

Document term frequency (IDF) reduces the number of times a word occurs in a document based on the frequency with which it appears in other documents, while term frequency (TF) does the opposite [40]. This is shown in Table 4.2 as a TF-IDF Vectorizer model.

$$TF(Word) = \frac{\text{No. of times same tokens in a doc}}{\text{Total no. of tokens in an in a doc.}} \tag{4.1}$$

$$IDF(Word) = \frac{\text{Total no. of doc. in a dataset}}{\text{No. of doc. with the same token in it.}} \tag{4.2}$$

Table 4.2: Model of TF-IDF Vectorizer

	W1	W2	W3	W4	W5	W6	….
Document 1	0.9	0.4	0.1	0.14	0.6	0.8	….
Document 2	0.2	0.2	0.1	0.4	0.4	0.4	….
Document 3	0.1	0.1	0.4	0.9	0.8	0.1	….
Document 4	0.2	0.1	0.9	0.4	0.4	0.1	….

4.5 Classification

Some machine learning methods are employed here, but first, we split the data and then use different classifiers on each piece of data.

These attributes of the training dataset are learned by utilizing a supervised machine learning technique, in which a model learns using labeled training data. A classification is made by the ML algorithm when fresh data is fed into it, using what it has already learned.

4.5.1 Split Data

Splitting the data set into two groups, training, and testing requires dividing the data into two groups. We partitioned the dataset after feature extraction. To train and test the model, we employed an 80/20 ratio, which means that 80 percent of the data is used for training and the remaining 20 percent is used for testing. Figure 4.13 depicts the number of observations per class after the data were divided into groups.

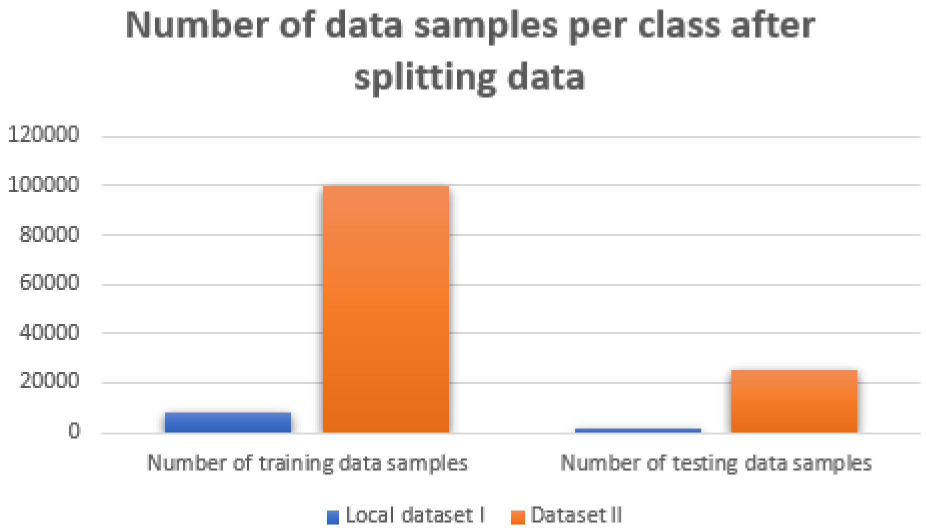


Figure 4.13: Division of data samples in training and testing

4.5.2 Machine learning model

In computer science, classifiers are algorithms that can foretell the labels of data that have not yet been seen. There are a variety of classifiers that may be used for a variety of purposes. A good choice is essential to the program's performance. The form and size of a dataset have a major role in the decision. When calculating a classifier's error in classification, a classifier's bias and variance are added together.

A classifier's bias is minimal if the model accurately predicts the distribution of the data. The size of the training set has no bearing on this. When a classifier has a large bias, it may underfit, which indicates that it overlooks key relationships between features.

Training sets with low variability are more likely to provide a low variance classifier. Training set size is an important consideration. Overfitting may occur when the classifier models randomness in the dataset rather than the underlying relations.

The volatility in a tiny dataset might be rather substantial. Naive Bayes, a classifier capable of dealing with large variation, is a viable option in this scenario. By examining each attribute independently, this classifier ignores the wide range of possible outcomes. The form of the data and the breadth of the issue become more relevant as the dataset grows. Classifying text is the crux of our issue. Sebastiani [48] looked at a variety of text categorization classifiers and discovered the following:

1. Methods such as classifiers built on the foundation of support vector machines, examples of classifiers, and regression models all perform admirably well.
2. Two of the most popular approaches for analyzing data: are neural networks and linear classifiers.
3. Batch linear classifiers and Nave Bayes classifiers are the poorest of the learning-based classifiers in terms of performance.
4. There is not enough evidence to draw any conclusions on the effectiveness of decision trees.

As seen in the figure above, we employ a variety of classifiers. The findings of these classifiers are different. In addition, the complexity of each algorithm varies. Algorithms like the following are employed:

- SVM
- Random Forest
- Logistic Regression
- Mutlinomial Naïve Bayes
- KNN

4.6 Evaluate model

An assessment method known as a 10-fold validation set is used. It is feasible to prevent overfitting the model by randomly splitting the dataset into 10 folds, as discussed in section 2.5.1. The larger the dataset, the more accurate a Machine Learning model will be [23]. Because of this, an efficient method is to investigate the model's behavior as a function of number the of training samples [23]. To minimize overfitting, a learning curve is an effective way to ensure that the model can handle additional training data. A classification report, as detailed in section 2.5.2, is useful for a better comprehension of the model's performance. When assessing a model's precision, recall, and f1-score, all three are taken into account.

Chapter 5: Experiment & Results

As the goal of this study is to propose an explanation framework, to test the effectiveness of proposed ML models, we use the same experimental settings on the Local dataset I and the Consumer complaints dataset II.

5.1 Local Dataset1

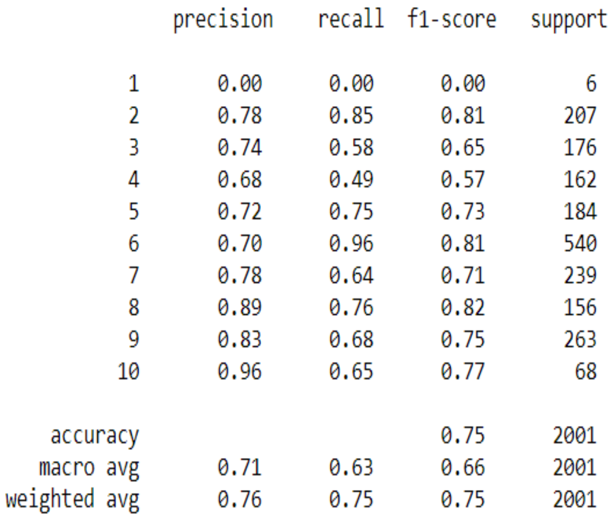
As we already know that in our local dataset, we have 10 classes and 10,000 samples in the dataset. Which is further divided into 80% training and 20% testing. We apply two different techniques of feature extraction and then we apply ML models to each of the techniques.

5.1.1 Count Vector

After applying the Count Vector technique in feature extraction and employing 5 different ML models. We acquire different results.

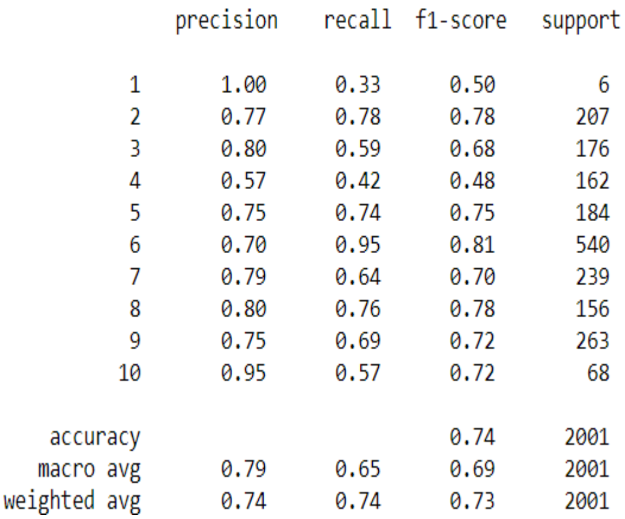
5.1.1.1 Classification reports

With the help of a classification report, we can easily know what is the precision, recall f1-score, and support(data samples) in each class after applying the count vector. Also, find out overall accuracy. Following are the testing classification reports produced by five different ML models.



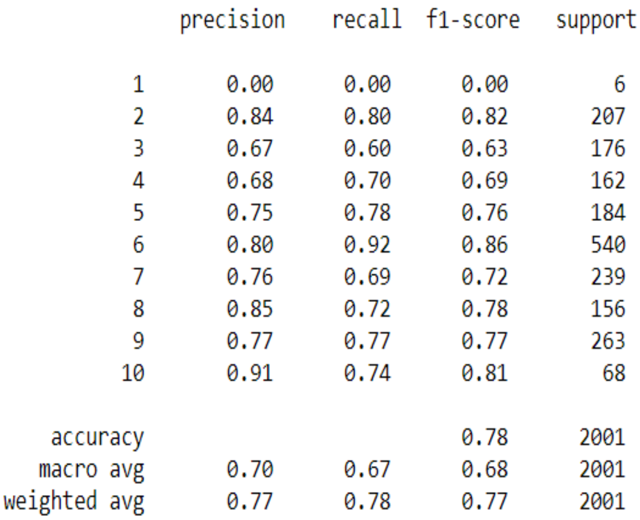
```
              precision    recall  f1-score   support

           1       0.00      0.00      0.00         6
           2       0.78      0.85      0.81       207
           3       0.74      0.58      0.65       176
           4       0.68      0.49      0.57       162
           5       0.72      0.75      0.73       184
           6       0.70      0.96      0.81       540
           7       0.78      0.64      0.71       239
           8       0.89      0.76      0.82       156
           9       0.83      0.68      0.75       263
          10       0.96      0.65      0.77        68

    accuracy                           0.75      2001
   macro avg       0.71      0.63      0.66      2001
weighted avg       0.76      0.75      0.75      2001
```

(a) Random Forest



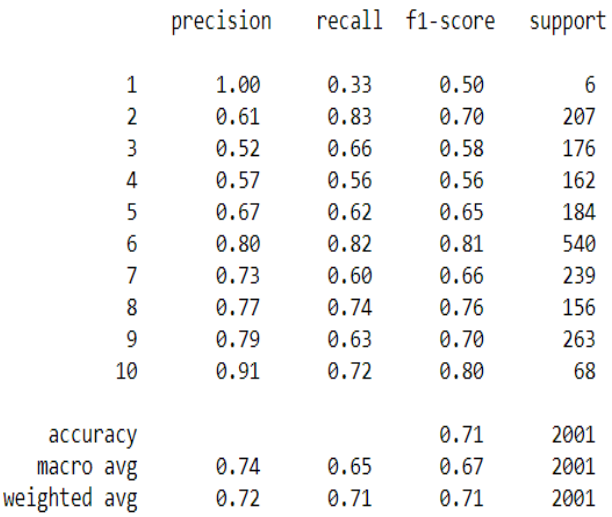
```
              precision    recall  f1-score   support

           1       1.00      0.33      0.50         6
           2       0.77      0.78      0.78       207
           3       0.80      0.59      0.68       176
           4       0.57      0.42      0.48       162
           5       0.75      0.74      0.75       184
           6       0.70      0.95      0.81       540
           7       0.79      0.64      0.70       239
           8       0.80      0.76      0.78       156
           9       0.75      0.69      0.72       263
          10       0.95      0.57      0.72        68

    accuracy                           0.74      2001
   macro avg       0.79      0.65      0.69      2001
weighted avg       0.74      0.74      0.73      2001
```

(b) Mutlinomial Naïve Bayes



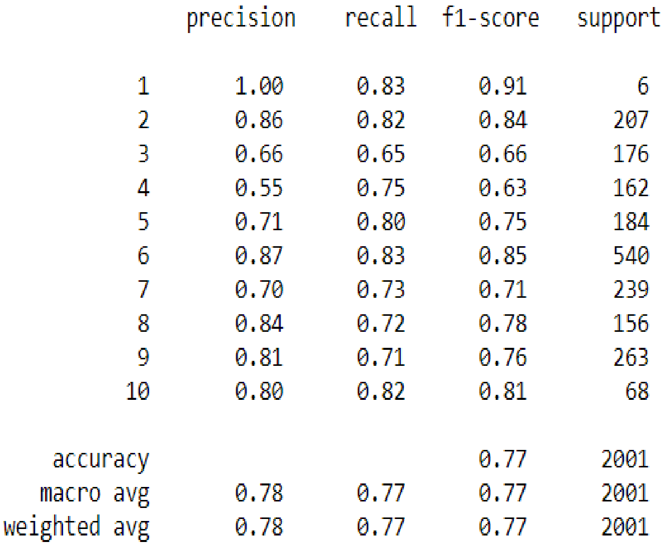
```
              precision    recall  f1-score   support

           1       0.00      0.00      0.00         6
           2       0.84      0.80      0.82       207
           3       0.67      0.60      0.63       176
           4       0.68      0.70      0.69       162
           5       0.75      0.78      0.76       184
           6       0.80      0.92      0.86       540
           7       0.76      0.69      0.72       239
           8       0.85      0.72      0.78       156
           9       0.77      0.77      0.77       263
          10       0.91      0.74      0.81        68

    accuracy                           0.78      2001
   macro avg       0.70      0.67      0.68      2001
weighted avg       0.77      0.78      0.77      2001
```

(c) Logistic Regression



```
              precision    recall  f1-score   support

           1       1.00      0.33      0.50         6
           2       0.61      0.83      0.70       207
           3       0.52      0.66      0.58       176
           4       0.57      0.56      0.56       162
           5       0.67      0.62      0.65       184
           6       0.80      0.82      0.81       540
           7       0.73      0.60      0.66       239
           8       0.77      0.74      0.76       156
           9       0.79      0.63      0.70       263
          10       0.91      0.72      0.80        68

    accuracy                           0.71      2001
   macro avg       0.74      0.65      0.67      2001
weighted avg       0.72      0.71      0.71      2001
```

(d) KNN



```
              precision    recall  f1-score   support

           1       1.00      0.83      0.91         6
           2       0.86      0.82      0.84       207
           3       0.66      0.65      0.66       176
           4       0.55      0.75      0.63       162
           5       0.71      0.80      0.75       184
           6       0.87      0.83      0.85       540
           7       0.70      0.73      0.71       239
           8       0.84      0.72      0.78       156
           9       0.81      0.71      0.76       263
          10       0.80      0.82      0.81        68

    accuracy                           0.77      2001
   macro avg       0.78      0.77      0.77      2001
weighted avg       0.78      0.77      0.77      2001
```

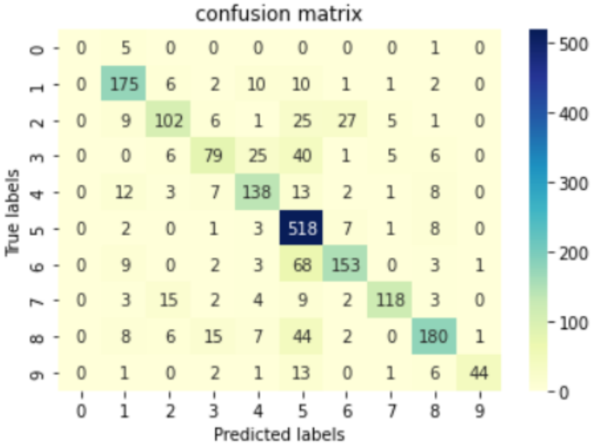
(e) SVM

Figure 5.1: Classification reports of Local dataset I from various ML models by Count Vector technique

From the above Classification reports, we can see the accuracies of different classes from the count vector technique with various classifiers. In fig.(c) Logistic Regression gives good results in class 6 that’s why overall Logistic Regression has better accuracy than others.

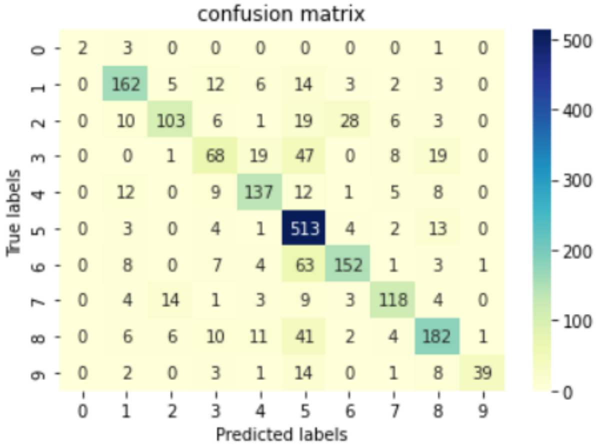
5.1.1.2 Confusion Matrix

With the help of the confusion matrix, we will know each class data more accurately. This helps in evaluating the performance of a classification model. The matrix compares the actual true value with the classified value given by the Ml model. The following figure shows the graphs of different ML models in which columns have 0-9 (I to 10) true class whereas rows have 0-9 (1-10) predicted class.



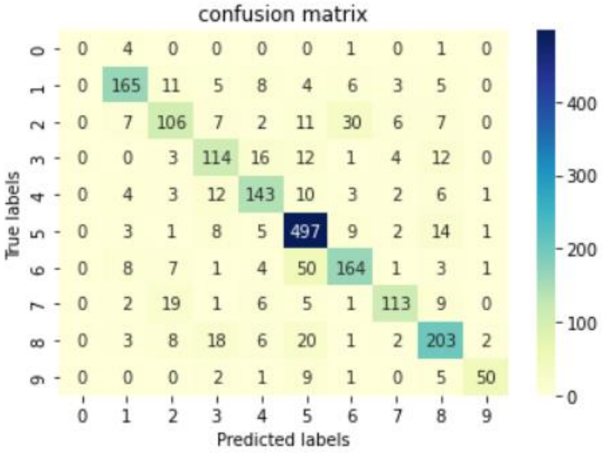
	0	1	2	3	4	5	6	7	8	9
0	0	5	0	0	0	0	0	0	1	0
1	0	175	6	2	10	10	1	1	2	0
2	0	9	102	6	1	25	27	5	1	0
3	0	0	6	79	25	40	1	5	6	0
4	0	12	3	7	138	13	2	1	8	0
5	0	2	0	1	3	518	7	1	8	0
6	0	9	0	2	3	68	153	0	3	1
7	0	3	15	2	4	9	2	118	3	0
8	0	8	6	15	7	44	2	0	180	1
9	0	1	0	2	1	13	0	1	6	44

(a) Random Forest



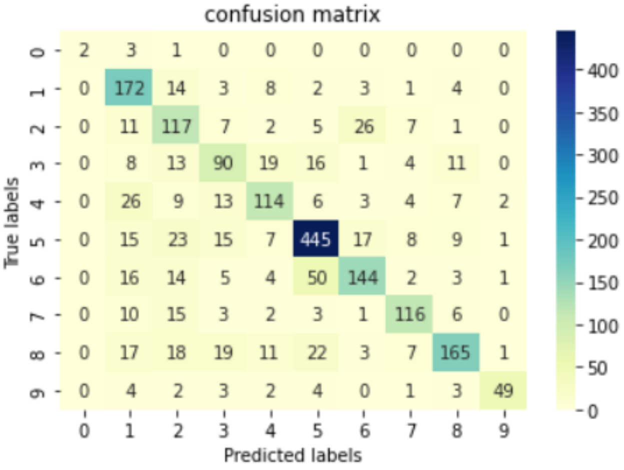
	0	1	2	3	4	5	6	7	8	9
0	2	3	0	0	0	0	0	0	1	0
1	0	162	5	12	6	14	3	2	3	0
2	0	10	103	6	1	19	28	6	3	0
3	0	0	1	68	19	47	0	8	19	0
4	0	12	0	9	137	12	1	5	8	0
5	0	3	0	4	1	513	4	2	13	0
6	0	8	0	7	4	63	152	1	3	1
7	0	4	14	1	3	9	3	118	4	0
8	0	6	6	10	11	41	2	4	182	1
9	0	2	0	3	1	14	0	1	8	39

(b) Mutlinomial Naïve Bayes



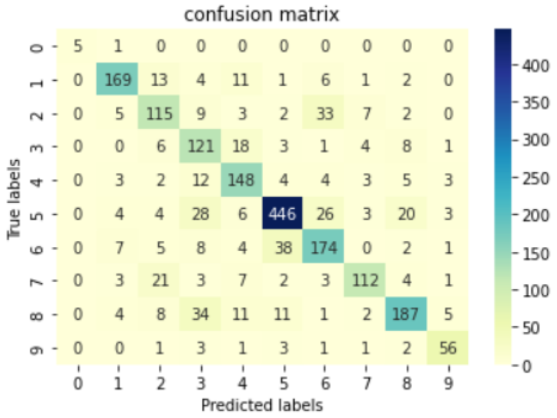
	0	1	2	3	4	5	6	7	8	9
0	0	4	0	0	0	0	1	0	1	0
1	0	165	11	5	8	4	6	3	5	0
2	0	7	106	7	2	11	30	6	7	0
3	0	0	3	114	16	12	1	4	12	0
4	0	4	3	12	143	10	3	2	6	1
5	0	3	1	8	5	497	9	2	14	1
6	0	8	7	1	4	50	164	1	3	1
7	0	2	19	1	6	5	1	113	9	0
8	0	3	8	18	6	20	1	2	203	2
9	0	0	0	2	1	9	1	0	5	50

(c) Logistic Regression



	0	1	2	3	4	5	6	7	8	9
0	2	3	1	0	0	0	0	0	0	0
1	0	172	14	3	8	2	3	1	4	0
2	0	11	117	7	2	5	26	7	1	0
3	0	8	13	90	19	16	1	4	11	0
4	0	26	9	13	114	6	3	4	7	2
5	0	15	23	15	7	445	17	8	9	1
6	0	16	14	5	4	50	144	2	3	1
7	0	10	15	3	2	3	1	116	6	0
8	0	17	18	19	11	22	3	7	165	1
9	0	4	2	3	2	4	0	1	3	49

(d) KNN



(e) SVM

Figure 5.2: Confusion matrix of Local dataset I from various ML models by Count Vector technique

From the above figures, we can observe class 4 and 9 which is plotted on a (3,3) & (8,8) have the most misclassified data samples with other classes.

5.1.1.3 Cross-Validation

This is a very important step. We perform 10 cross-validations to analyze the dataset independently. In the following Table 5.1, 10 different scores and mean value from each classifier is mentioned

Table 5.1: Cross-Validation of Local dataset I from various ML models by Count Vector technique

Cross validation	Random Forest	Mutlinomial Naïve Bayes	SVM	Logistic Regression	KNN
1st Score	70.4	68.8	69.9	72.5	67.3
2nd Score	77.6	77.0	80.7	80.3	72.8
3rd Score	77.9	76.3	79.7	81	73.4
4th Score	77.2	76.8	78.2	79.1	71.4
5th Score	77.6	74.4	78.7	78.2	71.3
6th Score	78.7	76.7	77.6	78.3	73.7

7th Score	78.8	78.2	82.2	81.3	73.1
8th Score	78.2	75.4	80	80.7	72.5
9th Score	59.4	57.7	60.1	60.3	52.1
10th Score	57.8	55.8	59.9	59.6	53.7
Mean	73.4	71.7	74.6	75.1	68.1

From the table, we find that logistic regression gives the highest acracy in 10 Cross-validation. Thus, below is the figure of the confusion matrix of 10 cross-validations from logistic regression. Logistic regression has the highest mean value of 75.1 of all other classifiers. most miss classification is occurring in class 7 which is plotted in (6,6).

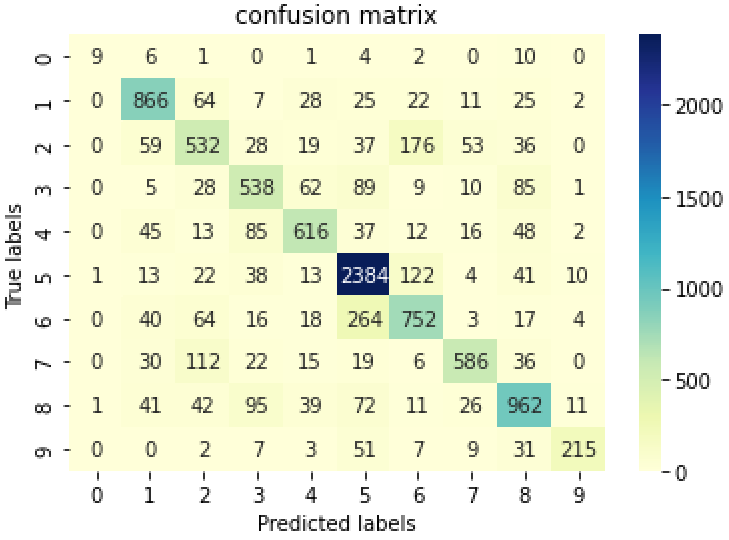


Figure 5.3: Confusion matrix from Count Vector of 10-Cross Validation (Logistic Regression)

5.1.1.4 Accuracy Graph

The following accuracy graph presents the testing and cross-validation accuracies on various classifiers.

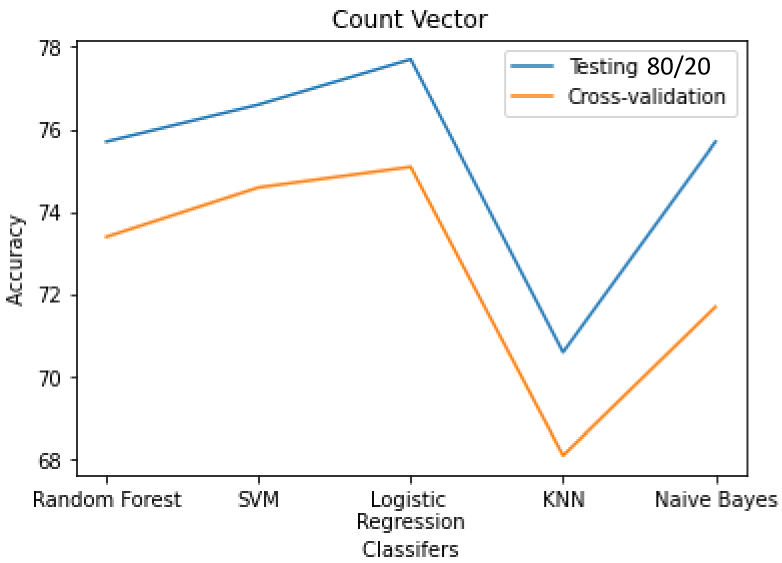


Figure 5.4: Accuracy graph on Local Dataset I from Count Vector Technique

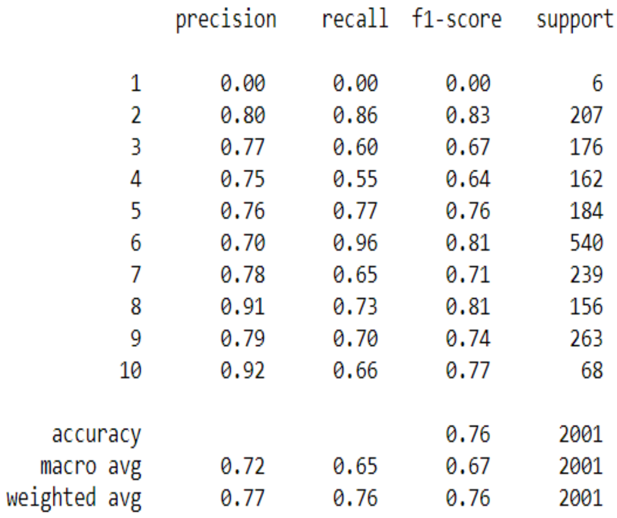
This graph shows Logistic Regression is the most accurate classifier in testing as well as in 10 cross-validations for count vector technique is given Local Dataset-1.

5.1.2 TF-IDF

There is another technique after pre-processing stage we perform TF-IDF in feature extraction and then apply different classifiers to acquire various results for a better understanding of the findings.

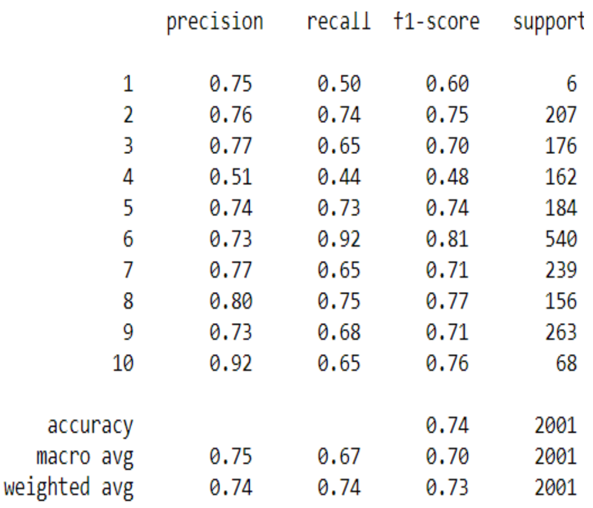
5.1.2.1 Classification Report

With the help of a classification report, we can easily know what is the precision, recall f1-score, and support(data-samples) in each class after applying TF-IDF. Also, find out overall accuracy. Following are the testing classification reports produced by five different ML models.



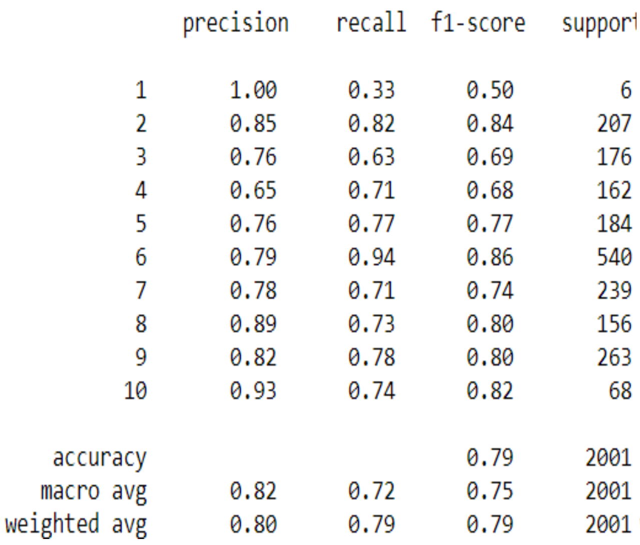
```
              precision    recall  f1-score   support

           1       0.00      0.00      0.00         6
           2       0.80      0.86      0.83       207
           3       0.77      0.60      0.67       176
           4       0.75      0.55      0.64       162
           5       0.76      0.77      0.76       184
           6       0.70      0.96      0.81       540
           7       0.78      0.65      0.71       239
           8       0.91      0.73      0.81       156
           9       0.79      0.70      0.74       263
          10       0.92      0.66      0.77        68

    accuracy                           0.76      2001
   macro avg       0.72      0.65      0.67      2001
weighted avg       0.77      0.76      0.76      2001
```

(a) Random Forest



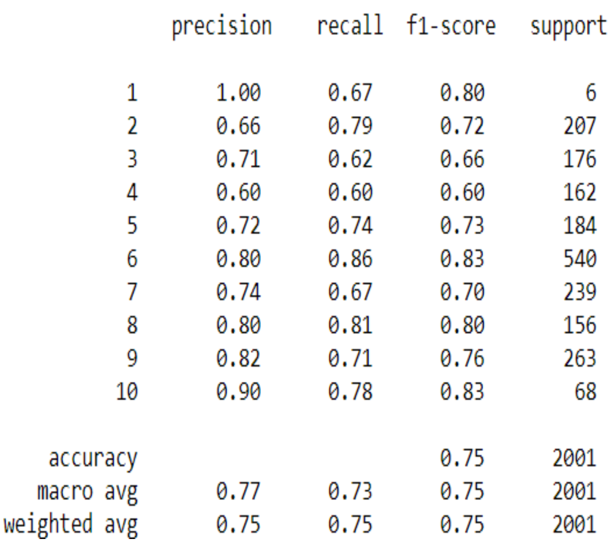
```
              precision    recall  f1-score   support

           1       0.75      0.50      0.60         6
           2       0.76      0.74      0.75       207
           3       0.77      0.65      0.70       176
           4       0.51      0.44      0.48       162
           5       0.74      0.73      0.74       184
           6       0.73      0.92      0.81       540
           7       0.77      0.65      0.71       239
           8       0.80      0.75      0.77       156
           9       0.73      0.68      0.71       263
          10       0.92      0.65      0.76        68

    accuracy                           0.74      2001
   macro avg       0.75      0.67      0.70      2001
weighted avg       0.74      0.74      0.73      2001
```

(b) Mutlinomial Naïve Bayes



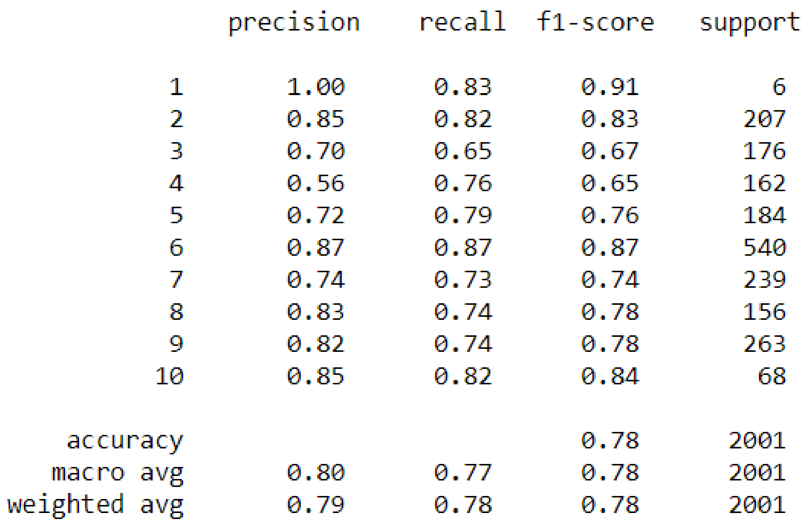
```
              precision    recall  f1-score   support

           1       1.00      0.33      0.50         6
           2       0.85      0.82      0.84       207
           3       0.76      0.63      0.69       176
           4       0.65      0.71      0.68       162
           5       0.76      0.77      0.77       184
           6       0.79      0.94      0.86       540
           7       0.78      0.71      0.74       239
           8       0.89      0.73      0.80       156
           9       0.82      0.78      0.80       263
          10       0.93      0.74      0.82        68

    accuracy                           0.79      2001
   macro avg       0.82      0.72      0.75      2001
weighted avg       0.80      0.79      0.79      2001
```

(c) Logistic Regression



```
              precision    recall  f1-score   support

           1       1.00      0.67      0.80         6
           2       0.66      0.79      0.72       207
           3       0.71      0.62      0.66       176
           4       0.60      0.60      0.60       162
           5       0.72      0.74      0.73       184
           6       0.80      0.86      0.83       540
           7       0.74      0.67      0.70       239
           8       0.80      0.81      0.80       156
           9       0.82      0.71      0.76       263
          10       0.90      0.78      0.83        68

    accuracy                           0.75      2001
   macro avg       0.77      0.73      0.75      2001
weighted avg       0.75      0.75      0.75      2001
```

(d) KNN



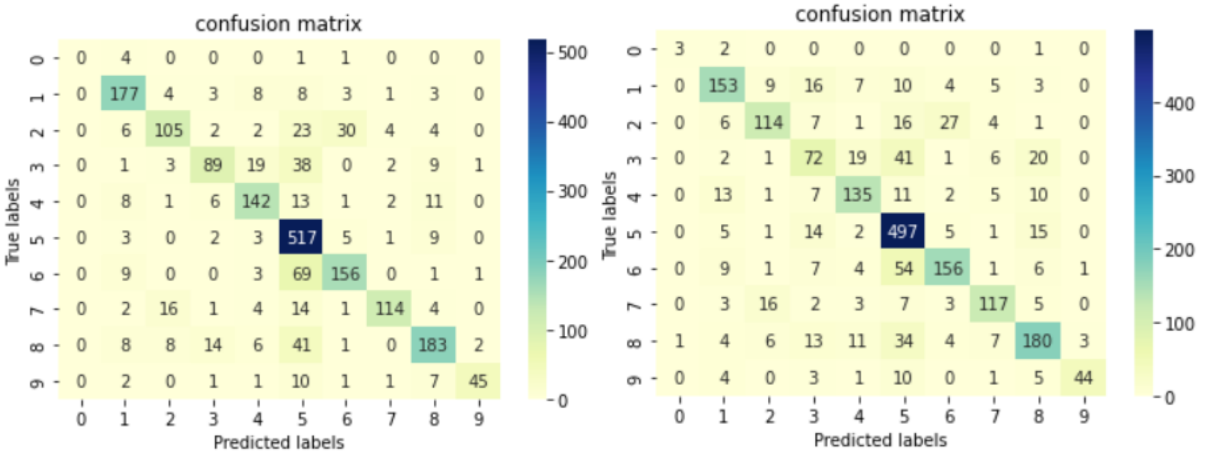
```
              precision    recall  f1-score   support

           1       1.00      0.83      0.91         6
           2       0.85      0.82      0.83       207
           3       0.70      0.65      0.67       176
           4       0.56      0.76      0.65       162
           5       0.72      0.79      0.76       184
           6       0.87      0.87      0.87       540
           7       0.74      0.73      0.74       239
           8       0.83      0.74      0.78       156
           9       0.82      0.74      0.78       263
          10       0.85      0.82      0.84        68

    accuracy                           0.78      2001
   macro avg       0.80      0.77      0.78      2001
weighted avg       0.79      0.78      0.78      2001
```

(e) SVM

Figure 5.5: Classification reports of Local dataset I from various ML models by TF-IDF technique

5.1.2.2 Confusion Matrix

With the help of the confusion matrix, we will know each class data more accurately. This helps in evaluating the performance of a classification model. The matrix compares the actual true value with the classified value given by the Ml model. The following figure shows the graphs of different ML models in which columns have 0-9 (I to 10) true class whereas rows have 0-9 (1-10) predicted class.

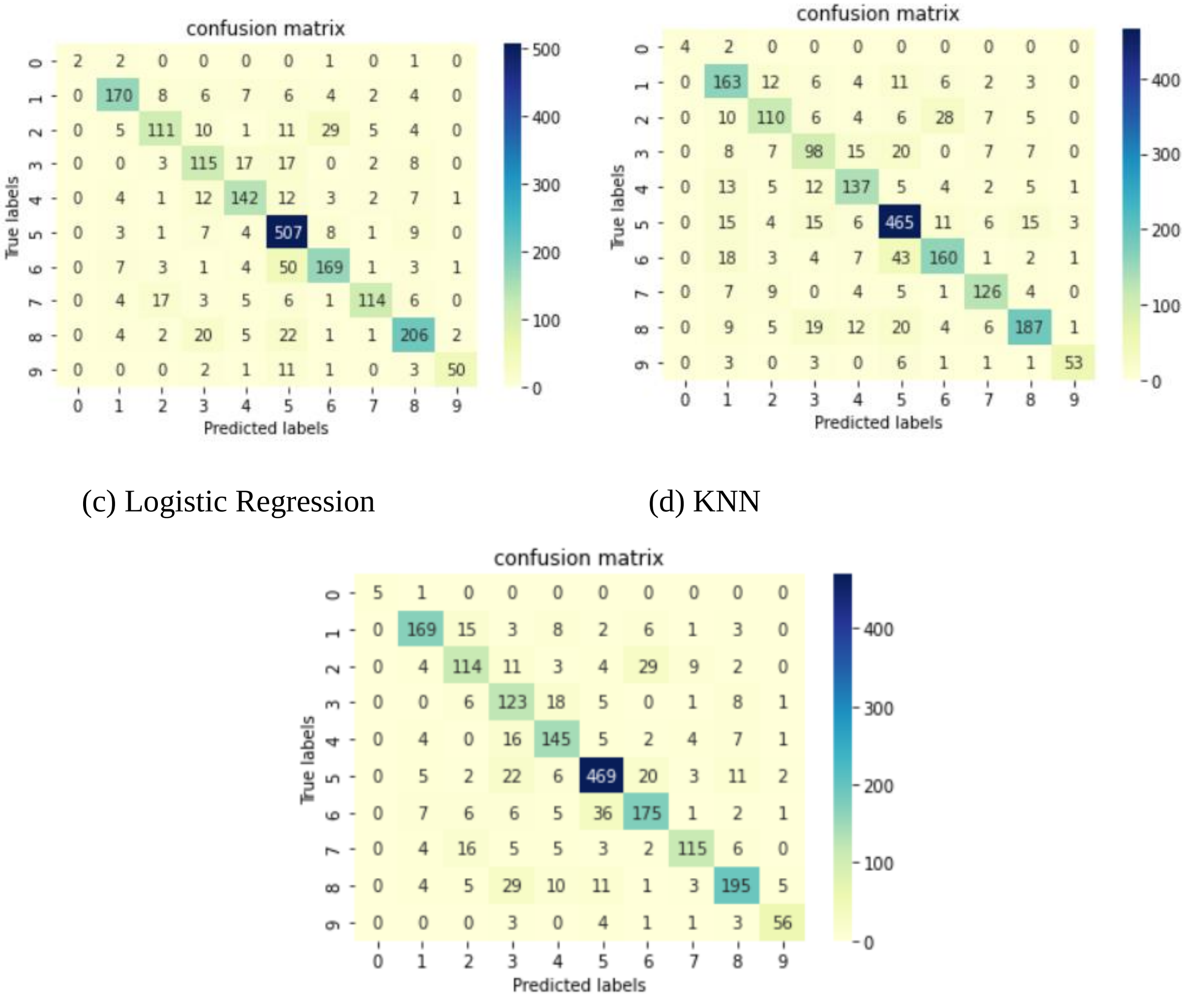


	0	1	2	3	4	5	6	7	8	9
0	0	4	0	0	0	1	1	0	0	0
1	0	177	4	3	8	8	3	1	3	0
2	0	6	105	2	2	23	30	4	4	0
3	0	1	3	89	19	38	0	2	9	1
4	0	8	1	6	142	13	1	2	11	0
5	0	3	0	2	3	517	5	1	9	0
6	0	9	0	0	3	69	156	0	1	1
7	0	2	16	1	4	14	1	114	4	0
8	0	8	8	14	6	41	1	0	183	2
9	0	2	0	1	1	10	1	1	7	45

(a) Random Forest

	0	1	2	3	4	5	6	7	8	9
0	3	2	0	0	0	0	0	0	1	0
1	0	153	9	16	7	10	4	5	3	0
2	0	6	114	7	1	16	27	4	1	0
3	0	2	1	72	19	41	1	6	20	0
4	0	13	1	7	135	11	2	5	10	0
5	0	5	1	14	2	497	5	1	15	0
6	0	9	1	7	4	54	156	1	6	1
7	0	3	16	2	3	7	3	117	5	0
8	1	4	6	13	11	34	4	7	180	3
9	0	4	0	3	1	10	0	1	5	44

(b) Mutlinomial Naïve Bayes



	0	1	2	3	4	5	6	7	8	9
0	2	2	0	0	0	0	1	0	1	0
1	0	170	8	6	7	6	4	2	4	0
2	0	5	111	10	1	11	29	5	4	0
3	0	0	3	115	17	17	0	2	8	0
4	0	4	1	12	142	12	3	2	7	1
5	0	3	1	7	4	507	8	1	9	0
6	0	7	3	1	4	50	169	1	3	1
7	0	4	17	3	5	6	1	114	6	0
8	0	4	2	20	5	22	1	1	206	2
9	0	0	0	2	1	11	1	0	3	50

(c) Logistic Regression

	0	1	2	3	4	5	6	7	8	9
0	4	2	0	0	0	0	0	0	0	0
1	0	163	12	6	4	11	6	2	3	0
2	0	10	110	6	4	6	28	7	5	0
3	0	8	7	98	15	20	0	7	7	0
4	0	13	5	12	137	5	4	2	5	1
5	0	15	4	15	6	465	11	6	15	3
6	0	18	3	4	7	43	160	1	2	1
7	0	7	9	0	4	5	1	126	4	0
8	0	9	5	19	12	20	4	6	187	1
9	0	3	0	3	0	6	1	1	1	53

(d) KNN

	0	1	2	3	4	5	6	7	8	9
0	5	1	0	0	0	0	0	0	0	0
1	0	169	15	3	8	2	6	1	3	0
2	0	4	114	11	3	4	29	9	2	0
3	0	0	6	123	18	5	0	1	8	1
4	0	4	0	16	145	5	2	4	7	1
5	0	5	2	22	6	469	20	3	11	2
6	0	7	6	6	5	36	175	1	2	1
7	0	4	16	5	5	3	2	115	6	0
8	0	4	5	29	10	11	1	3	195	5
9	0	0	0	3	0	4	1	1	3	56

(e) SVM

Figure 5.6: Confusion matrix of Local dataset I from various ML models by TF-IDF technique

From the above figure, we can see that our classifiers miss classifying class 7 with class 5.

5.1.2.3 Cross-Validation

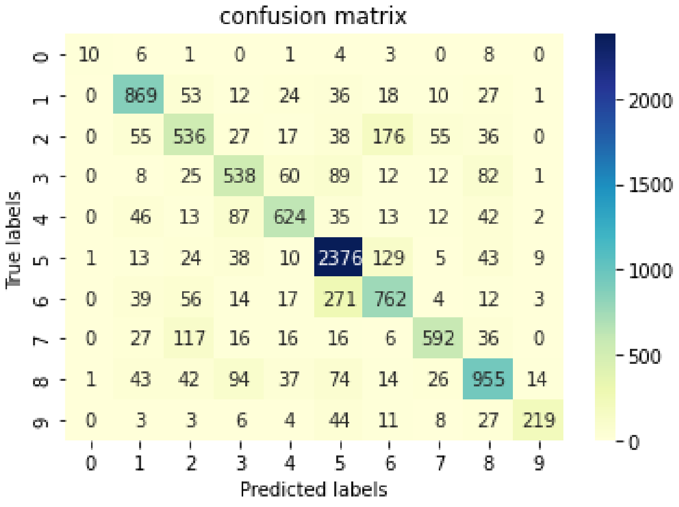
This is a very important step. We perform 10 cross-validations to analyze the dataset independently.

Table 5.2: Cross-Validation of Local dataset I from various ML models by TF-IDF technique

Cross validation	Random Forest	Mutlinomial Naïve Bayes	SVM	Logistic Regression	KNN

1st Score	70.4	69.6	70.5	73.8	70.1
2nd Score	78.6	76.1	82.1	81.4	75.8
3rd Score	77.6	76.9	81.7	82.6	76.8
4th Score	77.2	77.2	79.3	80.8	77.6
5th Score	77.4	74.7	80	80.7	74.9
6th Score	77.5	76.6	79.4	78.7	77
7th Score	79.4	79	83.1	82.7	77.9
8th Score	76.2	77.7	81.4	83.8	77.4
9th Score	58.6	57.2	60.9	61.4	56.2
10th Score	58	56.8	60	59.6	57.6
Mean	73	73	75.8	77	72.1

Below is the figure of the confusion matrix of 10 cross-validations from logistic regression. Logistic regression has the highest mean value of 77 of all other classifiers. most miss classification is occurring in class 7 which is plotted in (6,6).



	0	1	2	3	4	5	6	7	8	9
0	10	6	1	0	1	4	3	0	8	0
1	0	869	53	12	24	36	18	10	27	1
2	0	55	536	27	17	38	176	55	36	0
3	0	8	25	538	60	89	12	12	82	1
4	0	46	13	87	624	35	13	12	42	2
5	1	13	24	38	10	2376	129	5	43	9
6	0	39	56	14	17	271	762	4	12	3
7	0	27	117	16	16	16	6	592	36	0
8	1	43	42	94	37	74	14	26	955	14
9	0	3	3	6	4	44	11	8	27	219

Figure 5.7: Confusion matrix from TF-IDF of 10-Cross Validation (Logistic Regression)

5.1.2.4 Accuracy Graph

The following accuracy graph presents the testing and cross-validation accuracies on various classifiers.

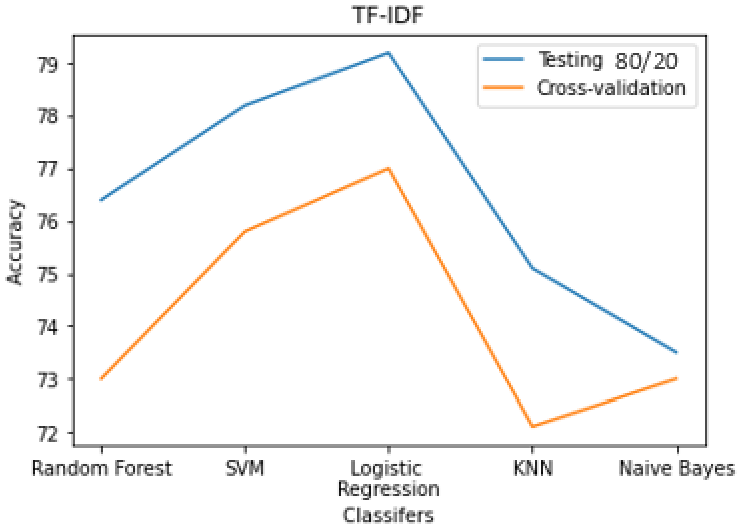


Figure 5.8: Accuracy graph on Local Dataset I from TF-IDF Technique

With the help of this graph, we find that logistic regression is the best classifier in terms of testing and cross-validation accuracy among all other classifiers.

5.1.3 Accuracies of local dataset I

The following table shows the accuracies of all classifiers with both techniques. Three different accuracies sets are mentioned. From these, we can see the overfitting in Random Forest & SVM.

Table 5.3: Overall Accuracies of the local dataset I.

Classifiers	**80/20** **Training Accuracy**		**80/20** **Testing Accuracy**		**10 fold CV** **Testing Accuracy**	
	Count Vector	TF-IDF	Count Vector	TF-IDF	Count Vector	TF-IDF
Random Forest	99.8%	99.8%	75.7%	76.4%	73.4%	73%

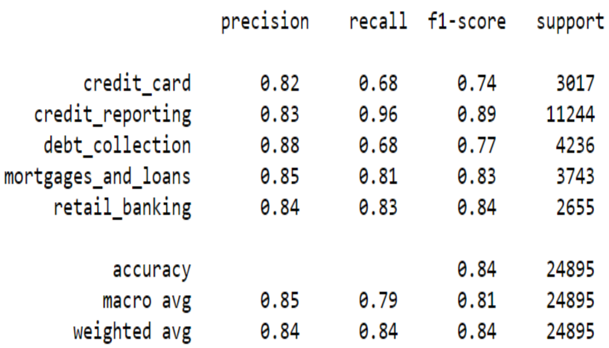
Mutlinomial Naïve Bayes	77.5%	82.6%	75.7%	73.5%	71.7%	73%
SVM	86.6%	91.3%	76.6%	78.2%	74.6%	75.8%
Logistic Regression	83.2%	86.5%	77.7%	79.2%	75.1%	77%
KNN	78.9%	78.3%	70.6%	75.1%	68.1%	72.1%

5.2 Consumer Complaints Dataset 2

As previously discussed in section 4.2.2 that in dataset 2, we have 5 classes and after pre-processing the data we have 1,24,473 samples in the dataset. Which is further divided into 80% training and 20% testing. We apply two different techniques of feature extraction and then we apply various ML models to each of the techniques.

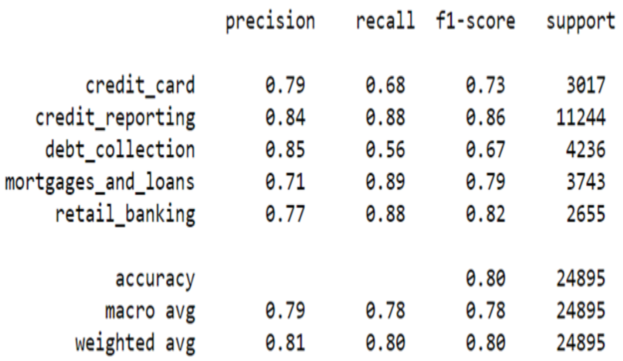
5.2.1 Count Vector

5.2.1.1 Classification reports



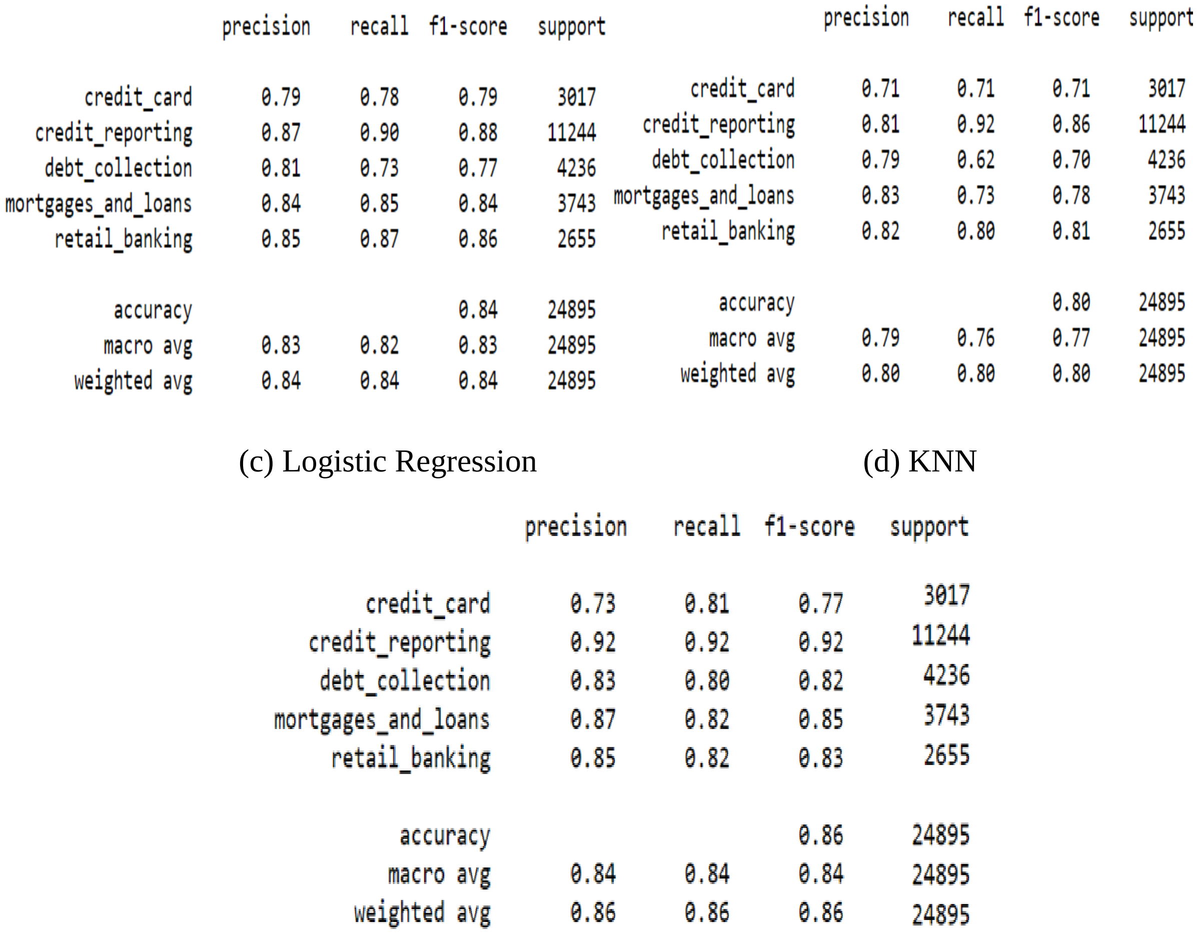
```
                   precision    recall  f1-score   support

       credit_card       0.82      0.68      0.74      3017
  credit_reporting       0.83      0.96      0.89     11244
   debt_collection       0.88      0.68      0.77      4236
mortgages_and_loans       0.85      0.81      0.83      3743
    retail_banking       0.84      0.83      0.84      2655

          accuracy                           0.84     24895
         macro avg       0.85      0.79      0.81     24895
      weighted avg       0.84      0.84      0.84     24895
```

(a) Random Forest



```
                   precision    recall  f1-score   support

       credit_card       0.79      0.68      0.73      3017
  credit_reporting       0.84      0.88      0.86     11244
   debt_collection       0.85      0.56      0.67      4236
mortgages_and_loans       0.71      0.89      0.79      3743
    retail_banking       0.77      0.88      0.82      2655

          accuracy                           0.80     24895
         macro avg       0.79      0.78      0.78     24895
      weighted avg       0.81      0.80      0.80     24895
```

(b) Mutlinomial Naïve Bayes



```
                  precision    recall  f1-score   support

      credit_card       0.79      0.78      0.79      3017
 credit_reporting       0.87      0.90      0.88     11244
  debt_collection       0.81      0.73      0.77      4236
mortgages_and_loans     0.84      0.85      0.84      3743
   retail_banking       0.85      0.87      0.86      2655

         accuracy                           0.84     24895
        macro avg       0.83      0.82      0.83     24895
     weighted avg       0.84      0.84      0.84     24895
```

(c) Logistic Regression

```
                  precision    recall  f1-score   support

      credit_card       0.71      0.71      0.71      3017
 credit_reporting       0.81      0.92      0.86     11244
  debt_collection       0.79      0.62      0.70      4236
mortgages_and_loans     0.83      0.73      0.78      3743
   retail_banking       0.82      0.80      0.81      2655

         accuracy                           0.80     24895
        macro avg       0.79      0.76      0.77     24895
     weighted avg       0.80      0.80      0.80     24895
```

(d) KNN

```
                  precision    recall  f1-score   support

      credit_card       0.73      0.81      0.77      3017
 credit_reporting       0.92      0.92      0.92     11244
  debt_collection       0.83      0.80      0.82      4236
mortgages_and_loans     0.87      0.82      0.85      3743
   retail_banking       0.85      0.82      0.83      2655

         accuracy                           0.86     24895
        macro avg       0.84      0.84      0.84     24895
     weighted avg       0.86      0.86      0.86     24895
```

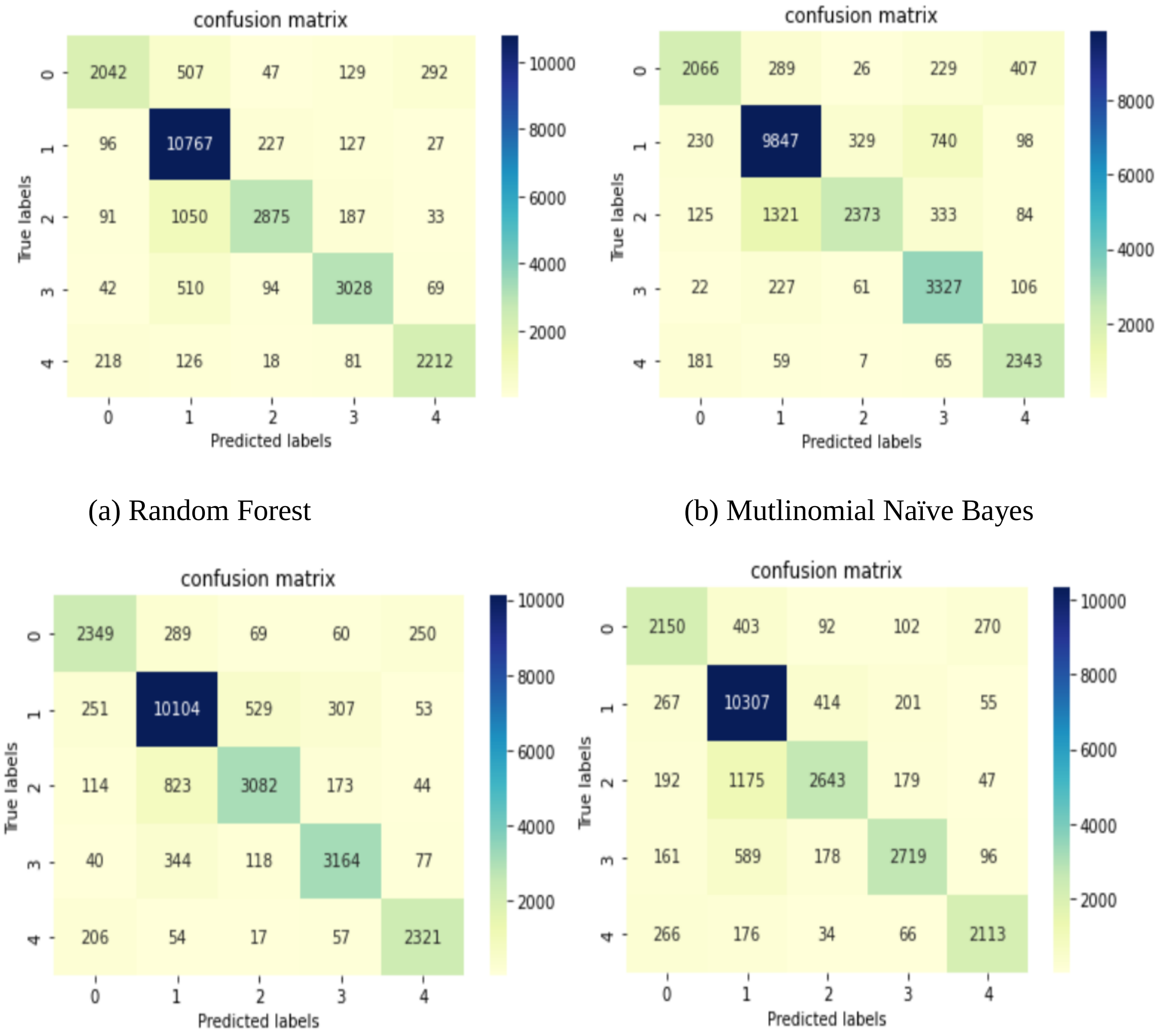
(e) SVM

Figure 5.9: Classification reports of Consumer Complaints Dataset II from various ML models by Count Vector technique

From the above Classification reports, we can see the accuracies of different classes from the count vector technique with various classifiers. All class 1 has very less accuracy in all classifiers with the testing data samples of 3017.

5.2.1.2 Confusion matrix

With the help of the confusion matrix, we will know each class data more accurately. This helps in evaluating the performance of a classification model. The matrix compares the actual true value with the classified value given by the Ml model. The following figure shows the graphs of different ML models in which columns have 0- 4 (I to 5) true class whereas rows have 0-4 (1-5) predicted class.

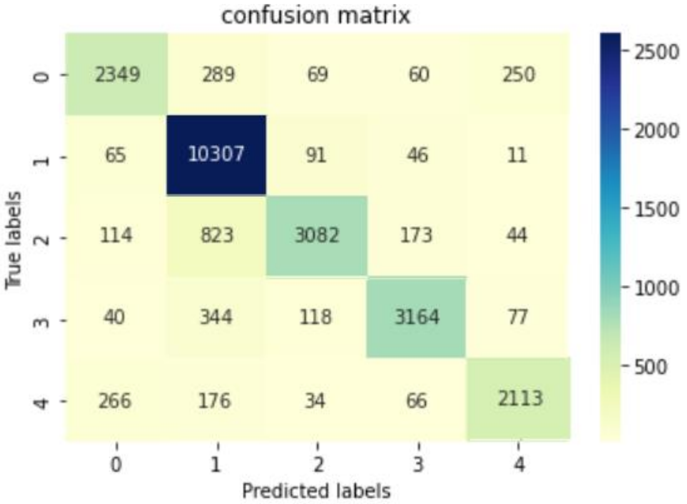


(a) Random Forest

(b) Mutlinomial Naïve Bayes

(c) Logistic Regression

(d) KNN



(e) SVM

Figure 5.10: Confusion matrix of Consumer Complaints Dataset II from various ML models by Count Vector technique

From the above Figure 5.10, in all classifiers class, 1 is mostly misclassified with other classes. Most data samples of class1 are classified with class 2 and class 5.

5.2.1.3 Cross-Validation

This is a very important step. We perform 10 cross-validations to analyze the dataset independently. We practice only 4 classifiers.

Table 5.4: Cross-Validation of Consumer Complaints Dataset II from various ML models by Count Vector technique

Cross validation	Random Forest	SVM	Logistic Regression	KNN	Mutlinomial Naïve Bayes
1st Score	81	91.4	82.2	77.1	78.8
2nd Score	82.1	90.9	84.4	79.8	81.1
3rd Score	78.1	89.2	80.4	73.4	76.1
4th Score	82.4	82.8	84.4	78.8	81.1

5th Score	83.5	87.3	84.7	79.1	81.4
6th Score	80.7	75	82.4	76.3	78.7
7th Score	83	74.5	83.1	79.4	80.3
8th Score	84.2	81	85.6	79.9	83.1
9th Score	79.8	77.2	82.5	77	79
10th Score	83.5	81.3	84.7	79.6	80.2
Mean	81.8	83	83.4	78	80

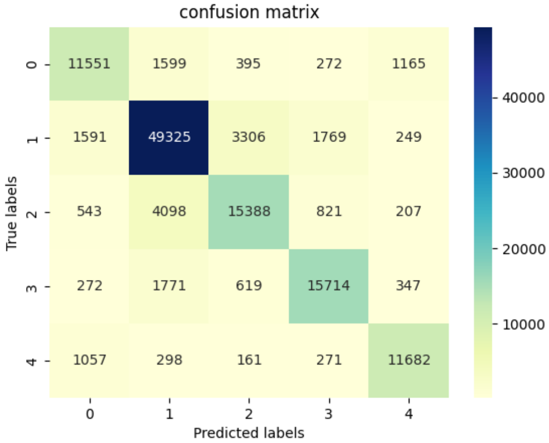


Figure 5.11: Confusion matrix from Count Vector of 10-Cross Validation (logistic Regression) on Consumer Complaints Dataset

5.2.1.4 Accuracy Graph

below the accuracy graph for Kaggle dataset II, present the testing, and cross-validation accuracies on various classifiers.

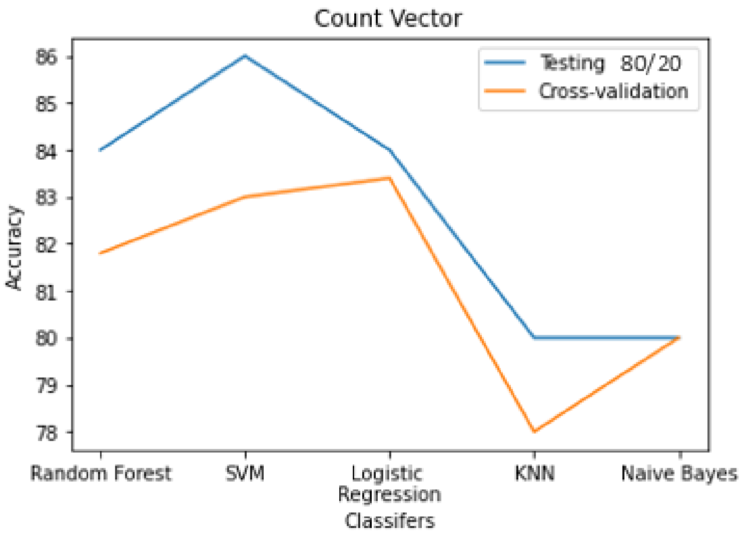
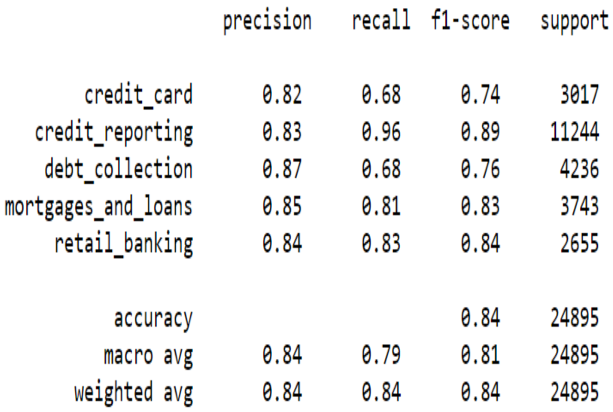


Figure 5.12: Accuracy graph on Consumer Complaints Dataset II from Count Vector Technique

5.2.2 TF-IDF

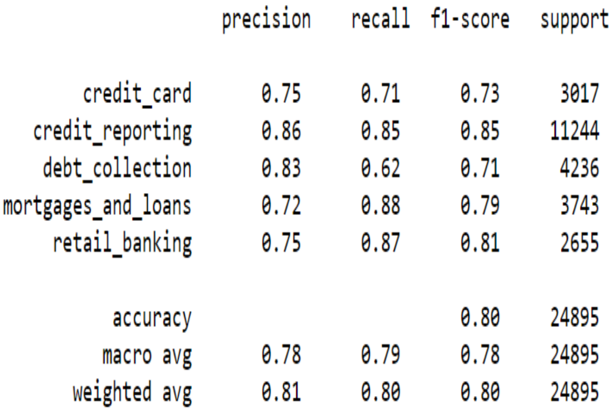
5.2.2.1 Classification Report

From the following Classification reports, we can see the accuracies of different classes from the count vector technique with various classifiers. All class 1 has very less accuracy in all classifiers with the testing data samples of 3017. The highest accuracies obtain by SVM from the TF-IDF technique. The second highest accuracy the Kaggle dataset II achieve is from Logistic regression.



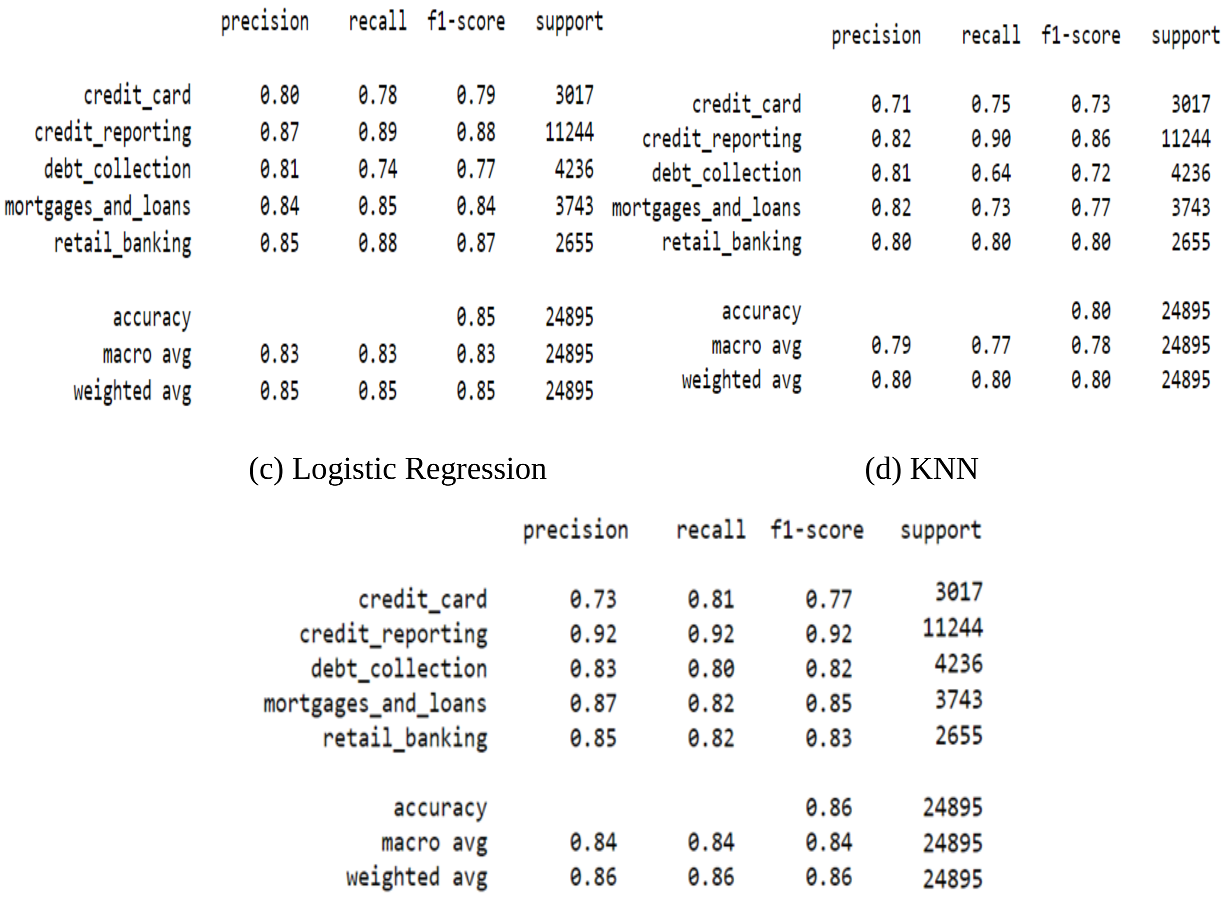
```
                  precision    recall  f1-score   support

     credit_card       0.82      0.68      0.74      3017
credit_reporting       0.83      0.96      0.89     11244
 debt_collection       0.87      0.68      0.76      4236
mortgages_and_loans    0.85      0.81      0.83      3743
  retail_banking       0.84      0.83      0.84      2655

        accuracy                           0.84     24895
       macro avg       0.84      0.79      0.81     24895
    weighted avg       0.84      0.84      0.84     24895
```

(a) Random Forest



```
                  precision    recall  f1-score   support

     credit_card       0.75      0.71      0.73      3017
credit_reporting       0.86      0.85      0.85     11244
 debt_collection       0.83      0.62      0.71      4236
mortgages_and_loans    0.72      0.88      0.79      3743
  retail_banking       0.75      0.87      0.81      2655

        accuracy                           0.80     24895
       macro avg       0.78      0.79      0.78     24895
    weighted avg       0.81      0.80      0.80     24895
```

(b) Mutlinomial Naïve Bayes



```
                   precision    recall  f1-score   support

       credit_card       0.80      0.78      0.79      3017
  credit_reporting       0.87      0.89      0.88     11244
   debt_collection       0.81      0.74      0.77      4236
mortgages_and_loans      0.84      0.85      0.84      3743
    retail_banking       0.85      0.88      0.87      2655

          accuracy                           0.85     24895
         macro avg       0.83      0.83      0.83     24895
      weighted avg       0.85      0.85      0.85     24895
```

(c) Logistic Regression

```
                   precision    recall  f1-score   support

       credit_card       0.71      0.75      0.73      3017
  credit_reporting       0.82      0.90      0.86     11244
   debt_collection       0.81      0.64      0.72      4236
mortgages_and_loans      0.82      0.73      0.77      3743
    retail_banking       0.80      0.80      0.80      2655

          accuracy                           0.80     24895
         macro avg       0.79      0.77      0.78     24895
      weighted avg       0.80      0.80      0.80     24895
```

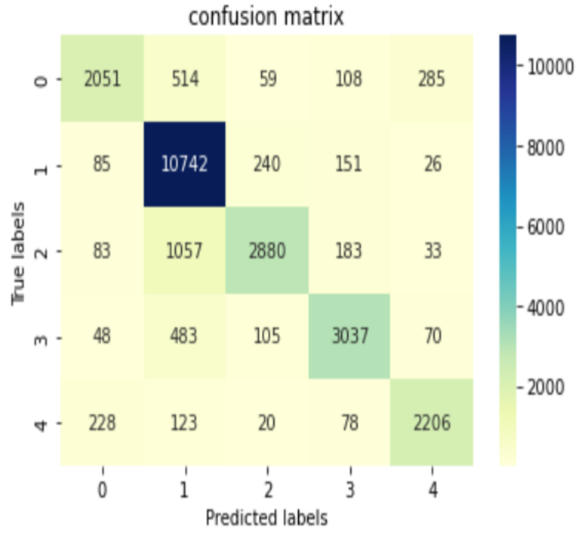
(d) KNN

```
                   precision    recall  f1-score   support

       credit_card       0.73      0.81      0.77      3017
  credit_reporting       0.92      0.92      0.92     11244
   debt_collection       0.83      0.80      0.82      4236
mortgages_and_loans      0.87      0.82      0.85      3743
    retail_banking       0.85      0.82      0.83      2655

          accuracy                           0.86     24895
         macro avg       0.84      0.84      0.84     24895
      weighted avg       0.86      0.86      0.86     24895
```

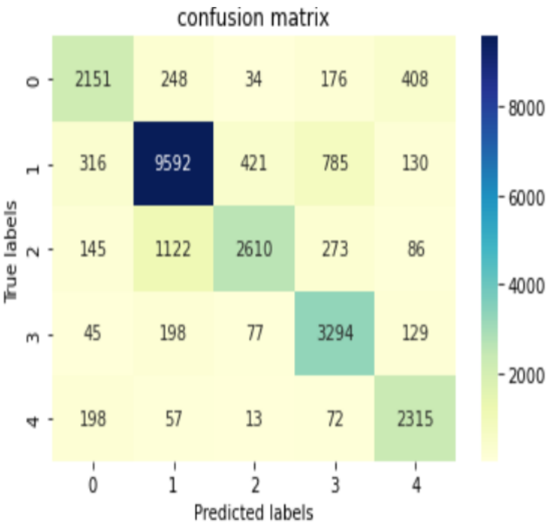
(e) SVM

Figure 5.13: Classification reports of Consumer Complaints Dataset II from various ML models by TF-IDF technique

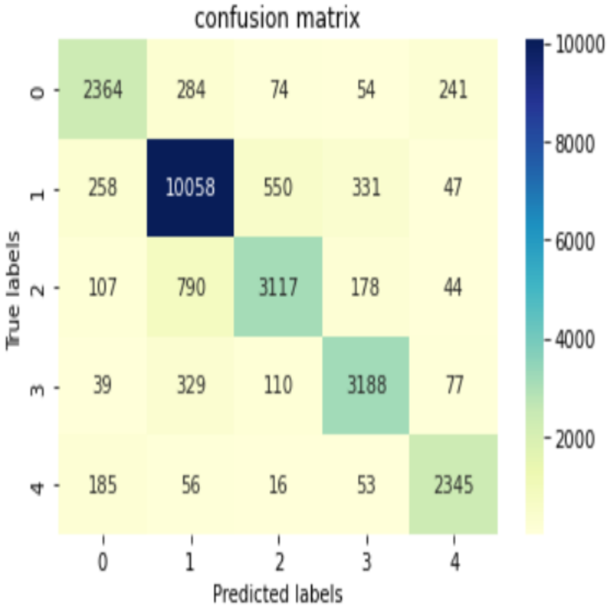
5.2.2.2 Confusion Matrix



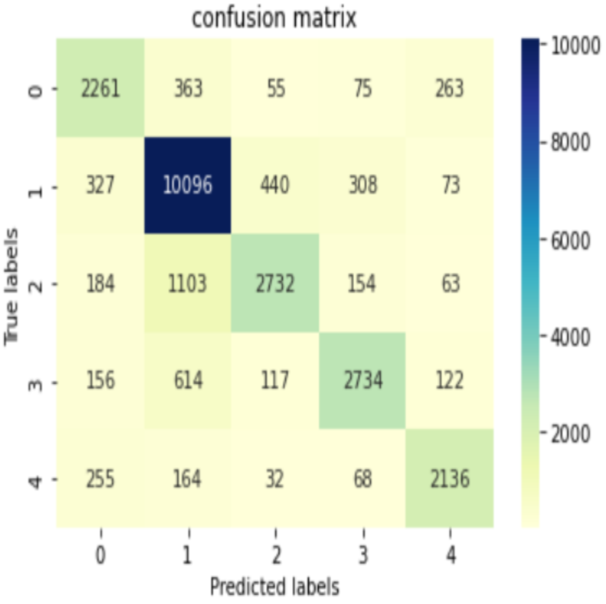
(a) Random Forest



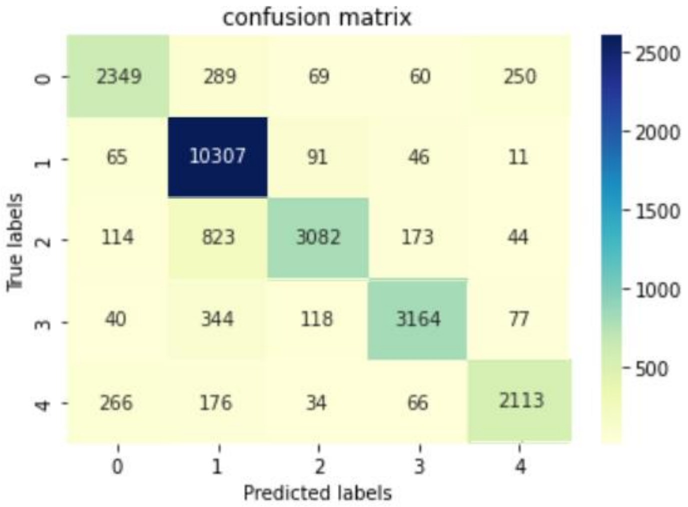
(b) Mutlinomial Naïve Bayes



(c) Logistic Regression



(d) KNN



(e) SVM

Figure 5.14: Confusion matrix of Consumer Complaints Dataset II from various ML models by TF-IDF technique

From the above figures, we can observe class 3 which is plotted on (2,2) misclassified data samples with other classes. From this, we can see class 3 is mostly miss-matched with class 2.

5.2.2.3 Cross-Validation

This is a very important step. We perform 10 cross-validations to analyze the dataset independently. We practice only 4 classifiers.

Table 5.5: Cross-Validation of Consumer Complaints Dataset II from various ML models by TF-IDF technique

Cross validation	Random Forest	SVM	Logistic Regression	KNN	Mutlinomial Naïve Bayes
1st Score	81	91.3	82.1	76.3	76.9
2nd Score	82.2	91.2	85.1	79.3	80.6
3rd Score	78.3	89.8	80.6	73.2	75.3
4th Score	82.3	83	84.8	78.2	80.4

5th Score	83.2	87.5	84.7	78.4	79.5
6th Score	80.7	75.5	82.5	76.3	76.9
7th Score	82.9	75	83.4	78.4	78.9
8th Score	84.5	81.7	85.9	79.7	82.3
9th Score	80.1	78.7	82.6	76.3	77.9
10th Score	83.5	82	84.9	79.4	79.7
Mean	83.6	81.9	83.7	77.6	78.8

From the above table, it can be concluded that SVM and Logistic regression both give good results. Logistic regression is .1 % better invalidating the data samples. Following Figure 5.15 give the cross-validation of Logistic Regression.

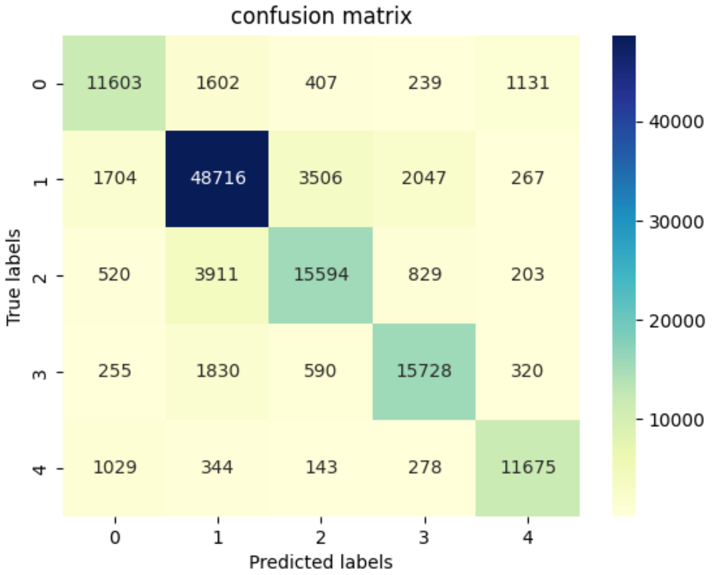


Figure 5.15: Confusion matrix from TF-IDF of 10-Cross Validation (logistic Regression) on Consumer Complaints Dataset

5.2.2.4 Accuracy Graph

Following the accuracy graph for dataset II, present testing, and cross-validation accuracies on various classifiers. This graph show that Logistic Regression and SVM both are good classifiers for the TIFD technique.

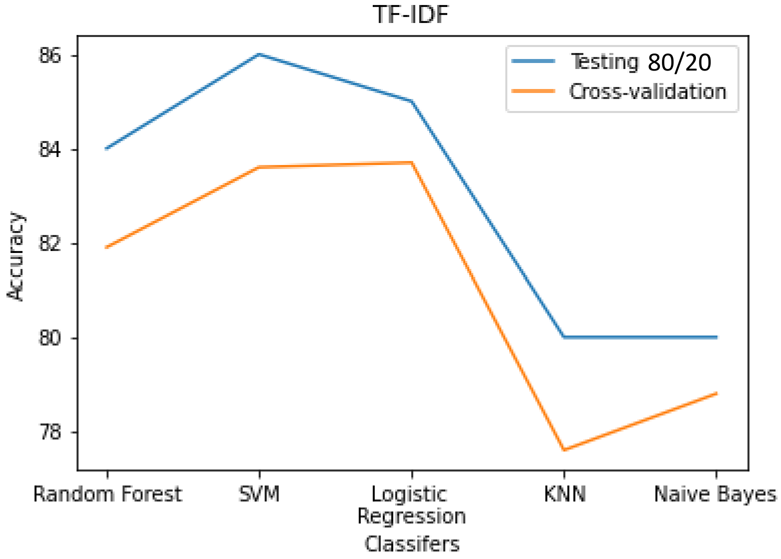


Figure 5.16: Accuracy graph on Consumer Complaints Dataset II from TF-IDF Technique

5.2.3 Accuracies of Kaggle Dataset II

From the following detailed Table 5.6 of all accuracies regarding training, testing & 10-cross validation from both techniques, it can be seen that there is overfitting in the count vector technique. As the Naïve Bayes is not calculated in cross-validation. Thus it is not mentioned.

Table 5.6: Overall Accuracies of Consumer Complaints Dataset II.

Classifiers	**80/20** **Training Accuracy**		**80/20** **Testing Accuracy**		**10 fold CV** **Testing Accuracy**	
	Count Vector	TF-IDF	Count Vector	TF-IDF	Count Vector	TF-IDF
Random Forest	98%	99%	84%	84%	81.8%	81.9%

SVM	96%	89%	86%	86%	83%	83.6%
Logistic Regression	91%	93%	84%	85%	83.4%	83.7%
KNN	88%	83%	80%	80%	78%	77.6%
Mutlinomial Naïve Bayes	83%	90%	80%	80%	80%	78.8%

5.3 Error analysis

We used random SVM for dataset I because of the wide range of ML models' accuracy. Using both vectorization methods, we examined the accuracy confusion matrix. Confusion matrices generated using the count vector approach and the TF-IDF technique are shown in Table VII and Table VIII, respectively.

While examining the incorrectly classified cases, many problems were discovered. As a result of this lack of detail, some complaints are based only on a brief paragraph that does not adequately describe the class. Other languages, such as Urdu, were utilized in the comments. Complaints of semantic overlap have been made by some. Examples include class III (IT), which overlaps class II (HRD division). The complaints have indeed been misclassified in all of the following examples. When writing a brief complaint, it is easy to fall into the trap of merely mentioning a few issues. The human operator had categorized a few of these complaints incorrectly, and we were able to detect them.

Naive Bayes and Count Vectorizer are used in our system, and the training recall accuracy is 86.5 percent in dataset II; in contrast, they attained an accuracy of 86 percent as training recall in the paper [49]. However, this degree of precision was achieved even though duplicate samples and complaints were not eliminated.

5.4 Imbalanced data

Data from the real world is not always evenly distributed. There may be more data points in certain classes than in others. Taking this into consideration, the classifier may acquire a biasness toward larger classes. To some, it may seem like a biased classifier is doing well. There are 10 classes ranging from one to ten in the supplied data set I. Class 6 has 540 data points, while class 1 only has six. An efficiency of 0.8 would be attained if the classifier developed baisness for Class 6 and classified everything in that class. Even though all data points of class 1 were classified incorrectly, this seems to be a respectable result.

There are many ways to address this issue. To get classes of comparable size, you may resample the dataset. Over-sampling is utilized when there is a tiny dataset. This signifies that the dataset is enriched by the addition of tiny class instances. Overfitting might occur if there are duplicates in the data.

Under-sampling is utilized when the dataset is huge. This implies that huge class instances will be wiped off. When under- or over-sampling isn't an option due to the size of the dataset, alternative techniques such as reinforcing distinct misclassification costs may be used. Misclassifying a big class will cost less than misidentifying a small class using this approach. This encourages the classifier to classify the smaller category more often.

5.5 Comparing the Consumer Complaints results

As our dataset 2 is taken from Kaggle and there is also an article written on dataset 2. Thus, we compare our paper results with results that are displayed on the website.[50]. Some of the classifiers are not used in our given paper. They used both techniques with different classifiers with 80% training dataset and 20% testing dataset. Following are the accuracies that are displayed on the website.

Table 5.7: Accuracies of Consumer Complaints Dataset [50]

Reference	**Classifiers**	**80/20 Training Accuracy**		**80/20 Testing Accuracy**	
		Count Vector	**TF-IDF**	**Count Vector**	**TF-IDF**
GIT-HUB	Mutlinomial NB	83%	-	80%	-

	Decision Tree	-	85%	-	81%
	Gradient Boosting	-	88%	-	86%
Proposed System	SVM	96%	89%	86%	86%
	Logistic Regression	91%	93%	84%	85%
	Random Forest	98%	99%	84%	84%
	Mutlinomial NB	83%	90%	80%	80%

Chapter 6: Conclusion and Future work

The outcomes establish an image that a supervised machine learning algorithms is an excellent tool for designing an automatic classification of complaints categorization and autonomous system. The English language complaints belonging to various departments are used in our purposed system. As learning from the training data may take time but once the model is trained, it generates prediction in no time. For classification, the main issue was a class imbalance, so the first task was to come up with the best technique. In our system, we have used multiple classifiers, and two different approaches in the feature extraction. The results shown from the TF-IDF model perform better than the Count vector technique. Classifiers such as logistic regression and SVM using TF-IDF techniques generate better results. The results achieved on the Consumer Complaint dataset II will serve as a baseline as it is international dataset. Additionally, our proposed model achieved competitive performance on the local dataset I. This system can be used in any organization in classifying and distributing the large data of complaints to their designated centers.

7.1 Future Work

In the future, we can use different feature selection methods for the classification of complaints by selecting the best features. Additionally, we can implement different deep learning models that will give better results. We can also include more than 10 classes in our dataset or, increase the data samples, and with a better machine-learning algorithm get more accurate results.

7.2 Limitations

Open-source libraries were used to implement the classification algorithms in this study. The experiments only covered five different types of classifiers to evaluate.

English language is another limitation in natural language processing. Thus, \it contains only one dictionary and reduced the complexity for cleaning/transforming texts.

References

[1]. W. D. Eggers, N. Malik, and M. Gracie, "Natural Language Processing Examples in Government Data," *Deloitte Insights*, 2019.

[2]. R. Kowalski, M. Esteve, and S. Jankin Mikhaylov, "Improving public services by mining citizen feedback: An application of natural language processing," *Public Adm.*, vol. 98, no. 4, 2020, doi: 10.1111/padm.12656.

[3]. E. Momeni, C. Cardie, and N. Diakopoulos, "A survey on assessment and ranking methodologies for user-generated content on the web," *ACM Computing Surveys*, vol. 48, no. 3. 2015. doi: 10.1145/2811282.

[4]. J. Liu, W. C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," 2017. doi: 10.1145/3077136.3080834.

[5]. S. Codish and R. N. Shiffman, "A model of ambiguity and vagueness in clinical practice guideline recommendations.," AMIA Annu. Symp. Proc., 2005.

[6]. B. Zhao, "Clinical Data Extraction and Normalization of Cyrillic Electronic Health Records Via Deep-Learning Natural Language Processing," JCO Clin. Cancer Informatics, no. 3, 2019, doi: 10.1200/cci.19.00057.

[7]. Wendy G. Lehnert and Martin H. Ringle. "Strategies for Natural Language Processing". 2014. doi: 10.4324/9781315802671.

[8]. R. Dale, "The commercial NLP landscape in 2017," *Natural Language Engineering*, vol. 23, no. 4. 2017. doi: 10.1017/S1351324917000237.

[9]. Amazon Web Services Inc., "Amazon Machine Learning - Predictive Analytics with AWS," Amazon Web Services, 2018. www.predictiveanalyticstoday.com (accessed Dec. 19, 2020).

[10]. "Tokenization," Nlp.stanford.edu, 2022. https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html (accessed Aug. 19, 2021).

[11]. "N-grams, multiword expressions, lexical bundles | Sketch Engine," Sketch Engine, 2022. https://www.sketchengine.eu/userguide/user-manual/n-grams/ (accessed Jan. 19, 2022).

[12]. M. Sanderson, Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze,"Introduction to Information Retrieval, Cambridge University Press. 2008. ISBN-13 978-0-521-86571-5, xxi + 482 pages.," Nat. Lang. Eng., vol. 16, no. 1, 2010, doi: 10.1017/s1351324909005129.

[13]. S. Miller, J. Guinness, and A. Zamanian, "Name tagging with word clusters and discriminative training," 2004.

[14]. H. Alpert, “Classifying Complaints with Natural Language Processing,” Towards Data Science, May 05, 2021.

[15]. J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” 2014. doi: 10.3115/v1/d14-1162.

[16]. P. Oram, “ WordNet: An electronic lexical database. Christiane Fellbaum (Ed.). Cambridge, MA: MIT Press, 1998. Pp. 423. ,” *Appl. Psycholinguist.*, vol. 22, no. 1, 2001, doi: 10.1017/s0142716401221079.

[17]. E. Yan, J. Song, C. Liu, J. Luan, and W. Hong, “Comparison of support vector machine, back propagation neural network and extreme learning machine for syndrome element differentiation,” Artif. Intell. Rev., vol. 53, no. 4, 2020, doi: 10.1007/s10462-019-09738-z.

[18]. Rohith Gandhi, “Naive Bayes Classifier,” Towards Data Science, 2022. https://towardsdatascience.com/naivebayes-classifier-81d512f50a7c (accessed Feb. 19, 2022).

[19]. Scikit Learn Documentation can be accessed at http://scikitlearn.org/stable/documentation.html

[20]. V. Kurama, V. Kurama, “Introduction To Machine Learning,” Towards Data Science, 2017.

[21]. “Datasets and Machine Learning.” https://skymind.ai/wiki/datasets-ml (accessed Mar. 20, 2019).

[22]. “Overfitting and Underfitting With Machine Learning Algorithms.” https://machinelearningmastery.com/overfittingand-underfitting-with-machine-learning-algorithms/ (accessed May. 03, 2019).

[23]. Jacob T and Vanderplas, “Python data science handbook: tools and techniques for developers,” OReilly, pp. 363–370, 2016.

[24]. “Support Vector Machines,” MonkeyLearn, 2019. https://monkeylearn.com/text-classificationsupport-vector-machines-svm/. (accessed Mar. 13, 2019).

[25]. Vasilis Vryniotis, “The Naive Bayes Text Classifier,” 2013.

[26]. T. Hastie, R. Tibshirani, and J. Friedman, Elements of Statistical Learning 2nd ed., vol. 27, no. 2. 2009.

[27]. Ho, T. Kam, “Random Decision Forest,” in Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 1995, pp. 278–282.

[28]. C. Nguyen, Y. Wang, and H. N. Nguyen, “Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic,” J. Biomed. Sci. Eng., vol. 06, no. 05, 2013, doi: 10.4236/jbise.2013.65070.

[29]. S. E. Robertson and K. S. Jones, “Relevance weighting of search terms,” J. Am. Soc. Inf. Sci., vol. 27, no. 3, 1976, doi: 10.1002/asi.4630270302.

[30]. V. B. S. Prasath et al., “Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier -- A Review,” Aug. 2017, doi: 10.1089/big.2018.0175.

[31]. “10-fold Crossvalidation,” OpenML. https://www.openml.org/a/estimation-procedures/1 (accessed Apr. 12, 2019).

[32]. “10-fold Crossvalidation,” ResearchGate. https://www.researchgate.net/figure/10-fold-cross-validation-procedure_fig3_239386696 (accessed Apr. 12, 2019).

[33]. “Classification Report,” Yellowbrick, 2016. https://www.scikit-yb.org/en/latest/api/classifier/classification_report.html (accessed Apr. 28, 2019).

[34]. “Confusion Matrix in Machine Learning,” GeeksforGeeks. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/ (accessed May 28, 2019).

[35]. K. Fortney, “Pre-Processing in Natural Language Machine Learning,” Towards Data Science, 2017. https://towardsdatascience.com/pre-processing-in-natural-language-machine-learning-898a84b8bd47 (accessed Jan. 05, 2020).

[36]. T. Joachims, “Text categorization with Support Vector Machines: Learning with many relevant features”, 2022.

[37]. T. Ahmed, A. Pelaez, and M. Ghassemi, “Sentiment analysis of IMDb movie reviews,” Semant. Sch., pp. 1–7, 2015.

[38]. I. B. N. Sanditya Hardaya, A. Dhini, and I. Surjandari, “Application of text mining for classification of community complaints and proposals,” in 2017 3rd International Conference on Science in Information Technology (ICSITech), Oct. 2017, pp. 144–149. doi: 10.1109/ICSITech.2017.8257100.

[39]. Kulkarni, C.S., Bhavsar, A.U., Pingale, S.R. and Kumbhar, S.S, “BANK CHAT BOT – An Intelligent Assistant System Using NLP and Machine Learning,” International Research Journal of Engineering and Technology, vol. 04 , no. 05 , May -2017

[40]. Tutika, A. and Nagesh, M.Y.V., "Restaurant reviews classification using NLP Techniques," Journal of Information and Computational Science, vol. 9, no. 11, 2019.

[41]. Barbosa, L., Filgueiras, J., Rocha, G., Cardoso, H.L., Reis, L.P., Machado, J.P., Caldeira, A.C. and Oliveira, A.M., "Automatic Identification of Economic Activities in Complaints," in In International Conference on Statistical Language and Speech Processing, 2019, October. HJHK

[42]. Dien, T.T., Loc, B.H. and Thai-Nghe, N., "Article classification using natural language processing and machine learning," in International Conference on Advanced Computing and Applications (ACOMP) , 2019, November.

[43]. Liu, H., Yin, Q. and Wang, W.Y., 2018. Towards explainable NLP: A generative explanation framework for text classification. arXiv preprint arXiv:1811.00196.

[44]. Vijayaraghavan, S., Wang, Y., Guo, Z., Voong, J., Xu, W., Nasseri, A., Cai, J., Li, L., Vuong, K., and Wadhwa, E., 2020. Fake news detection with different models. arXiv preprint arXiv:2003.04978.

[45]. Razno, M., "Machine learning text classification model with NLP approach," Computational Linguistics and Intelligent Systems, vol. 2, pp. 71-73, 2019.

[46]. Polyakov, E.V., Voskov, L.S., Abramov, P.S. and Polyakov, S.V., "Generalized approach to sentiment analysis of short text messages in natural language processing," Информационно-управляющие системы, no. 1, pp. 2-14, 2020. HGHVGH

[47]. "Python Programming Tutorials," Pythonprogramming.net. https://pythonprogramming.net/stop-words-nltk-tutorial/ (accessed Feb. 15, 2020).

[48]. F. Sebastiani, "Machine learning in automated text categorization," ACM Comput. Surv., vol. 34, no. 1, pp. 1–47, 2002.

[49]. T. Mikolov, W. T. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," 2013.

[50]. "complaint-content-classification-nlp/5_model_refinement.ipynb at main halpert3/complaint-content-classification-nlp," GitHub. https://github.com/halpert3/complaint-content-classification-nlp/blob/main/notebooks/5_model_refinement.ipynb (accessed Jul. 20, 2021).