Optimized Machine Learning Framework for SMS Spam Detection and Classification:A Comparative Evaluation

Firas Zawaideh[1], Qusay Bsoul[2] , Ala Alzoubi[3] , Nardine T. Botros[4] , Moaz T. Fawzy[4], Diaa Salama AbdElminaam[5,6] , Nour Mostafa[7,*]

[1]Cybersecurity Department, Faculty of Science and Information Technology, Jadara University, Irbid, Jordan
[2]Cybersecurity Department, College of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan
[3]Faculty of Information Technology, Applied Science Private University, Amman, Jordan
[4]Faculty of Engineering , Misr International University , cairo , Egypt
[5]MEU Research Unit, Middle East University, Amman, Jordan
[6]Jadara Research Center, Jadara University, Irbid, Jordan
[7]College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait

Emails: F.zawaideh@jadara.edu.jo;q.bsoul@aau.edu.jo; a_alzoubi@asu.edu.jo; nardine1908660@miuegypt.edu.eg; moaz1916682@miuegypt.edu.eg; diaa.salama@miuegypt.edu.eg; nour.moustafa@aum.edu.kw

Abstract

The proliferation of spam messages via SMS poses significant challenges to digital communication security. This paper presents an optimized framework for detecting SMS spam using advanced machine learning algorithms and natural language processing (NLP) techniques. Two datasets, the Filtering Mobile Phone Spam Dataset and the SMS Spam Collection Dataset, were utilized to evaluate the performance of various classifiers, including Multinomial Naive Bayes, K-Nearest Neighbors, Support Vector Classifier, Decision Trees, and AdaBoost. The methodology encompasses comprehensive data preprocessing steps, such as tokenization, stopword removal, and text normalization, followed by feature extraction using TF-IDF and Bag-of-Words models. The classifiers' performances were evaluated using accuracy, precision, recall, and F1-score, alongside cross-validation techniques. Results indicate that Support Vector Classifier and AdaBoost consistently achieved superior accuracy in distinguishing between spam and ham messages. The study underscores the importance of data preprocessing and model optimization in enhancing spam detection accuracy, offering valuable insights for improving SMS filtering systems in cybersecurity applications.

Keywords: SMS spam detection; Machine learning classifiers; Natural Language Processing (NLP); Feature extraction techniques; Naive Bayes classifier; Support Vector Classifier (SVC); Bag of Words (BoW) model; Spam vs Ham classification; Hyperparameter optimization

1 introduction

At present, there are more than 8.58 billion cellular phones in use worldwide. In the pursuit of effective communication, two prevailing methods have emerged: voice calls and text messaging, commonly known as SMS, facilitated by cellular communication systems. Communication encompasses a variety of forms and objectives, extending beyond interpersonal interactions. Numerous companies and organizations employ SMS

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

as a means to engage and communicate with their clients, customers, or targeted segments of the population, irrespective of whether it involves announcements or advertisements, owing to its expediency and widespread accessibility. Despite the perception of SMS as a somewhat antiquated technology, it continues to function adeptly, thus ensuring its relevance and sustainability.

However, this convenience comes at the cost of compromised security, making it a vulnerable domain susceptible to exploitation by hackers and spammers who employ malicious links to exploit, blackmail, and threaten innocent people. Consequently, it becomes imperative to impose restrictions and implement systems to detect and avert the transmission of spam messages to end users. To achieve this objective, the development of a system capable of distinguishing between spam and non-spam messages (referred to as "HAM") becomes essential. Remarkably, several systems and applications have emerged that primarily focus on detecting spam messages through the utilization of Machine Learning (ML) , operating within the domain of Natural Language Processing (NLP).

The integration of Machine Learning (ML) techniques within the domain of Natural Language Processing (NLP) has revolutionized the field of language analysis and comprehension. When taught on large amounts of textual data, machine learning (ML) algorithms may identify complex patterns and correlations in language, allowing automated systems to carry out difficult language-related tasks. One key use of machine learning (ML) in natural language processing (NLP) is sentiment analysis, which enables companies to automatically categorize text as positive, negative, or neutral. This allows for the extraction of insightful data from social media posts and consumer feedback. Moreover, machine learning algorithms have made it easier to create reliable machine translation systems, which enable the correct and smooth translation of text between languages. Through the utilization of machine learning (ML) to interpret and produce text that resembles human language, chatbots and virtual assistants with natural language processing (NLP) skills can converse with users in an efficient manner and offer personalized support and information. The aforementioned examples highlight the exceptional potential of machine learning in natural language processing (NLP), opening doors for developments in a range of fields such as text summarization, content classification, and information extraction. .

The objective of this paper is to modify and enhance the detection of SPAM messages by using Machine learning algorithms through appropriate data processing and removing stopwords. Besides using Term Frequency Inverse Document Frequency (TF-IDF) vectorizing and Bag-of-Words (BoW) models as future extraction. In addition, using Naïve Bayes, KNN, SVC, Decision Tree, and Boosting classifiers to demonstrate the results and efficiency.

As a summary, our contributions are as follows:

- The implementation starts with collecting SMS data, whether SPAM or HAM, to provide a well-qualified system to detect spam messages.
- An extensive preprocessing stage guarantees the improvement of the dataset by thoroughly enhancing it, cleaning the data, managing any missing values, checking for nulls, encoding, converting messages into lowercase, and removing punctuations and stopwords. This process aims to optimize the datasets for predictive modeling.
- Two distinct strategies, 10-fold cross-validation and a holdout method, are employed for tailored data splitting based on algorithm characteristics, optimizing performance and generalization.
- Each data set is processed by two different, sufficient methods. Porter stemmer and TfidVectorizer are used in method 1 processing while Lemmitization and Bag of words are used in method 2 preprocessing.
- Enhance and optimize the model's performance by using 5-fold cross-validation which is a sufficient strategy for customizing data splitting according to algorithm characteristics.
- Classifiers play a main role in identifying and improving the model's performance. The variety of applied classifiers which are KNN, SVC, Naive Bayes, Adaboost, and Decision tree demonstrates the output distinctly.
- The final stage measures performance, prediction, and evaluation by utilizing performance metrics such as accuracy, F1-score, precision, and recall. This approach provides quantitative insights into the strengths and weaknesses of each algorithm.

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

The rest of the paper is organized as follows: In Section 2, an overview of related work is presented. Section 3 introduces the proposed methodology for the SMS SPAM Detection System. Subsequently, Section 3.1 focuses on the critical phase of data collection. In Section 3.2, The paper explores the complexities of data preprocessing in depth. Following this, Section 3.3 provides insights into the data-splitting process. Moving forward, Section 3.6 explores the recommendation models based on ML algorithms. The subsequent section, Section 3.7, highlights the pivotal role of performance metrics in evaluating the efficacy of the recommendation systems. In Section 4, Illustrating an analysis of the performance of recommendation systems. Finally, Section 5 offers a conclusion and outlines avenues for future work.

2 related work

In this section we want to discuss the problem statement from previous papers which published in this topic and see what is the methodology used to solve the problem and how many data set used and which take the best result also will see the comparison between the methodologies in each research and what is the best methodology to give us the best accuracy, so let's go see the previous solutions for this problem.

In[1] ,the author discusses the problem of email spam which is using for illegal and unethical activities for example fraud and spreading.so, the paper aims to solve this problem and increasing the accuracy of spam detection by using machine learning algorithms to find the spam emails and these will happen by using different algorithms of machine learning such as Naïve Bayes, support vector machine-nearest neighbor, random forest, bagging and boosting. But before applying the algorithms author do text assessment of email content so he uses different methods like text analysis, white and blacklists of domain names and community-based techniques. The paper use two data set for train and test the model which are "spam.csv" and "email.csv". after applying the all classifier naïve bayes was the best scores and results.

In,[?] The author in these paper work on spam but in the IoT devices so he discuss the web spam problem. The paper aim to ensure IoT devices is free from spam. The author use 5 machine learning models to do that which is Bagged, Bayesian Generalized Linear, Boosted Linear , eXtreme Gradient Boosting and Generalized Lin ear Model with Stepwise Feature Selection. The paper also use REFIT smart data set which is smart home data set sponsored by Loughborough University this data set collected from 20 homes and this survey spanned 18 months. After the train and test we found the best result from model 5 which is Generalized Lin ear Model with Stepwise Feature Selection with accuracy 91.8

In[2] The paper discuss the problem of spam emails which is not only unwanted message but also contain malicious content so the paper aim to develop effective spam filtering using machine learning to protect users from these harmful emails. The researcher use many methods in these papers categorized this techniques into different groups and compare between the difference techniques used various metrics such as accuracy ,precision and recall. The paper use different datasets for train and test specially "ECML" and UCI datasets to train the model also use Enron dataset which contain a collection of emails used for test the model. The result for the algorithms was Hijawi et al achieved 99.3 spam detection using random forest algorithm on 6050 emails additionally Banday and Jan achieved 96.69 accuracy using SVM by using 8000 spam emails and the lowest accuracy was Verma and Sofat which achieved 89 on the Enron dataset

In[3]The author addressed the problem of spam in the email and IoT platforms and solved these problem by using machine learning algorithm different techniques such as SVM , MLP , Naïve Bayes , random forest , decision tree , KNN , additive regression and logistic regression then make a comparison for these techniques based on accuracy , precision and recall. Paper use "ECML" dataset and the UCI dataset and the model of MLP achieved the best result with accuracy 99 .

In[4]The purpose of this research study is to detect spam emails using machine learning techniques. For this study, a sample containing 5728 emails from kaggle.com was used as the basis for training different classifiers including the Naive Bayes classifier, the K-Nearest Neighbors (k-NN) algorithm as well as Support Vector Machine (SVM), Logistic Regression, Decision Tree and Random Forests. As a result of the experimental tests, the Naive Bayes algorithm predicts the spammer email with the highest accuracy of 99, precision of 97, recall of 99, F-measure of 98, and ROC Area 1.00. In addition, The performance of Logistic Regression is

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

also similar to Naive Bayes with an accuracy of 99. However, the K-Nearest Neighbors provides an accuracy of 90, Decision Tree 94, SVM 98, and Random Forest 95

In[5]This research work used two datasets for spam detection and sentiment analysis on Twitter. The results for spam detection showed that the Multinomial Naïve Bayes classifier performed well with a classification accuracy of 97.78 percent while the LSTM deep learning model performed well with a validation accuracy of 98.74. Sentiment analysis was better done by a Support Vector Machine classifier which yielded an accuracy rate of 70.56; however, the LSTM algorithm was the most accurate as it achieved 73.81 classification accuracy on the validation dataset. This study has employed two classifiers and models which if combined can produce a real-time spam detection and sentiment analysis system. As a result, the outcomes of this research have demonstrated that it is possible to develop a new modern system for addressing social media challenges.

In[6]The author discuss the lack of applicable twitter spam detection so the paper focus to solve this problem by using real-world datasets. The paper use machine learning algorithm to solve this problem such as KNN , K-KNN , Naive Bayes , Boosted Logistic Regression and Random Forest. The paper applying the study by using 20K tweets and extract account-based features to simplify spam detection. Also use 1.5 million public tweets to train the models. After divided the datasets to train test and validation and applying the datasets to different classifier , after comparing all results the best accuracy was from random forest model.

In[7]The author addressed the problem of spam which is using for illegal and unethical activities for example fraud and spreading also it threats to internet users. The paper aimed to solve this problem by using machine learning. The objective of this paper was evaluate five machine learning classifier which is logistic regression , decision tree , Naïve Bayes , random tree , SVM and KNN. The paper depend on train and test the model on UCI and then evaluate the different classifier using the weka tool and analyze various performance like accuracy , precision , recall ,F1 score. After these analysis the author found the best accuracy was from KNN and random tree achieved the same result.

In[8]The paper addressed identifying and classifying spam messages in sms by using machine learning techniques such as natural language processing (NLP) , feature engineering and supervised learning algorithms before chose the best model there are sum steps like data preprocessing , feature extraction , model training , evaluation and fine-tuning to determine the best performance. This paper doesn't mentioned the name of dataset used in the model and also the author doesn't discuss what is the best classifier.

In[9]The author discuss email spam which is not only unwanted message but also contain malicious content. The research focus on using machine learning techniques to develop a binary classifier which is can effectively distinguish between ham or spam. The paper use Azure-based platform to implement machine learning and compare these methods with different methods such as visual studio, hybrid analysis, and JoeSandbox cloud. The researchers improve the Vowpal Wabbit algorithm with Azure machine learning studio to detect spam or ham passed on classification model. The paper doesn't discus the datasets and also doesn't discuss the number of final accuracy.

In[10]The paper proposed that huge problems of SMS spam have been considered an important issue for users of the Internet since it is estimated that more than 60 of SMS traffic is spam, which is the main reason for bandwidth and storage overloads. This paper has developed a machine-learning-based system to filter SMS messages as spam for Indian users by merging Indian SMS data into the available global dataset. The SMS corpus is structured and analyzed by applying text-mining techniques. Further, different machine learning classifiers are experimented with the dataset. The results show that the machine learning approach is effective at detecting SMS spam. The Support Vector Machine (SVM) gives the best performance among all the experimented algorithms. This proves the possibility of dealing with the now fast-increasing SMS spam problem using text mining and machine learning.

In[11]The research article deals with spam in mobile message communication which focuses on its significant impact on security and performance. Many machine-learning algorithms such as logistic regression (LR), k-nearest neighbor (K-NN), and decision tree (DT) are used,in this paper, as classifiers to separate the spam and non-spam messages into different classes. A dataset of SMS spam messages is collected from Kaggle.com and it contains 5572 instances, 4900 non-spam messages, and 672 instances of spam messages and it is split into 70 for training and 30 for testing. The results indicate that logistic regression works with high performance rather than the other classifiers, achieving an accuracy of 99. This high level of accuracy shows the effectiveness of the proposed method in improving spam detection in mobile message communications.

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

3 Methodology

Figure 1 , and Algorithm 1 shows the scheme of work in six steps

Algorithm 1 Spam Detection Framework

Input: SMS dataset D, feature extraction method F, classifiers $\{C_1, C_2, ..., C_n\}$
Output: Classified messages as Spam or Ham
procedure DATA COLLECTION
    Gather SMS spam datasets, e.g., Filtering mobile phone spam dataset, SMS spam collection dataset
end procedure
procedure DATA PREPROCESSING
    Convert all text to lowercase
    Remove punctuation, digits, and non-word characters
    Tokenize text into individual words
    Remove stopwords (common words like ”the”, ”and”, etc.)
    Apply stemming (Porter Stemmer) or lemmatization based on method choice
end procedure
procedure FEATURE EXTRACTION
    Option 1: Apply TF-IDF Vectorizer
    Option 2: Apply Bag of Words model
end procedure
procedure DATA SPLITTING
    Split data into training set D_{train} and testing set D_{test} (e.g., 80/20 split)
end procedure
procedure MODEL TRAINING
    for each classifier $C_i \in \{C_1, C_2, ..., C_n\}$ **do**
        Train C_i on D_{train}
        Optimize hyperparameters using grid search
    end for
end procedure
procedure MODEL EVALUATION
    for each classifier $C_i \in \{C_1, C_2, ..., C_n\}$ **do**
        Evaluate C_i on D_{test} using metrics: accuracy, precision, recall, F1-score
    end for
end procedure
procedure CROSS VALIDATION
    Apply 10-fold cross-validation to each classifier C_i
end procedure
procedure MODEL SELECTION
    Select the classifier with the best performance based on the evaluation metrics
end procedure

- Data Collection
- Data Preprocessing
- Data Splitting
- Optimization for Machine Learning (ML)
- Recommendation based on Machine Learning
- Predictions and evaluations in form of metrics

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

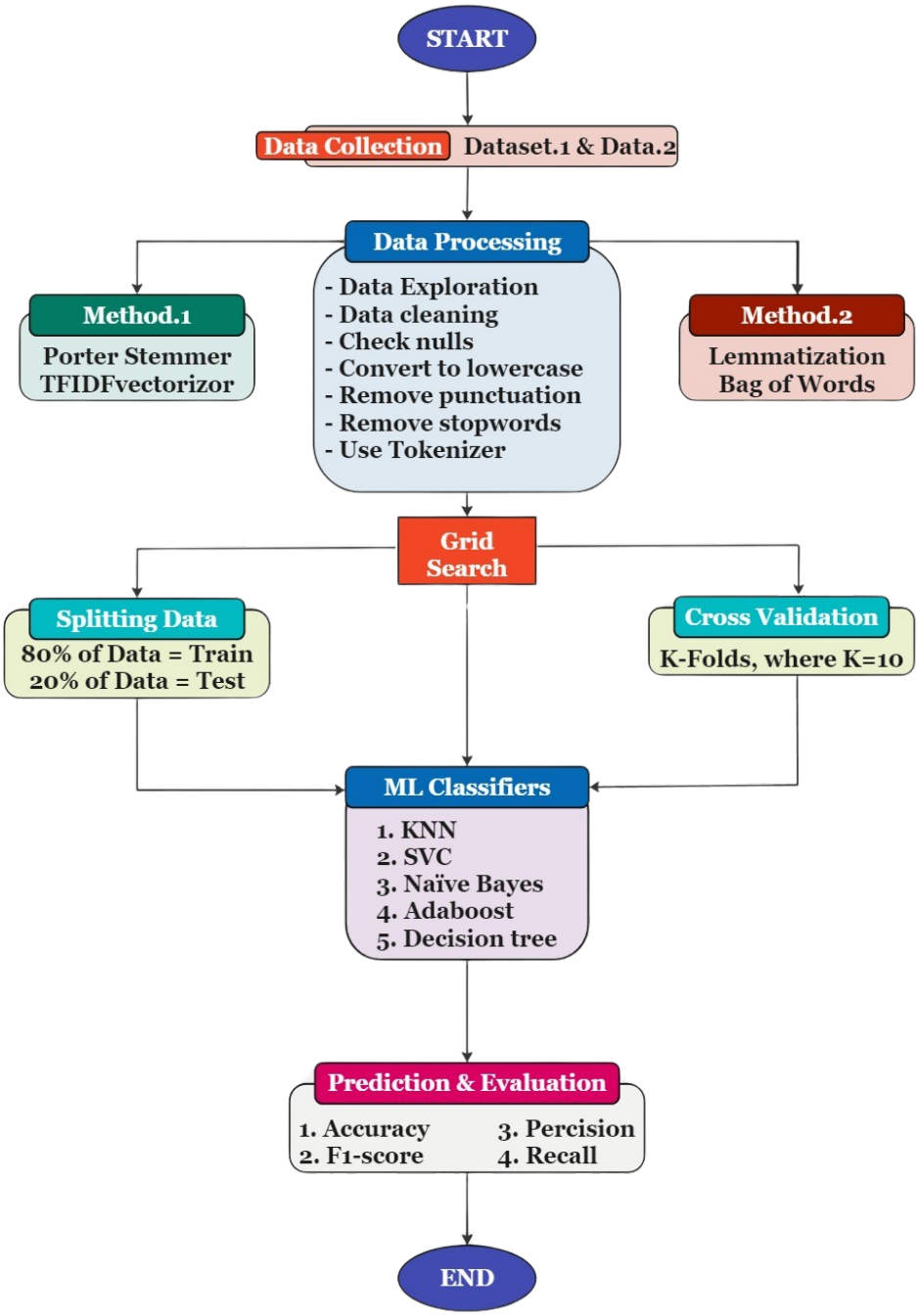

Figure 1: The flow of machine learning.

3.1 Data collection

The process of gathering, quantifying, and interpreting precise research insights through the application of established, verified methodologies is known as data collection. To put it concisely, data collection is the act of obtaining information for a particular goal. It can be applied to improve goods and services, find answers to research issues, and make well-informed business decisions. Prior to gathering data, we must decide what information is required and how it will be gathered. A hypothesis can also be assessed using the data that has been gathered. The most crucial and initial stage of research is typically data collection.It depends from the information that we collected, we find different fields of use for data collection. These datasets (SMS Spam collection, Filtering mobile phone spam)provide us huge amount of SMS messages.

Table 1 provides detailed information about the datasets used in our analysis.

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

Table 1: Datasets Information

Dataset Name Size	Year	Source
Filtering mobile phone spam 480.88 kB	2018	Kaggle
SMS Spam collection 483.7 kB	2017	Kaggle

1. **Filtering mobile phone spam**:[?] SMS 425 spam messages were manually taken from the Grumbletext website. mobile phone users publicly report receiving spam SMS messages; most of messages do so without disclosing the specific spam message they received. It took a lot of effort and time that identify the words of the spam messages in the claims, which required closely going through hundreds of web sites. a subset of 3,375 SMS ham messages that were selected at random, a corpus of roughly 10,000 authentic messages gathered for documentation at the National University of Singapore's Department of Computer Science.
2. **SMS Spam Collection (spam or legitimate) Dataset** [?] : It is collection of messages in form of SMS gathered for SMS spam research. It has a collection of 5,574 English SMS messages and it has spam or Ham messages.Mobile phone users publicly report receiving spam SMS messages; most of them do so without disclosing the specific spam message they received. It took a lot of effort and time to identify the text of the spam messages in the claims, which required closely going through hundreds of web sites.A subset of 3,375 SMS ham messages that were selected at random from sources

3.2 Data Preprocessing

Data preprocessing is the process of producing raw data for machine learning models. This is the first step for building a machine learning model. This step is the most difficult and time-consuming part of data science. Data preprocessing is so much mean in machine learning algorithms to be able to reduce its complexity. The approach is applied to for all datasets.

- **Data Exploration:** We began by carefully examining both datasets to learn more about their characteristics, and any difficulties. which help in preprocessing process in future, and closely studied the general characteristics of the data.
  - In dataset one: Filtering mobile phone spam includes 747 spam messages ,and 5572 ham messages, as shown below(Figure 2).
  - In dataset two: SMS Spam Collection includes 747 spam messages ,and 5559 ham messages, as shown below(Figure 3).
- **Data Cleaning:** Taking care of abnormalities, inconsistencies, and mistakes in the datasets was known as data cleaning. This process included things like testing the coherence of the information, standardizing formats, and fixing data types. We refined the datasets for precise and insightful analysis by resolving inconsistencies.
- **Checking Nulls:** Machine learning frequently encounters missing values. This happens when a certain variable has insufficient data points, which leads to imprecise information and might compromise the precision and stability of your models. For your machine learning initiatives to produce reliable and unbiased results.

  For instance , We checked if there is nulls in both datasets and we did not find any nulls in datasets to improve our model and increasing accurices.
- **Lowercase conversion:** One of the most popular Python text preparation procedures is changing the text's case, preferably to lower case. However, since lower casing can cause information loss in some NLP situations, you do not have to perform this step each time you work on an NLP problem.

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

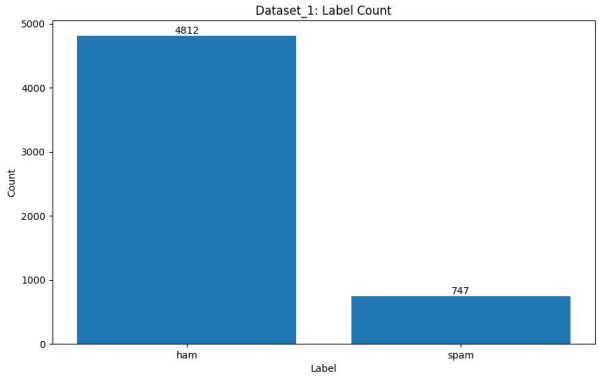

Figure 2: Dataset 1

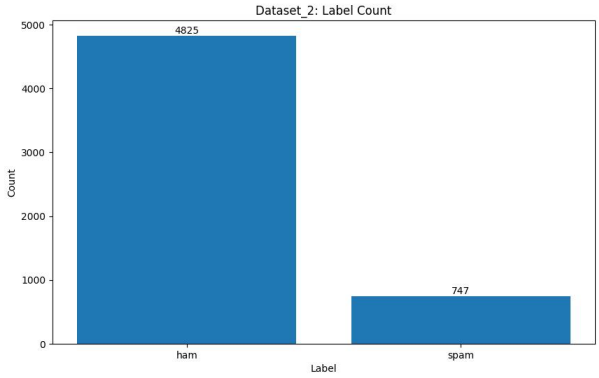

Figure 3: Dataset 2

- **Removing punctuation:** removing the extra spaces, digits and non word characters like punctuation, ascii etc. Avoiding information loss in some NLP situations.
- **Tokenizer:** As inclued the fields of Natural Language Processing (NLP) and machine learning techinques, tokenization is the methods that can break a squence of text into a smaller parts called tokens. These tokens can range in length from words to characters. This procedure is important primarily because it facilitates computer comprehension of human language by reducing it to smaller, more manageable chunks for analysis. Words or sub-words are commonly used as tokens in natural language processing. For many NLP activities, such as text processing, language modeling, and machine translation, tokenization is an essential first step
- **Stopwords:** The most frequently recurring words in any natural language are referred to as stopwords. These stopwords may not significantly alter the text's meaning for analyzing text data and creating NLP models. As a result, eliminating stopwords can increase the accuracy of our analysis by allowing us to concentrate on the text's most crucial information. Eliminating stopwords can have the benefit of shrinking the dataset, It decreases the time needed for natural language processing models in trainig phase.
- **Porter Stemming:** Algorithms known as stemmers are used to combine several word forms into one basic form. In essence, they accomplish this by cutting certain character strings from word token ends. Stemmers do not take prefixes into consideration. The majority of stemmers have a list of frequently used, language-specific suffixes that the algorithm compares input word tokens to. If a word in the algorithm matches one of the suffixes and removing the suffix does not break any of the program's predetermined guidelines. we applied it in our datasets as method 1.
- **lemmatization :** In natural language processing (NLP), lemmatization techniques refer to ways to recognize and convert words into their lemmas, or base or root forms. By aiding in text normalization, these

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

techniques enable more precise language processing and analysis across a range of NLP applications. There are three categories of lemmatization methods.The first is rule based lemmatization, a word's base or root form is obtained by applying predetermined rules. Rules and patterns in language are the foundation of rule-based lemmatization, as opposed to machine learning-based techniques that learn from data. The second is Dictionary-based lemmatization, which maps words to their matching base forms or lemmas using predetermined dictionaries or lookup tables, is the second type. To determine a word's lemma, each entry in the dictionary is compared. This approach works effectively for languages with clear rules. The third is machine learning-based lemmatization, which uses computer models to automatically figure out how words relate to each other in their simplest forms. Machine learning methods, like neural networks or statistical models, are trained on huge text datasets to generalize patterns, in unlike rule-based or dictionary-based approaches.we applied it in our datasets as method 2.

- **TF-IDF Vectorizer:** The text will be converted by TF-IDF into a meaningful representation of integers or numbers that can be used to fit predictions made by a machine learning system. By comparing the frequency of word that appears in text with the total number in papers. the TF-IDF Vectorizer calculates a word's originality. The TfidfVectorizer helps algorithms use word importance to predict outcomes by converting documents into a matrix of TF-IDF in form of features.we applied it in our datasets as method 1.
- **Bag of Words:** It is strategy that convert text document into numbers. This method works by converting text into vectors that depends on the frequency of words, without taking in consider the order or context of the words.we applied it in our datasets as method 2.
- **SMOTE** One of the most popular oversampling techniques for resolving imbalance issues is SMOTE (synthetic minority oversampling technique). By duplicating minority class cases at random, it seeks to achieve class distribution balance.
- **Encoding:** In order to transform categorical data into a format appropriate for model training, we used encoding techniques, as many machine learning algorithms demand numerical inputs. This stage makes sure that the information included in these variables can be interpreted and learned from by the algorithms in an efficient manner.label encoding is a method that we used for converting categorical variables into numerical notation.

3.3 Data splitting

It is one of methods that used for assessing machine learning performances that depends on splitting dataset,It works with any supervised learning method and may be applied to regression or classification tasks. The process includes dividing a dataset into two smaller groups. The first part is known as the training dataset. It is utlized to make it to fit the model. Instead of using the second part is used to train model, the input element is the model that can make predictions and compares them to the target values values. Fits the machine learning model with the help of the train dataset. Test Dataset: A machine learning model's fit is assessed using this dataset. We can divide our dataset into parts a testing and a training . To be able to do this, picking a random sample of approximately 80 percent of the rows and add them to your training set without replacing them. we add the remaining 30 percent to your test set.

3.4 Hyperparameters Optimization Methods

The process of determining the ideal hyperparameter settings for a machine learning system to attain peak performance is known as hyperparameter optimization. This is one of the trickiest parts of creating machine learning models, but it's essential for precise forecasts. Optimising default hyperparameter settings is necessary to achieve the best outcomes as they are not always suitable for a variety of applications. Hyperparameter optimization techniques were used in your project to improve the performance of the model. .Grid Search technique is used to find the optimal value for each parameter of Machine Leaning Model (Support Vector Classifier-SVC) described as:

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

- **Grid search** Grid Search is a crucial machine learning technique because it makes it possible to optimize hyper-parameters, which have a significant impact on a model's performance. Grid Search enhances the accuracy and generality of a model by determining the optimal set of hyper-parameters, leading to more accurate classifications.

3.5 Cross Validation

Cross validation is one way to evaluate the performance of a machine learning model. The basic idea of cross validation is to separate the data into training and testing sets, train the model on the training set, then evaluate its performance on the validation set. This method is repeated several times with different subsets of the training and testing data to get the average performance. The cross validation method that is most frequently used is k-fold cross validation. K subsets of the data are created for k-fold cross validation. Then, using one subset as the testing set and the remaining subsets as the training set, the model is trained and assessed k times. Next, the average performance over the k data points is used to estimate the model's performance..We have applied 10-fold validation technique on each Machine learning Model. as shown below(Figure 4).

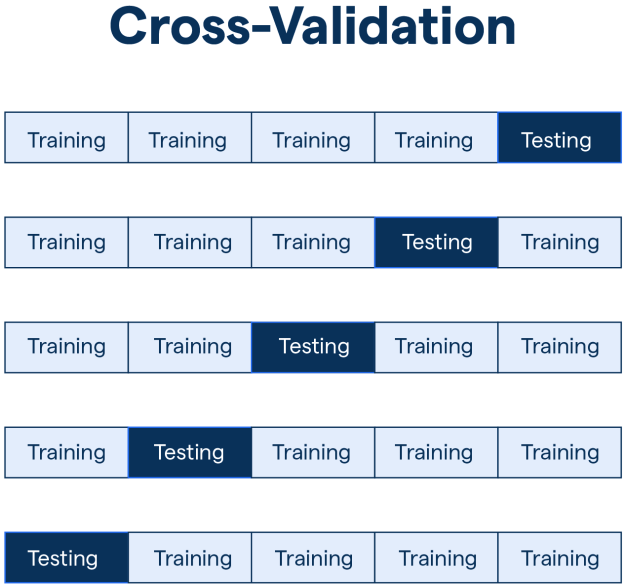

Figure 4: K-Fold

3.6 Recommendation bssed on ML

One kind of machine learning system that offers customers individualized suggestions based on their prior behaviours, preferences, and patterns is called a recommendation system. It is a sub-type of information filtering systems that suggests products to consumers based on their browsing habits or interest using algorithms. In this section, we used 5 machine learning models Multinomial Naive Bayes, K-Nearest Neighbors (KNN), Support vector classifer (SVC), Decision Tree ,and Ada boost.

- **Machine Learning Models**
  - **Multinomial Naive Bayes[?]**
    It is learning technique that depends on probability that is used in natural language processing (NLP). The method can detect the label of message like email or text nessage or articles, and is based Naive bayes method. It determines the lable from sample with the highest probability.

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

– **K-Nearest Neighbors (KNN)**[?] It is one of the most model in learning classification methods. It has high recommendations in recognizing patterns, for mining data, and intrusion detection. Its mean aim to have groups and measure the distance between data and classify them into classes by knowing the k-fold parameter.

Take into consideration the following table of data points, which has two characteristics, as an example. (Figure 5)

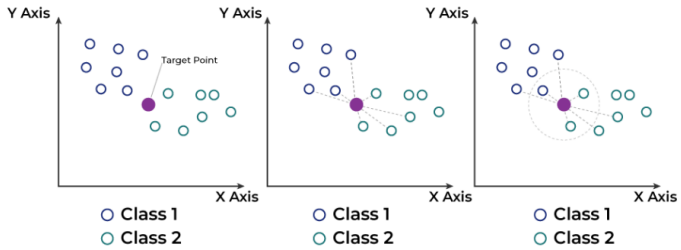

Figure 5: KNN

.

– **Support Vector Classifier**[?] The support vector Classifier (SVC) is a machine learning algorithm. It can determine the boundaries between data points based on labels,classes,and outputs. It is one of supervised learning models. It has the ability to solve difficult problems. SVCs are used in many parts like speech, natural language processing (NLP), healthcare, and signal processing applications. The SVC algorithm works by locating a hyperplane which is defined as line clearly divides data points into separate classes. The position of hyperplane is determinted to be able to where the classes being considered are separated by the greatest margin.(Figure 6)

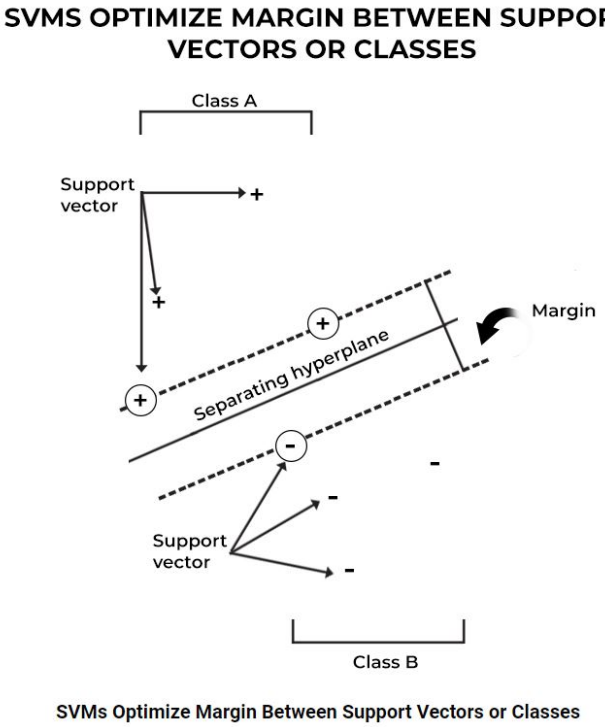

Figure 6: SVC

– **Decision Trees**[?] A decision tree is one of supervised learning algorithm that hepls in classification and regression process. It has a hierarchical tree structure that includes a root node, branches, internal nodes, and leaf nodes. Decision trees structure is determined from top to bottom. We find the roots in top and it is divied to roots to have many nodes, then this root is divided into several

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

nodes. It works like we have several if-else statment that check conditions if it is true or not then it goes to the following node. (Figure 7)

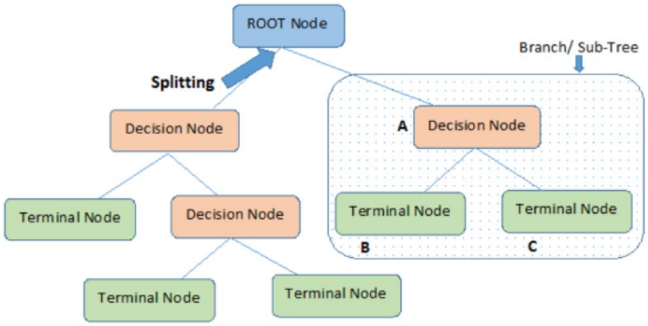

Figure 7: Decision tree

– **AdaBoost**[?] Adaptive Boosting, or AdaBoost, is an ensemble learning strategy that increases overall accuracy by combining several weak classifiers to build a strong classifier. The procedures that we followed are:
1-Setting Initial Weights: Give each training sample the same weight. 2-Make use of the weighted training data to train a weak classifier. Determine the classifier's weight and error to reduce the errors of dataset as much as possible. 3-Adjust sample weights. 4-Adjust weights to a normal value of one. 5-Final Prediction: Using a weighted of most votes, remove the results from every weak classifier. AdaBoost improves the performance of weak classifiers to produce a strong, accurate composite classifier by concentrating on difficult-to-classify data. (Figure 8)

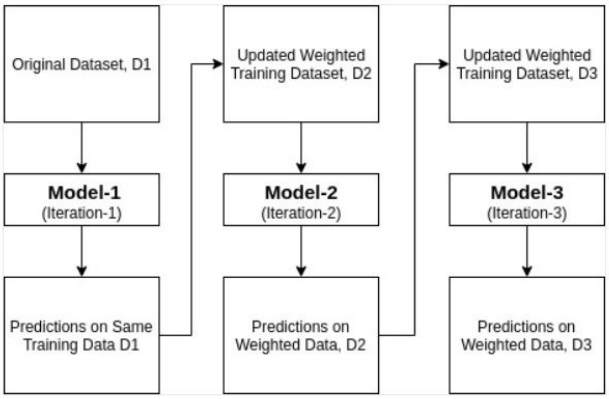

Figure 8: Ada-Boost

3.7 Performance metrics

In evaluating the performance of recommendation systems, We used accuracy,f1score,precision,Specificity, Sensitivity, recall,and Confusion Matrices.

- **Accuracy** Accuracy[?] is the most used performance measuring parameter for machine learning models is accuracy. Although it is a very simple indicator to measure, its trustworthiness can sometimes be challenging to determine depending on various conditions. It is employed in classification issues to determine whether percentage predictions made by a model are accurate. The equation for Accuracy is given by:

$$\text{Accuracy} = \frac{CorrectPredictions}{AllPredictions} \tag{1}$$

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

- **F1-score** F1-score[?] The F1 score can be like harmonic mean of both accuracy and recall, with a maximum score of 1 and a minimum score of 0. Recall and accuracy have equal parts to F1 score in terms of relative importance. The equation for F1-score is given by:

$$\text{F1-score} = \frac{2 * TP}{2 * TP + FP + FN} \tag{2}$$

  Where TP is the numbers of true positive, FN is the numbers of false negative,and FP is the numbers of false positive.

- **Precision** Precision[?] The precision of the model is determined by dividing all of its positive predictions by the percentage of genuine positive forecasts. The ratio of true positives to the total of true positives and false positives is used to compute it. The equation for Precision is given by:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3}$$

- **Recall** [?]Recall is a metric that is used to measure how often ML model identifies positive instances correct (true positives) out of all the actual positive samples in dataset. The equation for MAE is given by:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

  where TP represents True positive , FN represents False negative,

- **Sensitivity** [?] Sensitivity is used to measure how well Machine Learning Model avoids false negatives .The equation for MAE is given by:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{5}$$

  where TP represents True positive , FN represents False negative,

- **Specificity** [?] specificity is a measure of how can machine learning model can avoid false positive to show how will the model predtict true positives.The equation for specificity is:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{6}$$

  where TN represents True Negative , FP represents False Positive,

- **Confusion Matrix** [?] A confusion matrix provide the prediction results summary in form of matrix showing how many prediction is predicted by machine learning model are correct and incorrect per class, as shown below (Figure 9)

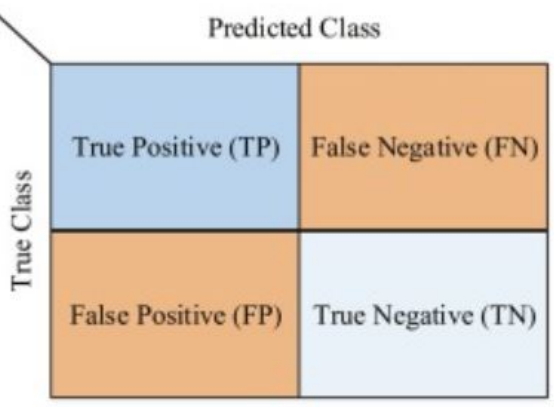

Figure 9: confusion matrix

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

4 Experimental Results and Discussion

In the experimental results and discussion phase, We entered the two datasets (Filtering mobile phone spam, SMS spam collections) into five machine learning models multinomial naive bayes,KNN,Decision,Ada boost,and SVC and we trained our model for these five machine learning models and we got high accuracy for each one .

4.1 Case Study I (Filtering mobile phone spam)

In this dataset we applied five classifiers which are Multinomial Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree,Support Vector Classifier and AdaBoost.

4.1.1 Method one

In method one, we applied Tokenizer and TF-IDE Vectorizer, as show below the evaluation.

- **KNN** the evaluation of KNN classifier includes confusion matrix as shown below (Figure 10) , also accuracy , f1 score, precision, Recall, shown below (Figure 11) and Cross validation shown below (Figure 12)

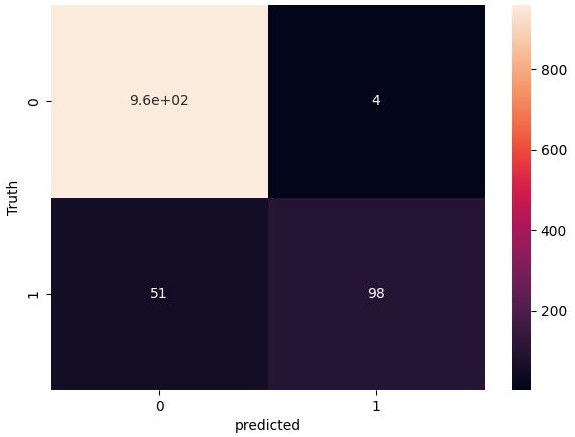

Figure 10: KNN confusion matrix

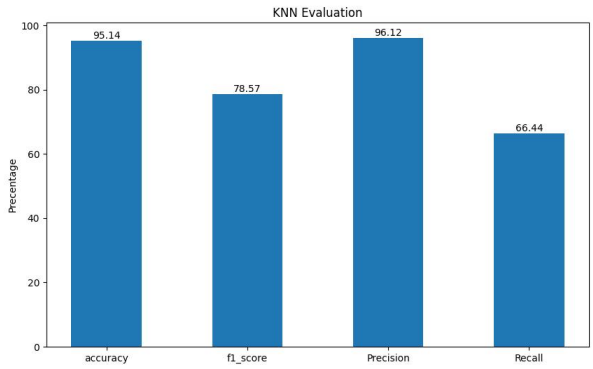

Figure 11: KNN Evaluation

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

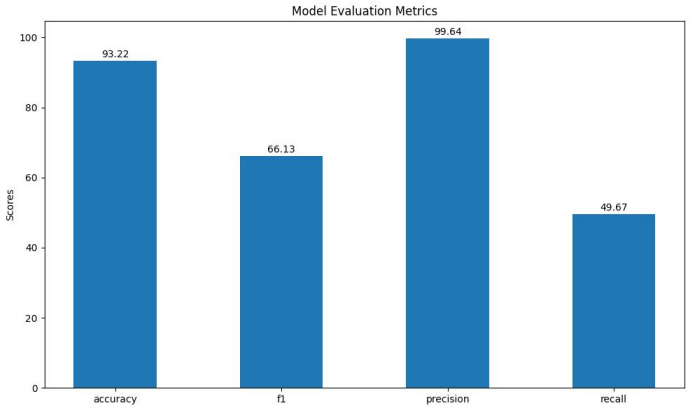

Figure 12: KNN Cross Validation

- **Support Vector Classifier** the evaluation of SVM classifier includes confusion matrix as shown below (Figure 13) , also accuracy , f1 score, precision, Recall, shown below (Figure 14) Cross validation as shown below (Figure 15)

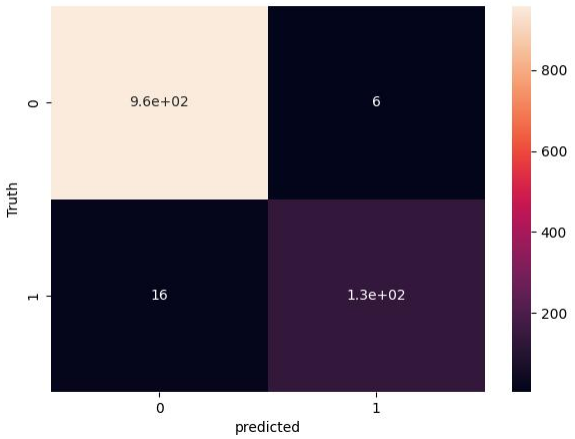

Figure 13: SVC confusion matrix

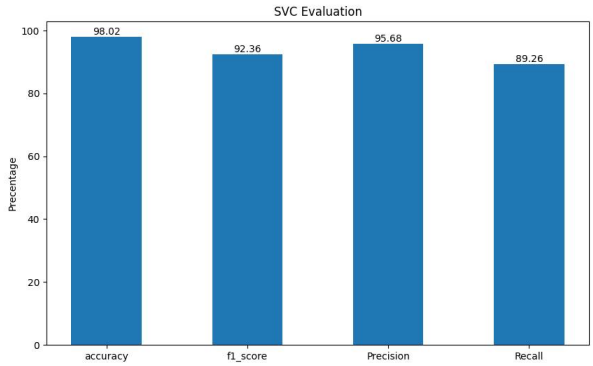

Figure 14: SVC Evaluation

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

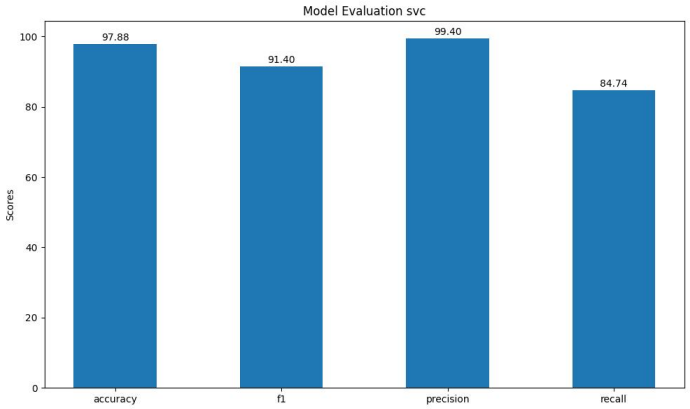

Figure 15: SVC Cross Validation

- **Decision Tree Classifier** The evaluation of Decision Tree classifier includes confusion matrix (Figure 16) , also accuracy , f1 score, precision, Recall, shown below (Figure 17) Cross validation as shown (Figure 18)

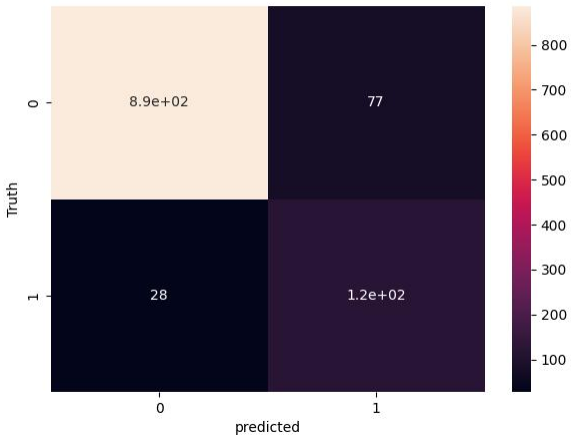

Figure 16: Decision Tree confusion matrix

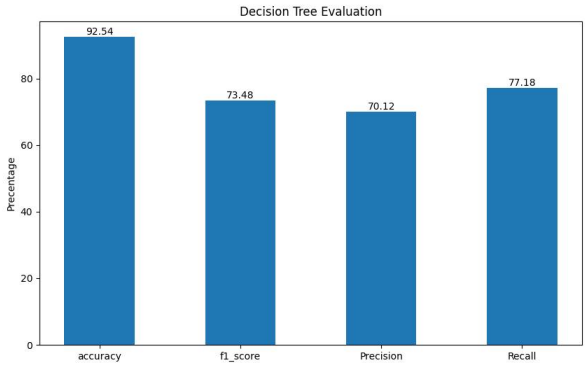

Figure 17: Decision Tree Evaluation

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

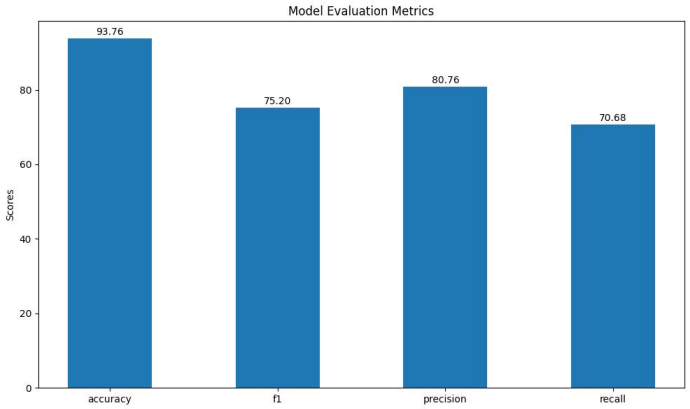

Figure 18: Decision Tree Cross Validation

- **Multinomial Naive Bayes** The evaluation of naive bayes classifier includes confusion matrix as shown below (Figure 19) , also accuracy , f1 score, precision, Recall, shown below (Figure 20) and Cross validation as shown below (Figure 21)

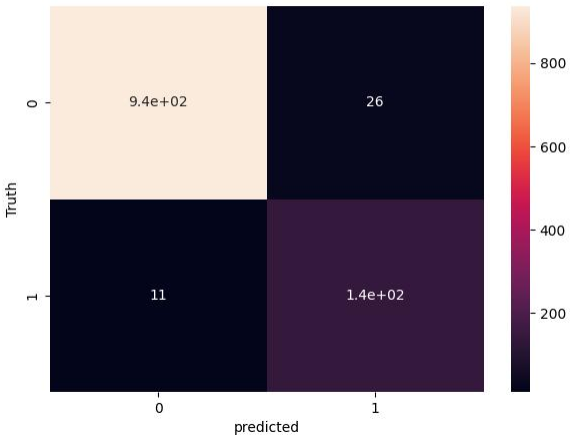

Figure 19: Naive Bayes confusion matrix

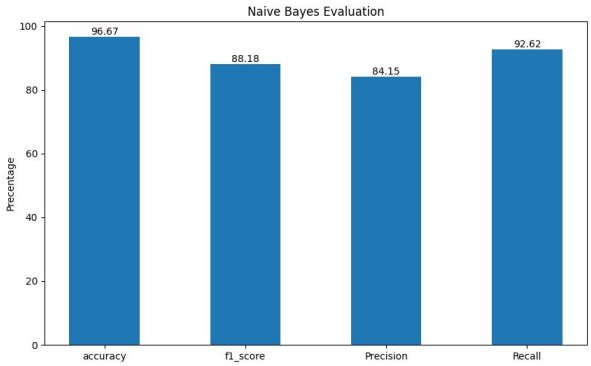

Figure 20: Naive Bayes Evaluation

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

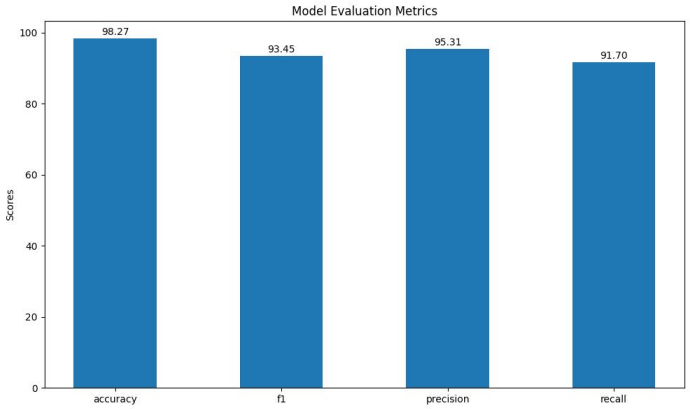

Figure 21: Naive Bayes Cross Validation

- **AdaBoost** The evaluation of AdaBoost classifier includes confusion matrix as shown below (Figure 22) , also accuracy , f1 score, precision, Recall, shown below (Figure 23) and Cross validation shown below (Figure 24)

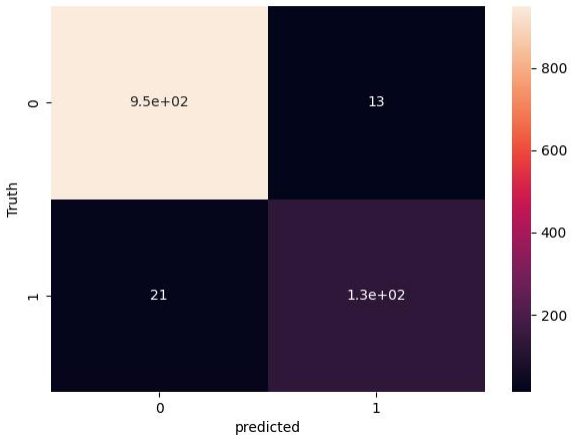

Figure 22: AdaBoost confusion matrix

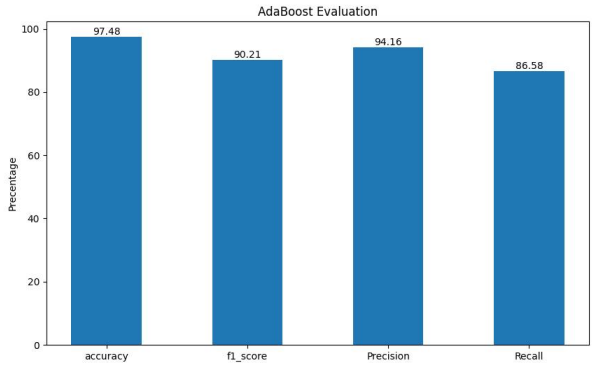

Figure 23: AdaBoost Evaluation

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

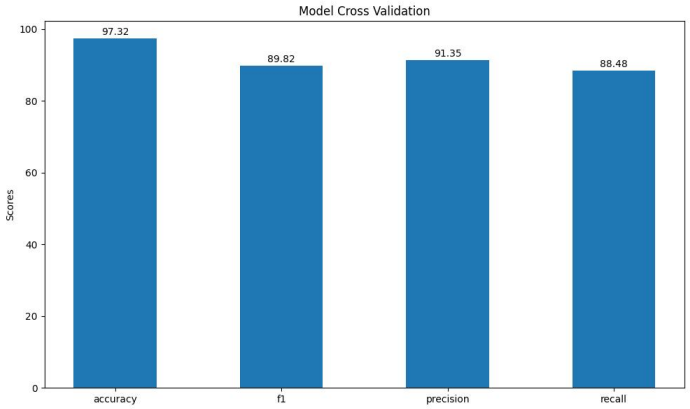

Figure 24: AdaBoost Cross Validation

4.1.2 Method Two

In method one, we applied lemmatization and Bag-of-Words (BoW) model, as show below the evaluation.

- **Multinomial Naive Bayes** the evaluation of Naive Bayes classifier includes confusion matrix as shown below (Figure 25) , also accuracy , f1 score, precision, Recall, shown below (Figure 26) and Cross validation shown below (Figure 27)

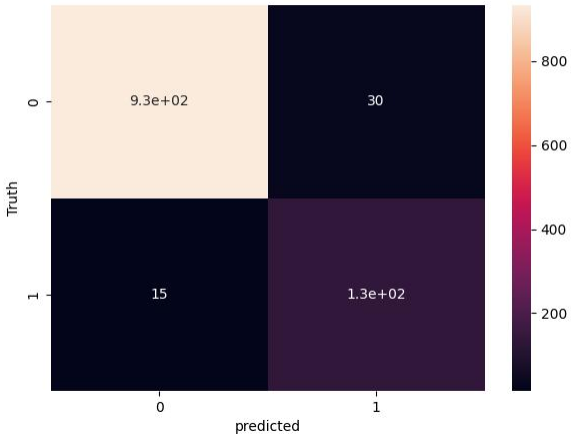

Figure 25: Naive Bayes confusion matrix

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

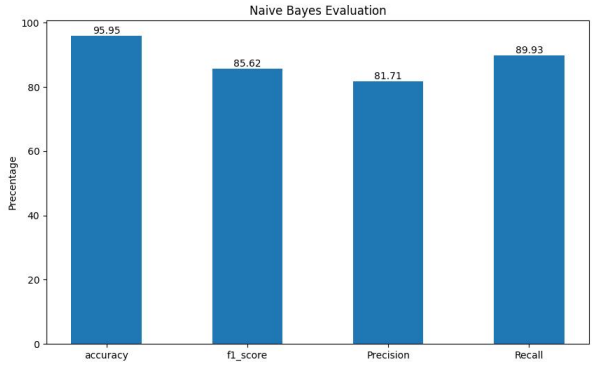

Figure 26: Naive Bayes Evaluation

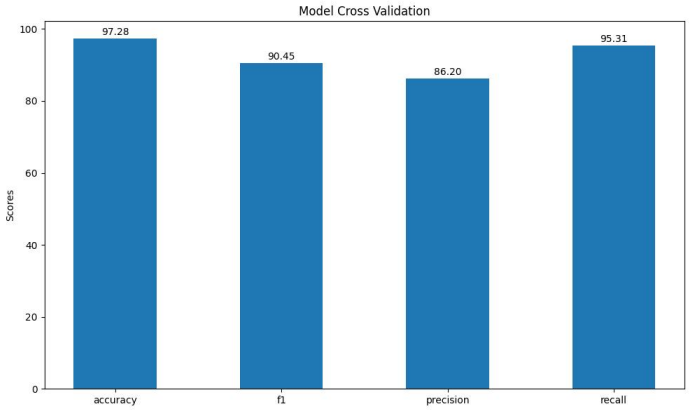

Figure 27: Naive Cross Validation

- **KNN** the evaluation of KNN classifier includes confusion matrix as shown below (Figure 28) , also accuracy , f1 score, precision, Recall, shown below (Figure 29) and Cross validation shown below (Figure 30)

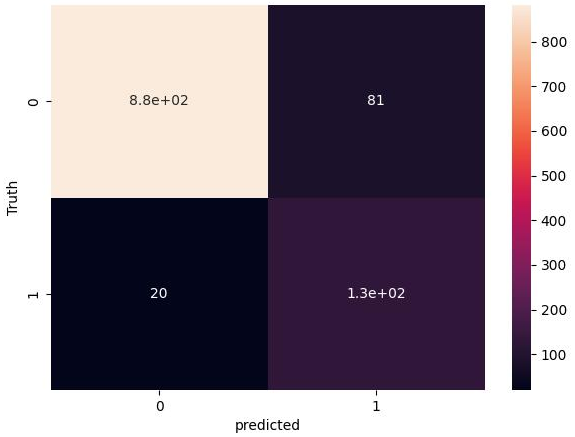

Figure 28: KNN Confusion Matrix

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

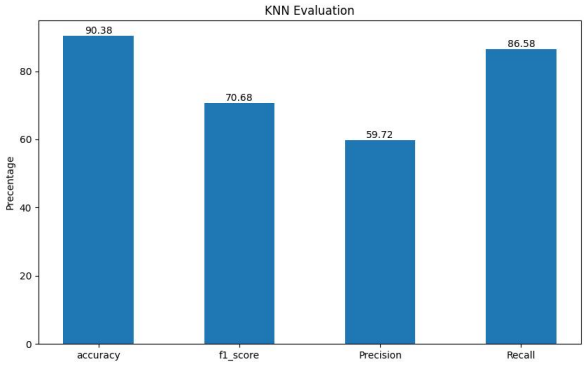

Figure 29: KNN Evaluation

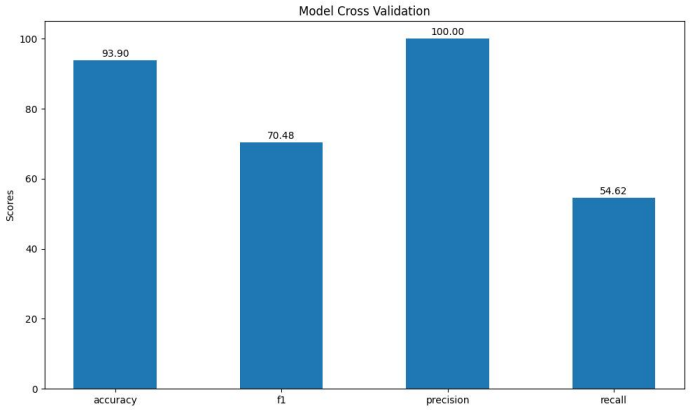

Figure 30: KNN Cross Validation

- **Decision Tree Classifier** The evaluation of Decision Tree classifier includes confusion matrix (Figure 31) , also accuracy , f1 score, precision, Recall, shown below (Figure 32) Cross validation as shown (Figure 33)

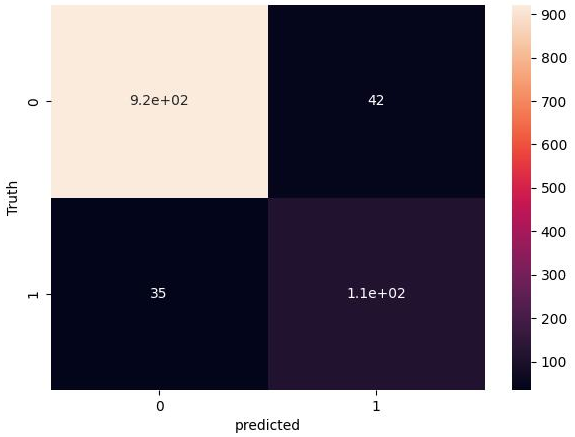

Figure 31: Decision Tree Confusion Matrix

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

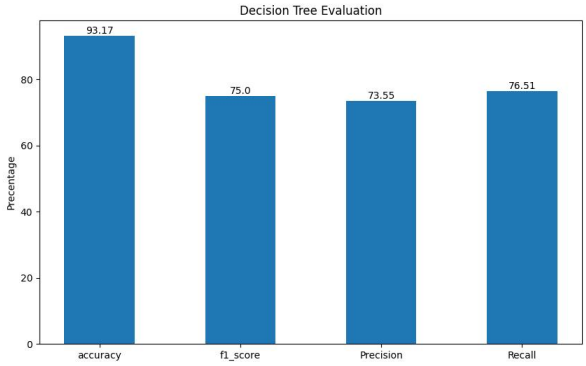

Figure 32: Decision Tree classifier Evaluation

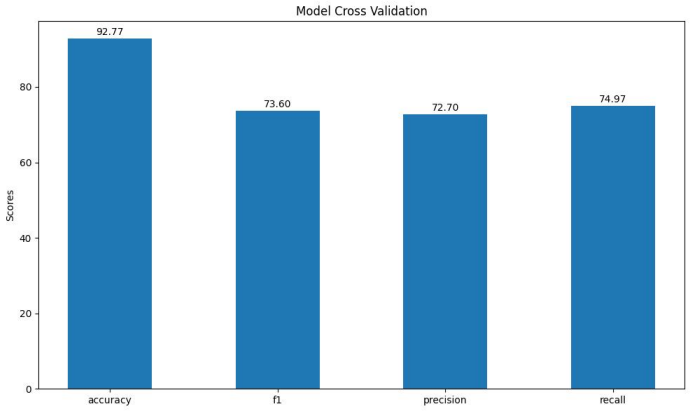

Figure 33: Decision Tree Cross Validation

- **Support Vector Classifier** The evaluation of SVC classifier includes confusion matrix as shown below (Figure 34) , also accuracy , f1 score, precision, Recall, shown below (Figure 35) Cross validation as shown below (Figure 36)

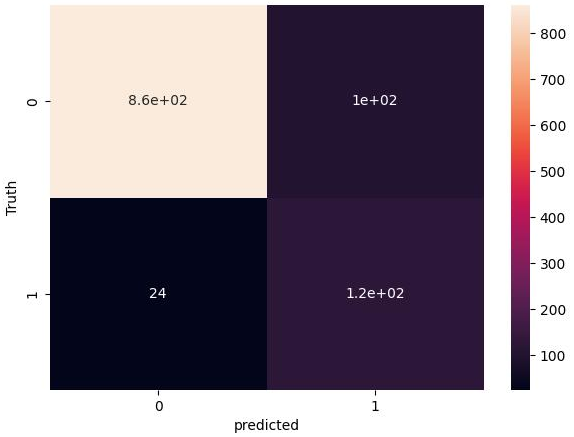

Figure 34: SVC Confusion Matrix

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

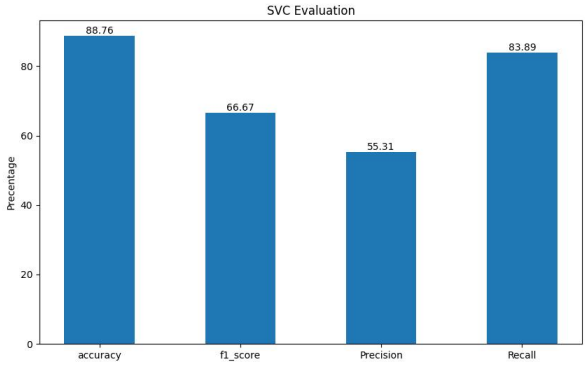

Figure 35: SVC Evaluation

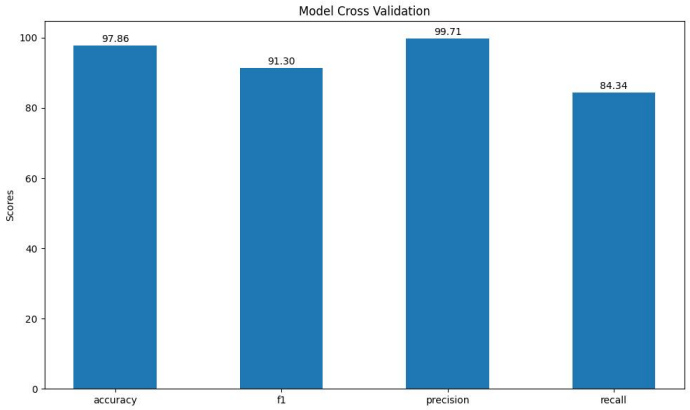

Figure 36: SVC Cross Validation

- **AdaBoost** The evaluation of AdaBoost classifier includes confusion matrix as shown below (Figure 37) , also accuracy , f1 score, precision, Recall, shown below (Figure 38) and Cross validation shown below (Figure 39)

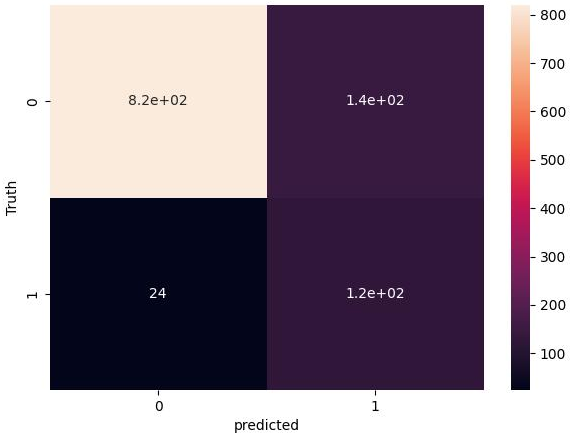

Figure 37: Ada-Boost confusion matrix

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

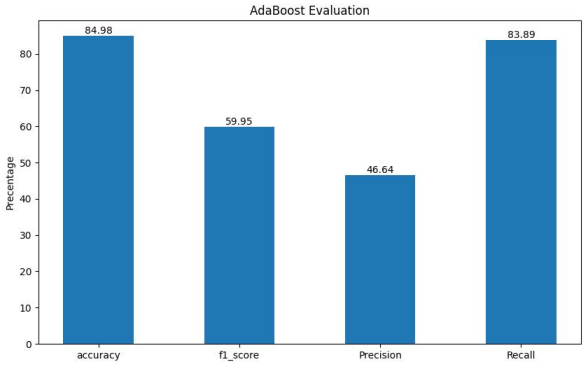

Figure 38: Ada-Boost Evaluation

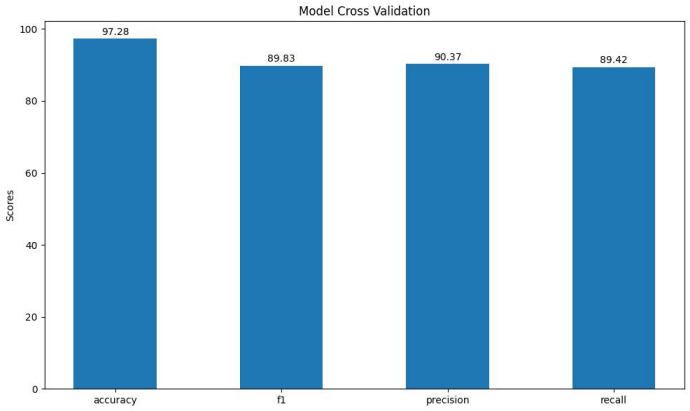

Figure 39: Ada-Boost Cross Validation

4.2 Case Study II (SMS spam collections)

In this dataset we applied five classifiers which are Multinomial Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree,Support Vector Classifier and AdaBoost.

4.2.1 Method one

In method one, we applied lemmatization and Bay of Words, as show below the evaluation.

- **Multinomial Naive Bayes** the evaluation of Naive Bayes classifier includes confusion matrix as shown below (Figure 40) , also accuracy , f1 score, precision, Recall, shown below (Figure 41) and Cross validation shown below (Figure 42)

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

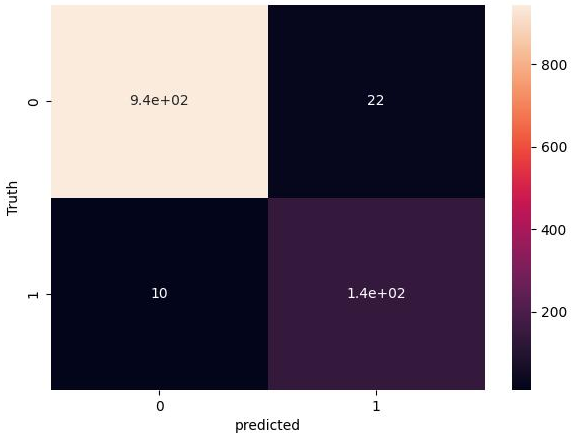

Figure 40: Naive Bayes Confusion Matrix

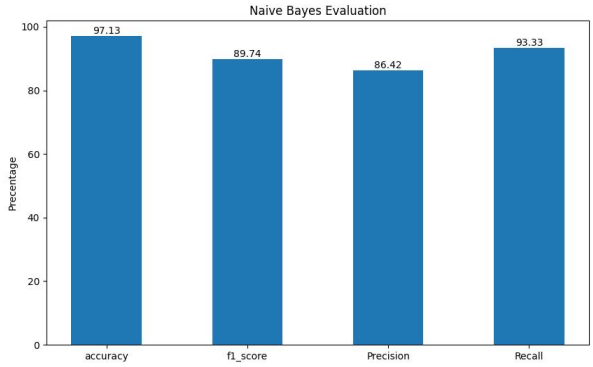

Figure 41: Naive Bayes Evaluation

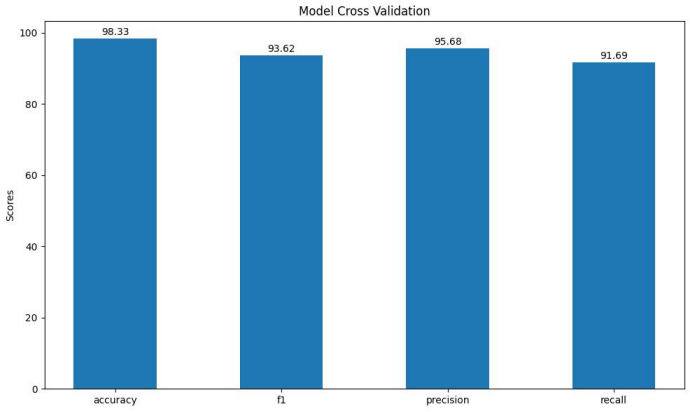

Figure 42: Naive Bayes Cross Validation

- **KNN** the evaluation of KNN classifier includes confusion matrix as shown below (Figure 43) , also accuracy , f1 score, precision, Recall, shown below (Figure 44) and Cross validation shown below (Figure 45)

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

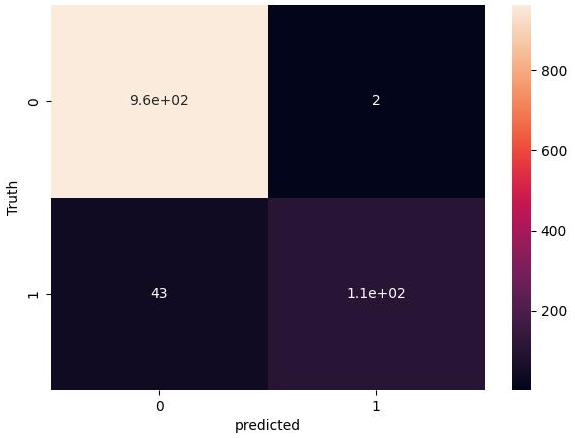

Figure 43: KNN classifier Confusion Matrix

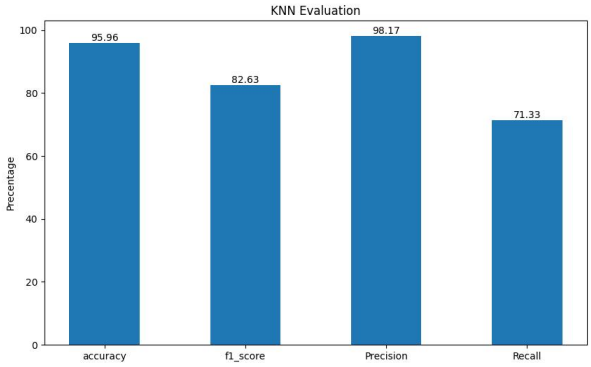

Figure 44: KNN classifier Evaluation

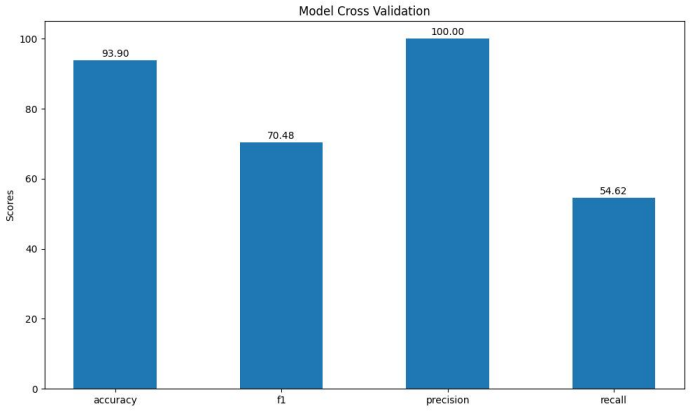

Figure 45: KNN classifier Cross Validation

- **Support Vector Classifier** The evaluation of SVC classifier includes confusion matrix as shown below (Figure 46) , also accuracy , f1 score, precision, Recall, shown below (Figure 47) Cross validation as shown below (Figure 48)

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

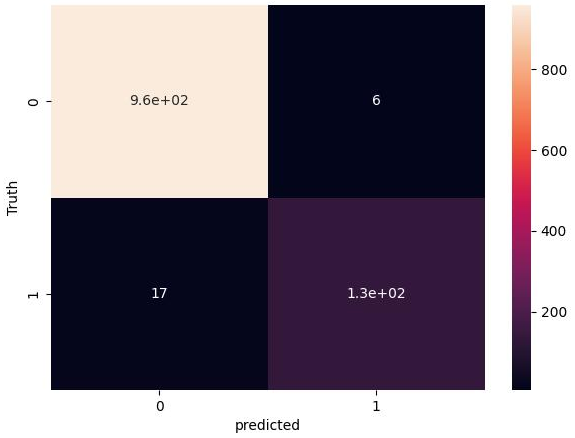

Figure 46: SVC classifier confusion matrix

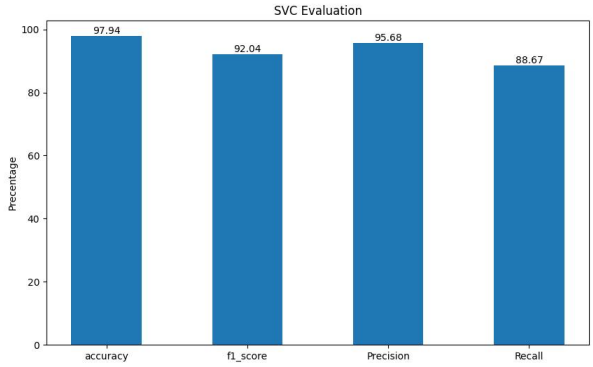

Figure 47: SVC classifier Evaluation

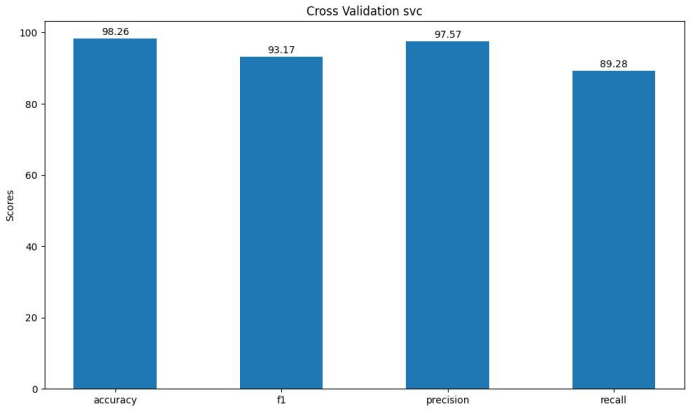

Figure 48: SVC classifier Cross Validation

- **Decision Tree Classifier** The evaluation of Decision Tree classifier includes confusion matrix (Figure 49) , also accuracy , f1 score, precision, Recall, shown below (Figure 50) Cross validation as shown (Figure 51)

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

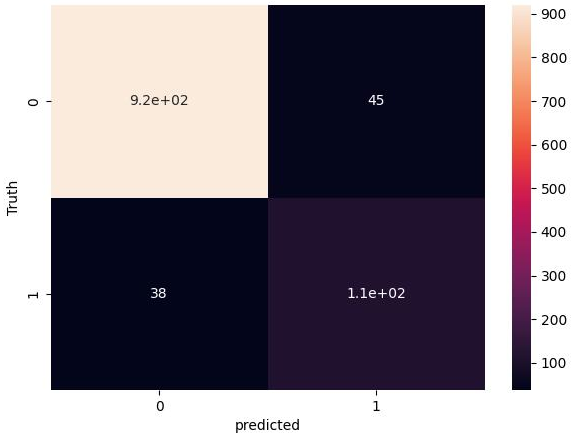

Figure 49: Decision tree classifier confusion matrix

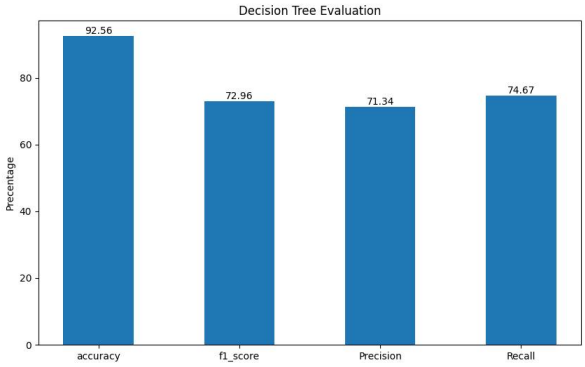

Figure 50: Decision tree classifier Evaluation

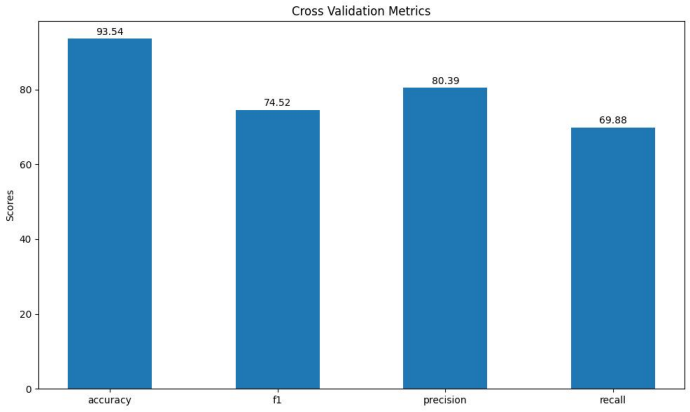

Figure 51: Decision tree Cross Validation

- **AdaBoost** The evaluation of AdaBoost classifier includes confusion matrix as shown below (Figure 52) , also accuracy , f1 score, precision, Recall, shown below (Figure 53) and Cross validation shown below (Figure 54)

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

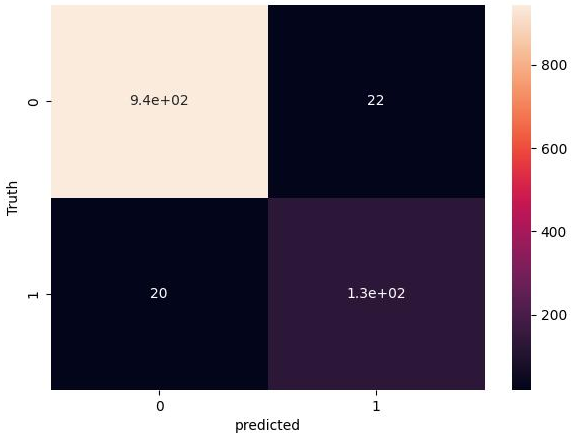

Figure 52: AdaBoost classifier confusion matrix

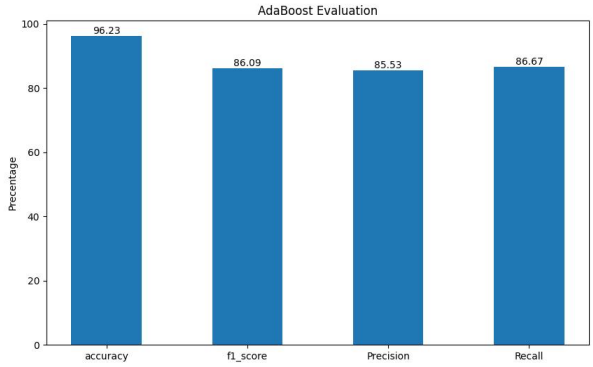

Figure 53: AdaBoost classifier Evaluation

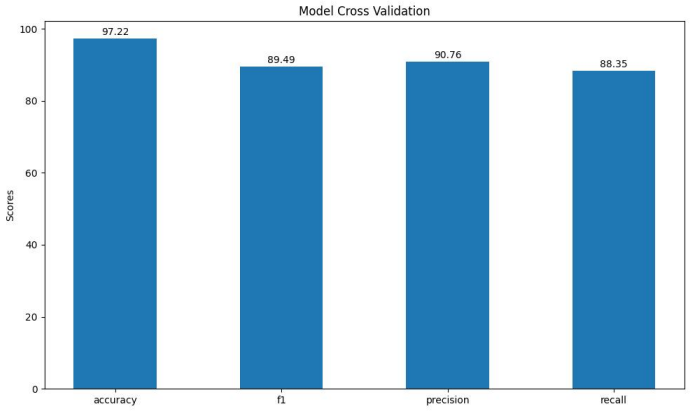

Figure 54: AdaBoost classifier Cross Validation

4.2.2 Method Two

In method one, we applied lemmatization and Bag of Worda, as show below the evaluation.

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

- **Multinomial Naive Bayes** the evaluation of Naive Bayes classifier includes confusion matrix as shown below (Figure 55) , also accuracy , f1 score, precision, Recall, shown below (Figure 56) and Cross validation shown below (Figure 57)

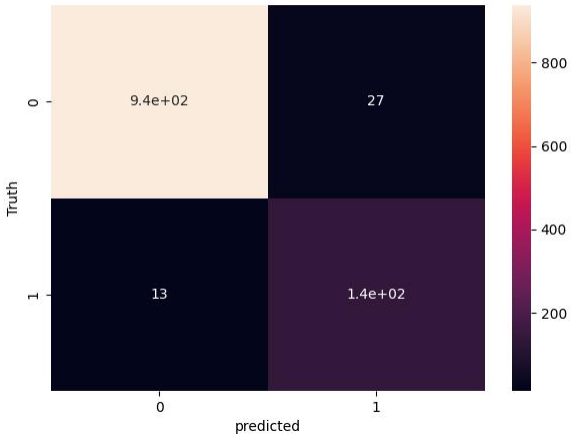

Figure 55: Naive Bayes classifier Confusion Matrix

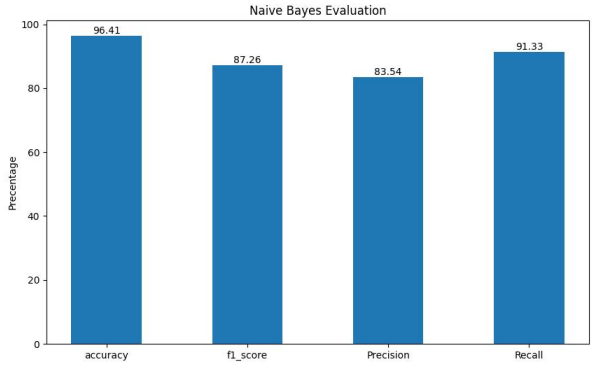

Figure 56: Naive Bayes Classifier Evaluation

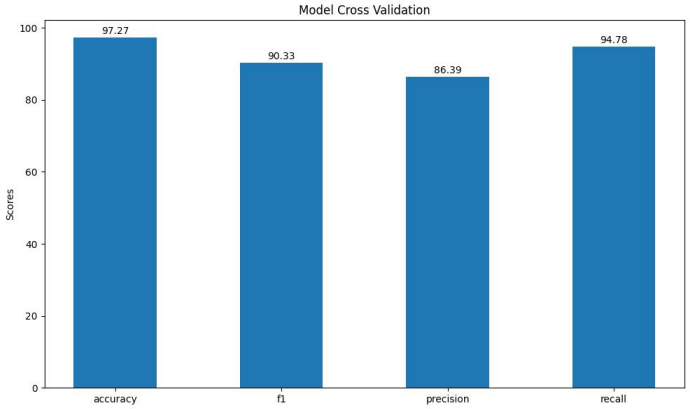

Figure 57: Naive Bayes classifier Cross Validation

- **KNN** The evaluation of KNN classifier includes confusion matrix as shown below (Figure 58) , also

accuracy , f1 score, precision, Recall, shown below (Figure 59) and Cross validation shown below (Figure 60)

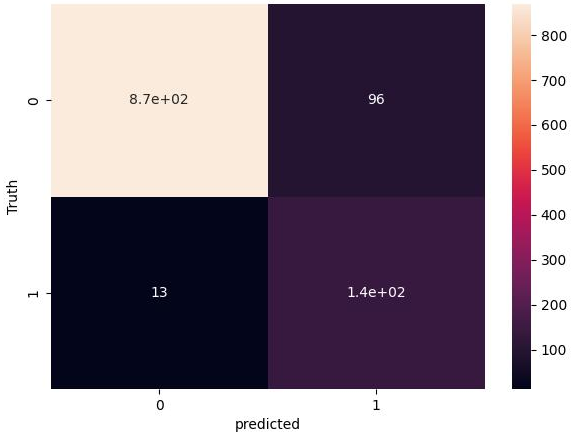

Figure 58: KNN Classifier Confusion Matrix

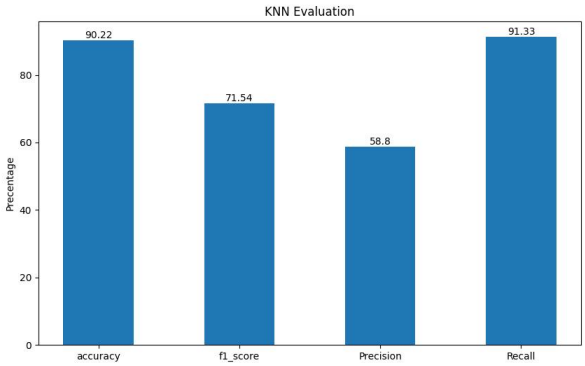

Figure 59: KNN Classifier Evaluation

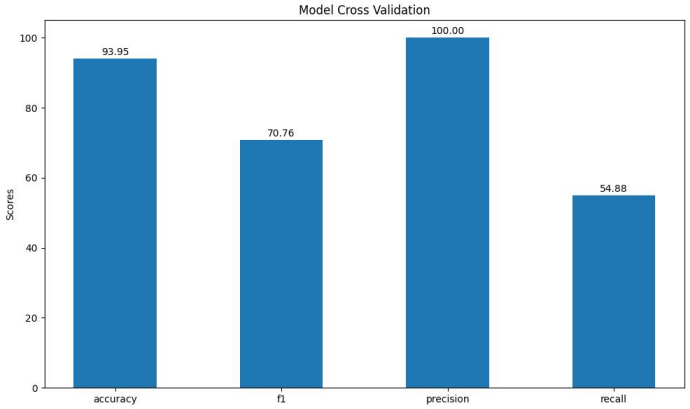

Figure 60: KNN Classifier Cross Validation

- **Support Vector Classifier** The evaluation of SVC classifier includes confusion matrix as shown below (Figure 61) , also accuracy , f1 score, precision, Recall, shown below (Figure 62) Cross validation as shown below (Figure 63)

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

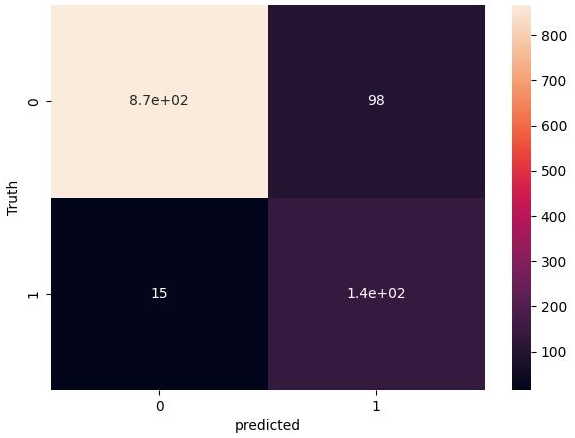

Figure 61: SVC classifier confusion matrix

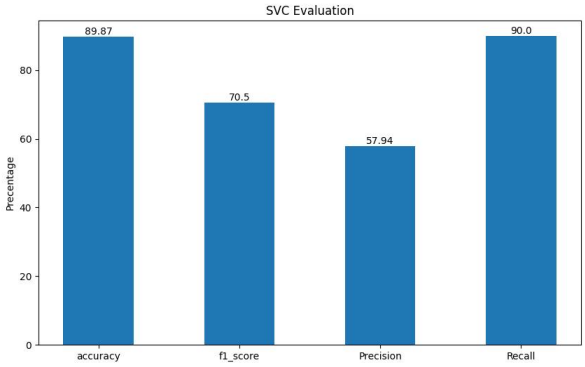

Figure 62: SVC classifier Evaluation

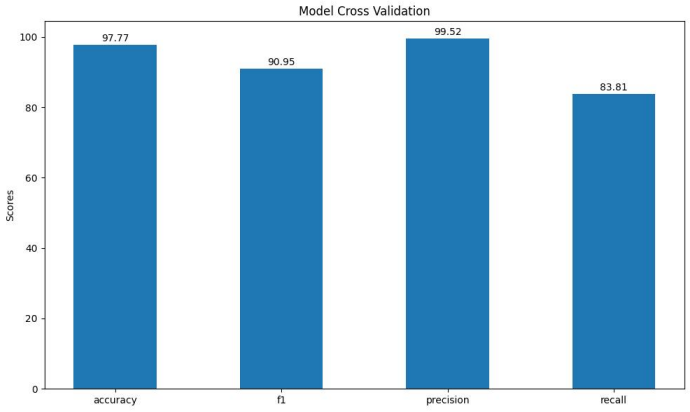

Figure 63: SVC classifier Cross Validation

- **Decision Tree Classifier** The evaluation of Decision Tree classifier includes confusion matrix (Figure 64) , also accuracy , f1 score, precision, Recall, shown below (Figure 65) Cross validation as shown (Figure 66)

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

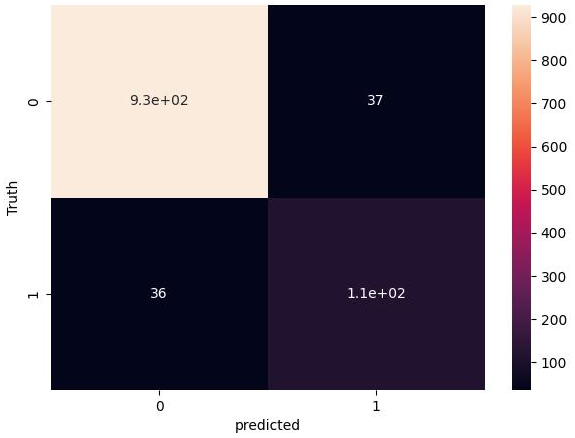

Figure 64: decision tree classifier confusion matrix

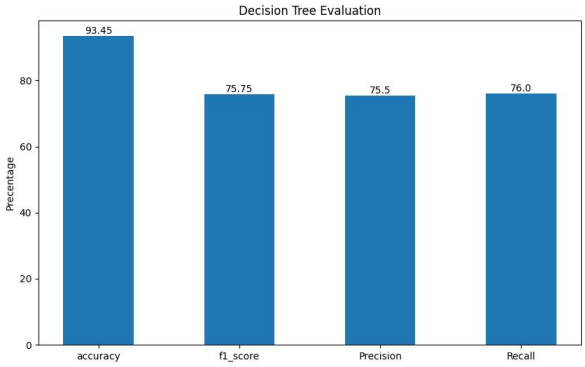

Figure 65: decision tree classifier Evaluation

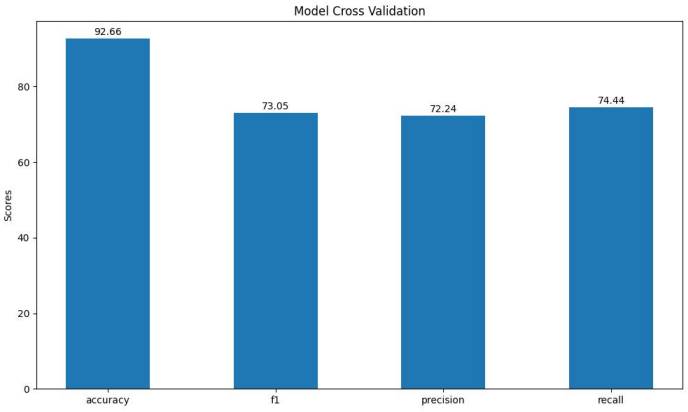

Figure 66: decision tree Cross Validation

- **AdaBoost** The evaluation of AdaBoost classifier includes confusion matrix as shown below (Figure 67) , also accuracy , f1 score, precision, Recall, shown below (Figure 68) and Cross validation shown below (Figure 69)

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

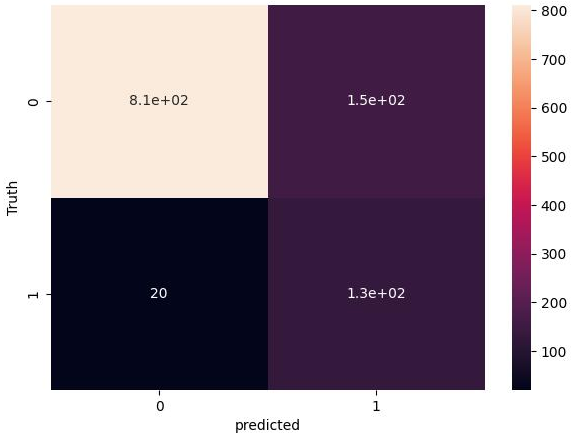

Figure 67: adaBoost classifier confusion matrix

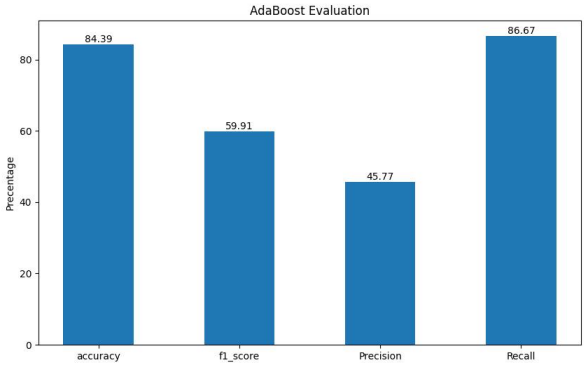

Figure 68: adaBoost classifier Evaluation

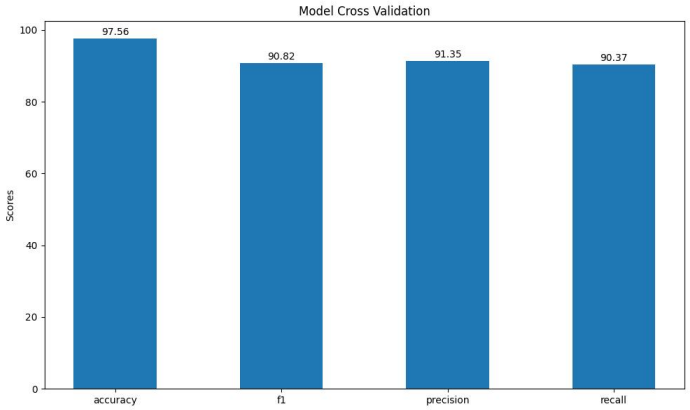

Figure 69: adaBoost classifier Cross Validation

DOI: https://doi.org/10.54216/FPA.180112
Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

5 Conclusion and future work

5.1 Case Study I (Filtering mobile phone spam Dataset)

In these dataset, after cleaning and checking nulls. we start to apply two different method for pre-processing first method, includes TfidVectorizer and Porter stemmer, then passing features through five Machine learning classifiers (Naive Bayes, KNN, Decision Tree, SVC, AdaBoost) the results shown as below (Figure 70)

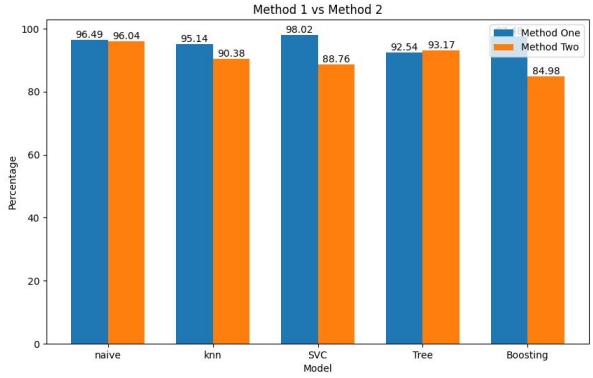

Figure 70: Method 1 vs Method 2

To conclude the results, Method one improve accuracy comparing to Method Two, Support Vector Classifier has the highest accuracy percentage comparing to other classifiers.

The K-Folds method for Cross Validation that it ensures that all the original dataset appear in both the training and test sets. these method is usually very useful when dealing with limited input data. In this dataset, we apply K-Folds method and The process begin by randomly dividing the dataset into 10 folds. Results as shown below (Figure 71)

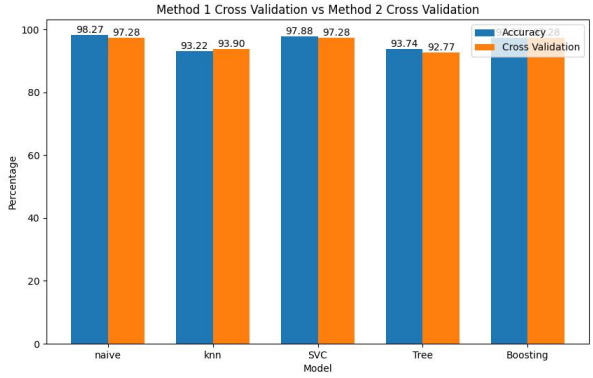

Figure 71: Method 1 Cross validation VS Method 2 Cross validation

To conclude the results, Method one has little bit improving in accuracy comparing to Method Two. Support Vector Classifier, Naive Bayes and Ada-Boosting have the highest accuracy percentage comparing to other classifiers.

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

5.2 Case Study II (SMS Spam Collection (spam or legitimate) Dataset)

In these dataset, after cleaning and checking nulls. we start to apply two different method for pre-processing first method, includes lemmetization and bag of words, then passing features through five Machine learning classifiers (Naive Bayes, KNN, Decision Tree, SVC, AdaBoost) the results shown as below (Figure 72)

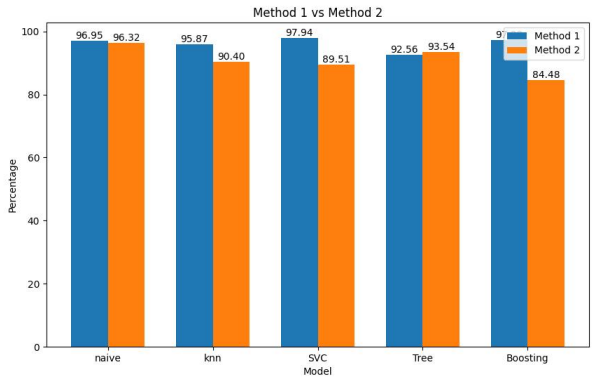

Figure 72: Method 1 vs Method 2

To conclude the results, Method one improve accuracy comparing to Method Two, Support Vector Classifier, KNN classifier, Ada-Boost classifier have the highest accuracy percentage comparing to other classifiers.

The K-Folds method for Cross Validation that it ensures that all the original dataset appear in both the training and test sets. these method is usually very useful when dealing with limited input data. In this dataset, we apply K-Folds method and The process begin by randomly dividing the dataset into 10 folds. Results as shown below (Figure 73)

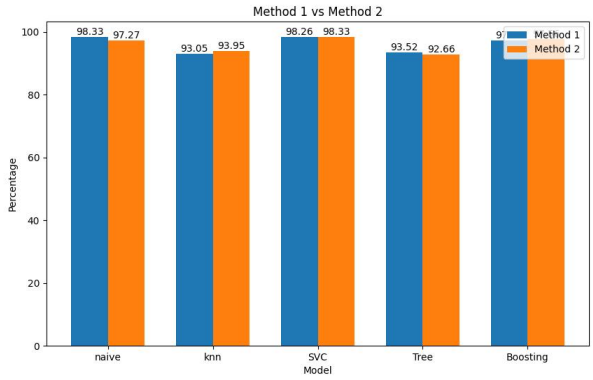

Figure 73: Method 1 Cross validation VS Method 2 Cross validation

To conclude the results, Method one has little bit improving in accuracy comparing to Method Two. Support Vector Classifier, Naive Bayes and Ada-Boosting have the highest accuracy percentage comparing to other classifiers.

The Accuracy of all classifiers models are so good so we don't need to integrate two datasets to be improve accuracy.

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

Author contributions

All authors contributed equally to this paper. All authors approved the work in this paper.

References

[1] N. Kumar, S. Sonowal, "Email spam detection using machine learning algorithms", Proc. of 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 108–113, 2020.

[2] P. T. Nallamothu, M. S. Khan, "Machine learning for SPAM detection", Asian Journal of Advances in Research, vol. 6, no. 1, pp. 167–179, 2023.

[3] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, T. Shah, "Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges", Security and Communication Networks, vol. 2022, pp. 1–19, 2022.

[4] Y. Kontsewaya, E. Antonov, A. Artamonov, "Evaluating the effectiveness of machine learning methods for spam detection", Procedia Computer Science, vol. 190, pp. 479–486, 2021.

[5] A. P. Rodrigues, R. Fernandes, A. Shetty, K. Lakshmanna, R. M. Shafi, "Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques", Computational Intelligence and Neuroscience, vol. 2022, 2022.

[6] N. Sun, G. Lin, J. Qiu, P. Rimba, "Near real-time twitter spam detection with machine learning techniques", International Journal of Computers and Applications, vol. 44, no. 4, pp. 338–348, 2022.

[7] S. Nandhini, J. Marseline, "Performance evaluation of machine learning algorithms for email spam detection", Proc. of 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), pp. 1–4, 2020.

[8] S. D. Gupta, S. Saha, S. K. Das, "SMS spam detection using machine learning", Journal of Physics: Conf. Ser., vol. 1797, no. 1, 2021.

[9] M. F. A. Kadir, A. F. A. Abidin, M. A. Mohamed, N. A. Hamid, "Spam detection by using machine learning based binary classifier", Indonesian Journal of Electrical Engineering and Computer Science, vol. 26, no. 1, pp. 310–317, 2022.

[10] M. R. Julis, S. Alagesan, "Spam detection in SMS using machine learning through text mining", International Journal of Scientific Technology Research, vol. 9, no. 02, 2020.

[11] L. GuangJun, S. Nazir, H. U. Khan, A. U. Haq, "Spam detection approach for secure mobile message communication using machine learning algorithms", Security and Communication Networks, vol. 2020, pp. 1–6, 2020.

[12] H. Sajedi, G. Z. Parast, F. Akbari, "SMS spam filtering using machine learning techniques: A survey", Machine Learning Research, vol. 1, no. 1, pp. 1–14, 2016.

[13] T. Almeida, J. M. Hidalgo, T. Silva, "Towards SMS spam filtering: Results under a new dataset", International Journal of Information Security Science, vol. 2, no. 1, pp. 1–18, 2013.

[14] L. Jiang, S. Wang, C. Li, L. Zhang, "Structure extended multinomial naive Bayes", Information Sciences, vol. 329, pp. 346–356, 2016.

[15] O. Bardhi, B. G. Zapirain, "Machine learning techniques applied to electronic healthcare records to predict cancer patient survivability", Computers, Materials & Continua, vol. 68, no. 2, pp. 1595–1613, 2021.

[16] S. Suthaharan, "Support vector machine", in Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning, pp. 207–235, 2016, Springer.

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024

[17] B. De Ville, ”Decision trees”, Wiley Interdisciplinary Reviews: Computational Statistics, vol. 5, no. 6, pp. 448–455, 2013.

DOI: https://doi.org/10.54216/FPA.180112

Received: July 04, 2024 Revised: September 28, 2024 Accepted: December 26, 2024