

A PROJECT REPORT  
on  
**" SPAM EMAIL CLASSIFIER "**

Submitted to  
KIIT Deemed to be University

BACHELOR'S DEGREE IN  
COMPUTER SCIENCE &  
ENGINEERING  
BY

**Abhay Nath Sharma**

ROLL NUMBER **2105337**

**Sayan Mondal**

ROLL NUMBER **21051512**

**Abhinav Bisht**

ROLL NUMBER **21051704**

**Manish Kumar**

ROLL NUMBER **21051659**

**Harsh Gupta**

ROLL NUMBER **21051651**

UNDER THE GUIDANCE OF  
**Dr. Soumya Ranjan Mishra**



SCHOOL OF COMPUTER ENGINEERING  
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY  
BHUBANESWAR, ODISHA -751024

# KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024



## CERTIFICATE

This is certify that the project entitled

**" SPAM EMAIL CLASSIFIER "**

submitted by -

**Abhay Nath Sharma**

**ROLL NUMBER 2105337**

**Sayan Mondal**

**ROLL NUMBER 21051512**

**Abhinav Bisht**

**ROLL NUMBER 21051704**

**Manish Kumar**

**ROLL NUMBER 21051659**

**Harsh Gupta**

**ROLL NUMBER 21051651**

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering ) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024 under our guidance.

Date: 28/03/2024

**Dr. Soumya Ranjan Mishra**  
**(Project Guide )**

## **Acknowledgement**

We are profoundly grateful to **Dr. Soumya Ranjan Mishra** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

# Abstract

Nowadays communication plays a major role in everything be it professional or personal. Email communication service is being used extensively because of its free use services, low-cost operations, accessibility, and popularity. Emails have one major security flaw that is anyone can send an email to anyone just by getting their unique user id. This security flaw is being exploited by some businesses and ill-motivated persons for advertising, phishing, malicious purposes, and finally fraud. This produces a kind of email category called SPAM.

Spam refers to any email that contains an advertisement, unrelated and frequent emails. These emails are increasing day by day in numbers. Studies show that around 55 percent of all emails are some kind of spam. A lot of effort is being put into this by service providers. Spam is evolving by changing the obvious markers of detection. Moreover, the spam detection of service providers can never be aggressive with classification because it may cause potential information loss to incase of a misclassification.

To tackle this problem we present a new and efficient method to detect spam using machine learning and natural language processing. A tool that can detect and classify spam. In addition to that, it also provides information regarding the text provided in a quick view format for user convenience.

## Table of Contents -

CHAPTER NO	TITLE	PAGE NO
	Abstract	1
	Table of Contents	2
1	<b>Introduction</b>	3
2	<b>Literature Review</b>	4
	2.1 introduction	
	2.2 Related work	
	2.3 Summary	
3	<b>Objectives and Scope</b>	5
4	<b>Experimentation and Methods</b>	6
	4.1 Introduction	6
	4.2 Requirements	7
	4.3 Workflow	8
	4.3.1 Data Processing	9
	4.4 Algorithms	10
5	<b>Results and Discussion</b>	17
6	<b>Conclusion and Future Scope</b>	20
	<b>References</b>	21

# 1. Introduction

Today, Spam has become a major problem in communication over internet. It has been accounted that around 55% of all emails are reported as spam and the number has been growing steadily. Spam which is also known as unsolicited bulk email has led to the increasing use of email as email provides the perfect ways to send the unwanted advertisement or junk newsgroup posting at no cost for the sender. This chances has been extensively exploited by irresponsible organizations and resulting to clutter the mail boxes of millions of people all around the world.

Spam has been a major concern given the offensive content of messages, spam is a waste of time. End user is at risk of deleting legitimate mail by mistake. Moreover, spam also impacted the economical which led some countries to adopt legislation

Text classification is used to determine the path of incoming mail/message either into inbox or straight to spam folder. It is the process of assigning categories to text according to its content. It is used to organized, structures and categorize text. It can be done either manually or automatically. Machine learning automatically classifies the text in a much faster way than manual technique. Machine learning uses pre-labelled text to learn the different associations between pieces of text and it output. It used feature extraction to transform each text to numerical representation in form of vector which represents the frequency of word in predefined dictionary.

Text classification is important to structure the unstructured and messy nature of text such as documents and spam messages in a cost-effective way. Machine learning can make more accurate precisions in real-time and help to improve the manual slow process to much better and faster analysing big data. It is important especially to a company to analyse text data, help inform business decisions and even automate business processes.

In this project, machine learning techniques are used to detect the spam message of a mail. Machine learning is where computers can learn to do something 10 without the need to explicitly program them for the task.

It uses data and produce a program to perform a task such as classification. Compared to knowledge engineering, machine learning techniques require messages that have been successfully pre-classified. The pre-classified messages make the training dataset which will be used to fit the learning algorithm to the model in machine learning studio.

A combination of algorithms are used to learn the classification rules from messages. These algorithms are used for classification of objects of different classes. These algorithms are provided with pre labelled data and an unknown text. After learning from the prelabelled data each of these algorithms predict which class the unknown text may belong to and the category predicted by majority is considered as final.

## 2. Literature Review

### 2.1 Introduction

This chapter discusses about the literature review for machine learning classifier that being used in previous researches and projects. It is not about information gathering but it summarize the prior research that related to this project. It involves the process of searching, reading, analysing, summarising and evaluating the reading materials based on the project. A lot of research has been done on spam detection using machine learning. But due to the evolvement of spam and development of various technologies the proposed methods are not dependable. Natural language processing is one of the lesser known fields in machine learning and it reflects here with comparatively less work present.

### 2.2 Related work

Spam classification is a problem that is neither new nor simple. A lot of research has been done and several effective methods have been proposed. i. M. RAZA, N. D. Jayasinghe, and M. M. A. Muslam have analyzed various techniques for spam classification and concluded that naïve Bayes and support vector machines have higher accuracy than the rest, around 91% consistently [1].

ii. S. Gadde, A. Lakshmanarao, and S. Satyanarayana in their paper on spam detection concluded that the LSTM system resulted in higher accuracy of 98%[2].

iii. P. Sethi, V. Bhandari, and B. Kohli concluded that machine learning algorithms perform differently depending on the presence of different attributes [3].

iv. H. Karamollaoglu, İ. A. Dogru, and M. Dorterler performed spam classification on Turkish messages and emails using both naïve Bayes classification algorithms and support vector machines and concluded that the accuracies of both models measured around 90% [4].

### 2.3 Summary

From various studies, we can take that for various types of data various models performs better. Naïve Bayes, random forest, SVM, logistic regression are some of the most used algorithms in spam detection and classification.

# 3. Objectives and Scope

## 3.1 Problem Statement

Spammers are in continuous war with Email service providers. Email service providers implement various spam filtering methods to retain their users, and spammers are continuously changing patterns, using various embedding tricks to get through filtering. These filters can never be too aggressive because a slight misclassification may lead to important information loss for consumer. A rigid filtering method with additional reinforcements is needed to tackle this problem.

## 3.2 Objectives

The objectives of this project are

- i. To create a ensemble algorithm for classification of spam with highest possible accuracy.
- ii. To study on how to use machine learning for spam detection.
- iii. To study how natural language processing techniques can be implemented in spam detection.
- iv. To provide user with insights of the given text leveraging the created algorithm and NLP.

## 3.3 Project Scope

This project needs a coordinated scope of work.

- i. Combine existing machine learning algorithms to form a better ensemble algorithm.
- ii. Clean, processing and make use of the dataset for training and testing the model created.
- iii. Analyse the texts and extract entities for presentation.

## 3.4 Limitations

This Project has certain limitations. i. This can only predict and classify spam but not block it. ii. Analysis can be tricky for some alphanumeric messages and it may struggle with entity detection. iii. Since the data is reasonably large it may take a few seconds to classify and analyse the message.



## 4. Experimentation and Methods

### 4.1 Introduction

This chapter will explain the specific details on the methodology being used to develop this project. Methodology is an important role as a guide for this project to make sure it is in the right path and working as well as plan. There is different type of methodology used in order to do spam detection and filtering. So, it is important to choose the right and suitable methodology thus it is necessary to understand the application functionality itself.

### 4.2 System Architecture

The application overview has been presented below and it gives a basic structure of the application.

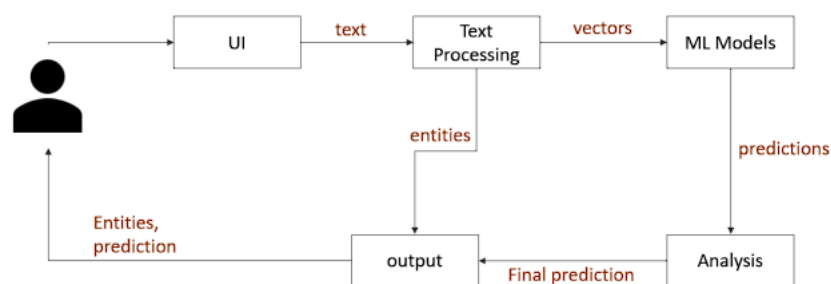


fig no. 4.1 Architecture

The UI, Text processing and ML Models are the three important modules of this project. Each Module's explanation has been given in the later sections of this chapter.

A more complicated and detailed view of architecture is presented in the workflow section.

### 4.3 Modules and Explanation

The Application consists of three modules.

- i. UI
- ii. Machine Learning
- iii. Data Processing

## I. UI Module

- a. This Module contains all the functions related to UI(user interface).
- b. The user interface of this application is designed using Streamlit library from python based packages.
- c. The user inputs are acquired using the functions of this library and forwarded to data processing module for processing and conversion.
- d. Finally the output from ML module is sent to this module and from this module to user in visual form.

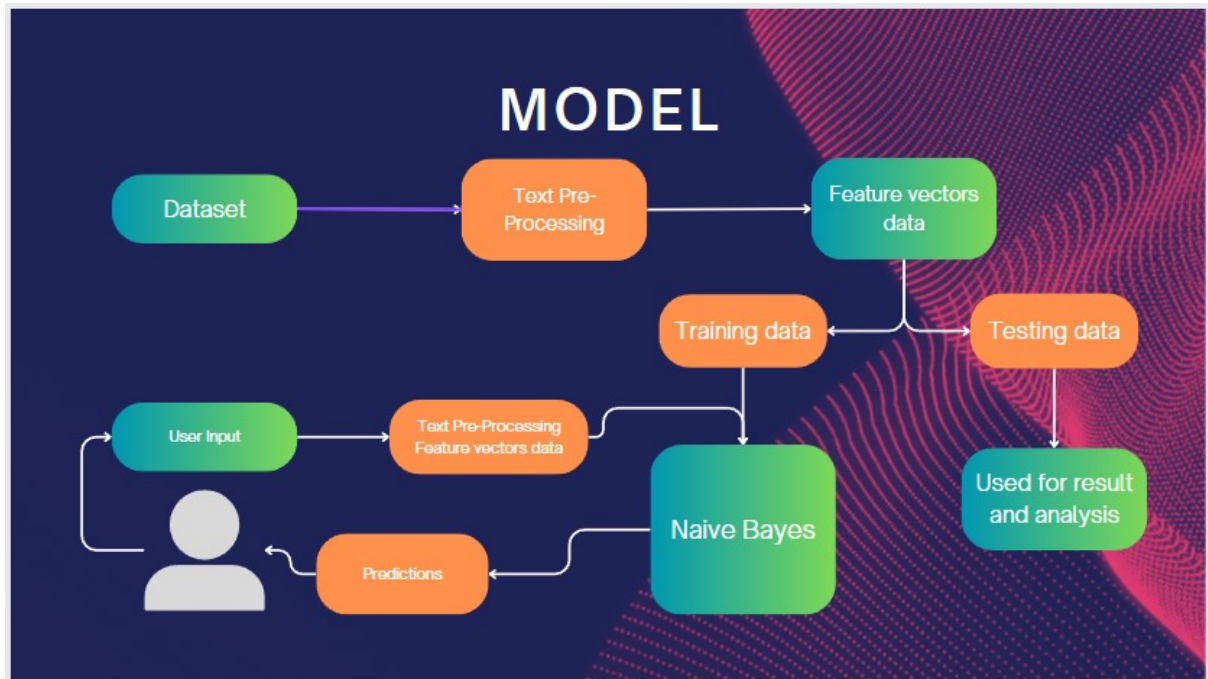
## II. Machine Learning Module

- a. This module is the main module of all three modules.
- b. This modules performs everything related to machine learning and results analysis.
- c. Some main functions of this module are
  - i. Training machine learning models.
  - ii. Testing the model
  - iii. Determining the respective parameter values for each model.
  - iv. Key-word extraction.
  - v. Final output calculation
- d. The output from this module is forwarded to UI for providing visual response to user

## III. Data Processing Module

- a. The raw data undergoes several modifications in this module for further process.
- b. Some of the main functions of this module includes
  - i. Data cleaning
  - ii. Data merging of datasets
  - iii. Text Processing using NLP
  - iv. Conversion of text data into numerical data(feature vectors).
  - v. Splitting of data.
- c. All the data processing is done using Pandas and NumPy libraries. d. Text processing and text conversion is done using NLTK and scikit-learn libraries.

## 4.4 WorkFlow



In the above architecture, the objects depicted in orange belong to a module called Data Processing. It includes several functions related to data processing, natural Language Processing. The objects depicted in Blue belong to the Machine Learning module. It is where everything related to ML is embedded. The red objects represent final results and outputs

#### 4.5.1 Data Collection and Description

- Data plays an important role when it comes to prediction and classification, the more the data the more the accuracy will be.
- The data used in this project is completely open-source and has been taken from various resources like Kaggle and UCI
- For the purpose of accuracy and diversity in data multiple datasets are taken. 2 datasets containing approximately over 5000 mails and their labels are used for training and testing the application.
- 1000 spam mails are taken for generalisation of data and to increase the accuracy.

#### 4.5.2 Data Processing

It consists of two main tasks

- Dataset cleaning

It includes tasks such as removal of outliers, null value removal, removal of unwanted features from data.

- Dataset Merging

After data cleaning, the datasets are merged to form a single dataset containing only two features(text, label). Data cleaning, Data Merging these procedures are completely done using Pandas library. Textual data processing

- Tag

removal Removing all kinds of tags and unknown characters from text using regular expressions through Regex library.

- Sentencing, tokenization

Breaking down the text(email/SMS) into sentences and then into tokens(words). This process is done using NLTK pre-processing library of python.

- Stop word removal

Stop words such as of , a ,be , ... are removed using stopwords NLTK library of python.

- Lemmatization

Words are converted into their base forms using lemmatization and pos-tagging This process gives key-words through entity extraction. This process is done using chunking in regex and NLTK lemmatization.

- Sentence formation

The lemmatized tokens are combined to form a sentence. This sentence is essentially a sentence converted into its base form and removing stop words. Then all the sentences are combined to form a text.

- While the overall data processing is done only to datasets, the textual processing is done to both training data, testing data and also user input data.

#### 4.5.2.1 Feature Vector Formation

- The texts are converted into feature vectors(numerical data) using the words present in all the texts combined
- This process is done using countvectorization of NLTK library.
- The feature vectors can be formed using two language models Bag of Words and Term Frequency-inverse Document Frequency.

##### 4.5.2.1.2 Term Frequency-inverse document frequency

Term frequency-inverse document frequency of a word is a measurement of the importance of a word. It compares the repentance of words to the collection of documents and calculates the score.

Terminology for the below formulae:

t – term(word)

d – document(set of words)

N – count of documents The TF-IDF process consists of various activities listed below.

- Term Frequency The count of appearance of a particular word in a document is called term frequency  **$tf(t, d) = \text{count of } t \text{ in } d / \text{number of words in } d$**
- Document Frequency Document frequency is the count of documents the word was detected in. We consider one instance of a word and it doesn't matter if the word is present multiple times.

**$df(t) = \text{occurrence of } t \text{ in documents}$**  iii) Inverse Document Frequency

- IDF is the inverse of document frequency.
- It measures the importance of a term t considering the information it contributes. Every term is considered equally important but certain terms such as (are, if, a, be, that, ..) provide little information about the document. The inverse document frequency factor reduces the importance of words/terms that has high recurrence and increases the importance of words/terms that are rare.

$$idf(t) = N/df$$

Finally, the TF-IDF can be calculated by combining the term frequency and inverse document frequency.

$$tf\_idf(t, d) = tf(t, d) * \log(N/(df + 1))$$

The process can be explained using the following example:

“Document 1 It is going to rain today.

Document 2 Today I am not going outside.

Document 3 I am going to watch the season premiere.”

The Bag of words of the above sentences is

[going:3, to:2, today:2, i:2, am:2, it:1, is:1, rain:1]

Then finding the term frequency:-

table no. 4.1 Term frequency

Words	IDF Value
Going	$\log(3/3)$
To	$\log(3/2)$
Today	$\log(3/2)$
I	$\log(3/2)$
Am	$\log(3/2)$
It	$\log(3/1)$
Is	$\log(3/1)$
rain	$\log(3/1)$

Then finding the inverse document frequency.

table no. 4.2 inverse document frequency

Words	Document1	Document2	Document3
Going	0.16	0.16	0.12
To	0.16	0	0.12
Today	0.16	0.16	0
I	0	0.16	0.12
Am	0	0.16	0.12
It	0.16	0	0
Is	0.16	0	0
rain	0.16	0	0

Applying the final equation the values of tf-idf becomes

Words/ documents	going	to	Today	i	am	if	it	rain
Document1	0	0.07	0.07	0	0	0.17	0.17	0.17
Document2	0	0	0.07	0.07	0.07	0	0	0
Document3	0	0.05	0	0.05	0.05	0	0	0

table no. 4.3 TF-IDF

Using the above two language models the complete data has been converted into two kinds of vectors and stored into a csv type file for easy access and minimal processing.

### 4.5.3 Data Splitting

The data splitting is done to create two kinds of data Training data and testing data. Training data is used to train the machine learning models and testing data is used to test the models and analyse results. 80% of total data is selected as testing data and remaining data is testing data.

### 4.5.4 Machine Learning

#### 4.5.4.1 Introduction

Machine Learning is process in which the computer performs certain tasks without giving instructions. In this case the models takes the training data and train on them. Then depending on the trained data any new unknown data will be processed based on the ruled derived from the trained data.

After completing the countvectorization and TF-IDF stages in the workflow the data is converted into vector form(numerical form) which is used for training and testing models.

The models used for the study include Naïve Bayes and a proposed model which was created using an ensemble approach

#### 4.5.4.2.1 Naïve Bayes Classifier

A naïve Bayes classifier is a supervised probabilistic machine learning model that is used for classification tasks. The main principle behind this model is the Bayes theorem.

Bayes Theorem: Naive Bayes is a classification technique that is based on Bayes' Theorem with an assumption that all the features that predict the target value are independent of each other. It calculates the probability of each class and then picks the one with the highest probability.

Naive Bayes classifier assumes that the features we use to predict the target are independent and do not affect each other. Though the independence assumption is never correct in real-world data, but often works well in practice. so that it is called "Naive" [14].

$$P(A|B) = (P(B|A)P(A))/P(B)$$

$P(A|B)$  is the probability of hypothesis A given the data B. This is called the posterior probability.

$P(B|A)$  is the probability of data B given that hypothesis A was true.

$P(A)$  is the probability of hypothesis A being true (regardless of the data). This is called the prior probability of A.

$P(B)$  is the probability of the data (regardless of the hypothesis) [15].

Naïve Bayes classifiers are mostly used for text classification. The limitation of the Naïve Bayes model is that it treats every word in a text as independent and is equal in importance but every word cannot be treated equally important because articles and nouns are not the same when it comes to language. But due to its classification efficiency, this model is used in combination with other language processing techniques.



#### 4.5.5 Experimentation

The process goes like data collection and processing then natural language processing and then vectorization then machine learning. The data is collected, cleaned, and then subjected to natural language processing techniques specified in section IV. Then the cleaned data is converted into vectors using Bag of Words and TF-IDF methods which goes like...

The Data is split into Training data and Testing Data in an 80-20 split ratio. The training and testing data is converted into Bag-of-Words vectors and TF-IDF vectors.

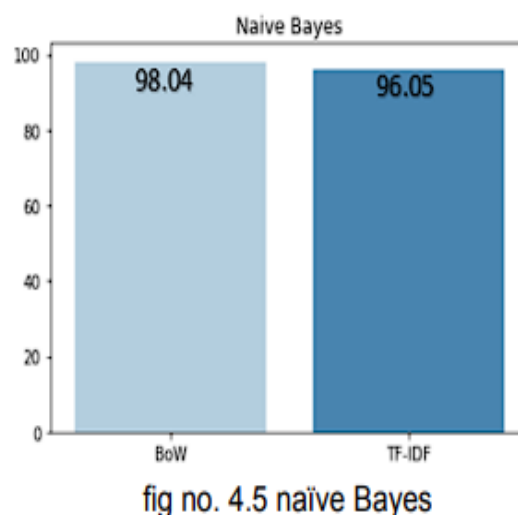
There are several metrics to evaluate the models but accuracy is considered for comparing BoW and TF-IDF models. Accuracy is generally used to determine the efficiency of a model.

##### **Accuracy:**

“Accuracy is the number of correctly predicted data points out of all the data points”.

##### **Naïve Bayes Classification algorithm:**

Two models, one for Bow and one for TF-IDF are created and trained using respective training vectors and training labels. Then the respective testing vectors and labels are used to get the score for the model.



The scores for Bag-of-Words and TF-IDF are visualized.

The scores for the Bow model and TF-IDF models are 98.04 and 96.05 respectively for using the naïve bayes model.

#### **4.5.6 User Interface(UI)**

Interface (UI) is an important component in this application. The user only interacts with the interface.

The UI of this project has been constructed with the help of an open source library called streamlit. The complete information and API reference sheet can be obtained from [here](#).

#### **4.5.7 Working Procedure**

The working procedure includes the internal working and the data flow of application.

i. After running the application some procedures are automated.

1. Reading data from file
2. Cleaning the texts
3. Processing
4. Splitting the data
5. Intialising and training the models

ii. The user just needs to provide some data to classify in the area provided.

iii. The provided data undergoes several procedures after submission.

1. Textual Processing
2. Feature Vector conversion
3. Entity extraction

iv. The created vectors are provided to trained models to get predictions.

v. After getting predictions the category predicted by majority will be selected.

vi. The accuracies of that prediction will be calculated

vii. The accuracies and entities extracted from the step 3 will be provided to user. Every time the user gives something new the procedure from step 2 will be repeated.

# 5. Result and Discussion

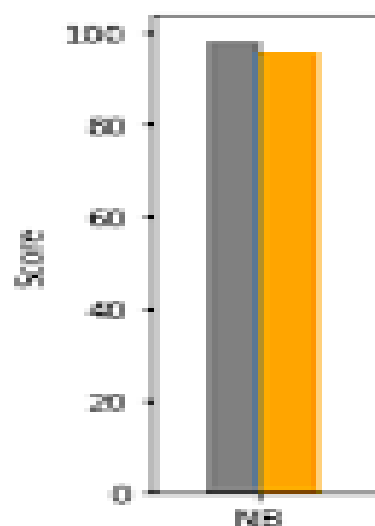
## 5.1 Language Model Selection

While selecting the best language model the data has been converted into both types of vectors and then the models been tested for to determine the best model for classifying spam. The results from individual models are presented in the experimentation section under methodology. Now comparing the results from the models.

Based on our experimentation, we conclude that TF-IDF is a more effective feature extraction technique for text classification compared to Bag of Words. TF-IDF's ability to capture word importance and handle stopwords better contributed to its superior performance in terms of accuracy.

TF-IDF outperformed BoW in terms of accuracy. This could be attributed to TF-IDF's ability to capture the importance of words in distinguishing between different documents. By considering both the term frequency and inverse document frequency, TF-IDF assigns higher weights to informative words and downweights common stopwords, leading to better discrimination between spam and non-spam messages.

score comparison



From the figure it is clear that TF-IDF proves to be better than BoW in every model tested. Hence TF-IDF has been selected as the primary language model for textual data conversion in feature vector formation.

## 5.2 Proposed Model results

To determine which model is effective we used three metrics Accuracy, Precision, and F1score.

The resulted values for the proposed model are

**Accuracy – 99.0**

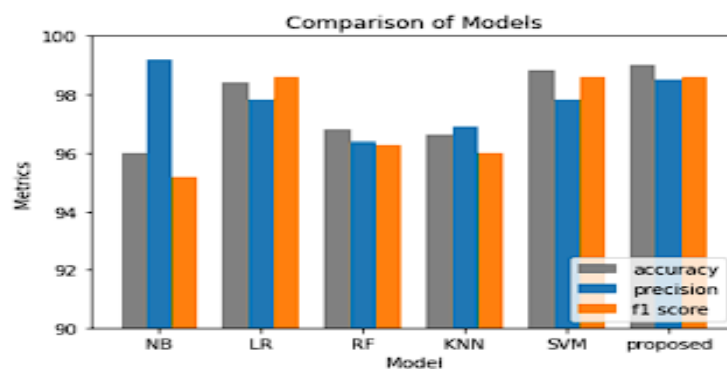
**Precision – 98.5**

**F1 Score – 98.6**

## 5.3 Comparison

The results from the proposed model has been compared with all the models individually in tabular form to illustrate the differences clearly.

Metric Model	Accuracy	Precision	F1 Score
Naïve Bayes	96.0	99.2	95.2
Logistic Regression	98.4	97.8	98.6
Random forest	96.8	96.4	96.3
KNN	96.6	96.9	96.0
SVM	98.8	97.8	98.6
Proposed model	99.0	98.5	98.6

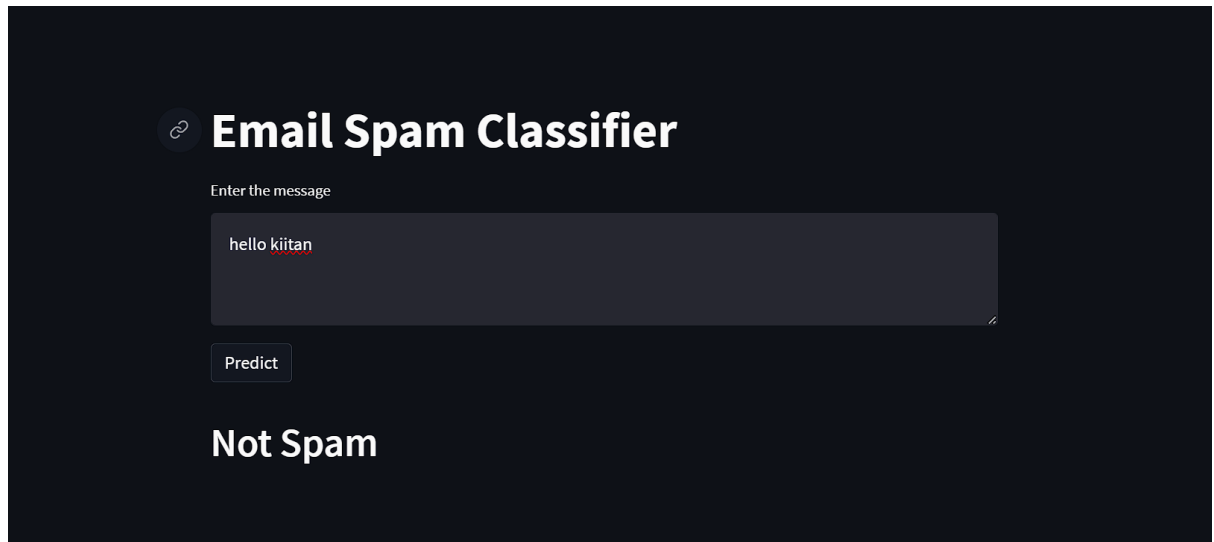


Here we can observe that our proposed model outperforms almost every other model in every metric. Only one model (naïve Bayes) has slightly higher accuracy than our model but it is considerably lagging in other metrics. The results are visually presented below for easier understanding and comparison.

From the above comparison bar chart we can clearly see that all models individually are not as efficient as the proposed method.

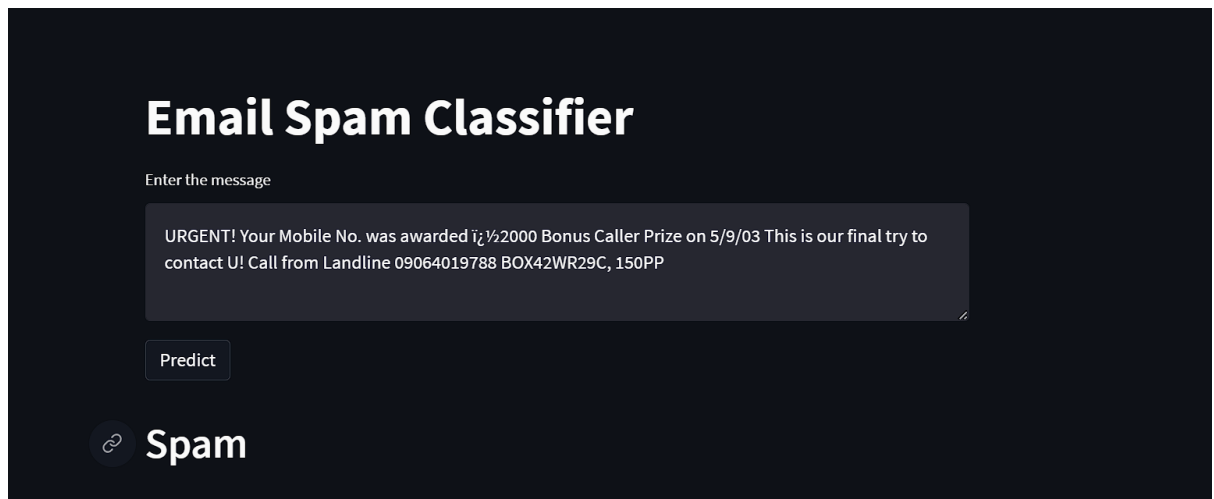
## 5.4 Result

### NOT SPAM DETECTION



The screenshot shows a web application titled "Email Spam Classifier". It has a dark theme. Below the title, there is a text input field with the placeholder "Enter the message". The input field contains the text "hello kiitan". Below the input field is a button labeled "Predict". Below the button, the prediction result "Not Spam" is displayed in a large, bold font.

### SPAM DETECTION



The screenshot shows the same "Email Spam Classifier" web application. The input field now contains a phishing message: "URGENT! Your Mobile No. was awarded ₹ 1/2000 Bonus Caller Prize on 5/9/03 This is our final try to contact U! Call from Landline 09064019788 BOX42WR29C, 150PP". The "Predict" button is visible. Below the button, the prediction result "Spam" is displayed in a large, bold font.

## **5.4 Summary**

There are two main tasks in the project implementation. Language model selection for completing the textual processing phase and proposed model creation using the individual algorithms. These two tasks require comparison from other models and select of various parameters for better efficiency.

During the language model selection phase two models, Bag of Words and TF-IDF are compared to select the best model and from the results obtained it is evident that TF-IDF performs better.

During the proposed model design various algorithms are tested with different parameters to get best parameters. Models are merged to form a ensemble algorithm and the results obtained are presented and compared above. It is clear from the results that the proposed model outperforms others in almost every metric derived.

# 6. Conclusion and Future Scopes

## 6.1 Conclusion

From the results obtained we can conclude that an ensemble machine learning model is more effective in detection and classification of spam than any individual algorithms. We can also conclude that TF-IDF (term frequency inverse document frequency) language model is more effective than Bag of words model in classification of spam when combined with several algorithms. And finally we can say that spam detection can get better if machine learning algorithms are combined and tuned to needs.

## 6.2 Future work

There are numerous applications to machine learning and natural language processing and when combined they can solve some of the most troubling problems concerned with texts. This application can be scaled to intake text in bulk so that classification can be done more effectively in some public sites.

Other contexts such as negative, phishing, malicious, etc., can be used to train the model to filter things such as public comments in various social sites. This application can be converted to online type of machine learning system and can be easily updated with latest trends of spam and other mails so that the system can adapt to new types of spam emails and texts.

## References

- [1] S. H. a. M. A. T. Toma, "An Analysis of Supervised Machine Learning Algorithms for Spam Email Detection," in International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), 2021.
- [2] S. Nandhini and J. Marseline K.S., "Performance Evaluation of Machine Learning Algorithms for Email Spam Detection," in International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020.
- [3] A. L. a. S. S. S. Gadde, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," in 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, 2021.
- [4] V. B. a. B. K. P. Sethi, "SMS spam detection and comparison of various machine learning algorithms," in International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), 2017.
- [5] G. D. a. A. R. P. Navaney, "SMS Spam Filtering Using Supervised Machine Learning Algorithms," in 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018. [6] S. O. Olatunji, "Extreme Learning Machines and Support Vector Machines models for email spam detection," in IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 2017.