Spam Detection in Twitter Using Machine Learning

Apra Kavdia
IT Department,
Smt. Kashibai Navale College of Engineering.

Preyasha Borse
IT Department,
Smt. Kashibai Navale College of Engineering.

Anushka Desai
IT Department,
Smt. Kashibai Navale College of Engineering.

Shubham Goel
IT Department,
Smt. Kashibai Navale College of Engineering.

Prof. Mrs. V. S. Khandekar, IT Department, Smt. Kashibai Navale College of Engineering.

***Abstract*-- With increase in the popularity on social media and millions and billions of people using it every day. This popularity of applications like Facebook, Twitter and Instagram grabs the attention of spammers. As through these they can trap genuine users with malicious activities. There is a significant amount of research done in this field. The primary focus of these researches is generally based on the accounts or users whose activity poses them as suspicious. These activities include posting of the same content, posting tweets that have no relevance to the trending topics and tagging them as one of the trending topics, sending bulk direct messages or users that have similar contents and are created on the same day. However, much of the research done focuses on spam accounts. There is little to none research done based on a model that marks tweets as spam and along with a sentiment analysis. In the proposed system, we propose a Machine learning system that would detect tweets as spam or ham. This spam detection would be done considering factors such as: shortened URLs, Emails that lead to malicious sites etc. The tweets would also be marked as spam based on the language. Using NLP, we would form a system that would mark tweets as spam if they have the potential to hurt sentiments of other users. The model would be trained and tested on a previously labelled dataset. This model would then be incorporated in a website that would take tweets as an input from the user. The result would be creation of a model that would give the tweet as spam or not based on the sentiment and spammer tactics.**

***Index Terms*-- Machine Learning, Spam Tweets, Ham Tweets, Sentiment Analysis, Natural Language Processing, Logistic Regression , Decision Tree, Random Forest, K - nearest Neighbour, Datasets, Feature Extraction**

NOMENCLATURE

ML: Machine Learning
NLP: Natural Language Processing
S^3D: Semi-Supervised Spam Detection
URL: Uniform Resource Locator
WEKA: Waikato Environment for Knowledge Analysis

I. INTRODUCTION

In this modern age, social media has become a part and parcel of everyone's life. Not only do people spend a significant amount of time using social media but also post a lot of personal stuff as well. Hence it becomes of utmost importance that there is a significant amount of security. Various social media sites which include

Twitter is a rapidly growing website and tweets are increasing day by day and so is spam and spammers. This popularity of tweets has led to several spam and exposure of sensitive information of users that can be dangerous if this information ends up in the wrong hands. Thus, it is necessary to detect and delete these spam Tweets.

Twitter is being used by countless users to stay connected professionally as well as informally. Not only commoners but many famous and socially as well as globally respected people use Twitter to stay connected to the masses.

The major motivation behind taking up this project is that we are the part of this modern age and hence it becomes necessary that the number of spam tweets be reduced making twitter a safe place to post up personal or professional tweets. The idea is to detect the spam tweets based on various algorithms and hence delete those spam malicious tweets.

Twitter, which was being started in 2006 has gained a lot of popularity and has been a rapidly growing website. With its continuously increasing popularity the number of spam tweets are also increasing which pose security threats. Hence it is necessary to detect as well as get rid of such spam tweets.

II. EXISTING WORK

There are various papers which give ideas about analyzing and predicting spam and ham tweets. The research shows that this article and papers proves to be very useful for determining the improved versions and methodologies for better results and appropriate analysis for tweets classification. As we look all over these research papers we get to know the diversities in approaches and methodologies.

A Performance Evaluation of Machine Learning Based Streaming Spam Tweets Detection is a research carried out in [11]. The research contained a performance evaluation based on three different aspects of the overall research i.e., dataset used, feature extracted and model made. This research gave an in detailed summary of the factors such as spam or non-spam ratio, training data size, data sampling, time related data and features discretization. Finally, the research shows models, features and data sets that are the most crucial and most used in creating a spam detection technique.

In [12], Detecting malicious tweets in trending topics using a statistical analysis of language research based on spam detection in trending topics. This research collected a dataset containing trending topics and extracted a feature that would be labelled as spam and non-spam and then this led to creation of a system that would be used in detection of spam tweets. Thus, use of an extension of the basic language models'

spammers are detected in trending topics. The system classified 89.3% and 93.7% in non-spam and spam, this percentage is correctly classified.

An Integrated approach for Malicious Tweets detection using NLP was carried out in [10]. This research is based on 2 aspects: identification of spam users without knowing its background and language analysis used for spamming in trending topics. They used SVM as their main algorithm for the classification. The use of WEKA led to the selection of SVM. The use of SVM experiments gave 95-97% results classifications as correct.

A Method based on NLP for Twitter Spam Detection [2] by Rahul, Kumar, Banani and Samir is based on the use of NLP to process language to find spammers. The detection techniques were based on how long a user is connected to another user. How many tweets per day do they post and when was this user created? This all was used to draw research that was helpful in detection of many spam users that were created automatically and whose purpose was to breach one's privacy.

A semi-supervised approach was used in the research done by Surendra and Auxin. In [3], they proposed a paper Semi-Supervised Spam Detection in Twitter Stream. The system mainly consisted of 2 modules: 1. Four light weight detectors and 2. Updating modules to periodically update the model. They created a S3D system which has 4 detectors that produced a system that provided a spamming pattern that was effective.

With the rapid increase in social network applications, people are using these platforms to voice their opinions with regard to daily issues. sentiment analysis (or opinion mining) is a common dialogue preparing task that aims to discover the sentiments behind opinions in texts on varying subjects. As such, the [9] paper explores the various sentiment analysis applied to Twitter data and their outcomes.

This span detection research is based upon the features extracted. A basic framework is suggested to detect malicious account holders in twitter. The system which they used works on machine learning based algorithms. In this [1], a system algorithm named Naïve Bayes classifier algorithm was used.

In this paper [4], They provide a survey and a comparative analysis of existing techniques for opinion mining like machine learning and lexicon-based approaches, together with evaluation metrics. Using various machine learning algorithms like Naive Bayes, Max Entropy, and Support Vector Machine, they provide research on twitter data streams. They have also discussed general challenges and applications of Sentiment Analysis on Twitter.

In the paper [5], they propose a framework which takes the user and tweet-based features along with the tweet text feature to classify the tweets. The benefit of using tweet text features is that we can identify the spam tweets even if the spammer creates a new account which was not possible only with the user and tweet-based features. They have evaluated our solution with four different machine learning algorithms namely.

III. PROPOSED METHODOLOGIES

The system would be built to mark streaming tweets as spam or ham. A taxonomy of spam detection is presented that classifies techniques based on their ability to detect: Fake content, URL based Detection, Detecting spam in trending topics, Fake user Identification. These would be the main points based on which the ML model would be trained and tested. This classification model would then be implemented to detect spam in streaming tweets. Accompanying this use of language would be done to determine if the sentiment of the tweet is positive, negative or neutral. NLP will be implemented to determine these results. The dataset extraction process involved:

a) General framework of proposed system:

The Framework Modules are:

Step 1 Data Collection: Collection of data is done in two ways: -

- Labelled data
- Live Tweets

Step 2 Training & Testing the Model: Based on labelled data, the model would be trained & tested.

Step 3 Classifying the User data: Three algorithms are used to classify if the data entered by the user is Spam or Not Spam. The algorithms are: -

- Naive Bayes
- Random Forest
- Decision tree
- K-nearest neighbor

Step 4 Twitter Stream: The raw data would then be fed to the classifier and the classifier would label them as spam or not, and also detect the sentiment of the tweet, then display the result.

Classification Result The entered tweet will be classified by the system model as SPAM or HAM, and the result will be displayed on the screen.

The below figure 1.1 shows the General Framework of our System.

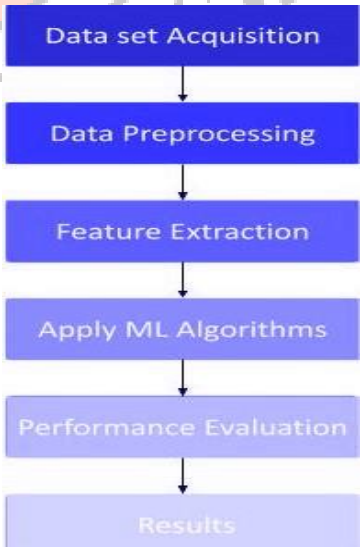

Figure 1.1 General Framework

b) Datasets:

An accurate global database (defined conditions with labels) is required to perform a wide range of learning tasks based on the spread of spam tweets. However, we have found that no data sets are publicly available specifically for our work. Because of this, we decided to collect tweets and make the world a reality.

Data Extraction - The tweet IDs were extracted from TwitterAPI .These tweet ids were in-turn used to extract the respective tweet attributes using Hydrator.An Hydrator is used to turn the Twitter IDs into Twitter JSON.The extracted dataset has two files :

1. File 1 contains tweets and their TweetID.
2. File2 contains TweetID and Spam or Ham columns. (1 : Spam & 0 : Ham).

These files were merged using the tweet ID. Hence, the final file contains Twitter id, Tweet text and spam and ham columns.

Balancing Data Set - Final Data-set contains 66,610 tweets in total, of which 4457 are spam tweets and 62153 are ham tweets.The following table 1.1 shows which features were extracted from the given dataset -

Table 1.1 Features Extracted from dataset

Feature Name	Values	Description
has_hashtag	binary	If tweet contains hashtag
num_hashtag	continuous	Number of hashtags in tweet
has_media	binary	If tweet contains media
has_URL	binary	If tweet contains URL
has_favourite_count	binary	If user has favorite count
has_place	binary	If tweet contains location
has_reTweet_count	binary	Count of tweet retweeted
is_retweet	binary	If tweet is retweet
has_user_description	binary	If user has description
has_user_followers_count	binary	Count of followers of user
is_user_verified	binary	If tweet is from verified user or not
length	continuous	Length of tweet
digit	continuous	Digits present in tweet
cap	continuous	Number of capital words in tweet

c) System Architecture:

Initially, we have used a raw dataset, which is further used for feature extraction. We have created a labelled dataset which consists of id and label of spam or ham. Then we have divided the dataset in 2 parts where one part is used as training dataset for the system and the other half is used as testing dataset on inputs provided.

To provide real time input to the system, the user must login (if already an existing user) or register (as a new user). The user is given the privilege of entering input as a tweet to the software and gives an input to the software system.

The system takes in input from the user and uses four algorithms i.e., Random Forest Algorithm / Naive Bayes Algorithm / Decision Tree / K - Nearest Neighbour. These algorithms are used to classify the input tweet as spam or ham. Then, the system displays the result as follows:

- If Tweet is spam: “The tweet entered is Spam Tweet”
- If Tweet is ham: “The tweet entered is Ham Tweet”

The below figure 1.2 shows the System Architecture of our Model.

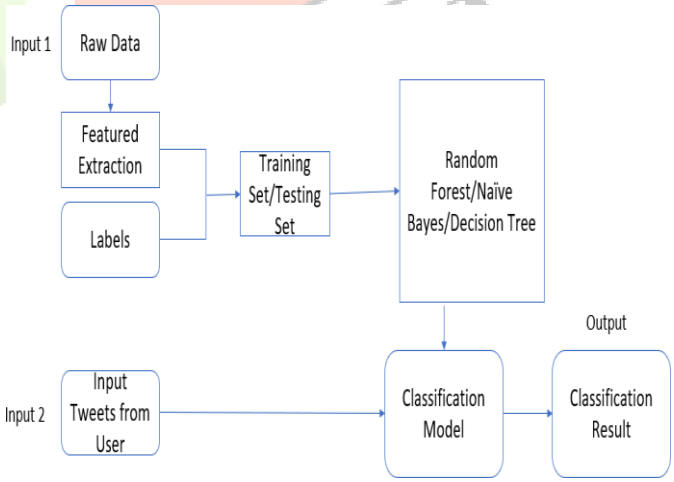

Figure 1.2 System Architecture

IV. Results

As we have used multiple ML models and algorithms for resulting different output parameters we have come up with a lot of defining results. In order to increase the accuracy, the algorithms are made to run multiple times with changed

settings and factors. Let us discuss the Performance matrix and results of all ML models used in the system.

a) Performance Matrix:

In order to evaluate the performance of spam detection approaches, some metrics imported from information retrieval are widely used by the researchers.

1. Positives and Negatives: Suppose there is a tweet t and the spam class S. The output of the classifier is whether it belongs to S or not. A common way to evaluate the classifier's performance is to use true positives (TP), false positives. ML-based spam detection process. (FP), true negatives (TN), and false negatives (FN) . These metrics are defined as follows:

a. TP tweets of class S correctly classified as belonging to class S.
b. FP tweets not belonging to class S incorrectly classified as belonging to class S
c. TN tweets not belonging to class S correctly classified as not belonging to class S.
d. FN tweets of class S incorrectly classified as not belonging to class S.

2. The relations of TP, FP, TN, and FN in social spam detection are shown in below table 1.2 as Evaluation Matrix. In order to measure the ability to detect spam, we also import true positive rate (TPR) and false positive rate (FPR).

Table 1.2 Evaluation Metrics

		Predicted	
		Spam	Ham
Actual	Spam	TP	FN
	Ham	FP	TN

a. TPR is defined as the ratio of those spam tweets correctly classified as belonging to class spam to the total number of tweets in class spam, it can be calculated by -

$$TPR = \frac{TP}{TP+FN}$$

b. FPR is defined as the ratio of those non spam tweets incorrectly classified as belonging to spam class S to the total number of nonspam tweets, and is calculated as -

$$FPR = \frac{FP}{FP+FN}$$

c. Precision, Recall, and Accuracy: Literature also uses precision, recall, and Accuracy to evaluate per-class performance.

i. Precision is defined as the ratio of those tweets that truly belong class S to those identified as class S, it can be calculated by -

$$Precision = \frac{TP}{TP+FP}$$

ii. Recall (which is also known as detection rate in the detection scenario) is defined as the ratio of those tweets correctly classified as belonging to class S to the total number of users in class S, it can be calculated by-

$$Recall = \frac{TP}{TP+FN}$$

iii. Accuracy is the fraction of predictions our model got right. combination of precision and recall, it is a widely adopt metric to evaluate per-class performance, it can be calculated by -

$$Accuracy = \frac{(TP+TN)}{(TP+TN+TF+Fn)}$$

b) Experimental Settings:

Random Forest Classifier: Random forest classifier is an ensemble machine learning algorithm used for classification and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class. Table 1.3 depicts the Confusion Matrix of Random Forest Classifiers.

Table 1.3 Random Forest Classifiers Results

True Positive Rate	78.33%
True Negative Rate	90.93%
Positive Predictive Value	91.76%
Negative Predictive Value	76.48%
False Positive Rate	9.07%
False Negative Rate	21.67%
False Discovery Rate	8.24%
Overall Accuracy	84.14%

Figures 1.3.1 and 1.3.2 depict ROC Curve and Precision Recall curve for Random Forest Classifier.

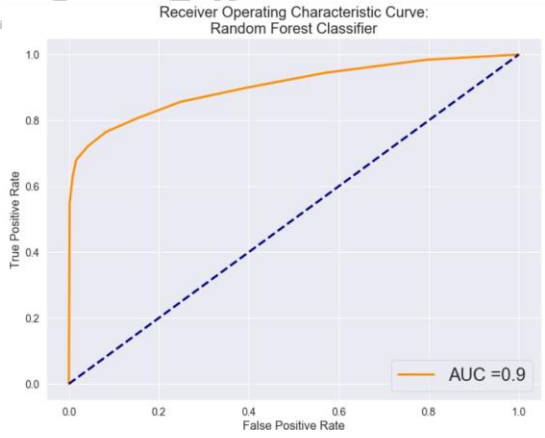

Figure 1.3.1 ROC Curve for Random Forest Classifier

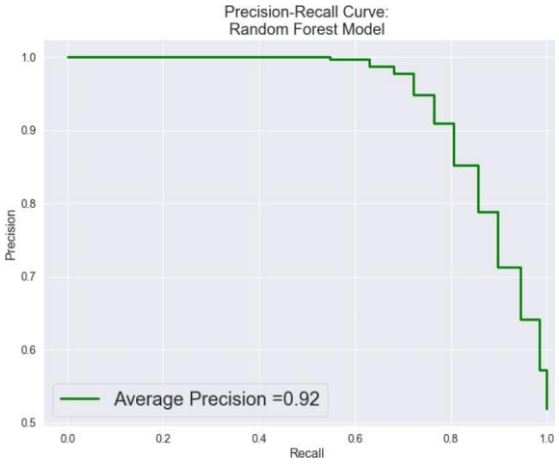

Figure 1.3.2 Precision-Recall Curve for Random Forest Classifier

K Nearest Neighbour: The K - nearest neighbour algorithm assumes that similar things exist in close proximity.KNN is an approach to data classification that estimates how likely a data point is to be a member of one group or the other depending on what group the data points nearest to it are. Table 1.4 depicts the Confusion Matrix of K - Nearest Neighbour.

Table 1.4 K - Nearest Neighbor Results

True Positive Rate	66.27%
True Negative Rate	83.32%
Positive Predictive Value	88.18%
Negative Predictive Value	56.81%
False Positive Rate	16.68%
False Negative Rate	33.73%
False Discovery Rate	11.82%
Overall Accuracy	72.19%

Figures 1.4.1 and 1.4.2 depict ROC Curve and Precision Recall curve for Random Forest Classifier.

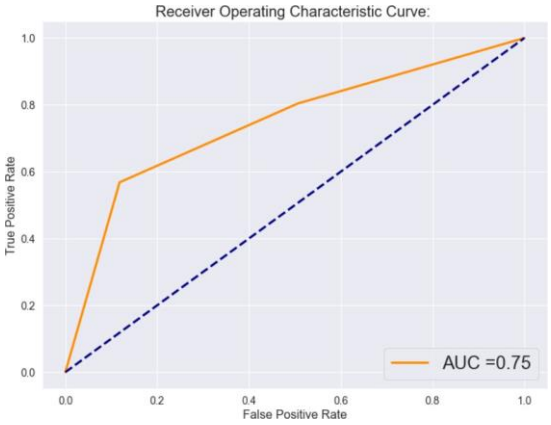

Figure 1.4.1 ROC Curve for K Nearest Neighbour

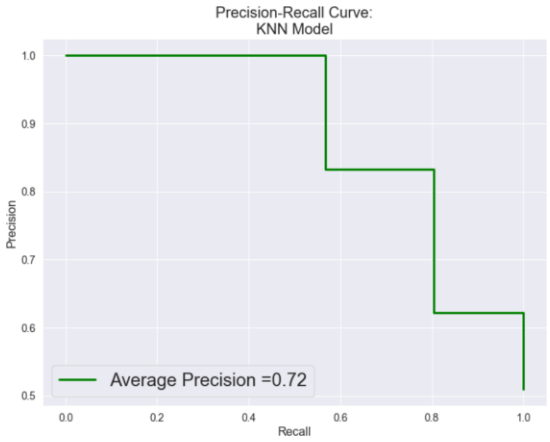

Figure 1.4.2 Precision-Recall Curve for K Nearest Neighbour

Decision Tree Classifier: Decision Tree is an ML model that splits the decisional tree nodes on all available variables and then selects the split which results in most homogeneous sub-nodes. Table 1.5 depicts the Confusion Matrix of Decision Tree Classifier.

Table 1.5 Decision Tree Classifier Results

True Positive Rate	75.64%
True Negative Rate	88.44%
Positive Predictive Value	90.22%
Negative Predictive Value	72.04%
False Positive Rate	11.56%
False Negative Rate	21.36%
False Discovery Rate	9.78%
Overall Accuracy	81.05%

Figures 1.5.1 and 1.5.2 depict ROC Curve and Precision Recall curve for Decision Tree Classifier.

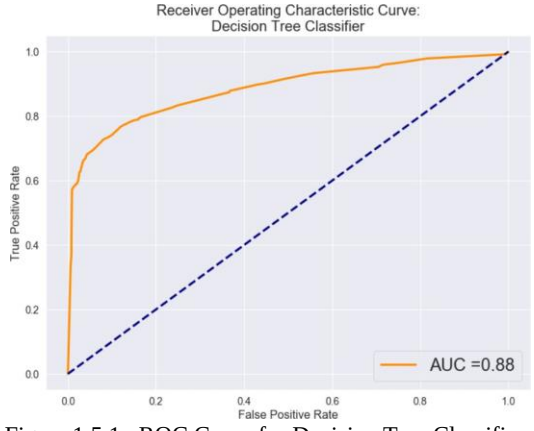

Figure 1.5.1 ROC Curve for Decision Tree Classifier

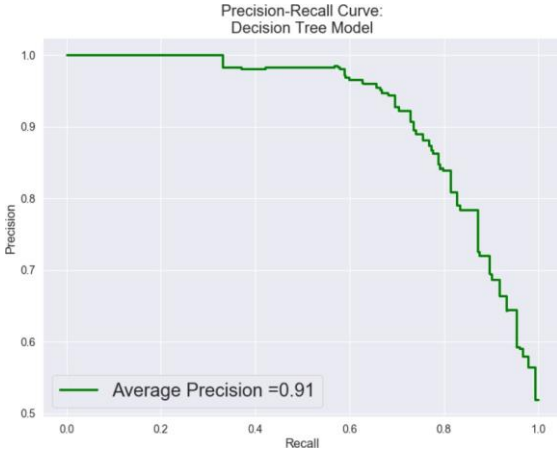

Figure 1.5.2 Precision-Recall Curve for Decision Tree Classifier

Naive Bayes: Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. A naive Bayes classifier considers each of these features to contribute independently to the probability, regardless of any possible correlations between their features. Table 1.6 depicts the Confusion Matrix of Naive Bayes.

Table 1.6 Naive Bayes Results

True Positive Rate	74.33%
True Negative Rate	98.37%
Positive Predictive Value	98.87%
Negative Predictive Value	66.53%
False Positive Rate	1.63%
False Negative Rate	25.67%
False Discovery Rate	1.13%
Overall Accuracy	82.54%

Figures 1.6.1 and 1.6.2 depict ROC Curve and Precision Recall curve for Naive Bayes.

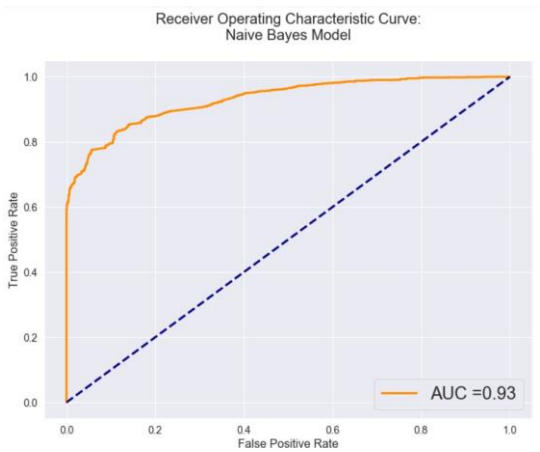

Figure 1.6.1 ROC Curve for Naive Bayes

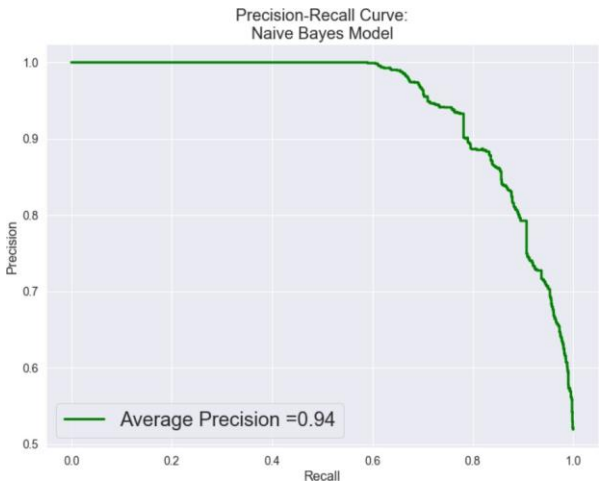

Figure 1.6.2 Precision-Recall Curve for Naive Bayes

c) Limitations:

In the above section, we have evaluated the impact of the dataset on the 4 machine learning classifiers. Each of these classfieres was trained with a dataset containing 66,610 tweets in total, of which 4,457 records were labeled as spam and 62,153 were labelled as non-spam tweets. To get a non biased evaluation from the classifier model.Balancing techniques were applied to the dataset.

i) Random Over Sampling : By applying random over sampling to our original dataset. We extracted a different dataset,which we consider as Dataset-A. The Dataset-A contains the same number of ham tweets and spam tweets.

ii) SMOTE-Tomek: By applying SMOTE-Tomek balancing technique to our original Dataset we have acquired Dataset-B.This dataset-B contains a reduced number of Ham tweets and an increased number of spam tweets.

After balancing both of these datasets (i.e dataset-A and database-B) these datasets were used to train our classifier models. And this resulted in increasing amounts of accuracy. Table 1.7 shows the Accuracy of classifier models.

Table 1.7 Model Accuracy

Model	Accuracy
Random Forest Classifier	94.78%
K - Nearest Neighbor	81.45%
Decision Tree Classifier	89.63%
Naive Bayes Algorithm	98.99%

The below figures 1.7.1 and 1.7.2 show ROC Curve and Precision Recall Curve of Models when balancing techniques were applied to the dataset.

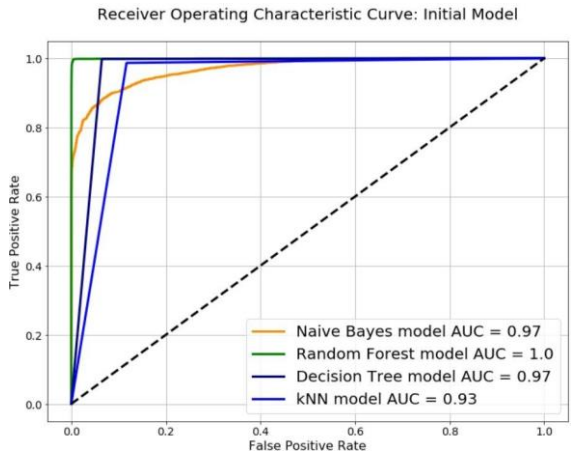

Figure 1.7.1 ROC for Overfitting

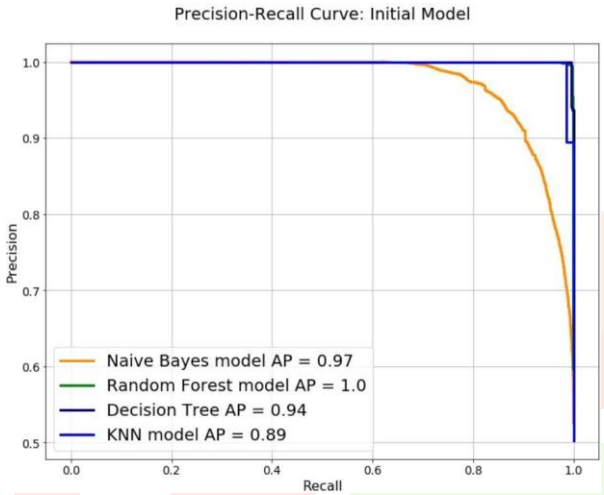

Figure 1.7.2 Precision Recall for Overfitting

In this section, we evaluate the performance of different classifiers on two kinds of datasets (randomly sampled and SMOTE), and find that classifiers have much better performance in detecting spam tweets when it was trained without original dataset which was balanced using the Random Undersampling technique.Making the size of the dataset as a limitation for the models to be accurate.

V. Conclusion

As of now, spamming has become one of the main issues and it should be solved in every social networking website. Spam Detection Framework on Twitter using Machine Learning helps people to solve their spam issues in an easy and accurate way so that spammers can be easily blocked and hence spam tweets are reduced.

The Machine Learning Model proposes and implements a framework for real time Spam Detection.We have extracted some new features from the dataset & the combination of features for detecting spam tweets has shown better performance in terms of accuracy, precision, recall, etc. The algorithms used for classing tweets as spam and ham in our models are Random Forest Classifier, Decision Tree Classifier, K - nearest Neighbour, Naive Bayes.

Our model takes input as spam tweets and classifies tweets asspam or ham.

Thus we can conclude that classifying tweets can be reliable for users. The purpose of this model is to reduce the divergence caused due to malicious tweets.

VI. References

[1] N. Noor Allema, “Spam Detection Framework for Twitter using ML”, IJITEE vol. 9, issue. 6, April 2020

[2] Ratul Chawdhury, “A Method based on NLP for Twitter Spam Detection”, vol. 2, issue. 6(2), 2020

[3] Surendra Sedhai, “Semi-Supervised Spam Detection in Twitter Stream", NTU Singapore, Feb 2017

[4] Vishal Kharde & S. Sonawane, “Sentiment Analysis of Twitter Data”, IJCA vol. 139 no. 11, April 2016

[5] Himank Gupta, “Framework of Real – Time Spam Detection in Twitter”, COMSNETS; IIT Hyderabad, January 10, 2018

[6] Akanksha S. Nagdeve & Manisha M. Ambekar, “Spam Detection by Designing Machine Learning Approach in Twitter Stream”, issue. 4, IEEE Paper, January 2021

[7] Miss. Richa Ramesh Sharma, Prof. Yogesh S. Patil & Prof. Dinesh D. Patil. “Twitter Spam Detection by Using Machine Learning Frameworks”, vol. 8, issue 5, May 2019

[8] K. Jino Abisha, J.Roshan Nilofer, A.Silviya, Dr. S. Raja Ratna “Detection of Twitter Spam's using Machine Learning Algorithm”, vol. 6, issue. 3, IJCSE Journal, 2019

[9] Abdullah Alsaeedi and Mohammad Zubair Khan, “A Study on Sentiment Analysis Techniques of Twitter Data”, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 10(2), 2019

[10] S. Gharge and M. Chavan, "An integrated approach for malicious tweets detection using NLP," International Conference on Inventive Communication and Computational Technologies (ICICCT), 2017.

[11] C. Chen et al., "A Performance Evaluation of Machine Learning-Based Streaming Spam Tweets Detection," in IEEE Transactions on Computational Social Systems, vol. 2, no. 3, September 2015.

[12] Juan Martinez-Romo, Lourdes Araujo,”Detecting malicious tweets in trending topics using a statistical analysis of language”, Expert Systems with Applications,vol. 40, issue. 8, 2020.