Automatic Tweet Spam Detection

Raneem Alqadi, Saslabeel Hammoudeh

1181712@student.birzeit.edu, 1180771@student.birzeit.edu

Abstract_ Twitter is a famous platform that allows individuals to express their view on certain topics that interest them and get updates about the current trending issues. But along with this increasing new ways was opened for spammers. Spammers find these online media an easy way to catch users in their malicious activities by spam messages. Their motive is to steal important information from users, exploit their privacy or antagonize the users. In this paper, we are developing a Twitter Based Spam Detection Model which will classify the tweet as spam or not using various features and classifiers.

Naïve Bayes, Random Forest and Neural Network classifiers gives recall 0.97, 0.98 , 0. 99 respectively after applying them in this project.

Keywords_ twitter, classification, detection, machine learning, spam, Naïve Bayes , Random Forest, Neural Network.

1. Introduction

Spam is irrelevant or unsolicited messages sent over the Internet. These are usually sent to a large number of users for a variety of use cases such as advertising, phishing, spreading malware, etc.

In the past, spam used to favor email as it was the primary communication way. Email addresses were relatively easy to harvest via chat rooms, websites, customer lists and that impact a user's address book. Eventually, email filters became more and more advanced, and to some extent, these technologies have greatly reduced spam emails.

Since then, spammers have moved onto a new target: social applications like Twitter and the other famous applications [1].

2. Related work

2.1. Twitter and Spam tweets

Twitter allows accounts to "follow" other different accounts that are of interest to them Compared to various other social media platforms, the connection between users is a bi-directional one and not a unidirectional connection since a particular user may not be following one of his followers. The user can like or retweet a tweet which implies sharing that tweet to his "followers". Every user has a unique Twitter username, and users can post tweets that talk about others by including their usernames beginning with "@" character which is referred to as "mention" on Twitter Users are promptly alerted with notifications when a mention, like, or retweet transpires on any of his tweets.

Twitter utilizes both manual and mechanized administrations to fight spammers. The manual method involves Twitter giving users a chance to report spammers via the spammers' profile pages.

A different method involves reporting spammers to the authority "@spam" account. Still, this strategy for revealing spam is obsolete. Likewise, Wang reports that this technique is mishandled by both inventors and spam. These manual methodologies are stressful and may not be sufficient to distinguish between all spammers because of billions of users [2].

Twitter utilizes different elements:

1. The publishing of a copy of the messages over numerous accounts or various copy of the messages in a single account.
2. The following/unfollowing of a huge number of accounts within a brief span.
3. The possessing of a huge number of spam protests documented against the account.
4. Forcefully liking, following, and retweeting.
5. The publishing of malignant connections.
6. The publishing of tweets which for the most part. comprise of connections rather than also publishing individual updates.
7. The presentation of inconsequential tweets to a trending subject to figure out what lead is thought to be spamming [2].

2.2. Naïve Bayes Classifier

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Naive Bayes can be extremely fast relative to other classification algorithms. It works on Bayes theorem of probability to predict the class of unknown data sets [3]. Bayes Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$p(A\ /B) = \frac{P(B\ /A)P(\ A)}{P(B)} \quad \ldots\ldots. 1$$

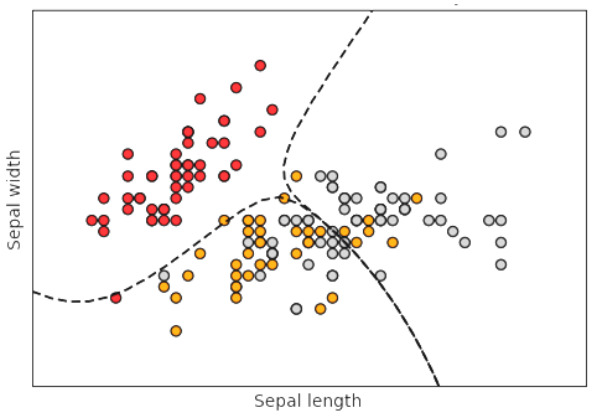

Figure 1: classification using Naive Bayes

2.3. Random Forest Classifier

Random forests or random decision forests are an supervised learning method for classification, regression including various operations because of that It is one of the most used algorithms, which is used by setting up a large number of decision trees at learning duration and outputting the class which is the approach of the mean prediction (regression) or classes (classification) of the separate trees. simply, random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. Random decision forests improve for the decision trees' manner of over-fitting to their learning plan. The grouping of training estimation can also be referred to as Random forests or random decision forests. It is managed by creating a large number of decision trees on the learning information array, after which result can be grouped because of the method of the classes.

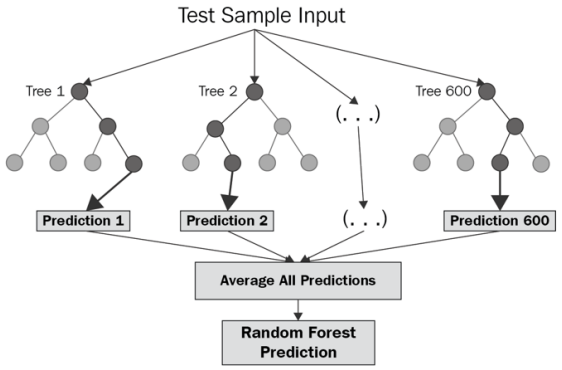

Figure 2: random forest algorithm

2.4. Neural Network Classifier

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated[4].

A neural network consists of units (neurons), arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer. Generally the networks are defined to be feed-forward: a unit feeds its output to all the units on the next layer, but there is no feedback to the previous layer. Weightings are applied to the signals passing from one unit to another, and it is these weightings which are tuned in the training phase to adapt a neural network to the particular problem at hand. This is the learning phase[5].

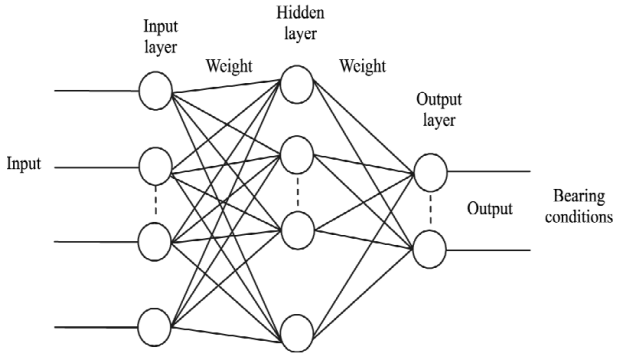

Figure 3: Neural network stages

3. Problem specification

The goal is to develop an automated solution that filters out the spam tweets based on extracted features from the tweet content and other attributes.

4. Data

Dataset was attached with project as a .csv file with **13234** sample. Each line of dataset contains the following fields:

Column Name	*Description*
id	The number of tweet
Tweet	This is the text that was tweeted
Following	The number of people the account that tweeted is following
Follower	The number of people following the account that tweeted.
Action	The total number of favorites, replies, and retweets of said tweet.
is_retweet	binary [0,1] value: If 0 its not a retweet, if 1 it is a retweet.
location	The self-written location provided by the user on their profile, May not exist, be "unkown", and is NOT standardized!.
Type	Type either Quality or Spam **(label).**

5. Why Python?

We chose Python to implement this project because python is the most popular language for AI and ML and based it on a trend search results on indeed.com [7]. there are a lot of reasons for these results:

✓ **A great library ecosystem**

A great choice of libraries is one of the main reasons Python is the most popular programming language used for AI. A library is a module or a group of modules published by different sources which include a pre-written piece of code that allows users to reach some functionality or perform different actions. The following are some of these libraries:

- **Scikit-learn** for handling basic ML algorithms like clustering, linear and logistic regressions, regression, classification, and others.
- **Pandas** for high-level data structures and analysis. It allows merging and filtering of data, as well as gathering it from other external sources like Excel.
- **Keras** for deep learning. It allows fast calculations and prototyping, as it uses the GPU in addition to the CPU of the computer.
- **TensorFlow** for working with deep learning by setting up, training, and utilizing artificial neural networks with massive datasets.
- **Matplotlib** for creating 2D plots, histograms, charts, and other forms of visualization.
- **NLTK** for working with computational linguistics, natural language recognition, and processing.
- **Scikit-image** for image processing.
- **PyBrain** for neural networks, unsupervised and reinforcement learning.
- **Caffe** for deep learning that allows switching between the CPU and the GPU and processing 60+ mln images a day using a single NVIDIA K40 GPU.

✓ **Flexibility**

Python for machine learning is a great choice, as this language is very flexible, It offers an option to choose either to use OOPs or scripting. There's also no need to recompile the source code, developers can implement any changes and quickly see the results. Programmers can combine Python and other languages to reach their goals.

✓ **platform Independence**

✓ **readability**

✓ **good visualization options**

Libraries like Matplotlib allow data scientists to build charts, histograms, and plots for better data comprehension, effective presentation, and visualization. Different application programming interfaces also simplify the visualization process and make it easier to create clear reports.

✓ **Community Support**

✓ **Growing Popularity[6]**

6. Proposed work (Stages)

The **5** main areas of the machine learning process:

1- **Data collection and preparation:** everything from choosing where to get the data, up to the point it is clean and ready for feature selection/engineering.

2- **Feature selection and feature engineering:** this includes all changes to the data from once it has been cleaned up to when it is ingested into the machine learning model.

3- **Choosing the machine learning algorithm and training our first model:** getting a "better than baseline" result upon which we can (hopefully) improve.

4- **Evaluating our model:** this includes the selection of the measure as well as the actual evaluation; seemingly a smaller step than others, but important to our end result.

5- **Model tweaking, regularization, and hyper parameter tuning:** this is where we iteratively go from a "good enough" model to our best effort[6].

These stages are displayed in this figure.

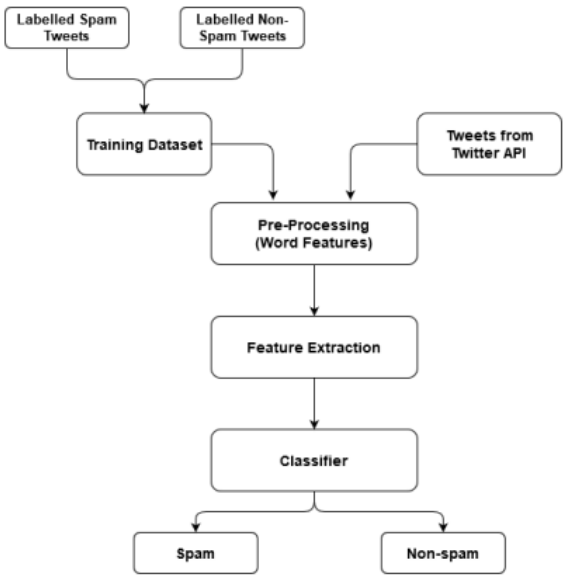

Figure 4: the ML strategy

6.1. Data collection and preparation

The quantity & quality of data dictate how accurate our model is so the outcome of this step is generally a representation of data which we will use for training. But dataset in our situation was given by Dr Aziz as mentioned in the Data section above.

We started preparing data for training by following these steps:

1- train.csv file was read using pandas library.

2- WordNetLemmatizer() method from nltk library was used to take the origin of words. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.

3- All non alphabet nor digit characters were replaced by space character.

4- The tweet was converted to lower case.

5- Stop words were removed.

6- to represent sentence by numbers Vectorization operation was needed. CountVectorizer is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. This is helpful when we have multiple such texts, and we wish to convert each word in each text into vectors. CountVectorizer creates a matrix in which each unique word is represented by a column of the matrix, and each text sample from the document is a row in the matrix. The value of each cell is nothing but the count of the word in that particular text sample. The CountVectorizer provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary.

After the preprocessing the data set was Split into training and evaluation sets with ratio 80% for training and 20% for testing.

6.2. Feature Selection

Choosing used features is an important process Therefore, we benefited from some papers and researches published in this field (**see references**), so determining the features to use was after trying and experimenting on more than one feature and more than one classifier as follows:

At the beginning, we tried to discover the affect of the tweet content alone, because we predict that it has the most affect in this process, and the obtained accuracy was about 70% for naïve bayes classifier.

After that we started to add features and noticing their impact, by adding the re_tweet feature the result The result is slightly improved. So we defined a new feature called ratio.

$$Ratio = \frac{\#followers}{\#followers+\#followings} \quad \ldots\ldots 2$$

This new feature has no clear effect on Naïve bayes and it didn't get better but it rises the random forest classifier result accuracy to %99 which is excellent accuracy (**note:** if ratio = $\frac{0}{0}$ it considered to be zero).

We decided to add a new feature which is the actions (The total number of favorites, replies, and retweets of said tweet) and the accuracy was excellent for all our three classifiers (Naïve Bayes, Random Forest, Neural Network) and **the obtained accuracies were 97%, 99%, 99% respectively. the final features that we decided to adopt are tweet text, actions, ratio.**

6.3. Choosing Models

We chose three classifiers to use in this project and they are:

1- Naïve Bayes
2- Random Forest
3- Neural Network

6.4. Training

The goal of training is to answer a question or make a prediction correctly as often as possible. Training was applied over 80% of dataset using the previous 3 classifiers.

6.5. Performance Measurements

Several performance measurements are suggested to estimate the classification model, and they include recall, accuracy, precision, and so on. It is not advisable to estimate the classifier utilizing a single performance measure so we used many of them.

We tested the model against previously unseen data Which are about 20% from all data set that we didn't use in training our model. And the performance measurements that we used are: Accuracy, precision, recall, f1, Confusion Matrix.

6.5.1. Confusion matrix

The total number of correct and incorrect predictions which are made by classifying the model compared to the actual outcome of the data is shown in a confusion matrix. The actual outcome is the targeted value. Matrix is NxN where N stands for a total number of target values. The data in the matrix is usually used to evaluate the performance of models.

Table 1:confusion matrix

Confusion Matrix	Predicted Positive	Predicted Negative
Actual Positive	True Positives (TP)	False Negatives (FN)
Actual Negative	False Positives (FP)	True Negatives (TN)

6.5.2. Accuracy

The most unlearned performance measure is the Accuracy. It is the rate of accurately classified tags on every forecast which can also be estimated by employing the formula below.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad \ldots\ldots. 3$$

6.5.3. Precision

The rate of the accurate affirmative tag over all the accurate forecasts, encompassing accurate inspections which are inaccurate, can be termed as Precision. It can be calculated using the formula below.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad \ldots. 4$$

6.5.4. Recall

Is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FN}} \quad \ldots. 5$$

6.5.5. F1 Scale

F1 score combines precision and recall into one measure.

$$F1 = \frac{2}{\frac{1}{precision}+\frac{1}{recall}} \quad \ldots. 6$$

7. Experiments and results

Our user interface was designed to be simple and creative at the same time. The idea is to make the buttons arranged in a sequential order in a way that shows the order of the tweet spam detection process started from preprocessing data, then feature extraction and training model after that checking performance measurements and showing a report.

The UI has a **tab bar** that contains three tabs for each classifier and this is a very easy way to move between them.

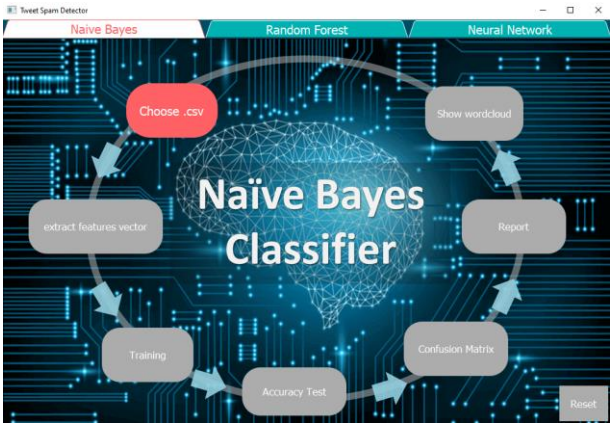

Figure 5: the main UI

After choosing the first button a file chooser window was appeared, then all that we must to do is to choose the data set csv file.

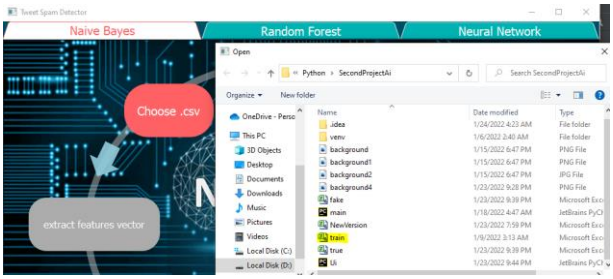

Figure 6: first step (choosing the csv file)

After all previous mentioned steps (preprocessing, feature extraction, training, testing) confusion matrix is shown as the next figure show.

Table 2: confusion matrix table of Naive bayes

TP = $\frac{656}{1817} \approx 36\%$	**FN** = $\frac{23}{1817} \approx 1.2\%$
FP = $\frac{40}{1817} \approx 2.2\%$	**TN** = $\frac{1128}{1817} \approx 62\%$

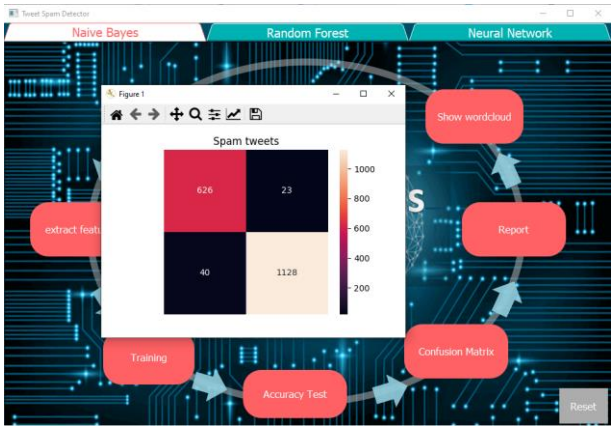

Figure 7: Confusion matrix

Based on confusion matrix a report was resulted contains some important performance measurements.

Table 3: evaluation table of Naive bayes

Accuracy	*Precision*	*recall*	*F1_Score*
97%	94%	96%	95%

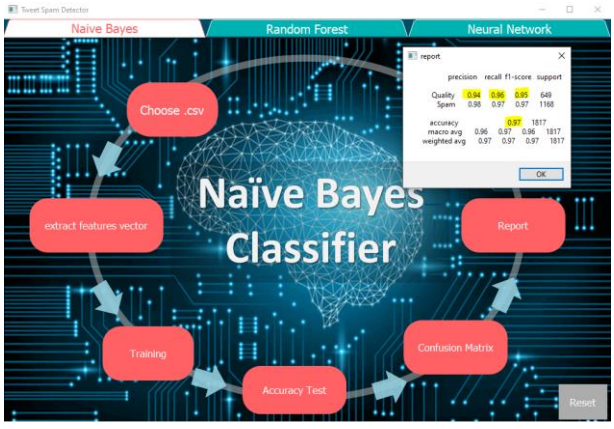

Figure 8: the resulting report

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. We used this concept to see which words are most repeated in spam tweets and the most repeated words in quality. Word cloud visualizations automatically apply lemmatization to our responses so we can visualize our responses by their key words. For example, "transforming" and "transformed" would both appear as the base word, "transform."

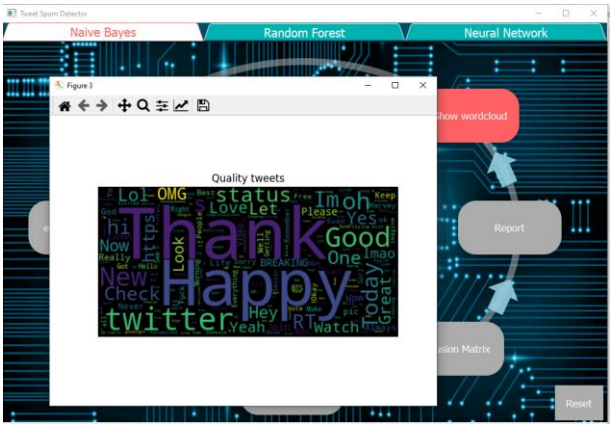

Figure 9: word cloud of spam tweets

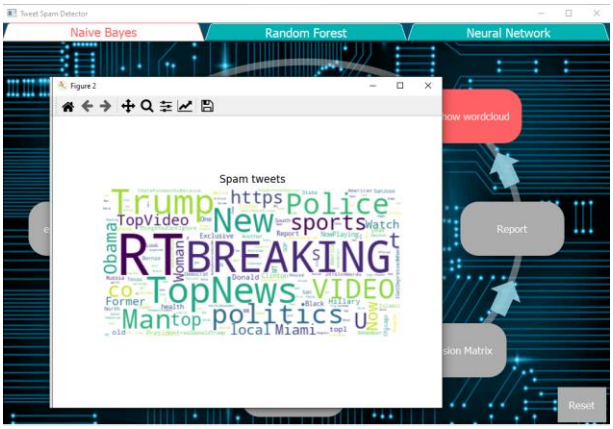

Figure 10: word cloud of quality tweets

A reset button was used to reset our system and begin a new cycle.

After Reset

Table 4: confusion matrix table of Random forest

$\mathbf{TP} = \frac{648}{1817} \approx 35\%$	$\mathbf{FN} = \frac{10}{1817} \approx 0.6\%$
$\mathbf{FP} = \frac{8}{1817} \approx 0.4\%$	$\mathbf{TN} = \frac{1151}{1817} \approx 63\%$

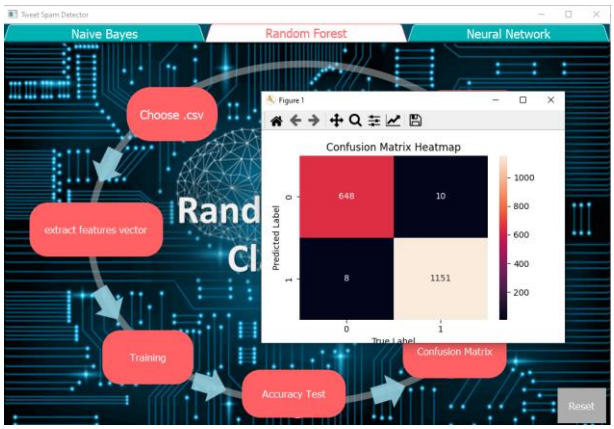

Figure 11: confusion matrix of random forest algorithm

Table 5evaluation table of Random forest

Accuracy	*Precision*	*recall*	*F1_Score*
97%	99%	98%	99%

Figure 12: report of random forest

Figure 13: Random forest ends

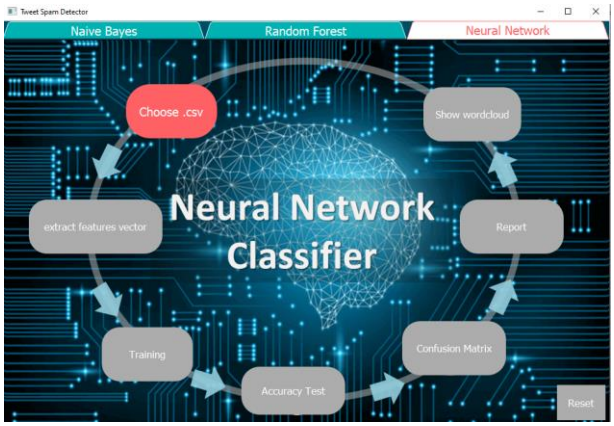

Figure 14: Neural Network UI

Table 6:confusion matrix table of Neural network

TP = $\frac{640}{1817} \approx 35\%$	**FN** = $\frac{9}{1817} \approx 0.5\%$
FP = $\frac{25}{1817} \approx 1.3\%$	**TN** = $\frac{1143}{1817} \approx 63\%$

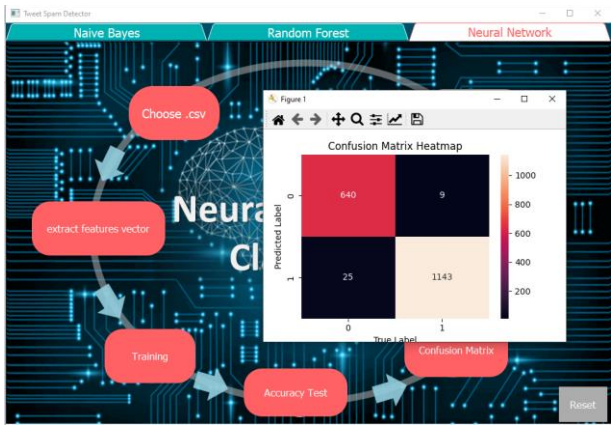

Figure 15: Neural Network Confusion Matrix

Table 7: evaluation table of Neural Network

Accuracy	*Precision*	*recall*	*F1_Score*
98%	96%	99%	97%

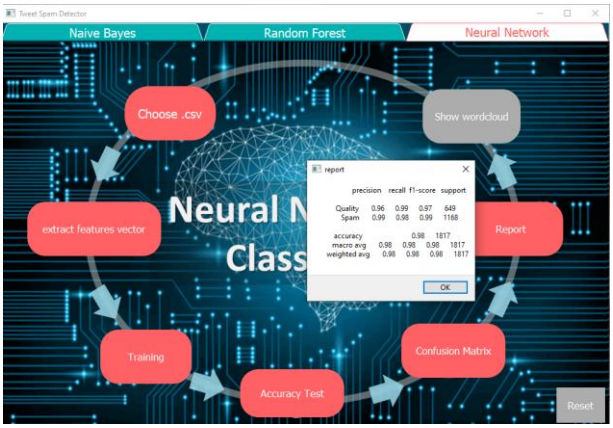

Figure 16: Neural Network Report

8. Conclusion

Table 8: Conclusion Table

	Naïve bayes	Random Forest	Neural Network
True	98.18%	99%	98.13%
False	1.82%	1%	1.87%
Accuracy	97%	99%	98%
Precision	94%	99%	96%
Recall	96%	98%	99%
F1-score	95%	99%	97%

When we want to detect an email if it is spam or quality we want to reduce the number of FP (a quality classified as spam) so we want to increase the Precision. **But** in tweeter the number of spam tweets is huge and because of repetition of information there is no significant effect when deleting something not spam while filtering tweets, our main interest here is to reduce FN (a spam classified as quality) so the best suitable performance measurement in this case is **recall. Although the Random Forest algorithm is better in most respects (Accuracy, Precision, F1-score) the Neural network algorithm has the best recall value, But there is no doubt that these two algorithms are better than Naïve bayes algorithm.**

9. References

[1] ***https://thenextweb.com/news/5-types-of-social-spam-and-how-to-prevent-them***. [Accessed: Jan. 19, 2022]

[2] AcademicianHelp, "Spam Detection on Twitter". [Online].Available:***https://www.academicianhelp.co.uk/assets/sampledocs/Dissertation__Spam_Detection_on_Twitter_20180927123949.pdf.*** [Accessed: Jan. 19, 2022]

[3] **Sunil Ray September 11,2017** ***https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/*** [Accessed: Jan. 20, 2022]

[4] ***https://www.robots.ox.ac.uk/~dclaus/digits/neural.htm#:~:text=Neural%20Networks%20as%20Classifiers,on%20to%20the%20next%20layer***. [Accessed: Jan. 20, 2022]

[5] ***https://wiki.pathmind.com/neural-network*** [Accessed: Jan. 21, 2022]

[6] ***https://towardsdatascience.com/8-reasons-why-python-is-good-for-artificial-intelligence-and-machine-learning-4a23f6bed2e6#:~:text=Python%20for%20machine%20learning%20is,and%20quickly%20see%20the%20results*.** [Accessed: Jan. 22, 2022]

[7] ***https://il.indeed.com/jobs?q=python&l&vjk=8c0fdccf036c206a***. [Accessed: Feb. 11, 2022]