

# **Email/SMS Spam Classifier**

## **A MINOR-II PROJECT REPORT**

*Submitted by*

**FAHAD RAFIQUE 2019-310-042**

*in partial fulfillment for the award of the degree of*

## **B. TECH COMPUTER SCIENCE & ENGINEERING**

*Under the supervision of*

**Ms. Shabina Ghafir**



**Department of Computer Science & Engineering  
School of Engineering Sciences & Technology**

**JAMIA HAMDARD**

New Delhi-110062

**(2022)**

## **DECLARATION**

I, **Mr. Fahad Rafique** a student of **Bachelors of Technology in Computer Science & Engineering (B.Tech CSE)**, **Enrolment No: 2019-310-042** hereby declare that the Project/Dissertation entitled **“Email/SMS Spam Classifier”** which is being submitted by me to the Department of Computer Science, Jamia Hamdard, New Delhi in partial fulfillment of the requirement for the award of the degree of **Bachelors of Technology Computer Science & Engineering (B.Tech CSE)**, is my original work and has not been submitted anywhere else for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

Fahad Rafique

2019-310-042

**Date: Dec 2022**

**Place: New Delhi**

## **ACKNOWLEDGEMENT**

I express my sincere thanks to Ms. Shabina Ghafir (Assistant Professor, Dept. of Computer Science and Engineering), my project in charge, who guided me through the project for her valuable suggestions and guidance for completing the project. This project has been a success only because of her guidance.

I deeply express my sincere thanks to our Head of Department Dr. Farheen Siddiqui for encouraging me and allowing me to present the project on the topic “Email/SMS Spam Classifier” at the department premises for the partial fulfilment of the requirements leading to the award of B.Tech degree.

I am also thankful to the whole computer science and engineering department for providing the technical support to carry out the project work, letting us utilize all the necessary facilities of the institute and providing guidance at each & every step during the project work.

# **INDEX**

TITLE	PAGE NO.
1. Objective	5
2. Introduction	6
3. Problem Statement	16
4. Software Requirements Specifications	17
5. Entity-Relationship Diagram	19
6. Snapshots of the different input and output screens	20
7. Conclusion	23
8. Limitation	24
9. Bibliography	26

## **OBJECTIVE**

The main aim of this project is to combine multiple services and open-source tools to make a system that will classify whether the e-mails and sms are Spam or non-spams.

# **INTRODUCTION**

In the modern world where digitization is everywhere, SMS has become one of the most vital forms of communication, unlike other chatting-based messaging systems like Facebook, WhatsApp, etc., SMS does not require an active internet connection at all. As we all know Hackers / Spammer tries to intrude into Mobile Computing Device, and SMS support for mobile devices had become vulnerable, as attacker tries to intrude into the system by sending an unwanted link, which on clicking those link the attacker can gain remote access over the mobile computing device. So, to identify those messages Authors have developed a system that will identify such malicious messages and will identify whether or not the message is SPAM or HAM (malicious or not malicious). The authors have created a dictionary using the TF-IDF Vectorizer algorithm, which includes all the features of words a SPAM SMS possess, based on the content of the text and referring I to the dictionary of the system and further it will be classifying the SMS and e-mail as spam or ham.

SMS is one of the most effective forms of communication. It is based on a cellular communication system and only your mobile phone needs to be in the coverage area of the network to send and receive messages.

Most people use this service for communication.

Various organizations using SMS to communicate with their clients, banks and other government agencies also use his SMS to communicate.

Many business organizations also use this service for advertising purposes. SMS therefore plays an important role as this framework does not require any active Internet connection.

The widespread use of SMS makes it one of the favorite places of hackers and spammers. Hackers can easily compromise someone else's cell phone by simply sharing or broadcasting a malicious link to their users. When an end-user clicks on a link or message sent by a hacker/spammer, her mobile device is automatically compromised. Once a hacker has control of your system, you can learn the rest of the ways hackers can exploit it. Therefore, limiting the content that end users receive has become very important. Therefore, we need a system to let end users know if a received message is spam. Non-spam messages he called HAM.

We identified the above problems and issues and developed a system that can detect whether it is spam or HAM from the content of a message using machine learning techniques. In this section, the author gave a brief overview of machine learning. Different types of machine learning and the techniques the authors used to develop machine learning models.

Each point in this article is considered in the context of a large social app that spans many countries.

Users are offered a level of free SMS messaging to any number and have proven to be an excellent service in areas with very limited internet connectivity and very high mobile data costs.

This naturally attracted the attention of scammers and scammers who found opportunities to make shady transactions very cheaply and led to many attempts to exploit the platform. This model and related procedures were

developed in response to this situation to maintain the quality of user experience within the app.

### **Machine Learning: -**

Machine learning is an interesting field because it covers important parts of different disciplines: statistics, artificial intelligence theory, data analysis, and numerical methods. Machine learning can be defined as the semi-automatic extraction of knowledge from datasets or data.

Let's split the definition into three components.

- i) First, machine learning always starts with data, with the goal of extracting insights from the user's data or dataset.
- ii) Second, machine learning involves some degree of automation rather than manually trying to extract insights from data.
- iii) Finally, machine learning is not fully automated. H.

A successful process requires human intervention to make many intelligent decisions. Simply put, machine learning is an application that allows you to improve your predictions with iterations and with experience. The process by which an application improves with experience is, of course, called training.

Larger iterations may be required to gradually improve results. During training, data is passed to a machine learning algorithm that improves internal representations and numerical parameters when discrepancies or training errors occur. The goal of this stage is to minimize the cost and error functions or adjust the internal weights of the algorithm to maximize the probability. As the algorithm improves in accuracy, it is called learning. Once the results are sufficiently accurate (also known as scoring), machine



learning applications can be used to solve the problems they were designed to solve.

**Machine learning can be divided into two categories:**

**a) Supervised learning**

**b) Unsupervised learning.**

**Supervised Learning: -**

Supervised learning, also known as predictive modeling, is the process of making predictions based on data. Examples of supervised learning include classification<sup>5</sup> and regression<sup>6</sup>. Training datasets for supervised learning are pre-tagged for known function values for classification problems or regression. Once the training is complete and the model has a minimum cost function on the training data set, we proceed to scoring where we can later predict values for new data.

**Classification:** It identifies group membership. That means that if we have multiple events characterized by input parameters, which can be labeled differently, we want our system to predict which label should be used.

**Regression:** Regression is a combination of multi-dimensional power supply and function interpolation. .

**Regression:** The regression problem is used to find the approximation of the function with a minimum error deviation or a cost function. In other words, the regression technique simply tries to predict numeric dependence, a function value.

## **Unsupervised Learning:**

Unsupervised learning is the process of extracting structure from data, or how to best represent data.

In unsupervised learning situations, where the algorithm automatically detects features in the data, it depends on the algorithm's goals and assumptions.

## **What is spam and why should it be prevented?**

Spam is unsolicited and unwanted messages sent electronically that may be malicious in content. I have. Email spam is sent and received over the Internet, while SMS spam is typically sent over cellular networks. A user who sends spam is called a "spammer". SMS messages are usually very cheap (if not free) for users to send, which makes them attractive for illegal exploits. To make matters worse, SMS is almost always considered a more secure and reliable form of communication than other sources. E.g. Email.

The dangers of spam messages to users are numerous. Unwanted ads, disclosure of personal information, falling victim to fraud or financial schemes, being directed to malware or phishing websites, or being unknowingly exposed to inappropriate content. For network operators, this leads to increased spam message operational costs.

In this case, spam is annoying to users, negatively impacting quality of service, and damaging your brand. This can lead to complaints, low ratings and even loss of users, not to mention users getting scammed.

## Differences with Email & Spam: -

The following table summarizes the main differences between spam in email and SMS.

SMS	EMAIL
Short messages	Can have any length
Sent (mostly) through mobile connections	Transmitted through any internet connection
Ambiguous intention	Greater length makes the intention clearer
Content can be plain text, string characters and possibly emojis	Has a subject, formatted text, multimedia content and attachments
Usually regarded as trustworthy	There is widespread awareness about spam emails

### Spammer's behavior

Spammers attempt to test an operator's anti-spam infrastructure by sending varying amounts of spam to determine if volume barriers are in place.

Using multiple numbers to send messages is very common and we rule out number blocking as an anti-spam strategy. requires some form of content-based filtering that

**In-Service Spam Filtering Status** At the beginning of the project, the only anti-spam measure was to block users whose number of SMS sent exceeded her daily and hourly thresholds. At that time, there was no content-based filtering or consideration based on the user's metadata. It was a rules-based system, very easy to circumvent, and used very little data.

## **STEPS INVOLVED IN BUILDING THE PROJECT: -**

1. Data cleaning
2. Exploratory Data Analysis
3. Text Preprocessing
4. Model building
5. Evaluation of the model on the basis of accuracy
6. Improvement in the model
7. Website

1. Data Cleaning: This is the first step that involves the importing of the dataset (for this model we have chosen it from the kaggle). The aim of information cleaning right here is to find the proper manner to rectify high-satisfactory issues like eliminating bad data, and filling in missing data to form the information efficient for the model.

2. Exploratory Data Analysis: This is the second step in which EDA has been conducted on the 2 Columns and exploring and gaining more knowledge about the data.

This dataset initially contains 87.35% of Ham and 12.63 Spam Data which is technically imbalanced for the model.

3. Text Preprocessing: This step involves the following transformations.

- Lower case
- Tokenization
- Removing special characters
- Removing stop words and punctuation
- Stemming (words with same meaning)

#### 4. Model building: -

Email spam detection is a classification problem. Some algorithms such as Naive Bayes Classifier and Decision Trees are good for spam detection. Algorithms such as KNN, linear regression, etc. do not perform well in practice due to their inherent drawbacks such as the curse of dimensionality.

This project trained multiple models and in the end compares their accuracy with each other.

- Naïve Bayes with TF-IDF approach: Naive Bayes is the simplest classification algorithm (quick to create and regularly used for spam detection). This is a common text classification problem that uses word frequency as a feature to determine whether a document belongs to one category or another (e.g., spam or legal, sports or politics, etc.).
- Decision Trees Decision trees are used for classification and regression. Decision analysis provides a visual and explicit representation of decisions and decisions using decision trees. The decision to make strategic splits greatly affects the accuracy of the tree. Decision criteria are different for classification trees and regression trees.

Information theory has a measure that defines this degree of confusion in a system known as entropy. If the sample is perfectly uniform, the entropy is zero, and if the model is evenly split

the entropy is 1. The split with the lowest entropy compared to the parent node and other splits is chosen. The lower the entropy, the better.

- SVM: SVM is a supervised machine learning algorithm that can be used for both classification and regression tasks. SVM is mainly used in classification related problems. The algorithm plots each data item as a point in an n-dimensional space (where n is the number of features it possesses). where the value of each feature is the value of a specific coordinate. Classification is then performed by finding hyperplanes that distinguish the two classes very well. The support vector machine is the boundary that best separates the two classes (hyperplane/line). If your data requires nonlinear classification, SVM can use kernels. This is a function that takes a low-dimensional input space and transforms it into a high-dimensional space. H. They transform inseparable problems into separable problems.
- Random Forest: A random forest is like a bootstrapping algorithm using a decision tree model (CART). Random Forest attempts to build multiple CART models with different samples and different initial variables. For example, build a CART model using a random sample of 100 observations and 5 randomly chosen initial variables. Repeat this process (for example) 10 times and make a final prediction for each observation. The final prediction is a function of each and single prediction of the model. This final prediction is simply the average of each prediction. Random Forest provides more accurate predictions in

many scenarios compared to simple CART/CHAID or regression models. These cases typically have a large number of predictors and a large sample size. This is so that we can obtain the variances of multiple input variables simultaneously, allowing a large number of observations to participate in the prediction.

## **PROBLEM STATEMENT**

Spam classification has historically been viewed as a binary classification problem. This is where the most original aspects of our approach become apparent. We abandon the classification related problem in the favor of a regression problem which aims to predict the spam probability of a text message.

- Large datasets of spam SMS are not publicly available. Even so, we never expect training on these datasets to yield good performance in our context.
- Lack of a pipeline to transform SMS logs into a structured and clean dataset.
- Apps are available in many countries and languages, adding to the complexity.
- The ultimate model should be deployed and integrated into the app's current infrastructure, taking the necessary precautions to avoid high costs and message delivery delays.
- Subjective Criteria for Labeling: Can religious propaganda be blocked even if it is not fraudulent or deceptive? What if the message is sent to thousands of users?
- Message Ambiguity: Even humans have difficulty distinguishing between genuine messages and spam.



# **SOFTWARE REQUIREMENT** **SPECIFICATIONS**

1. PyCharm
2. Jupyter Notebook
3. Streamlit (Frontend)
4. Some Preinstalled libraries (Pickle, Pandas, Scikits Learn etc)

## **PyCharm: -**

PyCharm is a computer programming integrated development environment (IDE) that focuses on the Python programming language. JetBrains, a Czech firm, developed it (formerly known as IntelliJ). It includes code analysis, a graphical debugger, an integrated unit tester, VCS integration, and Django and Anaconda support for web development and data science, respectively.

PyCharm is available in Windows, Mac OS X, and Linux versions. The Community Edition is licensed under the Apache License.

PyCharm is a collection of Python developer tools that are tightly integrated to create a simple environment for Python, web, and data science development.

## **Jupyter Notebook: -**

Jupyter notebooks are used for exploratory data analysis (EDA), data cleansing and transformation, data visualisation, statistical modelling, machine learning, and deep learning, among other data science tasks. Jupyter notebooks are particularly excellent for "presenting the work" done by your data team by combining code, markdown, links, and photos. They're simple to use and may be run cell by cell to see how the code works.

Through the online interface, Jupyter notebooks can be converted to a variety of common output formats (HTML, Powerpoint, LaTeX, PDF, ReStructuredText, Markdown, Python). Data scientists can easily share their findings with others because of this flexibility.

### Streamlit (Frontend): -

Streamlit is a fantastic new tool that allows engineers to quickly create highly dynamic online applications based on their data, machine learning models, or anything else.

The nicest part of Streamlit is that it doesn't require any web programming experience. You're good to go if you know Python.

### NumPy:-

It is used for working with arrays and linear algebra.

### Pandas: -

It is used for data analysis and data pre-processing, CSV file I/O (e.g. *pd.read\_csv*)

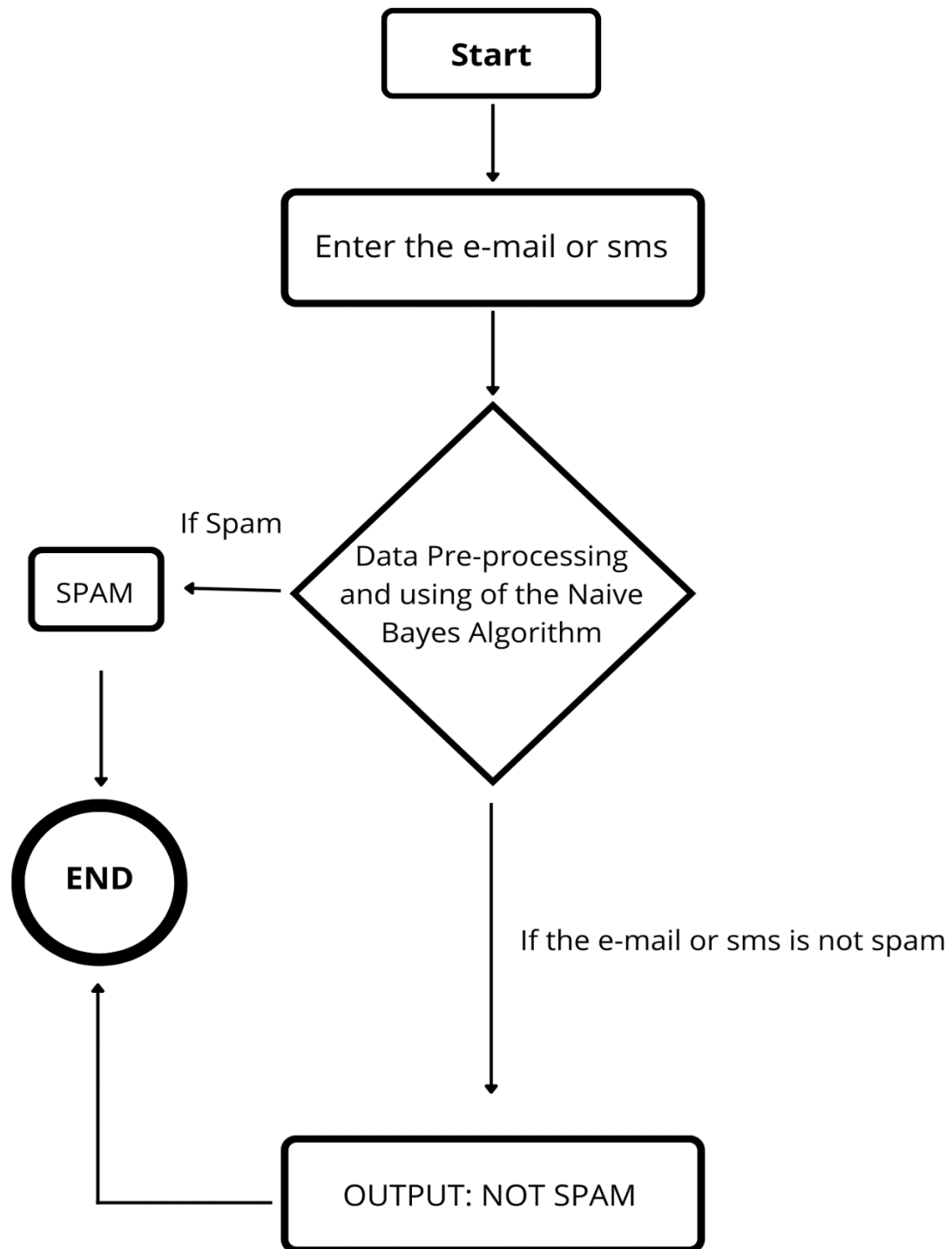
### Sklearn: -

It is used for making use of Machine learning tools. In this project we have used cosine similarity for recommendations.

### NLTK: -

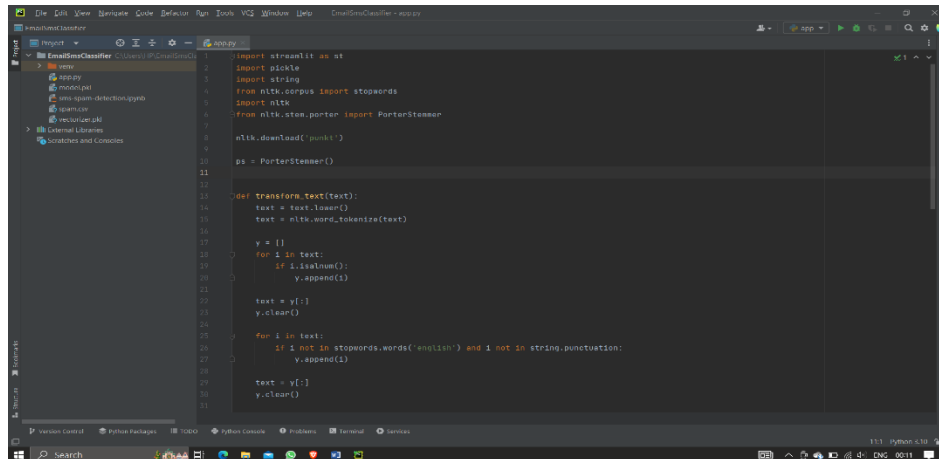
NLTK is a toolkit built for working with NLP in Python. Various word processing libraries are provided with many test datasets. NLTK allows you to perform various tasks such as: B. Tokenization, parse tree visualization, etc.

# ENTITY-RELATIONSHIP DIAGRAM



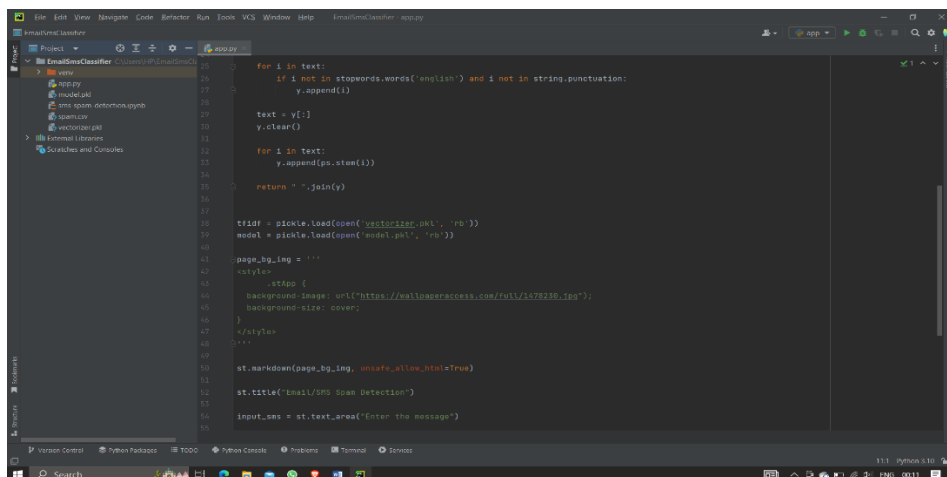
# SNAPSHOTS OF THE DIFFERENT INPUT AND OUTPUT SCREENS

## INPUTS: -



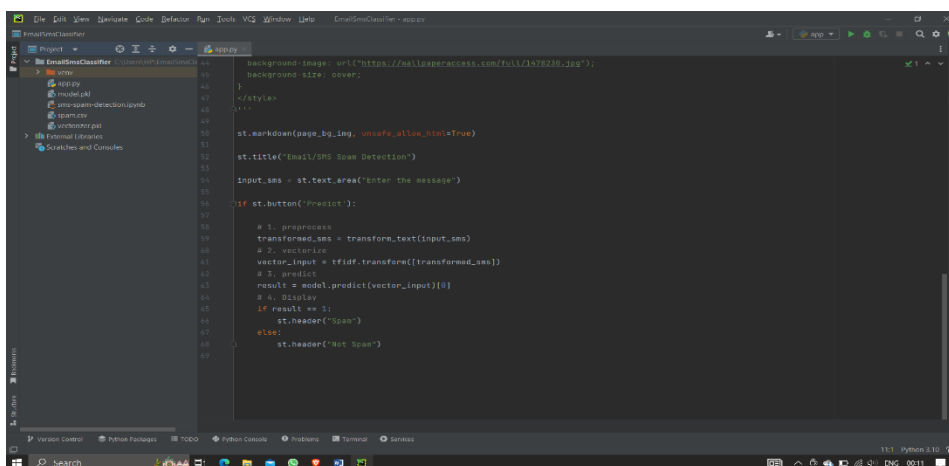
```
1 import streamlit as st
2 import pickle
3 import string
4 from nltk.corpus import stopwords
5 import nltk
6 from nltk.tokenize import PorterStemmer
7
8 nltk.download('punkt')
9
10 ps = PorterStemmer()
11
12
13
14 def transform_text(text):
15     text = text.lower()
16     text = nltk.word_tokenize(text)
17
18     y = []
19     for i in text:
20         if i.isalnum():
21             y.append(i)
22
23     text = y[:]
24     y.clear()
25
26     for i in text:
27         if i not in stopwords.words('english') and i not in string.punctuation:
28             y.append(i)
29     text = y[:]
30     y.clear()
```

Fig (1)



```
25
26
27     for i in text:
28         if i not in stopwords.words('english') and i not in string.punctuation:
29             y.append(i)
30
31     text = y[:]
32     y.clear()
33
34     for i in text:
35         y.append(ps.stem(i))
36
37     return " ".join(y)
38
39 tfidf = pickle.load(open('vectorizer.pkl', 'rb'))
40 model = pickle.load(open('model.pkl', 'rb'))
41
42 page_bg_img = '''
43 <style>
44     background-image: url("https://wallpaperaccess.com/full/1478236.jpg");
45     background-size: cover;
46 </style>
47 </style>
48 </style>
49
50 st.markdown(page_bg_img, unsafe_allow_html=True)
51 st.title('Email/SPM Spam Detection')
52 input_sms = st.text_area('Enter the message')
```

Fig (2)



```
53
54
55 if st.button('Predict'):
56     # 1. preprocess
57     transformed_sms = transform_text(input_sms)
58     # 2. vectorize
59     vector_input = tfidf.transform([transformed_sms])
60     # 3. predict
61     result = model.predict(vector_input)[0]
62     # 4. Display
63     if result == 1:
64         st.header('Spam')
65     else:
66         st.header('Not Spam')
```

Fig (3)

## OUTPUTS: -

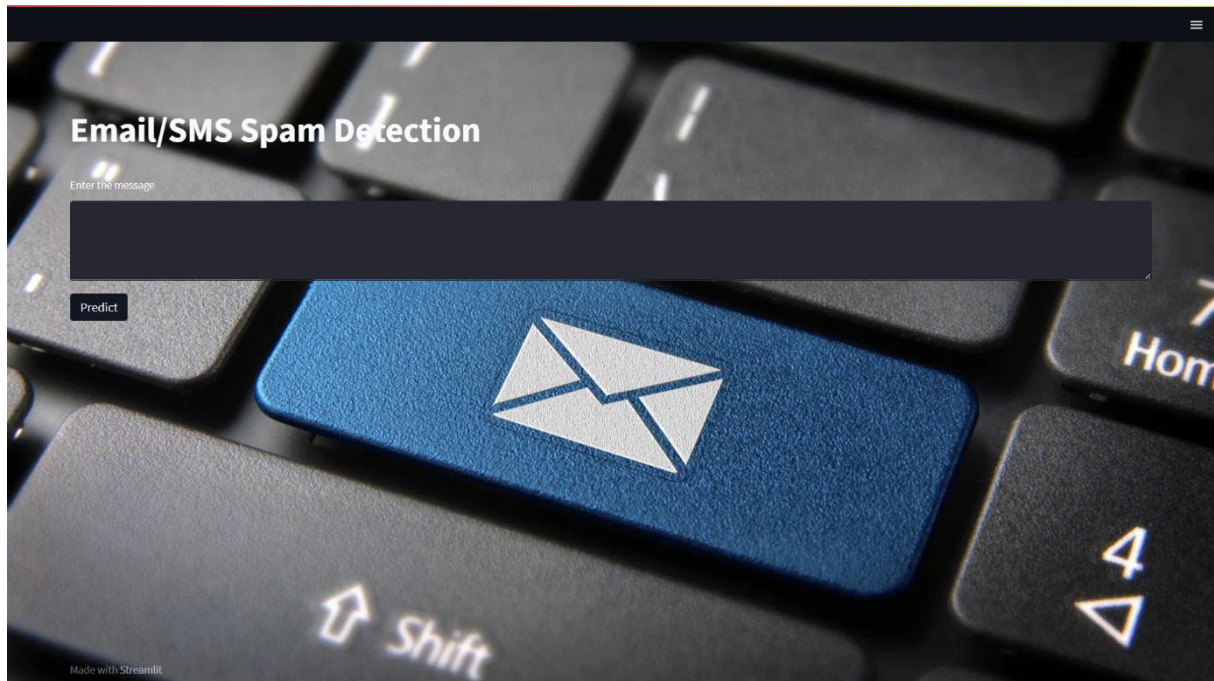


Fig (4): Main page of the application

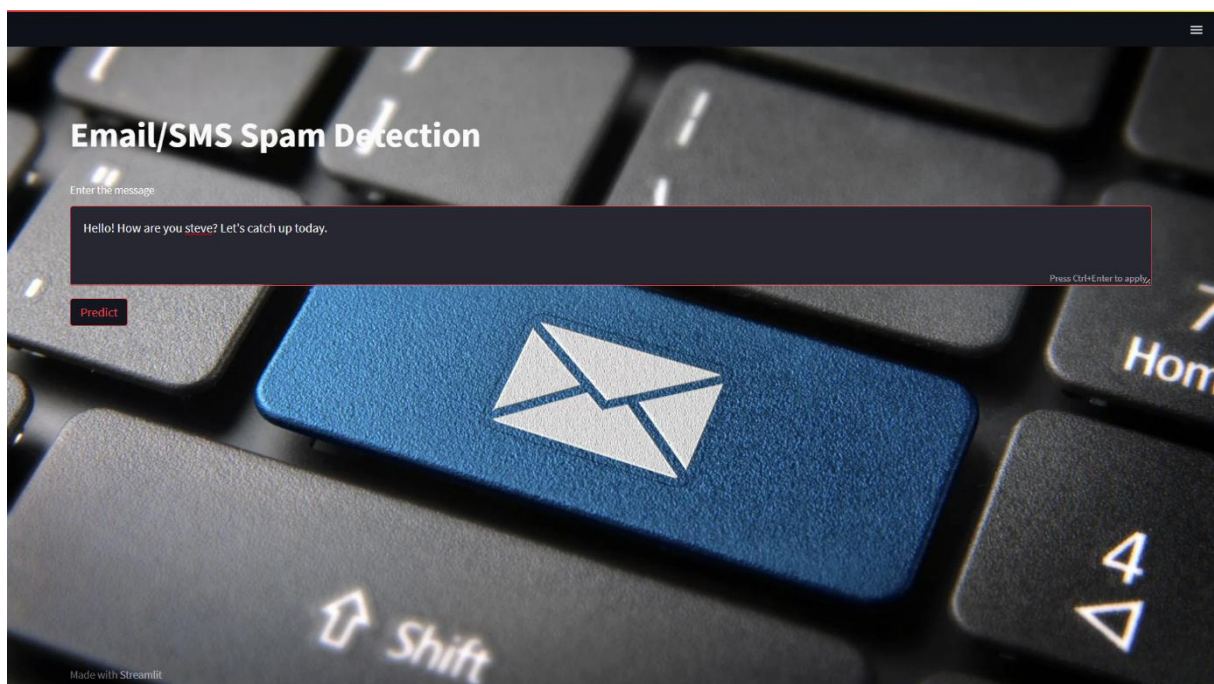


Fig (5): Give the input in the above given box



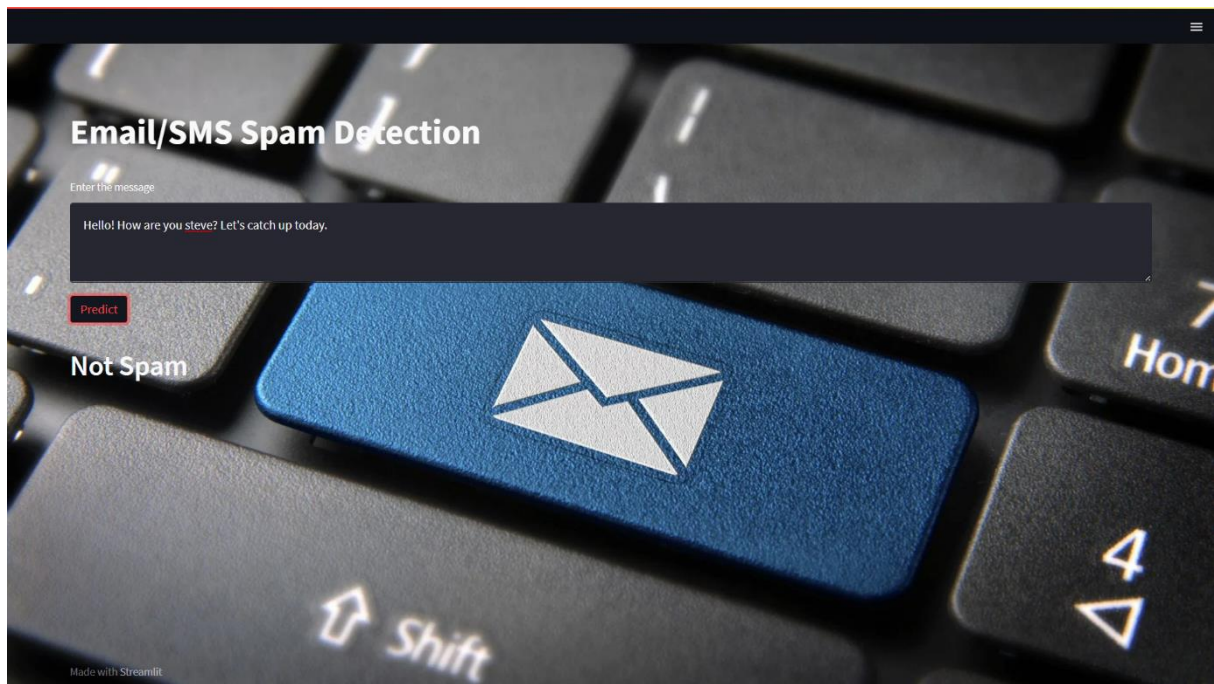


Fig (6): Use the predict button to demonstrate the text is spam or not spam (i.e. Not spam).

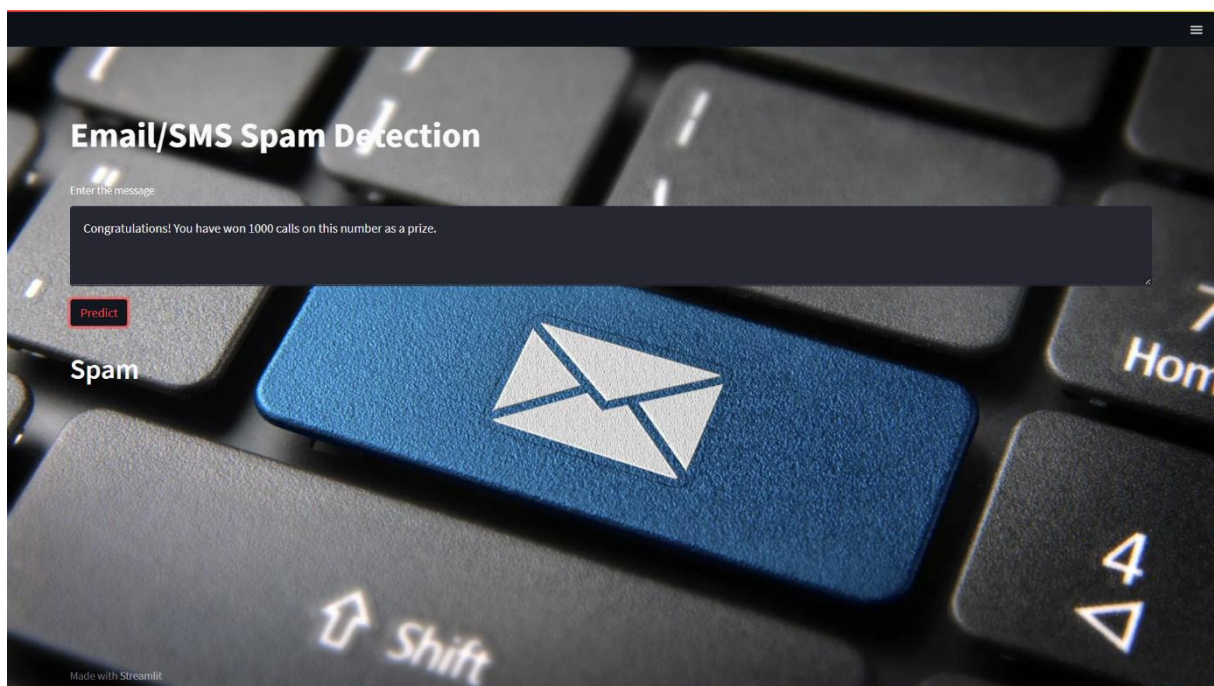


Fig (7): Spam Text

# CONCLUSION

In this project, I have learned how to approach the problem and use data preprocessing and data visualization to draw useful conclusions from your data that help you build better machine learning models.

To solve this classification problem, we used the Naive Bayes algorithm, specifically the Polynomial Naive Bayes algorithm. This is because it has the highest accuracy rating (i.e. the fewest false positives) and used his TFIDF for the vectorization method.

TF-IDF is an information retrieval technique that weights term frequency (TF) and its inverse document frequency (IDF). Each word or term that appears in the text has its own TF and IDF values.

Hyper parameter tuning of max\_features further improved the model.

The following techniques helped me understand how to build a text classification model and create a .pkl file to use over the network. This guide provides an overview of how to classify text messages as spam using different techniques.

You can probably imagine all the incredible possibilities and applications of this new information. This application may include chatbots, HR applications, and other systems. The possibilities are endless.

- Automatic labeling by frequency allows the meaningful creation of labeled datasets.
- Spam detection can be refactored as a regression problem, and an added message spam probability structure provides a more nuanced classification.
- The model successfully detects new spam patterns that were not detected in the training data set.

# **LIMITATIONS**

All machine learning projects have room for improvement and this is no exception. These are a few possible ones in order of expected impact.

## **Within the current framework: -**

1. Improved labeling. that is the most important. Data quality correlates very strongly with prediction quality.

- Use message clustering after feature extraction to group similar messages and manually tag representatives of the group. The label is then shared by all data points within the group.
- Come up with a better labeling heuristic.
- Use external data such as sender, carrier, location to identify spammers and mark all messages as spam.
- Use spam data from other sources (with caution) when useful for training.
- More manual labeling.

2. Experiment with different model architectures and optimize hyper parameters.

## **Outside the current framework: -**

Including information other than the content of the SMS may provide additional variables that help determine the action to take on the SMS after it has passed through the model. User metadata such as time spent in the app, frequency of messages, and number of previously blocked messages are great additions to the spam probability calculated from SMS content alone.



An integrated framework in which Spam probability is just one component will enhance the model's usefulness by reducing the number of false positives (from only blocking after a certain amount of warnings), not evaluating messages from known legitimate users and promptly block repeated spammers from using the app.

# **BIBLIOGRAPHY**

- <https://docs.streamlit.io/library/get-started>
- <https://www.nltk.org/book/>
- <https://stackoverflow.com/>
- <https://www.quora.com/What-are-the-advantages-of-using-a-naive-Bayes-for-classification>
- <https://towardsdatascience.com/document-feature-extraction-and-classification-53f0e813d2d3>