

## SPAM CLASSIFIER

19RH1A05E0,19RH1A05E3,19RH1A05E9  
II CSE C



### Introduction

In recent times, unwanted commercial bulk emails called spam has become a huge problem on the internet. The person sending the spam messages is referred to as the spammer. Such a person gathers email addresses from different websites, chatrooms, and viruses. Spam prevents the user from making full and good use of time, storage capacity and network bandwidth. The huge volume of spam mails flowing through the computer networks have destructive effects on the memory space of email servers, communication bandwidth, CPU power and user time . The menace of spam email is on

---

---

the increase on a yearly basis and is responsible for over 77% of the whole global email traffic. Users who receive spam emails that they did not request find it very irritating. It is also resulted to untold financial loss to many users who have fallen victim of internet scams and other fraudulent practices of spammers who send emails pretending to be from reputable companies with the intention to persuade individuals to disclose sensitive personal information like passwords, Bank Verification Number (BVN) and credit card numbers.

- According to stats we receive 40% of spam mails daily where we exclude social and promotional mails.

- In recent times, unwanted commercial bulk emails called spam has become a huge problem on the internet.

## **Problem**

Email spam is also termed as junk email, these are suspicious messages sent in bulk through emails. Most of the email spam messages are commercial in nature. They contain links that look genuine and convincingly familiar however the links lead to phishing websites that host malware. Spam emails can be annoying for users, but they bring more issues and risks with them.

## **Solution:**

---

Email spam detection in a logical, theoretically grounded manner, in order to facilitate the introduction of spam filtering techniques that could be operational in an efficient way. To effectively handle the threat posed by email spams, leading email providers have employed the combination of different machine learning (ML) techniques. All these tasks are done through Natural Language Processing (NLP), which *processes text into useful insights* that can be applied to future data. In the field of artificial intelligence, NLP is one of the most complex areas of research due to the fact that text data is contextual. It needs modification to make it machine-interpretable and requires multiple stages of processing for feature extraction.

## Model Overview:

Let's start with our spam detection data. We'll be using the open-source [Spambase dataset](#) from the UCI machine learning repository, a dataset that contains 5569 emails, of which 745 are spam. The target variable for this dataset is 'spam' in which a *spam email is mapped to 1* and anything else is mapped to 0. The target variable can be thought of as what you are trying to predict. In machine learning problems, the value of this variable will be modeled and predicted by other variables.

Data usually comes from a variety of sources and often in different formats. For this reason, transforming your raw data is essential. However, this transformation is not a simple process, as text data often contain redundant and repetitive words. This means

---

that processing the text data is the first step in our solution. The fundamental steps involved in text preprocessing are Cleaning the raw data Tokenizing the cleaned data.

Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Many machine learning practitioners believe that properly optimized feature extraction is the key to effective model construction.

In machine learning, scoring is the process of applying an algorithmic model built from a historical dataset to a new dataset in order to uncover practical insights that will help solve a business problem. Text processing is the automated process of analyzing text data for getting structured information.

Text generation is a subfield of natural language processing. It leverages knowledge in computational linguistics and artificial intelligence to automatically generate natural language texts, which can satisfy certain communicative requirements.

Model selection is the process of selecting one final machine learning model from among a collection of candidate machine learning models for a training dataset. Text data can be easily interpreted by humans. But for machines, reading and analyzing is a very complex task. To accomplish this task, we need to convert our text into a

---

machine-understandable format. Embedding is the process of converting formatted text data into numerical values/vectors which a machine can interpret.

Different performance metrics are used to evaluate different Machine Learning Algorithms. The key classification metrics: Accuracy, Recall, Precision, and F1- Score.

metrics used to evaluate a classification model are accuracy, precision, and recall.

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

### **Factors for classifying mails:**

We have many factors which effect on classifying our mails namely some of them are:

**Mail sender-** If their mail id is repeated many times under some promotions or ads we can simply classify them into spam.

**Content of mail-** By using text processing we can analyze all types of keywords and sentimental analysis is useful here.

**Subject of mail-** When we see some promotional or adds subject for example 70% off for this festive season so simple with help of subject we can classify them into spam.

---

**Received time** of mail is also an important factor because during any festive or sales period most of the advertisement mails are received.

### **Technology used:**

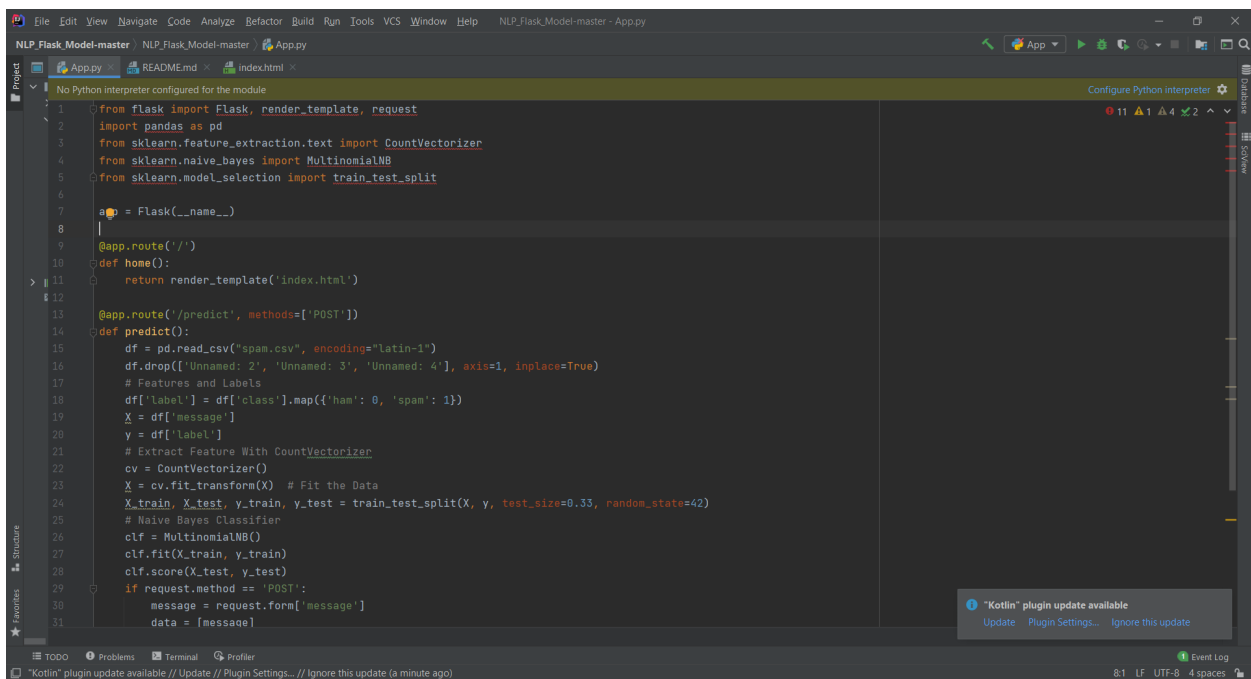
**Natural Language processing(NLP)** :Natural language processing involves the reading and understanding of spoken or written language through the medium of a computer. Through natural language processing, computers learn to accurately manage and apply overall linguistic meaning to text excerpts like phrases or sentences.

**Supervised ML**:supervised machine learning techniques namely Decision tree classifier, Multilayer Perceptron, Naïve Bayes Classifier are used for learning the features of spam emails and the model is built by training with known spam emails and legitimate emails. The results of the models are discussed.

**Python**: Python is used as backend development for our model. We also use flask as the front end to connect our local host server and backend part.

## CODE:

We developed our code using NLP as algorithm, ML as technology, and python as language and we developed our project using IntelliJ as IDE. By using stop words and some promotional keywords like 50% off and you won \$10M we improve our model and we developed a flask application with collaboration using HTML for user interface.



```
1 from flask import Flask, render_template, request
2 import pandas as pd
3 from sklearn.feature_extraction.text import CountVectorizer
4 from sklearn.naive_bayes import MultinomialNB
5 from sklearn.model_selection import train_test_split
6
7 app = Flask(__name__)
8
9 @app.route('/')
10 def home():
11     return render_template('index.html')
12
13 @app.route('/predict', methods=['POST'])
14 def predict():
15     df = pd.read_csv('spam.csv', encoding='latin-1')
16     df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)
17     # Features and Labels
18     df['label'] = df['class'].map({'ham': 0, 'spam': 1})
19     X = df['message']
20     y = df['label']
21     # Extract Feature With CountVectorizer
22     cv = CountVectorizer()
23     X = cv.fit_transform(X) # Fit the Data
24     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
25     # Naive Bayes Classifier
26     clf = MultinomialNB()
27     clf.fit(X_train, y_train)
28     clf.score(X_test, y_test)
29     if request.method == 'POST':
30         message = request.form['message']
31         data = [message]
```

The screenshot shows the IntelliJ IDE interface with a Python Flask application. The code implements a spam classification model using NLP (CountVectorizer and MultinomialNB) and ML (train\_test\_split). The application has two routes: a home page and a prediction endpoint. The prediction endpoint reads a CSV file, preprocesses the data, and uses a Naive Bayes classifier to predict the label of incoming messages. A notification at the bottom indicates a Kotlin plugin update is available.

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help NLP_Flask_Model-master - index.html
NLP_Flask_Model-master NLP_Flask_Model-master / templates / index.html
App.py README.md index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Spam Detection System</title>
6 <link href="https://fonts.googleapis.com/css?family=Pacifico" rel="stylesheet" type="text/css">
7 <link href="https://fonts.googleapis.com/css?family=Arimo" rel="stylesheet" type="text/css">
8 <link href="https://fonts.googleapis.com/css?family=Hind:300" rel="stylesheet" type="text/css">
9 <link href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300" rel="stylesheet" type="text/css">
10 <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
11 </head>
12
13 <body>
14 <div class="login">
15 <h1>Spam Detector for Emails</h1>
16
17 <form action="{{ url_for('predict') }}" method="POST">
18 <textarea name="message" rows="6" cols="50" required="required"></textarea>
19 <br> </br>
20 <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
21
22 <div class="results">
23
24 {%- if prediction == 1%}
25 <h2 style="color: red;">Looking Spam! , Be safe</h2>
26 {%- elif prediction == 0%}
27 <h2 style="color: green;">Not a Spam!</h2>
28 {%- endif %}
29
30 </div>
31
32 </div>
33
34 </body>
35 </html>
```

Kotlin" plugin update available  
Update Plugin Settings... Ignore this update

1000 Problems Terminal Profiler  
"Kotlin" plugin update available // Update // Plugin Settings... // Ignore this update (a minute ago)

Type here to search

25:28 LF UTF-8 Tab  
12:07  
07-01-2021

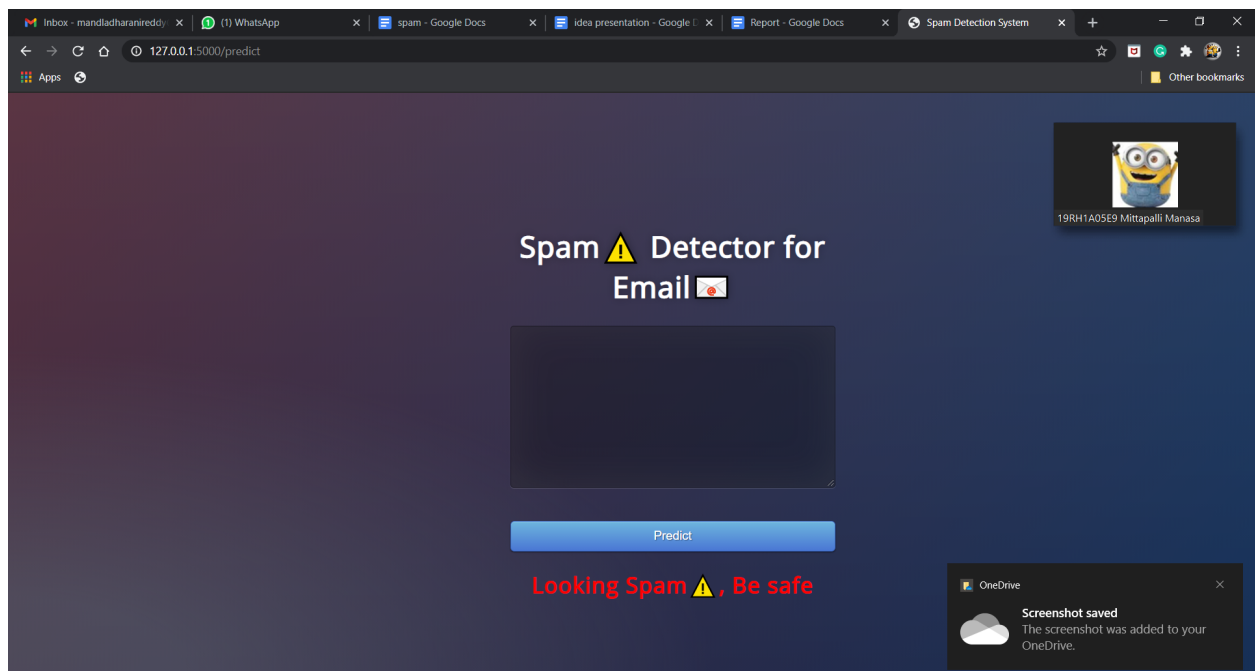
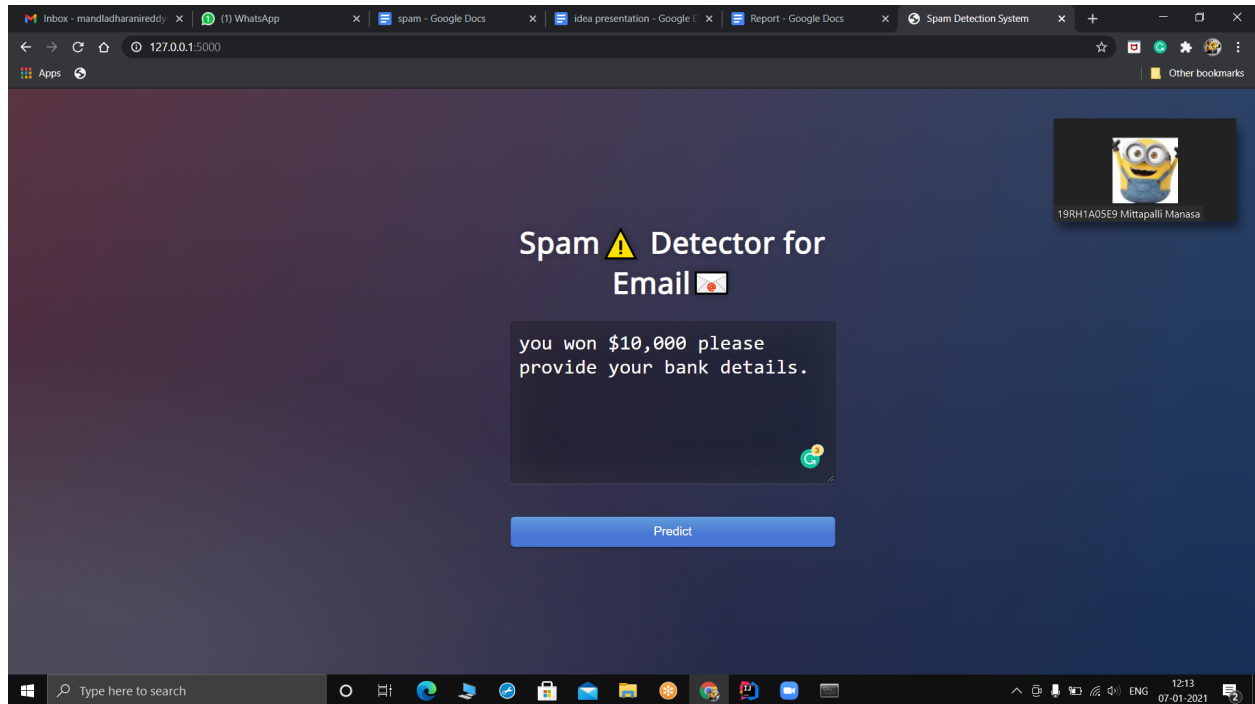
```
Anaconda Prompt (anaconda3) - python App.py run
(base) C:\Users\dhara>D:
(base) D:\>cd D:\NLP_Flask_Model-master
(base) D:\NLP_Flask_Model-master>python App.py run
* Serving Flask app "App" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

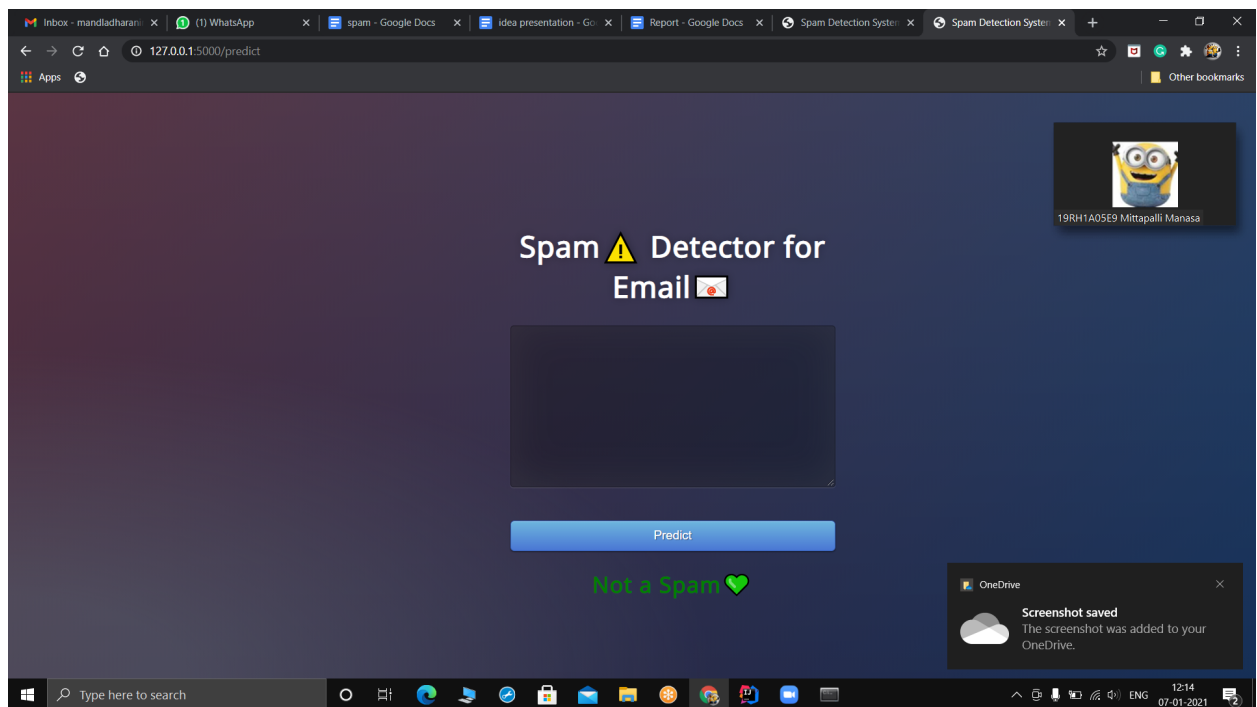
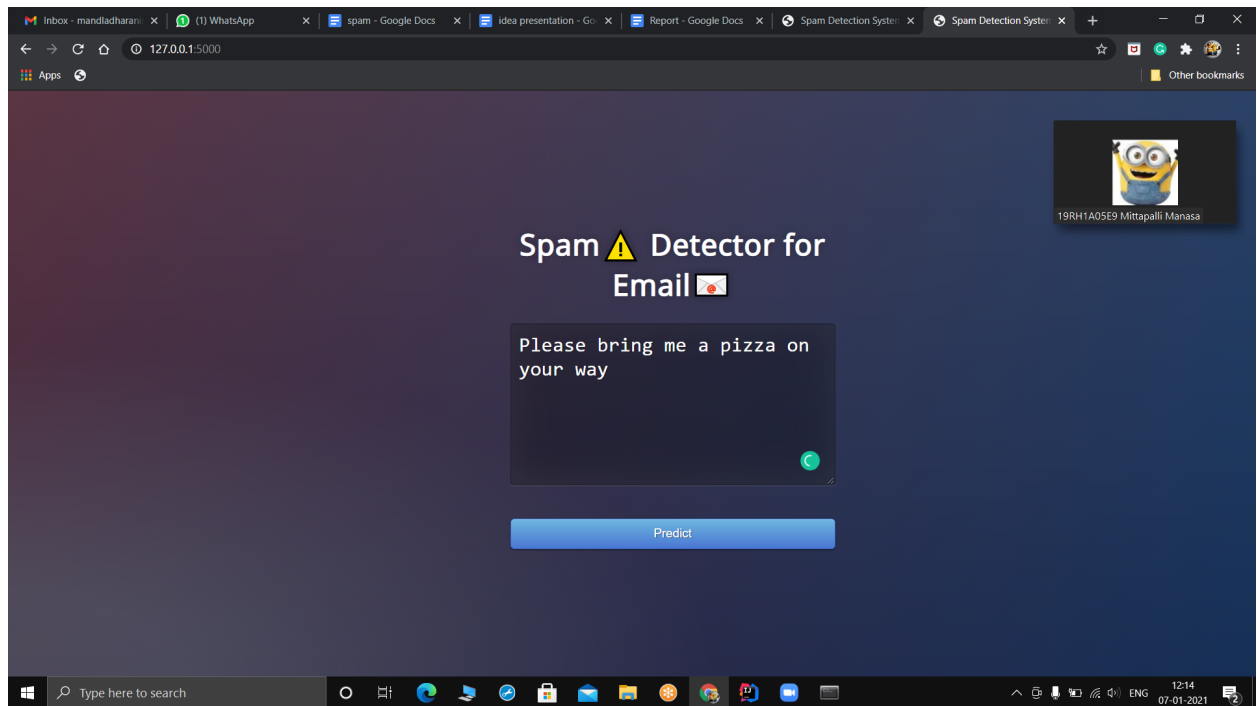
Type here to search

12:10  
07-01-2021



## OUTPUT SCREEN:





---

## **Future works:**

We try to improve our model accuracy by improving feature extraction and then we try to detect spam emails in real time. By comparing our metrics, accuracy, f-factor we improve our algorithm efficiency and we try to build an efficient spam filter.

## **Conclusion:**

We create a spam detection model by converting text data into vectors, creating a BiLSTM model, and fitting the model with the vectors. We also explored a variety of text processing techniques, text sequencing techniques, and deep learning models, namely RNN, LSTM, BiLSTM. Precision and recall are the two most widely used performance metrics for a classification problem to get a better understanding of the problem.

Precision is the fraction of the relevant instances from all the retrieved instances. When applying a model like this to real-world data, we still need to actively monitor the model's performance over time. We can also continue to improve the model by responding to results and feedback by doing things like adding features and removing misspelled words. The concepts and techniques learned in this article can be applied to a variety of natural language processing problems like building chatbots, text summarization, language translation models.