CSCE 5290: Natural Language Processing

Final Project

Group-2

Naveed Ahmed Mohammed - 11548614
Iliyas Ahmed Mohammed - 11550253
Mohammed Abdul Qayyoom Shaik - 1532186
Eswara Reddy Thimmapuram - 11506566

Introduction

Project Title:

Spam Message Classification

Goals and Objectives:

- **Motivation:**

  Nearly everyone now-a-days has a smart phone that at the very least has the basic feature such as messaging. Spam messages are unsolicited text messages that are forwarded to wide group of users that are typically sent for the aim of promoting products and services without their prior permission. The goal of this study is to identify whether a given message is spam or ham from the given message by using NLP techniques and machine learning algorithms.

- **Significance:**

  These days, the number of spam messages among scams has surged dramatically. These spam messages generally lure users to provide confidential and payment information by offering fraudulent and appealing deals. Using this spam detection model, we can easily identify spam messages i.e., fake messages that users are unable to recognize.

- **Objectives:**

Here, we have collected more than 1000 messages that are identified as ham & spam filled. Then, we will clean the data by eliminating punctuations and stop words using NLTK library.

We will use lemmatization i.e., normalization technique on created tokens to retrieve root words. In addition, we will apply count vectorization to eliminate words that are infrequent in the data. Furthermore, to predict the text messages we will use “Text Summarization” method to the given input sequence and then apply naïve Bayer classifier and SVM classifier from python “sklearn” package.

- **Features:**

The key feature of this project is to identify messages as either spam or ham. At first, the less common terms will be eliminated from the data once pre-processing and normalization techniques have been used. In order to determine if a message is spam or ham, we must train the model using the cleaned data and apply ML algorithms. As an example, the model will eliminate all the stop words and unnecessary text when we provide it with a message. Finally, after applying lemmatization to the message, the chosen algorithm will determine if the message is spam-or-ham filled.

Additional features:

We are adding extra feature i.e., Text Summarization to this Spam Message Classification model.

Functionality: For the input words sequence we are creating a frequency table and then tokenizing each sentence. Using the frequency method, we are finding score for each sentence and from that scores we are considering the average score of the sentences and finally generating the summary.

Background (Related Work):

The word ‘spam’ is defined as undesired text that is sent or received over social media platforms and messages. It is produced by spammers to divert consumers' attention from social media marketing and malware distribution, among other purposes. Spam is also present in product reviews that are posted on social networking websites. Liu & Pang (2018) estimate that between 30 and 35 percent of internet reviews are spam[1]. In a work published in 2018, "SMS Spam Filtering Using Supervised Machine Learning Algorithms," the techniques for categorizing spam messages are described. They used supervised machine learning algorithms like SVM and max entropy and performance has been accessed[2]. The classification of spam using SVM algorithm is summarized in the ‘Email Spam Classification by SVM’. In this paper, the performance of several kernel types has been assessed. The scope of this project is limited to one classification algorithm[3].

In the paper ‘A Machine Learning based Spam Detection Mechanism’ published in 2020, email spam detection was performed using Naïve Bayes algorithm including preprocessing, URL checking, tokenization and keyword checking. This paper is limited to one classification algorithm[4].

In "Email Spam Detection Using Mail Learning Techniques," which was published in 2020, different machine learning algorithms, including Naive Bayes, Support Vector Machine, Decision Tree, Neighborhood Neighbor, and Random Forest Classification, are assessed for paper spam emails. This study found that the Nave Bayes method worked well. The absence of testing the application on various data sets is a drawback of this study[5].

In "Content-Based Spam Detection in Email using Bayesian Classifier", published in 2015, the classification process is broken down into four parts in the paper: pre-processing, feature extraction, training, and classification. Its performance has been assessed, and it also explains how emails are categorized according to their content. Only one classification algorithm was covered in this study [6].

An Artificial Immune System (AIS) was created by Lutfun and Mainul[[7] for the classification of SMS. As an input spam filter, the system made advantage of a number of features. With the aid of a trained dataset that contained spam terms, phone numbers, etc., it was then utilized to categorize the text messages. The findings of this experiment demonstrated that when classifying messages as spam or not-spam, the Naive Bayesian algorithm performed superior in terms of accuracy and convergence speed. A two level stacked classifier was created by Narayan et al.[8] to distinguish between spam and legal SMS.

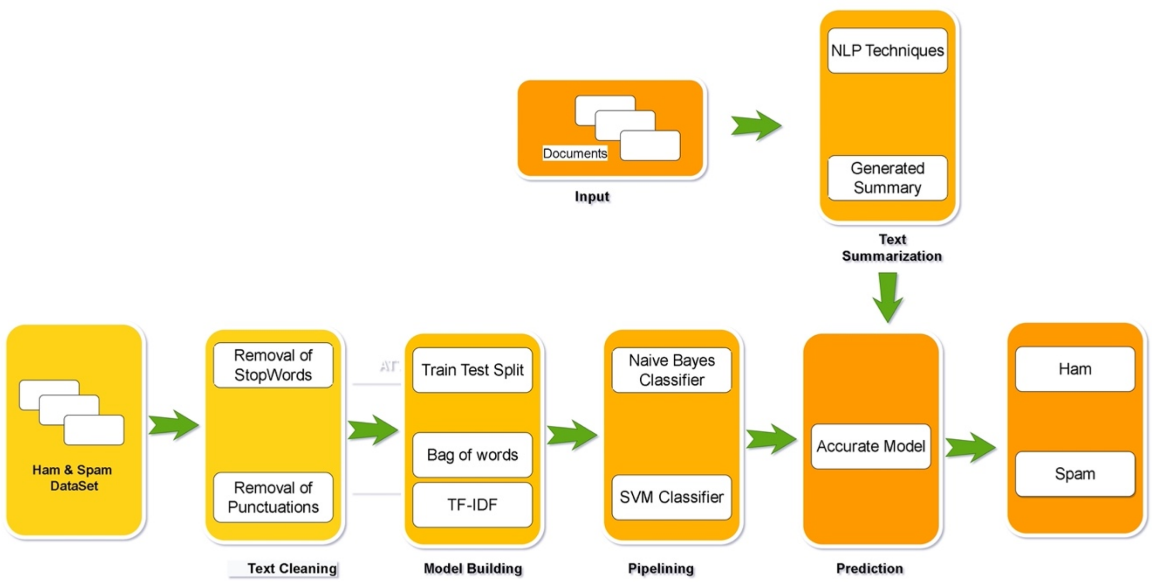
A selection of words whose individual probabilities are higher than a threshold are recorded in the classifier's first level. The picked words from the first level of classifier are inputted once that second level is called. They used various pairings of two-level machine learning classification algorithms, including Bayesian and SVM. Gomez et al.

[[9] examined how well Bayesian filtering techniques, which are used to prevent email spam, can be used to identify and thwart mobile spam. They preprocessed the communications using various tokenization techniques, picked out features, and evaluated their performance using several machine learning algorithms. They showed that, with the right feature extraction, Bayesian filtering approaches may be successfully used to SMS spam. In ‘Content-based SMS Spam Messages classification using Natural Language Processing and Machine Learning’ published in 2021, Sumahasan[[10] contrasted Naive Bayes with Support Vector methods using vector machines to classify SMS spam. Both models have been developed, trained, and evaluated using widely available standard datasets. The simulation's empirical findings demonstrated that the Nave Bayes-based proposed scheme outperformed the Support Vector Machine in terms of accuracy and processing speed.

For this project, we have collected some of the messages which are spam and we have understood why these messages are called spam, like they include some fraudulent offering and gifts to target the users. We have also studied about different algorithm like support vector machine(svm), random forest, naive bayer classifier and decision tree to classify the text, we have chosen the best and efficient algorithm which will help for this project.

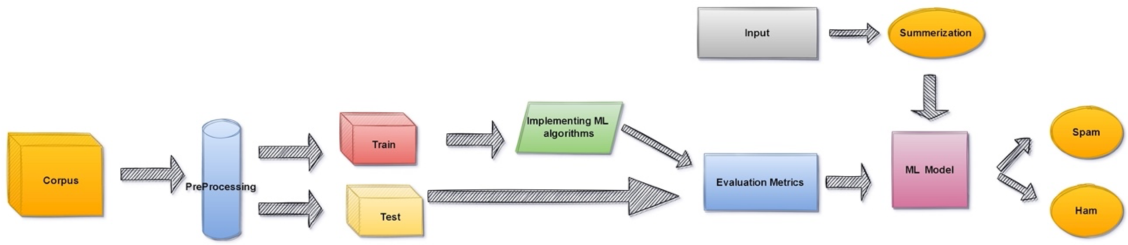
Model:

Architecture Diagram:



In the above image, you can see that data is cleaned and then we have removed stopwords and lemmatized the words and then build the model and then split into train and test sets and then summarized the input text to the model to give summarized text to the model to predict the results.

Workflow Diagram:



Workflow diagram explanation:

In this, we have taken the raw data and applied pre-processing techniques to clean the data by removing stopwords and punctuations. Then we have divided the dataset into test and train set. Then we have applied the ML algorithms to train the model with train set. Then we have evaluated the model with metrics and chosen the algo which has more accuracy. After the model is selected, we are applying nltk techniques to summarize the text. Finally, we have given the summarized message as input to the model to predict and the model classifies as ham or spam.

- **<u>Dataset:</u>**

Dataset which we have chosen has 2 columns namely, category and message. Basically, Category has two values “ham” and “spam,” which is given for text given in message column.

This dataset contains 5573 rows filled with text and its category, which will help to better train the model.

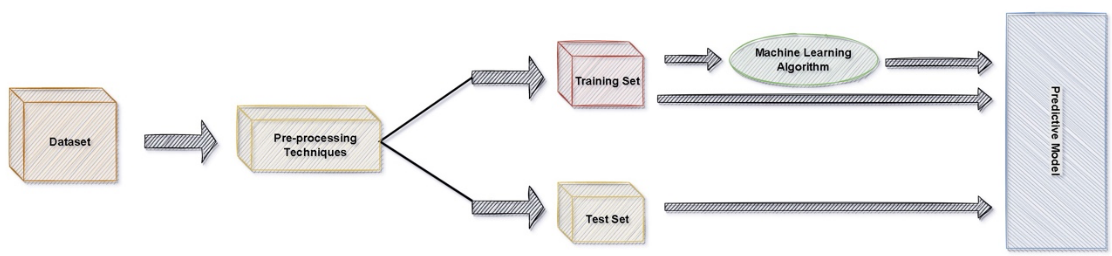
Train set:

	A	B
1	Category	Message
2	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
3	ham	Ok lar... Joking wif u oni...
4	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
5	ham	U dun say so early hor... U c already then say...
6	ham	Nah I don't think he goes to usf, he lives around here though
7	spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, ¬£1.50 to rcv
8	ham	Even my brother is not like to speak with me. They treat me like aids patent.
9	ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
10	spam	WINNER!! As a valued network customer you have been selected to receivea ¬£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
11	spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030
12	ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.
13	spam	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info
14	spam	URGENT! You have won a 1 week FREE membership in our ¬£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
15	ham	I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.
16	ham	I HAVE A DATE ON SUNDAY WITH WILL!!
17	spam	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL
18	ham	Oh k...i'm watching here:)
19	ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.
20	ham	Fine if that¬ís the way u feel. That¬ís the way its gota b
21	spam	England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/√∫1.20 POBOXox36504W45WQ 16+
22	ham	Is that seriously how you spell his name?

Test set:

	A
1	**Message**
2	No, I was trying it all weekend ;V
3	You know, wot people wear. T shirts, jumpers, hat, belt, is all we know. We r at Cribbs
4	Cool, what time you think you can get here?
5	Wen did you get so spiritual and deep. That's great
6	Have a safe trip to Nigeria. Wish you happiness and very soon company to share moments with
7	Hahaha..use your brain dear
8	Well keep in mind I've only got enough gas for one more round trip barring a sudden influx of cash
9	Yeh. Indians was nice. Tho it did kane me off a bit he he. We shud go out 4 a drink sometime soon. Mite hav 2 go 2 da works 4 a laugh soon. Love Pete x x
10	Yes i have. So that's why u texted. Pshew...missing you so much
11	No. I meant the calculation is the same. That <#> units at <#> . This school is really expensive. Have you started practicing your accent. Because its important. And have you decided if you are doing
12	Sorry, I'll call later
13	if you aren't here in the next <#> hours imma flip my shit
14	Anything lor. Juz both of us lor.
15	Get me out of this dump heap. My mom decided to come to lowes. BORING.
16	Ok lor... Sony ericsson salesman... I ask shuhui then she say quite gd 2 use so i considering...
17	Ard 6 like dat lor.
18	Why don't you wait 'til at least wednesday to see if you get your .
19	Huh y lei...
20	REMINDER FROM O2: To get 2.50 pounds free call credit and details of great offers pls reply 2 this text with your valid name, house no and postcode
21	This is the 2nd time we have tried 2 contact u. U have won the ¬£750 Pound prize. 2 claim is easy, call 087187272008 NOW1! Only 10p per minute. BT-national-rate.
22	Will √º b going to esplanade fr home?
23	Pity, * was in mood for that. So...any other suggestions?
24	The guy did some bitching but I acted like i'd be interested in buying something else next week and he gave it to us for free

Detail design of Features:



This model will help to predict the message either “ham” or “spam” and to summarize the given message using the frequency table by taking the average scores of the text and then providing the summary of the message as text.

We have taken large dataset with many messages categorize into ham and spam which will help model to train better. We have taken some test set with messages which are not classified, to give input to this model and get the result in “ham” or “spam.”

Here we are breaking-down the long texts into manageable sentences and paragraphs i.e., the summarization process. By using this technique, the text's meaning is maintained while important information is also extracted.

Here, we use an extractive method that chooses the top N sentences accurately and then summaries’ all the main idea on the article. These main ideas from the article are then rewritten in new words in extractive summaries.

Finally, we will summarize the given input message and give as output with message category.

Analysis of data:

After taking dataset, we can see that dataset contains noise which are with no meaning to the message. we have to remove noise with pre-processing techniques like Data cleaning which involves removal of NULL values, punctuations and stop words on dataset. Before data cleaning, for analysis we are storing the punctuation and length of the message in data frame with extra column, now we are splitting the column into ham and spam and calculating the mean of punctuations of spam and ham messages and mean of the length of both ham and spam. Now, we can see that the spam message has more punctuations than ham and length is also more for spam than ham.

We are using nltk stopwords and lemmatization from the NLTK library for cleaning, i.e., regex to remove stopwords and the stopwords which are not there in nltk package that would be lemmatize using lemmatization and after cleaning the resultant words is joined to form message and is stored in data frame. The cleaned message is now used for model to train which will be more efficient.

Using TF-IDF vectorizer, we will transform the input text into meaningful representations of integers which helps the machine learning classifiers to fit into the model for better predictions. It helps in comparing the number of times a word appears in a paper to the number of documents that appears in and finally determine how much the word is original.

Once the dataset is pre-processed, we divide into test and train set which will be used in developing the model with the help of ML algorithms to achieve the goal.

Implementation:

Pseudo code for the implementation of naïve bayes algorithm in Spam classification:

Input:

Training dataset Spam_ham_dataset,

Testing dataset {t1,t2,t3,….tn} predicted tokens

Output:

Input text with test class

Steps:

Read the train set of dataset Spam_ham_dataset,

Calculating the term frequencies from the each class using Tf-idf vectorizer.

Predicting on the test class and giving the accuracy of each class.

Pseudo code for the implementation of SVM algorithm in Spam classification:

Input:

Training dataset Spam_ham_dataset,

Testing dataset {t1,t2,t3,….tn} predicted tokens

Output:

Input text with test class

Steps:

Read the train set of dataset Spam_ham_dataset,

Calculating the term frequencies from the each class using Tf-idf vectorizer.

Predicting on the test class and giving the accuracy of each class.

After performing data cleaning, the dataset is ready to train the model. Now, we have split the dataset into message and labels in which messages are independent values and labels are dependent values. Currently, we have divided data into test and train using sk-learn library by giving percentage to both train and test.

Moreover, we are transforming text into understandable way to machine, for this we are using TF-IDF vectorizer. It is a common algorithm that converts text into a machine understandable number which calculate the weight of words of the overall document and normalize the data and using this data we fit in to the machine learning algorithm for the predictions.

For the implementation of TF-IDF vectorizer, Firstly we have imported count vectorizer to transform the text data into vectors on the base of frequent occurred words which can be denoted as bag of words and then we have imported TF-IDF vectorizer from the sklearn package and apply on to the data using fit_tranform, this will transform data into matrix. The matrix formed is of sentences and words.

Furthermore, we are going to use pipeline concept and import it from sklearn library. we use pipelining because it is repetitively go through the exact same steps for each test set in order to obtain predictions This pipeline will be helpful on test data to perform all the previous steps like vectorization of the data and classifying the data in a single cell. After that we are going to implement on various classifiers like SVM, naïve bayes, etc., From that we are going to choose the classifier with more accuracy.

Here, we will be importing the MultinomialNB model in order to fit it with X-train & y_train. Thus, the Pipeline performs vectorizing and make the predictions automatically. The accuracy we obtained using multinomial is 96.9%.

Here, we have the accuracy, but we cannot show that the model is working effectively. Thus, to know the model better we are performing evaluation metrics to calculate classification_report and the confusion_matrix and storing the reports that obtained.

Next, we had implemented SVM classifier to compare with the model which we build earlier. Here, we did the same steps as in multinomial i.e., we build a pipeline between vectorizer and the SVM classifier in order to fit in x-train and y-train and training the model.

Once the classifier model is implemented, then we can make predictions by providing various test cases and finally based on the output efficiency we can detect a good algorithm for this model. The accuracy we obtained using SVM is 98.6%.

We can conclude SVM is better model than the naïve bayes.

The next step is to integrate the spam detection with summarization. Firstly, we created a method for extracting the important features from the input document then collect all the keywords from the extracted features in a labelled structure. Next, we are building an analyzer to detect the negative labelled features from the input document to improve accuracy. Here, we have used nltk techniques to summarize the given input text and the generated summary is given to the spam classification model and the model will execute and predict the results into either ham or spam.

Results:

Visualizing the dataset in below image1. Then after we have added the punctuation length of each message in next column as shown in image2. We have cleaned the data and performed data cleaning, which involves in removal unnecessary data which is not meaningful to the message. Below is the screenshot of data cleaning, in which we have removed the null values, stopwords and punctuation as shown in below image.

Image-1:

```
messages.head()
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Image-2:

```
from nltk.corpus import stopwords
#Removal of extra characters and stop words and lemmatization
corpus = []
corpus_exp = []

#Skipping the 0th index (it's of Category)
for i in range(0,len(messages)):

    words = re.sub('[^a-zA-Z]',' ',messages['Message'][i])

    words = words.lower()
    #Splits into list of words
    words = words.split()


    #Lemmatizing the word and removing the stopwords
    words = [lemmatizer.lemmatize(word) for word in words if word not in set(stopwords.words('english'))]
    #print(words[0])
    #Again join words to form sentences
    words = ' '.join(words)

    corpus.append(words)
```

```
#What's in Corpus
corpus[0]
```

```
'go jurong point crazy available bugis n great world la e buffet cine got amore wat'
```

In the above image we can see that the we have applied nltk stopwords and removed stopwords and we have lemmatized the word.

Once the labelling is done, we are splitting the data into train and test set. So that further we can build the model classifier.

```
[ ] X_train , X_test , y_train , y_test = train_test_split(X , y, test_size = 0.33, random_state = 42)
```

```
X_train.head()
```

```
3235                                           yup comin
945     sent score sophas secondary application school...
5319                              kothi print marandratha
5528                             effect irritation ignore
247                                          asked call ok
Name: Message, dtype: object
```

We have used TF-IDF vectorizer which transform the input text into the understandable representation of integers so that the machine learning classifier fit into the model for better predictions.

We can see in the below image.

```
X_train_tfidf_vect

array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

```
[ ] X_train_tfidf_vect.shape

(3733, 5772)
```

Here, we are building a model by using naïve bayes classifier.

Naive Bayer Classifier

```
[ ] from sklearn.naive_bayes import MultinomialNB
```

Below is the prediction of the test data with testing score and the score we achieved is 96.9%

```
[ ] #Predictions of the test data
    y_preds_mnb

    array(['ham', 'ham', 'ham', ..., 'ham', 'ham', 'ham'], dtype='<U4')

    #Training score
    text_mnb.score(X_train,y_train)

    0.975354942405572

[ ] #Testing score
    text_mnb.score(X_test,y_test)

    0.9690048939641109
```

We are building confusion matrix and final report based on the prediction test sets for multinomial naïve bayes.

```
[ ] print(confusion_matrix(y_test,y_preds_mnb))

    [[1592    1]
     [  56  190]]

[ ] from sklearn.metrics import classification_report

    print(classification_report(y_test,y_preds_mnb))

                  precision    recall  f1-score   support

             ham       0.97      1.00      0.98      1593
            spam       0.99      0.77      0.87       246

        accuracy                           0.97      1839
       macro avg       0.98      0.89      0.93      1839
    weighted avg       0.97      0.97      0.97      1839
```

In the above image we can see the generated classification report and the confusion matrix report.

SVM classifier implementation:

In the below image, we can see that we have imported svm classifier amd we have fit the model into train and test sets.

SVM Classifier

```
[ ] from sklearn.svm import LinearSVC
```

```
#each tuple takes the name you decide , next you call what you want to occur
text_svm=Pipeline([('tfidf',TfidfVectorizer()),('svm',LinearSVC())])
```

```
#Now u can directly pass the X_train dataset.
text_svm.fit(X_train,y_train)

Pipeline(steps=[('tfidf', TfidfVectorizer()), ('svm', LinearSVC())])
```

In the below image, we can see the obtained score which is 98.6% using svm model.

```
[ ] #Predictions of the test data
    y_preds_svm

    array(['ham', 'ham', 'ham', ..., 'ham', 'ham', 'ham'], dtype=object)

[ ] #Training score
    text_svm.score(X_train,y_train)

    1.0

[ ] #Testing score
    text_svm.score(X_test,y_test)

    0.9869494290375204
```

Now, we have build the confusion matrix and classification of the report based on the prediction made by the model to detect the better enhancement of the model. We can see the obtained results in the below image.

```
[ ] from sklearn.metrics import confusion_matrix

print(confusion_matrix(y_test,y_preds_svm))

[[1589    4]
 [  20  226]]

[ ] from sklearn.metrics import classification_report

[ ] print(classification_report(y_test,y_preds_svm))

              precision    recall  f1-score   support

         ham       0.99      1.00      0.99      1593
        spam       0.98      0.92      0.95       246

    accuracy                           0.99      1839
   macro avg       0.99      0.96      0.97      1839
weighted avg       0.99      0.99      0.99      1839
```

Based on the scores of both the models we have concluded SVM classifier model is the best suitable model for the spam classification.

Here, we are trying to integrate spam classifier with the summarization. In the below image, using nltk, tokenized the sentences and we have removed stopwords, extra spaces and lemmatized the words as you can see in below image.

```
[ ] text = ' Congratulations, you have won a lottery of $5000. To Won Text on,555500 '
```

```
[ ] import nltk
    nltk.download('punkt')

    [nltk_data] Downloading package punkt to /root/nltk_data...
    [nltk_data]   Package punkt is already up-to-date!
    True
```

```
[ ] # Removing Square Brackets and Extra Spaces
    article_text = re.sub(r'[[0-9]*]', ' ', text)
    article_text = re.sub(r's+', ' ', text)
```

```
[ ] # Removing special characters and digits
    formatted_article_text = re.sub('[^a-zA-Z]', ' ', article_text )
    formatted_article_text = re.sub(r's+', ' ', formatted_article_text)
```

```
[ ] # Tokenizing the sentences
    sentence_list = nltk.sent_tokenize(article_text)
```

```
#removing stopwords from the text.
stopwords = nltk.corpus.stopwords.words('english')
word_frequencies = {}
for word in nltk.word_tokenize(formatted_article_text):
    if word not in stopwords:
        if word not in word_frequencies.keys():
            word_frequencies[word] = 1
        else:
            word_frequencies[word] += 1
```

Here, we are calculating the term frequencies of the weighted words by the occurrences of the words. After that we are calculating the sentences weights by adding the words weights.

To find the weighted frequency, divide the frequency of the word by the frequency of the most occurring word.

```
[ ] #below is to find weights frequency
    maximum_frequncy = max(word_frequencies.values())
    for word in word_frequencies.keys():
        word_frequencies[word] = (word_frequencies[word]/maximum_frequncy)
```

```
#below is to get score for each sentence
sentence_scores = {}
for sent in sentence_list:
    for word in nltk.word_tokenize(sent.lower()):
        if word in word_frequencies.keys():
            if len(sent.split(' ')) < 30:
                if sent not in sentence_scores.keys():
                    sentence_scores[sent] = word_frequencies[word]
                else:
                    sentence_scores[sent] += word_frequencies[word]
```

We have taken the top 7 sentences to get the summary of the input text.

```
[ ] #Here the heapq library has been used to pick the top 7 sentences to summarize the article.
    import heapq
    summary_sentences = heapq.nlargest(7, sentence_scores, key=sentence_scores.get)
    summary = ' '.join(summary_sentences)
    print(summary)

     Congratulation , you have won a lottery of $5000.
```

Finally, we are giving the generated summary text to the SVM classifier model.

```
# Directly predicting on the summary of the message
type(ref)
text_mnb.predict(ref)

array(['spam'], dtype='<U4')
```

In the below image, you can see the output of the final model which is the given input is the spam.

Project Management:

- ### Work Completed:

  ### Description:

  we have taken spam and ham dataset and we have applied the data cleaning techniques to clean the data and build the model using ML classifier. Next, we have created a text summarizer which can generate a summary and this summarizer got integrated by the ML model.

- ### Tasks/Responsibilities:

  All together we have taken some time on problem statement and researched on it and have taken appropriate dataset which better fit the requirements.

  Iliyas Ahmed– I have worked on the one more ML classifier which is SVM and fit the model with train and test sets. Generated the accuracy and predicted the results on test set.

  Abdul Qayyoom Shaik– For the better enhancement of the svm model I worked on evaluation metrics to generate confusion matrix and the classification reports.

  Naveed Ahmed– I have worked on the summarization and developed a text summary model using nltk techniques. I have calculated the weights of the words and then calculated the weights of the sentences and selected the top sentences to give summarization.

Eswara Reddy– I have worked on the integration of the summary which was generated by the summarization model to the SVM classifier. I have also tested the model with many examples and got the best results.

Contributions:

Iliyas Ahmed (25%)

Abdul Qayyoom Shaik (25%)

Naveed Ahmed (25%)

Eswara Reddy (25%)

References / Bibliography:

Kaggle website:
https://www.kaggle.com/datasets/team-ai/spam-text-message-classification

GitHub:
https://github.com/Naveed945/Spam-Message-classifier

Journal document:

https://jusst.org/wp-content/uploads/2021/08/Classification-of-Spam-Text-using-SVM.pdf

[1]https://www.sciencedirect.com/science/article/pii/S0957417418303749?casa_token=UD57BHqhamkAAAAA:eg0zyT9YZ5o9BR8PverPJt8QFl_m950MJYszhiLT68Q0HR1BbP9Xp96reHtwAia1axem6Oih

[2] https://ieeexplore.ieee.org/document/8442564

[3]https://www.researchgate.net/publication/220637882_A_study_of_spam_filtering_using_support_vector_machines

[4] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8802784/

[5] https://iarjset.com/wp-content/uploads/2021/06/IARJSET.2021.8632.pdf

[6]https://www.academia.edu/42943100/Improving_spam_email_detection_using_hybrid_feature_selection_and_sequential_minimal_optimisation
[7] https://www.mecs-press.org/ijitcs/ijitcs-v9-n7/IJITCS-V9-N7-5.pdf

[8]https://www.researchgate.net/publication/266654684_The_curse_of_140_characters_Evaluating_the_efficacy_of_SMS_spam_detection_on_Android

[9] https://www.ijcseonline.org/pub_paper/90-IJCSE-06523.pdf

[10]https://www.technoarete.org/common_abstract/pdf/IJERCSE/v8/i7/Ext_02649.pdf