Spam Identification using machine learning based approach

Submitted by:

Arsalan Aman [11504701]

Nayana Shrestha [11550256]

Roshan Sah [11574385]

Sourabh Yadav [11508089]

Abstract

In today's world, email seems to have become immensely prevalent. In actuality, it has recently been touted as the quickest, most ubiquitous, and quickest mode of communication. Notwithstanding the numerous advantages of email, its use has been hampered by the large number of unwanted and often malicious emails that must be recognized and separated as soon as possible using a spam identification mechanism. Spam identification is critical for protecting email users and preventing several recent undesirables uses toward which email have indeed been put. Regrettably, the potency of spam detection system capabilities has also been restricted and occasionally found to be unproductive due to the inherent flexibility of unprompted emails through the usages of mailing methods, necessitating the development of better junk mail sensing methods to accomplish stronger spam detection capability. Numerous spam detection algorithms have been suggested and evaluated in the research; however, the observed effectiveness indicates that additional work in this area is still needed to attain improved accuracy. Our team has decided to develop an algorithm that will categorize emails into spam or genuine, hoping that it will save people with various types of scams through email.

Introduction

In today's world, email seems to have become increasingly fashionable. In fact, most individuals can't imagine their lives devoid of sending and receiving emails on a regular basis. Email has been used for a variety of purposes, including but not confined to internal, inter-organizational, and global communication; ads, job recruiting procedures, and international communication. In summary, the significance of email in today's world cannot be overstated. Notwithstanding the numerous advantages of email, its use has been hampered by the large number of unwanted and often counterfeit emails that must be recognized and separated as soon as possible using a spam identification mechanism. Spam detection distinguishes between spam and non-spam emails, allowing spam email to be prevented from reaching users' inboxes. As a result, spam recognition is perhaps the foremost critical phase in the email filtration for preventing trash mails from reaching users' inboxes, especially in this era of massive junk mail owing to the existence of massive emailing services, which has increased the volume of spam emails.

Spam identification is unquestionably necessary to safeguard email recipients and avoid a variety of recent undesirable uses toward which emails have indeed been exposed. Regrettably, owing to the adaptability of unwanted emails via the utilization of mailing systems, the efficiency of spam recognizing systems has frequently been restricted, and even in some cases deemed useless or hacked, necessitating the development of stronger spam detection techniques. Numerous spams detecting algorithms have indeed been presented and evaluated in the research, however the claimed efficiency still calls for additional research in this area to improve accuracy.

In the proposed work, the main goal is classifying the email prompts as spam or non-spam. In the era of Artificial Intelligence, it's easy to think of solutions based on machine learning and ensembled based algorithms.

Related Work

Email Classification

This project has focused on classifying emails into custom folders that are relevant to the user. Getting a vast number of emails and being unable to categorize the important ones with the normal ones has been the major issue for all of us. This project solves this problem by automatically categorizing the emails into the folders to make our inbox more sorted and easier to read. Two different approaches have been used in this project —Naïve Bayes classifier and k-nearest neighbors' algorithm. The Naïve Bayes classifier is based on a probabilistic model, while the k-nearest neighbors' algorithm is based on a similarity measure with the training emails. These two approaches have been used in email classification, analyze the performance of these algorithms, and compare their results. [1]

Improving customer complaint management by automatic email classification using linguistic style features as predictors

Coussement and Van den Poel offered an automated e-mail categorization method for distinguishing complaints from non-complaints. They presented new ways of handling complaint emails using the machine learning algorithm. They show a boosting classifier that categorizes e-mails as complaints or non-complaints. The authors also contend that using linguistic characteristics can increase classification performance. In this study, linguistic style information from an email is utilized to determine if the incoming email is a complaint or not. [2]

Background

In the world of electronic mail, we often face the issue of spam email. Some professionals must deal with thousands of emails a day as it is their primary mode of communication. Some scammers can get access to some email IDs and send malicious emails to which these individuals fall prey. Our goal is to determine in each data set which of these emails is malicious and which ones are not. Through this classification, we can filter out the important emails which are meant for the intended recipient and mark the ones which are spam.

Experimental Methodology

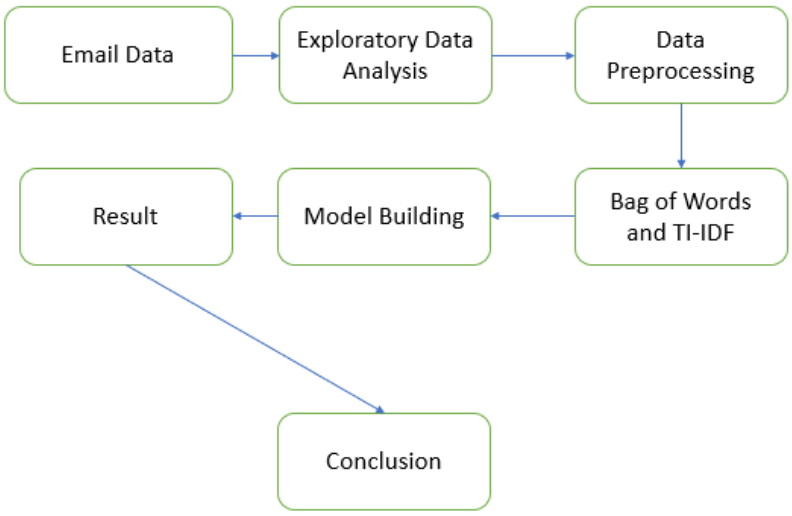


Figure 1: Flow-diagram

Data Accommodation

Data accommodation is perhaps the most important stage in putting the Machine Learning model into action. To build the machine, data is gathered and filtered in this stage. Data points in their native format are meaningless and convey no knowledge. This could occasionally include inaccurate numbers that jeopardize the model's integrity. Data points may have null values, which should be identified and addressed even before classifier is developed. Furthermore, datasets must be gathered from reliable sources; else, the model's efficiency could suffer.

Data Collection

Data collection is a strategy for obtaining and examining precise intelligence in order to provide replies to inquiries and audit the outcomes. It often relates to the integrity and amount of data, with the latter indicating the accuracy. Data Synthesis is the result of Data Collection. To use acquired data to construct Machine Learning algorithms, it must be captured and processed in a manner that is appropriate for the architecture in question. For database professionals, there is a limit to how much data can be absorbed. The sample of dataset with the features (label and text) is shown in figure 2.

	label	text
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...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...
10	ham	I'm gonna be home soon and i don't want to tal...
11	spam	SIX chances to win CASH! From 100 to 20,000 po...

Figure 2: Sample of Dataset

Figure 3 describes the number of spam and ham data in the whole dataset. We have a smaller number of spam data as compared to ham, so we use SMOTE function to balance the dataset

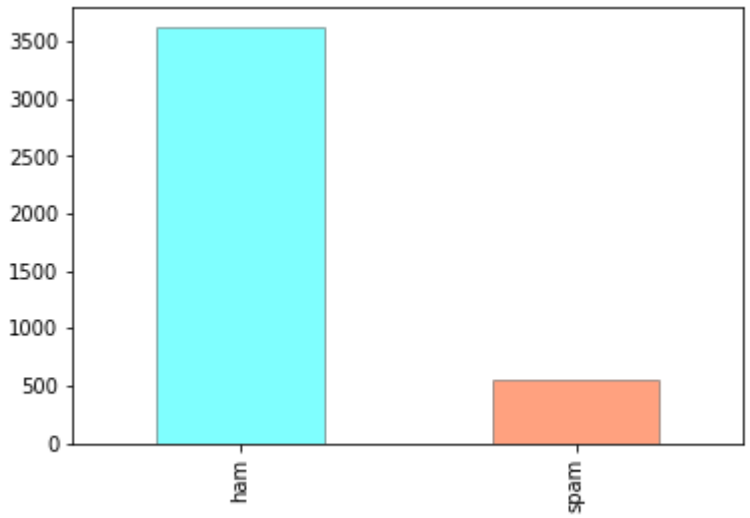


Figure 3: BarCharts for spam and ham dataset

Data Preparation

The most challenging and time-consuming step in Machine Learning modeling is data preparation. The fundamental reason for this is that each database is unique and tailored to certain tasks. This procedure establishes a framework within that we may assess the project's data preprocessing requirements. In this stage of the modeling process, the knowledge is readied for training. This process involves cleansing, which includes error detection and rectification, duplication elimination, normalization, and so on.

```
[ ]  #Showing the shape of dataset
     print(" no of rows", len(data))
     data.shape
```

```
 no of rows 5572
(5572, 2)
```

```
[ ]  data.describe()
```

	label	text
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

Figure 4: Dataset Shape and Description

Model Training

- **Data Split**

Data has become available for evaluation once it has been gathered and processed. The primary goal for preparation is to ensure that the training is as effective as possible. First and foremost, we divided the data into two sections for training: a training set and a testing set. Splitting is frequently done in a 70-30 ratio. 70 percent of the data is allocated to the training set, while 30% is allocated to the testing set. Although these division proportions are not set, they can be changed to meet specific needs.

- **Feature Extraction**

The dataset is again accessible for model calibration after the division has been completed successfully. The target/classifying features are taken from the sample and supplied to the classification model before training the model. To do evaluation on the desired object, many algorithms are used. There are several techniques to perform feature extractions, such as

Principal component analysis which is generally done for statistical datasets. Furthermore, Bag word approach is used for text-based datasets, where some set of words is considered as dictionary and defined as the set for representing identity. Moreover, there is a term frequency approach which is also used for text-based datasets. In this, term frequency and inversed term frequency is calculated to generalize the correlation among the data inputs corresponding to the predictor variable or target variable. We count the frequency of the text of both ham and spam label to compute the model through TF-IDF. The graph in the figure 5 describes the count of words of the text with respect to ham and spam label.

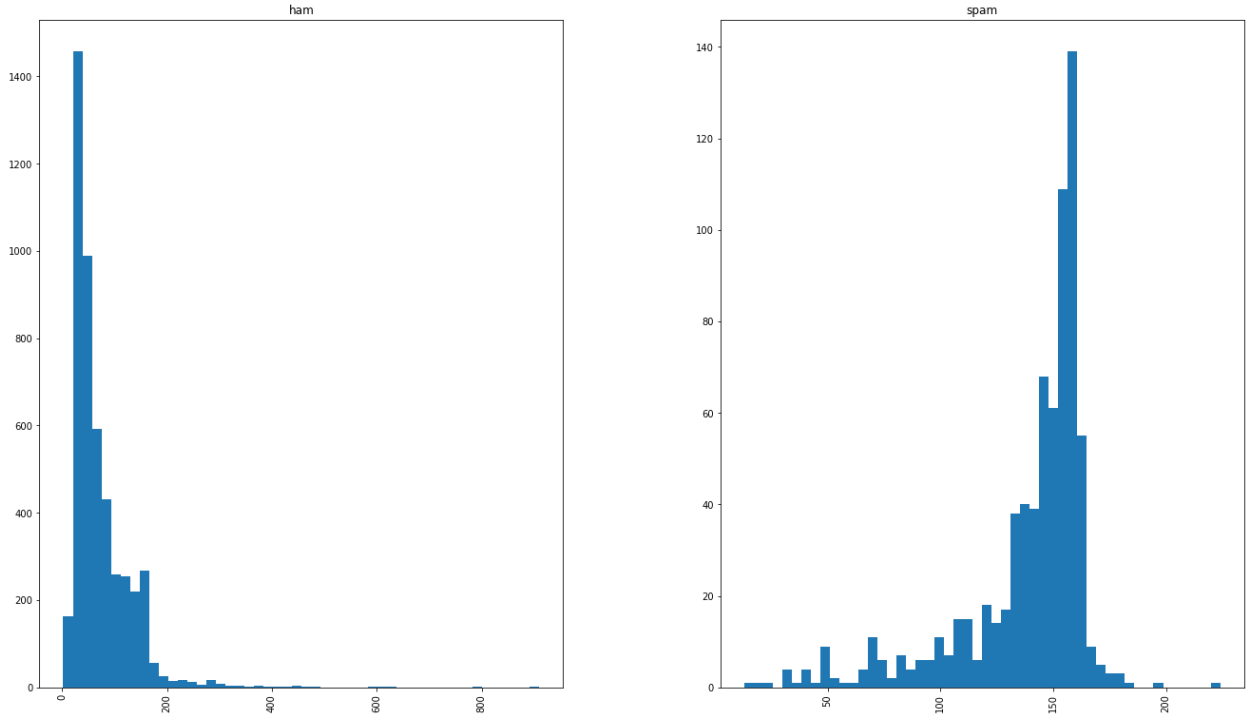


Figure 5: Frequency of ham and spam dataset

In figures 6 and 7, we visualize the most common words in the text used in ham and spam label respectively. We can analyze the quickly which keyword is spam or ham by seeing these words.

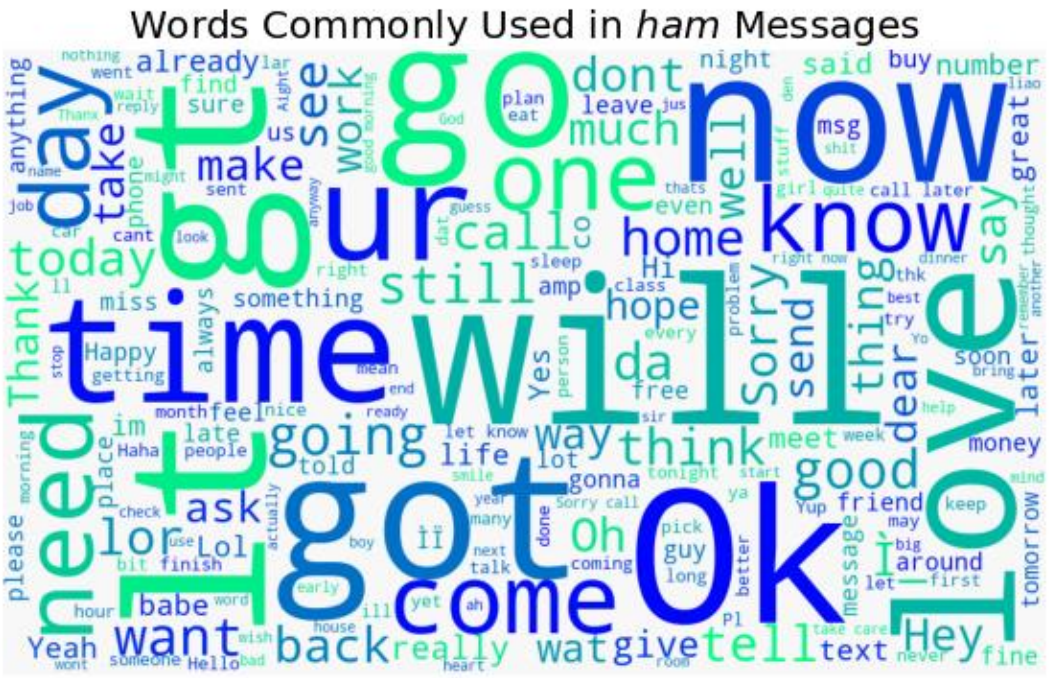


Figure 6: Common word in Ham dataset

Figure 7: Common word in spam Dataset

· **Model Fitting**

Once the categorizing trait has been retrieved, it is delivered to the classifiers. For doing evaluation on the retrieved characteristics, several algorithms are used. There are many different types of classifiers that may be used to do analytics, and several of them are given below:

§ ***Naïve Bayes***

It's a probabilistic classifier dependent on Bayes' Hypothesis and the premise of prediction independent. A Naive Bayes predictor, in basic words, posits that the existence of one characteristic in a category is independent to the possession of any subsequent characteristic. The Naive Bayes algorithm is simple to construct and is especially good for massive datasets. Naive Bayes is renowned to dominate even the most advanced categorization systems due to its minimalism.

§ ***Logistic Regression***

Logistic regression is a categorization approach derived from mathematics by machine learning. A quantitative strategy for assessing a data wherein one or more distinct variables predict a result is known as logistic regression. The goal of logistic regression is to determine the model that best describes the connection among the reliant and autonomous variables. Logistic regression is a machine learning classification approach. The reliant parameter is modeled using a logistic equation. The reliant variable is dualistic, which means that only two classifications are conceivable.

§ ***Multinomial Linear Support Vector Machine***

The SVM classifier computation goal is to identify a higher dimensional space in an N-dimensional space (N = the degree of variables) that distinguishes between data points. Support vectors represent datasets which are proximal towards the hyperplane and have an impact on the hyperplane's location and alignment. We optimize the classifier's range by utilizing such support vectors. The hyperplane's location will be altered if the support vectors are deleted. It's these factors that aid in the development of SVM.

§ ***Decision Tree***

A decision tree is a data flow diagram configuration in which every intrinsic entity symbolizes a functionality assessment (e.g., whether a coin flip will land heads or tails), for every leaf entity symbolizes a class label (choice taken after calculating all characteristics), and stems portray functionality combinations that led to such labels. The categorization criteria are represented by the pathways between root to leaf. In mathematics, data mining, and machine learning, a decision tree is among the prognostic modeling techniques. Decision trees are created using an algorithm that determines multiple ways to segment a data set depending on certain factors. It is amongst the foremost popular and useful supervised learning algorithms.

Model Evaluation

Overfitting occurs when a model has been trained well enough on the training set but inadequately on the testing set. It could only operate well enough on the training set and not on

unobserved or actual statistics. The algorithm is under-fitting if that underperforms on both the training and testing sets. Additional input is required to fit the classifier.

- **Accuracy Metrics**

One parameter for assessing classifier model is accuracy. Colloquially, accuracy refers to the percentage of correct forecasts made by the algorithm. The preceding is the precise notion of accuracy:

In terms of Binary Classification, Accuracy is defined in terms of the True Positive, True Negative, False Positive and False Negative.

For the matter of fact, while dealing with a class-imbalanced data collection, accuracy merely convey ’s entire situation.

- **Precision**

The accuracy is measured as the proportion of accurately described Positive samples to the overall count of Positive observations (either correctly or incorrectly). The precision of the algorithm in categorizing a data as positive is measured. It mainly gives the glimpse that how precisely a model an classify the positives. Mathematically it is represented as follows,

- **Recall**

The recall is determined by dividing the overall count of Positive observations by the proportion of Positive instances accurately categorized as Positive. The model's capability to recognize Positive inputs is measured by the recall. The greater the recall, the greater the number of positive samples found.

- **F1-Score**

The F-score or F-measure is a metric of a program's efficiency in binary categorization data analytics. It is determined using the study's precision and recall, with precision equaling the count of true positive outcomes divided by the total count of positive outcomes, along with those that were mislabeled, and recall equaling the count of true positive outcomes divided by the total count of samples which could perhaps had already been recognized as positive.

Results

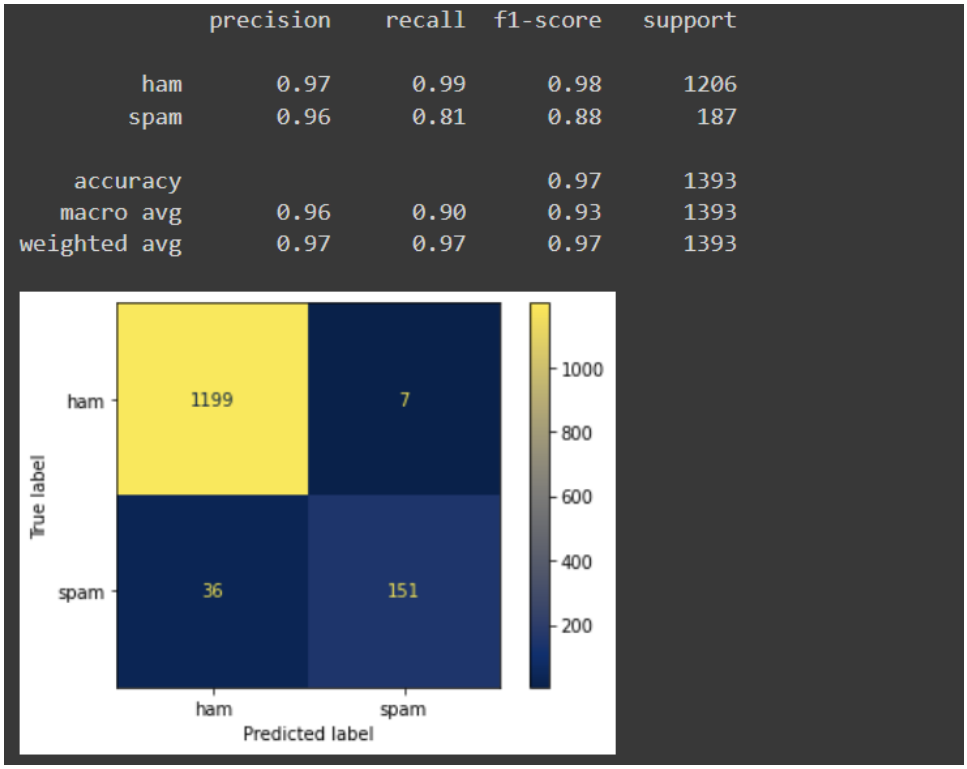
As discussed, we have used the following four models to get our results that is, Multinomial Naive Bayes, Logistic Regression, Multinomial Linear Support Vector machine and Decision Tree. Also, the two types of preprocessing used were bag of words and TF-IDF. For the evaluation of the implemented four models with both the preprocessing methods used, we have used the confusion matrix measures such as accuracy, precision, recall and F1 measure. We have compared the outcomes of each model with each preprocessing technique based on these measures to get the best model. We can see the differences of accuracy of the same model while

using the bag of words and TF-IDF. So, we can also say that the accuracy of the model depends on the preprocessing of the dat

A) While preprocessing data using bag of words

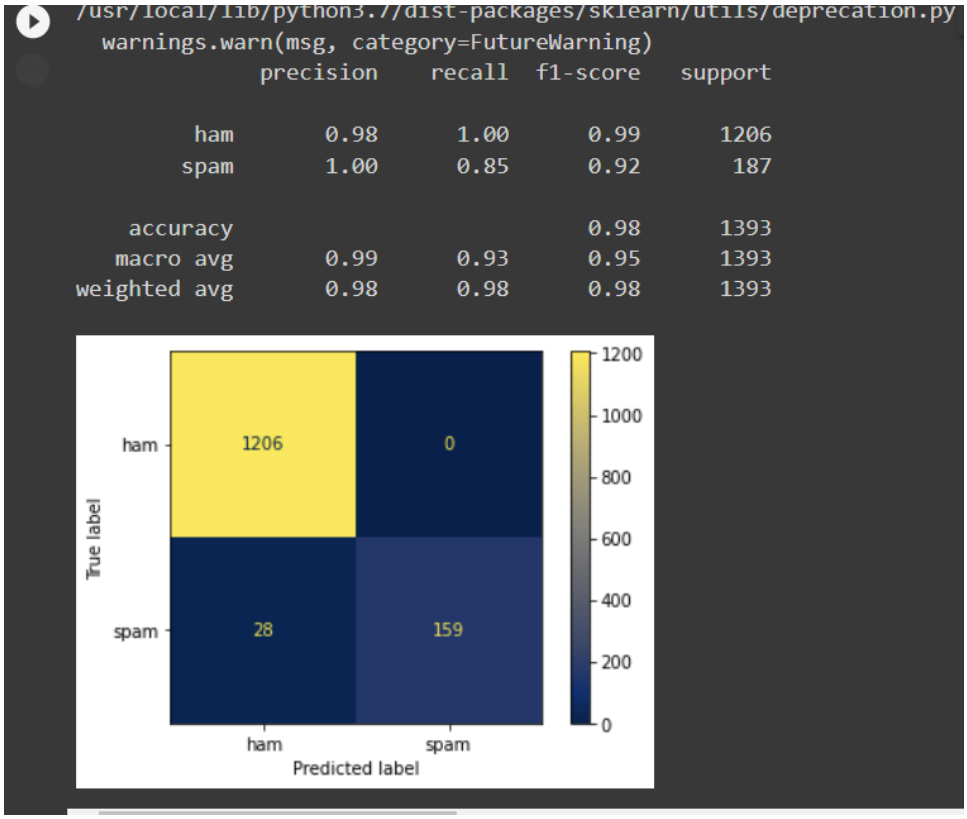
1. Multinomial Naive Bayes model

With the combination of bag of words and the multinomial naïve bayes model we get the accuracy of 0.97. The figure below illustrates the confusion matrix with its measure and the test accuracy gained through this model.



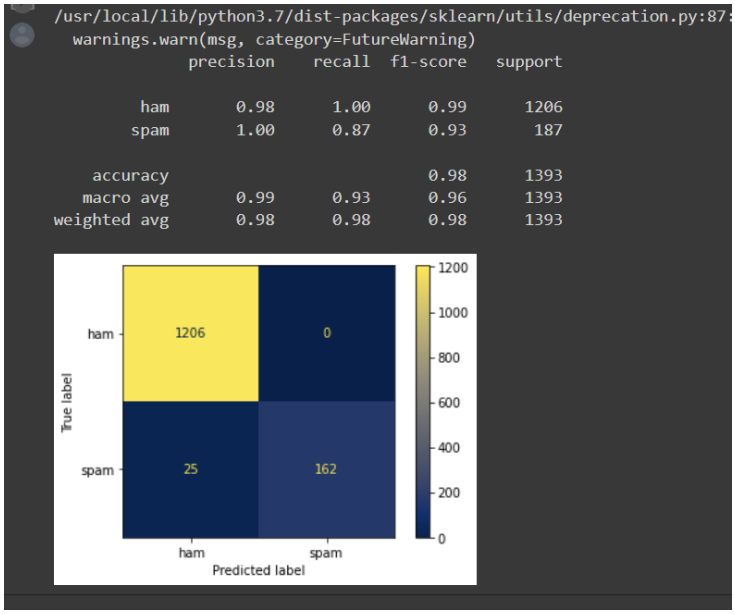
2. Logistic Regression Model

Below is the evaluation of logistic regression model when the data is preprocessed using the bag of word technique. The accuracy gained from this combination is 0.98 as seen in the figure. The confusion matrix with both ham and spam can be seen as well.



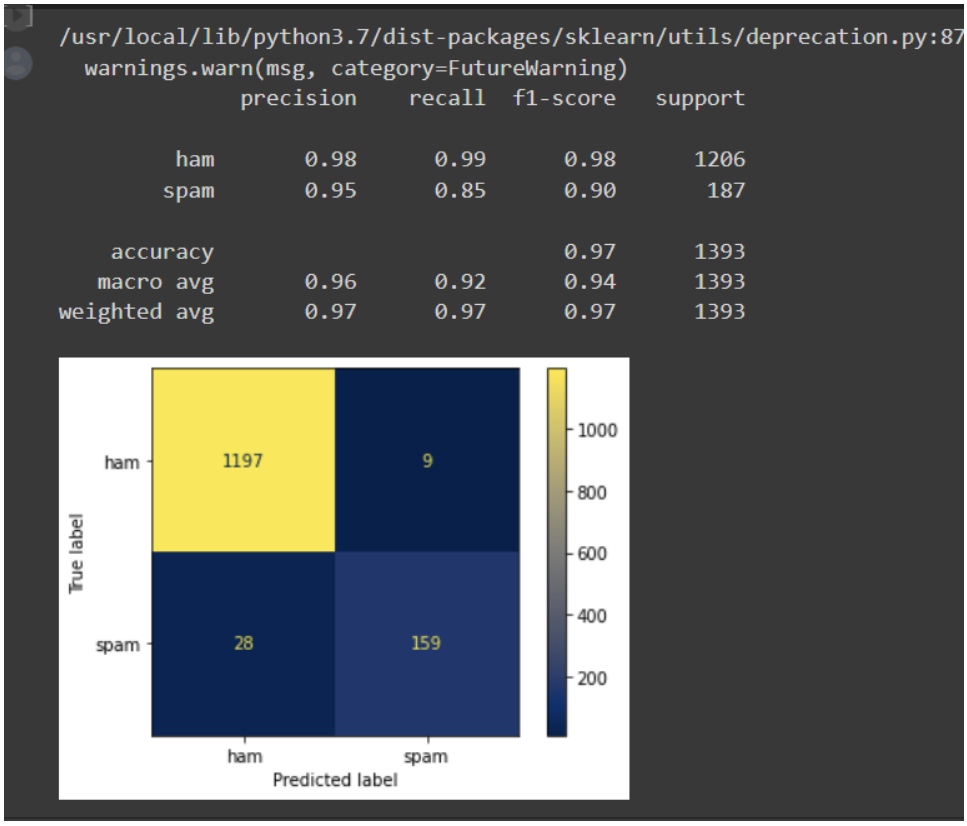
3. Multinomial Linear Support Vector machine model

With the combination of bag of words and the multinomial Linear Support Vector machine model we get the accuracy of 0.98. The figure below illustrates the confusion matrix with its measures(accuracy, precision, recall and F1 measure) and the test accuracy gained through this model.



4. Decision Tree model

With the combination of bag of words and the decision tree model we get the accuracy of 0.97. The figure below illustrates the confusion matrix with its measures (accuracy, precision, recall and F1 measure) and the test accuracy gained through this model.

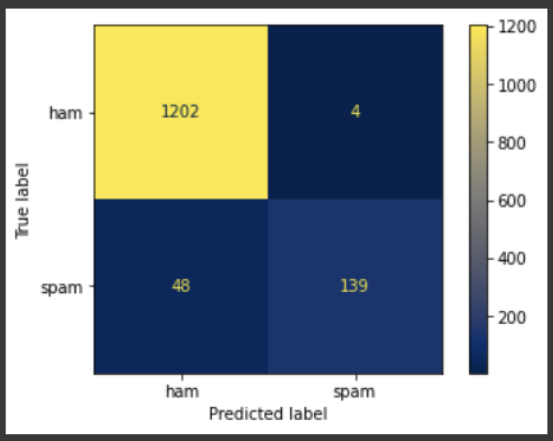


B) Data preprocessing using TF-IDF

1. Multinomial Naive Bayes model

Here we have used Multinomial Naive Bayes model on the preprocessed data through TF-IDF techniques. The accuracy we get here is 0.96 and the evaluation values of the confusion matrix is show in the figure below.

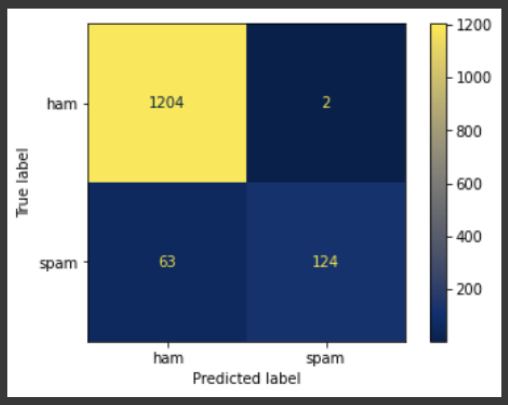
```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: Fu
  warnings.warn(msg, category=FutureWarning)
              precision    recall  f1-score   support

         ham       0.96      1.00      0.98      1206
        spam       0.97      0.74      0.84       187

    accuracy                           0.96      1393
   macro avg       0.97      0.87      0.91      1393
weighted avg       0.96      0.96      0.96      1393
```



2. Logistic Regression Model

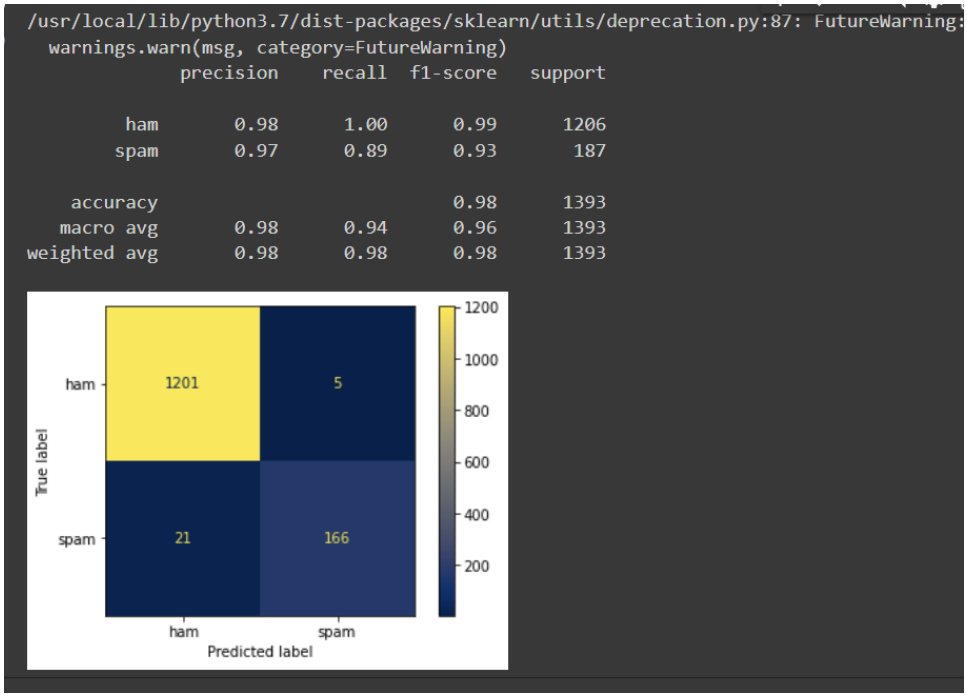
Below is the confusion matrix of the data when classified using logistic regression model. The accuracy we gained from this model is 0.95.

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarn
  warnings.warn(msg, category=FutureWarning)
              precision    recall  f1-score   support

         ham       0.95      1.00      0.97      1206
        spam       0.98      0.66      0.79       187

    accuracy                           0.95      1393
   macro avg       0.97      0.83      0.88      1393
weighted avg       0.95      0.95      0.95      1393
```



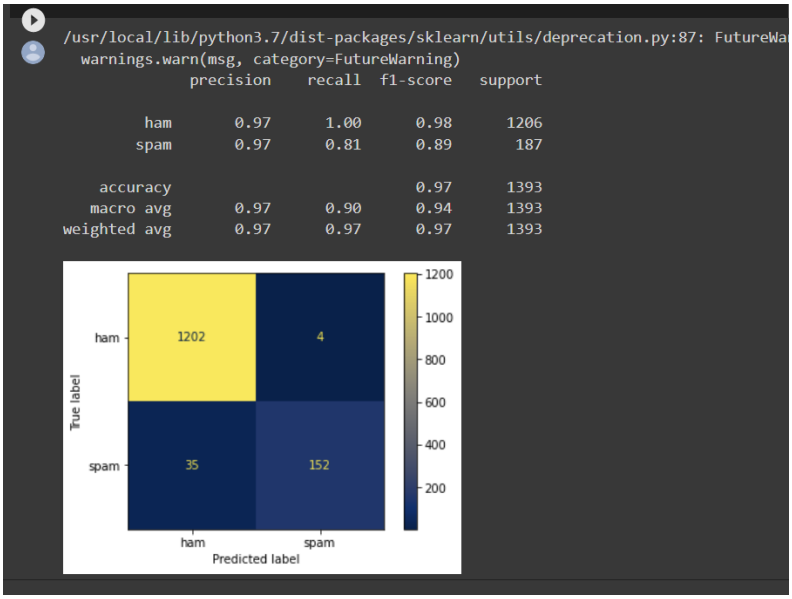
3. Multinomial Linear Support Vector machine model

With the combination of TF-IDF and the multinomial Linear Support Vector machine model we get the accuracy of 0.98. The figure below illustrates the confusion matrix with its measures (accuracy, precision, recall and F1 measure) and the test accuracy gained through this model.



4. Decision Tree model

Below is the evaluation of decision tree model when the data is preprocessed using the TF-IDF technique. The accuracy gained from this combination is 0.97 as seen in the figure. The confusion matrix with both ham and spam can be seen as well.



Conclusion

The proposed work introduces the spam detection model which has the ability to distinguish spam emails effectively. The computations done in the proposed models are based on machine learning based algorithms. The machine learning models are trained upon structured database on top of which models are trained after splitting the dataset into two parts, i.e., Training (80% of whole dataset) and Testing (20 % of whole dataset). Also, techniques like bag of words and Term Frequency and Inverse Term Frequency are also employed to perform effective pre-processing of the dataset. After building the models and performing model evaluation, it can be concluded that Logistic Regression, Linear Support Vector Machine are the best performing models having the same value of accuracy using the Bag of words but while using the TF-IDF, Linear Support Vector machine and Decision tree perform well as given below in the table 1 and 2. In the case of both Bag of words and TF-IDF, Linear Support Vector machine performs very good having the same value of 0.98 accurate rate which is high accuracy rate. Hence, we can say the Linear Support Vector Machine performs well in both cases.

Model	Accuracy	Target	Precision	Recall	F1 score
Multinomial Naive Bayes	0.97	Ham	0.97	0.99	0.98
		Spam	0.96	0.81	0.88
Logistic Regression	0.98	Ham	0.98	1.00	0.99
		Spam	1.00	0.85	0.92
Linear Support Vector machine	0.98	Ham	0.98	1.00	0.99
		Spam	1.00	0.87	0.93
Decision Tree	0.97	Ham	0.98	0.99	0.98
		Spam	0.95	0.85	0.90

Table 1: Using Bag of words

Model	Accuracy	Target	Precision	Recall	F1 score
Multinomial Naive Bayes	0.96	Ham	0.96	1.00	0.98
		Spam	0.97	0.74	0.84
Logistic Regression	0.95	Ham	0.95	1.00	0.97
		Spam	0.98	0.66	0.79
Linear Support Vector machine	0.98	Ham	0.98	1.00	0.99
		Spam	0.97	0.89	0.93
Decision Tree	0.97	Ham	0.97	1.00	0.98
		Spam	0.97	0.81	0.89

Table 2: Using term frequency-inverse document frequency (TF-IDF)

References

1. Timothyabwao. (2022, February 28). *Spam detection: Eda + bag-of-words model*. Kaggle. Retrieved March 1, 2022, from https://www.kaggle.com/code/timothyabwao/spam-detection-eda-bag-of-words-model
2. Shantanudhakadd. (2022, February 27). *Email spam classifier using naive bayes*. Kaggle. Retrieved March 1, 2022, from https://www.kaggle.com/code/shantanudhakadd/email-spam-classifier-using-naive-bayes
3. rmodi6. (n.d.). *RMODI6/email-classification: Classifying emails into custom user labels*. GitHub. Retrieved April 19, 2022, from https://github.com/rmodi6/Email-Classification
4. Coussement, K., & Poel, D. V. den. (2007, October 24). *Improving customer complaint management by automatic email classification using linguistic style features as predictors*. Decision Support Systems. Retrieved April 19, 2022, from https://www.sciencedirect.com/science/article/pii/S0167923607001820?via%3Dihub