

Innovators Hub

Project Report

Project Title: **Email Spam Detection**

Presented By: Dia Kaul  
 Minautee

Date (DD/MM/YYYY): 29/05/2021

1. Project Members
2. Dia Kaul: [diakaul2001@gmail.com](mailto:diakaul2001@gmail.com)
3. Minautee: [minauteestmic@gmail.com](mailto:minauteestmic@gmail.com)
4. Project Abstract(M)

Tons of emails are sent every day which includes spam mails as well. Due to their passage, people may miss out on important mail. Email providers have the huge task of filtering out the spam and making sure their users receive the messages that matter. Spam mails aim to draw attention to a cause or spread false information. Some of the examples include mails like YOU WON A LOTTERY, short body text, too many BCC recipients, etc. To avoid spamming of emails, one could block the sender’s email address or create a filter, but over the years, spammers began to use several tricky methods to overcome the filtering by changing words to the beginning or end of the message. The tedious task can be resolved with the help of Machine Learning. Through the machine learning approach, sample sets of pre-classified email messages are created which are then trained and tested to work. There are many algorithms that can be used to learn the classification rules from these email messages. Some of the algorithms that are commonly used are Naive Bayes, support vector machines, Neural Networks, K-nearest neighbour, etc. This project uses the Naive Bayes algorithm to successfully filter out spam emails from the ham ones.

1. Table of Contents
2. Introduction & Problem Statement
3. Approach to the Problem
4. Theoretical Results
5. Practical Results
6. Results Explanation
7. Conclusion
8. References
9. Introduction & Problem Statement

Problem Statement:

Build a spam email detector using AI/ML.  
(Do appropriate data pre-processing and post processing mechanisms as required.)

The project uses the sklearn libraries and the naïve bayes classifiers for the purpose of spam detection.

1. Approach to the Problem

The approach/steps that we took to solve the problem was:  
 1. We took a dataset of numerous emails

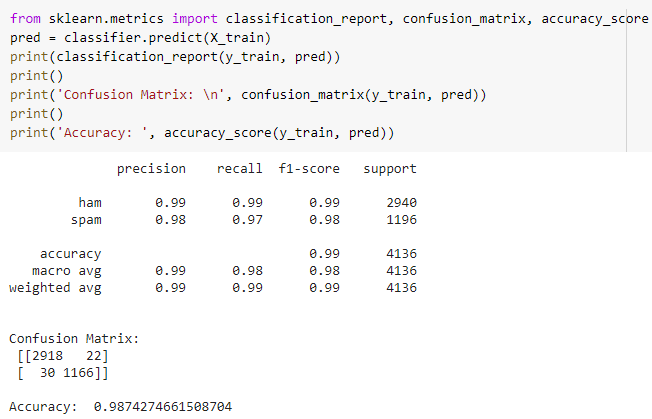
1. Import libraries
2. Import the dataset.(with markings as ‘ham’ or ‘spam’)
3. We drop the duplicate values
4. We check for null values.
5. Define process to remove punctuations
6. Remove the stop-words.
7. Converting the filtered text to token count matrix.( Basically mentions the number of words also repetitions)
8. Split dataset into training set(80%) and test set(20%).
9. Standardize the features.
10. Train and fit the model.
11. Test model.
12. Theoretical Results
    * We imported our **dataset spam\_ham\_dataset.csv** from Kaggle. After that implemented the program using **Google Colab**. Firstly, we imported libraries and loaded the dataset then.
    * After reading the dataset, we got the number of rows and columns and checked for duplicates to be removed. There were no duplicates found in the program.
    * Then the stopwords package was downloaded. [Stopwords: Common words that are used in spam mails]
    * In order to remove the filter spam from ham, firstly all the punctuations were removed, then stopwords were removed and a list of clean words was generated.
    * The tokenization was performed on the following list. [Tokenization: Splitting the data into minimal meaningful units.]
    * Then the list of words was converted into a matrix. For example:
    * Graphical user interface, text, application

      Description automatically generated
    * After that the data was split into training (80%) and testing (20%) and the shape was obtained.
    * Then, Naïve Bayes classifier was imported and trained. Predictions and actual values were printed then after
    * The Model was finally tested for training dataset for which we obtained an accuracy of 98.74% and for testing dataset we obtained an accuracy of 97.39%.

VII.Practical Results

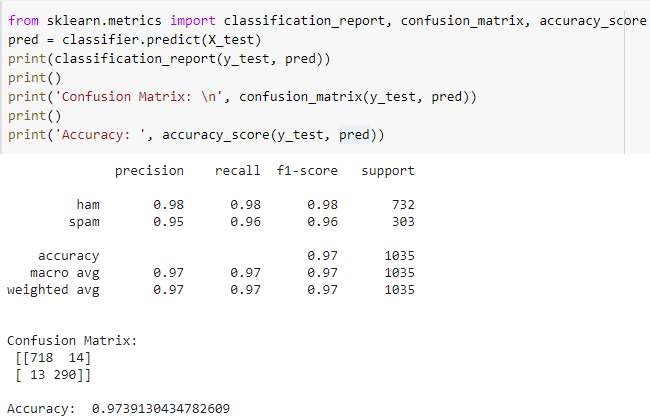
As described in problem approach and theoretical approach, after we have trained and tested the model and after that we print the classification report for trained and tested data both.

For trained data:



We received a good accuracy of 98.74% for our trained model. Also showing the confusion matrix, precision, recall f1-score (classification report)

For test data:

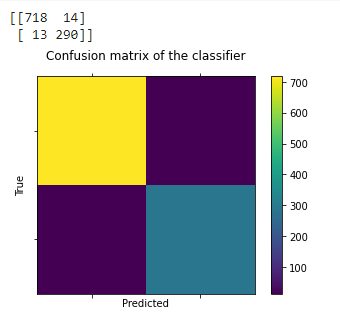


Finally for data to be tested we receive an accuracy of **97.39%.**  
With this we also print the confusion matrix and all the components of classification report.

1. Results Explanation

With an accuracy of 97.39% the model predicts if the mail is spam or not(or “ham”).

Confusion matrix:

Graphical user interface, application

Description automatically generated

As shown in the confusion matrix, the model can predict 718 proper mail(ham) and missed 14 i.e., the model failed in predicting the 14 mails as actual mails and predicted it as spam.  
Similarly the model successfully predicted 290 mails as spam but failed for predicting 13 as spam.

1. Conclusion

For this project, we successfully implemented a model for spam email detection using multiple tools like sklearn library, matplotlib, confusion matrix, nltk and most importantly naïve bayes classification from sklearn.

The model had a satisfactory accuracy of 97.39%.

1. References

* https://towardsdatascience.com/spam-detection-with-logistic-regression-23e3709e522
* https://www.researchgate.net/publication/50211017\_Machine\_Learning\_Methods\_for\_Spam\_E-Mail\_Classification
* https://www.python-course.eu/metrics.php
* https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/#:~:text=A%20Confusion%20matrix%20is%20an,by%20the%20machine%20learning%20model
* https://bdtechtalks.com/2020/11/30/machine-learning-spam-detection/