Spam Email's Classification

Article in Advance in Computer Science - Nov 2023

Author:

GitHub Link:- https://github.com/Mohanadsfe

Sentiment Analysis in English Texts

Mohanad Safi
Faculty of Computer Science & Mathematics,
The Ariel University, Israel.

ARTICLE INFO
Article history:
Received: 14 Oct, 2023
Accepted: 22 Nov, 2023
Online: 28 October, 2023
Keywords:
Spam Email's
Balanced Dataset
Unbalanced Dataset
Classification

ABSTRACT

Nowadays, emails are used in almost every field, from business to education. Emails have two subcategories, i.e., ham and spam. Email spam, also called junk emails or unwanted emails, is a type of email that can be used to harm any user by wasting his/her time, computing resources, and stealing valuable information. The ratio of spam emails is increasing rapidly day by day. Spam detection and filtration are significant and enormous problems for email and IoT service providers. Among all the techniques developed for detecting and preventing spam, filtering email is one of the most essential and prominent approaches. Several machine learning and deep learning techniques have been used, i.e., Naïve Bayes, decision trees, neural networks, and random forests. This paper surveys the machine learning techniques used for spam filtering techniques used in email and IoT platforms by classifying them into suitable categories. A comprehensive comparison of these techniques is also made based on accuracy, precision, recall, etc. In the end, comprehensive insights and future research directions are also discussed.

1. Introduction

In the era of information technology, information sharing has become very easy and fast. Many platforms are available for users to share information anywhere across the world. Among all information sharing mediums, email is the simplest, cheapest, and the most rapid method of information sharing worldwide. But, due to their simplicity, emails are vulnerable to different kinds of attacks, and the most common and dangerous one is spam

No one wants to receive emails not related to their interest because they waste receivers' time and resources. Besides, these emails can have malicious content hidden in the form of attachments or URLs that may lead to the host system's security breaches.

Spam is any irrelevant and unwanted message or email sent by the attacker to a significant number of recipients by using emails or any other medium of information sharing.

So, it requires an immense demand for the security of the email system. Spam emails may carry viruses, rats, and Trojans. Attackers mostly use this technique for luring users towards online services. They may send spam emails that contain attachments with the multiple-file extension, packed URLs that lead the user to malicious and spamming websites and end up with some sort of data or financial fraud and identify theft

Many email providers allow their users to make keywords base rules that automatically filter emails. Still, this approach is not very useful because it is difficult, and users do not want to customize their emails, due to which spammers attack their email accounts.

In the last few decades, Internet of things (IoT) has become a part of modern life and is growing rapidly. IoT has become an essential component of smart cities. There are a lot of IoT-based social media platforms and applications. Due to the emergence of IoT, spamming problems are increasing at a high rate. The researchers proposed various spam detection methods to detect and filter spam and spammers. Mainly, the existing spam detection methods are divided into two types: behaviour pattern-based approaches and semantic pattern-based approaches. These approaches have their limitations and drawbacks. There has been significant growth in spam emails, along with the rise of the Internet and communication around the globe . Spams are generated from any location of the world with the Internet's help by hiding the attacker's identity. There are a plenty of antispam tools and techniques, but the spam rate is still very high. The most dangerous spams are malicious emails containing links to malicious websites that can harm the victim's data. Spam emails can also slow down the server response by filling up the memory or capacity of servers. To accurately detect spam emails and avoid the rising email spam issues, every organization carefully evaluates the available tools to tackle spam in their environment. Some famous mechanisms to identify and analyze the incoming emails for spam detection are Whitelist/Blacklist , mail header analysis, keyword checking, etc.

2. Literature Review

As social media websites have attracted millions of users, these websites store a massive number of texts generated by users of these websites. Researchers were interested in investigating these metadata for search purposes. In this section, a number of research papers that explored the analysis and classification of Spam Email's metadata were surveyed to investigate different text classification approaches and the text classification results.

Authors noticed that many Email's users use the URL section of the profile to point to their blogs, and the blogs provided valuable demographic information about the users. Using this method, the authors created a corpus of about 7142 Emails users labeled with their gender. Then author arranged the dataset for experiments as following: for each user; they specify four fields; the first field contains the text of the email and the Second One is Email-Address. After that, the author conducted the experiments and found that using all of the dataset fields while classifying Emails user's spam provides the best accuracy of 98%. Using Emails text only for classifying Spam-Email's provides an accuracy of 98%.

the authors used Machine Learning approaches for Spam-Email's Analysis. Authors constructed a dataset consisting of more than 7142 English Email's labeled as "3571 ham tweets and 3571spam". Several Machine Learning Algorithms were applied, such as Naive Bayes (NB), KNN, Support vector machine (SVM), RF, Logistic regression (LR) , and DT. The authors found that RF and SVM provided the most accurate results on classifying email's, while KNN classifier results were the least accurate results.

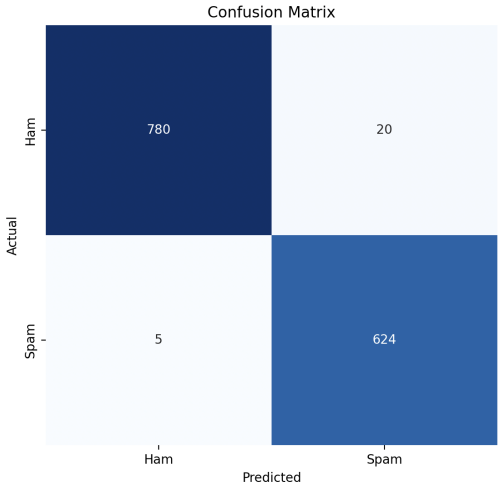
The author applied machine learning approaches using classifiers on the collected dataset. They reported the following: (1) The best classifier applied on the dataset is SVM, (2) Classifying a balanced set is challenging compared to the unbalanced set. The balanced set has fewer emai's than the unbalanced set, which may negatively affect

the classification's reliability. The author investigated the effects of applying preprocessing methods before the spam classification of the text. The authors used classifiers and Two datasets to evaluate the preprocessing method's effects on the classification. Experiments were conducted, and researchers reported the following findings: **`Replace contractions`** has no much effect, Removing stop words have a slight effect, Removing Numbers have no effect, Expanding Acronym improved the classification performance, and the same preprocessing methods have the same effects on the classifier's performance, SVM and RF classifiers showed more sensitivity than LR and KNN classifiers. In conclusion, the classifier's performance for spam email's analysis was improved after applying preprocessing methods.

3. Proposed Approach

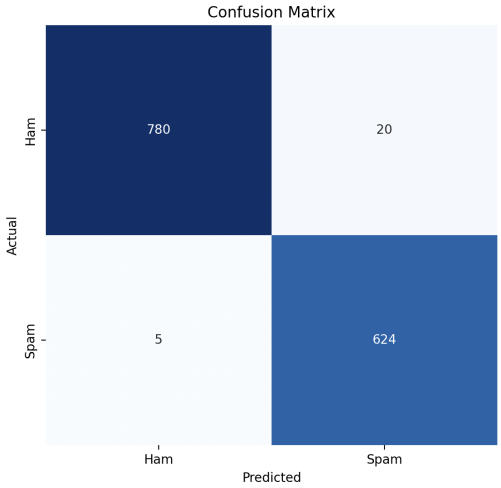
In this work, the author implemented and evaluated different classifiers in classifying the spam-emai's of the emails. Classifiers were applied on both balanced and unbalanced datasets. Classifiers used are Decision Tree, Naïve Bayes, Random Forest, K-NN, LR, and Random Tree.

Algorithm LR

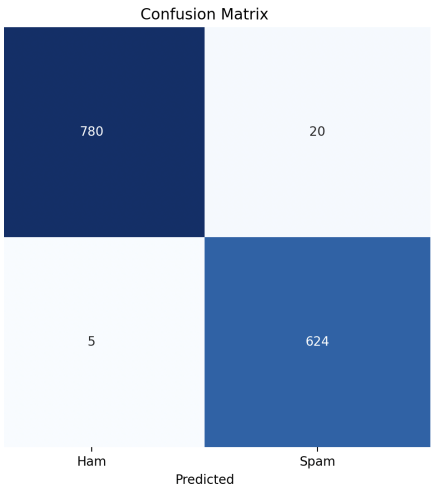


ALGORITHMES	ACC_TRAIN	ACC_TEST
KNN	**0.911**	**0.907**
LR	**0.999**	**0.982**
SVM	**0.999**	**0.982**
RF	**0.999**	**0.976**
DT	**0.999**	**0.972**
NB	**0.957**	**0.918**
KNN_less_P	**0.925**	**0.916**

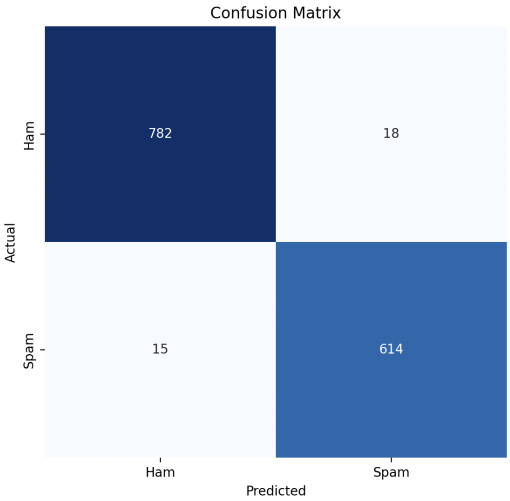
SVM



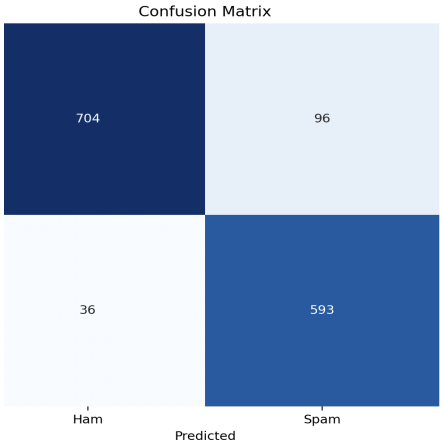
DT



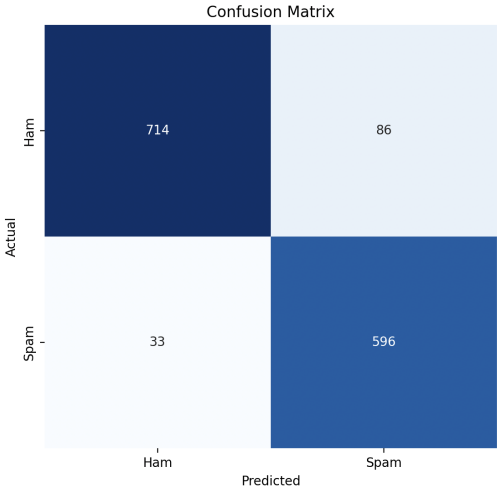
RF



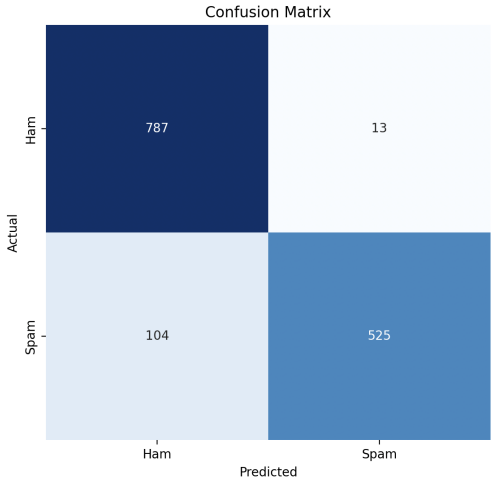
KNN-WithALLProcessDATA



KNN-WithOutALLProcessDATA

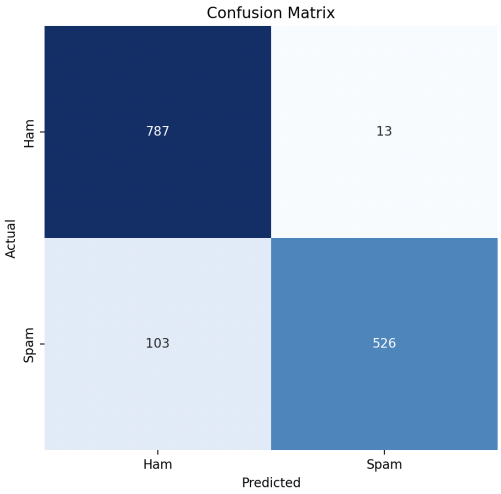


NB



ALGORITHMES	ACC_TRAIN	ACC_TEST
KNN	**0.911**	**0.907**
LR	**0.984**	**0.974**
SVM	**0.999**	**0.982**
RF	**0.999**	**0.982**
DT	**0.999**	**0.972**
NB	**0.958**	**0.918**
KNN_less_P	**0.925**	**0.916**

NB:with 3 features and all process data!



The algorithm KNN and RF are better without stop_words.

The algorithm KNN and all other algorithms is better without `fix_contractions(email)` then with.also removing stop_words does not effect.

NB is better when not using `fix_contractions(email)`,also removing stop_words does not effect.
Also, the same result when using 1 feature "message" ,and when using three features "message", "key_words", "contains_money_keywords".

The other algorithms is better when using 3 features than one.

4. Experiment Setup

In this section, the dataset is described as well as the settings and evaluation techniques are used in the experiments have been discussed. The prediction for the email category is tested twice— the first time on an unbalanced data set and the second time on a balanced dataset as below.

- Experiments on the unbalanced dataset: Decision Tree, Naïve Bayes, Random Forest, K-NN, LR, and Random Tree classifiers were applied.
- Experiments on the balanced dataset: In this experiment, the challenges related to unbalanced datasets were tackled by manual procedures to avoid biased predictions and misleading accuracy. The majority class in each dataset almost equalized with the minority classes, i.e., many positive, negative, and neutral, practically the same in the balanced dataset.

4.1. Dataset Description We obtained a dataset from Kaggle, one of the largest online data science communities in this work. It consists of more than 7142 emails, labeled either (ham or spam).

4.2. Dataset Cleansing

Summary for whole code

Data Loading: The code loads email data from a CSV file. It uses the Pandas library to read the data into a DataFrame.

Data Preprocessing:

- The code handles missing values by replacing them with empty strings.
- A list of specific keywords associated with spam emails is defined.

Text Preprocessing:

- Various text preprocessing functions are defined to clean and prepare the email text. These include converting text to lowercase, handling contractions, removing special characters, replacing URLs, and addressing word and punctuation repetition.

4.3. Dataset Training Each of the datasets was divided into two-part. The first part contains 80% of the total number of emails of the data set, and it is used to train the machine to classify the data under one attribute, which is used to classify the emails to either (ham or spam). The remaining 20% of emails were used to classify emails attribute to (ham or spam), i.e., test set.

Also using `random_state=3 for get random emails from data set, stratify=Y for get equal numbers emails that classify 'ham' or 'spam'. That mean to be balance ,when training.`

Unbalanced data set, more spam emails than ham

ıser has requested an enhancement of the downloaded

ALGORITHMES	**ACC_TRAIN**	**ACC_TEST**
KNN	**0.931**	**0.931**
LR	**0.986**	**0.981**
SVM	**0.999**	**0.985**
RF	**0.999**	**0.984**
DT	**0.999**	**0.975**
NB	**0.958**	**0.918**
KNN_less_P	**0.925**	**0.916**

Summary for whole code

Tokenization and Stemming:

- Tokenization and stemming are applied to the email text. The NLTK library is used for tokenization, and stemming is performed using the SnowballStemmer from NLTK.

Feature Engineering:

- Features related to spam keywords and the presence of specific currency symbols are created based on the email content.

TF-IDF Vectorization:

- The code uses the TF-IDF vectorization technique to convert the email text data into numerical feature vectors.

Label Encoding:

- The labels for spam and ham emails are encoded as 0 and 1, respectively.

Data Splitting:

- The dataset is split into training and test sets using the `train_test_split` **function from scikit-learn.**

Model Selection:

- **The code allows the user to select from a menu of classification algorithms, including K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machine (SVM), Random Forest, Decision Tree, and Naive Bayes.**

Model Training and Evaluation:

- **The selected classification model is trained on the training data.**
- **The code evaluates the model's accuracy on both the training and test data.**

Confusion Matrix:

- **A confusion matrix is generated and displayed using seaborn and matplotlib to visualize the model's performance.**

Real-Time Classification:

- **The code allows the user to input an email text and uses the trained model to classify it as spam or ham in real-time.**

ıser has requested an enhancement of the downloaded