Spam Filtering based on Naive Bayes Classification

Shabnam rani May 21, 2018

Abstract

Spam has been a major problem for every email user. the current spam filter only filters out emails that have been previously marked as spam by user. Spammed e-mail may contain many copies of the same message, commercial advertisement or other irrelevant. In previous research, different filtering techniques are used to detect these e-mails such as using Random Forest, Support Vector Machine (SVM) and Neutral Network. In this paper, we test Naive Bayes algorithm for e-mail spam filtering on two datasets and test its performance, i.e., Spam Data from Kaggle datasets. The performance of the dataset is evaluated based on their accuracy. The result shows that the type of email as spam or not with 99.x test accuracy.

Introduction

E-mail service is one of the most popular Internet communication services. Thousands of companies, organizations and individuals use e-mail every day and get benefit from it. However, the amount of spam emails always hang around us and bring down our productivity. We urgently need a spam filtering to clean up our network environment.

We have focused on spam filtering using Naive Bayes Classifier. Instead of focusing on increasing spam precision rate, we try to preserve all non-spam emails as the first priority. In the real world applications and services, that is what we should do.

Stop words are the words we want to filter out before training the classifier. They are high frequency words that are not giving any additional information and doesn't help in labelling.Stopwords actually confuse the classifier.

First of all we have datasets on which we will train our classifier and dataset contains spam emails and ham emails.

Working of naive bays classifier.

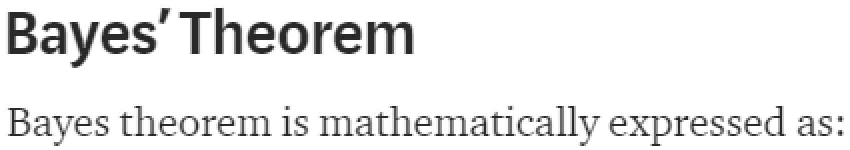
Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

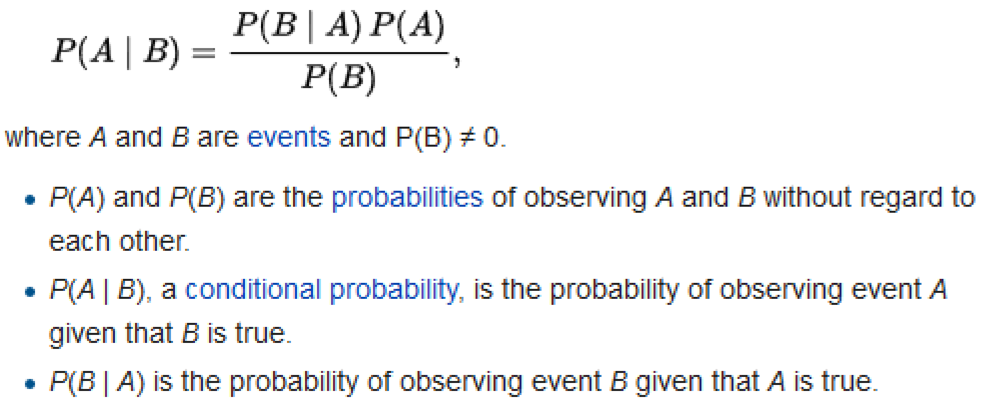
Methodology

The following fomula for naive bays classification is being used in this paper



Bayes' Theorem

Bayes theorem is mathematically expressed as:



$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$

where *A* and *B* are events and P(B) ≠ 0.

- *P*(*A*) and *P*(*B*) are the probabilities of observing *A* and *B* without regard to each other.
- *P*(*A* | *B*), a conditional probability, is the probability of observing event *A* given that *B* is true.
- *P*(*B* | *A*) is the probability of observing event *B* given that *A* is true.

The whole methodology is based on this theorem .After training of the datasets if you give an email as an input to your classifier it would classify it as spam or ham .

Example :

Let's suppose the suspected message contains the word "replica". Most people who are used to receiving e-mail know that this message is likely to be spam, more precisely a proposal to sell counterfeit copies of well-known brands of watches. The spam detection software, however, does not "know" such facts; all it can do is compute probabilities.

- is the probability that a message is a spam, knowing that the word "replica" is in it;
- is the overall probability that any given message is spam;
- is the probability that the word "replica" appears in spam messages;
- is the overall probability that any given message is not spam (is "ham");
- is the probability that the word "replica" appears in ham messages.

Conclusion

The aim of this project is to demonstrate the high performance of a simple statistical filtering method: naive Bayes classification. Given an email it would classify it as spam or ham .

Github Link : https://github.com/shabnamrani31/ML-semProject