AIM:

Build a spam classifier from scratch

detail information relating to the data-set:

I have taken the dataset from Kaggle website

The link of accessing dataset is

https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

the total number of data in this dataset is 5572

feature extracted:

from the dataset I am using two feature v1,v2 representing label and email/sms.

Label is ham or spam. And email/spam is some text

After that I am extracting all the unnecessary things/word, symbols from the email/sms like , . etc and I am making all words which are same but represented differently to one kind of word for example I have converted love, loving, loved to love.

Algorithm Used:

I have implemented Naive Bayes Algorithm to implement the binary classification for the email/sms

Explanation of the data,:

Generally, for the text data, naïve bayes algorithm works best. And our email/sms is also a text data hence I am using this algorithm.

First I read the data file and did preprocessing. It includes the cleaning and analysing the data.

As I mentioned above, I extracted two features label and email/sms from the data set.

and during preprocessing I added few useful feature in dataframe for analying the dataset clearly. Some feature are “number of words”,”number of characters”, “number of sentences” etc.

after analyzing the data I got to know that my dataset contains 87.367% of ham data and 12.633% spam data.

The piechart is shown below in figure 1.0

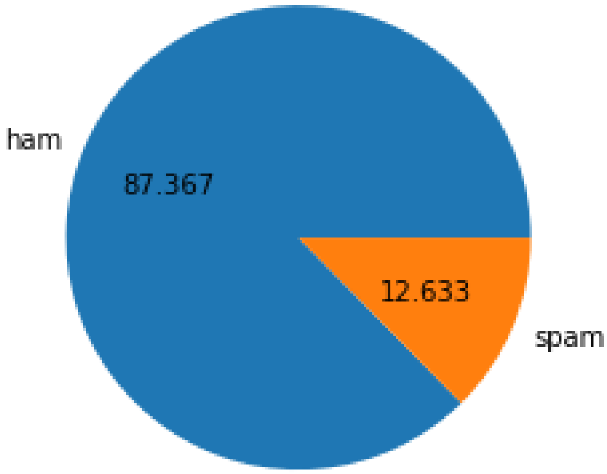

Fig-1.0

some figures representing the relation between the different type of feature are given below in fig 2.0 and fig 3.0

the red is for spam and for non spam it is green

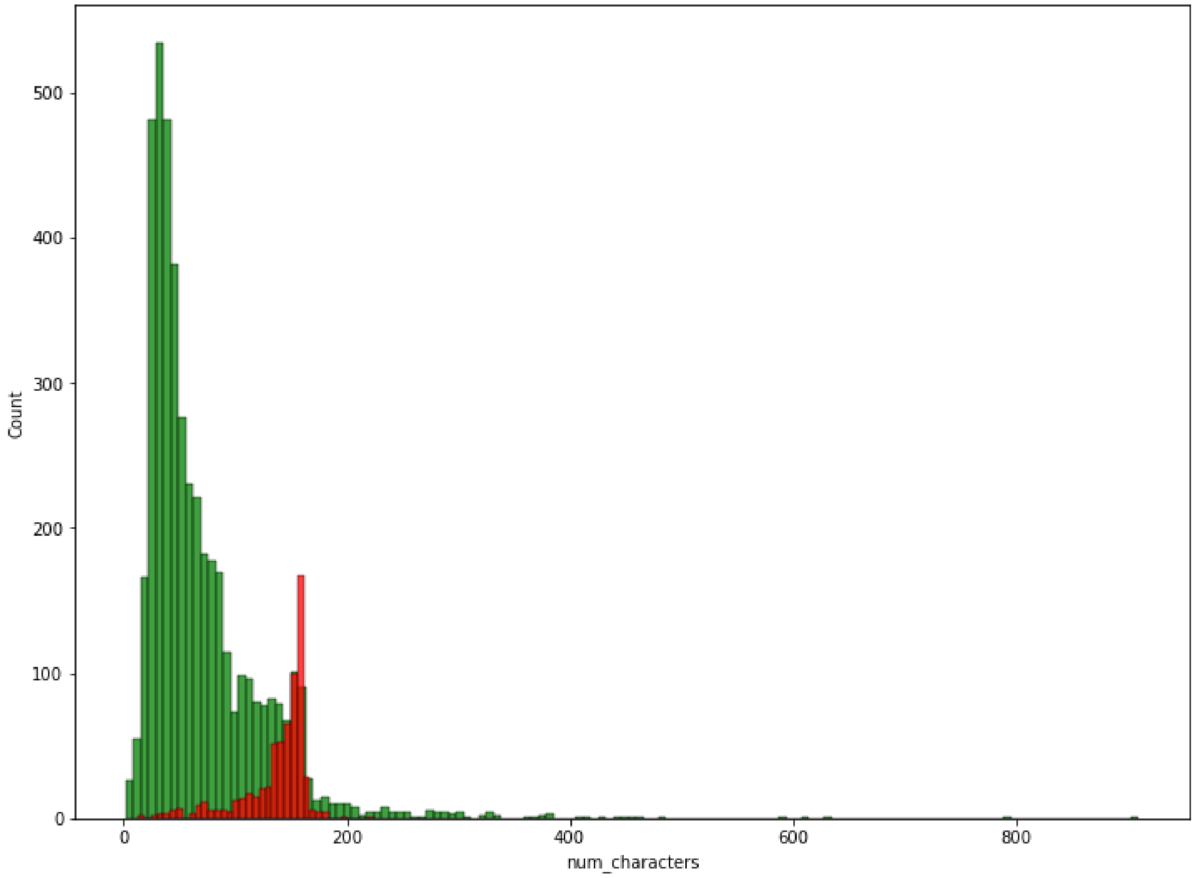

Fig-2.0

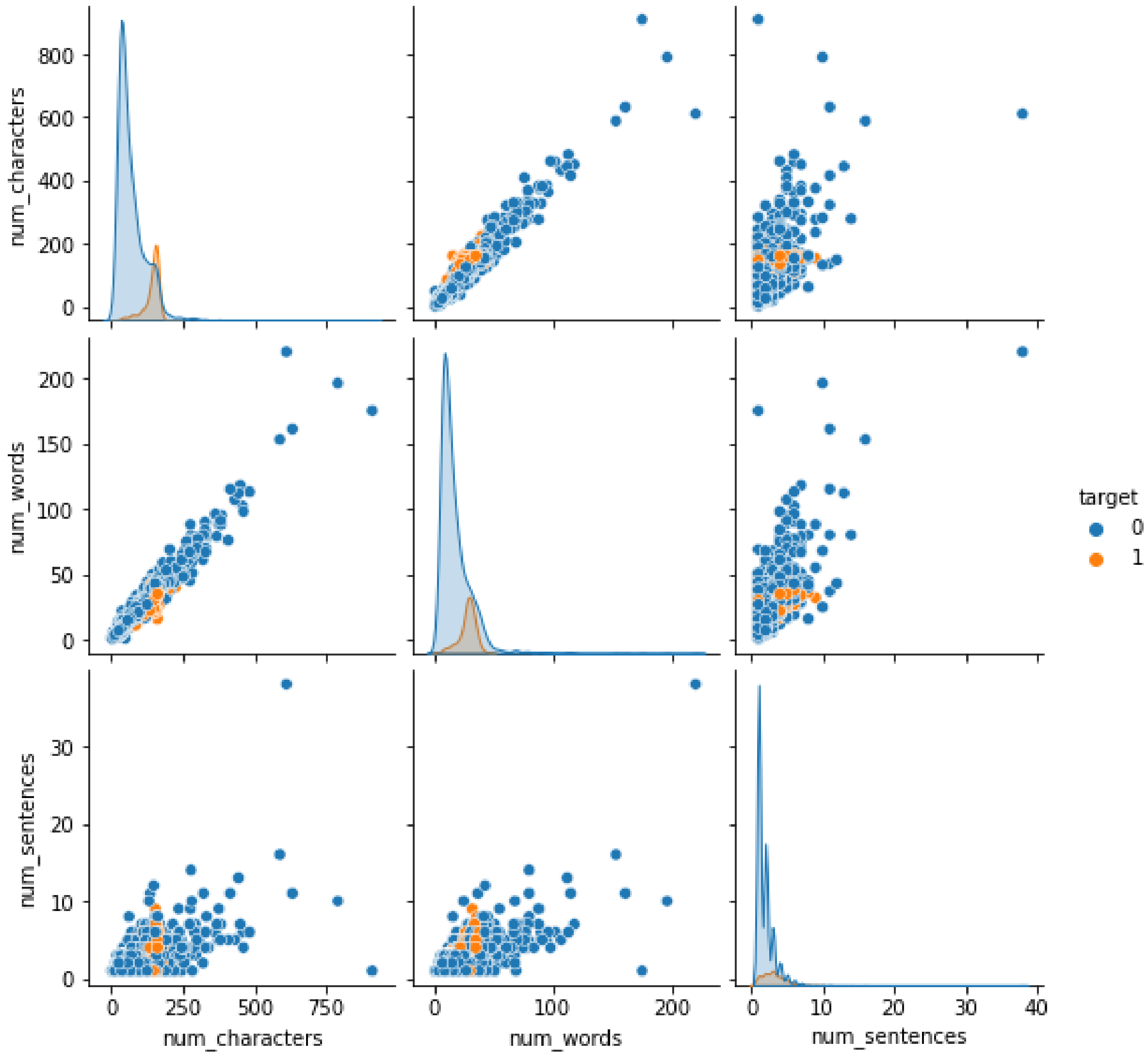

Fig-3.0

Target-1 represents spam emails/sms

Target-0 represents ham email/sms

Explanation of algorithm:

As I mentioned above I have used naïve bayes algorithm to classify emails.

First I have cleaned the data and remove unnecessary things which are not useful for classification.

Then represents each data in vector of 0,1 so that I can run my algorithm efficiently. And I also created a dictionary which contains all the unique words which can be obtained in the email/sms.

And through data I estimated all the necessary parameters and after applying bayes theorem if probability of email/sms being spam is more than probability of email/sms being non spam the the email/sms is classified as spam otherwise non spam

I divided the data into training and test data in the ratio of 85% and 15%

After that I build the model using 85% training data and then test on test data then its giving accuracy of 66%. But if I run my algorithm again and again then this accuracy will vary because each time my training data and test data is randomly assigned.