

Spam detector

Spam/ham mails



July 3, 2022

Hamza muhammad asghar

BCS-183127

# **What is Spam?**

Spam email is unsolicited and unwanted junk email sent out in bulk to an indiscriminate recipient list. Typically, spam is sent for commercial purposes. It can be sent in massive volume by botnets, networks of infected computers.

# Common types of spam

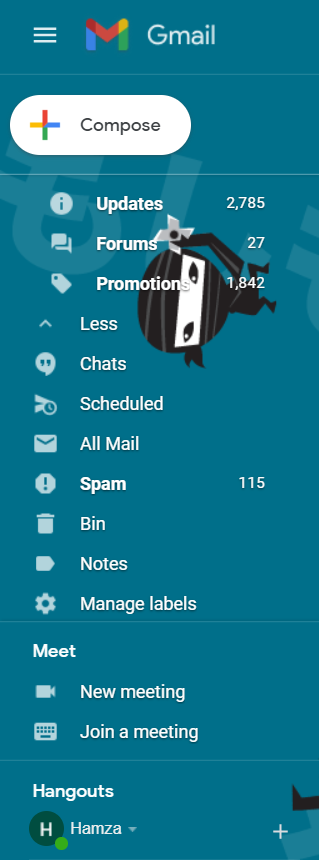
* Commercial advertisements
* Antivirus warnings
* Email spoofing
* Sweepstakes winners
* Money scams

# How do you stop spam email?

This is where the magic of data science comes into place.

Spam email can be difficult to stop, as it can be sent from botnets. Botnets are a network of previously infected computers. As a result, the original spammer can be difficult to trace and stop.

If you receive a message that appears to be spam--for example, if you don’t recognize the sender--mark the message as spam in your email application. Don't click any links or attached files, including opt-out or unsubscribe links. Spammers sometimes include these links to confirm that your email address is legitimate, or the links may trigger malicious webpages or downloads.

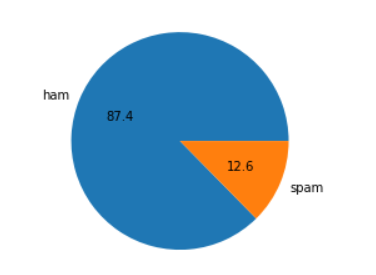
A screen shot of a typical inbox in which the spam mails are filtered and put into Spam folder.

# Data Set

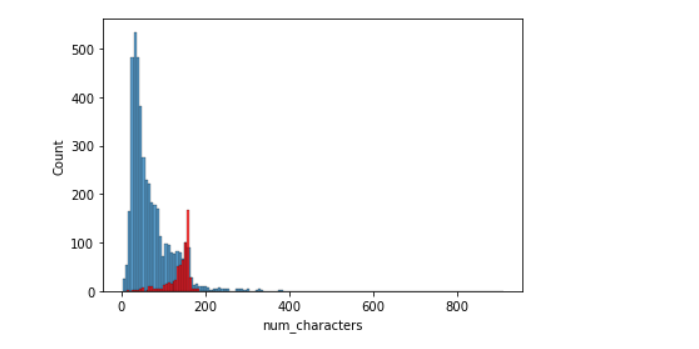
I got my data set from kaggle.com.

I analyzed my data step by step and performed all the steps.

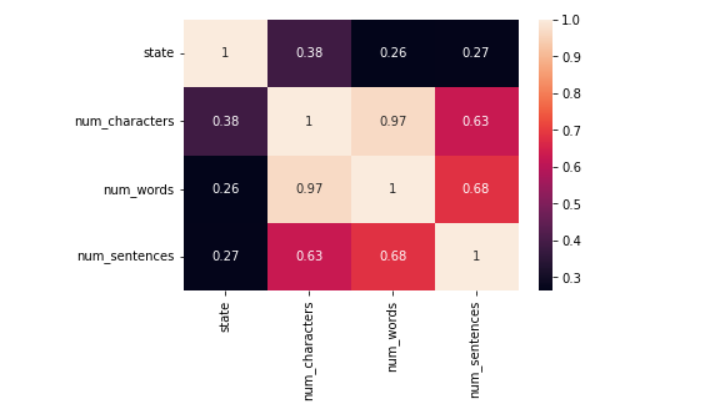
* 1st I got the raw data from kaggle.com
* Then data processing was performed.
* Then data was cleaned for any duplications etc.
* After cleaning the data, I performed exploratory data analysis in which I applied different statistical analysis.
* A good algorithm is the key to success here provided the above steps were done carefully.
* So I picked up Naïve Bayes algorithm and used it in this model.



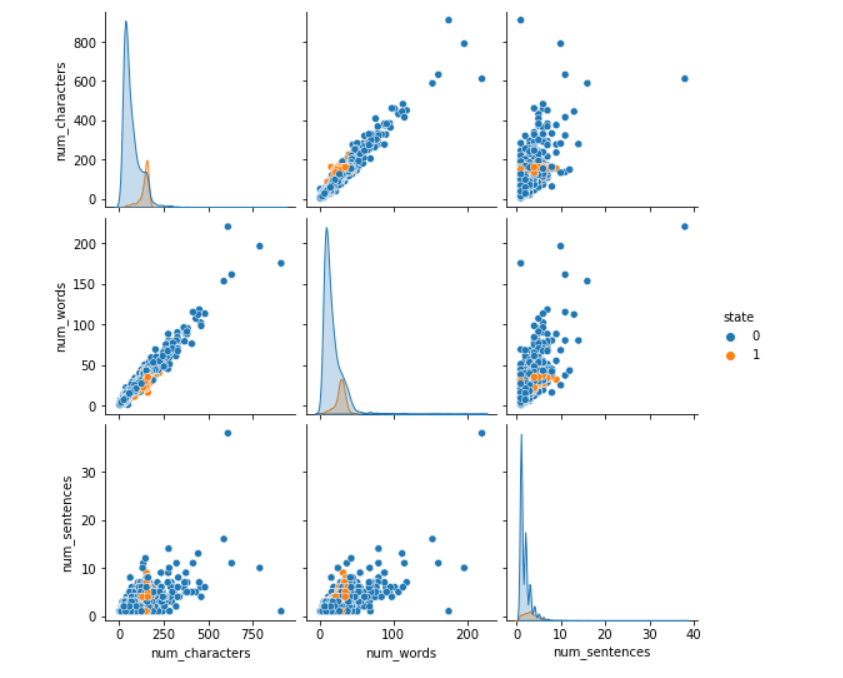
Pie chart of the data set



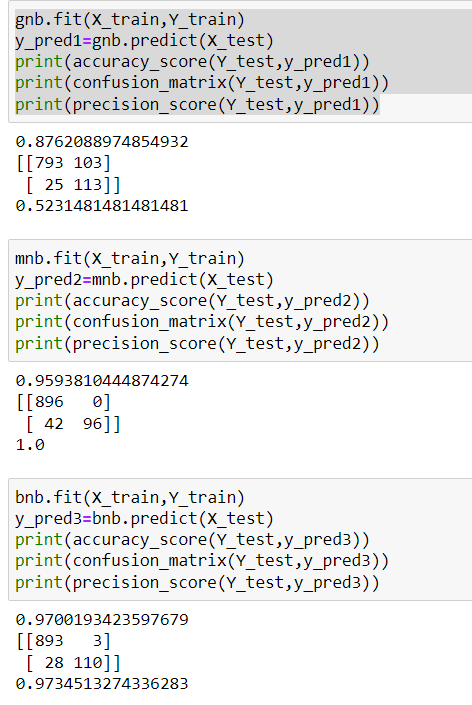
Histogram of the data set



Heat map of the data set



Pair plot of the data set



Training and testing different models for selecting the best.

# **Code**

import numpy as np

df = pd.read\_csv("spam.csv")

df.sample(5)

df.info()

df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)

df.sample(5)

df.rename(columns={'v1':'state','v2':'text'},inplace=True)

df.sample(5)

from sklearn.preprocessing import LabelEncoder

encoder=LabelEncoder()

df.isnull().sum()

df.duplicated().sum()

df=df.drop\_duplicates(keep='first')

df['state'].value\_counts()

import nltk

df['num\_characters']=df['text'].apply(len)

df.info()

df['num\_words']=df['text'].apply(lambda x:len(nltk.word\_tokenize(x)))

df['num\_sentences']=df['text'].apply(lambda x:len(nltk.sent\_tokenize(x)))

df[['num\_characters','num\_words','num\_sentences']].describe()

df[df['state']==0][['num\_characters','num\_words','num\_sentences']].describe()

df[df['state']==1][['num\_characters','num\_words','num\_sentences']].describe()

def transform\_text(text):

text=text.lower()

text=nltk.word\_tokenize(text)

y=[]

for i in text:

if i.isalnum():

y.append(i)

text = y[:]

y.clear()

for i in text:

if i not in stopwords.words('english') and i not in string.punctuation:

y.append(i)

text=y[:]

y.clear()

for i in text:

y.append(ps.stem(i))

return " ".join(y)

from nltk.stem.porter import PorterStemmer

ps=PorterStemmer()

df['transformed\_text']=df['text'].apply(transform\_text)

from sklearn.feature\_extraction.text import CountVectorizer,TfidfVectorizer

cv = CountVectorizer()

tfidf= TfidfVectorizer()

X=tfidf.fit\_transform(df['transformed\_text']).toarray()

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(X,Y,test\_size=0.2,random\_state=2)

from sklearn.naive\_bayes import GaussianNB,MultinomialNB,BernoulliNB

from sklearn.metrics import accuracy\_score,confusion\_matrix,precision\_score

gnb = GaussianNB()

mnb = MultinomialNB()

bnb = BernoulliNB()

gnb.fit(X\_train,Y\_train)

y\_pred1=gnb.predict(X\_test)

print(accuracy\_score(Y\_test,y\_pred1))

print(confusion\_matrix(Y\_test,y\_pred1))

print(precision\_score(Y\_test,y\_pred1))

mnb.fit(X\_train,Y\_train)

y\_pred2=mnb.predict(X\_test)

print(accuracy\_score(Y\_test,y\_pred2))

print(confusion\_matrix(Y\_test,y\_pred2))

print(precision\_score(Y\_test,y\_pred2))

bnb.fit(X\_train,Y\_train)

y\_pred3=bnb.predict(X\_test)

print(accuracy\_score(Y\_test,y\_pred3))

print(confusion\_matrix(Y\_test,y\_pred3))

print(precision\_score(Y\_test,y\_pred3))

In terminal we have given a command *streamlit run website.py* which takes us to a local server and gives us a GUI to test out the different messages to check whether they are spam or not.

# **Screenshot of detection**

