**Email Spam Classifier:**

Downloaded the dataset from Kaggle named as spam\_ham\_dataset

**Step 1: Data Cleaning:**

1]Checking Null values:

{As there no null values in our dataset and it is a crucial step to remove null values because Models like CountVectorizer/TfidfVectorizer **cannot process NULL values**If we had null values we will be removing it by using three steps  
1)Remove rows with null text (df = df.dropna(subset=['message'])

2)Replace null with empty string(df['message'] = df['message'].fillna("")

3)Replace null with a placeholder token(df['message'] = df['message'].fillna("no content"))}

2]Dropping unnecessary columns:

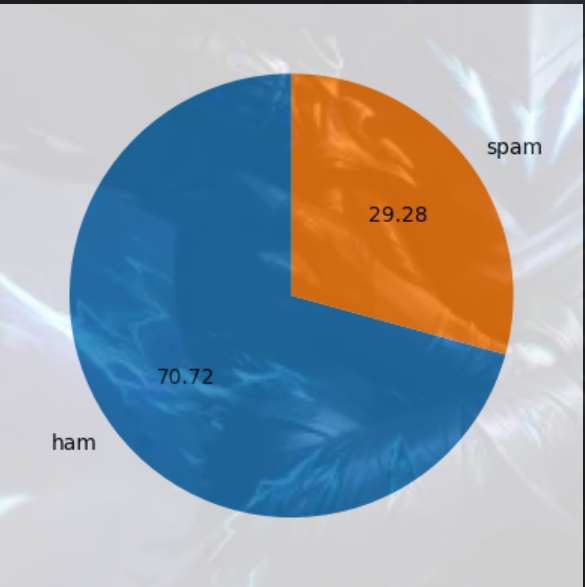
(df=df.drop(columns=['Unnamed: 0'])

3]Renaming column name for better understanding

4]Check for duplicated values:  
if found duplicate values drop the row

**Step 2:Exploratory Data Analysis:**

1]Counting Spam and ham messages and also plotting pie chart for it



2]Importing nltk library and download punkt(pre trained tokenizer) from nltk

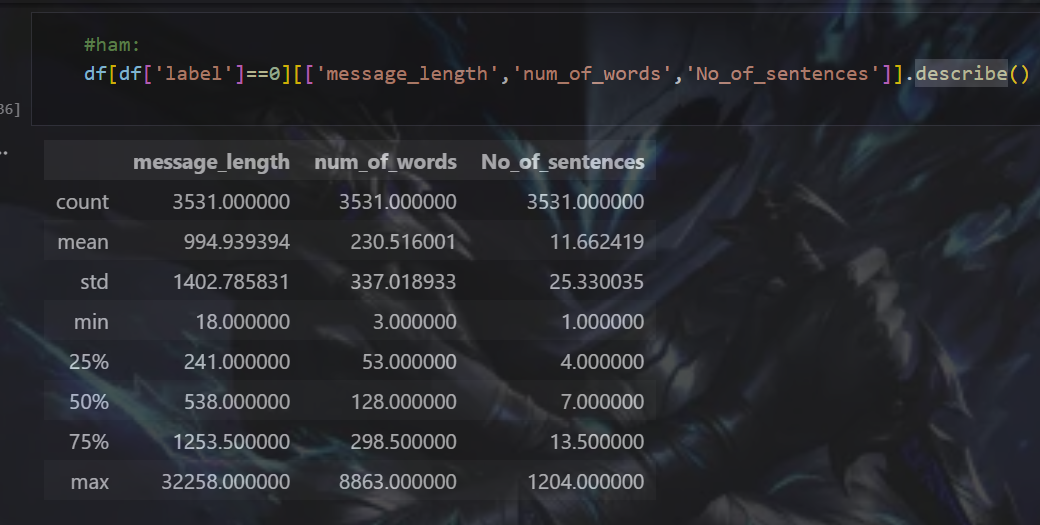
3]Counting length of each message and sort ascending=False

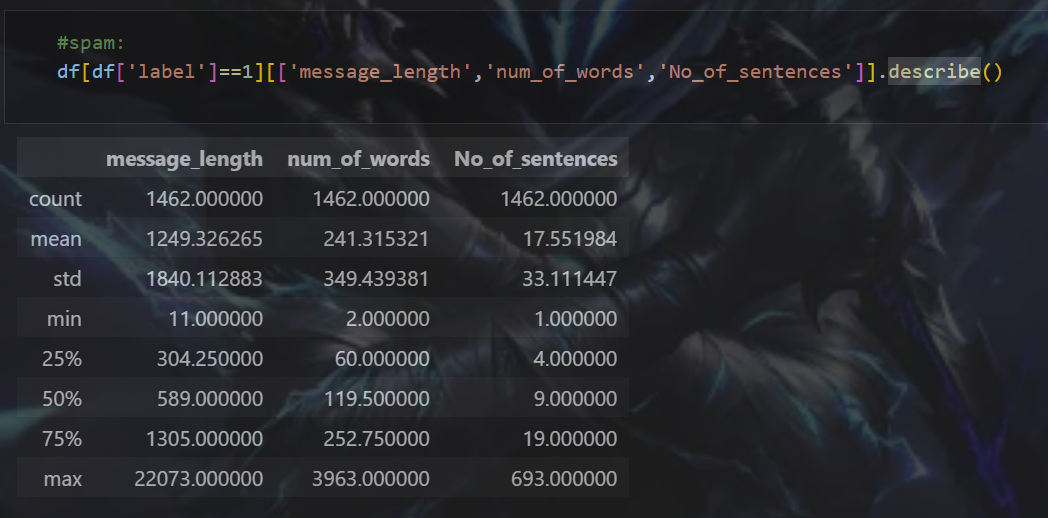


It is calculating number of rows in that particular column

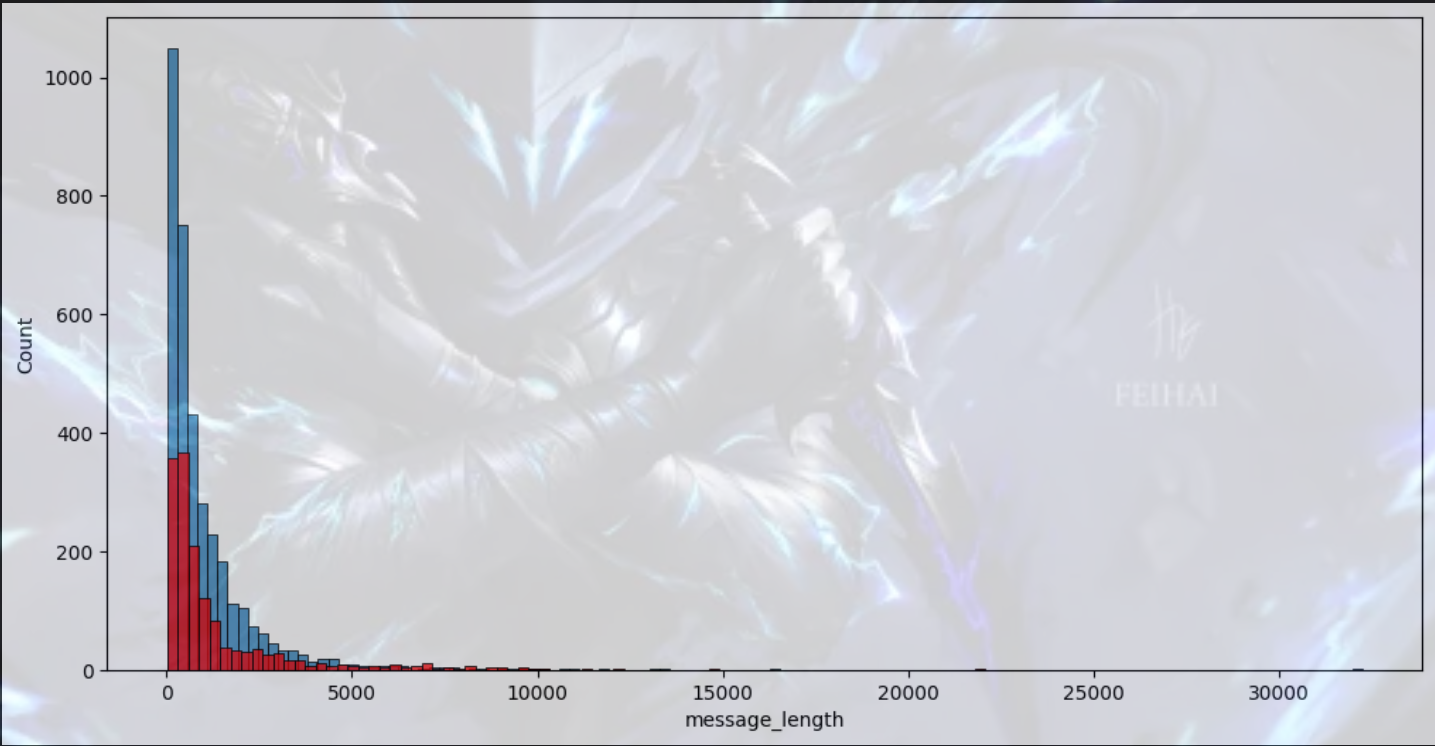
4]Splitting message into words the calculate num of words

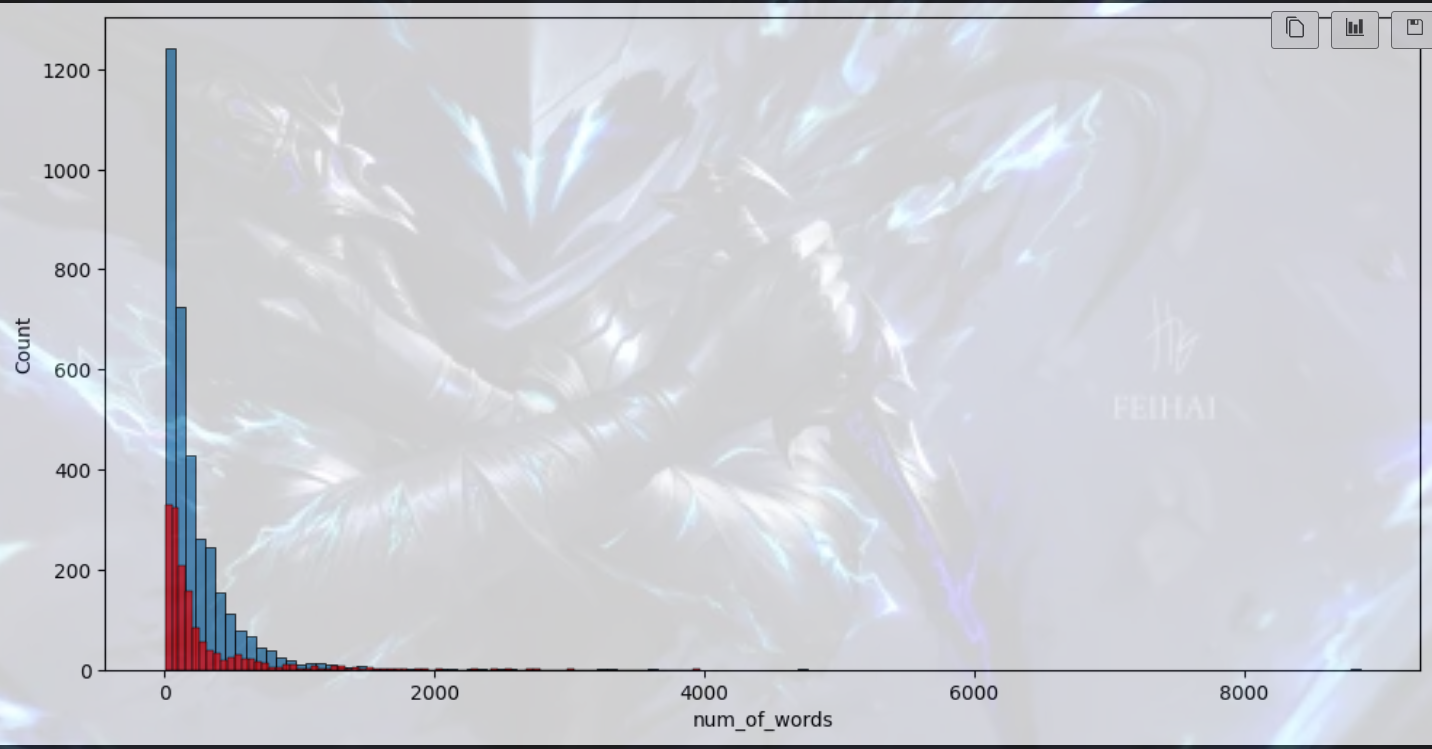
5]Splitting into sentences and calculate the no of sentences





Also we can plot histogram for all the comparisons



****

**Step 3: Data Preprocessing:**

**1) Lower case:**

.lower()

**2)Tokenization:**

nltk.word\_tokenize()

**3)Removing special characters:**

i.isalnum()

**4)Removing stop words and punctuation:**

nltk.download('stopwords')

from nltk.corpus import stopwords

stopwords.words('english')

import string

string.punctuation

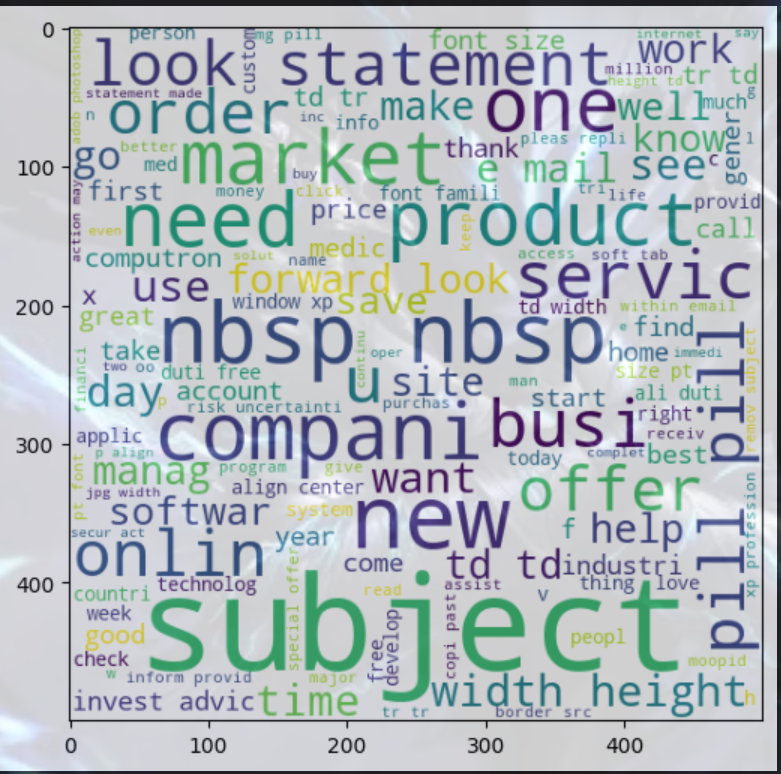
**5)stemming:**

from nltk.stem.porter import PorterStemmer

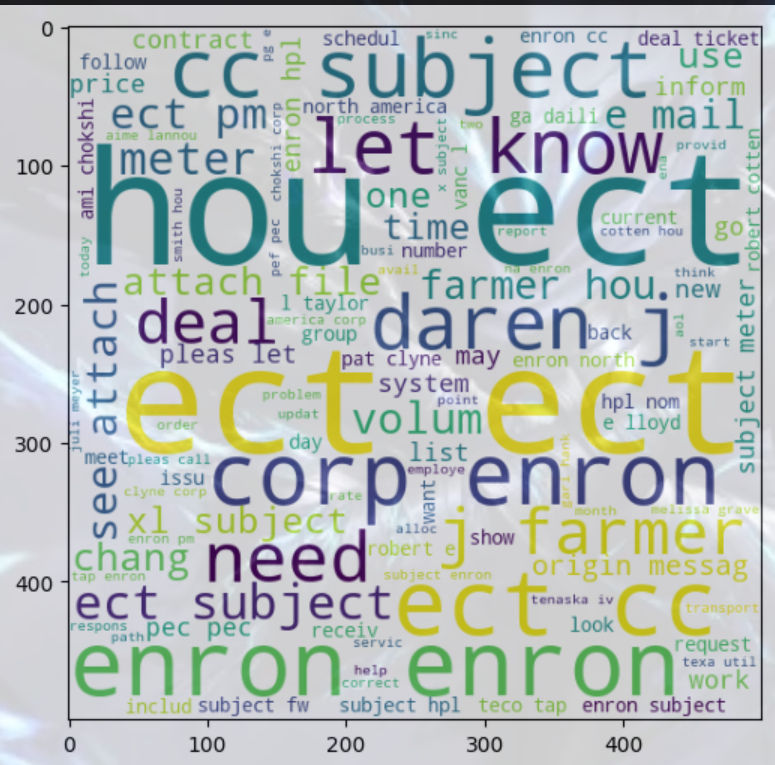
Representing frequently occurring words using wordcloud for spam and ham:

from wordcloud import WordCloud

Spam:

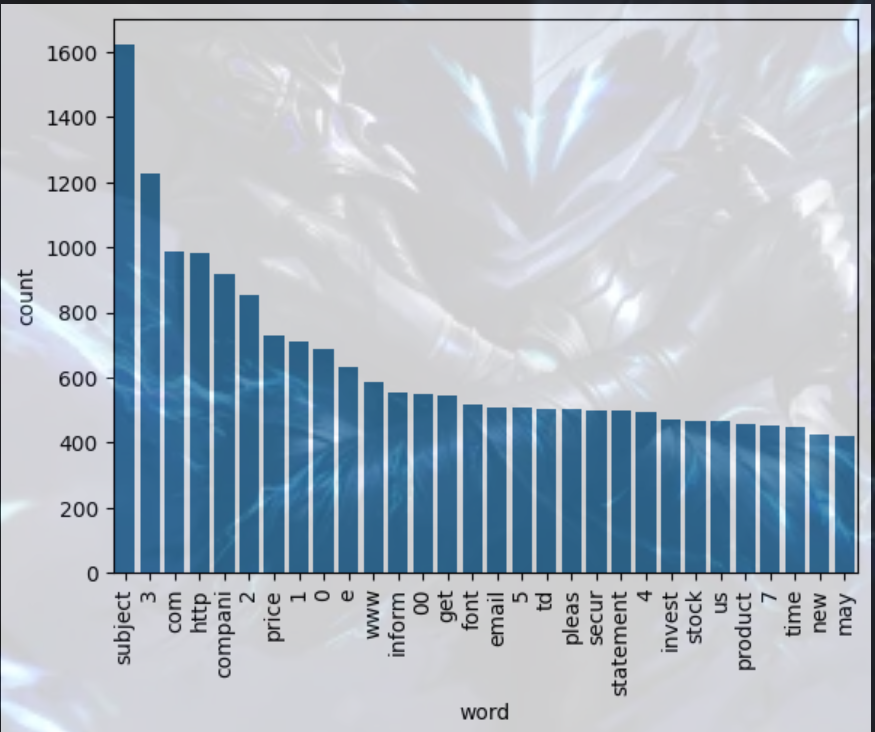


Ham:

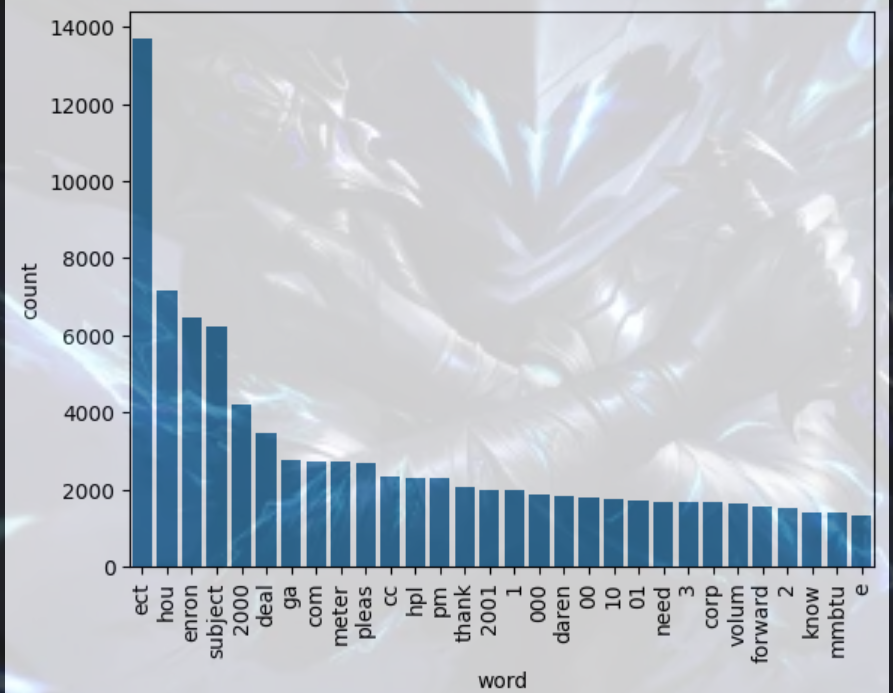


**spam corpus:(All spam words in a one single list)**

from collections import Counter



**Ham Corpus: (All ham words in a one single list)**

****

**Step 4: Model Building:**

#So we need number input to our model because any ml model takes number input and now we need to convert transformed message into numbers (vectorized it using bag of words)

1)Installing scikit learn:

Feature extraction and vectorization

Array is created with numerical data and stored in X

Y stores the label 0 and 1 from the col of dataset which is used for training the dataset

2)Count vectorizer is used for vectorization:

Now time for training and splitting data

3)from sklearn.model\_selection import train\_test\_split:

80% data-> Training

20% data->Testing

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

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

mnb = MultinomialNB()

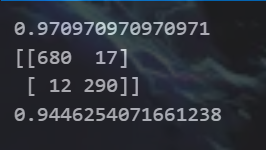
mnb.fit(X\_train, Y\_train)

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

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

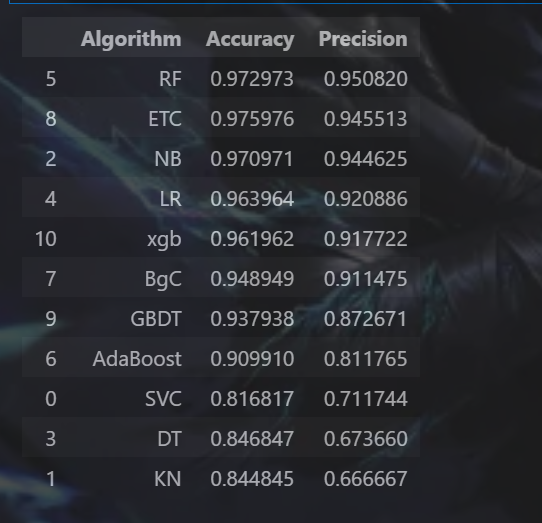
print(confusion\_matrix(Y\_test , Y\_pred2))

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



4)cv+mnb was the best but also tried other algorithm what if they works best

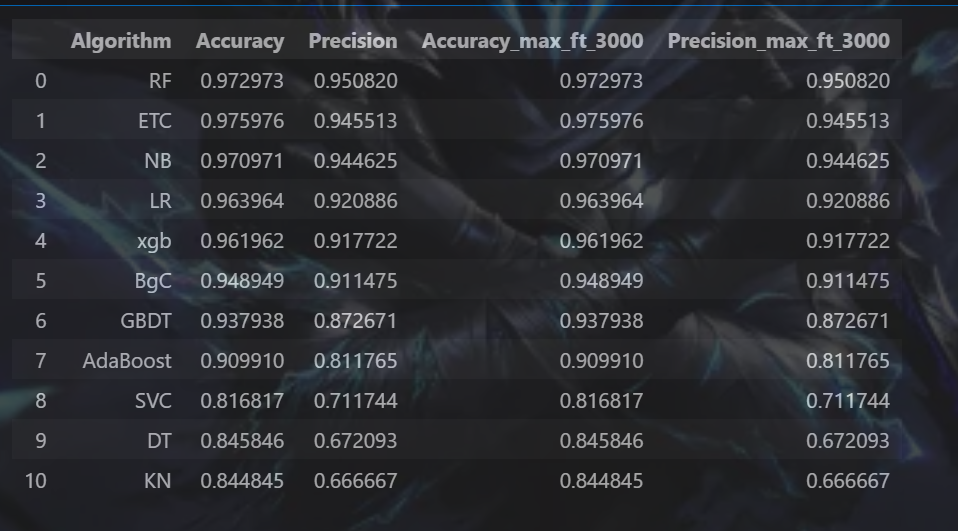
Comparing different ml models to check their accuracy and precision with our dataset



But still cv+mnb remains best

**Step 5: Improving model performance**

For improving accuracy and precision I have passed max\_features = 3000 results remain same

  
and also did voting classifier but not much improvement so I will not go with voting classifier  
then tried stacking but yet no improvements

Every Python file should import its own libraries.

**Step 6:Creating app.py:**

import streamlit as st

import pickle

import string

import nltk

nltk.download('punkt')

nltk.download('stopwords')    #For streamlit ;requirements.txt

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

ps= PorterStemmer()

def transform\_text(text):

    text = text.lower()

    text = nltk.word\_tokenize(text)    #not the text is converted to list

    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)

cv = pickle.load(open('vectorizer.pkl','rb'))

model = pickle.load(open('model.pkl','rb'))

st.title("Email Spam Classifier")

input\_email = st.text\_area("Enter the message")

if st.button('Predict'):

    # Check for empty input

    if input\_email.strip() == "":

        st.warning("Please enter a message!")

    else:

        # 1. Preprocess

        transformed\_email = transform\_text(input\_email)

        # 2. Vectorize

        vector\_input = cv.transform([transformed\_email])

        # 3. Predict

        result = model.predict(vector\_input)[0]

        # 4. Display

        if result == 1:

            st.header("Spam")

        else:

            st.header("Ham")

**Project is completed**

To run the project: In vs terminal type(streamlit run app.py)

**Also I have uploaded in github**

For pushing the code in github:(in the terminal of app.py)

git init

git add .

git commit -m "Initial commit"

git remote add origin https://github.com/Mantasha-205/Project\_Repository.git

git push -u origin main

Now my code is on github

If any changes made in code or ipynb file we need to again add it to git , commit and push  
and now we deploy in streamlit cloud so that it would be publicly available

**My project link:**

https://mantasha-205-project-repository-app-gugb0v.streamlit.app/

My model works correct for following mails:  
**✅ HAM (Not Spam) – should predict "ham"**

**1️⃣**

**Subject:** Meeting Reminder  
“Hey team, just a reminder that our project meeting is scheduled for 3 PM tomorrow. Please bring your progress updates.”

**2️⃣**

**Subject:** Invoice Attached  
“Dear John,  
Your invoice for the month of October is attached. Let me know if you have any questions.  
Regards, Finance Team”

**3️⃣**

**Subject:** Schedule Update  
“Hi, please note that your dentist appointment is confirmed for Monday at 11 AM.”

**4️⃣**

**Subject:** Family Dinner  
“Hey, we’re planning a family dinner this Sunday. Let me know if you can join!”

**5️⃣**

**Subject:** Delivery Notification  
“Your package has been shipped and is expected to arrive on Wednesday. Track your order through our official website.”

**🚨 SPAM Emails – should predict "spam"**

**6️⃣**

**Subject:** Congratulations! You Won $5,000  
“You have been selected for a cash prize of $5000! Click the link to claim your reward immediately!”

**7️⃣**

**Subject:** Your Account Will Be Closed  
“Your bank account will be suspended in 24 hours. Verify your identity by clicking the link below.”

**8️⃣**

**Subject:** Urgent: Update Required  
“Your device has been infected with 3 viruses. Install the antivirus now to protect your data!”

**9️⃣**

**Subject:** Limited Time Offer  
“Buy 1 get 3 free!!! Visit our website now to grab the unbelievable offer before it ends!”

**🔟**

**Subject:** Earn Money Fast  
“Work from home and earn $5000 per week with zero experience. Sign up now!”

To host the web:

streamlit run "c:/Email Spam Classifier/app.py"