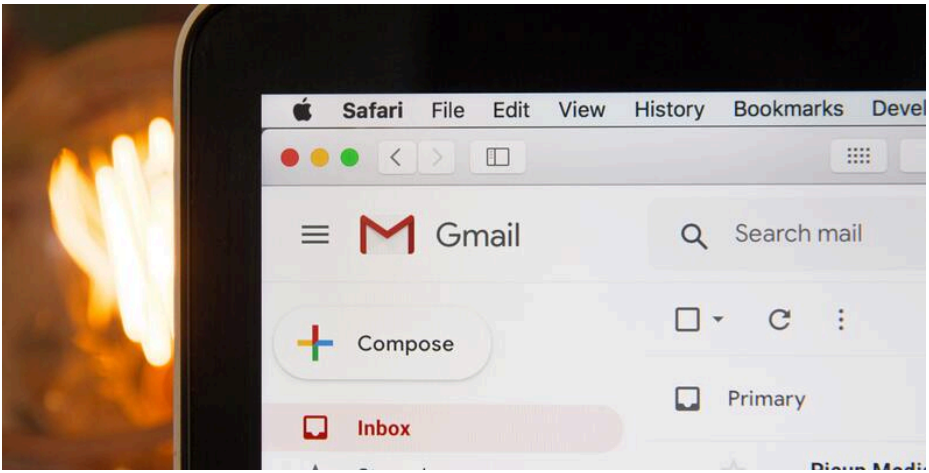


Import Libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import re
df=pd.read_csv('/content/spam.csv')
df
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will ü b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...

Next steps: Generate code with df View recommended plots

```
df.head()
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then sav...

Next steps: Generate code with df View recommended plots

```
df.head()
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then sav...

Next steps: Generate code with df View recommended plots

```
df.dtypes
```

```
Category    object
Message     object
dtype: object
```

```
df.isna().sum()
```

```
Category    0
Message     0
dtype: int64
```

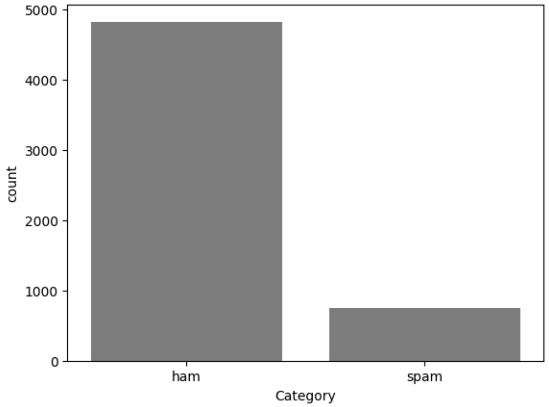
plotting Distribution of spam vs. ham

```
df['Category'].value_counts()
```

```
ham     4825
spam     747
Name: Category, dtype: int64
```

```
sns.countplot(x='Category',data=df,color='grey')
```

```
<Axes: xlabel='Category', ylabel='count'>
```




```
df['Category']=df['Category'].map({'ham':0,'spam':1})
df
```

	Category	Message
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	1	This is the 2nd time we have tried 2 contact u...
5568	0	Will ü b going to esplanade fr home?
5569	0	Pity, * was in mood for that. So...any other s...

Next steps: Generate code with df View recommended plots

Preprocessing

```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
True
```

```
tweets=df.Message
tweets
```

```
0       Go until jurong point, crazy.. Available only ...
1                           Ok lar... Joking wif u oni...
2       Free entry in 2 a wkly comp to win FA Cup fina...
3       U dun say so early hor... U c already then say...
4       Nah I don't think he goes to usf, he lives aro...
                              ...
5567    This is the 2nd time we have tried 2 contact u...
5568                 Will ü b going to esplanade fr home?
5569    Pity, * was in mood for that. So...any other s...
5570    The guy did some bitching but I acted like i'd...
5571                          Rofl. Its true to its name
Name: Message, Length: 5572, dtype: object
```

```
from nltk import TweetTokenizer
tk=TweetTokenizer()
tweets=tweets.apply(lambda x:tk.tokenize(x)).apply(lambda x:" ".join(x))
tweets
```

```
0       Go until jurong point , crazy .. Available onl...
1                         Ok lar ... Joking wif u oni ...
2       Free entry in 2 a wkly comp to win FA Cup fina...
3       U dun say so early hor ... U c already then sa...
4       Nah I don't think he goes to usf , he lives ar...
                              ...
5567    This is the 2nd time we have tried 2 contact u...
5568               Will ü b going to esplanade fr home ?
5569    Pity , * was in mood for that . So ... any oth...
5570    The guy did some bitching but I acted like i'd...
5571                           Rofl . Its true to its name
Name: Message, Length: 5572, dtype: object
```

```
tweets=tweets.str.replace('[^a-zA-Z0-9]+',' ')
tweets
```

```
<ipython-input-59-243a49c37bfd>:1: FutureWarning: The default value of regex will char
  tweets=tweets.str.replace('[^a-zA-Z0-9]+',' ')
0       Go until jurong point crazy Available only in ...
1                           Ok lar Joking wif u oni
2       Free entry in 2 a wkly comp to win FA Cup fina...
3            U dun say so early hor U c already then say
4       Nah I don t think he goes to usf he lives arou...
                            ...
5567    This is the 2nd time we have tried 2 contact u...
5568                 Will b going to esplanade fr home
5569    Pity was in mood for that So any other suggest...
5570    The guy did some bitching but I acted like i d...
5571                          Rofl Its true to its name
Name: Message, Length: 5572, dtype: object
```

```
from nltk.tokenize import word_tokenize
tweets=tweets.apply(lambda x:' '.join([w for w in word_tokenize(x) if len(w)>=3]))
tweets
```

```
0       until jurong point crazy Available only bugis ...
1                                 lar Joking wif oni
2       Free entry wkly comp win Cup final tkts 21st M...
3                    dun say early hor already then say
4         Nah don think goes usf lives around here though
                            ...
5567    This the 2nd time have tried contact have won ...
5568                         Will going esplanade home
5569        Pity was mood for that any other suggestions
5570    The guy did some bitching but acted like inter...
5571                           Rofl Its true its name
Name: Message, Length: 5572, dtype: object
```

```
from nltk.stem import SnowballStemmer
stemmer=SnowballStemmer('english')
tweets=tweets.apply(lambda x:[stemmer.stem(i.lower()) for i in tk.tokenize(x)]).apply(lamb
tweets
```

```
0       until jurong point crazi avail onli bugi great...
1                                   lar joke wif oni
2       free entri wkli comp win cup final tkts 21st m...
3                    dun say earli hor alreadi then say
4           nah don think goe usf live around here though
                            ...
5567    this the 2nd time have tri contact have won th...
5568                           will go esplanad home
5569             piti was mood for that ani other suggest
5570    the guy did some bitch but act like interest b...
5571                            rofl it true it name
Name: Message, Length: 5572, dtype: object
```

```
from nltk.corpus import stopwords
sw=stopwords.words('english')
tweets=tweets.apply(lambda x:[i for i in tk.tokenize(x) if i not in sw]).apply(lambda x:'
tweets
```

```
0       jurong point crazi avail onli bugi great world...
1                                   lar joke wif oni
2       free entri wkli comp win cup final tkts 21st m...
3                           dun say earli hor alreadi say
4                     nah think goe usf live around though
                              ...
5567    2nd time tri contact 750 pound prize claim eas...
5568                                  go esplanad home
5569                                piti mood ani suggest
5570    guy bitch act like interest buy someth els nex...
5571                                    rofl true name
Name: Message, Length: 5572, dtype: object
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer()
train_data=vec.fit_transform(tweets)
```

```
print(train_data)
```

```
  (0, 6579)     0.20216031597468262
  (0, 999)      0.36596689778277197
  (0, 2952)     0.16965264899544422
  (0, 1758)     0.3091703141145051
  (0, 1531)     0.3493549619752681
  (0, 6747)     0.24533301256575607
  (0, 2989)     0.2017861574399963
  (0, 1533)     0.3091703141145051
  (0, 4474)     0.1750991615757075
  (0, 1183)     0.2737313043998896
  (0, 1957)     0.2834161703187832
  (0, 4767)     0.24988993515576846
  (0, 3543)     0.36596689778277197
  (1, 4472)     0.5884936620961707
  (1, 6673)     0.46469076328470776
  (1, 3512)     0.494457003144156
  (1, 3675)     0.4396020657733696
  (2, 70)       0.26961406544295236
  (2, 1070)     0.1919355645505515
  (2, 5025)     0.18542681454450735
  (2, 6321)     0.14233815390204066
  (2, 5773)     0.22578140384429926
  (2, 4982)     0.1893353125177682
  (2, 5062)     0.18542681454450735
  (2, 766)      0.25346248369348356
  :     :
  (5567, 4650)  0.23275121270305701
  (5567, 5025)  0.24602192076488918
  (5567, 4474)  0.17929202107361064
  (5568, 2473)  0.782681381846728
  (5568, 2914)  0.43953417232463426
  (5568, 3189)  0.4407034897404325
  (5569, 4721)  0.6038288697475673
  (5569, 4147)  0.5021550803083344
  (5569, 5875)  0.5295640248979575
  (5569, 1016)  0.3206133417588562
  (5570, 1378)  0.3680669195553694
  (5570, 869)   0.37545486243003046
  (5570, 3393)  0.3284468927008843
  (5570, 3031)  0.26276106116829073
```

```
  (5570, 2851)  0.34640473630614693
  (5570, 2402)  0.30705779715092285
  (5570, 1558)  0.2570316492387874
  (5570, 5631)  0.25840107844655774
  (5570, 4307)  0.264307219819181
  (5570, 3752)  0.19826108170693374
  (5570, 6612)  0.22415005603621865
  (5570, 2759)  0.20028360803386733
  (5571, 5198)  0.6981587787831502
  (5571, 6288)  0.5274275139134303
  (5571, 4235)  0.4841430957631416
```

```
x=train_data
x
```

```
<5572x6885 sparse matrix of type '<class 'numpy.float64'>'
        with 44122 stored elements in Compressed Sparse Row format>
```

```
y=df['Category']
y
```

```
0       0
1       0
2       1
3       0
4       0
       ..
5567    1
5568    0
5569    0
5570    0
5571    0
Name: Category, Length: 5572, dtype: int64
```

splitting training and testing data

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

```
<3900x6885 sparse matrix of type '<class 'numpy.float64'>'
        with 31062 stored elements in Compressed Sparse Row format>
```

model creation

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import BernoulliNB
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix,accuracy_score
from sklearn.metrics import classification_report
k_model=KNeighborsClassifier(n_neighbors=7)
n_model=BernoulliNB()
s_model=SVC()
r_model=RandomForestClassifier()
d_model=DecisionTreeClassifier(criterion='entropy')
lst_model=[k_model,n_model,s_model,r_model,d_model]
```

```
for i in lst_model:
  print('model is',i)
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  print("*"*100)
  print(confusion_matrix(y_test,y_pred))
  print("Accuracy score is",accuracy_score(y_test,y_pred))
  print(".......classification Report........")
  print(classification_report(y_test,y_pred))
```

```
model is KNeighborsClassifier(n_neighbors=7)
****************************************************************************************************
[[1448    0]
 [ 181   43]]
Accuracy score is 0.8917464114832536
.......classification Report........
              precision    recall  f1-score   support

           0       0.89      1.00      0.94      1448
           1       1.00      0.19      0.32       224

    accuracy                           0.89      1672
   macro avg       0.94      0.60      0.63      1672
weighted avg       0.90      0.89      0.86      1672

model is BernoulliNB()
****************************************************************************************************
[[1445    3]
 [  33  191]]
Accuracy score is 0.9784688995215312
.......classification Report........
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1448
           1       0.98      0.85      0.91       224

    accuracy                           0.98      1672
   macro avg       0.98      0.93      0.95      1672
weighted avg       0.98      0.98      0.98      1672

model is SVC()
****************************************************************************************************
[[1447    1]
 [  43  181]]
Accuracy score is 0.9736842105263158
```

```
.......classification Report........
              precision    recall  f1-score   support

           0       0.97      1.00      0.99      1448
           1       0.99      0.81      0.89       224

    accuracy                           0.97      1672
   macro avg       0.98      0.90      0.94      1672
weighted avg       0.97      0.97      0.97      1672

model is RandomForestClassifier()
**********************************************************************************
[[1448    0]
 [  40  184]]
Accuracy score is 0.9760765550239234
.......classification Report........
              precision    recall  f1-score   support

           0       0.97      1.00      0.99      1448
           1       1.00      0.82      0.90       224

    accuracy                           0.98      1672
```