```
Import Dependencies
```

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
import nltk
import re
from nltk.stem import WordNetLemmatizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
```

 $\label{lem:df_train} $$ df_train=pd.read_csv("/content/SMS_train (1).csv",encoding='ISO-8859-1') $$ df_train $$ $$$

1	Label	Message_body	S. No.	
	Non-Spam	Rofl. Its true to its name	1	0
	Non-Spam	The guy did some bitching but I acted like i'd	2	1
	Non-Spam	Pity, * was in mood for that. Soany other s Non-S		2
	Non-Spam	4 Will ü b going to esplanade fr home?		3
	Spam	5 This is the 2nd time we have tried 2 contact u		4
	Non-Spam	hows my favourite person today? r u workin har Non-Spam		952
	Non-Spam	How much you got for cleaning	954	953
	Non-Spam	55 Sorry da. I gone mad so many pending works wha Non-Spam		954
	Non-Spam	Wat time ü finish? Non-Spa		955
	Non-Spam	Just glad to be talking to you.	957	956

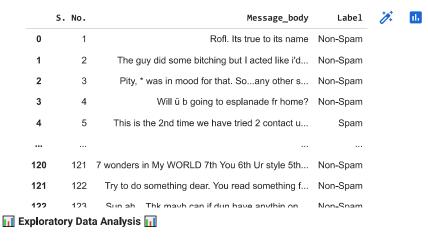
957 rows × 3 columns

 $\label{local_model} $$ df_test=pd.read_csv("/content/SMS_test (2).csv",encoding='ISO-8859-1') $$ df_test $$$

	S. No.	Message_body	Label
0	1	UpgrdCentre Orange customer, you may now claim	Spam
1	2	Loan for any purpose £500 - £75,000. Homeowner	Spam
2	3	Congrats! Nokia 3650 video camera phone is you	Spam
3	4	URGENT! Your Mobile number has been awarded wi	Spam
4	5	Someone has contacted our dating service and e	Spam
120	121	7 wonders in My WORLD 7th You 6th Ur style 5th	Non-Spam
121	122	Try to do something dear. You read something f	Non-Spam
122	123	Sun ah Thk mayb can if dun have anythin on	Non-Spam
123	124	SYMPTOMS when U are in love: "1.U like listeni	Non-Spam
124	125	Great. Have a safe trip. Dont panic surrender	Non-Spam

125 rows × 3 columns

df=pd.concat([df_train,df_test],axis=0)
df



df.head()

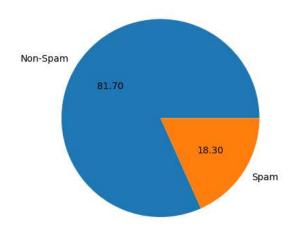
	Message_body	Label	1	ılı
0	Rofl. Its true to its name	0		
1	The guy did some bitching but I acted like i'd	0		
2	Pity, * was in mood for that. Soany other s	0		
3	Will ü b going to esplanade fr home?	0		
4	This is the 2nd time we have tried 2 contact u	1		

df.tail()

```
Message_bodyLabe11207 wonders in My WORLD 7th You 6th Ur style 5th...0121Try to do something dear. You read something f...0122Sun ah... Thk mayb can if dun have anythin on....0123SYMPTOMS when U are in love: "1.U like listeni...0124Great. Have a safe trip. Dont panic surrender ...0
```

df.info()

```
Message_body
                                                            Label
       0
                                  Rofl. Its true to its name Non-Spam
       1
               The guy did some bitching but I acted like i'd... Non-Spam
       2
                Pity, * was in mood for that. So...any other s... Non-Spam
df.isna().sum()
     Message_body
     dtype: int64
      120 7 wonders in Mv WORLD 7th You 6th Ur style 5th... Non-Spam
df['Label'].unique()
     array(['Non-Spam', 'Spam'], dtype=object)
          OVARDTONIO I II I I
                                        0.4 (1.19) (P. 4 - 2
Pie Chart 
# Calculating the non-pam & spam
df['Label'].value_counts()
     Non-Spam
                 198
     Spam
     Name: Label, dtype: int64
# we use matplotlib for better understanding of percentage of ham & spam
import matplotlib.pyplot as plt
plt.pie(df['Label'].value_counts(), labels=['Non-Spam','Spam'],autopct="%0.2f")
plt.show()
```



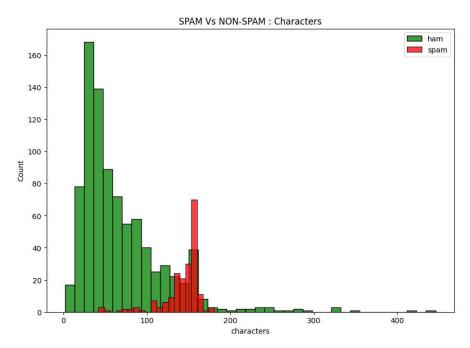
Obsrevations:¶

Data is Imblanced

Having Less Spam Messages in Dataset

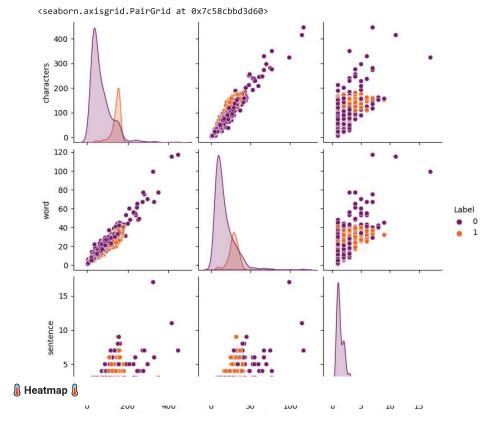
```
# Total No. of Characters in Data
df["characters"] = df["Message_body"].apply(len)
# Total No. of Words in Data
df["word"] = df["Message_body"].apply(lambda x:len( nltk.word_tokenize(x)))
# Total No. of Sentence
df["sentence"] = df["Message_body"].apply(lambda x:len(nltk.sent_tokenize(x)))
df
```

	Message_body	Label	characters	word	sentence	1
0	Rofl. Its true to its name	0	26	7	2	
1	The guy did some bitching but I acted like i'd	0	125	27	1	
2	Pity, * was in mood for that. Soany other s	0	57	15	2	
3	Will ü b going to esplanade fr home?	0	36	9	1	
4	This is the 2nd time we have tried 2 contact u	1	160	35	4	
120	7 wonders in My WORLD 7th You 6th Ur style 5th	0	151	33	1	
121	Try to do something dear. You read something f	0	54	11	2	
Histogram Plot						
123	SYMPTOMS when U are in love: "1.U like listeni	0	137	30	1	
sns.histpl	• •		-	•		



Nair Plot

Relation between the columns
sns.pairplot(df,hue="Label",palette='inferno')



#correlation and heatmap
sns.heatmap(df.corr(),annot = True)

<ipython-input-46-2d3646dd07cf>:1: FutureWarning: The default value of numeric_only in DataFrame.corr :
 sns.heatmap(df.corr(),annot = True)



df['Label']=df['Label'].map({'Spam':1,'Non-Spam':0})
df['Label']

```
8/1/23, 2:54 PM
         124
         Name: Label, Length: 1082, dtype: int64
    msg=df.Message_body
    msg
         0
                                       Rofl. Its true to its name
         1
                The guy did some bitching but I acted like i'd...
                Pity, * was in mood for that. So...any other s...
         2
         3
                             Will ü b going to esplanade fr home?
                This is the 2nd time we have tried 2 contact u...
         4
                7 wonders in My WORLD 7th You 6th Ur style 5th...
         120
         121
                Try to do something dear. You read something f...
         122
                Sun ah... Thk mayb can if dun have anythin on....
                SYMPTOMS when U are in love: "1.U like listeni...
         123
         124
                Great. Have a safe trip. Dont panic surrender ...
         Name: Message body, Length: 1082, dtype: object
    msg=msg.str.replace('[^a-zA-Z0-9]+',' ')
    msg
           msg=msg.str.replace('[^a-zA-Z0-9]+',' ')
                                        Rofl Its true to its name
                The guy did some bitching but I acted like i d...
         2
                Pity was in mood for that So any other suggest...
         3
                               Will b going to esplanade fr home
         4
                This is the 2nd time we have tried 2 contact u...
                7 wonders in My WORLD 7th You 6th Ur style 5th...
         120
         121
                Try to do something dear You read something fo...
                Sun ah Thk mayb can if dun have anythin on Thk...
         123
                SYMPTOMS when U are in love 1 U like listening...
         124
                 Great Have a safe trip Dont panic surrender all
         Name: Message_body, Length: 1082, dtype: object
    from nltk.stem import SnowballStemmer
    nltk.download('punkt')
    stemmer=SnowballStemmer('english')
    from nltk.tokenize import word tokenize
    msg=msg.apply(lambda x:[stemmer.stem(i.lower()) for i in word_tokenize(x)]).apply(lambda x:" ".join(x))
    msg
         [nltk_data] Downloading package punkt to /root/nltk_data...
         [nltk_data] Unzipping tokenizers/punkt.zip.
         0
                                          rofl it true to it name
                the guy did some bitch but i act like i d be i...
         1
                   piti was in mood for that so ani other suggest
         2
         3
                                    will b go to esplanad fr home
         4
                this is the 2nd time we have tri 2 contact u u...
                7 wonder in my world 7th you 6th ur style 5th ...
         120
```

```
<ipython-input-23-2edc65e333b0>:1: FutureWarning: The default value of regex will change from True to False in a future version.
```

tri to do someth dear you read someth for exam

great have a safe trip dont panic surrend all

sun ah thk mayb can if dun have anythin on thk...

symptom when u are in love 1 u like listen son...

Name: Message_body, Length: 1082, dtype: object

121

122

123

124

tnen ,

```
'there',
       'these',
      'they',
       'this',
       'those'
      'through',
       'to',
'too'
      'under',
       'until',
       'up',
       've',
       'very',
       'was',
       'wasn',
       "wasn't",
       'we',
       'were'
       'weren'
       "weren't",
       'what',
       'when',
       'where',
       'which',
       'while',
       'who',
       'whom',
       'whv'.
       'will',
       'with',
       'won',
       "won't",
       'wouldn'
       "wouldn't",
       'y',
       'you',
       "you'd",
       "you'll",
       "you're",
       "you've",
       'ýour',
       'yours<sup>'</sup>,
       'yourself',
       'yourselves'}
msg = msg.apply(lambda x:" ".join([w for w in word_tokenize(x) if len(w)>=3]))
msg
     0
                                                 rofl true name
            the guy did some bitch but act like interest b...
     1
     2
                      piti was mood for that ani other suggest
     3
                                             will esplanad home
     4
            this the 2nd time have tri contact have won th...
     120
            wonder world 7th you 6th style 5th smile 4th p...
     121
                      tri someth dear you read someth for exam
            sun thk mayb can dun have anythin thk have boo...
     122
     123
            symptom when are love like listen song get sto...
                   great have safe trip dont panic surrend all
     Name: Message_body, Length: 1082, dtype: object
from \ sklearn.feature\_extraction.text \ import \ TfidfVectorizer
vec = TfidfVectorizer()
train_data = vec.fit_transform(msg)
print(train_data)
       (0, 1788)
                      0.5086856793431559
       (0, 2676)
                      0.5352804139572925
       (0, 2198)
                      0.6743246681420617
       (1, 1138)
                      0.17859686537379169
       (1, 1122)
                      0.145663771594808
       (1, 1185)
                      0.29630458162284
       (1, 403)
                      0.1431901961566285
       (1, 2816)
                      0.20590361556826678
       (1, 1820)
                      0.2394517606916168
       (1, 985)
                      0.27446966691539726
       (1, 2389)
                      0.24847164852111867
       (1, 628)
                      0.24847164852111867
       (1, 1421)
                      0.2878344860869548
       (1, 1566)
                      0.19634036591184884
       (1, 333)
                      0.32003595991710204
       (1, 627)
                      0.17175261485986118
```

```
(1, 551)
                    0.3388726145756919
       (1, 2383)
                    0.21463994052388569
       (1, 890)
                    0.22747381784095883
       (1, 1255)
                    0.2422681930852499
       (1, 2575)
                    0.12401068454258501
       (2, 2488)
                    0.4427583140191907
       (2, 1906)
                    0.3275992987888908
       (2, 406)
                    0.31470189817687855
       (2, 2573)
                    0.2161139388471382
       (1080, 533)
                    0.30671363172226634
       (1080, 2522) 0.30671363172226634
       (1080, 1577) 0.26818530154548237
       (1080, 405) 0.26818530154548237
       (1080, 2831) 0.21432440301643546
       (1080, 2393) 0.27756807507225284
       (1080, 2457) 0.18290541973404553
       (1080, 2829) 0.361497263190884
       (1080, 1616) 0.18636337962795835
       (1080, 432) 0.14955153880181346
       (1080, 2263) 0.17869709545182685
       (1080, 2934) 0.2503424874315121
       (1080, 1201) 0.3032061498908571
       (1080, 2871) 0.1988622562872253
       (1080, 1566) 0.17770768156613842
       (1080, 2575) 0.11224208092485075
       (1080, 1788) 0.23137346071933745
       (1081, 2507) 0.4518718587714747
       (1081, 1926) 0.4518718587714747
       (1081, 934)
                    0.2906397621652869
       (1081, 386) 0.2647617934578632
       (1081, 2220) 0.4267540009467252
       (1081, 2673) 0.35216998096773255
       (1081, 1242) 0.30057271029644284
       (1081, 1287) 0.2014118687677222
train data.shape
     (1082, 2943)
y=df['Label'].values
У
     array([0, 0, 0, ..., 0, 0, 0])
Algorithms: Accuracy and Precision 
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(train_data,y,test_size=0.30,random_state=42)
print(x_train.shape,y_train.shape,x_test.shape,y_test.shape)
     (757, 2943) (757,) (325, 2943) (325,)
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix,accuracy_score
svm\_model = SVC()
nb model = MultinomialNB()
rf_model = RandomForestClassifier()
de_model = DecisionTreeClassifier()
lst_model = [svm_model,nb_model,rf_model,de_model]
for i in lst_model:
 print(i)
 i.fit(x_train,y_train)
 y_pred =i.predict(x_test)
 print("----")
 print()
 print("Confusion Matrix:")
 print(confusion_matrix(y_test, y_pred))
 print("Classification report :")
  print()
  print(classification_report(y_test, y_pred))
  nnint/"Accuracy Cong." accuracy cong/y and y tost\\
```

```
print( Accuracy Score: ,accuracy_Score(y_preu,y_test))
print()
  SVC()
  _____
  Confusion Matrix:
  [[269 0]
   [ 21 35]]
  Classification report :
                          recall f1-score
               precision
                                         support
                   0.93
            0
                           1.00
                                    0.96
                                              269
            1
                   1.00
                           0.62
                                    0.77
                                              56
                                    0.94
                                              325
      accuracy
                   0.96
                           0.81
                                    0.87
     macro avg
                                              325
  weighted avg
                   0.94
                           0.94
                                    0.93
                                              325
  Accuracy Score: 0.9353846153846154
  MultinomialNB()
  Confusion Matrix:
  [[269 0]
   [ 26 30]]
  Classification report :
               precision
                          recall f1-score
                                          support
            0
                   0.91
                           1.00
                                    0.95
                                              269
            1
                   1.00
                           0.54
                                    0.70
                                              56
                                    0.92
                                              325
      accuracy
                           0.77
                   0.96
     macro avg
                                    0.83
                                              325
  weighted avg
                   0.93
                           0.92
                                    0.91
                                              325
  Accuracy Score: 0.92
   ****************
  RandomForestClassifier()
  Confusion Matrix:
  [[269 0]
   [ 14 42]]
  Classification report :
               precision
                          recall f1-score
                                          support
            0
                   0.95
                           1.00
                                    0.97
                                              269
                   1.00
                           0.75
                                    0.86
                                              56
                                    0.96
                                              325
      accuracy
     macro avg
                   0.98
                           0.88
                                    0.92
                                              325
```

✓ 0s completed at 1:22 PM