```
In [63]: #Importig Packages
   import numpy as np
   import pandas as pd
   import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
   from sklearn.feature_extraction.text import TfidfVectorizer
   from sklearn.naive_bayes import MultinomialNB

from wordcloud import WordCloud
```

Out[64]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy Available only	NaN	NaN	NaN
1	ham	Ok lar Joking wif u oni	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina	NaN	NaN	NaN
3	ham	U dun say so early hor U c already then say	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro	NaN	NaN	NaN
•••					
5567	spam	This is the 2nd time we have tried 2 contact u	NaN	NaN	NaN
5568	ham	Will Ì _ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. Soany other s	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows \times 5 columns

In [65]: data.isnull().sum()

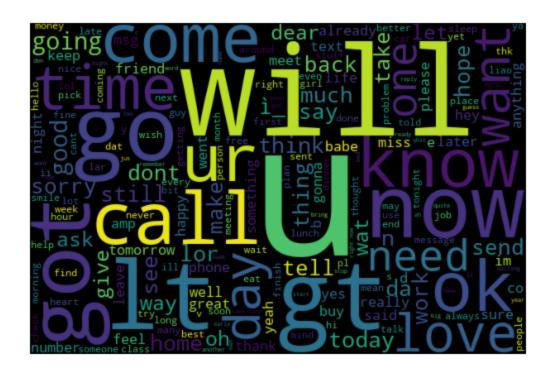
```
0
Out[65]: v1
           v2
                               0
           Unnamed: 2
                            5522
           Unnamed: 3
                            5560
           Unnamed: 4
                            5566
           dtype: int64
In [66]:
          data.drop(['Unnamed: 2','Unnamed: 3','Unnamed: 4'],axis=1,inplace=True)
          data.head()
In [67]:
Out[67]:
                 v1
                                                                v2
           0
                        Go until jurong point, crazy.. Available only ...
               ham
               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...
                        Nah I don't think he goes to usf, he lives aro...
           4
               ham
In [68]:
           data.columns = ['labels', 'messages']
           data.head()
Out[68]:
              labels
                                                          messages
           0
                ham
                         Go until jurong point, crazy.. Available only ...
           1
                ham
                                            Ok lar... Joking wif u oni...
           2
                      Free entry in 2 a wkly comp to win FA Cup fina...
               spam
                        U dun say so early hor... U c already then say...
           3
                ham
                        Nah I don't think he goes to usf, he lives aro...
           4
                ham
In [69]:
           data['b_labels'] = data.labels.map({'ham':0,'spam':1})
           data
```

Out[69]:		labels	messages	b_labels
	0	ham	Go until jurong point, crazy Available only	0
	1	ham	Ok lar Joking wif u oni	0
	2	spam	Free entry in 2 a wkly comp to win FA Cup fina	1
	3	ham	U dun say so early hor U c already then say	0
	4	ham	Nah I don't think he goes to usf, he lives aro	0
	•••			
	5567	spam	This is the 2nd time we have tried 2 contact u	1
	5568	ham	Will ì _ b going to esplanade fr home?	0
	5569	ham	Pity, * was in mood for that. Soany other s	0
	5570	ham	The guy did some bitching but I acted like i'd	0
	5571	ham	Rofl. Its true to its name	0

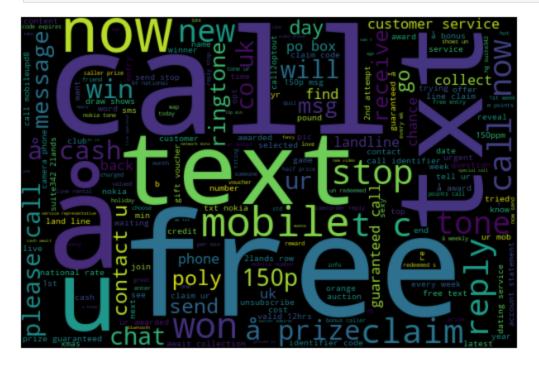
5572 rows × 3 columns

```
In [70]: x = data.messages
         y = data.b_labels.values
In [71]: x.head()
               Go until jurong point, crazy.. Available only ...
Out[71]: 0
                                   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...
               Nah I don't think he goes to usf, he lives aro...
         Name: messages, dtype: object
In [72]: data.b_labels.values
Out[72]: array([0, 0, 1, ..., 0, 0, 0], dtype=int64)
In [73]: #train test split
         x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.33)
In [74]: print(x_train.shape)
         print(y_train.shape)
         print(x_test.shape)
         print(y_test.shape)
        (3733,)
        (3733,)
        (1839,)
        (1839,)
```

```
In [75]: #vectorize
         tfidf = TfidfVectorizer()
         x_train = tfidf.fit_transform(x_train)
         x test = tfidf.transform(x test)
In [76]: | algo = MultinomialNB()
         algo.fit(x_train,y_train)
Out[76]: ▼ MultinomialNB
         MultinomialNB()
In [77]: print('Train Score : ',algo.score(x_train,y_train))
         print('Test Score : ',algo.score(x_test,y_test))
        Train Score: 0.9697294401285829
        Test Score: 0.9619358346927678
In [78]: pred = algo.predict(x_test)
         pred
Out[78]: array([1, 0, 0, ..., 0, 1, 0], dtype=int64)
In [79]: from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_scor
         ac = accuracy_score(pred,y_test)
         re = recall_score(pred,y_test)
         pr = precision_score(pred,y_test)
         f1 = f1_score(pred,y_test)
         print('Accuracy : ',ac)
         print('Recall : ',re)
         print('Precision : ',pr)
         print('F1 Score : ',f1)
        Accuracy: 0.9619358346927678
        Recall: 1.0
        Precision: 0.7107438016528925
        F1 Score: 0.8309178743961353
In [80]: #Data visualization using WordCloud
         def visualize(n):
             words = ''
             for i in data[data.labels==n]['messages']:
                 i = i.lower()
                 words += i
             wc = WordCloud(width=600, height=400).generate(words)
             plt.imshow(wc)
             plt.axis('off')
             plt.show()
In [81]: visualize('ham')
```



In [82]: visualize('spam')



In []: