In [ ]:
```
spam/ham classification using nlp:-
```

In [139]:
```
pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\sumee\anaconda3\lib\site-packag
es (3.6.1)
Requirement already satisfied: tqdm in c:\users\sumee\anaconda3\lib\site-packag
es (from nltk) (4.59.0)
Requirement already satisfied: regex in c:\users\sumee\anaconda3\lib\site-packa
ges (from nltk) (2021.4.4)
Requirement already satisfied: click in c:\users\sumee\anaconda3\lib\site-packa
ges (from nltk) (7.1.2)
Requirement already satisfied: joblib in c:\users\sumee\anaconda3\lib\site-pack
ages (from nltk) (1.0.1)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install transformers
```

```
Collecting transformers
  Downloading transformers-4.20.1-py3-none-any.whl (4.4 MB)
Requirement already satisfied: pyyaml>=5.1 in c:\users\sumee\anaconda3\lib\site
-packages (from transformers) (5.4.1)
Requirement already satisfied: requests in c:\users\sumee\anaconda3\lib\site-pa
ckages (from transformers) (2.25.1)
Requirement already satisfied: tqdm>=4.27 in c:\users\sumee\anaconda3\lib\site-
packages (from transformers) (4.59.0)
Requirement already satisfied: regex!=2019.12.17 in c:\users\sumee\anaconda3\li
b\site-packages (from transformers) (2021.4.4)
Collecting tokenizers!=0.11.3,<0.13,>=0.11.1
  Downloading tokenizers-0.12.1-cp38-cp38-win_amd64.whl (3.3 MB)
Requirement already satisfied: packaging>=20.0 in c:\users\sumee\anaconda3\lib
\site-packages (from transformers) (20.9)
Requirement already satisfied: numpy>=1.17 in c:\users\sumee\anaconda3\lib\site
-packages (from transformers) (1.20.1)
Requirement already satisfied: filelock in c:\users\sumee\anaconda3\lib\site-pa
ckages (from transformers) (3.0.12)
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.8.1-py3-none-any.whl (101 kB)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\sumee\ana
conda3\lib\site-packages (from huggingface-hub<1.0,>=0.1.0->transformers) (3.7.
4.3)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\sumee\anaconda3\lib
\site-packages (from packaging>=20.0->transformers) (2.4.7)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\sumee\anaconda3\li
b\site-packages (from requests->transformers) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\sumee\anaconda3\l
ib\site-packages (from requests->transformers) (2020.12.5)
Requirement already satisfied: idna<3,>=2.5 in c:\users\sumee\anaconda3\lib\sit
e-packages (from requests->transformers) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\sumee\anaconda
3\lib\site-packages (from requests->transformers) (1.26.4)
Installing collected packages: tokenizers, huggingface-hub, transformers
Successfully installed huggingface-hub-0.8.1 tokenizers-0.12.1 transformers-4.2
0.1
Note: you may need to restart the kernel to use updated packages.
```

In [80]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("darkgrid")
%matplotlib inline

import string
import nltk
from nltk.corpus import stopwords

from wordcloud import WordCloud
from sklearn.feature_extraction.text import CountVectorizer
from nltk.stem import WordNetLemmatizer

from sklearn.model_selection import train_test_split
from sklearn import metrics
```

In [81]:
```
messages = pd.read_csv('spam.csv',encoding = 'latin-1')
messages.head()
```

Out[81]:

	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

In [82]:
```
messages.tail()
```

Out[82]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
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. So...any 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

In [83]:
```
messages = messages.drop(labels = ["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"], axi
messages.columns = ["label", "message"]
```

In [84]:
```
messages.head()
```

Out[84]:

	label	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...

In [85]:
```
messages.info()
#There are total 5572 SMS in this dataset with 2 columns label and message.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   label    5572 non-null   object
 1   message  5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

In [86]:
```
messages.describe()
```

Out[86]:

	label	message
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

In [87]:
```
#Let's use groupby to use describe by label, this way we can begin to think about
```

In [88]:
```
messages.groupby('label').describe().T
```

Out[88]:

	label	ham	spam
message	count	4825	747
	unique	4516	653
	top	Sorry, I'll call later	Please call our customer service representativ...
	freq	30	4

In [89]:
```
messages['length'] = messages['message'].apply(len)
messages.head()
```

Out[89]:

	label	message	length
0	ham	Go until jurong point, crazy.. Available only ...	111
1	ham	Ok lar... Joking wif u oni...	29
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	ham	U dun say so early hor... U c already then say...	49
4	ham	Nah I don't think he goes to usf, he lives aro...	61

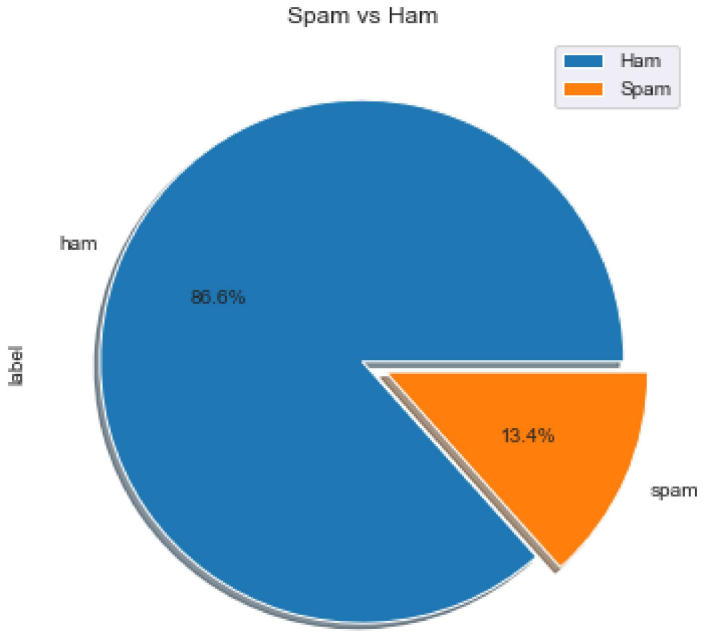
In [90]:
```
# Count the frequency of top 5 messages.
messages['message'].value_counts().rename_axis(['message']).reset_index(name='cou
```

Out[90]:

	message	counts
0	Sorry, I'll call later	30
1	I cant pick the phone right now. Pls send a me...	12
2	Ok...	10
3	Say this slowly.? GOD,I LOVE YOU & I NEED ...	4
4	Ok	4

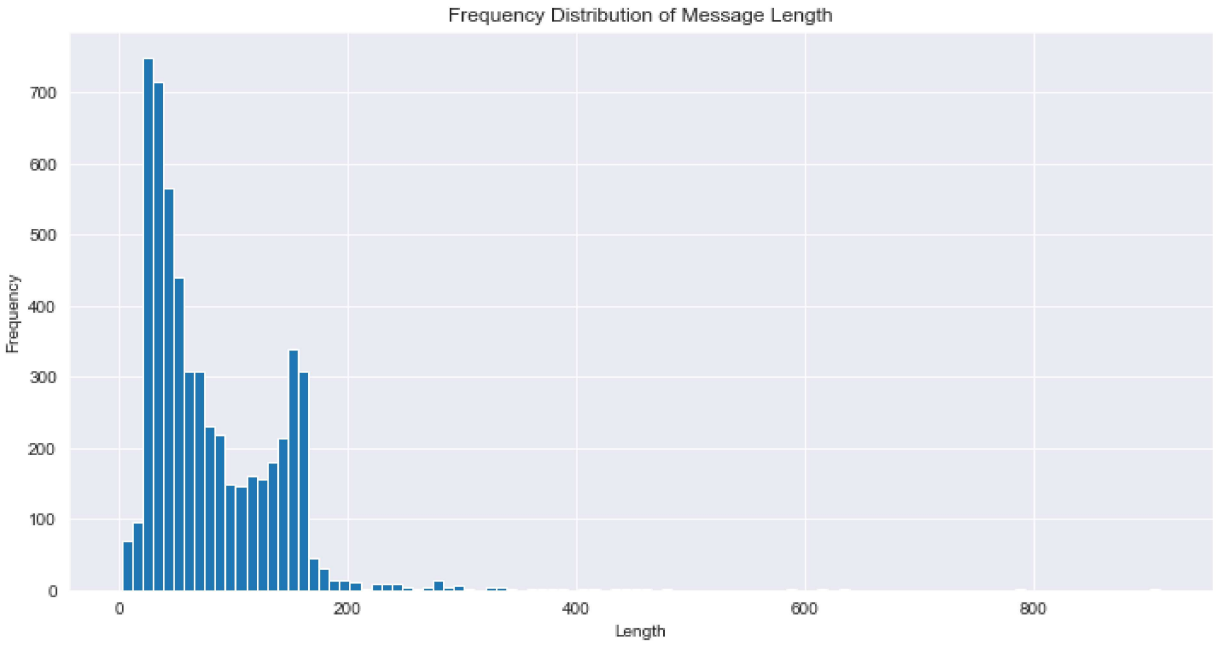
In [91]:
```
#Data Visualization:-
```

In [92]:
```
messages["label"].value_counts().plot(kind = 'pie',explode=[0, 0.1],figsize=(6, 6
plt.title("Spam vs Ham")
plt.legend(["Ham", "Spam"])
plt.show()
```



In [93]:
```
plt.figure(figsize=(12,6))
messages['length'].plot(bins=100, kind='hist') # with 100 length bins (100 length
plt.title("Frequency Distribution of Message Length")
plt.xlabel("Length")
plt.ylabel("Frequency")
```

Out[93]:
```
Text(0, 0.5, 'Frequency')
```



In [94]:
```
messages['length'].describe()
```

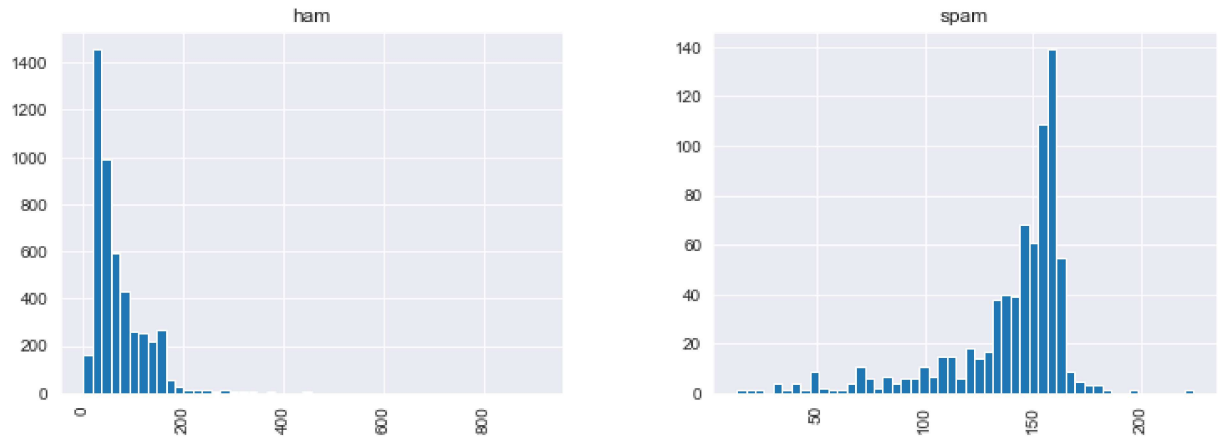
Out[94]:
```
count    5572.000000
mean       80.118808
std        59.690841
min         2.000000
25%        36.000000
50%        61.000000
75%       121.000000
max       910.000000
Name: length, dtype: float64
```

In [95]:
```
messages[messages['length'] == 910]['message'].iloc[0]
```

Out[95]:
```
"For me the love should start with attraction.i should feel that I need her eve
ry time around me.she should be the first thing which comes in my thoughts.I wo
uld start the day and end it with her.she should be there every time I dream.lo
ve will be then when my every breath has her name.my life should happen around
her.my life will be named to her.I would cry for her.will give all my happiness
and take all her sorrows.I will be ready to fight with anyone for her.I will be
in love when I will be doing the craziest things for her.love will be when I do
n't have to proove anyone that my girl is the most beautiful lady on the whole
planet.I will always be singing praises for her.love will be when I start up ma
king chicken curry and end up makiing sambar.life will be the most beautiful th
en.will get every morning and thank god for the day because she is with me.I wo
uld like to say a lot..will tell later.."
```

In [96]:
```
messages.hist(column='length', by='label', bins=50,figsize=(12,4))
```

Out[96]:
```
array([<AxesSubplot:title={'center':'ham'}>,
       <AxesSubplot:title={'center':'spam'}>], dtype=object)
```



In [97]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\sumee\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[97]:
```
True
```

In [98]:
```
def text_preprocess(mess):


    nopunc = [char for char in mess if char not in string.punctuation]


    nopunc = ''.join(nopunc)
    nopunc = nopunc.lower()

    nostop=[word for word in nopunc.split() if word.lower() not in stopwords.word

    return nostop
```

In [99]:
```
spam_messages = messages[messages["label"] == "spam"]["message"]
ham_messages = messages[messages["label"] == "ham"]["message"]
print("No of spam messages : ",len(spam_messages))
print("No of ham messages : ",len(ham_messages))
```

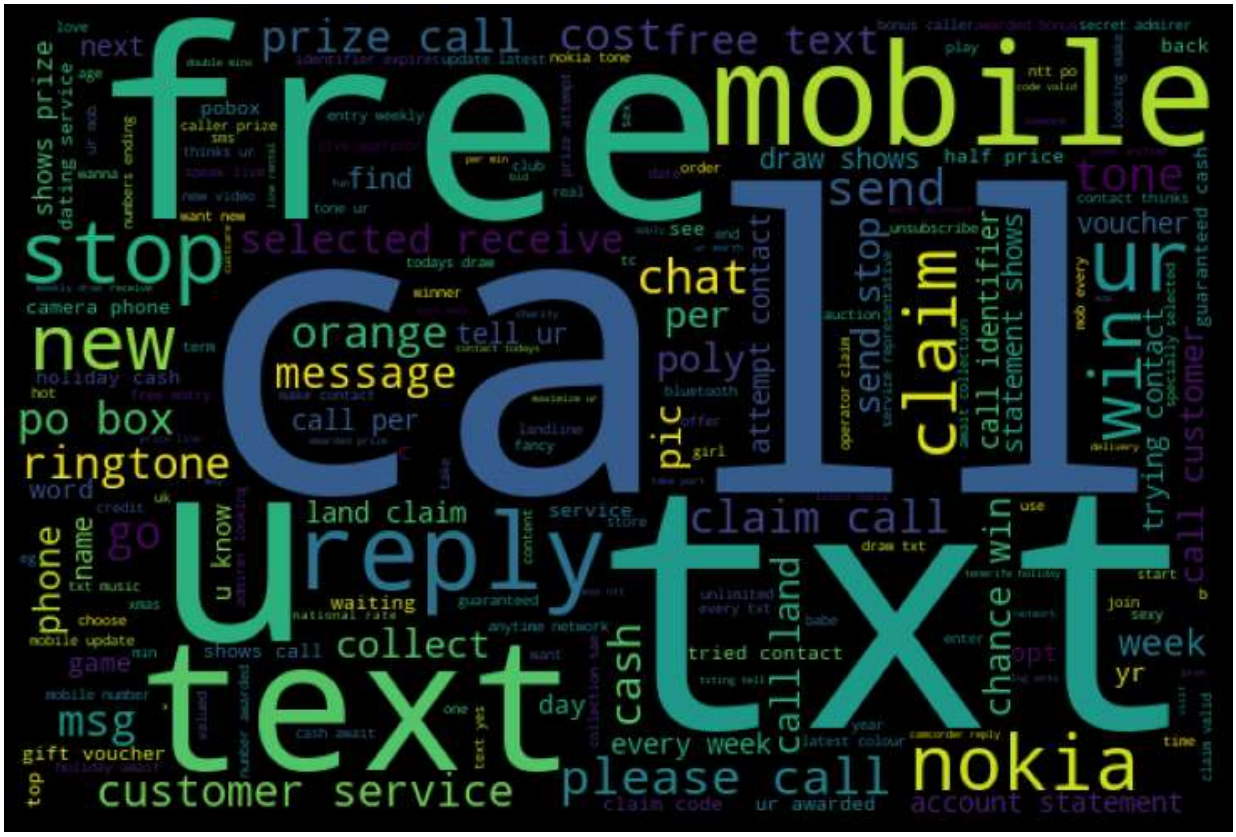
```
No of spam messages :  747
No of ham messages :  4825
```

In [100]:
```
spam_words = text_preprocess(spam_messages)
```

In [101]:
```
spam_words[:10]
```

Out[101]:
```
['free', 'entry', 'wkly', 'comp', 'win', 'fa', 'cup', 'final', 'tkts', 'may']
```

In [102]:
```
spam_wordcloud = WordCloud(width=600, height=400).generate(' '.join(spam_words))
plt.figure( figsize=(10,8), facecolor='k')
plt.imshow(spam_wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
plt.show()
```



In [103]:
```
print("Top 10 Spam words are :\n")
print(pd.Series(spam_words).value_counts().head(10))
```

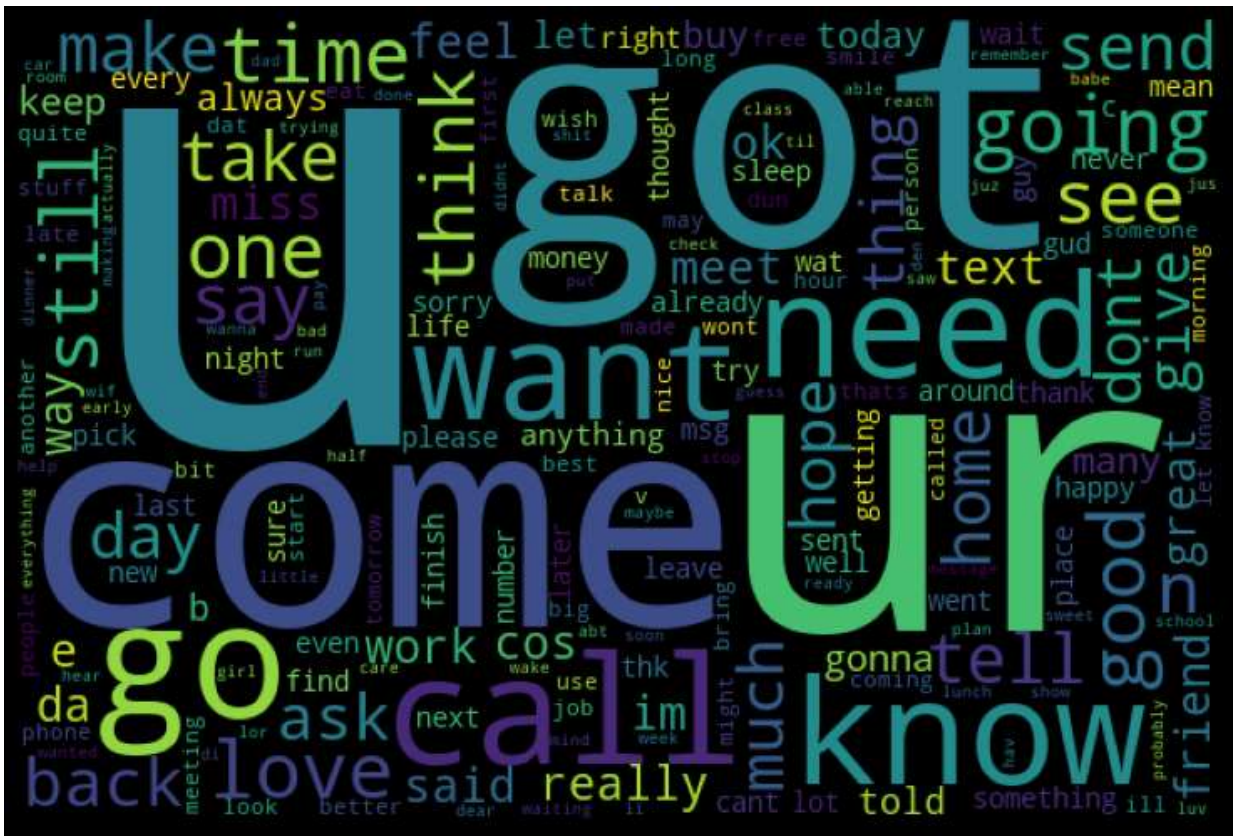
```
Top 10 Spam words are :

call      331
free      146
txt       136
ur        129
mobile    108
text      107
claim     105
reply      98
u          98
stop       78
dtype: int64
```

In [104]:
```
ham_words = text_preprocess(ham_messages)
```

In [105]:
```
ham_words[:10]
```

Out[105]:
```
['go',
 'jurong',
 'available',
 'bugis',
 'n',
 'great',
 'world',
 'la',
 'e',
 'cine']
```

In [106]:
```
ham_wordcloud = WordCloud(width=600, height=400).generate(' '.join(ham_words))
plt.figure( figsize=(10,8), facecolor='k')
plt.imshow(ham_wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
plt.show()
```



In [73]:
```
print("Top 10 Ham words are :\n")
print(pd.Series(ham_words).value_counts().head(10))
```

```
Top 10 Ham words are :

u         820
get       287
ur        235
go        231
got       216
like      215
know      202
come      201
call      200
going     151
dtype: int64
```

In [74]:
```
#Data transformation:-
```

In [75]:
```
messages.head()
```

Out[75]:

	label	message	length
0	ham	Go until jurong point, crazy.. Available only ...	111
1	ham	Ok lar... Joking wif u oni...	29
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	ham	U dun say so early hor... U c already then say...	49
4	ham	Nah I don't think he goes to usf, he lives aro...	61

In [76]:
```
messages["message"] = messages["message"].apply(text_preprocess)
```

In [77]:
```
messages.head()
```

Out[77]:

	label	message	length
0	ham	[go, jurong, point, crazy, available, bugis, n...	111
1	ham	[ok, lar, joking, wif, u, oni]	29
2	spam	[free, entry, wkly, comp, win, fa, cup, final,...	155
3	ham	[u, dun, say, early, hor, u, c, already, say]	49
4	ham	[nah, dont, think, goes, usf, lives, around, t...	61

In [78]:
```
messages["message"][7]
```

Out[78]:
```
['per',
 'request',
 'melle',
 'melle',
 'oru',
 'minnaminunginte',
 'nurungu',
 'vettam',
 'set',
 'callertune',
 'callers',
 'press',
 'copy',
 'friends',
 'callertune']
```

In [119]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [116]:
```
vectorizer = CountVectorizer()
bow_transformer = vectorizer.fit(messages['message'])

print("20 Bag of Words (BOW) Features: \n")
print(vectorizer.get_feature_names()[20:40])

print("\nTotal number of vocab words : ",len(vectorizer.vocabulary_))
```

```
20 Bag of Words (BOW) Features:

['0578', '06', '07', '07008009200', '07046744435', '07090201529', '0709029892
6', '07099833605', '07123456789', '0721072', '07732584351', '07734396839', '077
42676969', '07753741225', '0776xxxxxxx', '07781482378', '07786200117', '077xx
x', '078', '07801543489']

Total number of vocab words :  8672
```

In [113]:
```
message4 = messages['message'][3]
print(message4)
```

```
U dun say so early hor... U c already then say...
```

In [117]:
```
bow4 = bow_transformer.transform([message4])
print(bow4)
print(bow4.shape)
```

```
  (0, 1042)     1
  (0, 2802)     1
  (0, 2823)     1
  (0, 3927)     1
  (0, 6633)     2
  (0, 7024)     1
  (0, 7640)     1
(1, 8672)
```

In [114]:
```
print(bow_transformer.get_feature_names()[5945])
```

```
poortiyagi
```

In [120]:
```
messages_bow = bow_transformer.transform(messages['message'])
```

In [121]:
```
print('Shape of Sparse Matrix: ', messages_bow.shape)
print('Amount of Non-Zero occurences: ', messages_bow.nnz)
```

```
Shape of Sparse Matrix:  (5572, 8672)
Amount of Non-Zero occurences:  73916
```

In [122]:
```
from sklearn.feature_extraction.text import TfidfTransformer

tfidf_transformer = TfidfTransformer().fit(messages_bow)
```

In [123]:
```
tfidf4 = tfidf_transformer.transform(bow4)
print(tfidf4)
```

```
  (0, 7640)     0.2391367785302699
  (0, 7024)     0.2036385029167935
  (0, 6633)     0.588532244886041
  (0, 3927)     0.48845710205212745
  (0, 2823)     0.3528609993425001
  (0, 2802)     0.3250496221664022
  (0, 1042)     0.293626081506221
```

In [124]:
```
print(bow_transformer.get_feature_names()[5945])
print(bow_transformer.get_feature_names()[3141])
```

```
poortiyagi
fatty
```

In [125]:
```
print(tfidf_transformer.idf_[bow_transformer.vocabulary_['say']])
```

```
5.137052417837396
```

In [126]:
```
messages_tfidf = tfidf_transformer.transform(messages_bow)
print(messages_tfidf.shape)
```

```
(5572, 8672)
```

In [127]:
```
messages["message"][:10]
```

Out[127]:
```
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...
5    FreeMsg Hey there darling it's been 3 week's n...
6    Even my brother is not like to speak with me. ...
7    As per your request 'Melle Melle (Oru Minnamin...
8    WINNER!! As a valued network customer you have...
9    Had your mobile 11 months or more? U R entitle...
Name: message, dtype: object
```

In [128]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

vec = TfidfVectorizer(encoding = "latin-1", strip_accents = "unicode", stop_words
features = vec.fit_transform(messages["message"])
print(features.shape)

print(len(vec.vocabulary_))
```

```
(5572, 8402)
8402
```

In [129]:
```
msg_train, msg_test, label_train, label_test = \
train_test_split(messages_tfidf, messages['label'], test_size=0.2)
```

In [130]:
```
print("train dataset features size : ",msg_train.shape)
print("train dataset label size", label_train.shape)

print("\n")

print("test dataset features size", msg_test.shape)
print("test dataset lable size", label_test.shape)
```

```
train dataset features size :  (4457, 8672)
train dataset label size (4457,)


test dataset features size (1115, 8672)
test dataset lable size (1115,)
```

In [131]:
```
from sklearn.naive_bayes import MultinomialNB

clf = MultinomialNB()
spam_detect_model = clf.fit(msg_train, label_train)
```

In [132]:
```
predict_train = spam_detect_model.predict(msg_train)
```

In [133]:
```
print("Classification Report \n",metrics.classification_report(label_train, predi
print("\n")
print("Confusion Matrix \n",metrics.confusion_matrix(label_train, predict_train))
print("\n")
print("Accuracy of Train dataset : {0:0.3f}".format(metrics.accuracy_score(label_
```

```
Classification Report
               precision    recall  f1-score   support

          ham       0.96      1.00      0.98      3858
         spam       1.00      0.77      0.87       599

    accuracy                           0.97      4457
   macro avg       0.98      0.88      0.92      4457
weighted avg       0.97      0.97      0.97      4457



Confusion Matrix
 [[3858    0]
 [ 140  459]]


Accuracy of Train dataset : 0.969
```

In [134]:
```
print('predicted:', spam_detect_model.predict(tfidf4)[0])
print('expected:', messages['label'][3])
```

```
predicted: ham
expected: ham
```

In [ ]:
```
# Model evaluation:-
```

In [135]:
```
label_predictions = spam_detect_model.predict(msg_test)
print(label_predictions)
```

```
['ham' 'ham' 'ham' ... 'ham' 'ham' 'ham']
```

In [136]:
```
print(metrics.classification_report(label_test, label_predictions))
print(metrics.confusion_matrix(label_test, label_predictions))
```

```
              precision    recall  f1-score   support

         ham       0.96      1.00      0.98       967
        spam       1.00      0.74      0.85       148

    accuracy                           0.97      1115
   macro avg       0.98      0.87      0.92      1115
weighted avg       0.97      0.97      0.96      1115

[[967   0]
 [ 38 110]]
```

In [137]:
```
print("Accuracy of the model : {0:0.3f}".format(metrics.accuracy_score(label_test
```

```
Accuracy of the model : 0.966
```