In [2]:

```
# Import Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [3]:

```
# Warnings
import warnings
warnings.filterwarnings('ignore')
```

In [4]:

```
# Styles
plt.style.use('ggplot')
sns.set_style('whitegrid')

plt.rcParams['font.family'] = 'serif'
plt.rcParams['font.serif'] = 'Ubuntu'
plt.rcParams['font.monospace'] = 'Ubuntu Mono'
plt.rcParams['font.size'] = 10
plt.rcParams['axes.labelsize'] = 10
plt.rcParams['xtick.labelsize'] = 8
plt.rcParams['ytick.labelsize'] = 8
plt.rcParams['legend.fontsize'] = 10
plt.rcParams['figure.titlesize'] = 12
plt.rcParams['patch.force_edgecolor'] = True
```

In [5]:

```
# Text Preprocessing
import nltk
# nltk.download("all")
from nltk.corpus import stopwords
import string
from nltk.tokenize import word_tokenize
!pip install spacy
import spacy
nlp = spacy.load('en')
```

```
Requirement already satisfied: spacy in c:\users\bachh\anaconda3\lib\site-pa
ckages (2.1.4)
Requirement already satisfied: wasabi<1.1.0,>=0.2.0 in c:\users\bachh\anacon
da3\lib\site-packages (from spacy) (0.2.2)
Requirement already satisfied: numpy>=1.15.0 in c:\users\bachh\anaconda3\lib
\site-packages (from spacy) (1.15.4)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\bachh\anacond
a3\lib\site-packages (from spacy) (2.0.2)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\bachh\a
naconda3\lib\site-packages (from spacy) (1.0.2)
Requirement already satisfied: preshed<2.1.0,>=2.0.1 in c:\users\bachh\anaco
nda3\lib\site-packages (from spacy) (2.0.1)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\bachh\ana
conda3\lib\site-packages (from spacy) (2.21.0)
Requirement already satisfied: jsonschema<3.1.0,>=2.6.0 in c:\users\bachh\an
aconda3\lib\site-packages (from spacy) (2.6.0)
Requirement already satisfied: srsly<1.1.0,>=0.0.5 in c:\users\bachh\anacond
a3\lib\site-packages (from spacy) (0.0.5)
Requirement already satisfied: thinc<7.1.0,>=7.0.2 in c:\users\bachh\anacond
a3\lib\site-packages (from spacy) (7.0.4)
Requirement already satisfied: plac<1.0.0,>=0.9.6 in c:\users\bachh\anaconda
3\lib\site-packages (from spacy) (0.9.6)
Requirement already satisfied: blis<0.3.0,>=0.2.2 in c:\users\bachh\anaconda
3\lib\site-packages (from spacy) (0.2.4)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in c:\users\bachh\anaco
nda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (1.24.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\bachh\anaconda
3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2018.11.29)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\bachh\anaco
nda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\bachh\anaconda3\li
b\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2.8)
Requirement already satisfied: tqdm<5.0.0,>=4.10.0 in c:\users\bachh\anacond
a3\lib\site-packages (from thinc<7.1.0,>=7.0.2->spacy) (4.28.1)
```

In [6]:

```
texts = pd.read_csv("C:/Users/bachh/OneDrive/Desktop/Textbooks/Textbooks/TBANLT570/spam.csv

# Drop the extra columns and rename columns
texts = texts.drop(labels = ["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"], axis = 1)
texts.columns = ["category", "text"]
```

In [7]:

```
display(texts.head(n = 10))
```

	category	text
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...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

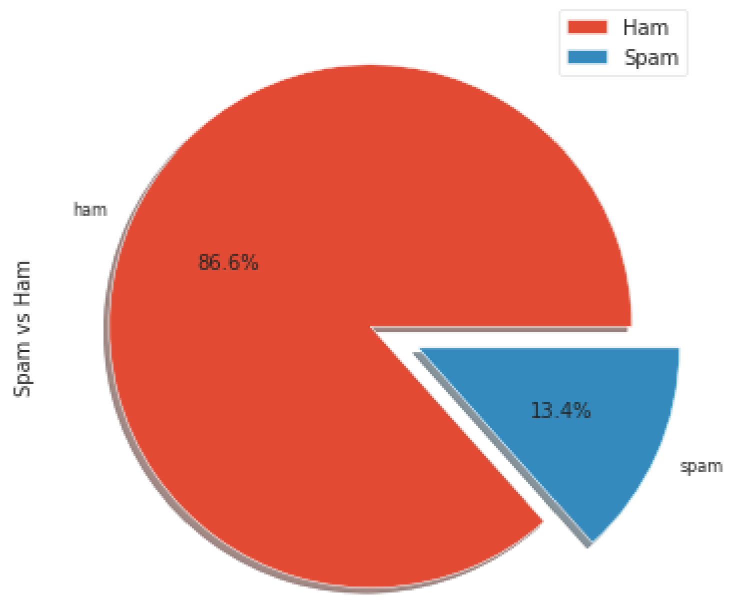
In [8]:

```
# Lets look at the dataset info to see if everything i
texts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
category    5572 non-null object
text        5572 non-null object
dtypes: object(2)
memory usage: 87.1+ KB
```

In [19]:

```
#Lets see what precentage of our data is spam/ham
texts["category"].value_counts().plot(kind = 'pie', explode = [0, 0.2], figsize = (6, 6), a
plt.ylabel("Spam vs Ham")
plt.legend(["Ham", "Spam"])
plt.show()
```



In [20]:

```
#top ham/spam messages
toptexts = texts.groupby("text")["category"].agg([len, np.max]).sort_values(by = "len", asc
display(toptexts)
```

text	len	amax
Sorry, I'll call later	30	ham
I cant pick the phone right now. Pls send a message	12	ham
Ok...	10	ham
Your opinion about me? 1. Over 2. Jada 3. Kusruthi 4. Lovable 5. Silent 6. Spl character 7. Not matured 8. Stylish 9. Simple Pls reply..	4	ham
Wen ur lovable bcums angry wid u, dnt take it seriously.. Coz being angry is d most childish n true way of showing deep affection, care n luv!.. kettoda manda... Have nice day da.	4	ham
Please call our customer service representative on FREEPHONE 0808 145 4742 between 9am-11pm as you have WON a guaranteed å£1000 cash or å£5000 prize!	4	spam
Okie	4	ham
Say this slowly.? GOD,I LOVE YOU &amp; I NEED YOU,CLEAN MY HEART WITH YOUR BLOOD.Send this to Ten special people &amp; u c miracle tomorrow, do it,pls,pls do it...	4	ham
7 wonders in My WORLD 7th You 6th Ur style 5th Ur smile 4th Ur Personality 3rd Ur Nature 2nd Ur SMS and 1st \Ur Lovely Friendship\"... good morning dear"	4	ham
Ok.	4	ham

In [21]:

```
# individual ham/spam words
spam_texts = texts[texts["category"] == "spam"]["text"]
ham_texts = texts[texts["category"] == "ham"]["text"]

spam_words = []
ham_words = []

# Since this is just classifying the message as spam or ham, we can use isalpha().
# This will also remove the not word in something like can't etc.
# In a sentiment analysis setting, its better to use
# sentence.translate(string.maketrans("", "", ), chars_to_remove)

def extractSpamWords(spamMessages):
    global spam_words
    words = [word.lower() for word in word_tokenize(spamMessages) if word.lower() not in st
    spam_words = spam_words + words

def extractHamWords(hamMessages):
    global ham_words
    words = [word.lower() for word in word_tokenize(hamMessages) if word.lower() not in stc
    ham_words = ham_words + words

spam_texts.apply(extractSpamWords)
ham_texts.apply(extractHamWords)
```

Out[21]:

```
0       None
1       None
3       None
4       None
6       None
7       None
10      None
13      None
14      None
16      None
17      None
18      None
20      None
21      None
22      None
23      None
24      None
25      None
26      None
27      None
28      None
29      None
30      None
31      None
32      None
33      None
35      None
36      None
37      None
38      None
```

```
        ...
5538    None
5539    None
5541    None
5542    None
5543    None
5544    None
5545    None
5546    None
5548    None
5549    None
5550    None
5551    None
5552    None
5553    None
5554    None
5555    None
5556    None
5557    None
5558    None
5559    None
5560    None
5561    None
5562    None
5563    None
5564    None
5565    None
5568    None
5569    None
5570    None
5571    None
Name: text, Length: 4825, dtype: object
```

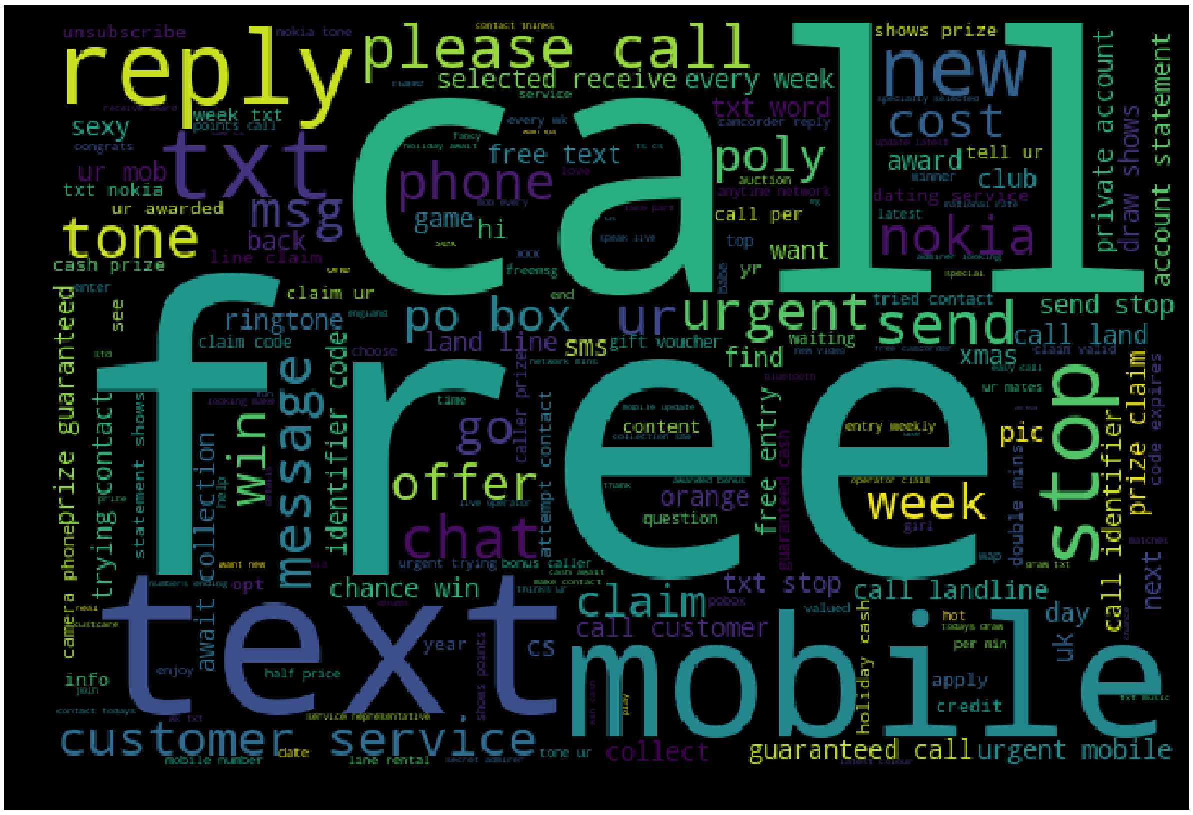
In [22]:

```
!pip install wordcloud
from wordcloud import WordCloud
```

```
Requirement already satisfied: wordcloud in c:\users\bachh\anaconda3\lib\sit
e-packages (1.5.0)
Requirement already satisfied: pillow in c:\users\bachh\anaconda3\lib\site-p
ackages (from wordcloud) (5.3.0)
Requirement already satisfied: numpy>=1.6.1 in c:\users\bachh\anaconda3\lib
\site-packages (from wordcloud) (1.15.4)
```

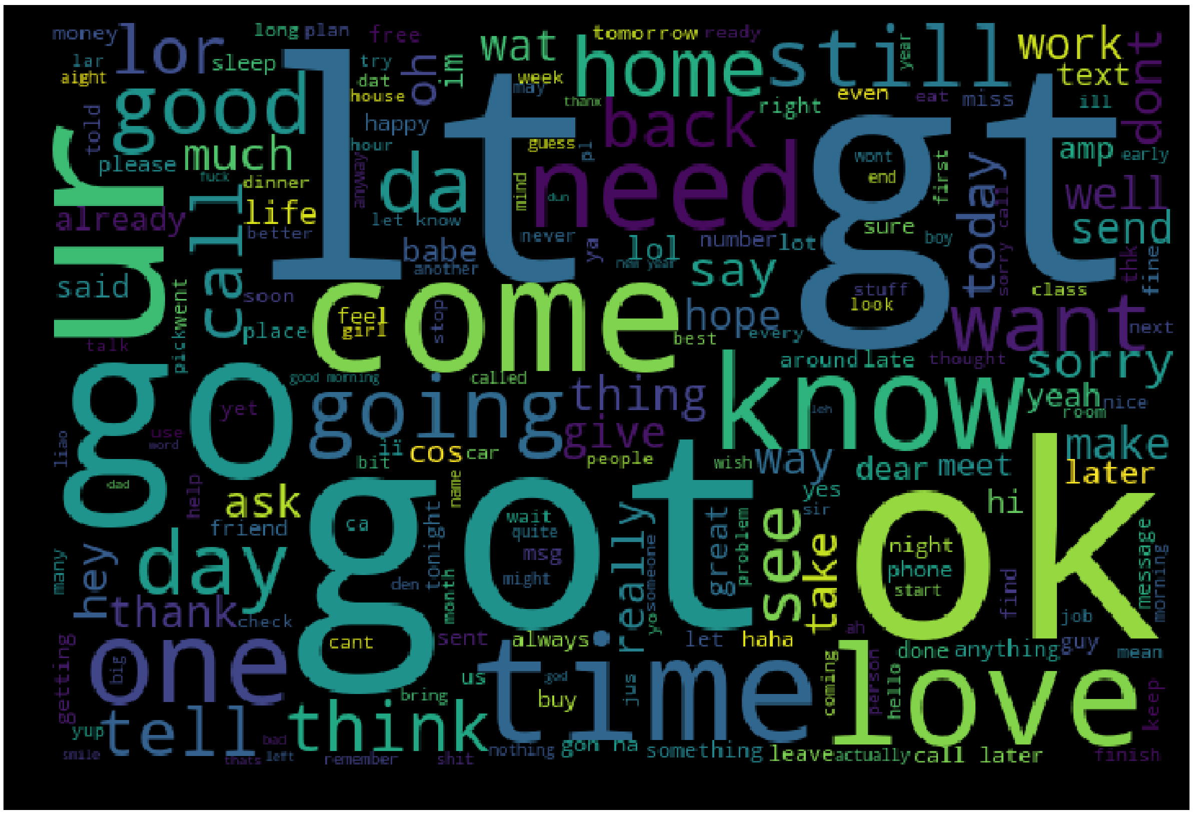
In [23]:

```
#Spam Word cloud

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 [24]:

```
#Ham word cloud

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 [25]:

```
# Top 10 spam words

spam_words = np.array(spam_words)
print("Top 10 Spam words are :\n")
pd.Series(spam_words).value_counts().head(n = 10)
```

```
Top 10 Spam words are :
```

Out[25]:

```
call      346
free      217
txt       156
ur        144
u         144
mobile    123
text      121
stop      114
claim     113
reply     104
dtype: int64
```

In [26]:

```
# Top 10 Ham words

ham_words = np.array(ham_words)
print("Top 10 Ham words are :\n")
pd.Series(ham_words).value_counts().head(n = 10)
```

```
Top 10 Ham words are :
```

Out[26]:

```
u       974
gt      318
lt      316
get     301
ok      246
go      246
got     242
ur      237
know    234
like    231
dtype: int64
```

In [27]:

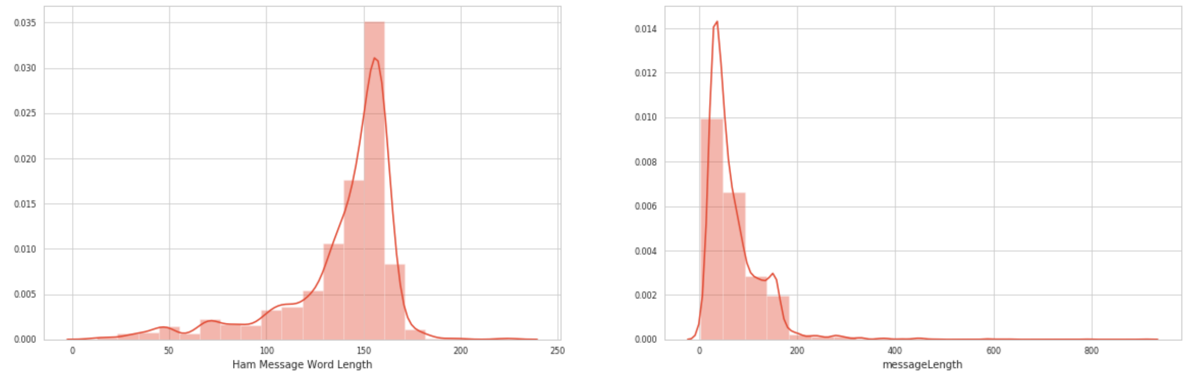
```
# indication of length
texts["messageLength"] = texts["text"].apply(len)
texts["messageLength"].describe()
```

Out[27]:

```
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: messageLength, dtype: float64
```

In [28]:

```
f, ax = plt.subplots(1, 2, figsize = (20, 6))

sns.distplot(texts[texts["category"] == "spam"]["messageLength"], bins = 20, ax = ax[0])
ax[0].set_xlabel("Spam Message Word Length")

sns.distplot(texts[texts["category"] == "ham"]["messageLength"], bins = 20, ax = ax[1])
ax[0].set_xlabel("Ham Message Word Length")

plt.show()
```



In [30]:

```
from nltk.stem import SnowballStemmer
stemmer = SnowballStemmer("english")

def cleanText(message):

    message = message.translate(str.maketrans('', '', string.punctuation))
    words = [stemmer.stem(word) for word in message.split() if word.lower() not in stopworc

    return " ".join(words)

texts["text"] = texts["text"].apply(cleanText)
texts.head(n = 10)
```

Out[30]:

	category	text	messageLength
0	ham	go jurong point crazi avail bugi n great world...	111
1	ham	ok lar joke wif u oni	29
2	spam	free entri 2 wkli comp win fa cup final tkts 2...	155
3	ham	u dun say earli hor u c alreadi say	49
4	ham	nah dont think goe usf live around though	61
5	spam	freemsg hey darl 3 week word back id like fun ...	148
6	ham	even brother like speak treat like aid patent	77
7	ham	per request mell mell oru minnaminungint nurun...	160
8	spam	winner valu network custom select receivea â£9...	158
9	spam	mobil 11 month u r entitl updat latest colour ...	154

In [31]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vec = TfidfVectorizer(encoding = "latin-1", strip_accents = "unicode", stop_words = "englis
features = vec.fit_transform(texts["text"])
print(features.shape)
```

```
(5572, 7903)
```

In [32]:

```
#model
def encodeCategory(cat):
    if cat == "spam":
        return 1
    else:
        return 0

texts["category"] = texts["category"].apply(encodeCategory)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, texts["category"], stratify =
```

In [33]:

```
from sklearn.model_selection import cross_val_score
from sklearn.metrics import fbeta_score

from sklearn.naive_bayes import MultinomialNB
gaussianNb = MultinomialNB()
gaussianNb.fit(X_train, y_train)

y_pred = gaussianNb.predict(X_test)

print(fbeta_score(y_test, y_pred, beta = 0.5))
```

```
0.933786078098472
```

In [ ]: