Libraries

```
# Import library
import pandas as pd
import numpy as np
import string
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from collections import Counter
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.model_selection import GridSearchCV
%matplotlib inline
```

NLP

```
import pandas as pd
df = pd.read_csv("train.csv")
```

```
df
```

	Tweet	following	followers	actions	is_retweet	location	Type	Unnamed: 7
0	Good Morning Love @LeeBrown_V	0.0	0.0	0.0	0.0	Pennsylvania, USA	Quality	NaN
1	'@realDonaldTrump @USNavy RIP TO HEROES'	42096.0	61060.0	5001.0	0.0	South Padre Island, Texas	Spam	NaN
2	Haven't been following the news but I understa...	0.0	0.0	NaN	0.0	Will never be broke ever again	Quality	NaN
3	pic.twitter.com/dy9q4ftLhZ What to do with pap...	0.0	0.0	0.0	0.0	Mundo	Quality	NaN
4	#DidYouKnow ► Mahatma Gandhi made a brief visi...	17800.0	35100.0	NaN	0.0	Nottingham, England	Quality	NaN
...	...	...	...	...	...	...	...	...
14894	#AllWentWrongWhen I told my hair stylist to "g...	695.0	533.0	868.0	1.0	United States	Spam	NaN

Pre - Processing

```
df = df.drop(["following","followers","actions","is_retweet","location","Unnamed: 7"],axis=1)
```

```
df
```

	Tweet	Type
0	Good Morning Love @LeeBrown_V	Quality
1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam
2	Haven't been following the news but I understa...	Quality
3	pic.twitter.com/dy9q4ftLhZ What to do with pap...	Quality

```
df
```

	Tweet	Type
0	Good Morning Love @LeeBrown_V	Quality
1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam
2	Haven't been following the news but I understa...	Quality
3	pic.twitter.com/dy9q4ftLhZ What to do with pap...	Quality
4	#DidYouKnow ▶ Mahatma Gandhi made a brief visi...	Quality
...	...	...
14894	#AllWentWrongWhen I told my hair stylist to "g...	Spam
14895	They don't have to like you, and you don't hav...	Quality
14896	#Miami Graham Nash Live at Parker Playhouse #...	Spam
14897	@bethannhamilton is in the business of one-upp...	Quality
14898	Chasing Success by Space Cadetz Listen up...	Spam

14899 rows × 2 columns

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

```
import string
string.punctuation
```

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

Removing Punctuations

```
# Function to remove punctuations.
def remove_punc(text):
    nonP_text = "".join([char for char in text if char not in string.punctuation])
    return nonP_text

df["body_text_clean"] = df["Tweet"].apply(lambda x: remove_punc(x))

df.head()
```

	Tweet	Type	body_text_clean
0	Good Morning Love @LeeBrown_V	Quality	Good Morning Love LeeBrownV
1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam	realDonaldTrump USNavy RIP TO HEROES
2	Haven't been following the news but I understa...	Quality	Havent been following the news but I understan...
3	pic.twitter.com/dy9q4ftLhZ What to do with pap...	Quality	pictwittercomdy9q4ftLhZ What to do with paper ...
4	#DidYouKnow ▶ Mahatma Gandhi made a brief visi...	Quality	DidYouKnow ▶ Mahatma Gandhi made a brief visit...

Tokenization

```
import re
```

```
#function to apply tokenization
def tokenize(text):
    tokens = re.split("\W+", text)# W+ means all capital, small alphabets and integers 0-9
    return tokens

df["body_text_tokenized"] = df["body_text_clean"].apply(lambda x: tokenize(x))

df.head()
```

	Tweet	Type	body_text_clean	body_text_tokenized
0	Good Morning Love @LeeBrown_V	Quality	Good Morning Love LeeBrownV	[Good, Morning, Love, LeeBrownV]
1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam	realDonaldTrump USNavy RIP TO HEROES	[realDonaldTrump, USNavy, RIP, TO, HEROES]
2	Haven't been following the news but I understa...	Quality	Havent been following the news but I understan...	[Havent, been, following, the, news, but, I, u...
3	pic.twitter.com/dy9q4ftLhZ What to do with pap...	Quality	pictwittercomdy9q4ftLhZ What to do with paper ...	[pictwittercomdy9q4ftLhZ, What, to, do, with, ...

```
!pip install nltk
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: nltk in /usr/local/lib/python3.9/dist-packages (3.8.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.9/dist-packages (from nltk) (4.65.0)
Requirement already satisfied: joblib in /usr/local/lib/python3.9/dist-packages (from nltk) (1.2.0)
Requirement already satisfied: click in /usr/local/lib/python3.9/dist-packages (from nltk) (8.1.3)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.9/dist-packages (from nltk) (2022.10.31)
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

Remove Stopwords

```
import nltk
stopwords = nltk.corpus.stopwords.words("english")

def remove_stopwords(token):
    text = [word for word in token if word not in stopwords]# to remove all stopwords
    return text

df["body_text_nonstop"] = df["body_text_tokenized"].apply(lambda x: remove_stopwords(x))
df.head()
```

	Tweet	Type	body_text_clean	body_text_tokenized	body_text_nonstop
0	Good Morning Love @LeeBrown_V	Quality	Good Morning Love LeeBrownV	[Good, Morning, Love, LeeBrownV]	[Good, Morning, Love, LeeBrownV]
1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam	realDonaldTrump USNavy RIP TO HEROES	[realDonaldTrump, USNavy, RIP, TO, HEROES]	[realDonaldTrump, USNavy, RIP, TO, HEROES]
2	Haven't been following the news but I understa...	Quality	Havent been following the news but I understan...	[Havent, been, following, the, news, but, I, u...	[Havent, following, news, I, understand, EFF, ...

Stemming

```
ps = nltk.PorterStemmer()

def stemming(t_text):
    text = [ps.stem(word) for word in t_text]
    return text

df["body_text_stemmed"] = df["body_text_nonstop"].apply(lambda x: stemming(x))
df.head()
```

	Tweet	Type	body_text_clean	body_text_tokenized	body_text_nonstop	
0	Good Morning Love @LeeBrown_V	Quality	Good Morning Love LeeBrownV	[Good, Morning, Love, LeeBrownV]	[Good, Morning, Love, LeeBrownV]	
1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam	realDonaldTrump USNavy RIP TO HEROES	[realDonaldTrump, USNavy, RIP, TO, HEROES]	[realDonaldTrump, USNavy, RIP, TO, HEROES]	
2	Haven't been following the news but I understa...	Quality	Havent been following the news but I understan...	[Havent, been, following, the, news, but, I, u...	[Havent, following, news, I, understand, EFF, ...	[
3	pic.twitter.com/dy9q4ftLhZ What to do with pap...	Quality	pictwittercomdy9q4ftLhZ What to do with paper ...	[pictwittercomdy9q4ftLhZ, What, to, do, with, ...	[pictwittercomdy9q4ftLhZ, What, paper, scissor...	[p

Lemmatization

```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
wn = nltk.WordNetLemmatizer()

def lemmatizer(t_text):
    text = [wn.lemmatize(word) for word in t_text]
    return text

df["body_text_lemmatized"] = df["body_text_stemmed"].apply(lambda x: lemmatizer(x))
df.head()
```

	Tweet	Type	body_text_clean	body_text_tokenized	body_text_nonstop	
0	Good Morning Love @LeeBrown_V	Quality	Good Morning Love LeeBrownV	[Good, Morning, Love, LeeBrownV]	[Good, Morning, Love, LeeBrownV]	
1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam	realDonaldTrump USNavy RIP TO HEROES	[realDonaldTrump, USNavy, RIP, TO, HEROES]	[realDonaldTrump, USNavy, RIP, TO, HEROES]	
2	Haven't been following the news but I understa...	Quality	Havent been following the news but I understan...	[Havent, been, following, the, news, but, I, u...	[Havent, following, news, I, understand, EFF, ...	[
3	pic.twitter.com/dy9q4ftLhZ What to do with pap...	Quality	pictwittercomdy9q4ftLhZ What to do with paper ...	[pictwittercomdy9q4ftLhZ, What, to, do, with, ...	[pictwittercomdy9q4ftLhZ, What, paper, scissor...	[p
	#DidYouKnow ►		DidYouKnow ►	[DidYouKnow, Mahatma	[DidYouKnow, Mahatma	[(

Data Saving - Pre Processed

```
df.to_csv("pre_processed_data.csv", sep=',')
```

```
import pandas as pd
import re
import string
import nltk

data = pd.read_csv("pre_processed_data.csv", sep=',')
```

```
data
```

	Unnamed: 0	Tweet	Type	body_text_clean	body_text_toker
0	0	Good Morning Love @LeeBrown_V	Quality	Good Morning Love LeeBrownV	['Good', 'Morning', 'l 'LeeBro
1	1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam	realDonaldTrump USNavy RIP TO HEROES	['realDonaldTr 'USNavy', 'RIP' 'HER
2	2	Haven't been following the news but I understand #EFF was doing the dumbest things	Quality	Havent been following the news but I understand EFF was doing the dumbest things	['Havent', 't 'following', 'the', 'r 'but', 'I', 'unders 'EFF', 'was', 'doir
3	3	pic.twitter.com/dy9q4ftLhZ What to do with paper scissors and glue http://paperlandmarks.com/pro...	Quality	pictwittercomdy9q4ftLhZ What to do with paper scissors and glue httppaperlandmarkscomproductthep...	['pictwittercomdy9q4f 'What', 'to', 'do', 'paper', 'scissors', 'glue',
4	4	#DidYouKnow ► Mahatma Gandhi made a brief visit to lecture in #Nottingham on 17 October 1931 [@M...	Quality	DidYouKnow ► Mahatma Gandhi made a brief visit to lecture in Nottingham on 17 October 1931 Mumbl...	['DidYouKnow', 'Maha 'Gandhi', 'made', 'a', ' 'visit', 'to', 'lecture 'N
...	...	...	...	...	
14894	14894	#AllWentWrongWhen I told my hair stylist to "go nuts".	Spam	AllWentWrongWhen I told my hair stylist to go nuts	['AllWentWrongWhe 'told', 'my', 'hair', 's 'to', 'go',
14895	14895	They don't have to like you, and you don't have to care.	Quality	They dont have to like you and you dont have to care	['They', 'dont', 'have 'like', 'you', 'and', 'dont', 'have', 'to', '
14896	14896	#Miami Graham Nash Live at Parker Playhouse #local	Spam	Miami Graham Nash Live at Parker Playhouse local	['Miami', 'Graham', 'N 'Live', 'at', 'Pa 'Playhouse', '

Apply Count Vectorizer

'she',

1. Convert the text into lower case
2. Remove punctuations
3. Split the text into tokens
4. Remove all stop words
5. Stems each word in the text using the porter stemmer algorithm

```
def clean_text(text):
    text = "".join([word.lower() for word in text if word not in string.punctuation])
    tokens = re.split('\W+', text)
    text = [ps.stem(word) for word in tokens if word not in stopwords]
    return text
```

1. We are using Count vectorizer to transform the text data into a matrix of token counts.
2. The function, get_feature_names_out method return the list of feature names(tokens) in the order they appear in the matrix. These feature names represent the vocabolary of the CountVectorizer model from the input data

```
from sklearn.feature_extraction.text import CountVectorizer

count = CountVectorizer(analyzer=clean_text)
x_count = count.fit_transform(data["Tweet"])

x_count.shape
count.get_feature_names_out()
```

```
array(['', '0', '000', ..., 'j e t', 'n o t', 's t r e a m'], dtype=object)
```

```
x_count_df = pd.DataFrame(x_count.toarray(), columns=count.get_feature_names_out())
x_count_df.head(10)
```

	0	000	000000	0000ampictwittercomenr5wmejyn	0005	002	003share	005	0056	...	pictwittercomr
0	0	0	0	0	0	0	0	0	0	...	
1	0	0	0	0	0	0	0	0	0	...	
2	0	0	0	0	0	0	0	0	0	...	
3	0	0	0	0	0	0	0	0	0	...	
4	0	0	0	0	0	0	0	0	0	...	
5	0	0	0	0	0	0	0	0	0	...	
6	0	0	0	0	0	0	0	0	0	...	
7	1	0	0	0	0	0	0	0	0	...	
9	0	0	0	0	0	0	0	0	0	...	

Vectorize Raw Data:

N Grams

Apply CountVectorizer(N-Grams)

[ ] ↳ *3 cells hidden*

2. TF - IDF

Apply TF - IDFvectorizer

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vect = TfidfVectorizer(analyzer=clean_text)
X_tfidf = tfidf_vect.fit_transform(data['Tweet'])
print(X_tfidf.shape)
print(tfidf_vect.get_feature_names_out())
```

```
(14899, 32599)
['' '0' '000' ... 'j e t' 'n o t' 's t r e a m']
```

```
X_tfidf_df = pd.DataFrame(X_tfidf.toarray(), columns=tfidf_vect.get_feature_names_out())
X_tfidf_df.head(10)
```

	0	000	000000	0000ampictwittercomenr5wmejyn	0005	002	003share	005	0056	...	pictwi
0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
2	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
3	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
4	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
5	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
6	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
7	0.328324	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
8	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
9	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

```
print(X_tfidf_df.loc[(X_tfidf_df!=0).any(axis=1)])
```

```
                    0  000  000000  0000ampictwittercomenr5wmejyn  0005  002  \
0      0.000000  0.0  0.0     0.0                            0.0   0.0  0.0
1      0.000000  0.0  0.0     0.0                            0.0   0.0  0.0
2      0.000000  0.0  0.0     0.0                            0.0   0.0  0.0
3      0.000000  0.0  0.0     0.0                            0.0   0.0  0.0
4      0.000000  0.0  0.0     0.0                            0.0   0.0  0.0
...         ...  ...  ...     ...                            ...   ...  ...
14894  0.000000  0.0  0.0     0.0                            0.0   0.0  0.0
14895  0.000000  0.0  0.0     0.0                            0.0   0.0  0.0
14896  0.000000  0.0  0.0     0.0                            0.0   0.0  0.0
14897  0.000000  0.0  0.0     0.0                            0.0   0.0  0.0
14898  0.169263  0.0  0.0     0.0                            0.0   0.0  0.0

       003share  005  0056  ...  티파챗pictwittercomrhkf05rsid  화양연화pt1  화양연화pt2  \
0           0.0  0.0   0.0  ...                          0.0      0.0      0.0
1           0.0  0.0   0.0  ...                          0.0      0.0      0.0
2           0.0  0.0   0.0  ...                          0.0      0.0      0.0
3           0.0  0.0   0.0  ...                          0.0      0.0      0.0
4           0.0  0.0   0.0  ...                          0.0      0.0      0.0
...         ...  ...   ...  ...                          ...      ...      ...
14894       0.0  0.0   0.0  ...                          0.0      0.0      0.0
14895       0.0  0.0   0.0  ...                          0.0      0.0      0.0
14896       0.0  0.0   0.0  ...                          0.0      0.0      0.0
14897       0.0  0.0   0.0  ...                          0.0      0.0      0.0
14898       0.0  0.0   0.0  ...                          0.0      0.0      0.0

       화양연화youngforev  c l o s e r  c o m e  d o  j e t  n o t  s t r e a m
0                 0.0     0.0   0.0  0.0  0.0  0.0     0.0
1                 0.0     0.0   0.0  0.0  0.0  0.0     0.0
2                 0.0     0.0   0.0  0.0  0.0  0.0     0.0
3                 0.0     0.0   0.0  0.0  0.0  0.0     0.0
4                 0.0     0.0   0.0  0.0  0.0  0.0     0.0
...               ...     ...   ...  ...  ...  ...     ...
14894             0.0     0.0   0.0  0.0  0.0  0.0     0.0
14895             0.0     0.0   0.0  0.0  0.0  0.0     0.0
14896             0.0     0.0   0.0  0.0  0.0  0.0     0.0
14897             0.0     0.0   0.0  0.0  0.0  0.0     0.0
14898             0.0     0.0   0.0  0.0  0.0  0.0     0.0

[14887 rows x 32599 columns]
```

Feature Engineering: Feature Creation

```
import pandas as pd

data = pd.read_csv("pre_processed_data.csv", sep=',')


data
```

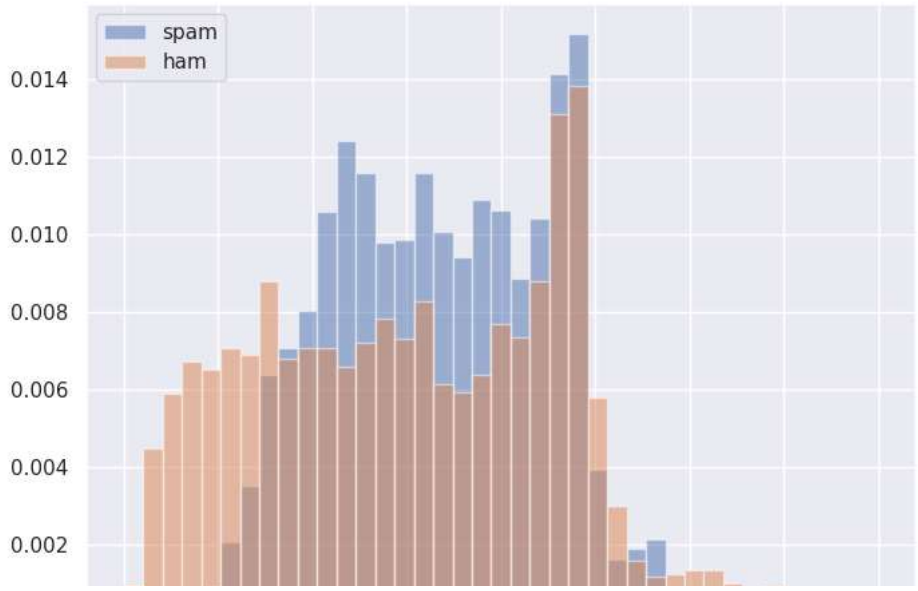
	Unnamed: 0	Tweet	Type	body_text_clean	body_text_toker
0	0	Good Morning Love @LeeBrown_V	Quality	Good Morning Love LeeBrownV	['Good', 'Morning', 'I 'LeeBro
1	1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam	realDonaldTrump USNavy RIP TO HEROES	['realDonaldTr 'USNavy', 'RIP' 'HER
2	2	Haven't been following the news but I understand #EFF was doing the dumbest things	Quality	Havent been following the news but I understand EFF was doing the dumbest things	['Havent', 't 'following', 'the', 'r 'but', 'I', 'unders 'EFF', 'was', 'doir
3	3	pic.twitter.com/dy9q4ftLhZ What to do with paper scissors and glue http://paperlandmarks.com/pro...	Quality	pictwittercomdy9q4ftLhZ What to do with paper scissors and glue httppaperlandmarkscomproductthep...	['pictwittercomdy9q4f 'What', 'to', 'do', 'paper', 'scissors', 'glue',

Create feature for text message length and % of punctuation in text

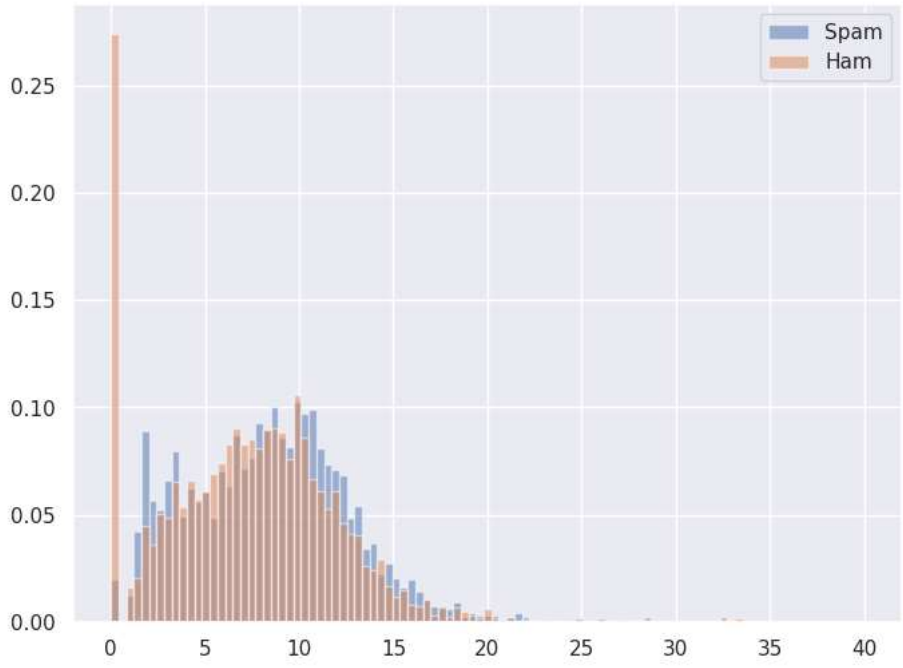
```
import string

# Function to calculate length of message excluding space
data['body_len'] = data['Tweet'].apply(lambda x: len(x) - x.count(" "))

data.head()

def count_punct(text):
    count = sum([1 for char in text if char in string.punctuation])
    return round(count/(len(text) - text.count(" ")), 3)*100

data['punct%'] = data['Tweet'].apply(lambda x: count_punct(x))

data.head()
```

	Unnamed: 0	Tweet	Type	body_text_clean	body_text_tokenized
0	0	Good Morning Love @LeeBrown_V	Quality	Good Morning Love LeeBrownV	['Good', 'Morning', 'Love' 'LeeBrownV'
1	1	'@realDonaldTrump @USNavy RIP TO HEROES'	Spam	realDonaldTrump USNavy RIP TO HEROES	['realDonaldTrump' 'USNavy', 'RIP', 'TO' 'HEROES'
2	2	Haven't been following the news but I understand #EFF was doing the dumbest things	Quality	Havent been following the news but I understand EFF was doing the dumbest things	['Havent', 'been' 'following', 'the', 'news' 'but', 'I', 'understand' 'EFF', 'was', 'doing', ..
3	3	pic.twitter.com/dy9q4ftLhZ What to do with paper scissors and glue http://paperlandmarks.com/pro...	Quality	pictwittercomdy9q4ftLhZ What to do with paper scissors and glue httppaperlandmarkscomproductthep...	['pictwittercomdy9q4ftLhZ' 'What', 'to', 'do', 'with' 'paper', 'scissors', 'and' 'glue', 'htt..
4	4	#DidYouKnow ► Mahatma Gandhi made a brief visit to lecture in #Nottingham on 17 October 1931 [@M...	Quality	DidYouKnow ► Mahatma Gandhi made a brief visit to lecture in Nottingham on 17 October 1931 Mumbl...	['DidYouKnow', 'Mahatma' 'Gandhi', 'made', 'a', 'brief' 'visit', 'to', 'lecture', 'in' 'Notti..

```
import seaborn as sns
#Setting the size and grid for plotting
sns.set(rc= {"figure.figsize": (8, 6)})

bins = np.linspace(0, 200, 40)

plt.hist(data[data['Type']=='Spam']['body_len'], bins, alpha=0.5, density=True, label='spam')
plt.hist(data[data['Type']=='Quality']['body_len'], bins, alpha=0.5, density=True, label='ham')
plt.legend(loc='upper left')
plt.show()
```




```
bins = np.linspace(0, 40, 100)

plt.hist(data[data['Type']=='Spam']['punct%'], bins, alpha=0.5, density=True, label='Spam')
plt.hist(data[data['Type']=='Quality']['punct%'], bins, alpha=0.5, density=True, label='Ham')
plt.legend(loc='upper right')
plt.show()
```



Split into train/test

```
from sklearn.model_selection import train_test_split

from sklearn import metrics

X = data.body_text_clean
y = data.Type
print(X.shape)
print(y.shape)
```

```
(14899,)
(14899,)
```

```
# split X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(11174,)
(3725,)
(11174,)
(3725,)
```

Vectorize text

```
from sklearn.feature_extraction.text import CountVectorizer

# instantiate the vectorizer
vect = CountVectorizer()
vect.fit(X_train)
```

```
CountVectorizer()
```

```
# learn training data vocabulary, then use it to create a document-term matrix
X_train_dtm = vect.transform(X_train)

# equivalently: combine fit and transform into a single step
X_train_dtm = vect.fit_transform(X_train)

# examine the document-term matrix
X_train_dtm
```

```
<11174x30785 sparse matrix of type '<class 'numpy.int64'>'
        with 126337 stored elements in Compressed Sparse Row format>
```

```
# transform testing data (using fitted vocabulary) into a document-term matrix
X_test_dtm = vect.transform(X_test)
X_test_dtm
```

```
<3725x30785 sparse matrix of type '<class 'numpy.int64'>'
        with 34747 stored elements in Compressed Sparse Row format>
```

```
from sklearn.feature_extraction.text import TfidfTransformer

tfidf_transformer = TfidfTransformer()
tfidf_transformer.fit(X_train_dtm)
tfidf_transformer.transform(X_train_dtm)
```

```
<11174x30785 sparse matrix of type '<class 'numpy.float64'>'
        with 126337 stored elements in Compressed Sparse Row format>
```

Models

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
spam_classifier = DecisionTreeClassifier()
spam_classifier.fit(X_train_dtm, y_train)
# Use the classifier for predictions
predictions = spam_classifier.predict(X_test_dtm)

print(metrics.confusion_matrix(y_test, predictions))
print(metrics.accuracy_score(y_test, predictions))
print(metrics.classification_report(y_test, predictions))
```

```
[[1300  516]
 [ 595 1314]]
0.701744966442953
```

```
                  precision    recall  f1-score   support

         Quality       0.69      0.72      0.70      1816
            Spam       0.72      0.69      0.70      1909

        accuracy                           0.70      3725
       macro avg       0.70      0.70      0.70      3725
    weighted avg       0.70      0.70      0.70      3725
```

Random Forest

```
from sklearn.ensemble import RandomForestClassifier

spam_classifier = RandomForestClassifier(n_estimators=200, random_state=0)
spam_classifier.fit(X_train_dtm, y_train)
# Use the classifier for predictions
predictions = spam_classifier.predict(X_test_dtm)

print(metrics.confusion_matrix(y_test, predictions))
print(metrics.accuracy_score(y_test, predictions))
print(metrics.classification_report(y_test, predictions))
```

```
    [[1554  262]
     [ 688 1221]]
    0.7449664429530202
                  precision    recall  f1-score   support

         Quality       0.69      0.86      0.77      1816
            Spam       0.82      0.64      0.72      1909

        accuracy                           0.74      3725
       macro avg       0.76      0.75      0.74      3725
    weighted avg       0.76      0.74      0.74      3725
```

Multinomial Naive Bayers

```
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()

%time nb.fit(X_train_dtm, y_train)

y_pred_class = nb.predict(X_test_dtm)
print(y_pred_class)

from sklearn import metrics
print(metrics.accuracy_score(y_test, y_pred_class))

print(metrics.confusion_matrix(y_test, y_pred_class))
print(metrics.classification_report(y_test, y_pred_class))
```

```
    CPU times: user 22.4 ms, sys: 1.83 ms, total: 24.2 ms
    Wall time: 24.2 ms
    ['Spam' 'Spam' 'Spam' ... 'Spam' 'Quality' 'Quality']
    0.7865771812080536
    [[1530  286]
     [ 509 1400]]
                  precision    recall  f1-score   support

         Quality       0.75      0.84      0.79      1816
            Spam       0.83      0.73      0.78      1909

        accuracy                           0.79      3725
       macro avg       0.79      0.79      0.79      3725
    weighted avg       0.79      0.79      0.79      3725
```

Logistic Regression normal and probabilistic

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(solver='liblinear')

%time logreg.fit(X_train_dtm, y_train)
```

```
y_pred_class = logreg.predict(X_test_dtm)

print(metrics.accuracy_score(y_test, y_pred_class))
print(metrics.classification_report(y_test, y_pred_class))

y_pred_prob = logreg.predict_proba(X_test_dtm)[:, 1]

print(metrics.roc_auc_score(y_test, y_pred_prob))
```

```
CPU times: user 406 ms, sys: 300 ms, total: 706 ms
Wall time: 373 ms
0.781476510067114
              precision    recall  f1-score   support

     Quality       0.73      0.87      0.80      1816
        Spam       0.85      0.69      0.77      1909

    accuracy                           0.78      3725
   macro avg       0.79      0.78      0.78      3725
weighted avg       0.79      0.78      0.78      3725

0.49663243089192627
```

SVM

```
#Import svm model
from sklearn import svm

#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel

#Train the model using the training sets
clf.fit(X_train_dtm, y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test_dtm)

print(metrics.accuracy_score(y_test, y_pred))
print(metrics.classification_report(y_test, y_pred))
```

```
0.7702013422818792
              precision    recall  f1-score   support

     Quality       0.72      0.86      0.78      1816
        Spam       0.84      0.69      0.75      1909

    accuracy                           0.77      3725
   macro avg       0.78      0.77      0.77      3725
weighted avg       0.78      0.77      0.77      3725
```

Real Time data analysis

```
import tweepy

#Put your Bearer Token in the parenthesis below
client = tweepy.Client(bearer_token='AAAAAAAAAAAAAAAAAAAAAHbJhgEAAAAAb0Wbujbwy1VtUOx9Ytmd4YG5YtA%3DhOfhBfhiqdue61IidFzhFwESIqm8FqswWbHNtwDjwW

# Get tweets that contain the hashtag #petday
# -is:retweet means I don't want retweets
# lang:en is asking for the tweets to be in english
query = '#CSKvsRR  lang:En'
tweets = client.search_recent_tweets(query=query, tweet_fields=['context_annotations', 'created_at'], max_results=100)

l=[]

for tweet in tweets.data:
    print(tweet.text)
    l.append(tweet.text)
    print("\n")
```

```
 RT @Harish_NS149: Thala Dhoni’s 200 IPL Match as the Captain for #CSK

 #CSKvsRR #IPL2023 #Thala200 #Yellove https://t.co/XUaj3GGFKv
```

```
RT @sexycricketshot: Angry MSD 🔥

CSK vs RR always remind me of this moment. Rarest scene ever in Ms Dhoni's career. #CSKvsRR
https://t.co/…


RT @TheDhoniEra: MS Dhoni felicitated with a momento on his 200*th IPL match as captain 😍

@MSDhoni #IPL2023 #CSKvsRR https://t.co/vkTdGzcy…


RT @NetMeds: #ContestAlert 🏏- 250 lucky winners will win a FREE 6-month Netmeds First membership.*

Rules:
- Comment with the right answer…


RT @Harish_NS149: Thala Dhoni’s 200 IPL Match as the Captain for #CSK

#CSKvsRR #IPL2023 #Thala200 #Yellove https://t.co/XUaj3GGFKv


RT @spyankit07: @NetMeds Ans-B) Blood group O  #ContestAlert 🏏
#CSKvsRR  @ritujha89646416 @wani_aakil @GauravSuradkar @NetMeds


@NipponIndia Path - A

@NipponIndia
#BlobbysFanFrenzy #NipponPaintIPL2023 #NipponPaint #NipponPaintIndia #contest #contestalert #giveaway #contestgiveaway #CSK #IPL2023

Tags :
@SmartAnand07
@KDivya18740584
@starkumud
@LaxmiPatell
@ujjawalkumar26


RT @Pratham_editz: My favourite #CSKvsRR moment, yours?

https://t.co/8nhk0CLQJ2


RT @KamalSi0071: Ms dhoni as a captain achievement 👑
Chennai Super Kings

Matches = 199
Won = 127
Runs = 5018
Highest batting average 6…
```

1

```
['RT @Harish_NS149: Thala Dhoni’s 200 IPL Match as the Captain for #CSK\n\n#CSKvsRR #IPL2023 #Thala200 #Yellove
https://t.co/XUaj3GGFKv',
 "RT @sexycricketshot: Angry MSD 🔥\n\nCSK vs RR always remind me of this moment. Rarest scene ever in Ms Dhoni's career.
#CSKvsRR\nhttps://t.co/…",
 'RT @TheDhoniEra: MS Dhoni felicitated with a momento on his 200*th IPL match as captain 😍\n\n@MSDhoni #IPL2023 #CSKvsRR
https://t.co/vkTdGzcy…',
 'RT @NetMeds: #ContestAlert 🏏- 250 lucky winners will win a FREE 6-month Netmeds First membership.* \n\nRules: \n- Comment with the
right answer…',
 'RT @Harish_NS149: Thala Dhoni’s 200 IPL Match as the Captain for #CSK\n\n#CSKvsRR #IPL2023 #Thala200 #Yellove
https://t.co/XUaj3GGFKv',
 'RT @spyankit07: @NetMeds Ans-B) Blood group O  #ContestAlert 🏏 \n#CSKvsRR  @ritujha89646416 @wani_aakil @GauravSuradkar @NetMeds',
 '@NipponIndia Path - A\n\n@NipponIndia\n#BlobbysFanFrenzy #NipponPaintIPL2023 #NipponPaint #NipponPaintIndia #contest #contestalert
#giveaway #contestgiveaway #CSK #IPL2023 #CSKvsRR \n\nTags :\n@SmartAnand07\n@KDivya18740584 \n@starkumud \n@LaxmiPatell
\n@ujjawalkumar26',
 'RT @Pratham_editz: My favourite #CSKvsRR moment, yours?\n\nhttps://t.co/8nhk0CLQJ2',
 'RT @KamalSi0071: Ms dhoni as a captain achievement 👑\nChennai Super Kings \n\nMatches = 199 \nWon = 127 \nRuns = 5018 \nHighest
batting average 6…',
 '#CSKvsRR\n#MSDhoni\nMS Dhoni receives a special memento on his 200th match as captain for CSK..... https://t.co/P33iWSrbAY',
 '@NetMeds Ans-B) Blood group O  #ContestAlert 🏏 \n#CSKvsRR  @ritujha89646416 @wani_aakil @GauravSuradkar @NetMeds',
 'RT @SiddiqTulip: You should instill the same morals in your child as this girl has. 👇\n\n#earthquake #CSKvsRR #TejRan
#ChampionsLeague \n#Sury…',
 '#Trending #shoppingstar #Oriele\n#CSKvsRR #ShopMyCloset #NEW\n#shorts #TikTok #Amazonギフトカード #Amazon #viral #YouTube\n#Online
#twitch #rosecoco\n#ChampionsLeague #TAEYANG\n#shopping #Ukraine\nWallet for ₹199 on flipkart https://t.co/0LazhsAHji
https://t.co/0SxEn4yOn2',
 'Special Twitter Cover Pic Edit For Thala @msdhoni \n\n#CSKvsRR #MSDhoni #Dhoni #CSK #Yellove https://t.co/3jSUC86g6G',
 '@VibhuBhola Wish to win.☝🏻 \n#IPL2023 #CSKvsRR #dhoni \n#SanjuSamson \n@VibhuBhola',
 'RT @77thHundredwhxn: GET READY FOR "JAIL CLASSICO" 🔥\n#CSKvsRR https://t.co/pSfGvEMhhJ',
```

```
 'My favourite #CSKvsRR moment, yours? https://t.co/sJ83ZJ7pXq',
 'RT @Ganesh_0718_: Oh Captain, Our Captain! 😍💛\n\n200 Whistles on this special match.\nThala #MSDhoni🦁 all set to lead the
Yellovely army for 2…',
 '💛\n.\n.\nDo follow @DhoniDhevudu For More\n.\n.\n.\n.\n.\n#MSDhoni🦁 #ChennaiSuperKings #CSKvsRR #Captain
https://t.co/yozSwteaPs',
 'THALA 200 Match as CSK Captain today. \n#CSKvsRR https://t.co/Ue5CPJYHNw',
 'RT @NipponIndia: Quote this tweet with your answer, tag and follow @NipponIndia and Use #BlobbysFanFrenzy\n\n#NipponPaintIPL2023
#NipponPaint…',
 'MS Dhoni felicitation ahead of his 200th match as CSK captain\n#CSKvsRR #CSKvRR #SuryakumarYadav #RohitSharma #MumbaiIndians
#MumbaiIndia #GautamGambhir #TimDavid #ViratKohli #wtcfinal https://t.co/INvuJ3wkoU',
 'RT @pk_views: Match dayyy 🐯🐯🐯\n\n#CSKvsRR https://t.co/GvdOX7jCAZ',
 'RT @CricketFantasyS: #Dream11 RT if it helps u\n\nRayudu, Samson, Rutu, Spin\n\nMore in Thread ☝\nRefer Carefully\n\nMore in PRIME
📩✍\nHand Written…',
 "RT @sexycricketshot: Angry MSD 🔥\n\nCSK vs RR always remind me of this moment. Rarest scene ever in Ms Dhoni's career.
#CSKvsRR\nhttps://t.co/…",
 'RT @anchor_apoorv: 💵💵CSK Vs RR MEGA MONEY CONTEST 💵💵\n\nIf Sanju Scores 90+ Runs \n&amp;amp; \nMS Dhoni Takes 3 Catches, I will
Paytm 1500 rupees to ev…',
 'RT @2Rocky__420: How many CSK fans online RT 💛\n\n#CSKvsRR https://t.co/GzxbOQc1iR',
 'Ashwin chahal lam world class spinners \nIraiva 🙇\u200dد\n#CSKvsRR',
 "It is almost like MSD is present world's Amarendra Baahubali @msdhoni #CSKvsRR @ChennaiIPL #IPL2023",
 "It's Captain Cool's 200th appearance for Chennai. 🥳🔥\n\nEven better- it's on his home turf!\n\nTell us your favourite memory of
his plays in the League!👇\n\n#PlayT20 #CricketTwitter #CSKvsRR",
 'RT @SiddiqTulip: You should instill the same morals in your child as this girl has. 👇\n\n#earthquake #CSKvsRR #TejRan
#ChampionsLeague \n#Sury…',
 '9 Finals - 4 Trophies\nThe GOAT Captain Of IPL 💥\n.\n#MSDhoni #CSKvsRR #CSKvRR #MSD #RvcjTelugu https://t.co/7YQiA0B3Rz',
 'RT @ZahidHa68: M S Dhoni action in today match. 🔥🔥\nChennai Super Kings\n#MSDhoni🦁 #CSKvsRR #IPL2023 #MIvDC \n#SuryakumarYadav
#RohitSharma #T…',
 'RT @Pratham_editz: My favourite #CSKvsRR moment, yours?\n\nhttps://t.co/8nhk0CLQJ2',
 'RT @TheDhoniEra: MS Dhoni felicitated with a momento on his 200*th IPL match as captain 😍\n\n@MSDhoni #IPL2023 #CSKvsRR
https://t.co/vkTdGzcy…',
 'RT @Pratham_editz: My favourite #CSKvsRR moment, yours?\n\nhttps://t.co/8nhk0CLQJ2',
```

```
df = pd.DataFrame({'tweets':l})
```

```
df
```

	tweets
0	RT @Harish_NS149: Thala Dhoni's 200 IPL Match as the Captain for #CSK\n\n#CSKvsRR #IPL2023 #Thal...
1	RT @sexycricketshot: Angry MSD 🔥\n\nCSK vs RR always remind me of this moment. Rarest scene ever...
2	RT @TheDhoniEra: MS Dhoni felicitated with a momento on his 200*th IPL match as captain 😍\n\n@MS...
3	RT @NetMeds: #ContestAlert 🏏- 250 lucky winners will win a FREE 6-month Netmeds First membership....
4	RT @Harish_NS149: Thala Dhoni's 200 IPL Match as the Captain for #CSK\n\n#CSKvsRR #IPL2023 #Thal...
...	...
95	RT @akhilkautilya1: #SiddhiIdnani letest 📷😍\n\n#Adi #ShahRukhKhan #SoppanaSundari #Rudhran #Keert...
96	#thala giving tips to #pathirana 🔥🔥🔥🔥🔥🔥🔥 #CSKvsRR #CSK #msd #MSDhoni🦁 #ipl #IPL2023 #ThalapathyV...
97	RT @IamPoojaSingh2: Viral Video from Jaipur\n\nMuslims in India🇮🇳👇\nMaximum share for the punish...
98	RT @sexycricketshot: Angry MSD 🔥\n\nCSK vs RR always remind me of this moment. Rarest scene ever...
99	RT @sexycricketshot: Angry MSD 🔥\n\nCSK vs RR always remind me of this moment. Rarest scene ever...

100 rows × 1 columns

```
df.describe()
```

	tweets
count	100
unique	64

```
# transform testing data (using fitted vocabulary) into a document-term matrix
X_test_real = vect.transform(df['tweets'])
X_test_real
```

```
<100x30785 sparse matrix of type '<class 'numpy.int64'>'
        with 998 stored elements in Compressed Sparse Row format>
```

```
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()

%time nb.fit(X_train_dtm, y_train)

y_pred_class = nb.predict(X_test_real)
print(y_pred_class)
```

```
CPU times: user 33.5 ms, sys: 0 ns, total: 33.5 ms
Wall time: 43 ms
['Spam' 'Spam' 'Spam' 'Quality' 'Spam' 'Spam' 'Quality' 'Spam' 'Spam'
 'Spam' 'Spam' 'Spam' 'Quality' 'Spam' 'Quality' 'Spam' 'Quality' 'Spam'
 'Spam' 'Spam' 'Quality' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam'
 'Quality' 'Quality' 'Spam' 'Quality' 'Spam' 'Spam' 'Spam' 'Spam'
 'Quality' 'Quality' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam'
 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam'
 'Spam' 'Spam' 'Spam' 'Quality' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam'
 'Spam' 'Quality' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Quality' 'Spam'
 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Spam' 'Quality'
 'Spam' 'Spam' 'Spam' 'Spam' 'Quality' 'Spam' 'Quality' 'Spam' 'Spam'
 'Spam' 'Spam' 'Spam' 'Quality' 'Spam' 'Spam' 'Spam']
```