```
# importing essentials to guide machine learning model
# stand 2 for email spam detection

import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
df= pd.read_csv("spam.csv",encoding='ISO-8859-1')
```

```
df.head()
```

Out[4]:

	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

```
# dropping unnamed columns
df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)
```

```
df.head()
```

Out[6]:

	v1	v2
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...

```
# detecting nul values of name and spaces
df.isnull().sum(axis=1)
```

Out[7]:

```
ham    4825
spam    747
Name: v1, dtype: int64
```

```
df['v2'].value_counts
```

Out[8]:

```
<bound method IndexOpsMixin.value_counts of 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...
...                                                   ...
5564    This is the 2nd time we have tried 2 contact u...
5565                 Will ì_ b going to esplanade fr home?
5566    Pity, * was in mood for that. So any other s...
5567    The guy did some bitching but I acted like i'd...
5568                         Rofl. Its true to its name
Name: v2, Length: 5572, dtype: object>
```

```
# renaming column headings for better search
df.rename(){'v1':'category', 'v2':'message'},axis=1,inplace=True)
df.head()
```

Out[9]:

	category	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...

```
# given info about shape of the data set
df.shape
```

Out[10]:

```
(5572, 2)
```

```
df.info()
```

Out[11]:

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

```
# check percentage of data that needs to be balanced
print(str(round(747/4825,2)*100))
```

Out[12]:

```
15.0
```

```
df.category.value_counts()
```

Out[13]:

```
ham     4825
spam     747
Name: category, dtype: int64
```

```
# dropping for null values
df.isnull().sum()
```

Out[14]:

```
category    0
message     0
dtype: int64
```

```
# checking for duplicacy
df.duplicated().sum()
```

Out[15]:

```
403
```

```
# dprop duplicates
df = df.drop_duplicates(keep = 'first',inplace=True)
```

```
# initial shape of dataset
df.shape
```

Out[17]:

```
(5169, 2)
```

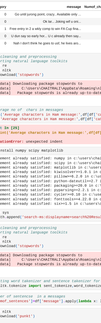
```
df['category'].value_counts(normalize=True)
```

Out[18]:

```
ham    0.87367
spam   0.12633
Name: category, dtype: float64
```

```
plt.figure(figsize=(8,6))
sns.countplot(df['category'])
plt.title("Count of spam and non-spam messages")
for bar, label in zip(ax.patches, df['category'].value_counts()):
    ax.text(bar.get_x()+0.3, bar.get_height()+50, str(round(bar.get_height()/5169*100,2))+'%', fontsize=15, color='black')
plt.xlabel("category")
plt.ylabel("Count")
```

Out[19]:




```
plt.pie(df['category'].value_counts(),labels=['ham','spam'],autopct='%0.2f%%',explode=[0.1,0])
```

Out[20]:


```
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\hsai\anaconda3\lib\site-packages (3.7)
Requirement already satisfied: tqdm in c:\users\hsai\anaconda3\lib\site-packages (from nltk) (4.64.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\hsai\anaconda3\lib\site-packages (from nltk) (2022.3.15)
Requirement already satisfied: joblib in c:\users\hsai\anaconda3\lib\site-packages (from nltk) (1.1.0)
Requirement already satisfied: click in c:\users\hsai\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: colorama in c:\users\hsai\anaconda3\lib\site-packages (from click->nltk) (0.4.4)
```

```
# nbdca cleaning and preprocessing
# stripping natural language tookits
import re
import nltk
nltk.download('stopwords')
```

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

Out[22]:

```
True
```

```
# tokenizing number of characters in message
df['Number_characters'].apply(lambda x: len(x))
df.head()
```

Out[23]:

	category	message	Number_characters
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

```
# nbdca cleaning and preprocessing
# stripping natural language tookits
import re
import nltk
nltk.download('stopwords')
```

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

Out[24]:

```
True
```

```
# per average no of chars in messages
print('Average characters in ham message',df[df['category']==0]['Number_characters'].mean())
print('Average characters in spam message',df[df['category']==1]['Number_characters'].mean())
```

```
Average characters in ham message 70.91660221588273
Average characters in spam message 137.891271056312
```

```
# Input In [26]:
    print('Average characters in ham message',df[df['category']==0]['Number_characters'].mean())

IndentationError: unexpected indent
```

```
!pip install numpy matplotlib
```

```
Requirement already satisfied: numpy in c:\users\hsai\anaconda3\lib\site-packages (1.21.5)
Requirement already satisfied: matplotlib in c:\users\hsai\anaconda3\lib\site-packages (1.5.3)
Requirement already satisfied: cycler>=0.10 in c:\users\hsai\anaconda3\lib\site-packages (0.11)
Requirement already satisfied: fonttools>=4.0.1 in c:\users\hsai\anaconda3\lib\site-packages (from matplotlib) (3.3.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hsai\anaconda3\lib\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hsai\anaconda3\lib\site-packages (from matplotlib) (1.4.2)
Requirement already satisfied: packaging>=20.0 in c:\users\hsai\anaconda3\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\hsai\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hsai\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\hsai\anaconda3\lib\site-packages (from pytz-deprecated-tz3) (2022.1)
Requirement already satisfied: tzlocal>=1.5.1 in c:\users\hsai\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (5.10.0)
```

```
import sys
sys.path.append('search-ms:displaynames=%22%5bC%3A%2fUsers%2fCHAITRALI%2fAppDataLocal%2fLAUNCHCLICK%2f')
```

```
# nbdca cleaning and preprocessing
# stripping natural language tookits
import re
import nltk
nltk.download('stopwords')
```

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

Out[28]:

```
True
```

```
# tokenizing word tokenizer and sentence tokenizer for finding no of words and sentences in a messages
from nltk.tokenize import sent_tokenize,word_tokenize
```

```
# tokenizer of sentence in a messages
df['Number_sentences']['df['message'].apply(lambda x: len(nltk.word_tokenize(x)))
```

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

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

Out[30]:

```
True
```

```
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\hsai\anaconda3\lib\site-packages (3.7)
Requirement already satisfied: regex>=2021.8.3 in c:\users\hsai\anaconda3\lib\site-packages (from nltk) (2022.3.15)
Requirement already satisfied: tqdm in c:\users\hsai\anaconda3\lib\site-packages (from nltk) (4.64.0)
Requirement already satisfied: joblib in c:\users\hsai\anaconda3\lib\site-packages (from nltk) (1.1.0)
Requirement already satisfied: click in c:\users\hsai\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: colorama in c:\users\hsai\anaconda3\lib\site-packages (from click->nltk) (0.4.4)
```

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

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

Out[32]:

```
True
```

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

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

Out[33]:

```
True
```

```
import re
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.corpus import stopwords
```

```
from nltk.tokenize import sent_tokenize, word_tokenize
```

```
# tokenizer of sentences in a messages
df['Number_sentences'] = df['message'].apply(lambda x: len(nltk.sent_tokenize(x)))
```

```
# tokenizer of word in a messages
df['Number_sentences'] = df['message'].apply(lambda x: len(nltk.word_tokenize(x)))
```

```
df.head()
```

Out[38]:

	category	message	Number_characters	Number_sentences	
0	ham	Go until jurong point, crazy.. Available only ...	111	24	
1	ham	Ok lar... Joking wif u oni...	29	8	
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	
3	ham	U dun say so early hor... U c already then say...	49	13	
4	ham	Nah I don't think he goes to usf, he lives aro...	61	15	

```
# statistical summary of ham messages
df [df['category']==0].[['Number_characters', 'Number_sentences']].describe()
```

Out[39]:

	Number_characters	Number_sentences
count	4516.000000	4516.000000
mean	70.459256	17.123782
std	56.358207	13.493725
min	2.000000	1.000000
25%	34.000000	8.000000
50%	52.000000	13.000000
75%	90.000000	22.000000
max	910.000000	220.000000

```
# statistical summary of ham spam
df[df['category']==1].[['Number_characters', 'Number_sentences']].describe()
```

Out[40]:

	Number_characters	Number_sentences
count	653.000000	653.000000
mean	137.891271	27.467688
std	30.137753	7.008418
min	13.000000	2.000000
25%	132.000000	25.000000
50%	149.000000	29.000000
75%	157.000000	32.000000
max	224.000000	46.000000

```
ham =df[df['category']==0].['Number_characters']
spam =df[df['category']==1].['Number_characters']
plt.figure(figsize=(8,6))
sns.hist(ham,ham_colors='b',label='ham')
sns.hist(spam,spam_colors='r',label='spam')
plt.title('Number of characters')
plt.xlabel('Length')
plt.ylabel('Count')
plt.legend();
```

Out[41]:


```
print(df[df['category']==0].['Number_characters'].skew())
print(df[df['category']==0].['Number_characters'].kurtosis())
```

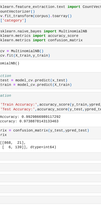
Out[42]:

```
1.8094264171341539
4.7132837993351211
```

```
plt.figure(figsize=(12,8))
sns.hist(plot=df[df['category']==0].['Number_characters'],color='yellow')
sns.hist(plot=df[df['category']==1].['Number_characters'],color='blue')
```

```
<bxes(ubplot:ylabels'Number_characters', ylabel='Count'>
```

Out[43]:




```
plt.figure(figsize=(8,6))
figure = sns.boxplot(x='category',y='Number_characters',data=df)
```

Out[44]:


```
plt.figure(figsize=(8,6))
figure = sns.boxplot(x='category',y='Number_sentences',data=df)
```

Out[45]:


```
# nbdca preprocessing
```

```
import nltk
import re
from nltk.corpus import stopwords

# import these modules
from nltk.stem import PorterStemmer
ps = PorterStemmer()
```

```
df['message'][0]
```

Out[47]:

```
'Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...'
```

```
# removing stopwords, punchuation,special characters and applying porterstemming
```

```
corpus = []
for i in range(len(df)):
    s = re.sub('[^a-zA-Z]+', " ",df.loc[i,'message'])
    s = s.lower()
    s = s.split()
    s = [ps.stem(word)for word in s if not word in set (stopwords.words('english'))]
    s = " ".join(s)
    corpus.append(s)
```

```
# tokenizing text data splits
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split( x1, y, test_size=0.2, random_state=42)
```

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=2500)
x1 = cv.fit_transform(corpus).toarray()
y = df['category']
```

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

```
model_cv = MultinomialNB()
model_cv.fit(X_train,y_train)
```

```
MultinomialNB()
```

```
# prediction
y_pred_train = model_cv.predict(X_train)
cpred_train = model_cv.predict(X_train)
```

```
# evaluation
```

```
print('Train Accuracy:',accuracy_score(y_train,cpred_train))
print('Test Accuracy:', accuracy_score(y_test,cpred_test))
```

```
Train Accuracy: 0.9929966031188
Test Accuracy: 0.9738878143133462
```

```
cf_matrix = confusion_matrix(y_test,cpred_test)
cf_matrix
```

```
array([[965,  21],
       [  8, 120]], dtype=int64)
```