

```
In [1]: #importing essentials to guide machine learning model
#task 2 for email spam detection

import pandas as pd
import numpy as np
import os
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

In [3]: df= pd.read_csv("spam.csv",encoding="ISO-8859-1")

In [4]: 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 dont think he goes to usf, he lives aro...      NaN      NaN      NaN

In [5]: # dropping unnamed columns
df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)

In [6]: 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 dont think he goes to usf, he lives aro...

In [7]: #checking for no. of hams and spams
df['v1'].value_counts()

Out[7]:
ham      4825
spam      747
Name: v1, dtype: int64

In [8]: df['v2'].value_counts()
<bound method IndexOpsMixin.value_counts of 0      Go until jurong point, crazy.. Available only ...
1      Free entry in 2 a wkly comp to win FA Cup fina...
2      U dun say so early hor... U c already then say...
3      Nah I dont't think he goes to usf, he lives aro...
4      ...
5567     This is the 2nd time we have tried 2 contact u...
5568     Will I b going to esplanade fr home?
5569     Pity, * was in mood for that. So...any other s...
5570     The guy did some bitching but I acted like l'd...
5571     Rofl. Its true to its name
Name: v2, Length: 5572, dtype: object>

In [9]: # 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 dont think he goes to usf, he lives aro...

In [10]: ## gives info. about shaoe of the data set
df.shape

Out[10]:
(5572, 2)

In [11]: df.info()
<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

In [12]: # check percentage of data that needs to be balanced
print(str(round(747/4825,2))+'%')
0.15%

In [13]: # drop duplicacy
df['category'] = pd.get_dummies(df['category'], drop_first=True)

In [14]: #checking for null values
df.isnull().sum()

Out[14]:
category      0
message      0
dtype: int64

In [15]: # checking for duplicacy
df.duplicated().sum()

Out[15]:
403

In [16]: #drop duplicates
df = df.drop_duplicates(keep = 'first', ignore_index=True)

In [17]: #final shape of dataset
df.shape

Out[17]:
(5169, 2)

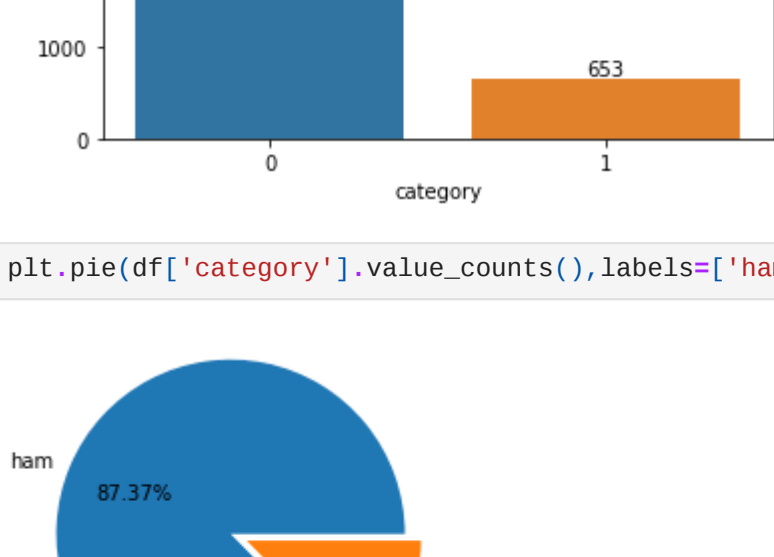
In [ ]:

In [18]: df['category'].value_counts(normalize=True)

Out[18]:
0      0.87367
1      0.12633
Name: category, dtype: float64

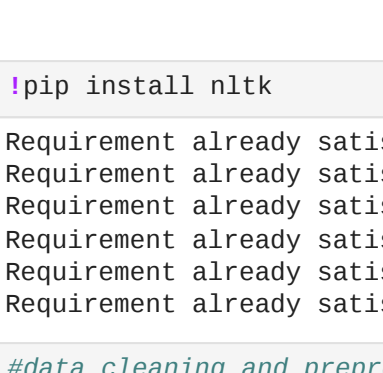
In [19]: plt.figure(figsize=(4,3))
ax = sns.countplot(df['category'])
plt.title("Count of spam and non-spam messages")
for label in ax.containers:
    ax.bar_label(label)

Count of spam and non-spam messages



In [20]: plt.pie(df['category'].value_counts(), labels=['ham', 'spam'], autopct='%0.2f%%', explode=[0.1,0]);

ham
spam



In [21]: !pip install nltk
Requirement already satisfied: nltk in c:\users\chaitrali\anaconda3\lib\site-packages (3.7)
Requirement already satisfied: tqdm in c:\users\chaitrali\anaconda3\lib\site-packages (from nltk) (4.64.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\chaitrali\anaconda3\lib\site-packages (from nltk) (2022.3.15)
Requirement already satisfied: joblib in c:\users\chaitrali\anaconda3\lib\site-packages (from nltk) (1.1.0)
Requirement already satisfied: click in c:\users\chaitrali\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: colorama in c:\users\chaitrali\anaconda3\lib\site-packages (from click->nltk) (0.4.4)

In [22]: #data cleaning and preprocessing
# importing natural language toolkits
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!
True

In [ ]:

In [23]: #including number of characters in message
df['Numof_characters']=df['message'].apply(len)
df.head()

Out[23]:
   category      message  Numof_characters
0         0  Go until jurong point, crazy.. Available only ...      111
1         0                Ok lar... Joking wif u oni...       29
2         1  Free entry in 2 a wkly comp to win FA Cup fina...     155
3         0  U dun say so early hor... U c already then say...      49
4         0  Nah I dont think he goes to usf, he lives aro...      61

In [24]: #data cleaning and preprocessing
# importing natural language toolkits
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!
True

In [25]: # average no of chars in messages
print('Average characters in Ham message:',df[df['category']==0]['Numof_characters'].mean())
print('Average characters in Ham message:',df[df['category']==1]['Numof_characters'].mean())

Input In [25]
print('Average characters in Ham message:',df[df['category']==0]['Numof_characters'].mean())
^
IndentationError: unexpected indent

In [26]: !pip install numpy scipy matplotlib
Requirement already satisfied: numpy in c:\users\chaitrali\anaconda3\lib\site-packages (1.21.5)
Requirement already satisfied: scipy in c:\users\chaitrali\anaconda3\lib\site-packages (1.7.3)
Requirement already satisfied: matplotlib in c:\users\chaitrali\anaconda3\lib\site-packages (3.5.1)
Requirement already satisfied: tqdm in c:\users\chaitrali\anaconda3\lib\site-packages (from matplotlib) (4.3.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\chaitrali\anaconda3\lib\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\chaitrali\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: packaging>=20.0 in c:\users\chaitrali\anaconda3\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\chaitrali\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: cycler>=0.10 in c:\users\chaitrali\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\chaitrali\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: sio>=1.5 in c:\users\chaitrali\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.10.0)

In [27]: import sys
sys.path.append('search-ms:displayname=search%20results%20in%20MAININ%20(C%3A)&crumb=location:C%3A%5Cnltk')

In [28]: #data cleaning and preprocessing
# importing natural language toolkits
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!
True

In [29]: #importing word tokenizer and sentence tokenizer for finding no of words and sentennces in a messages
from nltk.tokenize import sent_tokenize,word_tokenize

In [30]: #number of sentence in a messages
df['Numof_sentences']=df['message'].apply(lambda x: len(nltk.word_tokenize(x)))

In [31]: 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!
True

In [32]: !pip install nltk
Requirement already satisfied: nltk in c:\users\chaitrali\anaconda3\lib\site-packages (3.7)
Requirement already satisfied: regex>=2021.8.3 in c:\users\chaitrali\anaconda3\lib\site-packages (from nltk) (2022.3.15)
Requirement already satisfied: tqdm in c:\users\chaitrali\anaconda3\lib\site-packages (from nltk) (4.64.0)
Requirement already satisfied: click in c:\users\chaitrali\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: joblib in c:\users\chaitrali\anaconda3\lib\site-packages (from nltk) (1.1.0)
Requirement already satisfied: colorama in c:\users\chaitrali\anaconda3\lib\site-packages (from click->nltk) (0.4.4)

In [33]: 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!
True

In [34]: 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!
True

In [35]: import re
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords

In [36]: from nltk.tokenize import sent_tokenize, word_tokenize

In [37]: #number of sentences in a messages
df['Numof_sentences'] = df['message'].apply(lambda x: len(nltk.sent_tokenize(x)))

In [38]: #number of word in a messages
df['Numof_sentences'] = df['message'].apply(lambda x: len(nltk.word_tokenize(x)))

In [39]: df.head()

Out[39]:
   category      message  Numof_characters  Numof_sentences
0         0  Go until jurong point, crazy.. Available only ...      111          24
1         0                Ok lar... Joking wif u oni...       29           8
2         1  Free entry in 2 a wkly comp to win FA Cup fina...     155          37
3         0  U dun say so early hor... U c already then say...      49          13
4         0  Nah I dont think he goes to usf, he lives aro...      61           15

In [40]: #statistical summary of ham messages
df[df['category']==0][['Numof_characters','Numof_sentences']].describe()

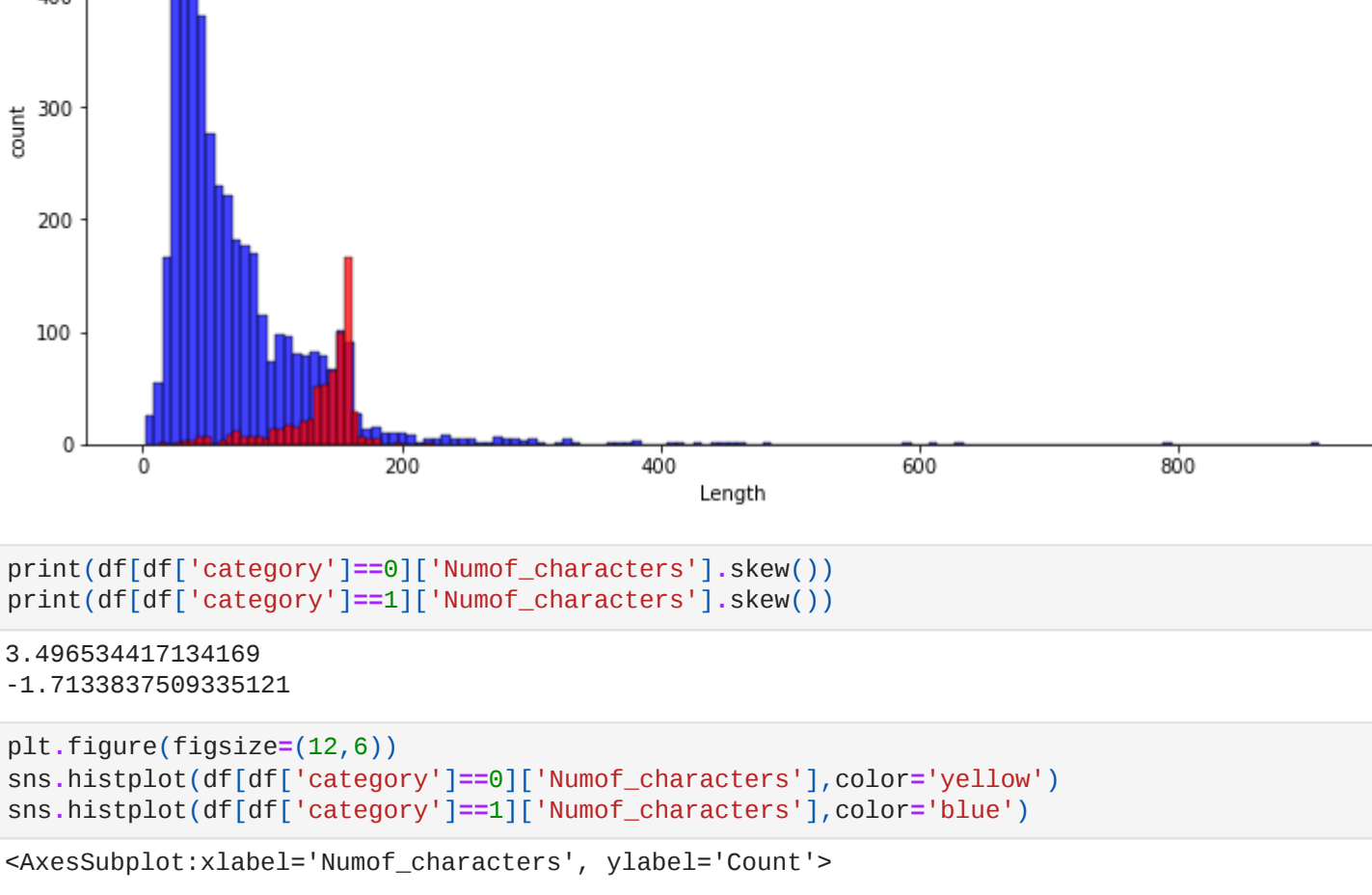
Out[40]:
   Numof_characters  Numof_sentences
count      4516.000000      4516.000000
mean         70.459256        17.120903
std         56.358207        13.493725
min           2.000000         1.000000
25%          34.000000         8.000000
50%          52.000000        13.000000
75%          90.000000        22.000000
max         910.000000        220.000000

In [41]: #statistical summary of ham spam
df[df['category']==1][['Numof_characters','Numof_sentences']].describe()

Out[41]:
   Numof_characters  Numof_sentences
count           653.000000           653.000000
mean        137.891271          27.667688
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

In [42]: ham=df[df['category']==0][['Numof_characters']]
spam=df[df['category']==1][['Numof_characters']]
plt.figure(figsize=(12,6))
sns.histplot(ham,color='b',label='Ham')
sns.histplot(spam,color='r',label='spam')
plt.title('Number of characters')
plt.xlabel('length')
plt.ylabel('count')
plt.legend();

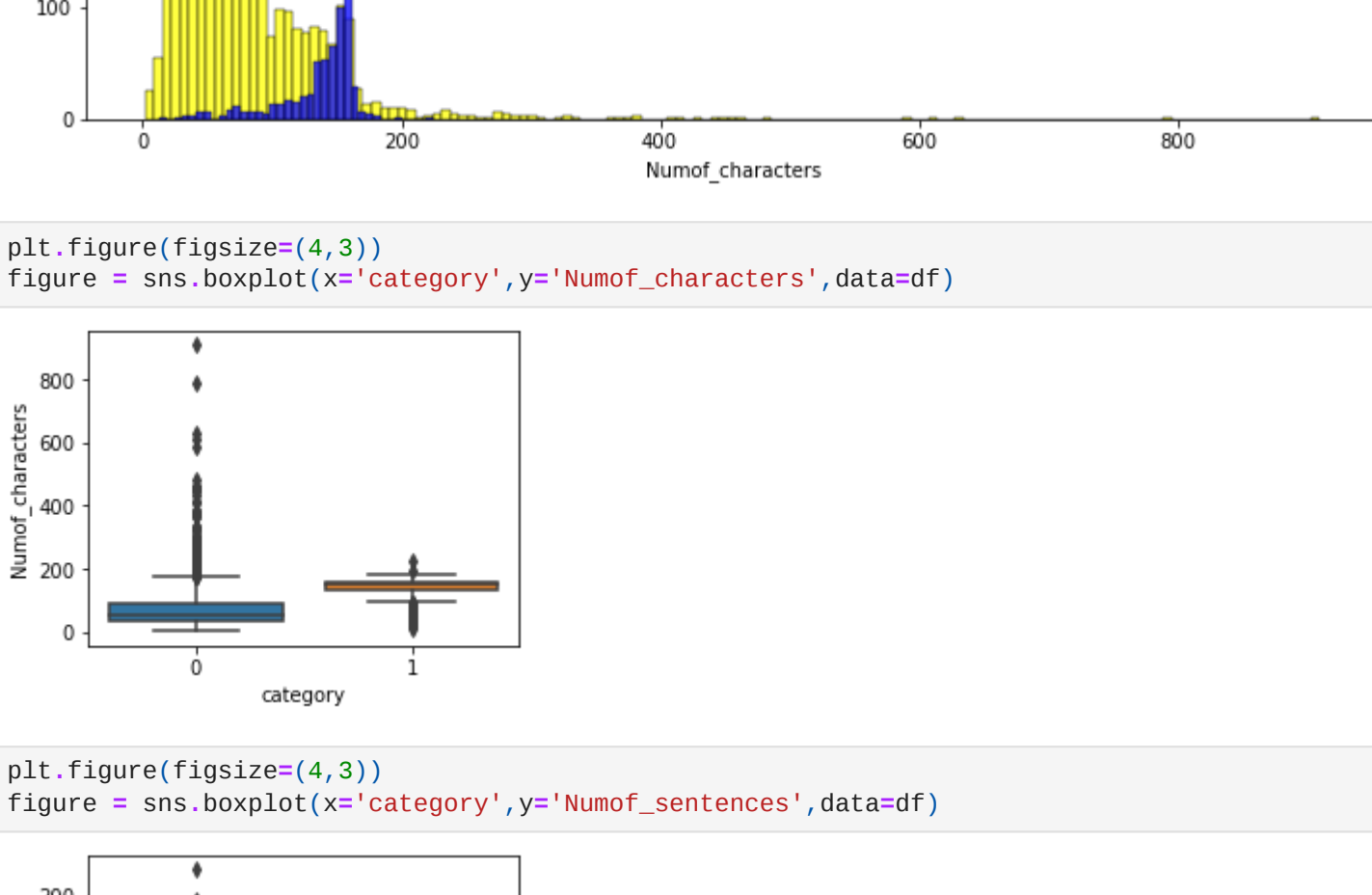
Number of characters



In [43]: print(df[df['category']==0][['Numof_characters']].skew())
print(df[df['category']==1][['Numof_characters']].skew())
3.496534417134169
-1.713383750935121

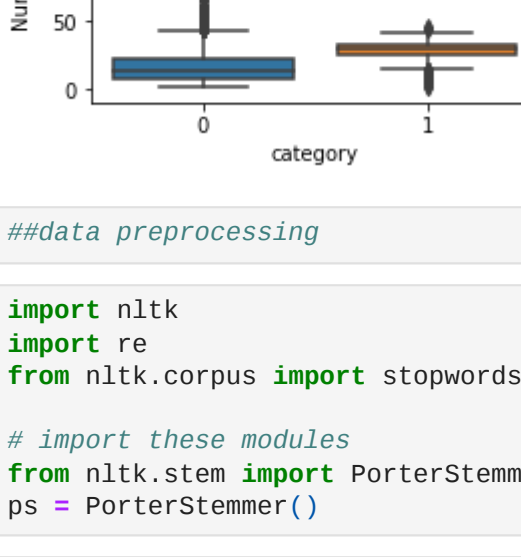
In [44]: plt.figure(figsize=(12,6))
sns.histplot(df[df['category']==0][['Numof_characters']],color='yellow')
sns.histplot(df[df['category']==1][['Numof_characters']],color='blue')
<AxesSubplot:xlabel='Numof_characters', ylabel='Count'>

Count



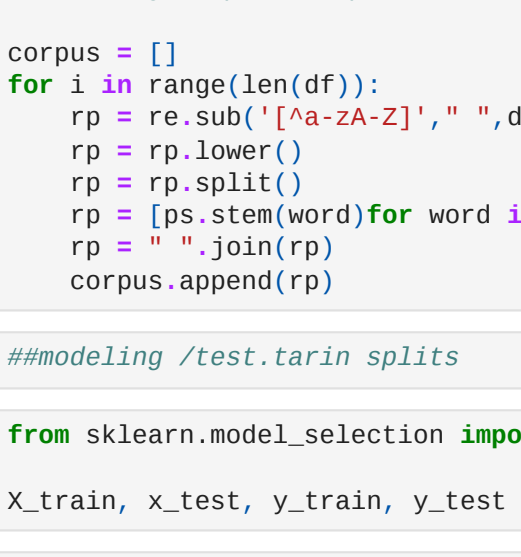
In [45]: plt.figure(figsize=(4,3))
figure = sns.boxplot(x='category',y='Numof_characters',data=df)

Numof_characters



In [46]: plt.figure(figsize=(4,3))
figure = sns.boxplot(x='category',y='Numof_sentences',data=df)

Numof_sentences



In [47]: ##data preprocessing

In [68]: import nltk
import re
from nltk.corpus import stopwords

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

In [69]: df['message'][0]

Out[69]:
'Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...'

In [70]: #removing stopwords, punctuation, special characters and applying porterstemming
corpus = []
for i in range(len(df)):
    rp = re.sub('[^a-zA-Z]'," ",df.loc[i,'message'])
    rp = rp.lower()
    rp = ps.stem(rp)
    rp = ps.stem(rp)
    for word in rp if not word in set (stopwords.words('english'))]]
    corpus.append(rp)

In [71]: ##modeling /test.tarin splits

In [73]: 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)

In [74]: from sklearn.feature_extraction.text import CountVecorizer
cv = CountVecorizer()
x1 = cv.fit_transform(corpus).toarray()
y = df['category']

In [75]: from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix

In [76]: model_cv = MultinomialNB()
model_cv.fit(X_train,y_train)

Out[76]:
MultinomialNB()

In [77]: #prediction
ypred_test = model_cv.predict(x_test)
ypred_train = model_cv.predict(X_train)

In [78]: #evaluation
print('Train Accuracy:',accuracy_score(y_train,ypred_train))
print('Test Accuracy:',accuracy_score(y_test,ypred_test))
Train Accuracy: 0.9929866989117292
Test Accuracy: 0.9738878143133463

In [79]: cf_matrix = confusion_matrix(y_test,ypred_test)
cf_matrix

Out[79]:
array([[ 868, 21],
       [ 6, 139]], dtype=int64)

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```