In [1]:

**import** numpy **as** np **import** pandas **as** pd *# import nltk*

In [2]:

df\_sms**=**pd.read\_csv('spam.csv') df\_sms.head()

Out[2]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **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 | [3]: |  |  |  |  |

df\_sms**=**df\_sms.drop(["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"] , axis**=**1) df\_sms**=**df\_sms.rename(columns**=**{"v1":"label","v2":"sms-text"})

In [4]:

df\_sms.head()

Out[4]:

**label sms-text**

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

In [5]:

print(len(df\_sms))

5572

In [6]:

df\_sms.shape

Out[6]:

(5572, 2)

In [7]:

df\_sms.tail(5)

Out[7]:

|  |  |  |
| --- | --- | --- |
|  | **label** | **sms-text** |
| **5567** | spam | This is the 2nd time we have tried 2 contact u... |
| **5568** | ham | Will �\_ b going to esplanade fr home? |
| **5569** | ham | Pity, \* was in mood for that. So...any other s... |
| **5570** | ham | The guy did some bitching but I acted like i'd... |
| **5571** | ham | Rofl. Its true to its name |

In [8]:

df\_sms.label.value\_counts()

Out[8]:

ham 4825

spam 747

Name: label, dtype: int64

In [9]:

df\_sms.head()

Out[9]:

**label sms-text**

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

In [10]:

df\_sms.describe()

Out[10]:

|  |  |  |
| --- | --- | --- |
|  | **label** | **sms-text** |
| **count** | 5572 | 5572 |
| **unique** | 2 | 5169 |
| **top** | ham | Sorry, I'll call later |
| **freq** | 4825 | 30 |

df\_sms['length']**=**df\_sms['sms-text'].apply(len)

In [12]:

df\_sms.head(3)

Out[12]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **label** | **sms-text** | **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 |
| In | [13]: |  |  |

**import** matplotlib.pyplot **as** plt

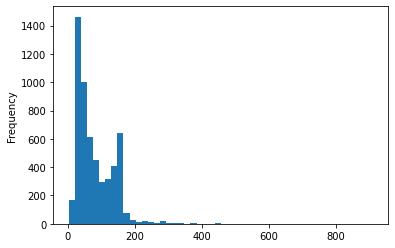
**import** seaborn **as** sns

**%**matplotlib inline

df\_sms['length'].plot(bins**=**50,kind**=**'hist')

Out[13]:

<AxesSubplot:ylabel='Frequency'>



In [14]:

*#Implemenation of Bag of words approach*

*#step1:Convert all strings to their lower case form*

documents**=**['Hello,how are you!',

'Win money ,win from home.', 'Call me now.'

'Hello,Call hello you tomorrow? '] lower\_case\_documents**=**[]

lower\_case\_documents**=**[d.lower() **for** d **in** documents] print(lower\_case\_documents)

['hello,how are you!', 'win money ,win from home.', 'call me now.hello,call hello you tomorrow? ']

In [16]:

*# Step 2:Removing all punctuations*

sans\_punctuation\_documents**=**[]

**import** string

**for** i **in** lower\_case\_documents:

sans\_punctuation\_documents.append(i.translate(str.maketrans("","",string.punctuation)))

In [17]:

sans\_punctuation\_documents

Out[17]:

['hellohow are you',

'win money win from home',

'call me nowhellocall hello you tomorrow ']

In [18]:

*#Step 3:Tokenization*

preprocessed\_documents**=**[[w **for** w **in** d.split()] **for** d **in** sans\_punctuation\_documents] preprocessed\_documents

Out[18]:

[['hellohow', 'are', 'you'],

['win', 'money', 'win', 'from', 'home'],

['call', 'me', 'nowhellocall', 'hello', 'you', 'tomorrow']]

In [19]:

*#Step 4:Count frequencies*

frequency\_list**=**[]

**import** pprint

**from** collections **import** Counter

frequency\_list**=**[Counter (d)**for** d **in** preprocessed\_documents ] pprint.pprint(frequency\_list)

[Counter({'hellohow': 1, 'are': 1, 'you': 1}),

Counter({'win': 2, 'money': 1, 'from': 1, 'home': 1}),

Counter({'call': 1,

'me': 1,

'nowhellocall': 1,

'hello': 1,

'you': 1,

'tomorrow': 1})]

doc\_array**=**frequency\_list doc\_array

Out[20]:

[Counter({'hellohow': 1, 'are': 1, 'you': 1}),

Counter({'win': 2, 'money': 1, 'from': 1, 'home': 1}),

Counter({'call': 1,

'me': 1,

'nowhellocall': 1,

'hello': 1,

'you': 1,

'tomorrow': 1})]

In [21]:

**from** sklearn.feature\_extraction.text **import** CountVectorizer count\_vector**=**CountVectorizer()

In [22]:

count\_vector.fit(documents)

count\_vector.get\_feature\_names\_out()

Out[22]:

array(['are', 'call', 'from', 'hello', 'home', 'how', 'me', 'money', 'now', 'tomorrow', 'win', 'you'], dtype=object)

In [23]:

doc\_array**=**count\_vector.transform(documents).toarray() doc\_array

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[23]: |  | | | | | | | | | | |
| array([[1, | 0, | 0, | 1, | 0, | 1, | 0, | 0, | 0, | 0, | 0, | 1], |
| [0, | 0, | 1, | 0, | 1, | 0, | 0, | 1, | 0, | 0, | 2, | 0], |
| [0, | 2, | 0, | 2, | 0, | 0, | 1, | 0, | 1, | 1, | 0, | 1]], dtype=int64) |

In [24]:

frequency\_matrix**=**pd.DataFrame(doc\_array,columns**=**count\_vector.get\_feature\_names\_out()) frequency\_matrix

Out[24]:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **are** | **call** | **from** | **hello** | **home** | **how** | **me** | **money** | **now** | **tomorrow** | **win** | **you** |
| **0** 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| **1** 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 | 0 |
| **2** 0 | 2 | 0 | 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |

**from** sklearn.model\_selection **import** train\_test\_split

X\_train,X\_test,y\_train,y\_test**=**train\_test\_split(df\_sms['sms-text'],df\_sms['label'],test\_size

In [26]:

count\_vector**=**CountVectorizer()

In [27]:

training\_data**=**count\_vector.fit\_transform(X\_train)

In [28]:

training\_data

Out[28]:

<4457x7733 sparse matrix of type '<class 'numpy.int64'>'

with 59215 stored elements in Compressed Sparse Row format>

In [29]:

testing\_data**=**count\_vector.transform(X\_test)

In [30]:

**from** sklearn.naive\_bayes **import** MultinomialNB naive\_bayes**=**MultinomialNB()

naive\_bayes.fit(training\_data,y\_train)

Out[30]:

MultinomialNB()

In [31]:

predictions**=**naive\_bayes.predict(testing\_data)

In [32]:

predictions

Out[32]:

array(['ham', 'ham', 'ham', ..., 'ham', 'ham', 'ham'], dtype='<U4')

In [33]:

**from** sklearn.metrics **import** accuracy\_score,precision\_score,recall\_score,f1\_score print('Accuracy score:{}'.format(accuracy\_score(y\_test,predictions)))

Accuracy score:0.9856502242152466

In [34]:

print('Precision score:{}'.format(precision\_score(y\_test,predictions,pos\_label**=**'spam'))) print('Recall score:{}'.format(recall\_score(y\_test,predictions,pos\_label**=**'spam')))

print('f1 score:{}'.format(f1\_score(y\_test,predictions,pos\_label**=**'spam')))

Precision score:0.9424460431654677 Recall score:0.9424460431654677

f1 score:0.9424460431654677

In [35]:

**from** sklearn.metrics **import** classification\_report print(classification\_report(predictions,y\_test))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| ham | 0.99 | 0.99 | 0.99 | 976 |
| spam | 0.94 | 0.94 | 0.94 | 139 |
| accuracy |  |  | 0.99 | 1115 |
| macro avg | 0.97 | 0.97 | 0.97 | 1115 |
| weighted avg | 0.99 | 0.99 | 0.99 | 1115 |

In [44]:

**from** sklearn.naive\_bayes **import** MultinomialNB spam\_filter**=**MultinomialNB()

predictions**=**spam\_filter.fit(training\_data,y\_train)

In [45]:

predictions**=**spam\_filter.predict(testing\_data)

In [46]:

count**=**0

**for** i **in** range(len(y\_test)):

**if** y\_test.iloc[i] **!=**predictions[i]: count**+=**1

print('Total number of test cases',len(y\_test)) print('Number of wrong predictions',count)

Total number of test cases 1115 Number of wrong predictions 16

In [47]:

**from** sklearn.model\_selection **import** cross\_val\_score model**=**MultinomialNB()

scores**=**cross\_val\_score(model,X\_train,y\_train,scoring**=**'accuracy',cv**=**5,n\_jobs**=-**1)

c:\users\dell\appdata\local\programs\python\python39\lib\site-packages\sklea rn\model\_selection\\_validation.py:372: FitFailedWarning:

1. fits failed out of a total of 5.

The score on these train-test partitions for these parameters will be set to nan.

If these failures are not expected, you can try to debug them by setting err or\_score='raise'.

Below are more details about the failures:

1 fits failed with the following error:

Traceback (most recent call last):

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\model\_selection\\_validation.py", line 681, in \_fit\_and\_score

estimator.fit(X\_train, y\_train, \*\*fit\_params)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\naive\_bayes.py", line 663, in fit

X, y = self.\_check\_X\_y(X, y)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\naive\_bayes.py", line 523, in \_check\_X\_y

return self.\_validate\_data(X, y, accept\_sparse="csr", reset=reset)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\base.py", line 572, in \_validate\_data

X, y = check\_X\_y(X, y, \*\*check\_params)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\utils\validation.py", line 956, in check\_X\_y

X = check\_array(

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\utils\validation.py", line 738, in check\_array

array = np.asarray(array, order=order, dtype=dtype)

File "C:\Users\DELL\AppData\Roaming\Python\Python39\site-packages\numpy\co re\\_asarray.py", line 83, in asarray

return array(a, dtype, copy=False, order=order)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\pandas\core\series.py", line 768, in array

return np.asarray(self.array, dtype)

File "C:\Users\DELL\AppData\Roaming\Python\Python39\site-packages\numpy\co re\\_asarray.py", line 83, in asarray

return array(a, dtype, copy=False, order=order)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\pandas\core\arrays\numpy\_.py", line 203, in array

return np.asarray(self.\_ndarray, dtype=dtype)

File "C:\Users\DELL\AppData\Roaming\Python\Python39\site-packages\numpy\co re\\_asarray.py", line 83, in asarray

return array(a, dtype, copy=False, order=order)

ValueError: could not convert string to float: 'Free Msg: get Gnarls Barkley s \\Crazy\\" ringtone TOTALLY FREE just reply GO to this message right no

w!"'

4 fits failed with the following error:

Traceback (most recent call last):

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag

es\sklearn\model\_selection\\_validation.py", line 681, in \_fit\_and\_score estimator.fit(X\_train, y\_train, \*\*fit\_params)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\naive\_bayes.py", line 663, in fit

X, y = self.\_check\_X\_y(X, y)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\naive\_bayes.py", line 523, in \_check\_X\_y

return self.\_validate\_data(X, y, accept\_sparse="csr", reset=reset)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\base.py", line 572, in \_validate\_data

X, y = check\_X\_y(X, y, \*\*check\_params)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\utils\validation.py", line 956, in check\_X\_y

X = check\_array(

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\sklearn\utils\validation.py", line 738, in check\_array

array = np.asarray(array, order=order, dtype=dtype)

File "C:\Users\DELL\AppData\Roaming\Python\Python39\site-packages\numpy\co re\\_asarray.py", line 83, in asarray

return array(a, dtype, copy=False, order=order)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\pandas\core\series.py", line 768, in array

return np.asarray(self.array, dtype)

File "C:\Users\DELL\AppData\Roaming\Python\Python39\site-packages\numpy\co re\\_asarray.py", line 83, in asarray

return array(a, dtype, copy=False, order=order)

File "c:\users\dell\appdata\local\programs\python\python39\lib\site-packag es\pandas\core\arrays\numpy\_.py", line 203, in array

return np.asarray(self.\_ndarray, dtype=dtype)

File "C:\Users\DELL\AppData\Roaming\Python\Python39\site-packages\numpy\co re\\_asarray.py", line 83, in asarray

return array(a, dtype, copy=False, order=order)

ValueError: could not convert string to float: 'Sleeping nt feeling well' warnings.warn(some\_fits\_failed\_message, FitFailedWarning)

In [48]:

scores

Out[48]:

array([nan, nan, nan, nan, nan])

In [ ]: