Natural Language Processing : Sentiment Analysis

In [1]:

```
import numpy as np
import pandas as pd
import nltk
```

In [2]:

```
df_review = [line.rstrip() for line in open('/tmp/amazon_alexa.tsv')]
print (len(df_review))
#rstrip() plus a list comprehension to get a list of all the lines of text reviews.
```

```
3151
```

In [4]:

```
import pandas
df_review = pandas.read_csv('/tmp/amazon_alexa.tsv', sep='\t')
df_review.head()
```

Out[4]:

	rating	date	variation	verified_reviews	feedback
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1

In [5]:

```
#basic data analysis
df_review.describe()
```

Out[5]:

	rating	feedback
count	3150.000000	3150.000000
mean	4.463175	0.918413
std	1.068506	0.273778
min	1.000000	0.000000
25%	4.000000	1.000000
50%	5.000000	1.000000
75%	5.000000	1.000000
max	5.000000	1.000000

In [6]:

```
df_review.groupby('rating').describe()
```

Out[6]:

	feedback							
	count	mean	std	min	25%	50%	75%	max
rating								

	feedback							
	count	mean	std	min	25%	50%	75%	max
rating								
1	161.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	96.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	152.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0
4	455.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0
5	2286.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0

In [7]:

```
#New column to find length of text
df_review['length'] = df_review['verified_reviews'].apply(len)
df_review.head()
```

Out[7]:

	rating	date	variation	verified_reviews	feedback	length
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1	13
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1	9
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1	195
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1	172
4	5	31-Jul-18	Charcoal Fabric	Music	1	5

In [8]:

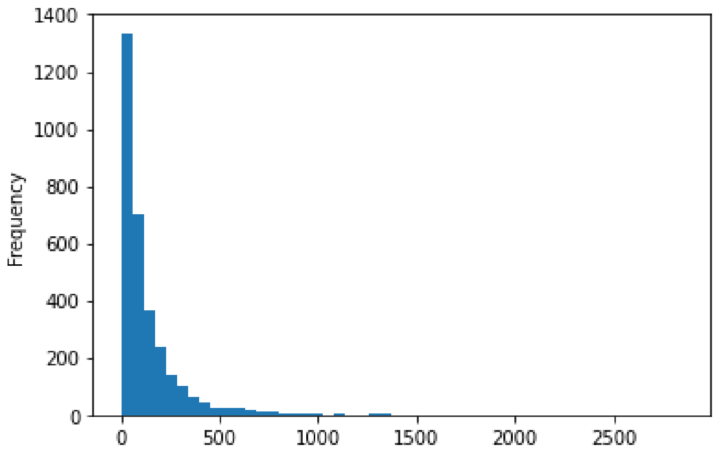
```
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

In [9]:

```
df_review['length'].plot(bins=50, kind='hist')
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb0676e3890>
```



In [10]:

```
df_review.length.describe() #TO find max length review
```

Out[10]:

```
count    3150.000000
mean      132.049524
std       182.099952
min         1.000000
25%        30.000000
50%        74.000000
75%       165.000000
max      2851.000000
max       2851.000000
Name: length, dtype: float64
```

2851 is max length text so we need to mask this first.

In [11]:

```
df_review[df_review['length'] == 2851]['verified_reviews'].iloc[0]
#syntax to find max length
```

Out[11]:

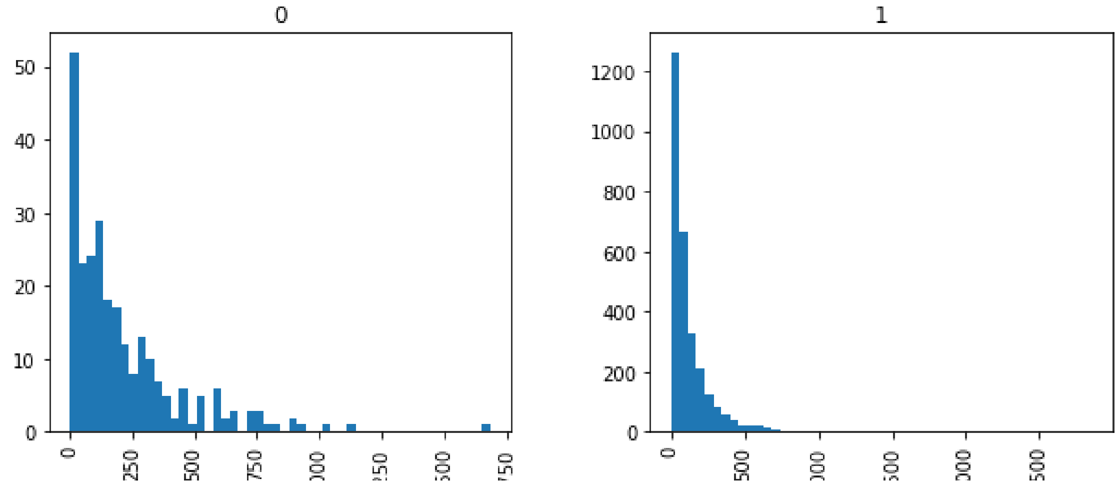
```
"Incredible piece of technology.I have this right center of my living room on an island k
itchen counter. The mic and speaker goes in every direction and the quality of the sound
is quite good. I connected the Echo via Bluetooth to my Sony soundbar on my TV but find t
he Echo placement and 360 sound more appealing. It's no audiophile equipment but there is
good range and decent bass. The sound is more than adequate for any indoor entertaining a
nd loud enough to bother neighbors in my building. The knob on the top works great for ad
justing volume. This is my first Echo device and I would imagine having to press volume b
uttons (on the Echo 2) a large inconvenience and not as precise. For that alone I would r
ecommend this over the regular Echo (2nd generation).The piece looks quality and is quite
sturdy with some weight on it. The rubber material on the bottom has a good grip on the g
ranite counter-- my cat can even rub her scent on it without tipping it over.This order c
ame with a free Philips Hue Bulb which I installed along with an extra one I bought. I pu
t the 2 bulbs into my living room floor lamp, turned on the light, and all I had to do wa
s say &#34;Alexa, connect my devices&#34;. The default names for each bulb was assigned a
s &#34;First light&#34; and &#34;Second light&#34;, so I can have a dimmer floor lamp if
I just turned on/off one of the lights by saying &#34;Alexa, turn off the second light&#3
4;. In the Alexa app, I created a 'Group' with &#34;First light&#34; and &#34;Second ligh
t&#34; and named the group &#34;The light&#34;, so to turn on the lamp with both bulbs sh
ining I just say &#34;Alexa, turn on The light&#34;.I was surprised how easily the bulbs
connected to the Echo Plus with its built in hub. I thought I would have to buy a hub bri
dge to connect to my floor lamp power plug. Apparently there is some technology built dir
ectly inside the bulb! I was surprised by that. Awesome.You will feel like Tony Stark on
this device. I added quite a few &#34;Skills&#34; like 'Thunderstorm sounds' and 'Quote o
f the day' . Alexa always loads them up quickly. Adding songs that you hear to specific p
laylists on Amazon Music is also a great feature.I can go on and on and this is only my s
econd day of ownership.I was lucky to buy this for $100 on Prime Day, but I think for $15
0 is it pretty expensive considering the Echo 2 is only $100. In my opinion, you will be
paying a premium for the Echo Plus and you have to decide if the value is there for you:1
) Taller and 360 sound unit.2) Volume knob on top that you spin (I think this is a huge b
enefit over buttons)3) Built in hub for Hue bulbs. After researching more, there are some
cons to this setup if you plan on having more advanced light setups. For me and my floor
lamp, it's just perfect.I highly recommend it and will buy an Echo dot for my bedroom now
."
```

In [12]:

```
#distinguish between positive and negative review
df_review.hist(column='length', by='feedback', bins=50,figsize=(10,4))
```

Out[12]:

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x7fb0675523d0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7fb0670cf910>],
      dtype=object)
```



In [13]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [14]:

```
# Importing the dataset
dataset = pd.read_csv('/tmp/amazon_alexa.tsv', delimiter = '\t', quoting = 3)
```

In [16]:

```
# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus=[]
for i in range(0,3150):
    review = re.sub('[^a-zA-Z]', ' ', dataset['verified_reviews'][i] )
    review=review.lower()
    review=review.split()
    ps=PorterStemmer()
    review=[ps.stem(word) for word in review if not word in set(stopwords.words('english
'))]
    review=' '.join(review)
    corpus.append(review)
```

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

In [17]:

```
# creating the Bag of words Model
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer(max_features=1500)
X=cv.fit_transform(corpus).toarray()
y=dataset.iloc[:,4].values
```

In [18]:

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state
= 0)
```

In [19]:

```
# Fitting Random Forest classifier with 100 trees to the Training set
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 100, criterion = 'entropy', random_st
ate = 0)
classifier.fit(X_train, y_train)
```

Out[19]:

```
RandomForestClassifier(criterion='entropy', random_state=0)
```

In [20]:

```
# Predicting the Test set results
y_pred = classifier.predict(X_test)
```

In [21]:

```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [22]:

```
cm
```

Out[22]:

```
array([[ 18,  36],
       [  0, 576]])
```

In [ ]:

```

```