Customer Support Analysis

Written by Domas Budrys

In [26]:

```
import csv

column_tweet_id = []
column_author_id = []
column_inboud = []
column_created_at = []
column_text = []
column_response_tweet_id = []
column_in_response_to_tweet_id = []



with open ('twcs.csv', encoding='utf-8') as csvfile:

    reader = csv.DictReader(csvfile)

    for row in reader:


        column_tweet_id.append(row['tweet_id'])
        column_text.append(row['text'])
```

Twitter ID and Text is being split into separate list. Because of this, we are able to sort and manipulate data using separate lists and not the entire file.

In [27]:

```
#Question 1
count_rows = sum(1 for line in column_tweet_id)
print ('Number of Tweets is: ', count_rows)
```

```
Number of Tweets is:  2811774
```

Question 1

We are assigning number of tweets to the variable **count_rows**. Then, `sum(1 for line in column_tweet_id)` is used to assign number "1" to every row in **column_tweet_id** and with the sum() function we are able get the total of all number "1"s.

In [28]:

```
#Question 2
tweet_269_index = column_tweet_id.index('269')
text = column_text[tweet_269_index]

print("Tweet ID :  ", column_tweet_id[tweet_269_index])
print(text)
```

```
Tweet ID :   269
@115770 こんにちは、アマゾン公式です。Fire TV Stickが見れないというのは、どのような状況でしょうか。一般的なトラブルシューティングを記載したヘルプがございますので、ご参照ください。https://t.co/2pbG55qJ7h ET
```

Question 2

Because earlier we stored all the values of tweet_id into list *column_tweet_id* now we are able to search this list for specific value.

By using `column_tweet_id.index('269')` we are able to specify that we are searching for tweet_id=269 and the index of it will be stored to the variable **tweet_269_index** . Then, we are able to use this index variable to search for a text related to tweet_id=269

In [29]:

```
#Question 3
ascii_char_set = set('0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!"#$%&\'()*+,-
./:;?@[\\]^_`{|}~ \t\n\r\x0b\x0c')

less_than = 0

for i in column_text:

    text_len = len(i)
    ascii_count = sum(char in ascii_char_set for char in i)


    if ((ascii_count / text_len ) < 0.5):
        less_than +=1


print ('The number of tweets that contains less than 50% of English '
       'characters is:',less_than)
```

```
The number of tweets that contains less than 50% of English characters is: 20025
```

Question 3

In Question 3 we are using variable **ascii_char_set** to store the set of characters that represents ASCII. Then, using `for i in column_text:` we are able to read every line from **column_text** and store it in variable **i**. After that, we are assigning the length of **i** to **text_len**. Now, in **ascii_count** we are able to store the sum of all characters of **i** that match any value in **ascii_char_set**. Finally, we make an if statement to compare if average value is less than 0.5 (50%)

In [30]:

```
#Question 4
import re

name_list=[]

for i in column_text:

    name_list += re.findall("[@]\w+", i)

unique_names = list(set(name_list))

print('Number of unique twitter names : ',len(unique_names))
```

```
Number of unique twitter names :  716567
```

Question 4

First, we must import regular expression library 're'. Then, we create an empty list **name_list** to store future name values. Then, in for loop we use re function *findall* where `[@]\w+` means that every word starting character @ and ending with any other value `\w+` will be stored to the **name_list**. Furthermore, `unique_names = list(set(name_list))` is used to store unordered unique values of **name_list**. Finally, we are able to get the count of unique twitter names by using *len()* fuction

In [31]:

```
#Question 5
import collections
print ('10 of the most used twitter names and their count :')
for i in collections.Counter(name_list).most_common(10):
    print (i)
```

```
10 of the most used twitter names and their count :
('@AmazonHelp', 136815)
('@AppleSupport', 98024)
```

```
('@AmericanAir', 50507)
('@Uber_Support', 47226)
('@Delta', 42559)
('@115858', 40726)
('@VirginTrains', 37592)
('@SouthwestAir', 34375)
('@Tesco', 34087)
('@SpotifyCares', 31214)
```

Question 5

collections must be imported at first. Then with `collections.Counter(name_list)` we are able get the count of every twitter name used in this data file. Next, we can use one of the built in functions for Counter and get the value of 10 `most_common` values.

In [32]:

```
#Question 6

hashtag_list=[]

for i in column_text:

    hashtag_list += re.findall("[#]\w+", i)

unique_hashtag = list(set(hashtag_list))

print('Number of unique twitter hashtags(#) : ', len(hashtag_list))
```

```
Number of unique twitter hashtags(#) :  241942
```

Question 6

For this question we will be using 're' library again. We create an empty list **hashtag_list** to store future hashtag values. Then, in for loop we use re function *findall()* where `[#]\w+` means that every word starting character # and ending with any other value \w+ will be stored to the hashtag_list, because of this a single # will not be added to the list. Furthermore, `unique_hashtag = list(set(hashtag_list))` is used to store unordered unique values of hashtag_list. Finally, we are able to get the count of unique twitter names by using len() fuction

In [33]:

```
#Question 7
print ('10 of the most used twitter hashtags and their count :')
for i in collections.Counter(hashtag_list).most_common(10):
    print (i)
```

```
10 of the most used twitter hashtags and their count :
('#mobile_Care', 2151)
('#AATeam', 1621)
('#fail', 1604)
('#Amazon', 1385)
('#iPhoneX', 1247)
('#iOS11', 1226)
('#hppsdr', 1208)
('#help', 1116)
('#mobile_CareXI', 1050)
('#CustomerService', 1030)
```

Question 7

Using with collections.Counter(name_list) we are able get the count of every hashtag used in this data file. Next, we can use one of the built in functions for Counter and get the value of 10 most_common values

In [34]:

```
#Question 8
```

```
all_words = []
for i in column_text:

    all_words += re.findall(r'[a-zA-Z]+', i )


#nltk.download("stopwords")
#from nltk.corpus import stopwords

#stop_words = stopwords.words('english')

#for w in list(all_words):  # iterating on a copy since removing will mess things up
#    if w in stop_words:
#        all_words.remove(w)

for i in collections.Counter(all_words).most_common(20):
    print (i)
```

```
('to', 1679423)
('the', 1408351)
('you', 1264804)
('I', 1154133)
('t', 1020842)
('a', 885713)
('and', 801336)
('for', 754446)
('your', 683002)
('co', 655927)
('https', 654523)
('this', 532376)
('it', 497247)
('on', 485821)
('is', 482405)
('can', 474957)
('in', 467208)
('us', 445446)
('with', 443049)
('We', 429121)
```

Question 8

With the first for loop we are able to find only the words that start with English letters (ignoring lowercase or uppercase) and assign to the list **all_words** Then, we are using the Counter() to display 20 most common words: `for i in collections.Counter(all_words).most_common(20)`

Commented code should be used to remove stop words from the list of **all_words**. I was not able to see the result of this code because my computer was taking way too long to execute it.

```
nltk.download("stopwords")
from nltk.corpus import stopwords

stop_words = stopwords.words('english')

for w in list(all_words):  # iterating on a copy since removing will mess things up
    if w in stop_words:
        all_words.remove(w)
```