**Project Info 5810**

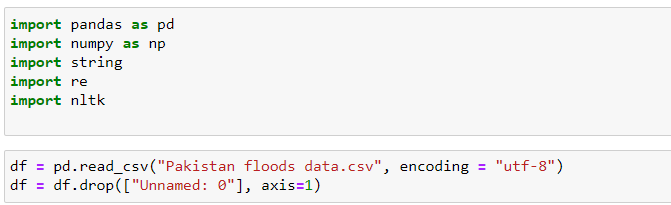
**Second submission**

**Data Downloading:**

For this project, we need tweet data from Twitter and for downloading the data, we have used web scraping with API.For this code, we need to create a Twitter developer account to get a few keys to run the process but we already created the account. We have used a few keys such as API key, api\_secret\_key, access\_token, and access\_token\_secret. Here we have to give a keyword to search that hashtag might be anything that has already been used by someone on Twitter.

So here we have used Pakistan Floods as a keyword. This keyword must be a hashtag on Twitter. After running a cell, it will display the tweets, likes, time, and source of the keyword. The total number of tweets is 1511. Later we need to save files by giving a proper path.

**Data Cleaning:**

As we mentioned, we collected the data from Twitter using tweeter API. For cleaning the data, first, we make a python file using python 3. We imported some useful libraries such as pandas, NumPy, re, and nltk. Pandas library is the most useful library for data analysis and we use this library to import the data into the python program and which is a text file. Re library comes with several functions that employ regular expressions for string matching. For reading the text file we have to read the file using utf-8 format because we cannot read the text file directly

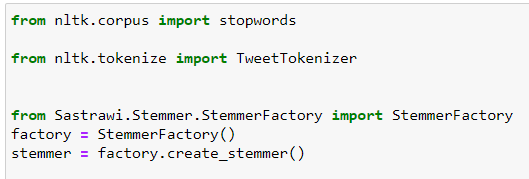
For cleaning the tweets data, we created a function that will clean the data line by line. So with this function, we are going to get rid of the @RT, three dots at the last of the line, semicolon, unusual words, ampersands (&), user names, and the website address.

First, we are removing the (@Username) pattern using regex (regular) expression "@[\w]\*" and then removing the retweet using this regular expression r"^RT\s\w\*" , this will remove the match string from the data set and replace with blank space. Also remove the commons (,) if it is present in the tweet with regular expression r",".

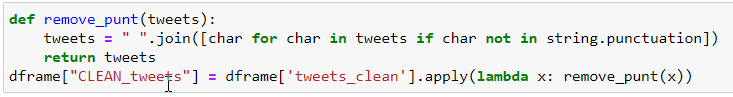
Second, we remove all the places, where @username comes in the tweets. For removing the user name that comes in between the tweets, we use the r“@\w+”. this regular expression removes the @username and replaces it with a blank space.

Third, we want to remove all the hyperlinks from the tweets data, for this, we use the r"(http[s]?:\\/\\/(www\\.)?|ftp:\\/\\/(www\\.)?|www\\.){1}([0-9A-Za-z-\\.@:%\_\+~#=]+)+((\\.[a-zA-Z]{2,3})+)(/(.)\*)?(\\?(.)\*)?" regular expression that will remove all most all hyperlink and replace it with blank space. We also want to remove the emoji from the data. For removing the emoticons, symbols & pictographs, transport & map symbols, flags (iOS), and Chinese char. 

define the function that can remove all kinds of emoji and pass the tweets to this function.

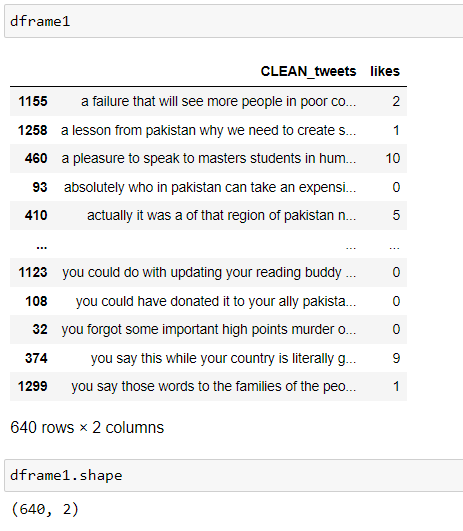
Next, we need to import the nltk library and from nltk library, we need to import some useful packages which are going to use the remove punctuations and do some further processing of our data. From nltk, we import TweetTokenizer, which separates the sentences into individual words. So, why do we use the word\_tokenizer to split the words, even the split method also can do the same thing because it does not do the simple split based on the spaces, it separated the words and punctuations over there and assign the output of the word tokenizer into the variable for further use.

Now, we want to remove the punctuation from the tweets, so, we created an empty list and we have a method that checks if it has no word in the string.punctuation will append the list.

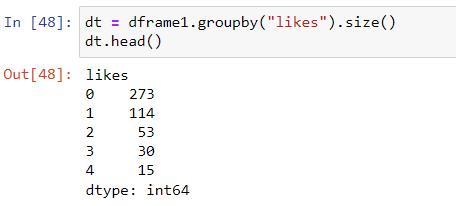
We have a list at in our hand but if we want or string, we convert the list into the string using “ ”.join(list\_name)

**Analysis:**

Initially, we had 1511 tweets in our data set and after cleaning the data set we have 640 tweets in our clean data set. Final data after cleaning is showing below.



Likes sorted maximum likes in this data set is 273.



Line graph of the likes

