BIG DATA PROGRAMMING - FINAL REPORT

1. PROJECT TITLE AND TEAM MEMBERS

International Student Data Analysis using Big Data.

- Tanvi Jain
- Saikumar Reddy Papagari
- Amarnadha Reddy Ankireddypalli
- Thotakura Naga Mounika

2. INTRODUCTION

This project focuses on extraction of dataset (Twitter Data) from the Twitter API regarding all the information related to international students and using different tools like Hive, Hue, Scala, pySpark, Spark SQL and Solr to show different aspects of the data in a visualization form using Tableau and Seaborn.

GitHub Link: https://github.com/xlr8r53/BDP-Project-Team-6/wiki/Final-Report-Submission

3. BACKGROUND

Twitter sentiment analysis is done in different ways before, using panda is the most common way. We will be using different hashtags which are related to international students and events related to international students to get a hold of better results based on those hashtags and more information about what events really took place and different aspects they bring out. Here are some references of projects people have done implementing twitter analysis with big data, we are going to follow some aspects of these projects, but our dataset is new and is extracted only for this project, so we could only find references related to those projects which have totally different dataset and completely different objectives.

1. https://github.com/dsuarez993/bigdata-realtime-twitter-analysis
2. https://github.com/6vedant/TwitterAnalyticsHadoop
3. https://www.toptal.com/apache/apache-spark-streaming-twitter

4. GOALS AND OBJECTIVES

Motivation:

Studying abroad is a journey of education and discovery. There are currently over 1 million international students from more than 220 countries, coming to the United States annually. The

individuals from this group belong to the international student community and we came to the United States to get a higher education. There are numerous situations where we are relied upon to follow those standards which the overall US citizens are not expected to follow in the long run since we have a place with the foreigner gathering. So, this gave us the establishment for this task, and we chose to feature the fundamental information like percentage of students going to the United States for education, the probability of getting a work VISA, Immigration rules change for F-1 Visa during COVID, Jobs for international students during Covid (H1B sponsorship).

Significance:

Big data tools help to analyze the huge data which helps to provide efficient results. The sentimental analysis provides a brief understanding of various challenges that international students are currently facing and impact of covid-19 on the visa assurance. Finally, we are using Spark using python and Scala for writing the queries by visualizing with Seaborn and Tableau.

Objectives:

The objective of this project is to get twitter data extraction using Twitter Data Analysis and then cleaning the data, performing sentimental analysis, and importing files into Hadoop where we used different tools like Hue, Hive, Solr, Scala, Spark SQL and pySpark.

Features:

The main feature of the project is to collect the Real timed tweets from the twitter API, also by performing the ETL which means we preprocess the data using Texthero and extract the necessary data and then we load the extracted data in our HIVE. Performed topic modelling using LDA and genism model, visualized the top 4 topics by t-SNE visualization and performed analysis on data using PySpark, Solr and Scala

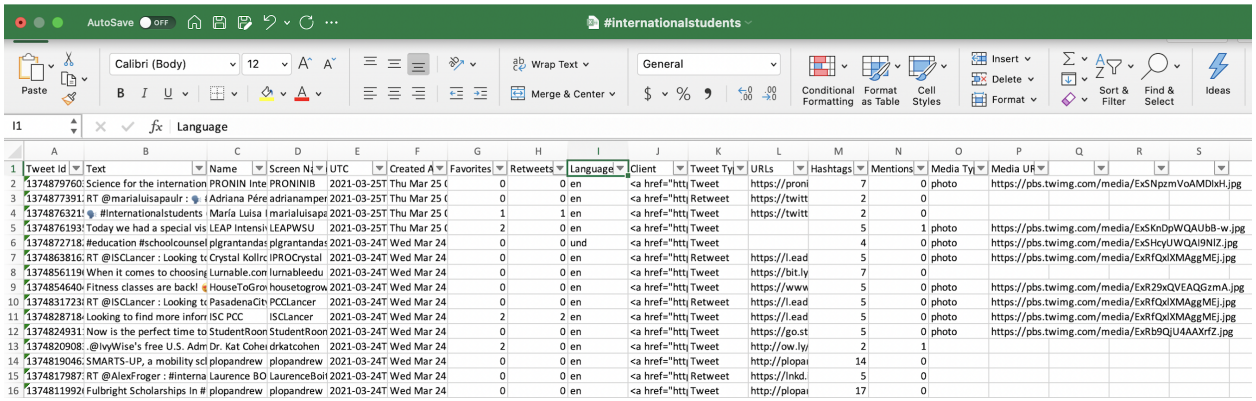
5. DATASET:

We collected the data using twitter API using developer account and API keys. We have used different hashtags to get data related to 3 different scenarios as follows:

1. Generic International Student Data (#F1visa, #intlstudents, #internationalstudents, #studyinUSA)

2. Immigration rules change for F1 Visa during COVID in 2020 (#AbolishICE)

3. Jobs for international students during covid (H1B sponsorship) (#H1B, #h1bjobs) From the extraction we have the information about the tweet itself like this:



Tweet Id	Text	Name	Screen Na	UTC	Created A	Favorites	Retweets	Language	Client	Tweet Ty	URLs	Hashtags	Mentions	Media Ty	Media UR
1374879760	Science for the internation	PRONIN Inte	PRONINIB	2021-03-25T	Thu Mar 25 (	0	0	en	<a href="htt	Tweet	https://proni	7	0	photo	https://pbs.twimg.com/media/ExSNpzmVoAMDlxH.jpg
1374877391	RT @marialuisapaulr :	Adriana Pére	adrianamper	2021-03-25T	Thu Mar 25 (	0	0	en	<a href="htt	Retweet	https://twitt	2	0		
1374876321	#Internationalstudents	María Luisa I	marialuisapa	2021-03-25T	Thu Mar 25 (	1	1	en	<a href="htt	Tweet	https://twitt	2	0		
1374876193	Today we had a special vis	LEAP Intensi	LEAPWSU	2021-03-25T	Thu Mar 25 (	2	0	en	<a href="htt	Tweet		5	1	photo	https://pbs.twimg.com/media/ExSKnDpWQAUbB-w.jpg
1374872718	#education #schoolcounsel	plgrantandas	plgrantandas	2021-03-24T	Wed Mar 24	0	0	und	<a href="htt	Tweet		4	0	photo	https://pbs.twimg.com/media/ExSHcyUWQAI9NIZ.jpg
1374863816	RT @ISCLancer : Looking t	Crystal Kollrc	IPROCrystal	2021-03-24T	Wed Mar 24	0	0	en	<a href="htt	Retweet	https://l.ead	5	0	photo	https://pbs.twimg.com/media/ExRfQxlXMAggMEj.jpg
1374856119	When it comes to choosing	Lurnable.com	lurnableedu	2021-03-24T	Wed Mar 24	0	0	en	<a href="htt	Tweet	https://bit.ly	7	0		
1374854640	Fitness classes are back!	HouseToGro	housetogrow	2021-03-24T	Wed Mar 24	0	0	en	<a href="htt	Tweet	https://www	5	0	photo	https://pbs.twimg.com/media/ExR29xQVEAQGzmA.jpg
1374831723	RT @ISCLancer : Looking t	PasadenaCity	PCCLancer	2021-03-24T	Wed Mar 24	0	0	en	<a href="htt	Retweet	https://l.ead	5	0	photo	https://pbs.twimg.com/media/ExRfQxlXMAggMEj.jpg
1374828718	Looking to find more infor	ISC PCC	ISCLancer	2021-03-24T	Wed Mar 24	2	2	en	<a href="htt	Tweet	https://l.ead	5	0	photo	https://pbs.twimg.com/media/ExRfQxlXMAggMEj.jpg
1374824931	Now is the perfect time to	StudentRoon	StudentRoon	2021-03-24T	Wed Mar 24	0	0	en	<a href="htt	Tweet	https://go.st	5	0	photo	https://pbs.twimg.com/media/ExRb9QjU4AAXrfZ.jpg
1374820908	.@IvyWise's free U.S. Adm	Dr. Kat Cohe	drkatcohen	2021-03-24T	Wed Mar 24	2	0	en	<a href="htt	Tweet	http://ow.ly/	2	1		
1374819046	SMARTS-UP, a mobility scl	plopandrew	plopandrew	2021-03-24T	Wed Mar 24	0	0	en	<a href="htt	Tweet	http://plopa	14	0		
1374817987	RT @AlexFroger : #interna	Laurence BO	LaurenceBoi	2021-03-24T	Wed Mar 24	0	0	en	<a href="htt	Retweet	https://lnkd.	5	0		
1374811992	Fulbright Scholarships In #	plopandrew	plopandrew	2021-03-24T	Wed Mar 24	0	0	en	<a href="htt	Tweet	http://plopa	17	0		

The information about the user as well - It has all the different fields, which we will filter and use later using big data tools to perform queries and visualization.

User Id	Name	Screen Nam	UTC	Created At	Followers	Following	Favorites	Tweets	Lists	Bio	Location	URL	Verified	Default Profile
171719416	PRONIN Inte	PRONINIB	2010-07-28T	Wed Jul 28 0	173	315	210	349	1	We've been	Sydney, New	http://www.	FALSE	FALSE
1158836808	Adriana Pére	adrianamper	2019-08-06T	Tue Aug 06 2	324	582	6690	1508	1	Guayaquileñ	Perla del Pac	https://ndsm	FALSE	TRUE
1066522985	María Luisa I	marialuisapa	2018-11-25T	Sun Nov 25 0	323	388	2156	1515	2	Talent & Incl	Caracas, Ven	https://mari	FALSE	FALSE
1920828079	LEAP Intensi	LEAPWSU	2013-09-30T	Mon Sep 30	452	253	1784	3770	5	Wright State	Dayton, OH	http://www.	FALSE	FALSE
1227615420	plgrantandas	plgrantandas	2020-02-12T	Wed Feb 12	8	160	9	135	0	International		http://www.	FALSE	TRUE
198577337	Crystal Kollrc	IPROCrystal	2010-10-04T	Mon Oct 04 1	65	120	144	1521	2	Lies, damn li			FALSE	TRUE
1077204432	Lurnable.com	lurnableedu	2013-01-10T	Thu Jan 10 1	39	14	14	793	3	Lurnable is tl		http://www.	FALSE	FALSE
8955770950	HouseToGrov	housetogrow	2017-08-10T	Thu Aug 10 0	33	306	4	412	1	#Charity that	Sydney, New	http://house	FALSE	FALSE
78736148	PasadenaCity	PCCLancer	2009-09-30T	Wed Sep 30	6046	338	10010	10033	166	The official T	Pasadena, C/	http://pasad	FALSE	FALSE
386812395	ISC PCC	ISCLancer	2011-10-07T	Fri Oct 07 23	59	9	34	524	0	The official T	Pasadena, C/	http://www.	FALSE	TRUE
1275139023	StudentRoon	StudentRoon	2020-06-22T	Mon Jun 22 1	6	16	4	65	0	Building the			FALSE	TRUE
7358628181	Dr. Kat Cohe	drkatcohen	2016-05-26T	Thu May 26 1	11020	531	44957	21727	154	Founder and	New York, N	http://IvyWis	FALSE	FALSE

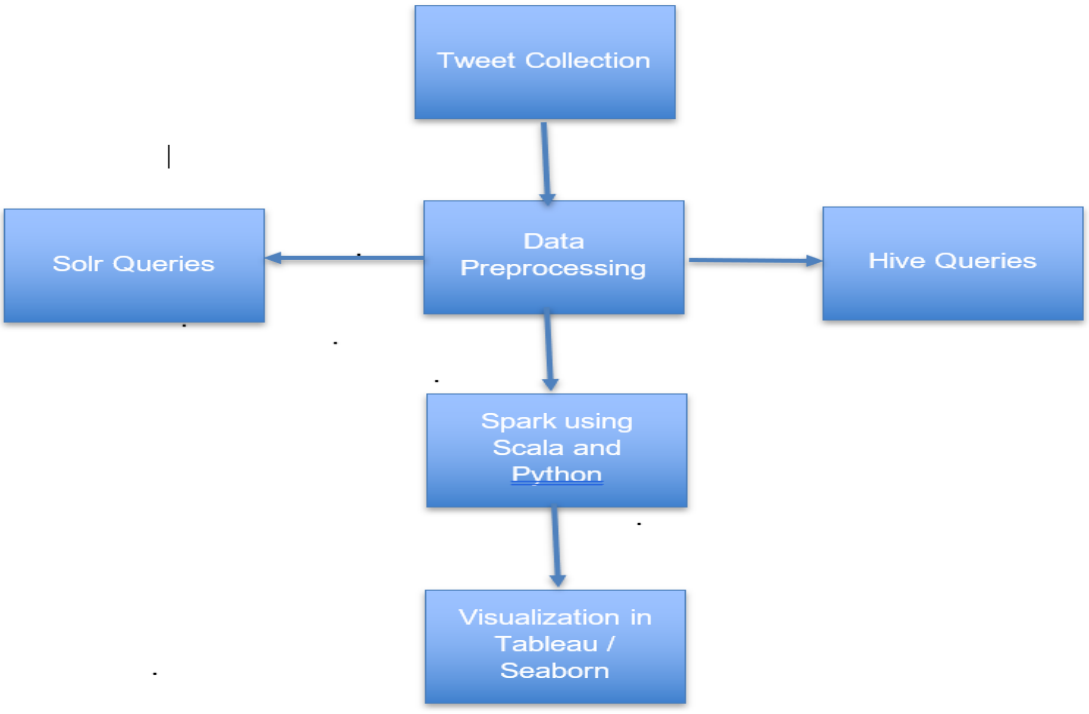
It has all the different fields, which we will filter and use later using big data tools to perform queries and visualization.

Features and their description:

Feature	Description
Tweet Id	Unique Id of user
Text	UTF-8 text Tweet data
Name	Name of the user
Screen Name	Profile or Screen Name of the user
UTC	Tweet created timestamp in UTC
Created At	Tweet created timestamp
Favorites	Count of people favorited the tweet
Retweets	Count of retweets of the tweet
Language	Language opted by the user
Client	Type of device used to post the tweet
Tweet Type	Type of tweet (Tweet, Retweet or Reply)
URLs	Different URLs included in the tweet
Hashtags	Count of people favorited the tweet
Mentions	Count of people favorited the tweet
Media Types	Types of media files included in the tweet
Media URLs	Types of media URLs included in the tweet

Table: Description of Attributes of Data

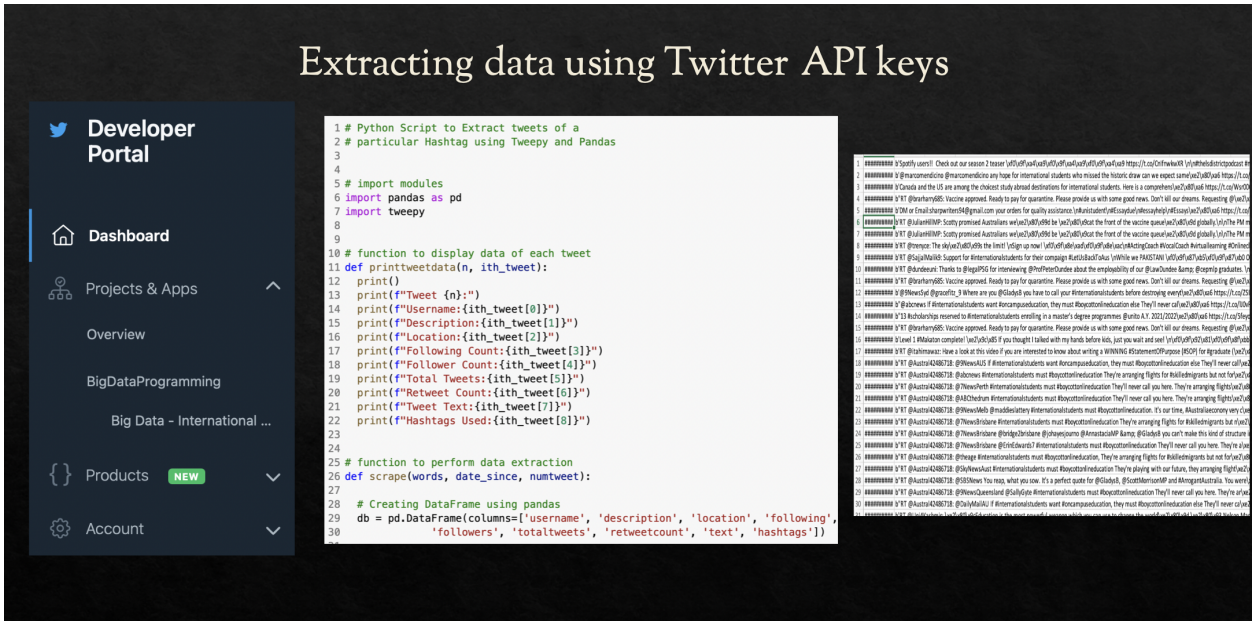
Detail design of Features with Project Workflow:



6. DATA ANALYSIS:

We have extracted dataset using twitter API, these datasets are extracted using different hashtags focusing on different events related to international students, dividing the events and dataset is crucial since we want this project to be informational and focus on different aspects through different datasets. We will use sentiment analysis on the text (tweet) and use different queries to extract important features and visualize them in Tableau.

Data Collection:



Data Preprocessing:

a.

Preprocessing

```
def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for i in r:
        input_txt = re.sub(i, '', input_txt)

    return input_txt

# remove twitter handles (@user)
train['tidy_tweet'] = np.vectorize(remove_pattern)(train['tweets'], "@[\w]*")

# remove special characters, numbers, punctuations
train['tidy_tweet'] = train['tidy_tweet'].str.replace("[^a-zA-Z#]", " ")

#removing words less than length 2
train['tidy_tweet'] = train['tidy_tweet'].apply(lambda x: ' '.join([w for w in x.split() if len(w)>2]))

tokenized_tweet = train['tidy_tweet'].apply(lambda x: x.split())
tokenized_tweet.head()
```

```
0    [any, hope, for, international, students, who,...
1    [Canada, and, the, are, among, the, choicest, ...
2    [Vaccine, approved, Ready, pay, for, quarantin...
3    [Email, sharpwriters, com, your, orders, for, ...
4    [Scotty, promised, Australians, cat, the, fron...
Name: tidy_tweet, dtype: object
```

```
#stemming
from nltk.stem.porter import *
stemmer = PorterStemmer()

tokenized_tweet = tokenized_tweet.apply(lambda x: [stemmer.stem(i) for i in x]) # stemming
tokenized_tweet.head()
```

```
[32]  train.head()
```

	tweets	tidy_tweet
0	b'@marcomendicino @marcomendicino any hope for...	ani hope for intern student who miss the histo...
1	b'Canada and the US are among the choicest stu...	canada and the are among the choicest studi ab...
2	b"RT @brarharry685: Vaccine approved. Ready to...	vaccin approv readi pay for quarantin pleas pr...
3	b'DM or Email:sharpwriters94@gmail.com your or...	email sharpwrit com your order for qualiti ass...
4	b'RT @JulianHillMP: Scotty promised Australian...	scotti promis australian cat the front the vac...

```
[41]  tidy_tweets = (train['tidy_tweet'])
```

b.

```
tw = hero.visualization.top_words(df["Text"].pipe(hero.clean)).head(10)
tw.plot.bar()
tw.head(10)
```

```
internationalstudents    4220
australians              3202
vaccine                  3144
abolishice               2916
letusbacktoaus           2504
queue                    2288
h1b                      2280
julianhillmp             1720
us                       1662
amp                      1640
Name: Text, dtype: int64
```

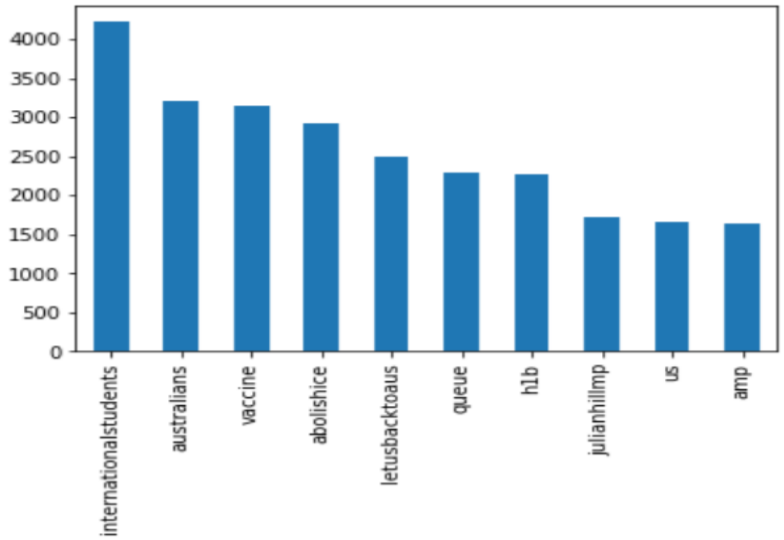


Fig. Plotting the Top words from the data.

c.

```
#Add pca value to dataframe to use as visualization coordinates
df['pca'] = (
              df['Text']
              .pipe(hero.tfidf)
              .pipe(hero.pca)
    )
#Add k-means cluster to dataframe
df['kmeans'] = (
              df['Text']
              .pipe(hero.tfidf)
              .pipe(hero.kmeans)
    )
df.head()
```

Fig: Adding PCA and k-means clusters to the preprocessed data.

d.

```
#generate scatter plot
hero.scatterplot(df, 'pca', color = 'kmeans', hover_data=['Text '] )
```

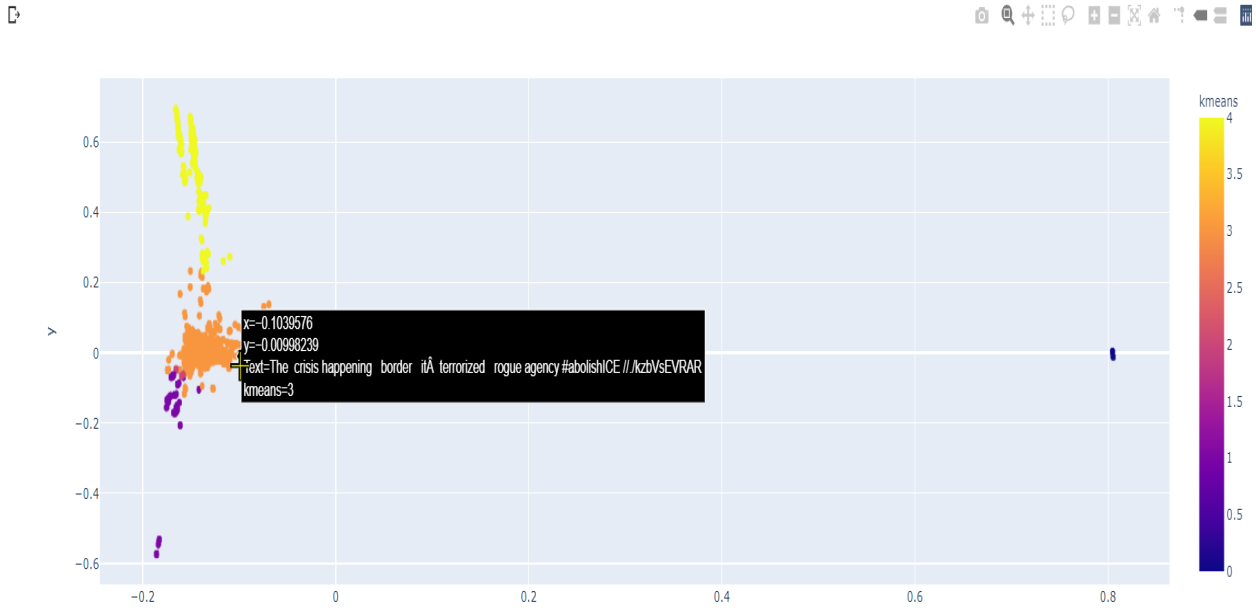
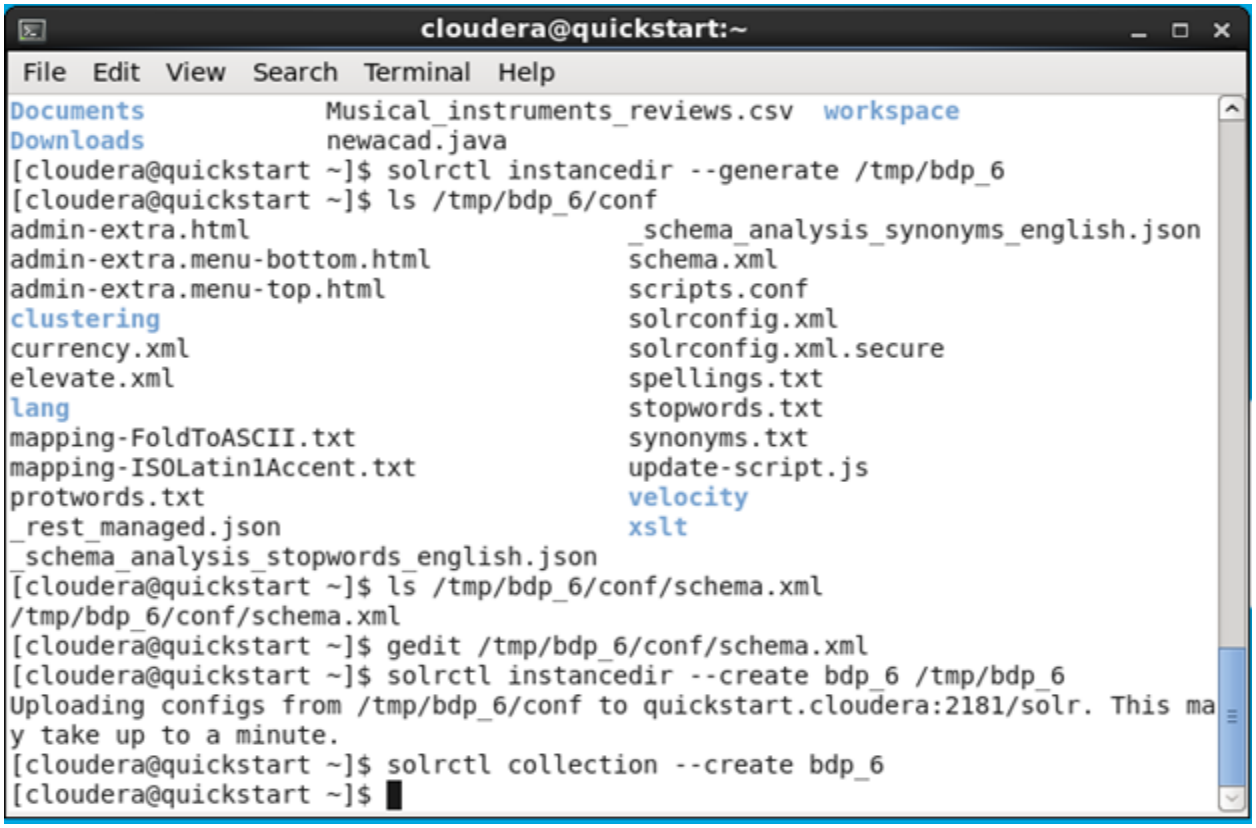


Fig: Scatter plot hovering the k-means and PCA clustered data

7. IMPLEMENTATION:

Solr:

1. Creation/generation of instance & Collection:

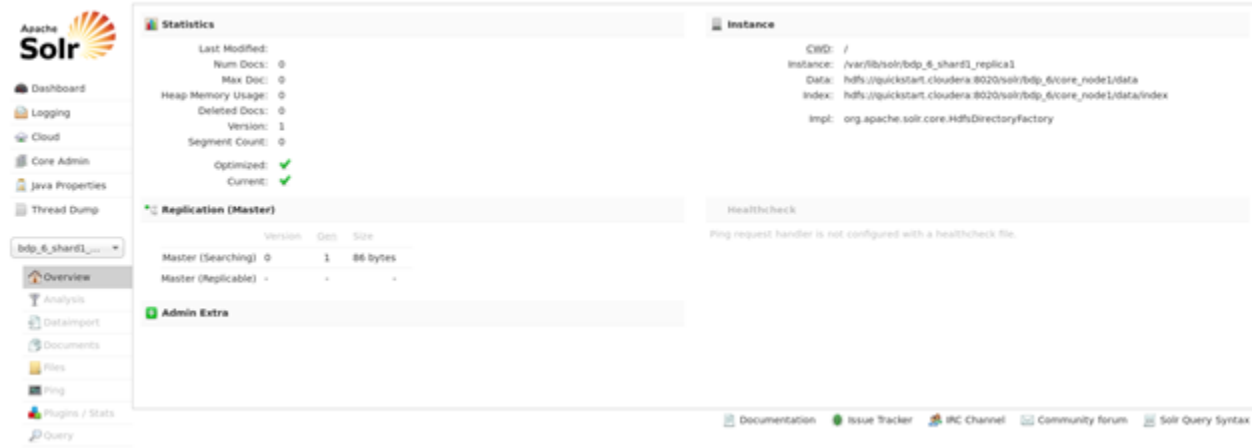


2. Edit the schema.xml created with the instance generation inside the configuration folder to change the attributes based on the dataset given:

schema.xml

```
      <field name="Tweet Id" type="string" indexed="true" stored="true" required="true" multiValued="false" />
      <field name="Text" type="string" indexed="true" stored="true" />
      <field name="Name" type="string" indexed="true" stored="true" />
      <field name="Screen Name" type="string" indexed="true" stored="true" />
      <field name="UTC" type="string" indexed="true" stored="true" />
      <field name="Created At" type="string" indexed="true" stored="true" />
      <field name="Favorites" type="string" indexed="true" stored="true" />
      <field name="Retweets" type="string" indexed="true" stored="true" />
      <field name="Language" type="string" indexed="true" stored="true" />
      <field name="Client" type="string" indexed="true" stored="true" />
      <field name="Tweet Type" type="string" indexed="true" stored="true" />
      <field name="URLs" type="string" indexed="true" stored="true" />
      <field name="Hashtags" type="string" indexed="true" stored="true" />
      <field name="Mentions" type="string" indexed="true" stored="true" />
      <field name="Media Type" type="string" indexed="true" stored="true" />
      <field name="Media URLs" type="string" indexed="true" stored="true" />
       <field name="" type="string" indexed="true" stored="true" />
      <field name="" type="string" indexed="true" stored="true" />
      <field name="" type="string" indexed="true" stored="true" />

<!-- points to the root document of a block of nested documents. Required for nested
     document support, may be removed otherwise
-->
<field name="_root_" type="string" indexed="true" stored="false"/>

<field name="sku" type="text_en_splitting_tight" indexed="true" stored="true" omitNorms="true"/>
<field name="name" type="text_general" indexed="true" stored="true"/>
<field name="manu" type="text_general" indexed="true" stored="true" omitNorms="true"/>
<field name="cat" type="string" indexed="true" stored="true" multiValued="true"/>
<field name="features" type="text_general" indexed="true" stored="true" multiValued="true"/>
<field name="includes" type="text_general" indexed="true" stored="true" termVectors="true" termPositions="true" termOffsets="true" />

<field name="weight" type="float" indexed="true" stored="true"/>
<field name="price"  type="float" indexed="true" stored="true"/>
<field name="popularity" type="int" indexed="true" stored="true" />
<field name="inStock" type="boolean" indexed="true" stored="true" />

<field name="store" type="location" indexed="true" stored="true"/>
```

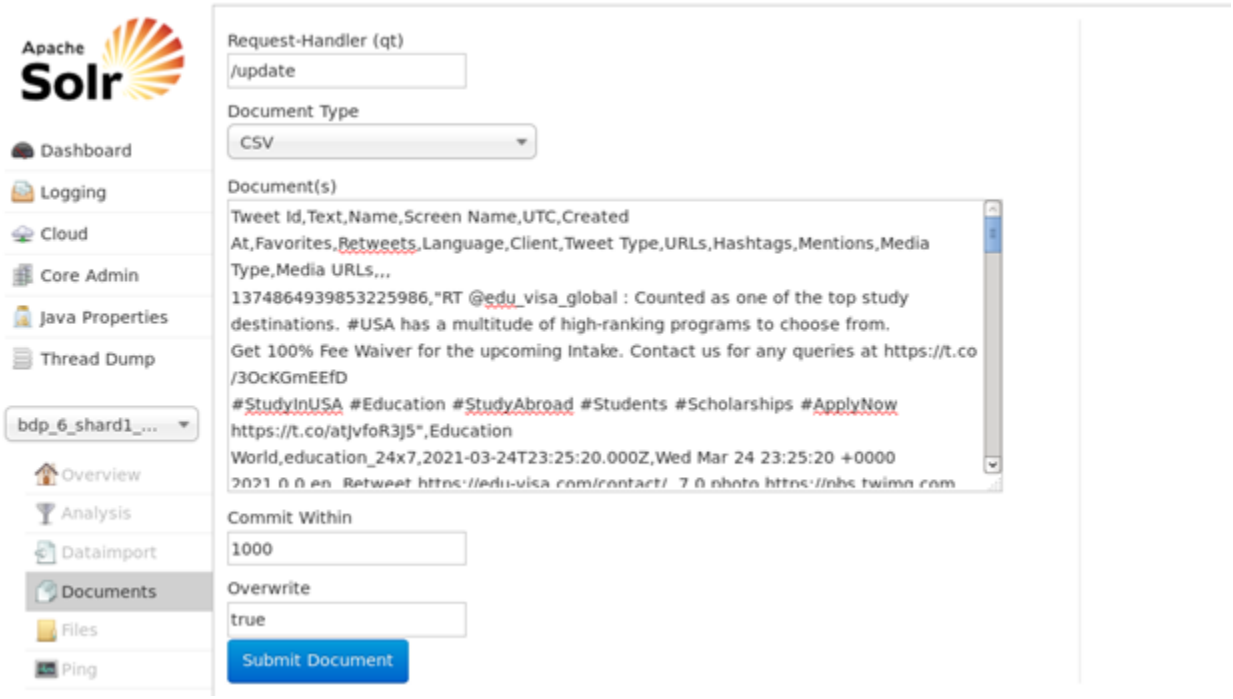
3. Open the solr browser in the web browser and select the create collection on the left side dropdown.



4.Set "Tweet Id" as the primary key.

```
<!-- Field to use to determine and enforce document uniqueness.
     Unless this field is marked with required="false", it will be a required field
  -->
  <uniqueKey> Tweet Id </uniqueKey>
```

5. Select the document type to csv. Then copy paste all the data inside the dataset into the documents field and submit the document.



Queries

1. Pulled the 10 records of data:

Request-Handler (qt)

/select

common

q

:

fq

sort

start, rows

0 10

fl

df

Raw Query Parameters

key1=val1&key2=val2

wt

json

indent

debugQuery

http://quickstart.cloudera:8983/solr/team6_shard1_replica1/select?q=*%3A*&wt=json&indent=true


```
{
  "responseHeader": {
    "status": 0,
    "QTime": 3,
    "params": {
      "indent": "true",
      "q": "*:*",
      "_": "1619793943113",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 78,
    "start": 0,
    "docs": [
      {
        "Tweet Id": "1374834250667716608",
        "Text": "RT @Maiquel59232858 : All categories of immigrant visas in Cuba #IR5Visa #F1Visa #F3Visa #F4Visa #f2AVisa
        "Name": "Neibis Aguilar",
        "Screen Name": "AguilarNeibis",
        "UTC": "2021-03-24T21:23:23.000Z",
        "Created At": "Wed Mar 24 21:23:23 +0000 2021",
        "Favorites": "0",
        "Retweets": "0",
        "Language": "en",
        "Client": "<a href=\"https://mobile.twitter.com\" rel=\"nofollow\">Twitter Web App</a>",
        "Tweet Type": "Retweet",
```


2. Pulled the tweets that are tweeted in English:

Request-Handler (qt)

/select

common

q

Language:*en*

fq

sort

start, rows

0 10

fl

df

Raw Query Parameters

key1=val1&key2=val2

wt

json

indent

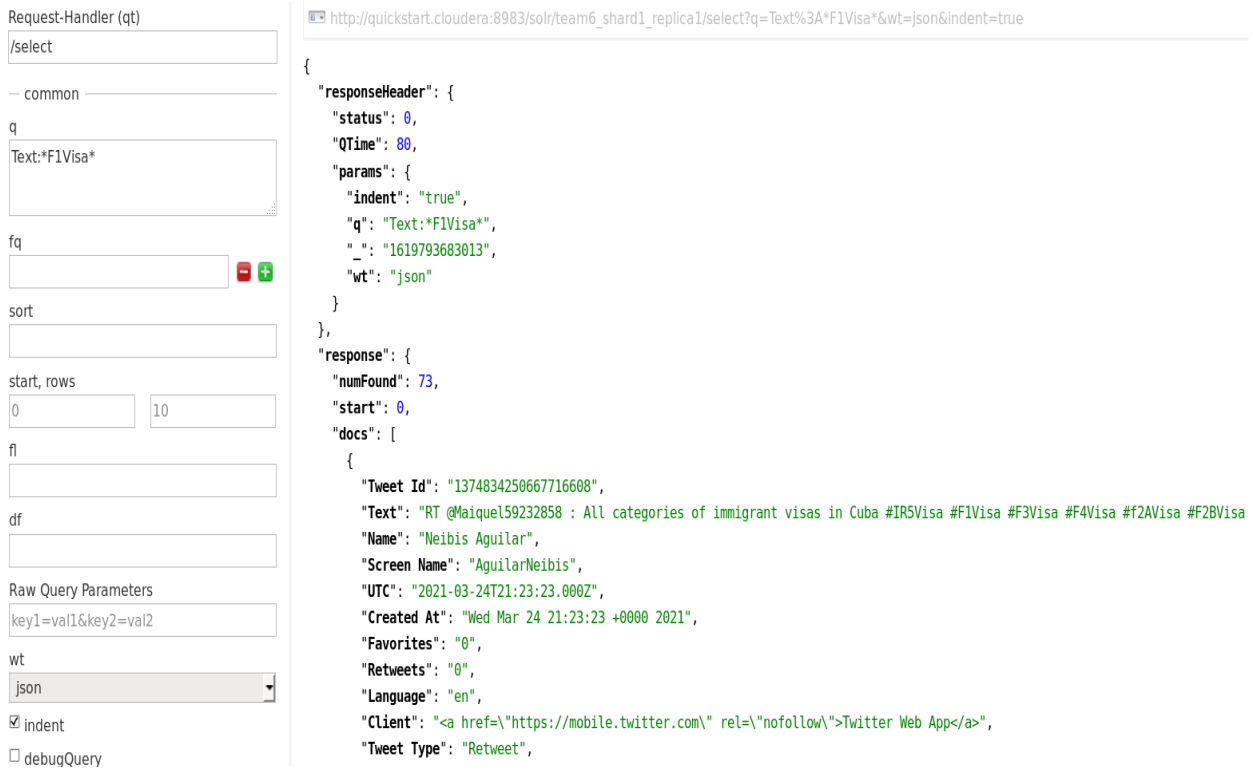
debugQuery

dismax

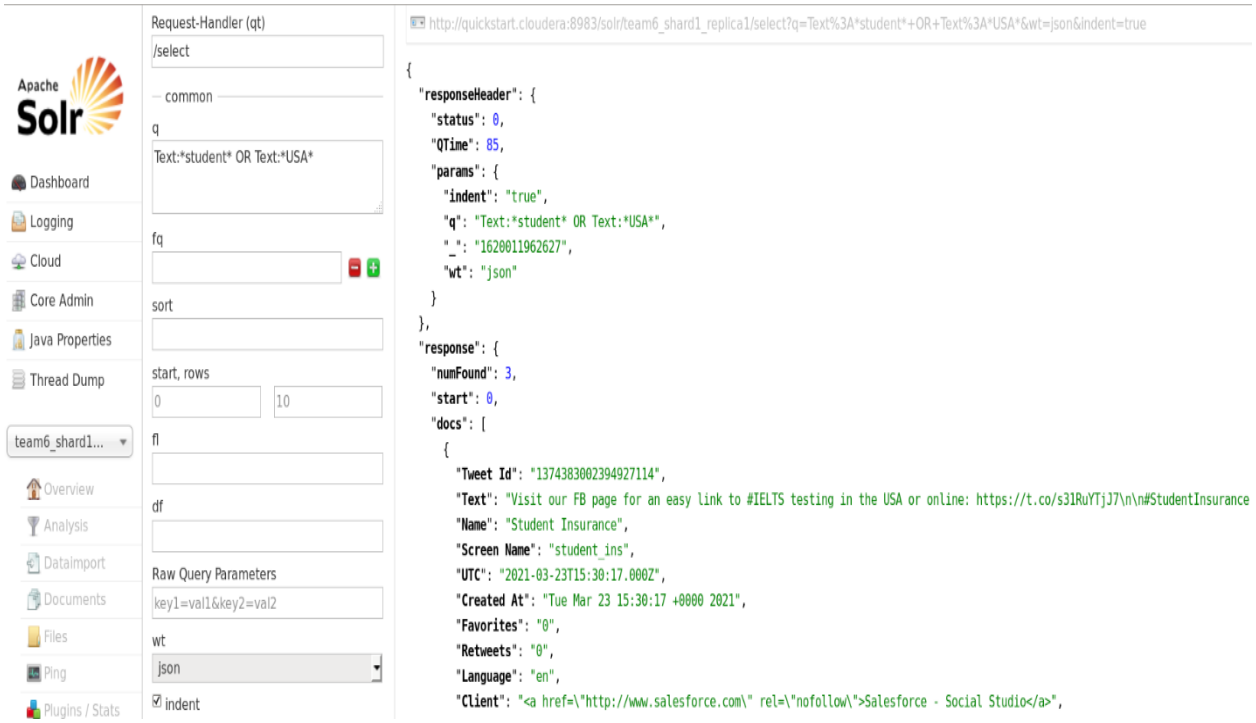
http://quickstart.cloudera:8983/solr/team6_shard1_replica1/select?q=Language%3A*en*&wt=json&indent=true


```
{
  "responseHeader": {
    "status": 0,
    "QTime": 82,
    "params": {
      "indent": "true",
      "q": "Language:*en*",
      "_": "1619753537943",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 77,
    "start": 0,
    "docs": [
      {
        "Tweet Id": "1374834250667716608",
        "Text": "RT @Maiquel59232858 : All categories of immigrant visas in Cuba #IR5Visa #F1Visa #F3Visa #F4Visa #f2AVisa #F2BVisa
        "Name": "Neibis Aguilar",
        "Screen Name": "AguilarNeibis",
        "UTC": "2021-03-24T21:23:23.000Z",
        "Created At": "Wed Mar 24 21:23:23 +0000 2021",
        "Favorites": "0",
        "Retweets": "0",
        "Language": "en",
        "Client": "<a href=\"https://mobile.twitter.com\" rel=\"nofollow\">Twitter Web App</a>",
        "Tweet Type": "Retweet",
        "Hashtags": "20",
        "Mentions": "0",
        " version ": 1696507291868594200
```

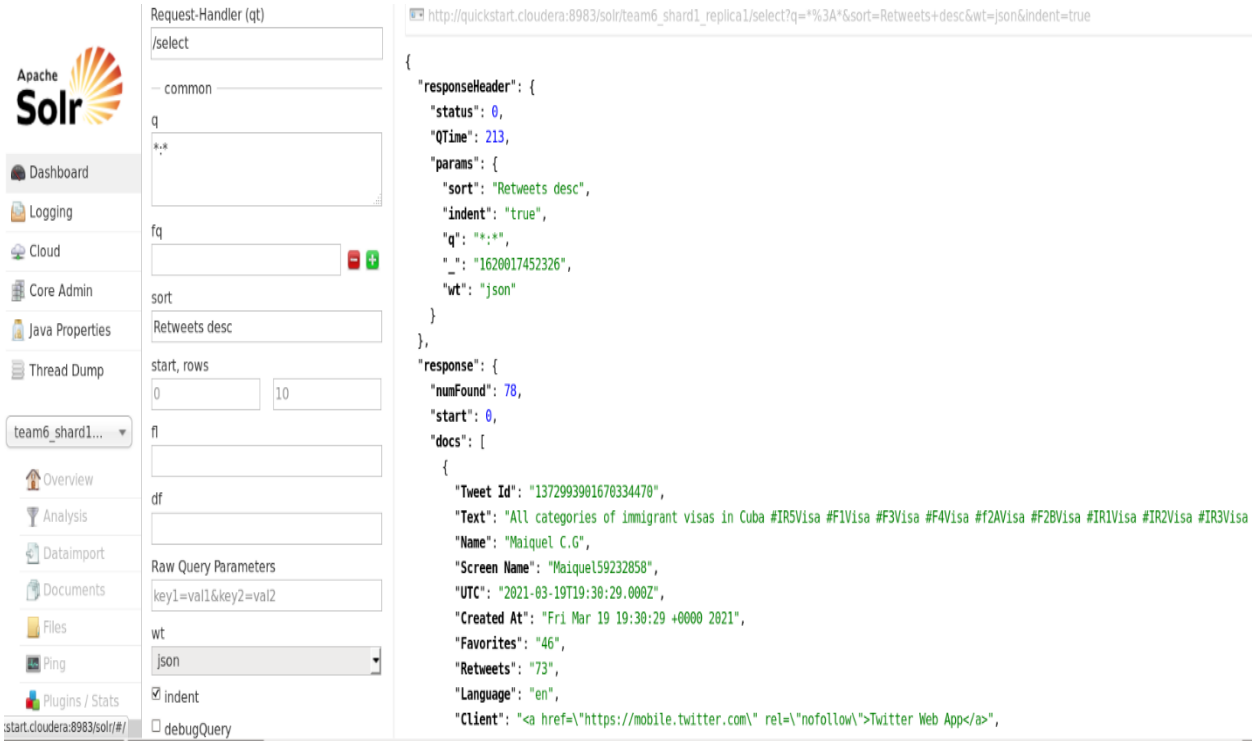
3. Pulled the data has 'F1Visa' included in the Text (Regex):



4. Collected the response with having Text:”student” and Text:”USA”:



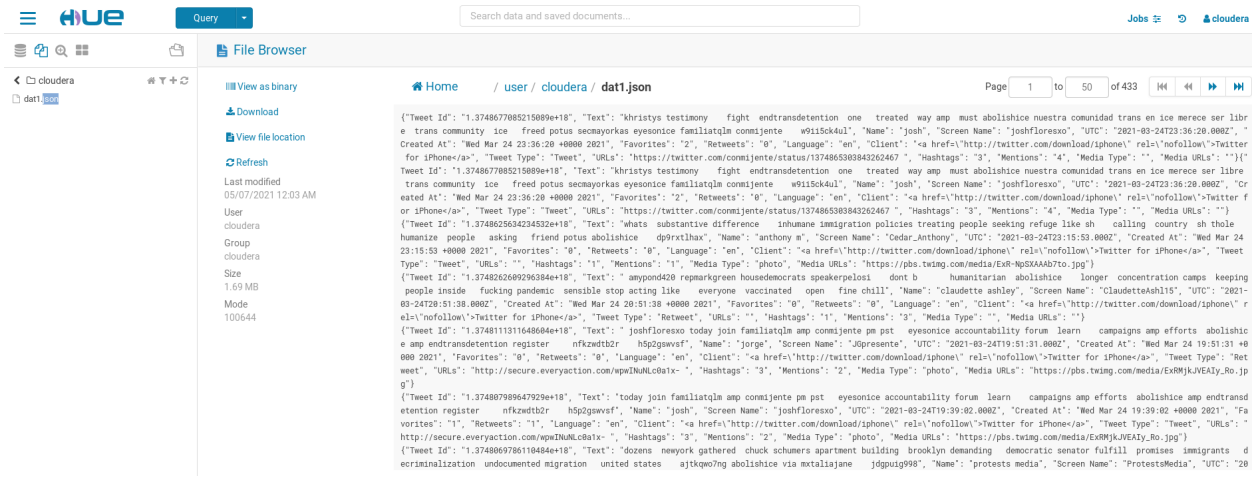
4. **Pulled the most retweeted tweets:**



Converting working dataset (in .csv) to .json file.

```
import json
import csv

with open('intlstudclean.csv', 'r') as input_file:
    reader = csv.DictReader(input_file)

    jsonoutput = 'dat1.json'
    with open(jsonoutput, 'a') as output_file:
        for row in reader:
            json.dump(row, output_file)
            output_file.write('\n')
```

Place the .json file in the cloudera working directory.



Spark using Scala

1. Starting the Scala using spark-shell command.

2. Load the 'dat1.json' into a data frame using sqlContext.read.json()

```
scala> val dfs = sqlContext.read.json("dat1.json")
dfs: org.apache.spark.sql.DataFrame = [Client: string, Created At: string, Favorites: string, Hashtags: string, Language: string, Media Type: string, Media URLs: string, Mentions: string, Name: string, Retweets: string, Screen Name: string, Text: string, Tweet Id: string, Tweet Type: string, URLs: string, UTC: string]

scala> dfs.show()
+--------------------+--------------------+---------+--------+--------+----------+--------------------+--------+--------------------+--------+--------------+--------------------+--------------------+----------+--------------------+--------------+
|              Client|          Created At|Favorites|Hashtags|Language|Media Type|         Media URLs|Mentions|                Name|Retweets|   Screen Name|                Text|            Tweet Id|Tweet Type|                URLs|           UTC|
+--------------------+--------------------+---------+--------+--------+----------+--------------------+--------+--------------------+--------+--------------+--------------------+--------------------+----------+--------------------+--------------+
|<a href="http://t...|Wed Mar 24 23:36:...|        2|       3|      en|          |                    |       4|               josh|       0|  joshfloresxo|khristys testimon...|1.374867708521508...|     Tweet|https://twitter.c...|03-24T23:36:...|
|<a href="http://t...|Wed Mar 24 23:15:...|        0|       1|      en|     photo|https://pbs.twimg...|       1|          anthony m|       0| Cedar_Anthony|whats  substantiv...|1.374862563423453...|     Tweet|                    |03-24T23:15:...|
|<a href="http://t...|Wed Mar 24 20:51:...|        0|       1|      en|          |                    |       3|   claudette ashley|       0|ClaudetteAshl15| amypond420 repma...|1.374826260929638...|   Retweet|                    |03-24T20:51:...|
|<a href="http://t...|Wed Mar 24 19:51:...|        0|       3|      en|     photo|https://pbs.twimg...|       2|              jorge|       0|    JGpresente| joshfloresxo tod...|1.374811131164860...|   Retweet|http://secure.eve...|03-24T19:51:...|
|<a href="http://t...|Wed Mar 24 19:39:...|        1|       3|      en|     photo|https://pbs.twimg...|       2|               josh|       1|  joshfloresxo|today join famili...|1.374807989647929...|     Tweet|http://secure.eve...|03-24T19:39:...|
|<a href="https://...|Wed Mar 24 19:35:...|        1|       2|      en|     video|https://video.twi...|       0|     protests media|       0| ProtestsMedia|dozens  newyork g...|1.374806978611048...|     Tweet|https://protests....|03-24T19:35:...|
|<a href="http://t...|Wed Mar 24 16:14:...|        0|       1|      en|          |                    |       0|      cozcacuauhtli|       0|      cozca503| xulxiyut cant sp...|1.374756574795767...|   Retweet|                    |03-24T16:14:...|
|<a href="http://t...|Wed Mar 24 16:10:...|       38|       1|      en|          |                    |       0|xul xi yut still ...|      11|      Xulxiyut|cant spell coloni...|1.374755625259868...|     Tweet|                    |03-24T16:10:...|
|<a href="http://t...|Wed Mar 24 13:03:...|        0|      19|      pt|     photo|https://pbs.twimg...|       0|               amam|       0|  amam_____|anticolonial amam...|1.374708410717986...|     Tweet|                    |03-24T13:03:...|
|<a href="https://...|Wed Mar 24 12:41:...|        0|       1|      en|          |                    |       1|               nlmh|       0|      NLMHlive|amazing work  mak...|1.374702978087674...|     Tweet|https://twitter.c...|03-24T12:41:...|
|<a href="http://t...|Wed Mar 24 03:56:...|        3|       1|      en|          |                    |       0|the road to aboli...|       0|   TENDEMANDS|cut funding  pros...|1.374570728796913...|     Reply|                    |03-24T03:56:...|
|<a href="http://t...|Wed Mar 24 03:28:...|        7|       2|      en|          |                    |       0|   yogottie perucha|       0|itsyo_qottieboo|today years ago i...|1.374563633829646...|     Tweet|                    |03-24T03:28:...|
|<a href="https://...|Wed Mar 24 03:20:...|        0|       4|      en|          |                    |       0|green party of ne...|       0| GreenPartyofNJ|hudson county pro...|1.374561836465524...|     Tweet|https://www.nj.co...|03-24T03:20:...|
|<a href="http://t...|Wed Mar 24 02:04:...|        0|       1|      en|          |                    |       0|        unimpressed|       0|  chilltadpole| happyfeminist  a...|1.374542603769766...|   Retweet|                    |03-24T02:04:...|
|<a href="http://t...|Wed Mar 24 01:50:...|        1|       1|      en|          |                    |       0|        cancelrent|       0| alejandro415sf| possible    happ...|1.374539073180762...|     Tweet|https://twitter.c...|03-24T01:50:...|
|<a href="http://t...|Wed Mar 24 00:11:...|        0|       6|      en|     photo|https://pbs.twimg...|       0|         salem snow|       0| Salem4Congress| maebe  girl im e...|1.374514115989831...|   Retweet|                    |03-24T00:11:...|
|<a href="http://t...|Tue Mar 23 18:04:...|        2|       3|      en|          |                    |       0|              ethan|       0|   EthanJClift| realize   kids d...|1.374421702755578...|     Tweet|                    |03-23T18:04:...|
```

3. Schema of the dataset.

```
scala> dfs.printSchema()
root
 |-- Client: string (nullable = true)
 |-- Created At: string (nullable = true)
 |-- Favorites: string (nullable = true)
 |-- Hashtags: string (nullable = true)
 |-- Language: string (nullable = true)
 |-- Media Type: string (nullable = true)
 |-- Media URLs: string (nullable = true)
 |-- Mentions: string (nullable = true)
 |-- Name: string (nullable = true)
 |-- Retweets: string (nullable = true)
 |-- Screen Name: string (nullable = true)
 |-- Text: string (nullable = true)
 |-- Tweet Id: string (nullable = true)
 |-- Tweet Type: string (nullable = true)
 |-- URLs: string (nullable = true)
 |-- UTC: string (nullable = true)
```

4. Display the Username and Text of top 20 tweets.

```
scala> dfs.select("Name", "Text").show()
+--------------------+--------------------+
|                Name|                Text|
+--------------------+--------------------+
|                josh|khristys testimon...|
|           anthony m|whats  substantiv...|
|    claudette ashley| amypond420 repma...|
|               jorge| joshfloresxo tod...|
|                josh|today join famili...|
|      protests media|dozens  newyork g...|
|       cozcacuauhtli| xulxiyut cant sp...|
|xul xi yut still ...|cant spell coloni...|
|                amam|anticolonial amam...|
|                nlmh|amazing work  mak...|
|the road to aboli...|cut funding  pros...|
|    yogottie perucha|today years ago i...|
|green party of ne...|hudson county pro...|
|         unimpressed| happyfeminist  a...|
|          cancelrent| possible    happ...|
|          salem snow| maebe  girl im e...|
|              ethan| realize   kids d...|
|       carolyn wolfe| ilovedogs4ever l...|
|justiceforfloyd f...|leaked photograph...|
|            angelica| jorgebepe today ...|
+--------------------+--------------------+
only showing top 20 rows
```

5. Count the number of Media Types present in the dataset.

```
scala> dfs.groupBy("Media Type").count().show()
+------------+-----+
|  Media Type|count|
+------------+-----+
|       photo|  964|
|animated_gif|   15|
|             | 1586|
|       video|   76|
+------------+-----+
```

6. Fetching the different languages in which the tweets are made about international students.

```
scala> val q3=sqlContext.sql("select Language, count(*) as count from stud  where Language is not null group by Language");
q3: org.apache.spark.sql.DataFrame = [Language: string, count: bigint]

scala> q3.show()
+--------+-----+
|Language|count|
+--------+-----+
|      fr|    6|
|      ta|    1|
|      te|    2|
|      th|    2|
|      tr|    4|
|      ur|    2|
|      in|    3|
|      pa|    7|
|      it|    6|
|     und|  208|
|      ja|    3|
|      pt|    9|
|      kn|    2|
|      ko|    1|
|      en| 2369|
|      es|   16|
+--------+-----+
```

7. Fetching users with a greater number of hashtags in their tweets on student data.

```
scala> val q2=sqlContext.sql("SELECT Name as Screen_Name,Text, Hashtags FROM stud  ORDER BY Hashtags DESC LIMIT 20");
q2: org.apache.spark.sql.DataFrame = [Screen_Name: string, Text: string, Hashtags: string]

scala> q2.show()
+-------------------+-------------------+--------+
|        Screen_Name|               Text|Hashtags|
+-------------------+-------------------+--------+
|stuart chen hayes...|ecmmason     abol...|       9|
|      louisa jourdan| louisjourdan10 b...|       9|
|   faith in new york|tomorrow join us ...|       9|
|       dr erin mason| schenhayes ecmma...|       9|
|genocide leader c...|banglaviral chad ...|       9|
|        gratefulness| complexion  prot...|       9|
|genocide leader c...|lora ries chad wo...|       9|
|             ap a t| endduopolynow se...|       9|
|       gratefulness|athe complexion  ...|       9|
|  gp monmouth county| greenpartyofnj r...|       9|
|                   d| abolishedwords b...|       9|
|joe biden is a j ...| greenpartyofnj g...|       9|
|        pat mcgroin|biden tells migra...|       9|
|      louisa jourdan| louisjourdan10 b...|       9|
|      louisa jourdan|breaking uk monar...|       9|
|      louisa jourdan|breaking uk monar...|       9|
|     end the duopoly|senators call   p...|       9|
|    challa law group|cases amp memos  ...|       9|
|     end the duopoly|senators call   p...|       9|
|          vvalidate|struggling  spot ...|       9|
+-------------------+-------------------+--------+
```

8. Fetching users who have more followers and are tweeting based on student data.

```
scala> val q4= sqlContext.sql("SELECT `Screen Name`, max(Favorites) as Favorites_count FROM stud  WHERE Text like '%international student%' group by `Screen Name` order by Favorites_count desc limit 15");
q4: org.apache.spark.sql.DataFrame = [Screen Name: string, Favorites_count: string]

scala> q4.show()
+--------------+--------------+
|   Screen Name|Favorites_count|
+--------------+--------------+
|  RKUniversity|             6|
|  Felician_edu|             5|
|opportunitiesfy|             5|
|Gurpart77345849|             4|
|GEMHub_official|             4|
|   GTaerospace|             4|
|StudyandStayPEI|             3|
|  VisaStudying|             3|
|RanaMUB53688693|             3|
|   EdUSAOAXACA|             2|
|       Epigeum|             2|
|alisonisonfigma|             2|
|AkshitChhabra26|             2|
|     sernexuss|             2|
|     StudyUCEM|             2|
+--------------+--------------+
```

9. Fetching users with a greater number of mentions in their tweets on student data.

```
scala> val q2=sqlContext.sql("SELECT Name as Screen_Name,Text, Mentions FROM stud  ORDER BY Mentions DESC LIMIT 20");
q2: org.apache.spark.sql.DataFrame = [Screen_Name: string, Text: string, Mentions: string]

scala> q2.show()
+------------------+------------------+--------+
|       Screen_Name|              Text|Mentions|
+------------------+------------------+--------+
|  full metal tuchas| rodneyr58127664 ...|      9|
|hi skilled immigrant| gcbcoalition day...|      9|
|     rodney roberts|dojtrump jbcarmod...|      9|
|hi skilled immigrant| gcbcoalition day...|      9|
|   majestic primate| rodneyr58127664 ...|      9|
|high skilled immi...| gcbcoalition day...|      9|
|     rodney roberts| latest h1bvisa a...|      9|
|hi skilled immigrant| gcbcoalition day...|      9|
|       protect jobs|h1b best wishes  ...|      9|
|hi skilled immigrant| gcbcoalition day...|      9|
|    gcbacklogwithms| gcbcoalition day...|      9|
|       protect jobs|joe biden giving ...|      9|
|high skilled immi...| gcbcoalition day...|      9|
|       protect jobs|tahminawatson eri...|      9|
|         keyur modi| gcbcoalition day...|      9|
|    gcbacklogwithms| gcbcoalition day...|      9|
|hi skilled immigrant| gcbcoalition day...|      9|
|hi skilled immigrant| gcbcoalition day...|      9|
|hi skilled immigrant| gcbcoalition day...|      9|
|hi skilled immigrant| gcbcoalition day...|      9|
+------------------+------------------+--------+
```

10. Fetching users having account names with a greater number of tweets on web series.

```
scala> val q4=sqlContext.sql("SELECT count(*) as count, Name from stud  where Name is not null group by Name order by count desc limit 10");
q4: org.apache.spark.sql.DataFrame = [count: bigint, Name: string]

scala> q4.show()
+-----+------------------+
|count|              Name|
+-----+------------------+
|  173|         australia|
|   98|   hireitpeople com|
|   44|  oya opportunities|
|   39|     rodney roberts|
|   32|                  |
|   32|           sanjeev|
|   19|       protect jobs|
|   18|hi skilled immigrant|
|   16|         amypond420|
|   15|         rup dhanda|
+-----+------------------+
```

11. Fetching users with a greater number of retweets for their tweets on student data.

```
scala> val q2=sqlContext.sql("SELECT Name as screen_name,Text,Retweets FROM stud  ORDER BY Retweets DESC LIMIT 20");
q2: org.apache.spark.sql.DataFrame = [screen_name: string, Text: string, Retweets: string]

scala> q2.show()
+------------------+------------------+--------+
|       screen_name|              Text|Retweets|
+------------------+------------------+--------+
|            sanjeev|jonasbrothers mic...|       9|
|     equal and fair|edusa india study...|       9|
|            sanjeev|flotus reminder d...|       9|
|        the h1b guy|   chance  sit   ...|       9|
|            sanjeev|flotus reminder d...|       9|
| detention watch dwn|new york join abo...|       9|
|            sanjeev|randpaul h1b prim...|       9|
|            sanjeev|jonasbrothers mic...|       9|
|    sikhnews247 com|pavitrachode stud...|       9|
|           strictly|republicans  maki...|       9|
|    rise and resist| manufactured mig...|       9|
|              mikko|really cool   pot...|       9|
|              mikko|really cool   pot...|       9|
|nord anglia educa...|weare proud    2n...|       9|
|    sikhnews247 com|pavitrachode stud...|       9|
|nord anglia educa...| proud    2nd yea...|       9|
|           strictly|republicans  maki...|       9|
|       avneet arora|internationalstud...|       9|
|        sbs punjabi|faced   uncertain...|      88|
|        sbs punjabi|faced   uncertain...|      88|
+------------------+------------------+--------+
```

Spark using Python

Considered python programming because it is much quicker than Scala data-frames in terms of execution time – Visualization is achieved with Matplotlib and Seaborn in python programming.

1. Download Java to run the Java Virtual Machine (JVM).

```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
```

```
[2]  !wget -q https://www-us.apache.org/dist/spark/spark-3.1.1/spark-3.1.1-bin-hadoop2.7.tgz
```

```
!tar xf spark-3.1.1-bin-hadoop2.7.tgz
```

2. Set the environment path. This will enable us to run Pyspark in the Collab environment.

```
[5]  import os

     os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"

     os.environ["SPARK_HOME"] = "/content/spark-3.1.1-bin-hadoop2.7"
```

3. Import SparkSession from pyspark.sql and create a SparkSession.

```
from pyspark.sql import SparkSession

spark = SparkSession.builder\
.master("local")\
.appName("Amar123")\
.config('spark.ui.port', '4050')\
.getOrCreate()
```

```
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
v3.1.1
Master
local
AppName
Amar123

4. Import the dataset and create data frames directly on import.

```
df=spark.read.csv(r"/content/drive/MyDrive/intlstudclean.csv", header=True)
df.createOrReplaceTempView("stuData")
```

```
df
```

```
DataFrame[Tweet Id: string, Text: string, Name: string, Screen Name: string, UTC: string, Created At: string,
```

5. Check for duplicate records and null values in the dataset.

```
[37] df=df.dropDuplicates()
```

```
[35] df.count()
```

```
2641
```

```
df.groupBy(df.columns).count().where(fn.col('count')>1).select(fn.sum('count')).show()
```

```
+----------+
|sum(count)|
+----------+
|      null|
+----------+
```

Queries:

1. Display first 100 rows order by the Date tweeted.

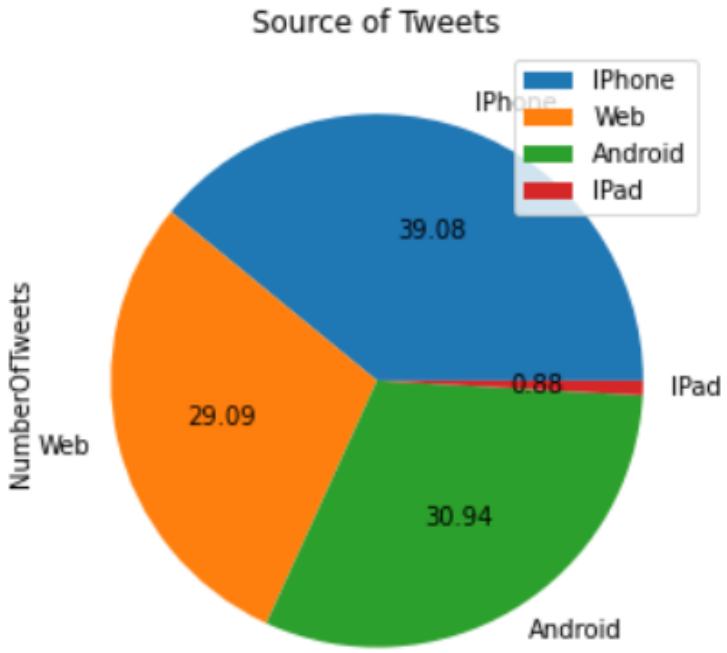
```
spark.sql("select `Created At`, `Screen Name`, Text from stuData ORDER BY `Created At` limit 25").show(100)
```

```
+--------------------+---------------+--------------------+
|          Created At|    Screen Name|                Text|
+--------------------+---------------+--------------------+
|Fri Mar 19 00:01:...|       coc_isp|curious   coc han...|
|Fri Mar 19 00:04:...|   amam_______|acab abolishice c...|
|Fri Mar 19 00:04:...|   amam_______|acab abolishice c...|
|Fri Mar 19 00:12:...|   PROTECTJOBS1|millions  america...|
|Fri Mar 19 00:32:...|   annarborandy|another reason  a...|
|Fri Mar 19 00:36:...|      DiyaCBose|want  stand   fam...|
|Fri Mar 19 00:56:...|      DoctorLix|according   dept ...|
|Fri Mar 19 00:58:...|    revmitulski|migrationisahuman...|
|Fri Mar 19 00:58:...|    revmitulski|migrationisahuman...|
|Fri Mar 19 00:59:...|      flaviajim|criminalization  ...|
|Fri Mar 19 01:03:...|         ACLUMN| doctorlix accord...|
|Fri Mar 19 01:17:...|       MoCo_DSA|thank  delegatest...|
|Fri Mar 19 01:30:...|   migratesmart|internationalstud...|
|Fri Mar 19 01:30:...|   VisaStudying|    popular study...|
|Fri Mar 19 01:33:...|    CarolStern1| moco dsa thank  ...|
|Fri Mar 19 01:37:...| GabrielAcevero|     also abolishice|
|Fri Mar 19 01:57:...|  TouchstoneEdu| experts help  pr...|
|Fri Mar 19 02:07:...|       VicGovAu| studymelbourne  ...|
|Fri Mar 19 02:09:...|        LEAPWSU|tbt scavenger hun...|
|Fri Mar 19 02:09:...|     FirstecInt|study work live  ...|
|Fri Mar 19 02:11:...|interstellaruth| interstellaruth ...|
|Fri Mar 19 02:11:...|interstellaruth| interstellaruth ...|
|Fri Mar 19 02:11:...|SunriseMaryland|abolishice dignit...|
|Fri Mar 19 02:14:...|    diversityup| leapwsu tbt scav...|
|Fri Mar 19 02:16:...| GreenPartyofNJ|cops used anti mu...|
+--------------------+---------------+--------------------+
```

2. Using the Union operation, count the number of tweets depending on the data sources.

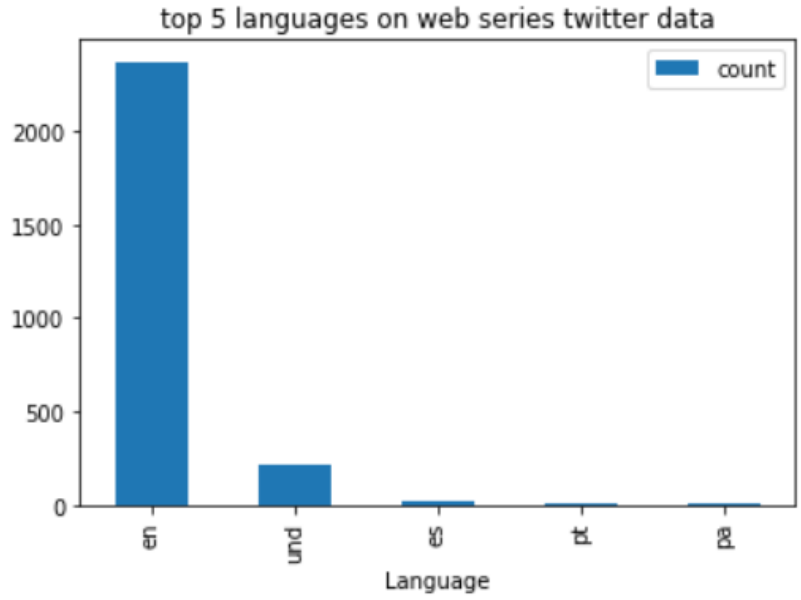
```
q1= spark.sql("select count(*) as NumberOfTweets, 'Android' as Source from stuData where Client like '%Twitter
pd1 = q1.toPandas()
print(pd1)
pd1.plot.pie(y='NumberOfTweets', labels=['IPhone', 'Web', 'Android', 'IPad'], autopct='%.2f', figsize=(5, 5),
                 title="Source of Tweets").figure
```

	NumberOfTweets	Source
0	888	Web
1	661	Android
2	703	IPhone
3	20	IPad



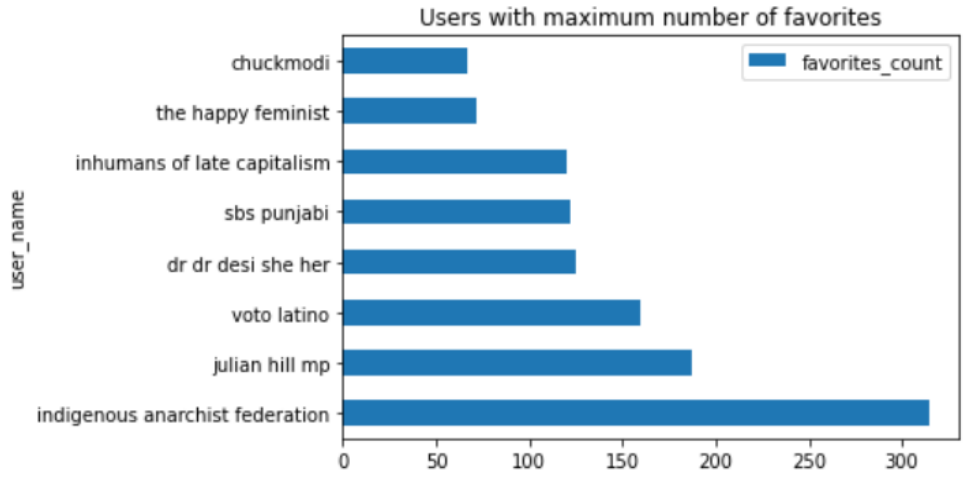
3. Count the tweets by grouping them into languages and sorting them in descending order, starting with the top five.

```
query3 = spark.sql("select count(*) as count,Language as Language from stuData where Language is not null group by Language order by count desc limit 5")
print(query3)
pd3 = query3.toPandas()
pd3.plot(kind="bar", x="Language", y="count", title="top 5 languages on web series twitter data").figure
```



4. Pull the top 10 users based on favorites count by dropping the duplicates.

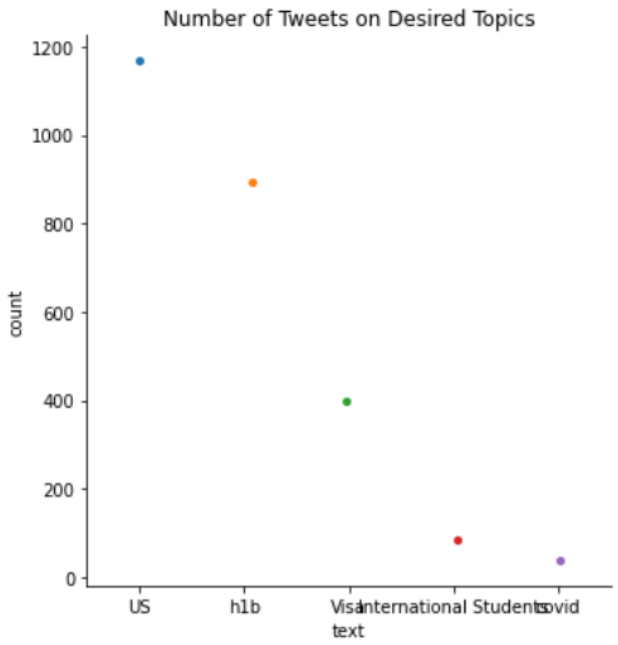
```
query4 = spark.sql(
        "select Name as user_name, int(Favorites) as favorites_count from stuData order by favorites_count desc limit 10").dropDuplicates()
pd4 = query4.toPandas()
pd4.plot.barh(x='user_name',y='favorites_count',title="Users with maximum number of favorites").figure
```



5. Pull the count of tweets based on the desired topics from the Text attribute.

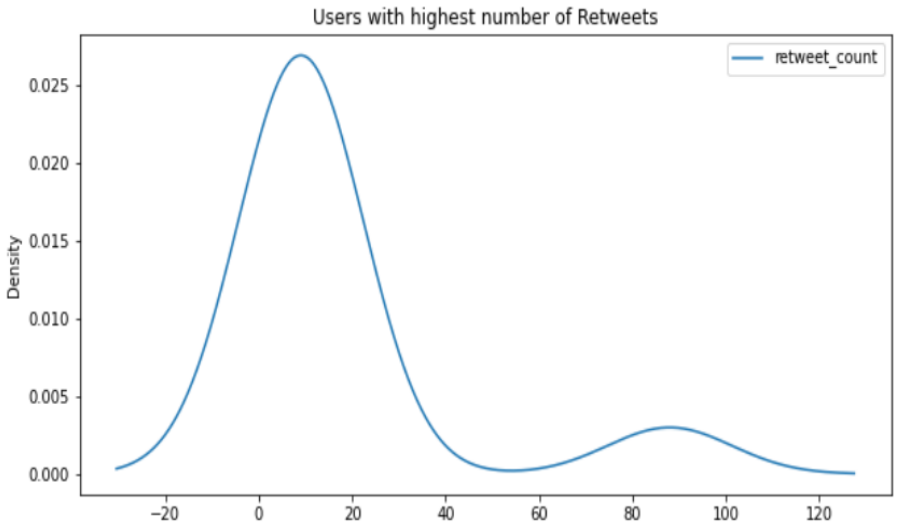
```
query6 = spark.sql(
        "select count(*) as count,q.Text as text from (select case when Text like '%f1visa%' then 'F1 Visa' when Text like '%internati
pd6 = query6.toPandas()
print(pd6)
plot= sns.catplot(x='text', y='count', data=pd6).set(title="Number of Tweets on Desired Topics")
plot.fig
```

	count	text
0	1169	US
1	893	h1b
2	398	Visa
3	85	International Students
4	38	covid



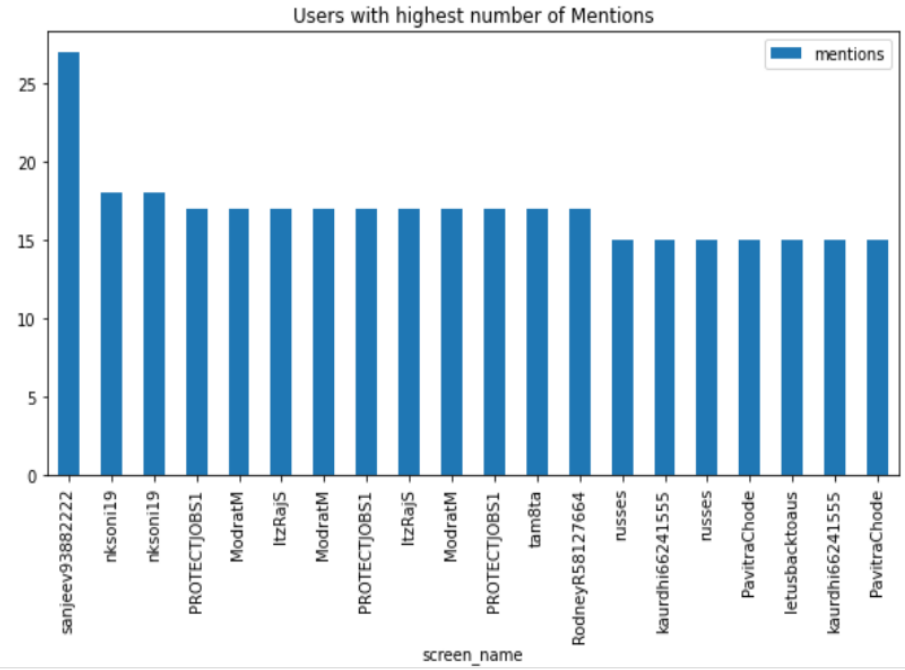
6. Display the screen names of tweets that are mostly retweeted.

```
query5 = spark.sql(
        "SELECT `Screen Name` as screen_name, int(Retweets) as retweet_count FROM stuData ORDER BY Retweets DESC LIMIT 20")
pd5 = query5.toPandas()
pd5.plot.density(x="screen_name", y="retweet_count",title='Users with highest number of Retweets',figsize=(10,5)).figure
```



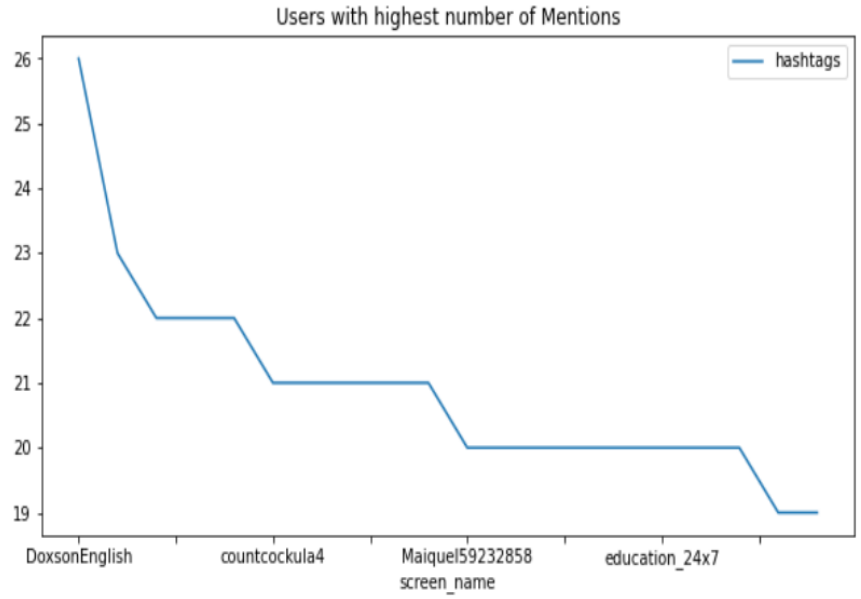
7. Display the screen name of tweets that has highest mentions.

```
query6 = spark.sql(
        "SELECT `Screen Name` as screen_name, int(Mentions) as mentions FROM stuData ORDER BY Mentions DESC LIMIT 20")
pd6 = query6.toPandas()
pd6.plot.bar(x="screen_name", y="mentions",title='Users with highest number of Mentions',figsize=(10,5)).figure
```



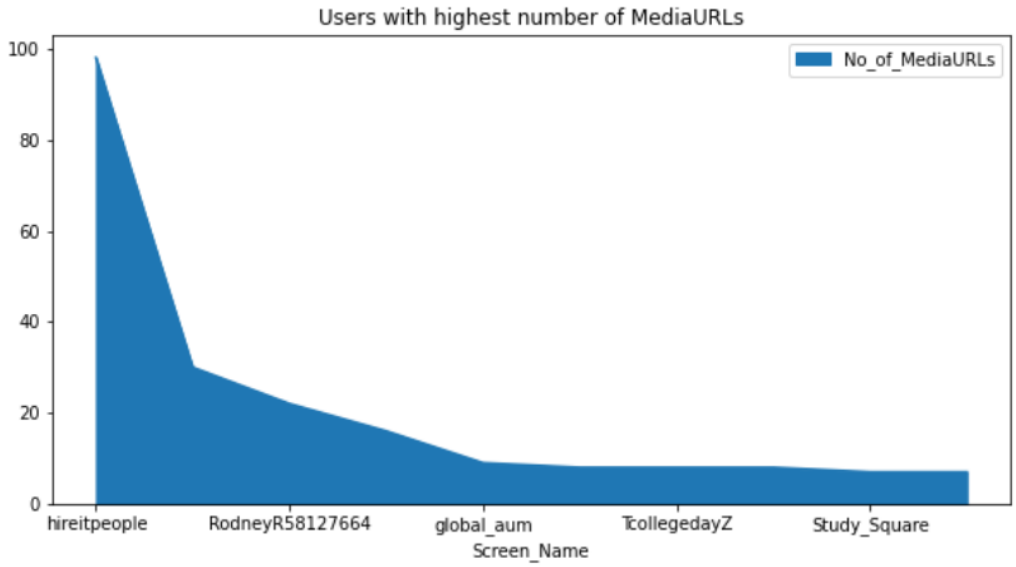
8. Display the screen name of tweets that has most hashtags in it.

```
query7 = spark.sql(
        "SELECT `Screen Name` as screen_name, int(Hashtags) as hashtags FROM stuData ORDER BY Hashtags DESC LIMIT 20")
pd7 = query7.toPandas()
pd7.plot.line(x="screen_name", y="hashtags",title='Users with highest number of Mentions',figsize=(10,5)).figure
```



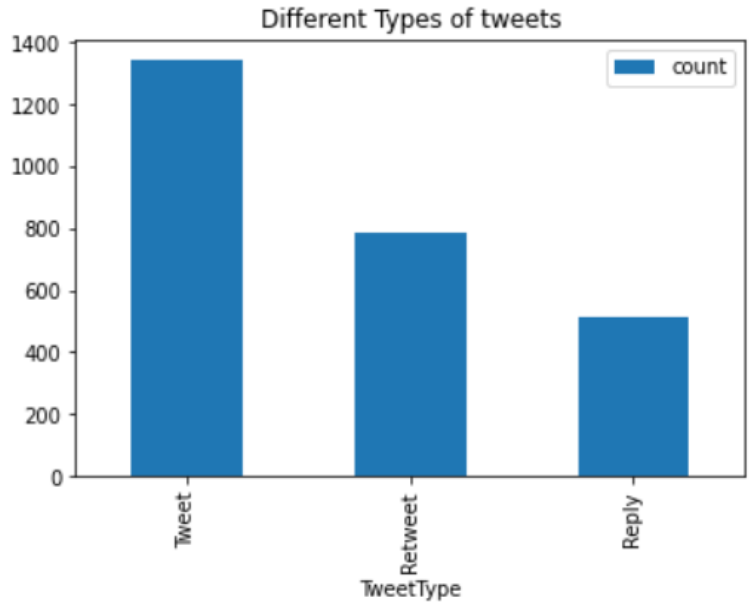
9. Plot the Users with the highest number of Media URLs.

```
query8= spark.sql("select `Screen Name` as Screen_Name,count(`Media URLs`) as No_of_MediaURLs from stuData group by `Screen Name`order by No_of_MediaURLs DESC

pd8 = query8.toPandas()
pd8.plot.area(x="Screen_Name", y="No_of_MediaURLs",title='Users with highest number of MediaURLs',figsize=(10,5)).figure
```



10. Display the count of different types of tweets (Retweet, Reply or Tweet).

```
query9 = spark.sql("select count(*) as count,`Tweet Type` as TweetType from stuData where `Tweet Type` is not null group by `Tweet Type` order by count desc limit 5")
print(query9)
pd9 = query9.toPandas()
pd9.plot(kind="bar", x="TweetType", y="count", title="Different Types of tweets").figure
```



11. Analysing the user data with pyspark:

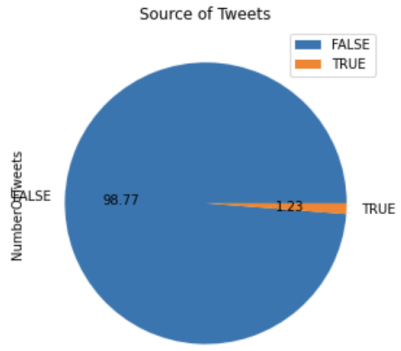
```
spark.sql("select `Tweets`, `Screen Name`, Location from userData ORDER BY `Tweets` desc limit 25" ).show(100)
```

```
+--------------------+--------------------+--------------------+
|              Tweets|         Screen Name|            Location|
+--------------------+--------------------+--------------------+
|                True|                 IPA|                null|
|Huntsville AL+Hat...|  minoring in music|                True|
|               False|   @QS_Intelligence|                null|
|               False|        Spouse visa|                null|
|               False|               music|                null|
|               False|   @QS_Intelligence|                null|
|               False|   @QS_Intelligence|                null|
|                9986|    GCBacklogWithMS|                null|
|               99711|        giddyupbill|                null|
|                 997|       chugh_ritesh|                null|
|                9969|    liberty_immigrn|                null|
|                 996|      essayexxperts|          MIAMI, USA|
|                 995|         ria47744809|                null|
|                 995|         ria47744809|                null|
|                 995|         ria47744809|                null|
|                9944|       AbreuMarioly|                null|
|                9928|    ComradeFeathers|                null|
|                 991|     Lakshmi_Sydney|                null|
|                  99|    hsktechnologies|New Jersey 08854,...|
|                  99|    Gurdeep29241677|                null|
|                9896|            OzLady0|Melbourne, Australia|
|               98885|     gingerjonesNYC|              NY, NY|
|                 987|         harithtwee|     Hyderabad, India|
|                9847|     hutchesonglenn|       Noongar boodja|
|                9847|        2colourinme|                null|
+--------------------+--------------------+--------------------+
```

12. Checking the number of verified and unverified accounts tweeted about the international student cause.

```
q1= spark.sql("select count(*) as NumberOfTweets, 'TRUE' from userData where Verified like '%T%' UNION select

pd1 = q1.toPandas()
print(pd1)
pd1.plot.pie(y='NumberOfTweets', labels=['FALSE', 'TRUE'], autopct='%.2f', figsize=(5, 5),
                  title="Source of Tweets").figure


```

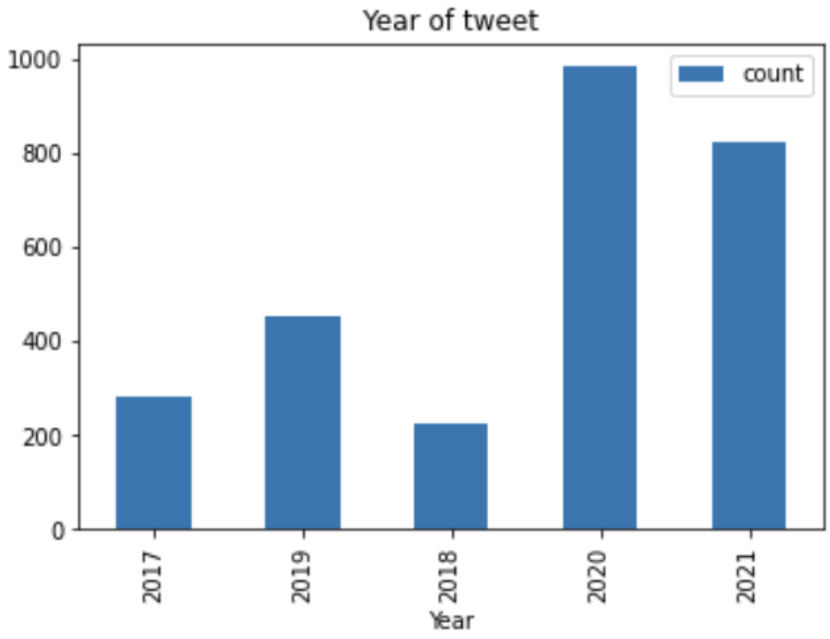
```
   NumberOfTweets   TRUE
0            4351  FALSE
1              54   TRUE
```



13. From this visualization it shows that there were more tweets in the year 2020 specifically as compared to other years since international students were in the talk in this year due to

the corona pandemic, when ICE decided to change the immigration rules and h1b sponsorship was affected due to lack of jobs and recession.

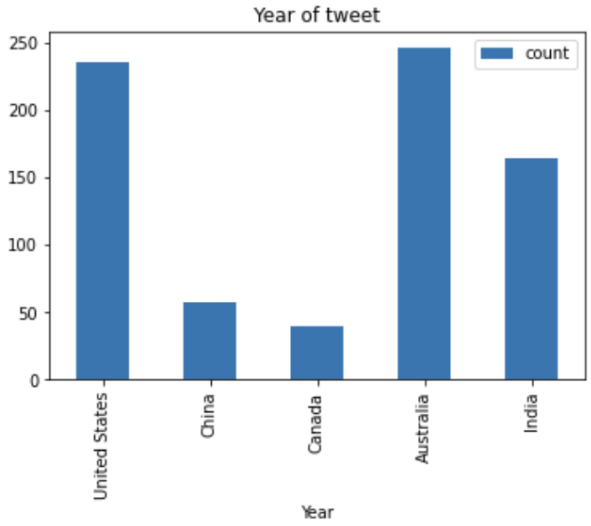
```
DataFrame[count: bigint, Year: string]
```



14. Here are the countries that tweeted most and the least regarding international student scenario in United States specifically.

```
l("select count(*) as count, 'United States' as Year from use

ndas()
r", x="Year", y="count", title="Year of tweet").figure
```

```
DataFrame[count: bigint, Year: string]
```



Pyspark, Spark SQL, NLP and SQL queries for both users and tweets:

Here are 2 datasets, one covers the information about the tweets, and other about the users.

Load / Cache Data

- Spark dataframe should split into partitions = 2-3x the no. threads available in your CPU or cluster. I have 2 cores, with 2 threads each = 4, and I chose 3x, ie. 12 partitions, based on experimentation.
- Then cache tables: you can see in Spark GUI that 12 partitions are cached for each file.
- The Shuffle Read is default to 200, we don't want this to be the bottleneck, so we set this equal to partitions in our data, using spark.sql.shuffle.partitions. This is specific to wide shuffle transformations (e.g. GROUP BY or ORDER BY) that may be performed later on, and how many partitions this operation sets up to read the data.

```
[22]  1 tweets = spark.read.csv(r"/content/intlstud.csv", header=True)
      2 users = spark.read.csv(r"/content/intluserdata_csv.csv", header=True)
```

```
[23]  1 spark.sql('SET spark.sql.shuffle.partitions=12')
```

```
DataFrame[key: string, value: string]
```

Changing the Spark SQL Dataframe Column type from one data type to another data to make the analysis more accurate and meaningful.

EDA

Users data

```
1 #change the Spark SQL DataFrame column type from one data type to another data
2 users = users.withColumn("Followers",col("Followers").cast("long"))
3 users = users.withColumn("Following",col("Following").cast("long"))
4 users = users.withColumn("Favorites",col("Favorites").cast("long"))
```

```
[68]  1 users.dropna()
```

```
DataFrame[User Id: string, Name: string, ScreenName: string, UTC: string, Created At: string, Followers: bigint,
```

```
[69]  1 users.printSchema()
```

```
root
 |-- User Id: string (nullable = true)
 |-- Name: string (nullable = true)
 |-- ScreenName: string (nullable = true)
 |-- UTC: string (nullable = true)
 |-- Created At: string (nullable = true)
 |-- Followers: long (nullable = true)
 |-- Following: long (nullable = true)
 |-- Favorites: long (nullable = true)
 |-- Tweets: string (nullable = true)
 |-- Lists: string (nullable = true)
 |-- Bio: string (nullable = true)
 |-- Location: string (nullable = true)
 |-- URL: string (nullable = true)
 |-- Verified: string (nullable = true)
 |-- Default Profile: string (nullable = true)
```

Spark SQL Queries

```
query = '''
SELECT ScreenName, Followers
FROM users
ORDER BY Followers DESC
'''
spark.sql(query).show()
```

```
+---------------+---------+
|     ScreenName|Followers|
+---------------+---------+
|      the_hindu|  6712772|
|       siasatpk|  1282984|
|       BT_India|  1103741|
|moneycontrolcom|  1056504|
|moneycontrolcom|  1056504|
|FinancialXpress|   717645|
|      ETNOWlive|   633186|
|    USinNigeria|   366188|
|    USinNigeria|   366188|
| W_Angels_Wings|   153401|
|    Matt_Pinner|   142387|
|  MikeCarlton01|   141252|
|    zellieimani|   139273|
|    npquarterly|   131410|
|     votolatino|   114563|
|    Study_INTNL|   106727|
|       NH_India|   104826|
|     julietkego|    63347|
|michaelamilesau|    62833|
|      dundeeuni|    60013|
+---------------+---------+
only showing top 20 rows
```

```
query = '''
SELECT ScreenName, Following
FROM users
ORDER BY Following DESC
'''
spark.sql(query).show()
```

```
+---------------+---------+
|     ScreenName|Following|
+---------------+---------+
|    Matt_Pinner|   144327|
| W_Angels_Wings|   124295|
|   smartdissent|    53687|
|    wordrefiner|    50729|
|      56perumal|    46539|
|   ResisterChic|    31081|
|      bigtickHK|    28913|
|      bigtickHK|    28913|
|    aajeel_dars|    24315|
|michaelamilesau|    21426|
| Kraven_Raven24|    20412|
|        IDPDRIE|    20201|
|  mikecoulson48|    19812|
|     julietkego|    18894|
|     Dicedotcom|    18362|
|     Dicedotcom|    18362|
|        rk70534|    17307|
|RedHairnBlkLthr|    16810|
|      AaronDodd|    16269|
|  DavidBeazley4|    15826|
+---------------+---------+
only showing top 20 rows
```

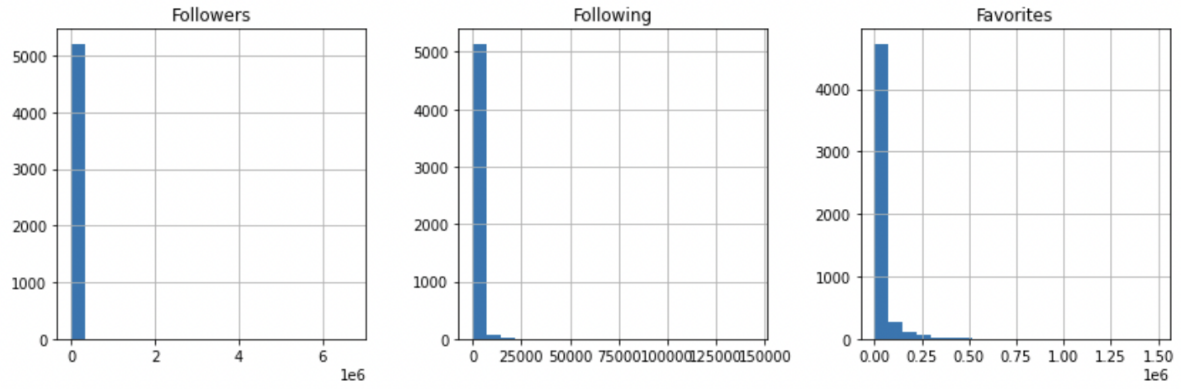
```
query = '''
SELECT name, Favorites
FROM users
ORDER BY Favorites DESC
'''
spark.sql(query).show()
```

```
+--------------------+---------+
|                name|Favorites|
+--------------------+---------+
|      m.e.h. #MaskUp|  1485858|
|        Déri Szilvia|  1306010|
|         Geoff Payne|  1245298|
|     Susan Mackay 😱|  1055133|
|             anodyne|  1046525|
|   Raimo Kangasniemi|   926954|
|SENSE💥OF💥OUTRAG...|   926117|
|"The cutie Goat-F...|   902959|
|  Christine McNichol|   812277|
|    NAZIR AHMED DARS|   737753|
|Death will not re...|   723494|
|  Battle Weary Woman|   685970|
|no justice - just...|   646488|
|Cassie, Bimbo at Law|   627215|
|       Leslie F. ✊🏿|   521528|
|🕯️DebbieN Credibl...|   516551|
|🌈Immigrants ...|   481996|
|🌈Immigrants ...|   481996|
|Tokyo Free Skater...|   473838|
|          Anne Day💜|   472313|
+--------------------+---------+
only showing top 20 rows
```

The total number of verified users who participated in tweeting for the international students, their followers, the people they are following and the favorites.

```
print('Total no. verified users on twitter:', users.filter('Verified == "TRUE"').count())
print("\n Below we see there are a few members with a large following. These could be influential i

fig, ax = plt.subplots(ncols=3, figsize=(14,4))
users.select('Followers').toPandas().hist(bins=20, ax=ax[0])
users.select('Following').toPandas().hist(bins=20, ax=ax[1])
users.select('Favorites').toPandas().hist(bins=20, ax=ax[2]);
```

```
Total no. verified congressmembers on twitter: 54

 Below we see there are a few members with a large following. These could be influential in pushing the
```



Tweets Data

Tweets data

```
[77]  1 tweets = tweets.withColumn("Favorites",col("Favorites").cast("int"))
      2 tweets = tweets.withColumn("Retweets",col("Retweets").cast("int"))
      3 tweets = tweets.withColumn("Hashtags",col("Hashtags").cast("int"))
      4 tweets = tweets.withColumn("Mentions",col("Mentions").cast("int"))
```

```
[78]  1 tweets.printSchema()
```

```
root
 |-- Tweet Id: string (nullable = true)
 |-- Text: string (nullable = true)
 |-- Name: string (nullable = true)
 |-- Screen Name: string (nullable = true)
 |-- UTC: string (nullable = true)
 |-- Created At: string (nullable = true)
 |-- Favorites: integer (nullable = true)
 |-- Retweets: integer (nullable = true)
 |-- Language: string (nullable = true)
 |-- Client: string (nullable = true)
 |-- Tweet Type: string (nullable = true)
 |-- URLs: string (nullable = true)
 |-- Hashtags: integer (nullable = true)
 |-- Mentions: integer (nullable = true)
 |-- Media Type: string (nullable = true)
 |-- Media URLs: string (nullable = true)
```

```
query = '''
SELECT tweets.Mentions AS Mentions, COUNT(*) as cnt
FROM tweets
GROUP BY Mentions
ORDER BY cnt DESC
'''
spark.sql(query).show()
```

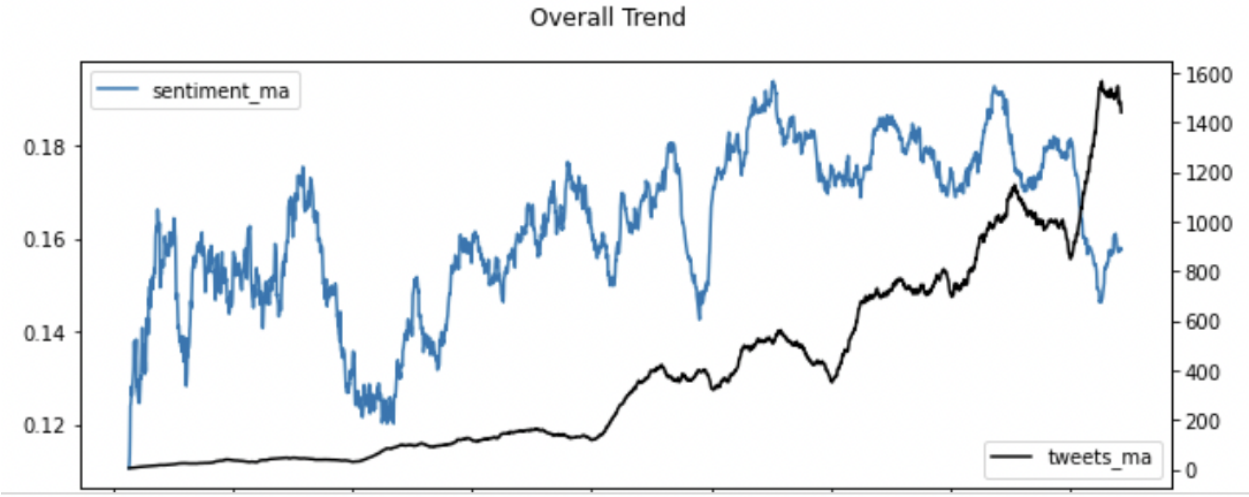
```
+--------+----+
|Mentions| cnt|
+--------+----+
|       0|6168|
|       1|1052|
|       2| 702|
|       3| 420|
|       7| 334|
|       4| 298|
|       5| 244|
|       9| 210|
|       6| 200|
|       8| 102|
|      10|  80|
|      15|  26|
|      13|  18|
|      14|  16|
|      12|  14|
|      17|  14|
|      11|  12|
|      18|   6|
|      27|   4|
+--------+----+
```

[97]
```
query = '''
SELECT tweets.Hashtags AS Hashtags, COUNT(*) as cnt
FROM tweets
GROUP BY Hashtags
ORDER BY cnt DESC
'''
spark.sql(query).show()
```

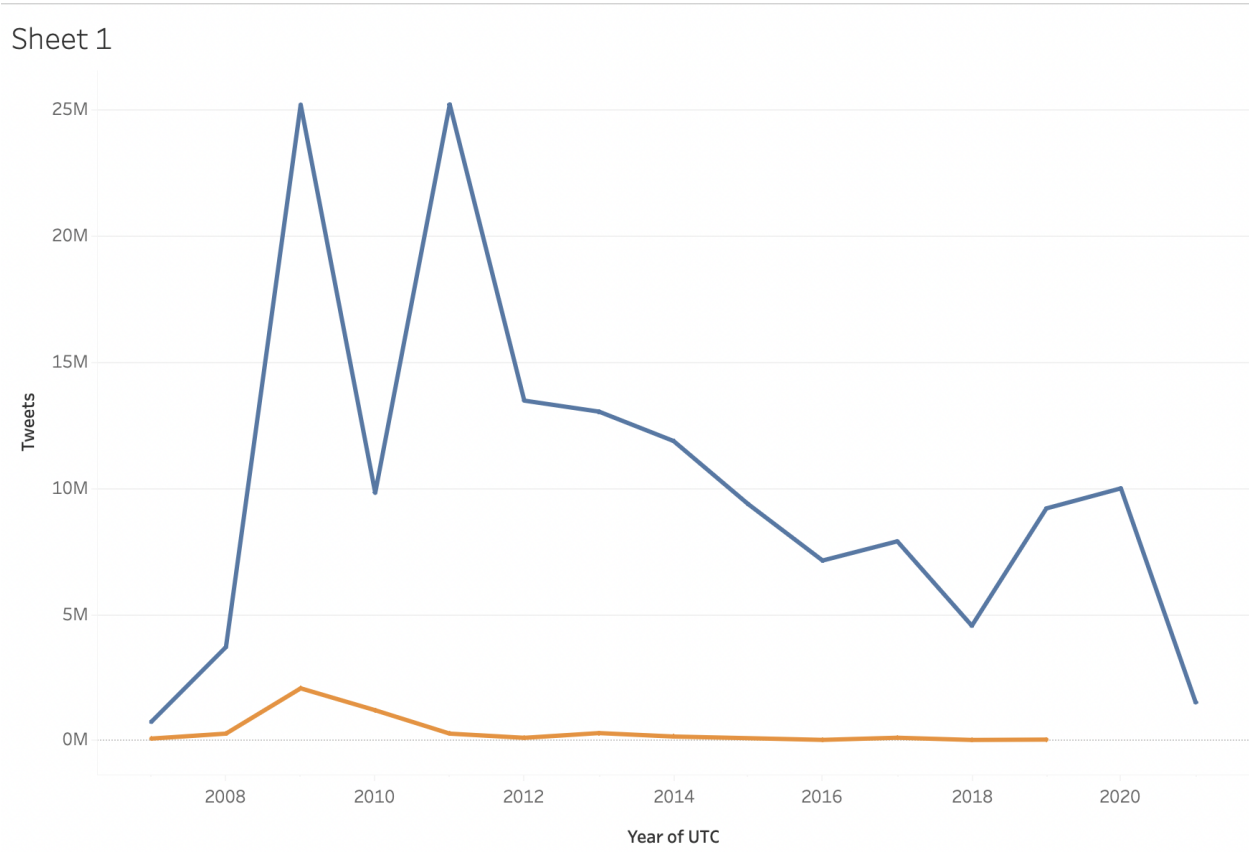
```
+--------+----+
|Hashtags| cnt|
+--------+----+
|       5|2062|
|       1|1686|
|       3|1450|
|       2|1242|
|       4|1070|
|       6| 532|
|       7| 342|
|       9| 264|
|       8| 224|
|      10| 180|
|      20| 166|
|      11| 158|
|      13| 148|
|      12| 110|
|      14|  98|
|      15|  72|
|      17|  34|
|      18|  22|
|      16|  20|
|      19|  16|
+--------+----+
only showing top 20 rows
```

Sentiment Analysis Trend with time for the year 2020:-

```
1 plot_trend('Overall Trend')
```



Visualization of the trend of the tweets by verified and non verified users for the different years regarding the international students and ICE immigration rule changes, It also includes the tweets about h1b sponsorship.



HIVE:

Created the database and database table in hive and loaded the data into hive table for AbolishICE dataset:

```
hive> create table AbolishICE (Tweet_Id string,Text string,Name string,Screen_Name string,UTC string,Created_At string,Favorites string,Retweets string,Language string,Client string,Tweet_Type string,URLs string,Hashtags string,Mentions
string,Media_Type string,Media_URLs string) row format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 0.353 seconds
```

```
hive> load data local inpath '/home/cloudera/Downloads/bdphive/AbolishICE.csv' into table AbolishICE;
Loading data to table default.abolishice
Table default.abolishice stats: [numFiles=1, totalSize=645903]
OK
Time taken: 0.598 seconds
hive> select * from AbolishICE limit 2;
OK
Tweet Id        Text    Name    Screen Name     UTC     Created At      Favorites       Retweets        Language        Client  Tweet Type      URLs    Hashtags        Mentions        Media Type      Media URLs
1374885870491197441     "RT @abolishICE_nynj : After living in the US for nearly his whole life  Hieu and 32 other Vietnamese refugees were deported by @POTUS and @VP last week. Please support his re-entry by donating to this fundraiser
set up by @vietlead and @MekongNYC. If you don't have Venmo let us know. #AbolishICE https://t.co/T71JNy7d8O"   carb    carbnim 2021-03-25T00:48:30.000Z        Thu Mar 25 00:48:30 +0000 2021  0       0       en      "<a href=""https://mo
bile.twitter.com"" rel=""nofollow"">Twitter Web App</a>"         Retweet         1       4       photo
Time taken: 0.382 seconds, Fetched: 2 row(s)
hive>
```

Created the database and database table in hive and loaded the data into hive table for F1visa dataset:

Visualized the output:

```
hive> create table F1visa (Tweet_Id string,Text string,Name string,Screen_Name string,UTC string,Created_At string,Favorites string,Retweets string,Language string,Client string,Tweet_Type string,URLs string,Hashtags string,Mentions stri
ng,Media_Type string,Media_URLss string) row format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 0.071 seconds
hive> load data local inpath '/home/cloudera/Downloads/bdphive/F1visa.csv' into table F1visa;
Loading data to table default.f1visa
Table default.f1visa stats: [numFiles=1, totalSize=40034]
OK
Time taken: 0.257 seconds
hive> select * from F1visa limit 3;
OK
Tweet Id        Text    Name    Screen Name     UTC     Created At      Favorites       Retweets        Language        Client  Tweet Type      URLs    Hashtags        Mentions        Media Type      Media URLs
1374834250667716608     "RT @Maiquel59232858 : All categories of immigrant visas in Cuba #IR5Visa #F1Visa #F3Visa #F4Visa #f2AVisa #F2BVisa #IR1Visa #IR2Visa #IR3Visa #IR4Visa #CR1Visa #CR2Visa #K1Visa #K2Visa #K3Visa #K4Visa #DV2020 #DV
2021 &amp;amp; #CFRP should be processed in our country      until when?     NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL
 #ConsularServisesInCuba"       Neibis Aguilar  AguilarNeibis   2021-03-24T21:23:23.000Z        Wed Mar 24 21:23:23 +0000 2021  0       0       en      "<a href=""https://mobile.twitter.com"" rel=""nofollow"">Twitter Web App</a>"   Retwe
et      20      0       NULL
Time taken: 0.064 seconds, Fetched: 3 row(s)
hive>
```

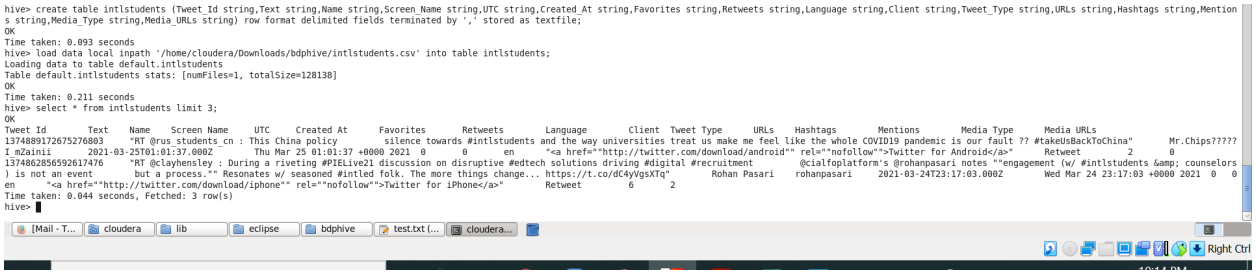
Created the database and database table in hive and loaded the data into hive table for h1b dataset:

Visualized the output:

```
hive> create table h1b (Tweet_Id string,Text string,Name string,Screen_Name string,UTC string,Created_At string,Favorites string,Retweets string,Language string,Client string,Tweet_Type string,URLs string,Hashtags string,Mentions string,
Media_Type string,Media_URLs string) row format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 0.106 seconds
hive> load data local inpath '/home/cloudera/Downloads/bdphive/h1b.csv' into table h1b;
Loading data to table default.h1b
Table default.h1b stats: [numFiles=1, totalSize=591164]
OK
Time taken: 0.339 seconds
hive> select * from h1b limit 2;
OK
Tweet Id        Text    Name    Screen Name     UTC     Created At      Favorites       Retweets        Language        Client  Tweet Type      URLs    Hashtags        Mentions        Media Type      Media URLs
1374887541896216576     "Mohammed Raheem's drive back home ?? home turned soar   please don't for brothers surgery https://t.co/Vhfd01vEiB #Sanjose #muslim #muslimah #MuslimLivesMatter #veinbust #Indian #H1B #f1 visa"       Al Hindi ????
indimuslim101   2021-03-25T00:55:08.000Z        Thu Mar 25 00:55:08 +0000 2021  0       0       en      "<a href=""http://twitter.com/download/iphone"" rel=""nofollow"">Twitter for iPhone</a>"         Tweet   https://www.gofundme.com/f/pl
ease-help-for-the-surgery-of-mohammed-raheem?utm_source=twitter&utm_medium=social&utm_campaign=m_pd+share-sheet         8       0
Time taken: 0.107 seconds, Fetched: 2 row(s)
hive>
```

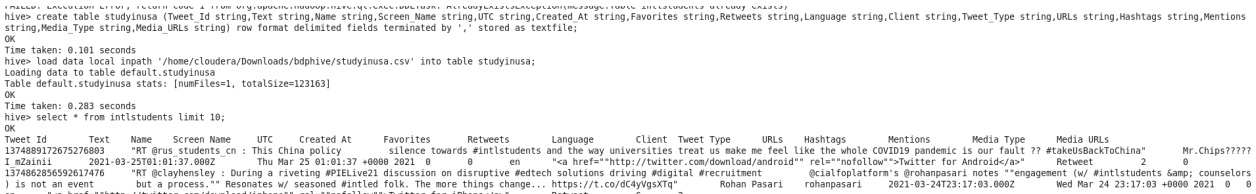
Created the database and database table in hive and loaded the data into hive table for intlstudents dataset:

Visualized the output:

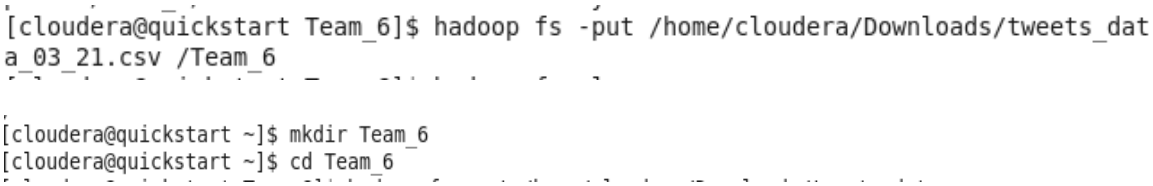


Created the database and database table in hive and loaded the data into hive table for study in USA dataset:

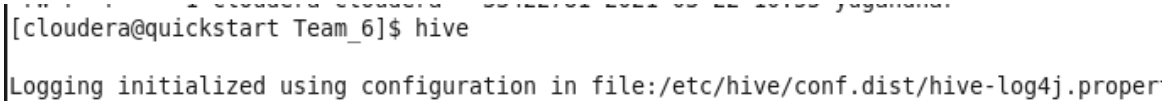
Visualized the output:

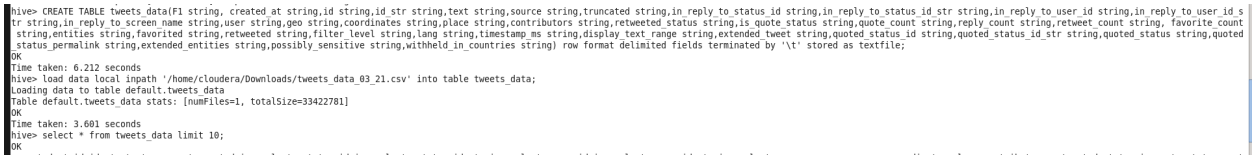


Imported file into hdfs



Created the database and database table in hive and loaded the data into hive table for General dataset:





Visualized the Output:

```
Remember when people were thinking that a student was gonna drop out from the war and make room for S…","<a href=""http://twitter.com/download/iphone"" rel=""nofollow"">Twitter for iPhone</a>",FALSE,,,,,,"{'id': 1215252128790450176, 'id_
str': '1215252128790450176', 'name': 'Lau □needs Ajax', 'screen_name': 'TOD0ROVS', 'location': 'Tahera's♥ | 22 | BLM', 'url': None, 'description': 'She/Her • I mostly rt fanarts of stuff I love and ramble about them — not spoiler free —
 Deku stan', 'translator_type': 'none', 'protected': False, 'verified': False, 'followers_count': 243, 'friends_count': 160, 'listed_count': 17, 'favourites_count': 40041, 'statuses_count': 30088, 'created_at': 'Thu Jan 09 12:41:15 +0000
 2020', 'utc_offset': None, 'time_zone': None, 'geo_enabled': False, 'lang': None, 'contributors_enabled': False, 'is_translator': False, 'profile_background_color': 'F5F8FA', 'profile_background_image_url': '', 'profile_background_image
_url_https': '', 'profile_background_tile': False, 'profile_link_color': '1DA1F2', 'profile_sidebar_border_color': 'C0DEED', 'profile_sidebar_fill_color': 'DDEEF6', 'profile_text_color': '333333', 'profile_use_background_image': True, 'p
rofile_image_url': 'http://pbs.twimg.com/profile_images/1371146374704680960/UKbLPL3__normal.jpg', 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1371146374704680960/UKbLPL3__normal.jpg', 'profile_banner_url': 'https://p
bs.twimg.com/profile_banners/1215252128790450176/1615741672', 'default_profile': True, 'default_profile_image': False, 'following': None, 'follow_request_sent': None, 'notifications': None}",,,,,"{'created_at': 'Thu Mar 18 03:32:23 +0000
 2021', 'id': 1372390398316843010, 'id_str': '1372390398316843010', 'text': '#MHA306\n#MHASpoilers\n\nRemember when people were thinking that a student was gonna drop out from the war and make ro… https://t.co/7laxi6znvG', 'source': '<a
href=""https://mobile.twitter.com"" rel=""nofollow"">Twitter Web App</a>', 'truncated': True, 'in_reply_to_status_id': None, 'in_reply_to_status_id_str': None, 'in_reply_to_user_id': None, 'in_reply_to_user_id_str': None, 'in_reply_to_sc
reen_name': None, 'user': {'id': 926200464453939200, 'id_str': '926200464453939200', 'name': 'AsarathaHS', 'screen_name': 'AsarathaHS', 'location': 'he/him', 'url': 'https://curiouscat.me/AsarathaHS', 'description': '99% of my tweets are
 about MHA.', 'translator_type': 'none', 'protected': False, 'verified': False, 'followers_count': 585, 'friends_count': 70, 'listed_count': 2, 'favourites_count': 8172, 'statuses_count': 3596, 'created_at': 'Thu Nov 02 21:32:53 +0000 20
17', 'utc_offset': None, 'time_zone': None, 'geo_enabled': False, 'lang': None, 'contributors_enabled': False, 'is_translator': False, 'profile_background_color': 'F5F8FA', 'profile_background_image_url': '', 'profile_background_image_ur
l_https': '', 'profile_background_tile': False, 'profile_link_color': '1DA1F2', 'profile_sidebar_border_color': 'C0DEED', 'profile_sidebar_fill_color': 'DDEEF6', 'profile_text_color': '333333', 'profile_use_background_image': True, 'prof
ile_image_url': 'http://pbs.twimg.com/profile_images/1372791666000556042/md5wt4sD_normal.jpg', 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1372791666000556042/md5wt4sD_normal.jpg', 'profile_banner_url': 'https://pbs.
twimg.com/profile_banners/926200464453939200/1615161438', 'default_profile': True, 'default_profile_image': False, 'following': None, 'follow_request_sent': None, 'notifications': None}, 'geo': None, 'coordinates': None, 'place': None, '
contributors': None, 'is_quote_status': False, 'extended_tweet': {'full_text': ""#MHA306\n#MHASpoilers\n\nRemember when people were thinking that a student was gonna drop out from the war and make room for Shinso?\n\ndamn y'all were righ
t it just wasnt who we expected LMFAOOO"", 'display_text_range': [0, 190], 'entities': {'hashtags': [{'text': 'MHA306', 'indices': [0, 7]}, {'text': 'MHASpoilers', 'indices': [8, 20]}], 'urls': [], 'user_mentions': [], 'symbols': []}}, '
quote_count': 171, 'reply_count': 44, 'retweet_count': 640, 'favorite_count': 4754, 'entities': {'hashtags': [{'text': 'MHA306', 'indices': [0, 7]}, {'text': 'MHASpoilers', 'indices': [8, 20]}], 'urls': [{'url': 'https://t.co/7laxi6znvG'
, 'expanded_url': 'https://twitter.com/i/web/status/1372390398316843010', 'display_url': 'twitter.com/i/web/status/1…', 'indices': [117, 140]}], 'user_mentions': [], 'symbols': []}, 'favorited': False, 'retweeted': False, 'filter_level':
 'low', 'lang': 'en'}",FALSE,0,0,0,0,"{'hashtags': [{'text': 'MHA306', 'indices': [16, 23]}, {'text': 'MHASpoilers', 'indices': [24, 36]}], 'urls': [], 'user_mentions': [{'screen_name': 'AsarathaHS', 'name': 'AsarathaHS', 'id': 926200464
453939200, 'id_str': '926200464453939200', 'indices': [3, 14]}], 'symbols': []}",FALSE,FALSE,low,en,1.61634E+12,,,,,,,,,        NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULLN
ULL     NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL
1,Sun Mar 21 16:20:26 +0000 2021,1.37367E+18,1.37367E+18,"@cephalus en I stand against all forms of racism, ableism, homophobia, transphobia, Islamophobia, hate; and pedophi… https://t.co/htKEN7BOsV","<a href=""http://twitter.com/downloa
```


Queries:

1. Viewing the number of tweets based on h1b:

```
hive> select count(*) from h1b where text like "%h1b%" or text like "%H1b%";
Query ID = cloudera_20210507105959_03858f06-a7f2-4773-8a73-1fb44698bf1e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1620363455197_0001, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1620363455197_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1620363455197_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-07 10:59:47,277 Stage-1 map = 0%,  reduce = 0%
2021-05-07 10:59:56,262 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.43 se
c
2021-05-07 11:00:03,736 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.48
sec
MapReduce Total cumulative CPU time: 2 seconds 480 msec
Ended Job = job_1620363455197_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.48 sec   HDFS Read: 601090
HDFS Write: 4 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 480 msec
OK
118
Time taken: 31.248 seconds, Fetched: 1 row(s)
hive> Display all possibilites? (y or n)
```

2. Viewing the number of tweets based on F1visa:

```
hive> select count(*) from F1visa where text like "%F1visa%" or text like "%f1visa%";
Query ID = cloudera_20210507111515_7b5c7459-e238-4552-85f6-952bd4707329
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1620363455197_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1620363455197_
0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1620363455197_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-07 11:16:05,010 Stage-1 map = 0%,  reduce = 0%
2021-05-07 11:16:12,470 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2021-05-07 11:16:20,941 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.28 sec
MapReduce Total cumulative CPU time: 2 seconds 280 msec
Ended Job = job_1620363455197_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.28 sec   HDFS Read: 50001 HDFS Write: 2 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 280 msec
OK
1
```

3. Details of tweets regarding international students that have a greater favourites count:

```
hive> select text,tweet_id,favorites as c from internationalstudents order by c desc limit 10;
Query ID = cloudera_20210507123434_51964827-c2e1-4675-9f31-a4975c2d3dc5
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1620363455197_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1620363455197_
0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1620363455197_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-07 12:34:46,487 Stage-1 map = 0%,  reduce = 0%
2021-05-07 12:34:51,835 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.79 sec
2021-05-07 12:34:59,303 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.6 sec
MapReduce Total cumulative CPU time: 1 seconds 600 msec
Ended Job = job_1620363455197_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.6 sec   HDFS Read: 8843 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 600 msec
OK
Time taken: 20.836 seconds
```

4. Details of tweets regarding international students using Order by:

```
hive> select text,tweet_id,name,retweets from internationalstudents order by retweets desc limit 10;
Query ID = cloudera_20210507124040_d459b52d-0048-4a6b-84be-48e2cbd93cbd
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1620363455197_0005, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1620363455197_
0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1620363455197_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-07 12:40:29,893 Stage-1 map = 0%,  reduce = 0%
2021-05-07 12:40:36,205 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.85 sec
2021-05-07 12:40:43,573 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.68 sec
MapReduce Total cumulative CPU time: 1 seconds 680 msec
Ended Job = job_1620363455197_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.68 sec   HDFS Read: 9144 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 680 msec
OK
Time taken: 20.888 seconds
```

5. Viewing the details of tweet id from h1b:

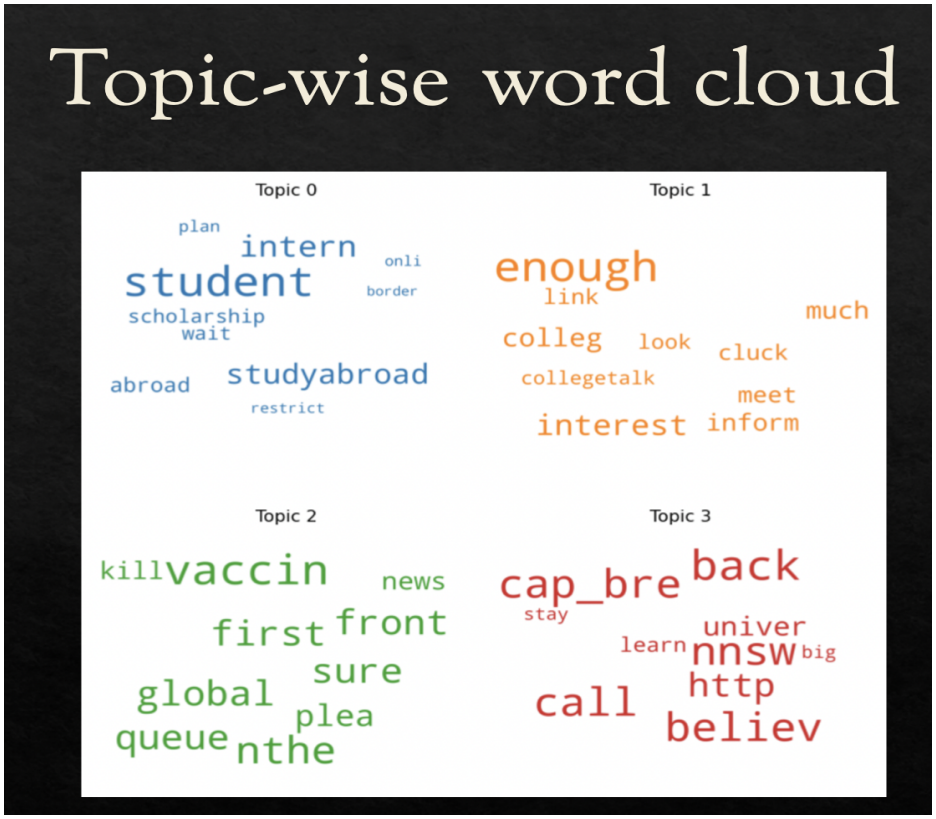
```
hive> select tweet_id,name from h1b cluster by tweet_id limit 10;
Query ID = cloudera_20210507124646_5291b3a4-f21b-4743-8c79-47dca1b7fb66
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1620363455197_0007, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1620363455197_
0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1620363455197_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-07 12:46:47,667 Stage-1 map = 0%,  reduce = 0%
2021-05-07 12:46:55,061 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.26 sec
2021-05-07 12:47:02,456 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.3 sec
MapReduce Total cumulative CPU time: 2 seconds 300 msec
Ended Job = job_1620363455197_0007
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
```

6. Details of retweets with limit from h1b:

```
hive> select retweets,tweet_type from h1b order by retweets limit 3;
Query ID = cloudera_20210507125858_dd7cc2c3-3136-41f2-9be4-f72f0d7ff820
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1620363455197_0017, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1620363455197_
0017/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1620363455197_0017
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-07 12:58:28,063 Stage-1 map = 0%,  reduce = 0%
2021-05-07 12:58:34,387 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.07 sec
2021-05-07 12:58:42,719 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.25 sec
MapReduce Total cumulative CPU time: 2 seconds 250 msec
Ended Job = job_1620363455197_0017
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.25 sec   HDFS Read: 599211 HDFS Write: 18 SUCCESS
```

8. RESULTS EVALUATION:

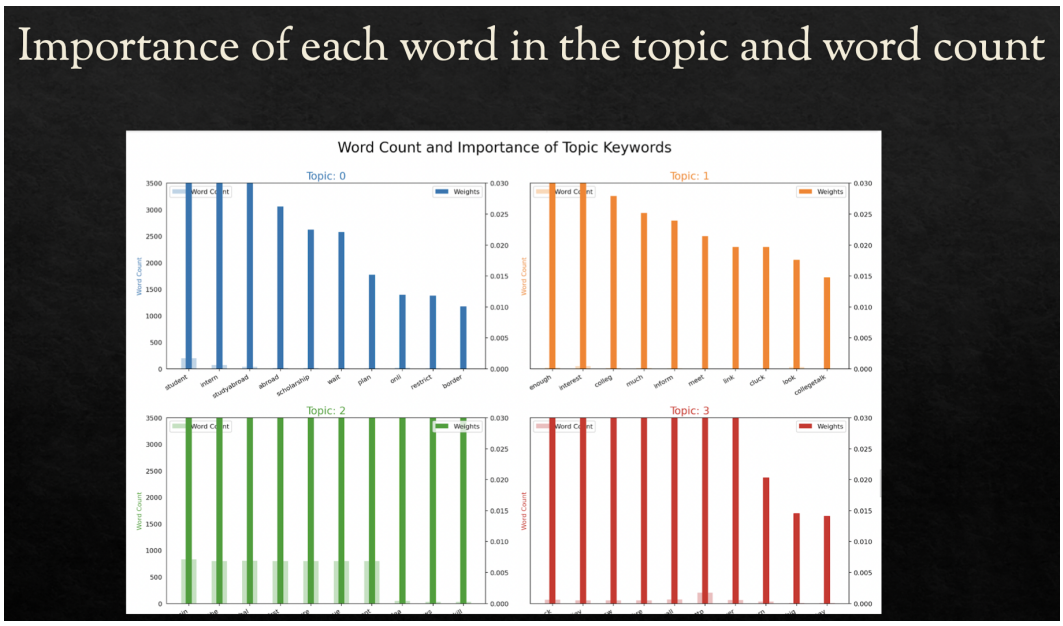
a.



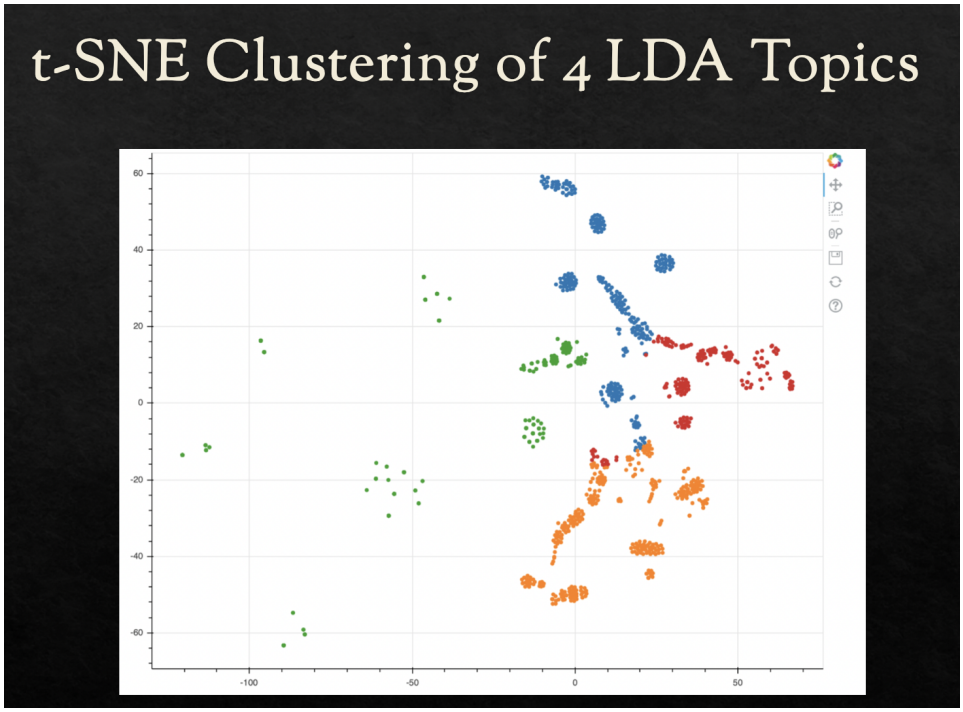
b.

```
text = tidy_tweets[0]

# Create and generate a word cloud image:
wordcloud = WordCloud().generate(text)

# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
```

c.


Importance of each word in the topic and word count
Word Count and Importance of Topic Keywords
Topic: 0
Topic: 1
Topic: 2
Topic: 3
Word Count
Weights
student
intern
studyabroad
abroad
scholarship
wait
plan
drill
restrict
border
enough
interest
colleg
much
inform
meet
link
cluck
look
collegetalk

d.


t-SNE Clustering of 4 LDA Topics
60
40
20
0
-20
-40
-60
-100
-50
0
50

e.

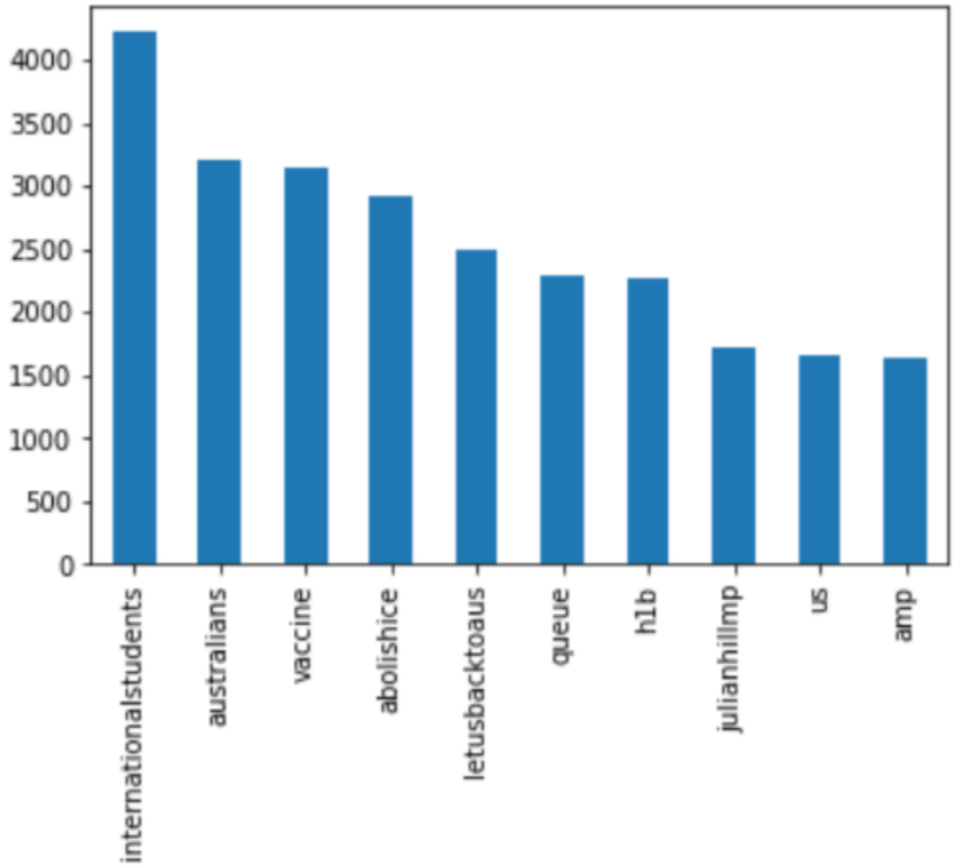

4000
3500
3000
2500
2000
1500
1000
500
0
internationalstudents
australians
vaccine
abolishice
letusbacktoaus
queue
h1b
julianhillmp
us
amp

Fig. Plotting top words from Text data

f.

```
hero.wordcloud(df["Text"].pipe(hero.clean))
```

Fig. Word Cloud generation using Texthero

g.

```
#generate scatter plot
hero.scatterplot(df, 'pca', color = 'kmeans', hover_data=['Text '] )
```

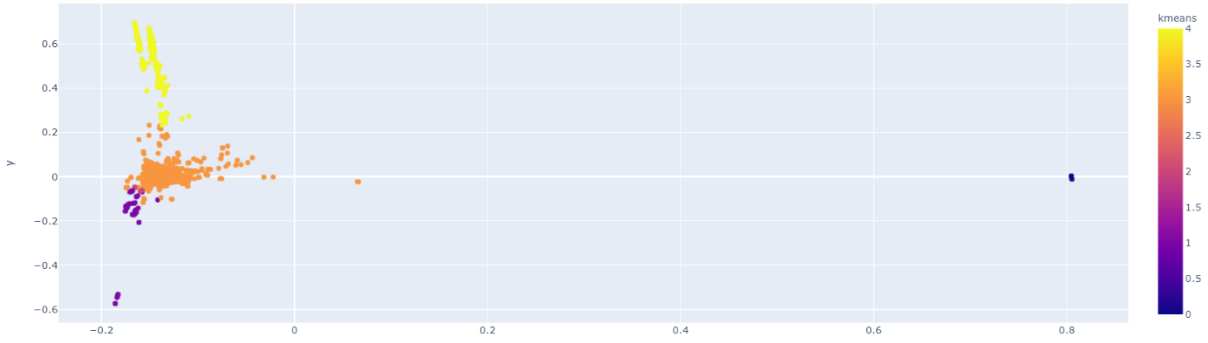


Fig. Scatter Plot of PCA and k-means cluster analysis

9. CONCLUSION:

Done the data extraction using Twitter API for different hashtags and implemented them in various Big Data tools like Hive, Spark and PySpark. The visualization is done using PySpark in the form of bar graphs, pie-charts, and scatter plots. The sentiment analysis of the tweets is

performed using pyspark and some visualization is done using different tools like TSNE and tableau.

10. FUTURE WORK:

- Implementing the better Machine Learning Algorithms.
- Working on more data sources such as Instagram, Facebook, and YouTube Data as well.
- Considering the project deployment in Docker.
- Better use of the big data tools by targeting on a specific predefined and focused dataset.
- A tool can be created using ML tools and analysis which on the basis of streaming tweets can tell the sentiments and prospective suggestions about a specific situation or crisis, which would help people calm down and be productive in the right track instead of panicking.

11. PROJECT MANAGEMENT:

Description:

All the tasks that were lined up for the final project have been completed successfully in a timely fashion. We were able to implement different big data tools to analyse the data about international students and how they were impacted by the COVID situation and crisis.

Work Completed :

We have completed the extraction of dataset from twitter using twitter API, completed the tweets preprocessing part, sentiment analysis of those tweets. The data which was extracted was converted from csv to json format. We have also worked on Hive, Scala and pySpark to perform different analyzing techniques. Loaded dataset into solr to analyze various aspects from the data. Visualized the results using Tableau and Seaborn.

Responsibility (Task, Person):

- Twitter data streaming (Tanvi, Mounika)
- Data processing (Tanvi, Amar)
- Conversion of csv to json (Saikumar)
- Spark streaming using Scala (Saikumar)
- Solr execution and queries (Amar)
- pySpark queries and plotting (Amar, Tanvi)
- Word cloud, PCA and k-means clustering using Texthero (Amar)

- Hive implementation (Mounika)
- Visualization in Seaborn (Amar)
- Word Cloud using pyspark (Tanvi)
- Topic modelling to find dominant terms (Tanvi)
- TSNE clustering & Visualization in Tableau (Tanvi)
- Sentiment analysis and trend analysis in pyspark(Tanvi)

Contributions (members/percentage):

- Tanvi Jain (25%)
- Amarnadha Reddy Ankireddypalli (25%)
- Naga Mounika Thotakura (25%)
- Saikumar Reddy Papagari (25%)

Issues/Concerns:

- The data we gathered from Twitter in the beginning is unprocessed. So, we faced challenges in combining all the extracted .csv files into one single file.
- Faced challenges in converting .csv file to .json file due to large data.

12. STORY TELLING:

	WHO	WHAT	WHEN	WHERE	WHY	HOW
CHAPTER 1	The international students who are studying in the United States of America (or overseas)	The problem mainly affected them in terms of finance, job hunting, allowance to work with any company at any time or lack of getting accepted by any company due to limited work authorization and stress because of	There is no specific time/place of problem, it is the general situational challenges that international students come across.	These challenges can be faced by international students in a foreign land, where they are a part of the immigrant community and do not have access to the perks that the citizens of that country get, which	Due to a set of immigration rules and less availability of jobs/ work authorization in desired fields.	There is n certain that chances happening these bec these kin incidents seen experience many internatio students the time.

		lack of stability in life.		makes life a bit harder for even the very basic requirements of living.		
CHAPTER 2	The international students who are studying in the United States of America (or overseas). The dataset is extracted using different hashtags to get data related to 3 different scenarios as follows: Generic International Student Data (#F1visa #intlstudents #internationalstudents #studyinUSA) Immigration rules change for F1 Visa during COVID in 2020 (#AbolishICE) Jobs for international students during covid (H1B sponsorship) (#H1B, #h1bjobs)	Yes, the data set records the targeted events, activities, behaviors, etc. in Assignment 1. This is fundamentally about the variables. It records the username, the location, the tweets which tell us about what the users really think about the specific event that happened.	The events take place on how people reacted to the challenges faced by International students during COVID such as immigration rules for F1visa, jobs for international students during covid(H1B sponsorship).	These challenges can be faced by any International Student in a foreign land, where they are a part of immigrant community and do not have access to the perks that the citizens of that country get, which makes life a bit harder for even the very basic requirements of living.	Due to the increase in the intake of international students over a period.	It happ during C as a r there wa massive Unavailabi of jobs graduates many have their jobs to pand immigratio rules ch for F1 during COVID.
CHAPTER 3 (Scientist and AI)	Students from other countries studying in the United States of America or Overseas	The preprocessed data can be fitted to any of the ML and Deep Learning models. The text is used for finding dominant topic through topic modeling and	The extracted data can be used to find the accuracy and runtime performance of best fitting ML model.	It was part of the Big Data Programming course offered in University of Missouri-Kansas City	Most of the unsupervised learning models work best for our extracted data.	The scientist use data i way that research requires based on tool. No in this me is static; data or sources m introduced

		sentiment analysis. All the other features of the data about tweet and users are used to visualize the important stats using pyspark.				required further rev
CHAPTER 4 (Users)	International students	The visualisation shows the analysis of twitter data on international students through which we analysed various challenges faced by them due to covid.	This project can be used to understand the present crisis due to covid such as travel ban, student's in-take, and visa issues.	This project is visualised using Seaborn and Tableau and can be deployed in Docker as well.	The visualised data can be used to comprehend the travel bans, college admissions, and visa issues due to covid crisis.	The internatio students utilize project wo plan overseas in a bette considerin the challe that may by oncourse their ove journey.
CHAPTER 5 (Society)	In this project the society of international students/immigrants in the USA are targeted during the covid crisis. The data scientists who have worked on this project are a part of the international student society who thought it was necessary to address this situation and analyse it through using big data tools. Data Scientists - Tanvi	Yes definitely there is a social and cultural impact through this project, since it will help to identify how people were effected and there was a drastic change in the nature of tweets for this situation.	The social impact takes place every now and then, though the data or the timeline of the data that this project is covering is for the year 2020 and 2021 specifically and comparing it	The cultural impact takes place in the setting of the USA and the immigration rules change due to the result of recession.	It is important for the upcoming international students and immigrants to have the knowledge about how trend and immigration rules change owing to the change in the world economy and	Specially the part c communit which has part percentag the socie play, ML sl have sp tools predefine that suggest guide p based on sentiment society t

	Jain, Saikumar Reddy Papagari, Amarnadha Reddy Ankireddypalli, Thotakura Naga Mounika.	There are no privacy concerns because the data that is analysed is publicly available on the social media platform tike twitter.	with the other previous years when COVID did not hit the world.		infrastructure. They should have a clear view and idea about how their life can be affected in a positive or a negative way as a result of recession.	specific scenario, that inste getting informatic from w resources, can be us a resource look forwa when situation occurs.

13. REFERENCES:

[1] https://towardsdatascience.com/try-texthero-the-absolute-simplest-way-to-clean-and-analyze-text-in-pandas-6db86ed14272

[2] https://docs.cloudera.com/documentation/enterprise/6/6.3/topics/cm_mc_solr_service.html

[3] https://spark.apache.org/sql/

[4] https://www.tutorialspoint.com/spark_sql/spark_sql_dataframes.html

[5] https://docs.cloudera.com/runtime/7.2.7/search-managing/topics/search-updating-the-schema-in-a-solr-collection.html

[6] https://github.com/dsuarez993/bigdata-realtime-twitter-analysis

[7] https://github.com/6vedant/TwitterAnalyticsHadoop

[8] https://www.toptal.com/apache/apache-spark-streaming-twitter

[9] https://hadoop.apache.org/