CS167 Twitter Analysis Report
Project Group D1

Minsoo Kim 862238343 - Task 1
Rajeev Thundiyil 862077977 - Task 2
Christian Boroff 862181900 - Task 3

Introduction

In our project we performed analysis on twitter data, and used Apache Spark (Beast) to do so. We used this big-data system because it is useful for working with dataframes and performing analysis on them, as it has sql and machine learning support, both of which we used throughout the project.

Within the project, we cleaned the data, assigned a topic to each tweet, and then built a machine learning model that predicted the topic of a tweet.

Task 1 - Minsoo Kim

I was tasked with the data preparation portion of the project. Specifically, I had to prepare the data for analysis later in the project. To give a brief summary, I first took the given data set and kept only the some of the attributes and created a new output file with it. Then I extracted the top 20 keywords from the newly created dataset.

In order to first load the data I did the following:

```
// Load the given input file using the json format.
    val tweetsDF = sparkSession.read.format("json")
      .option("sep", "\t")
      .option("inferSchema", "true")
      .option("header", "true")
      .load(inputFile)
```

Then, in order to keep only the relevant attributes, I ran the following:

```
// Keep only the following attributes {id, text, entities.hashtags.txt,
user.description, retweet_count, reply_count, and quoted_status_id}
    val clean_tweets_df: DataFrame = tweetsDF.selectExpr("id", "text",
"transform(entities.hashtags, x -> x.text) AS hashtags", "user.description AS
user_description", "retweet_count", "reply_count", "quoted_status_id")
```

The relevant attributes mentioned above included the following,

- {id, text, entities.hashtags.txt, user.description, retweet_count, reply_count, and quoted_status_id}

Tweets_1k.json

Running the `printSchema()` function, I got the following schema

```
root
 |-- id: long (nullable = true)
 |-- text: string (nullable = true)
 |-- hashtags: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- user_description: string (nullable = true)
 |-- retweet_count: long (nullable = true)
 |-- reply_count: long (nullable = true)
 |-- quoted_status_id: long (nullable = true)
```

Then running the `show()` on the `tweets_clean` dataset, I got the following

```
+------------------+--------------------+--------------------+--------------------+-------------+-----------+------------------+
|                id|                text|            hashtags|    user_description|retweet_count|reply_count|  quoted_status_id|
+------------------+--------------------+--------------------+--------------------+-------------+-----------+------------------+
|921633443934433280|       saya tahu dia ter...|                  []|I'm not here anymore|            0|          0|              null|
|921633444219596800|ふみとぉおおお👧👧👧💕💕...|                  []|野球ガールの頭やってます🙋甲子園...|            0|          0|921338657436459008|
|              null|0          |0           |... اللهم إجعلني خير|                  []|...اللهم إني أعوذ بك       |921633444131680256|
|921633443976568838|       find a boo &amp; ...|                  []|• TxSU • 🤘 HTX...|            0|          0|              null|
|921633443708096512|       He referred to hi...|                  []|   constantly grumpy|            0|          0|              null|
|921633444400115713|       @BarRefaeli my fi...|                  []|            transduce|            0|          0|              null|
|921633444437770241|       Idk bout y'all bu...|                  []|Sierra High Schoo...|            0|          0|              null|
|921633444223954944|       😂😂😂😂gwym mane...|                  []| David killed Goli...|            0|          0|921633138190815232|
|921633444773400576| I'm at 北京餃子 in 仙台...|                  []|五橋→元茶畑68→東京某学校/元I...|            0|          0|              null|
|921633444953772032|       I'm at G Tower in...|                  []|Zpace for share w...|            0|          0|              null|
|921633445326942211|       beach https://t.c...|                  []|they hurt u and t...|            0|          0|899837211901820928|
|921633445360623617|       @christophelicat ...|[Taquet, TeamSilv...|10 Années de dans...|            0|          0|              null|
|921633444693663745|       #NaturalCure #Hea...|[NaturalCure, Hea...|#HealthTips News ...|            0|          0|              null|
|921633445788258304|       @ShirazHassan my ...|                  []|Cheif operating o...|            0|          0|              null|
|921633445499097088|       So true! https://...|                  []|Area Manager for ...|            0|          0|921491726585487360|
|921633445729538048|       @Auisirilak1 ตอนไ...|                  []|                null|            0|          0|              null|
|921633444915904512|       เป็นคนเชื่อง่ายหัว...|                []|รักสันโดษ เจ็บแค่...|            0|          0|              null|
|921633445301964800|       5mentarios https:...|                  []|☠️:michellerosello|            0|          0|921501191401623552|
|921633446518079488|       Mas okay talaga k...|                  []|  hopeless romantic|            0|          0|              null|
|921633444496576512|       Paesaggiando http...|                  []|tremeran le fogli...|            0|          0|              null|
+------------------+--------------------+--------------------+--------------------+-------------+-----------+------------------+
only showing top 20 rows
```

This is how I wrote the new file for the clean dataset

```
// Store the output in a new JSON file named tweets_clean
    clean_tweets_df.write.mode(SaveMode.Overwrite).json("tweets_clean")
```

Then, I ran a top-k SQL query to select the top 20 most frequent hashtags. The below shows the SQL and how I collected the result in an array of keywords.

```
// run a top-k SQL query to select the top 20 most frequent hashtags as follows.
    clean_tweets_df.createOrReplaceTempView("clean_tweets")
    val frequent_hashtags = sparkSession.sql(
      """
      SELECT hashtag, count(*) as count
      FROM (
        SELECT explode(hashtags) as hashtag
        FROM clean_tweets
      ) t
      GROUP BY hashtag
```

```
       ORDER BY count DESC
       LIMIT 20
     """)

     val keywords: Array[String] = frequent_hashtags.select("hashtag").rdd.map(row =>
row.getString(0)).collect()
```

The result of the above code snippet was the following:

hashtag	count
ALDUBxEBLoveis	7
FurkanPalalı	5
no309	5
LalOn	5
sbhawks	3
DoktorlarDenkliki...	3
Benimisteğim	3
3 احتاج_بالوقت_هذا	
art	2
CNIextravaganza2017	2
love	2
happy	2
2 السعودية	
nowplaying	2
beautiful	2
türkiye	1
vegalta	1
KittyLive	1
鋼	1
tossademar	1

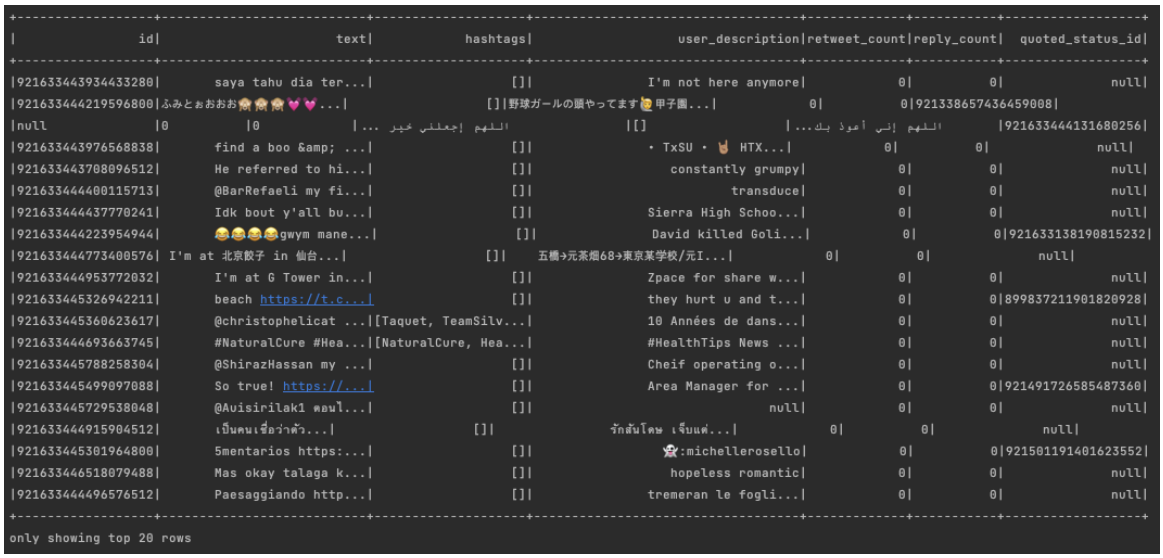
Tweets_10k.json

Following the same steps but for `Tweets_10k.json` gave me the following results

Running the `printSchema()` function on `tweets_clean` for 10k dataset, I got the following schema

```
root
 |-- id: long (nullable = true)
 |-- text: string (nullable = true)
 |-- hashtags: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- user_description: string (nullable = true)
 |-- retweet_count: long (nullable = true)
 |-- reply_count: long (nullable = true)
 |-- quoted_status_id: long (nullable = true)
```

Then running the `show()` on the `tweets_clean` for the 10k dataset, I got the following



id	text	hashtags	user_description	retweet_count	reply_count	quoted_status_id
921633443934433280	saya tahu dia ter...	[]	I'm not here anymore	0	0	null
921633444219596800	ふみとぉおおお👧👧👧💕💕...	[]	野球ガールの扉やってます🙋甲子園...	0	0	921338657436459008
null	0	0	اللهم اجعلني خير ...	[]	اللهم إني أعوذ بك...	921633444131680256
921633443976568838	find a boo & ...	[]	• TxSU • 🤘 HTX...	0	0	null
921633443708096512	He referred to hi...	[]	constantly grumpy	0	0	null
921633444400115713	@BarRefaeli my fi...	[]	transduce	0	0	null
921633444437770241	Idk bout y'all bu...	[]	Sierra High Schoo...	0	0	null
921633444223954944	😂😂😂😂gwym mane...	[]	David killed Goli...	0	0	921633138190815232
921633444773400576	I'm at 北京餃子 in 仙台...	[]	五橋→元茶畑68→東京某学校/元I...	0	0	null
921633444953772032	I'm at G Tower in...	[]	Zpace for share w...	0	0	null
921633445326942211	beach https://t.c...	[]	they hurt u and t...	0	0	899837211901820928
921633445360623617	@christophelicat ...	[Taquet, TeamSilv...	10 Années de dans...	0	0	null
921633444693663745	#NaturalCure #Hea...	[NaturalCure, Hea...	#HealthTips News ...	0	0	null
921633445788258304	@ShirazHassan my ...	[]	Cheif operating o...	0	0	null
921633445499097088	So true! https://...	[]	Area Manager for ...	0	0	921491726585487360
921633445729538048	@Avisirilak1 ตอนไ...	[]	null	0	0	null
921633444915904512	เป็นคนเชื่อว่าตัว...	[]	รักสันโดษ เจ็บแค่...	0	0	null
921633445301964800	5mentarios https:...	[]	🕸:michellerosello	0	0	921501191401623552
921633446518079488	Mas okay talaga k...	[]	hopeless romantic	0	0	null
921633444496576512	Paesaggiando http...	[]	tremeran le fogli...	0	0	null

only showing top 20 rows

Finally, the top 20 hashtags for the 10k dataset was the following

hashtag	count
ALDUBxEBLoveis	84
FurkanPalalı	51
no309	51
LalOn	51
chien	30
job	28
Hiring	22
sbhawks	16
Top3Apps	16
perdu	15
trouvé	15
CareerArc	14
Job	12
trumprussia	12
trndnl	12
Jobs	11
ShowtimeLetsCelebr8	9
hiring	9
impeachtrumppence	9
music	8

Task 2 - Rajeev Thundiyil

My task was to show the topic of the tweet by doing an array intersection of the most frequent hashtags from task 1 with the hashtags within the tweet itself. If it's a match, a new column is created with the topic shown, found through the intersection.

```
clean_tweets_df.createOrReplaceTempView( viewName = "tweets_clean")
//convert keywords to an array separated with , so it can be used for array intersect in a query
val topics: String = "'"+ keywords.mkString("','") + "'"

//dataframe
```

Taking the tweets_clean dataset from task 1, I use createorReplaceTempView in order to create a local temporary view. This will be used later in the query.

Taking the keywords string array from task 1, I parse through it and put in commas between each space. This is necessary so it can be parsed and properly used for the array_intersect function within the query below.

```
val topics_df: DataFrame = sparkSession.sql(
  sqlText = s"""
  SELECT id, text,element_at(t1.tweet_topic,1), user_description, retweet_count, reply_count, quoted_status_id
  FROM ( SELECT *, array_intersect(hashtags, array($topics)) AS tweet_topic FROM tweets_clean) AS t1 WHERE size(tweet_topic) > 0;
   """)


//write to json
topics_df.write.json( path = "tweets_topic.json")
topics_df.show()
val t4 = System.nanoTime()
```

The query picks up all of the data types in order, as given in the project specifications.

root
 |-- id: long (nullable = true)
 |-- text: string (nullable = true)
 |-- topic: string (nullable = true)
 |-- user_description: string (nullable = true)
 |-- retweet_count: long (nullable = true)
 |-- reply_count: long (nullable = true)
 |-- quoted_status_id: long (nullable = true)

I create a table called t1, and within it a new column named tweet_topic is created, in which the hashtags on the line get intersected with the frequent hashtags from task 1. If it gets matched, then it gets included in the datatype, tweet_topic. I use element_at in order to get only the first frequent hashtag matched with the current tweet being checked.

Results shown below

```
Operations on file 'Tweets_1k.json' took 7.5779234 seconds
+------------------+--------------------------------+-------------------------+--------------------------------+-------------+-----------+------------------+
|                id|                            text|element_at(tweet_topic, 1)|              user_description|retweet_count|reply_count|  quoted_status_id|
+------------------+--------------------------------+-------------------------+--------------------------------+-------------+-----------+------------------+
|921633446644080641|              #negramarooffici...|                     love|           Negramanteinside_...|            0|          0|              null|
|921633445045866497|              #CNIextravaganza2...|       CNIextravaganza2017|           Hebat Produknya H...|            0|          0|              null|
|921633449882128384|              #DoktorlarDenklik...|      DoktorlarDenkliki...|                       emin ben|            0|          0|              null|
|921633451773648896|              Na miss ko mag tw...|            ALDUBxEBLoveis|           Resilient. Object...|            0|          0|              null|
|921633452289642497|              #FurkanPalalı Değ...|              FurkanPalalı|                            null|            0|          0|              null|
|921633451714920448|              #KittyLive penuh ...|                 KittyLive|                        Semangat|            0|          0|              null|
|              null|              0|         0|           |...انا مقدر علي قلبي            |احتاج_بالوقت_هذا|          |...احتاج_بالوقت_هذا#              |921633453715738624|
|921633455766728704|              Don`t run from yo...|            ALDUBxEBLoveis|           BEHIND THE THICK ...|            0|          0|              null|
|921633464276996097|              Künefe Ocağı Rezi...|                   türkiye|           Bym Isı Rezistans...|            0|          0|              null|
|921633465921028096|ブルーマウンテンさんやん！\n追加...|                 sbhawks|野球垢です！ホークスファンです！|            0|          0|              null|
|921633468634632194|              Pastinya dong, ka...|       CNIextravaganza2017|           BLOGGER | BUZZER ...|            0|          0|921627555261702144|
|921633470413262848|              #FurkanPalalı Değ...|              FurkanPalalı|                            null|            0|          0|              null|
|921633476964646912|              Start by doing wh...|            ALDUBxEBLoveis|           BEHIND THE THICK ...|            0|          0|              null|
|921633490482946048|              Stay by Rihanna F...|                nowplaying|           A live stream of ...|            0|          0|              null|
|921633493569998848|              #FurkanPalalı Değ...|              FurkanPalalı|                            null|            0|          0|              null|
|921633494530392064|              Saturday let's go...|                nowplaying|               Student of Life!|            0|          0|              null|
|              null|              0|         0|           |...يومية - سياسي            |السعودية|          |...التعليق مع رجال أ              |921633495511896065|
|921633502453518336|              There are many th...|            ALDUBxEBLoveis|           BEHIND THE THICK ...|            0|          0|              null|
|921633507558002688|              #happy #Saturday ...|                    happy|           Pastry chef, pers...|            0|          0|              null|
|921633511752257536|              #FurkanPalalı Değ...|              FurkanPalalı|                            null|            0|          0|              null|
+------------------+--------------------------------+-------------------------+--------------------------------+-------------+-----------+------------------+
```

After running it for the tweets_10k.json file, there were **269 results** that were generated and shown to have the topic of the tweet shown.

Task3 - Christian Boroff

My task was to build a machine learning model that assigns a topic for each tweet using the classified tweets from the previous part. I used Apache Spark for this part, as it is very useful for machine learning, and building machine learning models. I made a pipeline for the model that includes a tokenizer, a HashingTF, a StringIndexer, and finally we used a LogisticRegression classifier to determine the topic of each tweet. My code for doing this can be seen below.

```
val tokenizer = new Tokenizer().setInputCol("text").setOutputCol("words")

val hashingTF = new HashingTF().setInputCol("words").setOutputCol("features")

val stringIndexer = new StringIndexer().setInputCol("element_at(tweet_topic, 1)").setOutputCol("label").setHandleInvalid("skip")

val logisticRegression = new LogisticRegression()

val pipeline = new Pipeline().setStages(Array(tokenizer, hashingTF, stringIndexer, logisticRegression))

val Array(trainingData, testData) = topics_df.randomSplit(Array(0.7, 0.3))

val logisticModel = pipeline.fit(trainingData)

val predictions = logisticModel.transform(testData)

predictions.select( col= "id",  cols= "text", "element_at(tweet_topic, 1)", "user_description", "label", "prediction").show( numRows= 50)
```

As you can see above I used a 70/30 split for the trainingData and testData respectively. I then trained the model on the first 70% of the data and tested it on the remaining testData, which resulted in the following output.

id	text	element_at(tweet_topic, 1)	user_description	label	prediction
921633452289642497	#FurkanPalalı Değ...	FurkanPalalı	null	1.0	1.0
921633470413262848	#FurkanPalalı Değ...	FurkanPalalı	null	1.0	1.0
921633493569998848	#FurkanPalalı Değ...	FurkanPalalı	null	1.0	1.0
921633502453518336	There are many th...	ALDUBxEBLoveis	BEHIND THE THICK ...	0.0	0.0
921633525505310722	あー落ちない #sbhawks	sbhawks	野球垢です！ホークスファンです！！ 6.0	6.0	
921633536628551680	Everyone's deserv...	ShowtimeLetsCelebr8	love + trust	8.0	0.0
921633595747213312	If you're looking...	job	Staffing and Recr...	3.0	3.0
921633595801792512	Can you recommend...	job	Quintiles and IMS...	3.0	3.0
921633599660658688	Sometimes people ...	ALDUBxEBLoveis	BEHIND THE THICK ...	0.0	0.0
921633653129486336	@kuriring01 @Bern...	ALDUBxEBLoveis	Businesswoman, lo...	0.0	0.0
921633666656284672	#FurkanPalalı Değ...	FurkanPalalı	null	1.0	1.0
921633666656284672	#FurkanPalalı Değ...	FurkanPalalı	null	1.0	1.0
921633683789856768	Can you recommend...	job	Follow us for job...	3.0	3.0
921633684708454400	See our latest #R...	job	Join a team that ...	3.0	3.0
921633691469787136	#trumprussia #imp...	trumprussia	#NYC Registered N...	11.0	11.0
921633698423955456	😂🤣\nla ako masa...	ALDUBxEBLoveis	CMSRN	0.0	0.0
921633705260548096	Ganda nila 😍 #Sh...	ShowtimeLetsCelebr8	Hotelier and a 👨...	8.0	8.0

I then needed to compute the precision and recall of my result on the 10k dataset, which I did with the following code.

```
// Compute the number of true positives, false positives, and false negatives for each class
val tp = (0 ≤ to ≤ 10).map(c => predictions.filter(col( colName = "label") === c && col( colName = "prediction") === c).count()).sum
val fp = (0 ≤ to ≤ 10).map(c => predictions.filter(col( colName = "label") =!= c && col( colName = "prediction") === c).count()).sum
val fn = (0 ≤ to ≤ 10).map(c => predictions.filter(col( colName = "label") === c && col( colName = "prediction") =!= c).count()).sum

// Compute overall precision and recall
val overallPrecision = tp.toDouble / (tp + fp)
val overallRecall = tp.toDouble / (tp + fn)

println(s"Overall Precision: $overallPrecision, Overall Recall: $overallRecall")
```

This code found the total number of true positives, false positives, and false negatives for all of our classes, and then used them to compute the overall precision and recall. The precision and recall we obtained for the 10k dataset were the same, both coming out to **0.94382**. This means that 94.38% of the time, my model correctly identified true positive cases, while about 5.62% of the time, it incorrectly identified positive cases as negative.