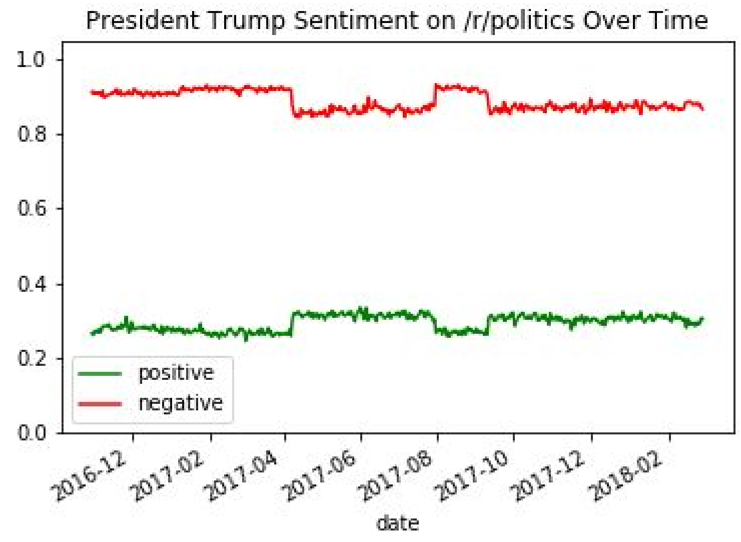
CS143 Project 2 Report

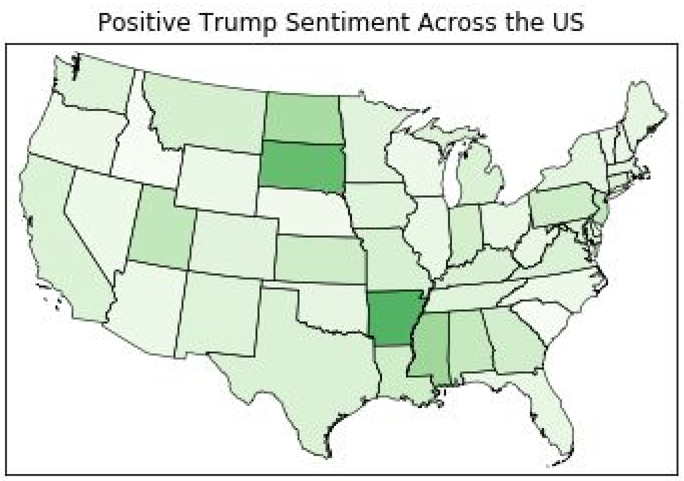
Xingyi Chen -205032924
Yucong Wang - 305036163
Jingjing Zhang - 904946929
Hui Wang - 205036597

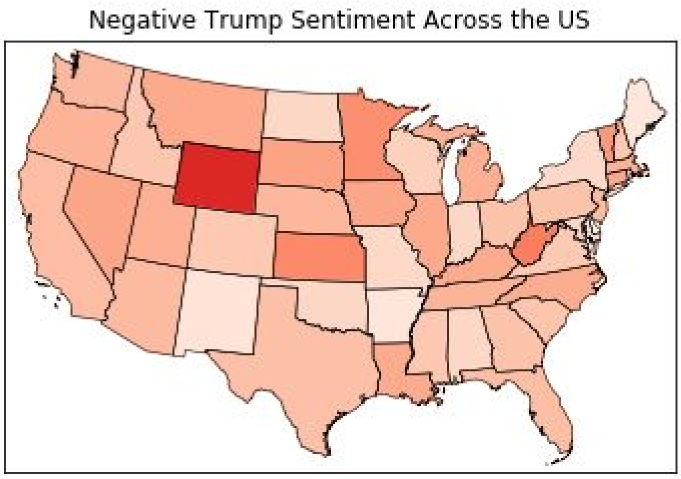
Part 1: Analysis of the result

1. **Create a time series plot (by day) of positive and negative sentiment. This plot should contain two lines, one for positive and one for negative. It must have data as an X axis and the percentage of comments classified as each sentiment on the Y axis.**

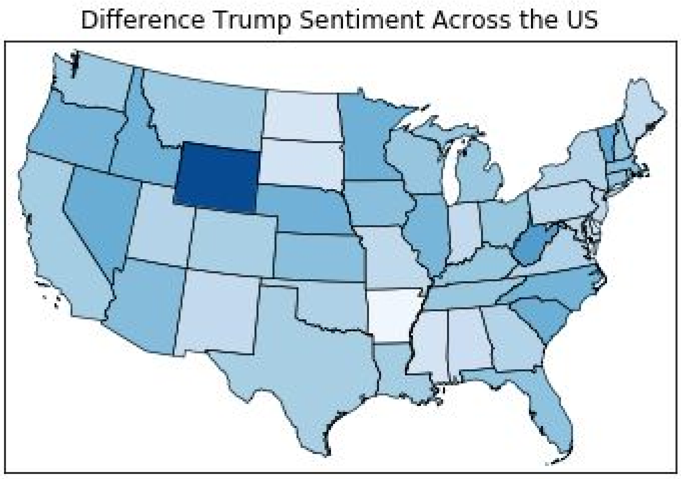


2. **Create 2 maps of the United States: one for positive sentiment and one for negative sentiment. Color the states by the percentage.**

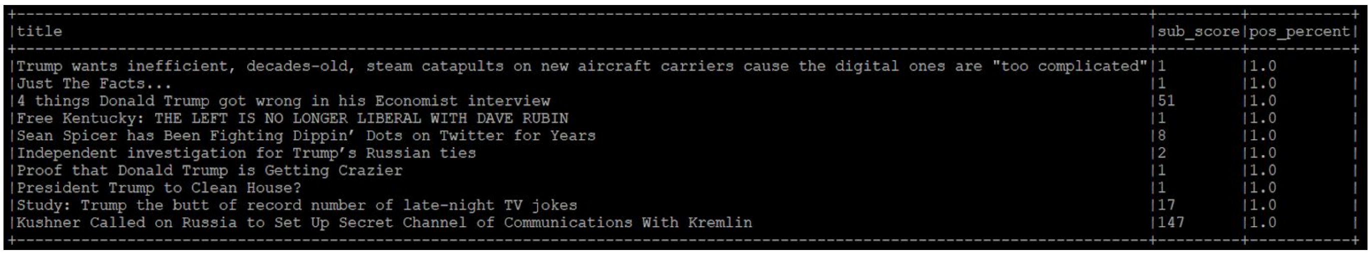




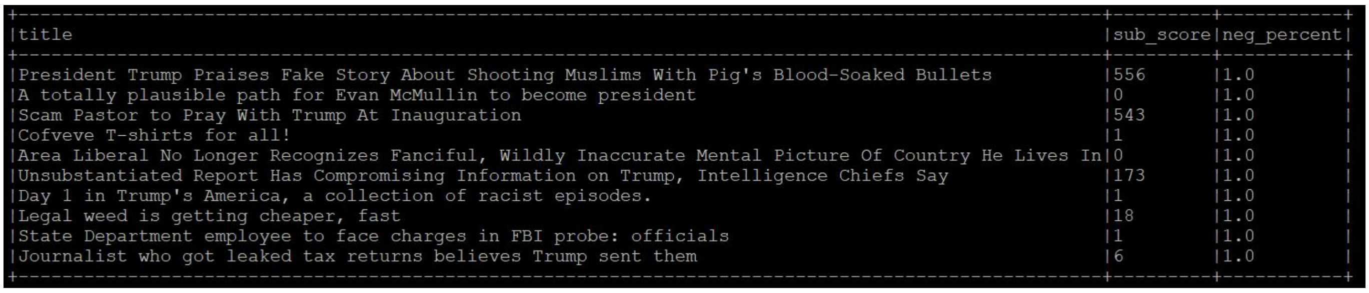
3. **Create a third map of the United States that computes the difference: %Positive - %Negative.**



4. **Give a list of the top 10 positive stories (have the highest percentage of positive comments) and the top 10 negative stories (have the highest percentage of negative comments). This is easier to do in Spark.**

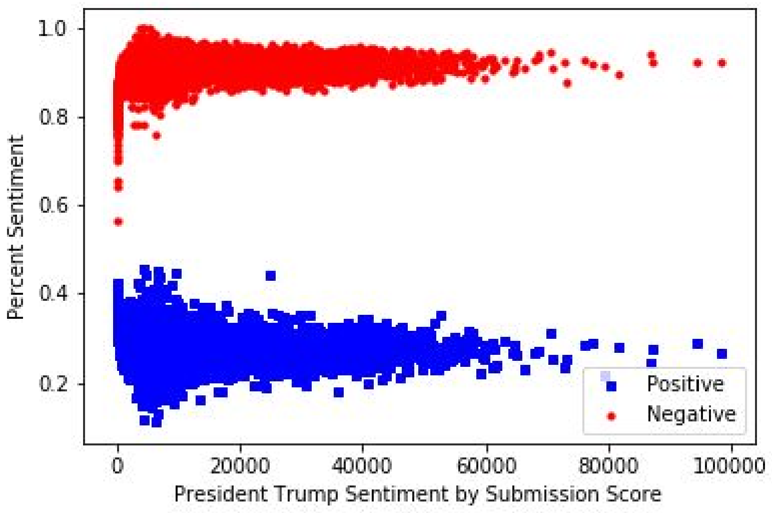


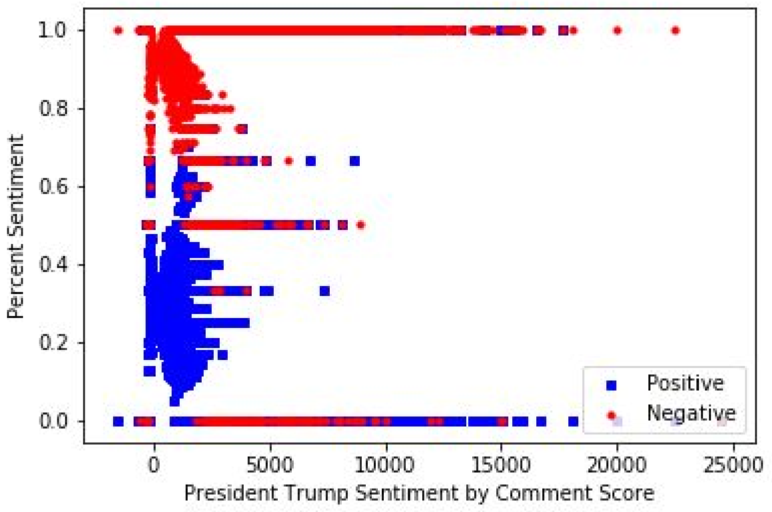
```
+--------------------------------------------------------------------------------------------------------------------+---------+-----------+
|title                                                                                                               |sub_score|pos_percent|
+--------------------------------------------------------------------------------------------------------------------+---------+-----------+
|Trump wants inefficient, decades-old, steam catapults on new aircraft carriers cause the digital ones are "too complicated"|1        |1.0        |
|Just The Facts...                                                                                                   |1        |1.0        |
|4 things Donald Trump got wrong in his Economist interview                                                          |51       |1.0        |
|Free Kentucky: THE LEFT IS NO LONGER LIBERAL WITH DAVE RUBIN                                                        |1        |1.0        |
|Sean Spicer has Been Fighting Dippin' Dots on Twitter for Years                                                     |8        |1.0        |
|Independent investigation for Trump's Russian ties                                                                  |2        |1.0        |
|Proof that Donald Trump is Getting Crazier                                                                          |1        |1.0        |
|President Trump to Clean House?                                                                                     |1        |1.0        |
|Study: Trump the butt of record number of late-night TV jokes                                                       |17       |1.0        |
|Kushner Called on Russia to Set Up Secret Channel of Communications With Kremlin                                    |147      |1.0        |
+--------------------------------------------------------------------------------------------------------------------+---------+-----------+
```



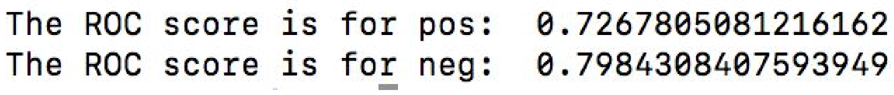
```
+--------------------------------------------------------------------------------------------------------+---------+-----------+
|title                                                                                                   |sub_score|neg_percent|
+--------------------------------------------------------------------------------------------------------+---------+-----------+
|President Trump Praises Fake Story About Shooting Muslims With Pig's Blood-Soaked Bullets               |556      |1.0        |
|A totally plausible path for Evan McMullin to become president                                          |0        |1.0        |
|Scam Pastor to Pray With Trump At Inauguration                                                          |543      |1.0        |
|Cofveve T-shirts for all!                                                                               |1        |1.0        |
|Area Liberal No Longer Recognizes Fanciful, Wildly Inaccurate Mental Picture Of Country He Lives In|0        |1.0        |
|Unsubstantiated Report Has Compromising Information on Trump, Intelligence Chiefs Say                   |173      |1.0        |
|Day 1 in Trump's America, a collection of racist episodes.                                              |1        |1.0        |
|Legal weed is getting cheaper, fast                                                                     |18       |1.0        |
|State Department employee to face charges in FBI probe: officials                                       |1        |1.0        |
|Journalist who got leaked tax returns believes Trump sent them                                          |6        |1.0        |
+--------------------------------------------------------------------------------------------------------+---------+-----------+
```

5. **Create TWO scatterplots where the X axis is the submission score, and a second where the X axis is the comment score, and the Y access is the percentage positive and negative. Use two different colors for positive and negative. This allows us to determine if submission score, or comment score can be used as a feature.**

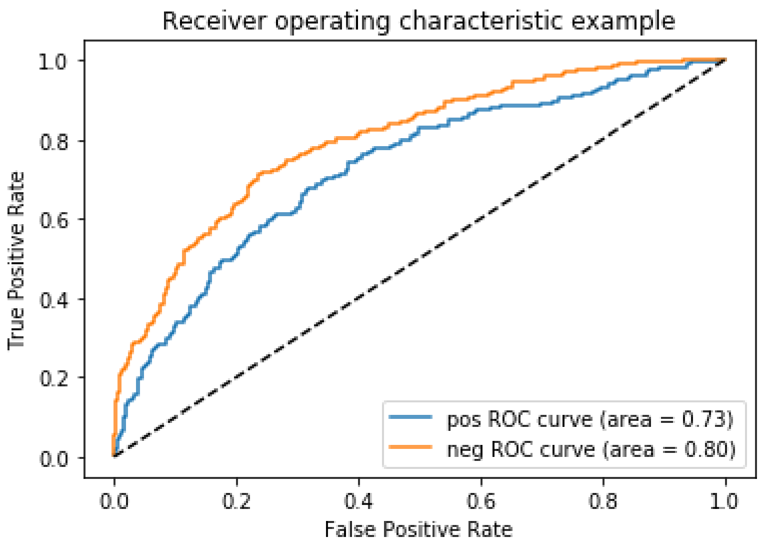




6. **Extra Credit: Produce the ROC curves for YOUR classifiers and compute the Area Under the Curve for each one, which is a measure of accuracy.**



```
The ROC score is for pos:  0.7267805081216162
The ROC score is for neg:  0.7984308407593949
```



The AUC is fairly high so the classifier performs really well on the test set.

7. **Write a paragraph summarizing your findings. What does /r/politics think about President Trump? Does this vary by state? Over time? By story/submission?**

Answer: /r/politics more tend to think negatively about Trump, since the percentage of the negative comments is much higher than positive one.

The result is apparently vary by state since there is obvious color difference among states which indicates that people in some state slightly tend to support President Trump and some state are not.

Also the result vary over time, during 2017.4 to 2017.8 and after 2017.9, people are likely to support Trump more than other times. Besides, it's reasonable that when the positive line is higher, the negative line becomes lower.

The percentage positive is always low along with the submission score, and the percentage negative is always high, so the submission might be a good feature to separate the negative and positive. On the other hand, the tend by comment score is not clear between positive and negative, so this may not be a good feature to choose.

8. **Extra credit part:**

We calculate the positive and negative percentage with title containing "trump" and make comparison with all the comments.
The positive percentage of reddits with a title including trump is 0.2975206611570248. The negative

percentage of reddits with a title including trump is 0.8394787031150668. Compared to the positive percentage and negative percentage of the whole reddits. Reddits whose title include trump is slightly more positive than the average.

Part 2: Question Answers

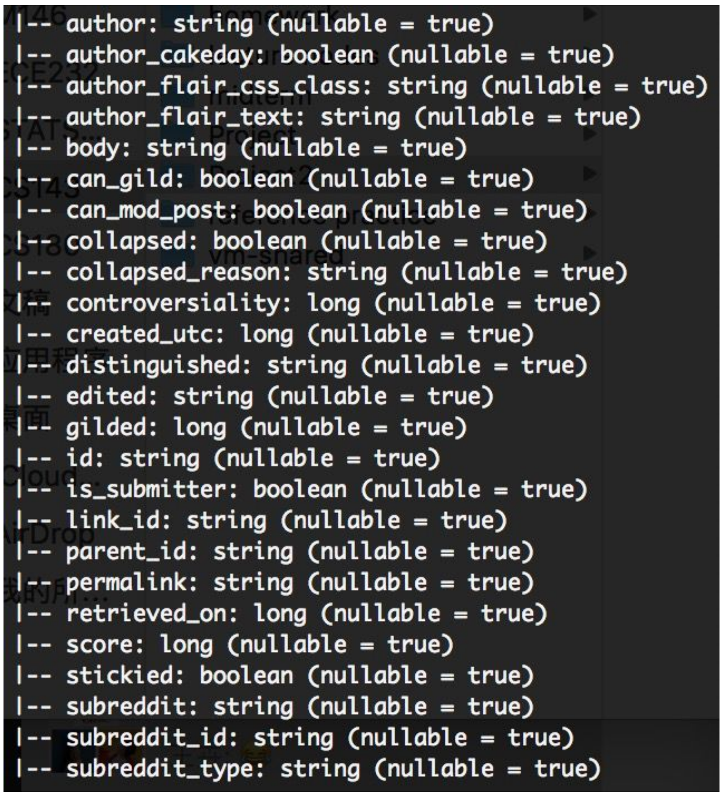
QUESTION 1: Take a look at labeled_data.csv. Write the functional dependencies implied by the data.

Answer: There are three FDs in the table:

$$\text{Input.id} \rightarrow \text{labeldem}, \text{Input.id} \rightarrow \text{labelgop}, \text{Input.id} \rightarrow \text{labeldjt}$$

QUESTION 2: Take a look at the schema for the comments dataframe. Forget BCNF and 3NF. Does the data frame look normalized? In other words, is the data frame free of redundancies that might affect insert/update integrity? If not, how would we decompose it? Why do you believe the collector of the data stored it in this way?

Answer: The schema of the dataframe is shown below:

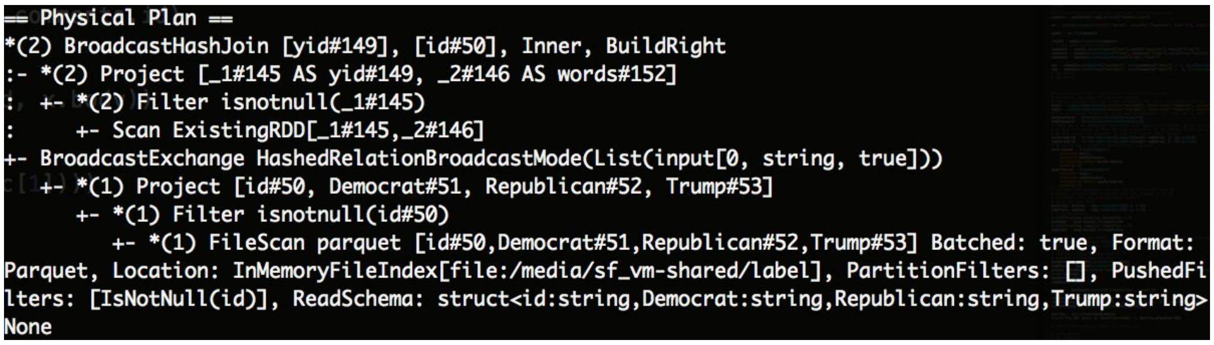


```
|-- author: string (nullable = true)
|-- author_cakeday: boolean (nullable = true)
|-- author_flair_css_class: string (nullable = true)
|-- author_flair_text: string (nullable = true)
|-- body: string (nullable = true)
|-- can_gild: boolean (nullable = true)
|-- can_mod_post: boolean (nullable = true)
|-- collapsed: boolean (nullable = true)
|-- collapsed_reason: string (nullable = true)
|-- controversiality: long (nullable = true)
|-- created_utc: long (nullable = true)
|-- distinguished: string (nullable = true)
|-- edited: string (nullable = true)
|-- gilded: long (nullable = true)
|-- id: string (nullable = true)
|-- is_submitter: boolean (nullable = true)
|-- link_id: string (nullable = true)
|-- parent_id: string (nullable = true)
|-- permalink: string (nullable = true)
|-- retrieved_on: long (nullable = true)
|-- score: long (nullable = true)
|-- stickied: boolean (nullable = true)
|-- subreddit: string (nullable = true)
|-- subreddit_id: string (nullable = true)
|-- subreddit_type: string (nullable = true)
```

The data frame is normalized, since the id can uniquely define the comment, so it's the key of the whole table. There's no redundancy in this schema, everything is decided directly by id. When we update a comment, we only need to take care of all the attributes of this comment if. When we insert of delete a comment, we need to take care of the subreddit_id and parent_id to keep the integrity of the schema. Because we do not need fast aggregation request on the data, so redundancy is not useful here, as a result having a normalized table is more space efficient then,

QUESTION 3: Pick one of the joins that you executed for this project. Rerun the join with .explain() attached to it. Include the output. What do you notice? Explain what Spark SQL is doing during the join. Which join algorithm does Spark seem to be using?

Answer: the explain of the join for the comments table and label table is shown below:



```
== Physical Plan ==
*(2) BroadcastHashJoin [yid#149], [id#50], Inner, BuildRight
:- *(2) Project [_1#145 AS yid#149, _2#146 AS words#152]
:  +- *(2) Filter isnotnull(_1#145)
:     +- Scan ExistingRDD[_1#145,_2#146]
+- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
   +- *(1) Project [id#50, Democrat#51, Republican#52, Trump#53]
      +- *(1) Filter isnotnull(id#50)
         +- *(1) FileScan parquet [id#50,Democrat#51,Republican#52,Trump#53] Batched: true, Format: 
Parquet, Location: InMemoryFileIndex[file:/media/sf_vm-shared/label], PartitionFilters: [], PushedFi
lters: [IsNotNull(id)], ReadSchema: struct<id:string,Democrat:string,Republican:string,Trump:string>
None
```

From the output we can see that Spark SQL did Broadcast Hash Join on two tables, which performs a hash inner join between two child relations. When the output RDD of this operator is being constructed, a Spark SQL job is asynchronously started to calculate the values for the broadcasted relation. This data is then placed in a Spark broadcast variable. The streamed relation is not shuffled.