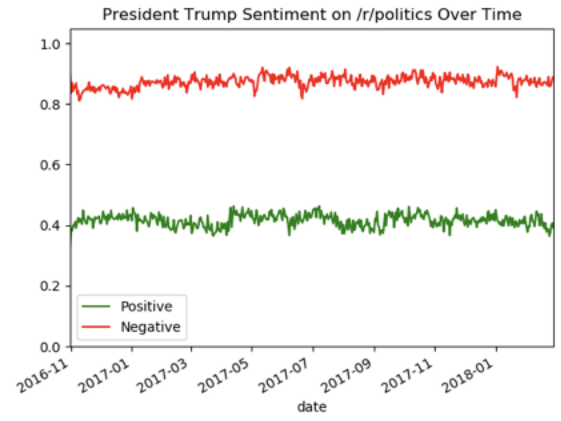
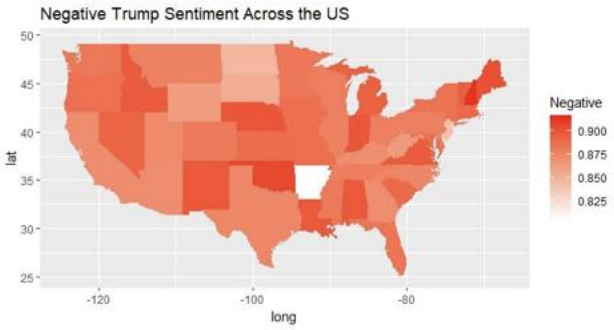
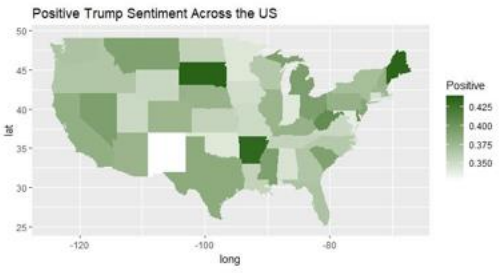
CS143 Project2 Report

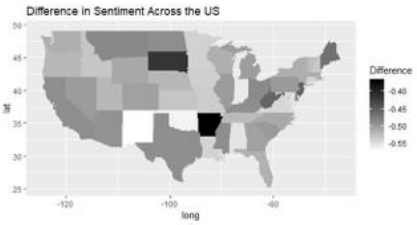
Zixuan Wang, Katherine Kang

Final Deliverable:







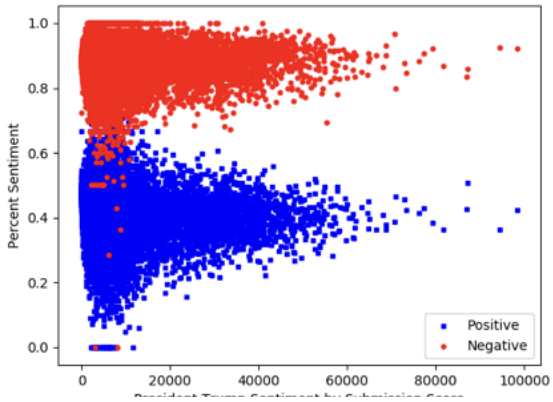


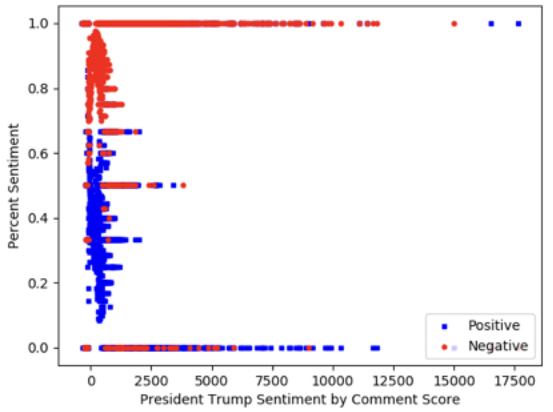
Top_Positive_Submissions

link_id	title	Positive	Negative
5e5tm4	Obama advice to daughters about election: 'Don't get into a fetal position'	1	0
5j1rz1	DNC Chair Brazile: Trump Victory 'Tainted' By Russian	1	0
5ens1q	Politifact: Truml claims he isn't literally Hitler, PANTS ON FIRE!	1	0
5cdd0t	Samantha Bee Slams White People for Voting Donald Trump	1	0.666666666666667
5fg0k7	Senate Democrats signal Sessions is in for a fight	1	1
5bxcc7	Live Election results. Doesn't look good for Trump unless he can hold onto Florida and NC.	1	1
5fgbpx	H.L. Mencken must've seen Trump coming?	1	1
5c64bg	Racism grips schools around the country in wake of Trump victory	1	0
5fywkx	The 'outrageous' 40-year-old film that predicted the future	1	1
5j95f0	'Faithless' electors rejected or forced to vote along party line for Clinton	1	0

Top_Negtive_Submissions

link_id	title	Positive	Negative
5b79cv	Woman who accused Donald Trump of child rape is dropping her lawsuit	0.8	1
5c0tmv	After Bush v. Gore, Obama, Clinton wanted Electoral College scrapped	0.333333333333333	1
5bdrtt	If you're in Philly, there are free rides via Uber and Lyft to the polls on Election Day	0	1
5ao8j6	Hillary Canceled Her Last Public Event Because The Crowd Yelled "Lock Her Up"	0.5	1
5bxyvk	Minnesota elects country's first Somali-American lawmaker	1	1
5bio8r	Top Democrats say Clinton took a real hit from Comey. But they're cautiously optimistic.	0	1
5b1263	Historic Mississippi black church burned and vandalized with 'Vote Trump' graffiti	1	1
5bm90b	Donald Trump's success reveals a frightening weakness in American democracy	0	1
5b3x4g	Donald Trump has never been closer to the presidency than he is at this moment	0.333333333333333	1
5bt8k0	Corbyn is 'The Maddest Person In The Room' - Says Bill Clinton	0	1





We are plotting our plots by using our model with a 0.2 threshold for positive and 0.25 threshold for negative. The whole dataset takes too much time and thus we sample the dataset with a percentage of 10. The plots are generated by those 10% data. From the first plot, we can see that the comments on /r/politics tend to be much more negative than positive President Trump. Our sentiment analysis makes sense here and we can conclude that on the politics thread of reddit, there are very few people supporting President Trump. After we examine the state color map file generated by our program, we find that the sentiment tends to vary across all states. From the two plots regarding submission score and comment score, respectively, we can see that the positive percentage tends to be lower than the negative percentage in both cases. Besides, the submission score clearly separates the positive from the negative more clearly than the comment score does since the negative percentages are always higher than the positive percentages using the submission score. Hence, the submission score may be a better feature for sentiment analysis than the comment score. Therefore, the sentiment analysis indicates that the politics thread on reddit is an environment where people tend to speak badly of President Trump.

Our Answers to the following questions:

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

Functional Dependencies: Input_id -> labeldem; Input_id -> labelgop; Input_id -> 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?

We think that the data frame does not look fully normalized. There exist some data that can be repeated by others, such as author and subreddit. We can further decompose the data frame by denoting some functional dependencies. For example, we could form another relation with all attributes related to the subreddit_id or author_id. We believe the collector of the data stored it in this way due to a reason that it can be useful for statistics analysis.

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?

The Spark uses Broad cast hash join algorithm. Hash joins are equijoins on large table with hash indexing and match data based on the join key. Spark uses this algorithm to perform efficient large joins and enable parallelization due to its distributed characteristics.

```
== Physical Plan ==
*(2) Project [Input_id#192, labeldem#193, labelgop#194, labeldjt#195, body#136]
+- *(2) BroadcastHashJoin [id#146], [Input_id#192], Inner, BuildRight
   :- *(2) Project [body#136, id#146]
   :  +- *(2) Filter isnotnull(id#146)
   :     +- *(2) FileScan json [body#136,id#146] Batched: false, Format: JSON, Location:
InMemoryFileIndex[file:/media/sf_vm-shared/comments-minimal.json.bz2], PartitionFilters: [],
PushedFilters: [IsNotNull(id)], ReadSchema: struct<body:string,id:string>
   +- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
      +- *(1) Project [Input_id#192, labeldem#193, labelgop#194, labeldjt#195]
         +- *(1) Filter isnotnull(Input_id#192)
            +- *(1) FileScan csv [Input_id#192,labeldem#193,labelgop#194,labeldjt#195] Batched:

false, Format: CSV, Location: InMemoryFileIndex[file:/media/sf_vm-shared/labeled_data.csv],
```

PartitionFilters: [], PushedFilters: [IsNotNull(Input_id)], ReadSchema:

struct<Input_id:string,labeldem:int,labelgop:int,labeldjt:int>

Note: We use Python to generate all the plots except the state map because we cannot fix Basemap issue. We use R provided by the professor to generate the three maps related to the states.