CS 143
Spring 2019
Ashley Wu & Cindy Cui

Project 2B Report

#TASK 2

QUESTION 1: Take a look at labeled_data.csv. Write the functional dependencies implied by the data.
Input_id -> labeled, labelgop,labeldjt

QUESTION 2: Take a look at the schema for comments. 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?
it does not look like it's normalized since subreddit_id -> subreddit. To decompose it, we would remove the subreddit column from the comments table and create another table with subreddit_id and subreddit. It is easier for users to access the data.

#TASK 10

QUESTION 1:
avg(pos) avg(neg)
0.395100219911059 0.891226708930468

Question 2:
here is the part of the solution for task 10. 2. the complete result is inside the q2.csv file
date avg(pos) avg(neg)
2017-08-11,0.31514030218933087,0.9004008633980882
2017-09-11,0.3436213991769547,0.8860082304526748
2017-01-06,0.3363770977295163,0.8654985192497532
2017-02-26,0.34297108673978066,0.8654037886340977
2017-09-28,0.3772436872528141,0.882871919683602
2017-01-27,0.34727200318598167,0.8837116686579052
2016-11-08,0.38060154944554153,0.8575117727479873
2016-12-19,0.3409720938828313,0.8231380521160598
2017-01-24,0.3534687900421014,0.8843126487278052
2017-06-29,0.3174298729321506,0.896667465835531
2017-09-29,0.36857391809468487,0.8846935811792042
2017-07-31,0.31567852437417654,0.8974967061923583
2017-02-16,0.34782608695652173,0.8989042064333687
2017-12-02,0.37171398527865407,0.868559411146162
2017-08-14,0.3562222222222222,0.8935555555555555
2017-10-23,0.3604060913705584,0.8937182741116751
2017-08-18,0.3483801295896328,0.9028077753779697
2017-04-09,0.2895238095238095,0.9052380952380953
2017-12-25,0.35779294653014787,0.878839590443686
2017-02-28,0.3573641809543483,0.8934104523858707
2018-01-23,0.354261220373172,0.8756933938477055

Question 3:
here is the results from question 3. The complete result is from q3.csv file
author_flair_text avg(pos) avg(neg)
Utah 0.304957905 0.902712816

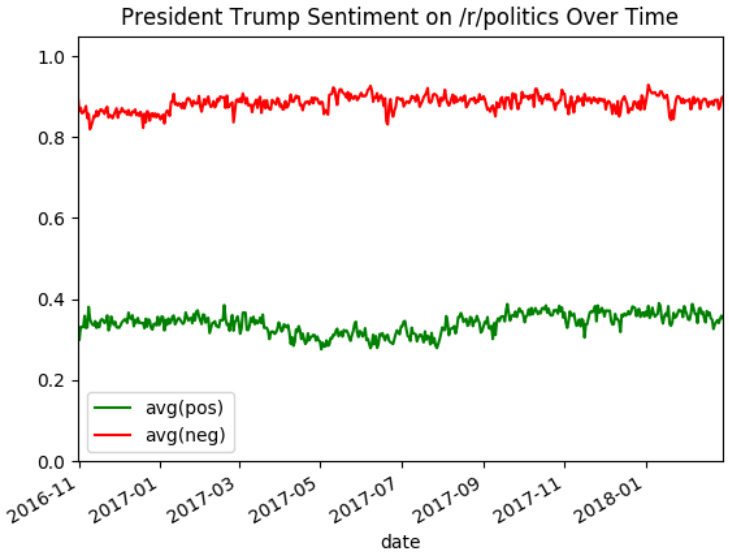
Hawaii	0.305882353	0.929411765
Minnesota	0.29430321	0.882909631
Ohio	0.32937365	0.883215057
Oregon	0.307421384	0.896981132
Arkansas	0.351791531	0.814332248
Texas	0.319204667	0.888677223
North Dakota	0.314189189	0.891891892
Pennsylvania	0.308708992	0.892140666
Connecticut	0.310173697	0.90942928
Nebraska	0.320193081	0.917940467
Vermont	0.274905422	0.89407314
Nevada	0.320777643	0.884568651
Washington	0.311721807	0.898345784
Illinois	0.310538266	0.885267798
Oklahoma	0.302978723	0.896170213
Delaware	0.35862069	0.951724138
Alaska	0.327022375	0.895008606
New Mexico	0.291588785	0.91588785
West Virginia	0.326454034	0.872420263

Question 4 :
here is the results from question 4. The complete result is from q4_c.csv and q4_s.csv file

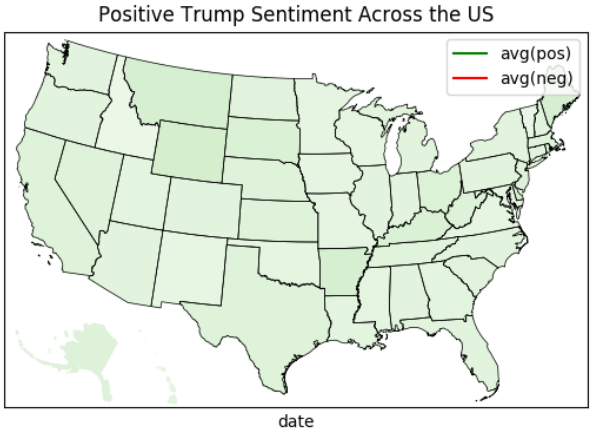
cscore	avg(pos)	avg(neg)
26	0.353249018752726	0.893807239424335
29	0.358018386108274	0.894279877425945
964	0.285714285714286	0.714285714285714
474	0.263157894736842	0.947368421052632
-91	0.375	0.875
1697	0	1
1950	1	0
2250	1	1
2040	0	1
1806	0	1
-251	1	0.5
1677	0	1
65	0.343203230148048	0.909825033647376
3806	0.333333333333333	1
191	0.31	0.94

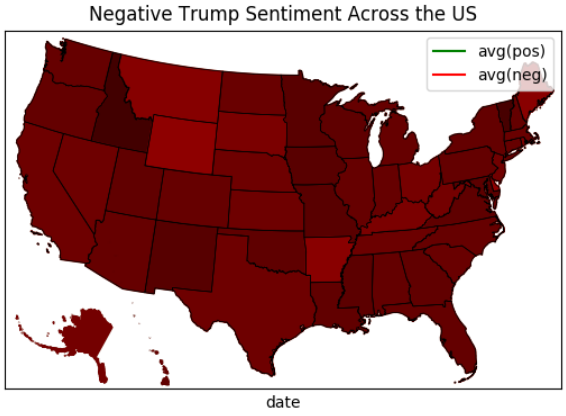
sscore	avg(pos)	avg(neg)
2214	0.32516339869281	0.866013071895425
7225	0.431372549019608	0.784313725490196
29	0.484835164835165	0.905494505494505
26	0.456817346563763	0.907019478133039
37884	0.369426751592357	0.837579617834395
2927	0.33695652173913	0.880434782608696
5385	0.323076923076923	0.907692307692308
15432	0.325925925925926	0.903703703703704
4894	0.422413793103448	0.913793103448276
2509	0.5	0.862068965517241
13723	0.448087431693989	0.918032786885246
12568	0.380952380952381	0.946428571428571
1950	0.439189189189189	0.891891891891892

#Final deliverable
1.

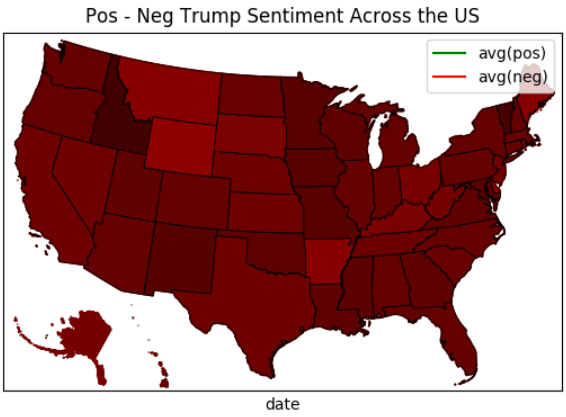


2.





3.

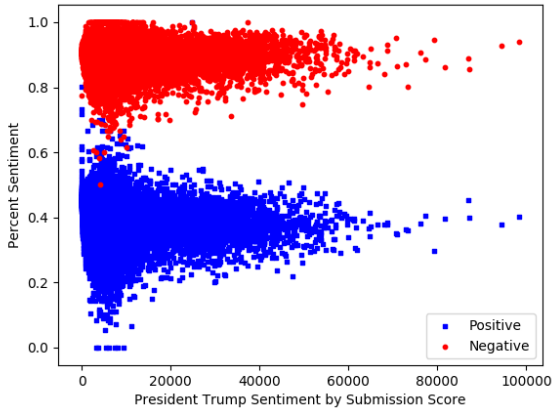


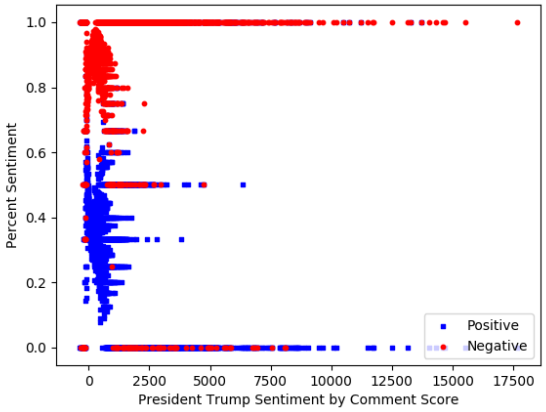
4.

title	avgpos	avgneg
Trump: 'Kelly is doing a fantastic job as chief of staff'	1.0	1.0
Trump Says He Is 'Never Going To Be Going Against' President Obama	1.0	1.0
Trump: 'We'll win' appeal of immigration ban	1.0	0.6666666666666666
Ivanka Trump's Husband, Jared Kushner, Met With Russian Officials Last Year and the FBI is Investigating Why	1.0	1.0
Americans Haven't Been This Optimistic About the Job Market in Over 30 Years	1.0	0.6666666666666666
Sad! Trump's 9 biggest unfulfilled campaign promises	1.0	1.0
UK Labour Party Leader: 'I hope our Government will condemn far-right retweets' by Trump	1.0	1.0
Davos leader booed for saying Trump has been victim of 'biased interpretations'	1.0	1.0
Trump may break another tradition – No pets in the White House	1.0	1.0
President Trump holds rally in Pensacola, Florida -- live stream	1.0	0.6666666666666666

title	avgpos	avgneg
Mueller requested DOJ hand over documents related to Comey firing: report	0.4444444444444444	1.0
Mike Huckabee reacts to Trump-Comey scandal with joke about the JFK assassination	0.5	1.0
South Dakota Republicans are about to get rid of the state's first independent ethics commission	0.5	1.0
Impeached Perjurer Bill Clinton: Trump Voters 'More Vulnerable To False Claims'	0.0	1.0
Watch a former Trump aide say incriminating things about Russia on live TV	0.0	1.0
Protesters call supporters of Trump's travel ban idiots and bigots. They're wrong.	0.3333333333333333	1.0
The alt-right is an attack on Western values. Liberals shouldn't surrender so easily.	0.5	1.0
Early Draft of Comey's Statement Called Clinton 'Grossly Negligent,' Grassley Says	0.4	1.0
Juan Williams: Trump and the new celebrity politics	1.0	1.0
Conway: Media Obsesses Over Trump's Tweets & Ignores His Policies	0.3333333333333333	1.0

5.





6.
Over time there are continuous high negative results then positive results towards trump. There is not much variable based on time. There are some variation by state, but the difference is not that obvious depending on the graph. For comment score, we see that larger comment score has stronger sentiment percentage.

Final Questions:
Question 1: same as Task2 question 1

Question 2 : same as Task2 question 2

Question 3:

```
== Physical Plan ==
*(2) BroadcastHashJoin [link_id#304], [sub_id#312], Inner, BuildRight
:- *(2) Project [id#14, body#4, created_utc#10L, substring(link_id#16, 4, 12) AS link_id#304, author_flair_text#3, score#20L AS cscore#305L]
:  +- *(2) Filter isnotnull(substring(link_id#16, 4, 12))
:     +- *(2) FileScan parquet [author_flair_text#3,body#4,created_utc#10L,id#14,link_id#16,score#20L] Batched: true, Format: Parquet, Location: InMemoryFileIndex[file:/media/sf_vm-shared/project_2a/CS_143_P2/comments.parquet], PartitionFilters: [], PushedFilters: [], ReadSchema: struct<author_flair_text:string,body:string,created_utc:bigint,id:string,link_id:string,score:big...
+- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
   +- *(1) Project [id#69 AS sub_id#312, title#106, score#92L AS sscore#313L]
      +- *(1) Filter isnotnull(id#69)
         +- *(1) FileScan parquet [id#69,score#92L,title#106] Batched: true, Format: Parquet, Location: InMemoryFileIndex[file:/media/sf_vm-shared/project_2a/CS_143_P2/submissions.parquet], PartitionFilters: [], PushedFilters: [IsNotNull(id)], ReadSchema: struct<id:string,score:bigint,title:string>
```

From the explain function we use to extract the physical layer of the join, we notice that Spark SQL could read schema, what kind of join we use, for our case, we use inner join, filter, file scan, project, parquet file in memory file index.

Spark SQL use the inner hash join algorithm to joint the submission and comments table.