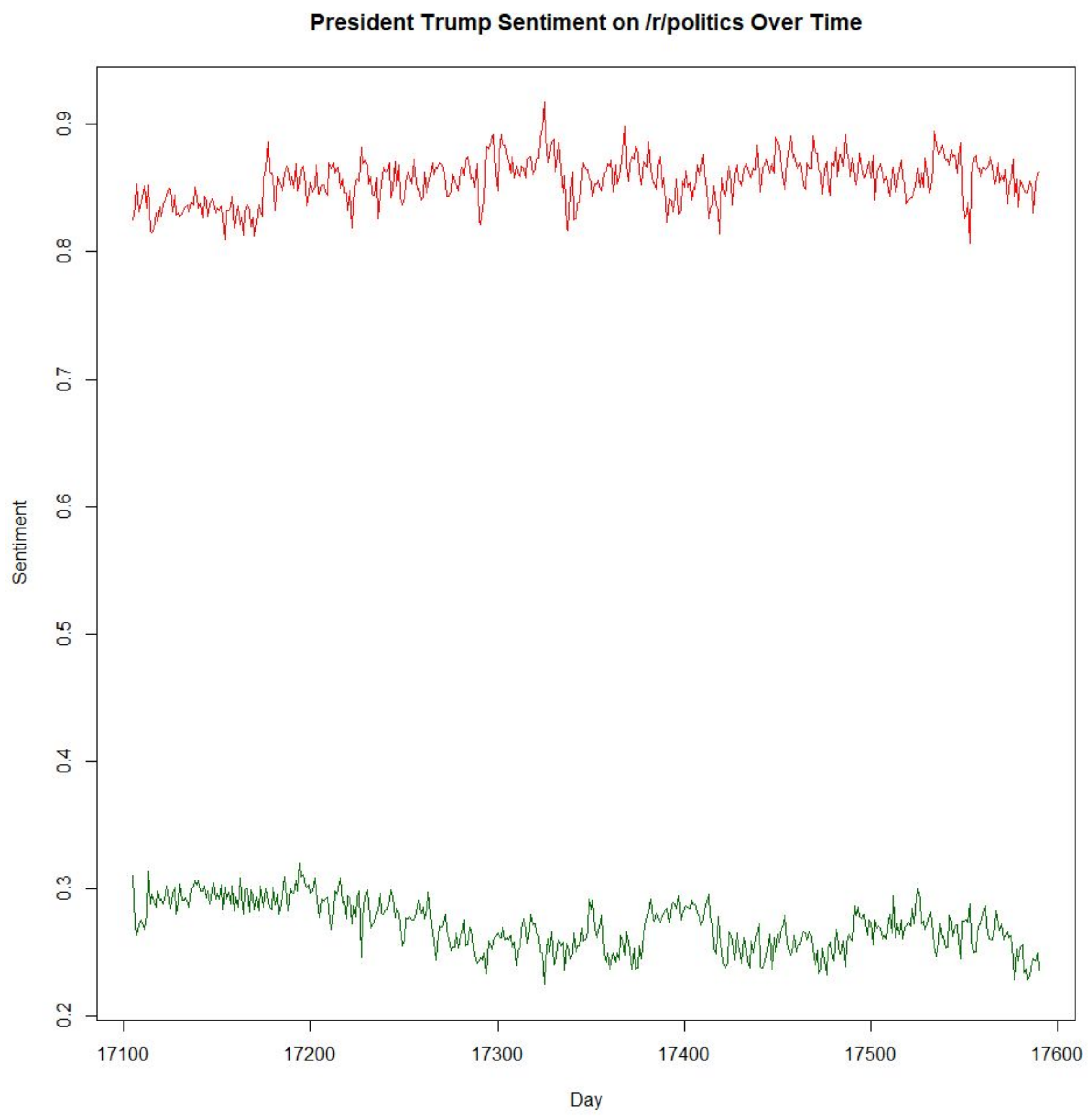
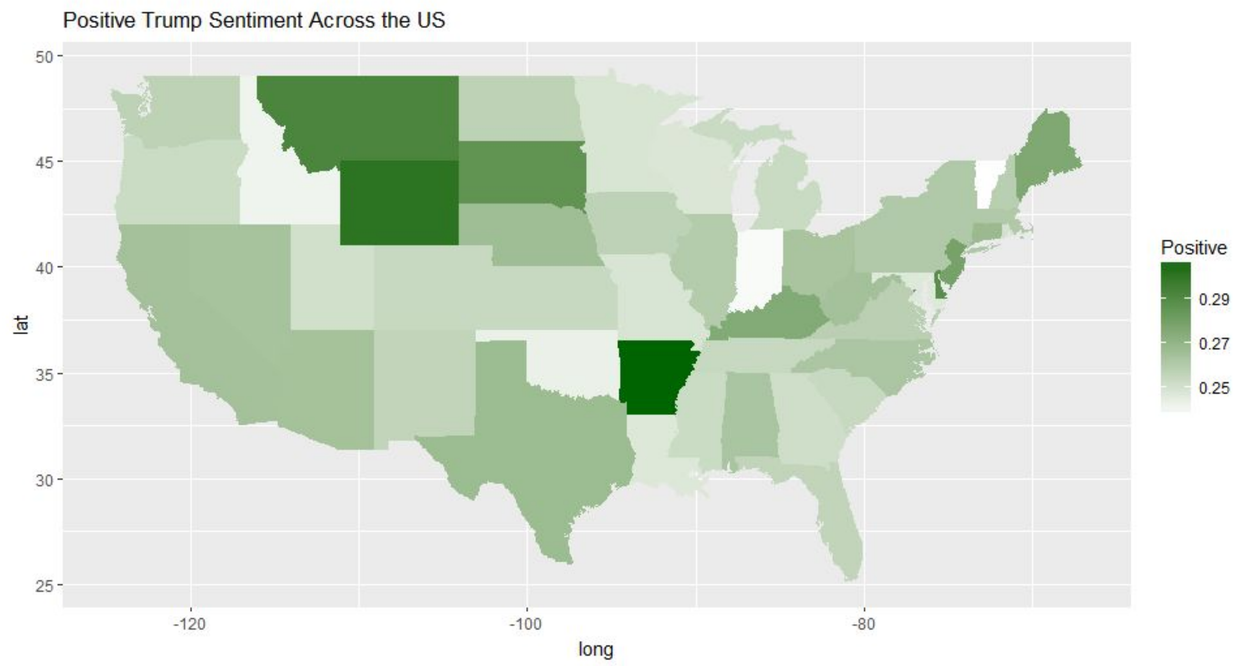
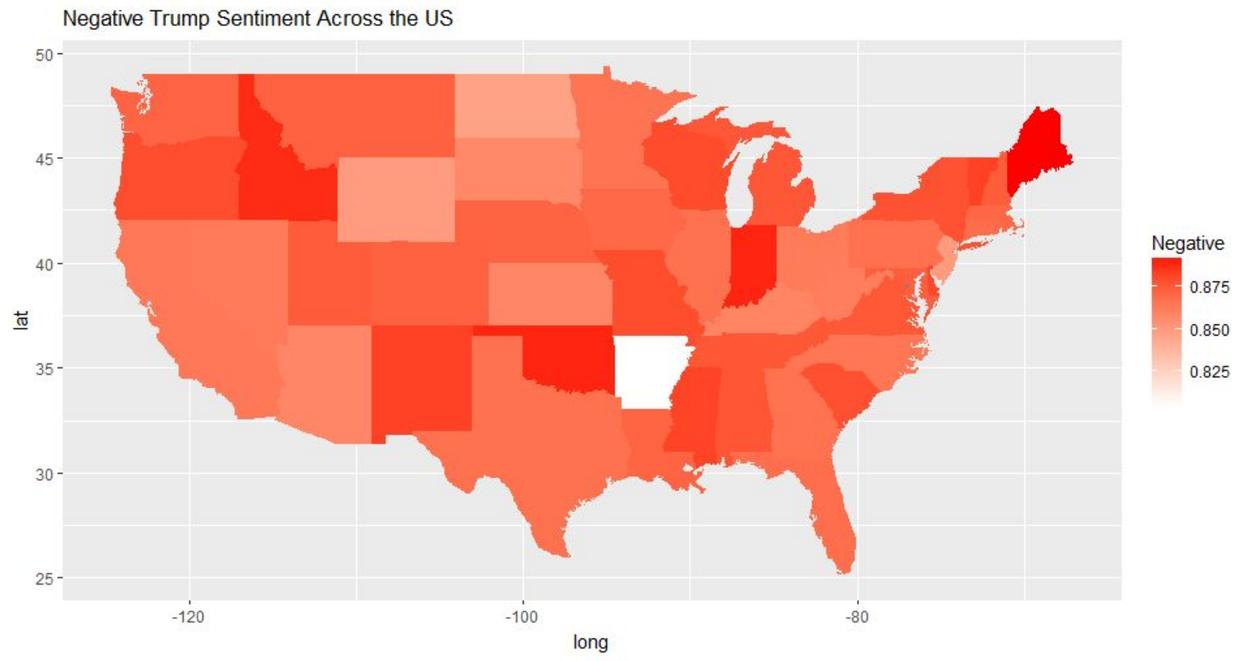
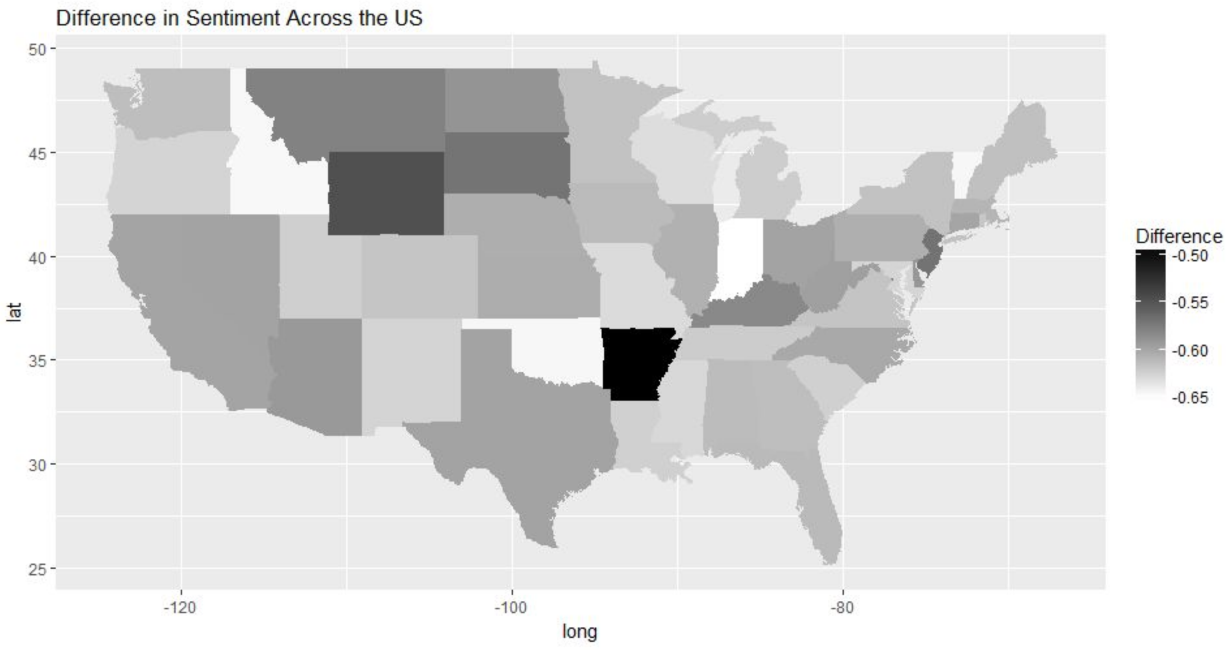
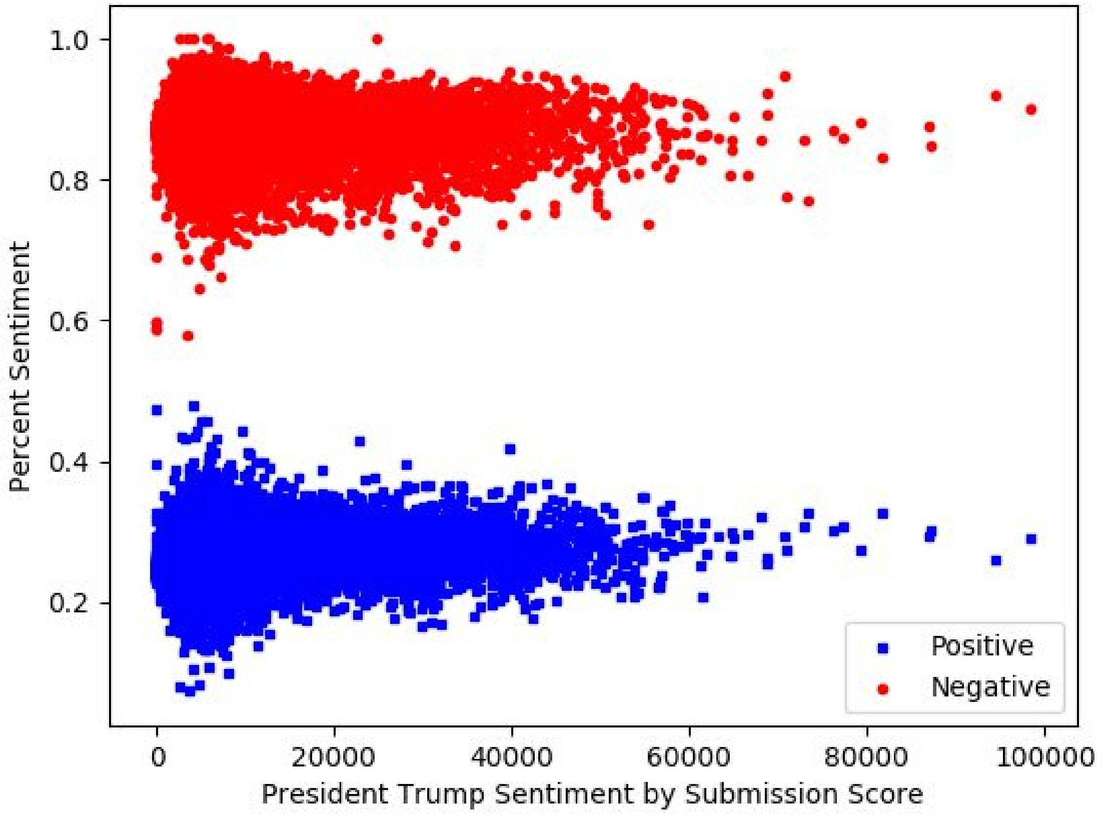
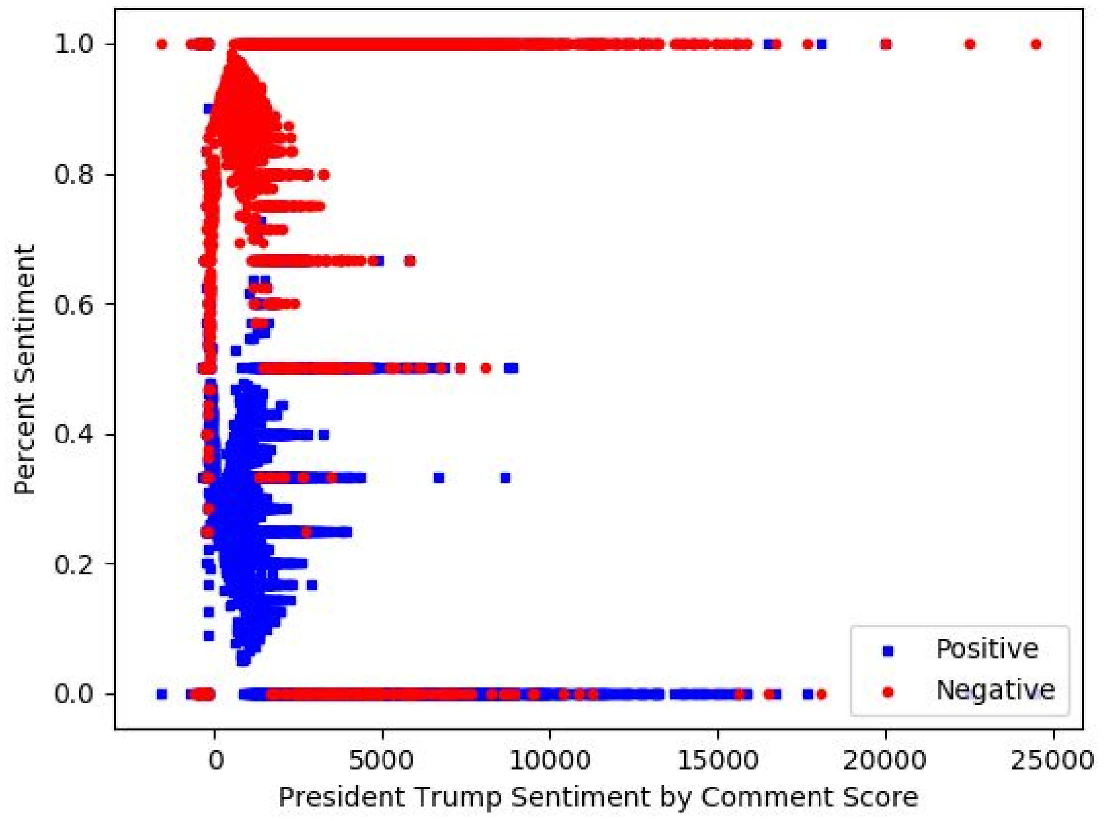

President Trump Sentiment on /r/politics Over Time
Sentiment
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Day
17100
17200
17300
17400
17500
17600


Positive Trump Sentiment Across the US
50
45
40
35
30
25
lat
-120
-100
-80
long
Positive
0.29
0.27
0.25


Negative Trump Sentiment Across the US
50
45
40
35
30
25
lat
-120
-100
-80
long
Negative
0.875
0.850
0.825


Difference in Sentiment Across the US
lat
long
50
45
40
35
30
25
-120
-100
-80
Difference
-0.50
-0.55
-0.60
-0.65


Percent Sentiment
President Trump Sentiment by Submission Score
1.0
0.8
0.6
0.4
0.2
0
20000
40000
60000
80000
100000
Positive
Negative



For reporting the posts with the most negative sentiment and most positive sentiment percentage-wise, I noticed that there were a plethora of submissions that had a 1.0 sentiment score. Because of this, I then added another column based on the number of comments and took the 10 highest scoring submissions that also had the highest number of comments, since it would be more difficult to hold a sentiment score of 1.0 across a higher number of comments. Another metric that I think could be good in creating this top-10 list would be submission score, since posts with a 1.0 score and a higher submission score would have had more eyes on the post/more user interaction, though less active engagement when compared to number of comments.

<u>Most Positive Submissions (number of comments):</u>

1. Trump pressured Park Service to find proof for his claims about inauguration crowd (34)
2. Trump Tells Israel to Hold Off on Building New Settlements (16)
3. ‘He will die in jail’: Intelligence community ready to ‘go nuclear’ on Trump, senior source says (14)
4. Trump says ‘I inherited a mess,’ blasts media and detractors at combative news conference (14)
5. Legal challenge to Arpaio pardon begins (14)

6. Obama administration is close to announcing measures to punish Russia for election interference (13)
7. Trump to order regulatory rollback Friday for finance industry starting with Dodd-Frank (13)
8. Upheaval is now standard operating procedure inside the White House (12)
9. Trump Nominates Neil Gorsuch to the Supreme Court (12)
10. Republicans vote to rebuke Elizabeth Warren, saying she impugned Session's character (11)

Most Negative Submissions (number of comments):

1. Secret Service: We don't have any Trump White House tapes (120)
2. President Trump Cannot Redeem Himself (90)
3. Melania's Swat Proves She Hates Donald Just As Much As America Does (86)
4. Manafort spokesman to testify before grand jury Friday (82)
5. Source: Bannon was going to defend Trump until Trump attacked him (82)
6. Trump, Who Cited Bone Spurs To Avoid Military Service, Claims He Would Have Fought Off Parkland Shooter Without A Weapon (75)
7. Scaramucci Says He And Trump Talked About Pardons In Oval Office Meeting (71)
8. Was Trump slurring his speech during Israel remarks? (68)
9. GOP senator: Flynn will 'likely' be asked to testify (66)
10. Scaramucci: I will never trust a reporter again (65)

Conclusions:

Based on the data produced by my sentiment models, /r/politics hold a negative sentiment about Donald Trump. About 80-90% of the comments in all submissions during the evaluated time period reflected a negative sentiment, while only 20-30% of the comments during the evaluated time period reflected a positive sentiment. While sentiments on a day-to-day basis saw minor swings, /r/politics did not significantly change in either positive or negative sentiment during the evaluated time period. Arkansas, Wyoming, and Montana were the three most positive states when it came to commenting about Donald Trump, while Idaho, Ohio, Maine, and Oklahoma held the greatest negative sentiment towards DJT. The score of a submission did not have much effect on the percentage of comments having a negative or positive sentiment. Submissions consistently had a much higher percentage of its comment show negative sentiments than positive sentiments. A majority of the comments with high comment scores had a negative sentiment while also not having a positive sentiment. Overall, /r/politics could be seen as having a bias against Donald Trump and reflects an environment that does not think highly of him.

Questions:

1. Since the question states to only look at labeled_data.csv, I'm assuming that the functional dependencies to be listed are also contained only within labeled_data.csv. With that, I see Input_id => labeldem, Input_id => labelgop, and Input_id => labeldjt since an Input_id is unique and has three corresponding values. Since Input_id's are unique, they would correspond to only one certain value of labeldem, labelgop, and labeldjt. Labeldem, labelgop, and labeldjt do not have any functional dependencies since multiple comment id's can have the same value for labeldem, labelgop, and labeldjt.
2. The data appears to be partially normalized. The data is at least in 1NF, with each field only having one value and rows having a unique identifier. However, there is some data that is repetitive, such as subreddit_id, subreddit, author, link_id, author_flair_text, author_flair_css_class, and author_cakeday. Because comments can be permalinked to their own URL and possibly be evaluated outside of a submission, I think the lack of complete normalization is intentional so that when an individual comment is pulled independent of other resources, this data can still be accessed.
3. I chose my initial join that looks like

```
    joinedDF = labeled.join(comments, labeled.Input_id == comments.id,
"inner").explain()
```

Output:

```
== Physical Plan ==
*(2) BroadcastHashJoin [Input_id#170], [id#14], Inner, BuildLeft
:- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
:  +- *(1) Project [Input_id#170, labeldem#171, labelgop#172, labeldjt#173]
:     +- *(1) Filter isnotnull(Input_id#170)
:        +- *(1) FileScan parquet
[Input_id#170,labeldem#171,labelgop#172,labeldjt#173] Batched: true, Format:
Parquet, Location:
InMemoryFileIndex[file:/media/sf_vm-shared/p2b/labeled.parquet],
PartitionFilters: [], PushedFilters: [IsNotNull(Input_id)], ReadSchema:
struct<Input_id:string,labeldem:string,labelgop:string,labeldjt:string>
+- *(2) Project [author#0, author_cakeday#1, author_flair_css_class#2,
author_flair_text#3, body#4, can_gild#5, can_mod_post#6, collapsed#7,
collapsed_reason#8, controversiality#9L, created_utc#10L, distinguished#11,
edited#12, gilded#13L, id#14, is_submitter#15, link_id#16, parent_id#17,
permalink#18, retrieved_on#19L, score#20L, stickied#21, subreddit#22,
subreddit_id#23, subreddit_type#24]
   +- *(2) Filter isnotnull(id#14)
      +- *(2) FileScan parquet
[author#0,author_cakeday#1,author_flair_css_class#2,author_flair_text#3,body#4,
can_gild#5,can_mod_post#6,collapsed#7,collapsed_reason#8,controversiality#9L,cr
eated_utc#10L,distinguished#11,edited#12,gilded#13L,id#14,is_submitter#15,link_
id#16,parent_id#17,permalink#18,retrieved_on#19L,score#20L,stickied#21,subreddi
t#22,subreddit_id#23,subreddit_type#24] Batched: true, Format: Parquet,
Location: InMemoryFileIndex[file:/media/sf_vm-shared/p2b/comments.parquet],
PartitionFilters: [], PushedFilters: [IsNotNull(id)], ReadSchema:
struct<author:string,author_cakeday:boolean,author_flair_css_class:string,autho
r_flair_text:strin...
```

A BroadcastHashJoin is used in Spark when one table is small enough to be able to fit in RAM. Spark broadcasts the data to all of its compute nodes to increase possible parallelization. Spark first hashes the smaller relation (labeled.parquet) and then proceeds to scan the larger relation, looking for matching hashes based on the join key. The algorithm in use is a hash join.