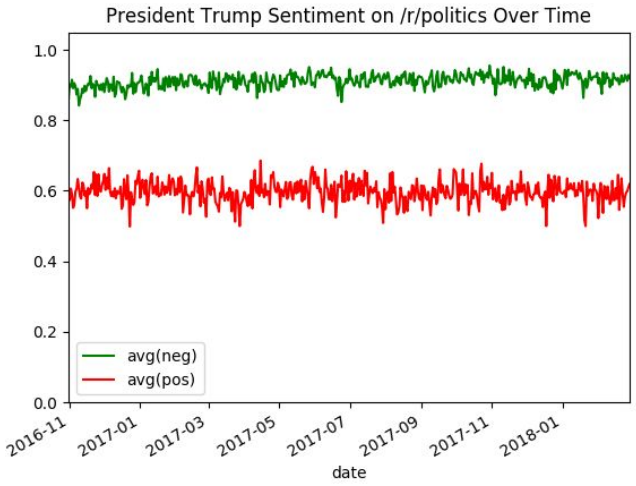
Mehar Nallamalli, Aditi Mittal
804769644, 104736807
June 2, 2019

Project 2b Report

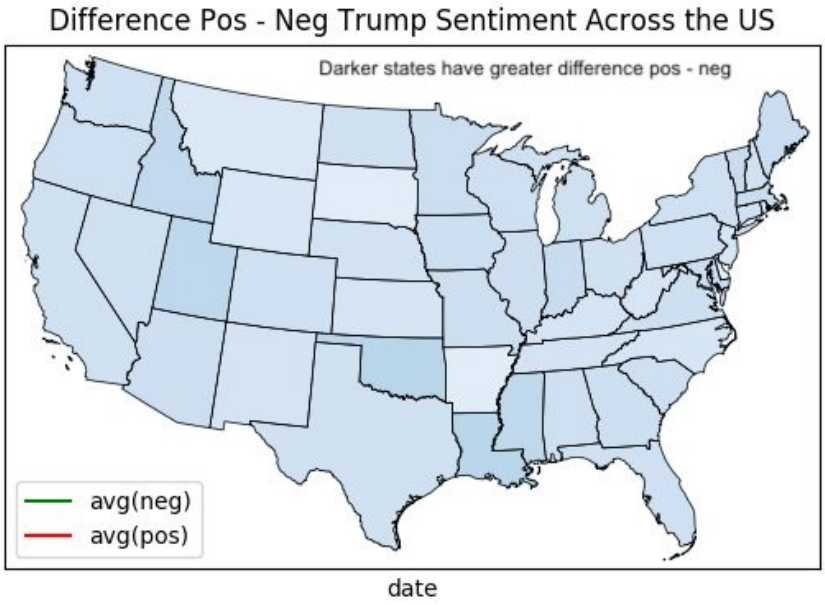
Plotting Exercises

1.



2.

3.



4.

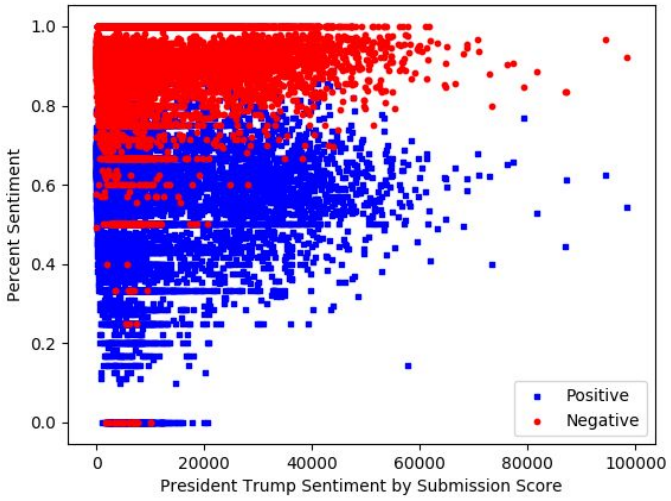
Top 10 positive stories

title	P
Trump: I never said 'Israel' in meeting with Russians	1.0
South Dakota Republicans are about to get rid of the state’s first independent ethics commission	1.0
“It’s back to work.” Trump’s schedule is nothing but livetweeting Fox lies	1.0
Bernie Sanders' Supporters Say the 'Justice Democrats' Can Move the Party Left to Fight Donald Trump	1.0
Chris Coons: Gorsuch nomination ‘almost certainly’ will require GOP to go ‘nuclear’	1.0
Trump adopting same behavior he criticized Clinton for	1.0
Conway: Media Obsesses Over Trump's Tweets & Ignores His Policies	1.0
Official to CNN: Drudge at White House 'all the time'	1.0
Trump to scrap Nasa climate research in crackdown on ‘politicized science’	1.0
Democratic Rep. Tulsi Gabbard 'Under Serious Consideration' for Trump Cabinet	1.0

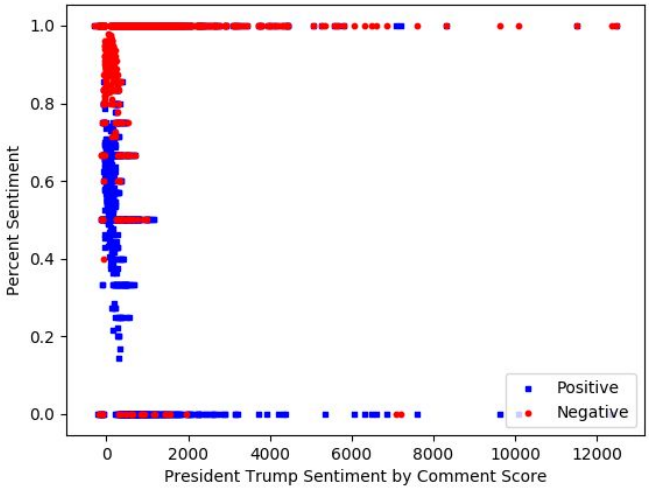
Top 10 negative stories

title	P
The White House is struggling to prevent a crippling exodus of foreign policy staffers eager to leave before the arrival of the Trump administration, according to current and former officials.	1.0
Trump adopting same behavior he criticized Clinton for	1.0
Democrats step up calls that Russian hack was act of war	1.0
James Comey in early talks with special counsel Mueller on Russia testimony, memos	1.0
Conservative columnist: Founding Fathers had Trump in mind when they included impeachment clause	1.0
Moore Doubles Down On Refusal To Concede Race, Suggests Voter Fraud	1.0
Trump Humiliated Jeff Sessions After Mueller Appointment	1.0
Sanders vows to 'radically transform the Democratic Party'	1.0
German foreign minister goes beyond other US allies to decry 'nepotism' of Ivanka Trump role	1.0
Paul Ryan Blocked the Delivery of 87,000 Petitions Asking Him Not to Defund Planned Parenthood	1.0

5.



a.



b.

6.

One thing I found by the submission score plot is that even though it looks like there are many negative sentiments, the range of negative comments is smaller

than positive comments. That means that there are people who don’t like Trump, but also, give him the benefit of doubt by only liking him a little bit. If we assume that 49% of positive Trump submissions can be considered a negative sentiment since there is more negative than positive, then only a small sliver in the middle are people who are positively opinionated about Trump. All 0 - 100% of Negative submissions are against Trump, and 49% of the positive comments are also against Trump. Interesting to see that people who are positive to Trump are majority only the blue in between .6 and .8 even though it looks like the blue spread is larger than red. The same goes for the sentiment score based on the comments.

Looking at plot 1, The positive timeline is lower than the negative timeline. The positive timeline also has more range than negative timeline which means that more people who are positive to Trump are more malleable to changing their opinions about him. The negative timeline is pretty stable and fluctuates less than the positive timeline.

Answers to Questions

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

ans:
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?

ans:

comment_id	created_utc	body	author_flair_text	comment_score	submission_id	title	submission_score	ngrams	pos	neg
dmmvmt3	1504700602	Read his official...	null	14	6ycafy	Pastor forced fro...	3130	[read, his, offic...	0	1
die2qd4	1496454687	One question duri...	null	3	6ew07f	Trump Campaign Se...	11209	[one, question, d...	0	1
do5ssey	1507621241	The 2000 run on a...	null	3	75apgj	ESPN's Stephen A....	6767	[the, 2000, run, ...	0	1
ddxqgbv	1487509130	I really hope we ...	null	2	5uxtk6	Are Liberals Help...	26	[i, really, hope,...	1	1
ddra2g1	1487122478	I am waiting till...	null	1	5u0i7m	CNN anchor: 'Zero...	28274	[i, am, waiting, ...	0	1
dclevsf	1484765882	Don't become "fak...	null	8	5opf7m	Trump rips NBC Ne...	6784	[don't, become, f...	1	1
da9cubk	1479723466	Remember to call ...	null	-20	5e1x77	Donald Trump's Sw...	1096	[, ,]	0	1
d9uvuyt	1478811992	I don't know, it ...	null	4	5ca15m	Best friends? Tru...	102	[, ,]	0	1
d9toto7	1478740445	They bet all they...	null	1	5c2ig1	Calls grow for Be...	34145	[they, bet, all, ...	0	1
dj2njje	1497809248	No one cares what...	null	-28	6i0zcx	Former Obama aide...	429	[no, one, cares, ...	1	1
ddvn4cl	1487365810	Serious question....	null	2	5undrc	One Million Peopl...	47763	[serious, questio...	0	1
dckh70s	1484707070	I am attending a ...	null	11	5okf1t	Trump inauguratio...	2125	[i, am, attending...	1	1
dckhwxu	1484708045	As an ex-teacher ...	null	7	5omcmg	Betsy DeVos says ...	3429	[as, an, ex-teach...	0	1
dpcge8f	1509822915	In the article it...	null	1	7as7kn	Donna Brazile: I ...	78	[in, the, article...	0	1
ducvo7x	1518812951	Y'all have the 'd...	null	3	7y162q	Preet Bharara tro...	11396	[y'all, have, the...	0	0
dlei4ln	1502320692	They could have ...	null	5	6sm791	Trump called for ...	39184	[they, could, hav...	0	1
dpawy8b	1509736589	Yep, American is ...	null	1	7aiyyp	Donald Trump twee...	33738	[, ,]	0	1
dpdzqv1	1509908607	It has everything...	null	1	7awv72	Donald Trump's pr...	23550	[, ,]	0	1
dnj3zhh	1506433053	I wonder if the a...	null	5	72j4vi	Steve Bannon trie...	4626	[i, wonder, if, t...	0	1
da67fgq	1479504906	You're way too op...	null	1	5dmv7n	Voters In Wyoming...	5414	[you're, way, too...	0	1

Above is the comments that dataframe that we generated. Looking at the table, it looks normalized. Each column is important to the integrity of the comment to its related post. There aren't any functional dependencies that might cause a duplicate entry or redundant information because the comment_id is related with the submission_id since that comment is for that specific post. We believe the collector stored the data this way so that he can search by submission_id, and see all the different comment_id's associated with that post. Obviously, the title, body, score, ngrams are dependent on the submission_id because each submission will have different information posted by different people. To ensure, we can also add a user_id to show which user has posted the original post.

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?

ans:

```
== Physical Plan ==
*(2) Project [id#14 AS comment_id#541, title_id#256, created_utc#10L, author_flair_text#3,
title#106, body#4, score#20L AS comment_score#542L, id#69 AS submission_id#543,
score#92L AS submission_score#544L]
+- *(2) BroadcastHashJoin [title_id#256], [id#69], Inner, BuildRight
   :- *(2) Project [author_flair_text#3, body#4, created_utc#10L, id#14, score#20L,
regexp_replace(link_id#16, ^t3_, ) AS title_id#256]
   :  +- *(2) Filter isnotnull(regexp_replace(link_id#16, ^t3_, ))
   :     +- *(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/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, score#92L, title#106]
         +- *(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/submissions.parquet], PartitionFilters: [],
PushedFilters: [IsNotNull(id)], ReadSchema: struct<id:string,score:bigint,title:string>
```

This output of the explain for our sql query uses a broadcasthashjoin rather than a simple hashjoin. After some research, we found that broadcast join is useful if one of structures is relatively small. Otherwise it can be significantly more expensive than a full shuffle. The join is happening on the IDs of the tables that we specified 256 and 69 and is doing an inner join, slowly building the table right. After it hashjoin, it will project the selected columns which the select operator.