Spring 2019 CS143

Project 2B - Spark Machine Learning Project

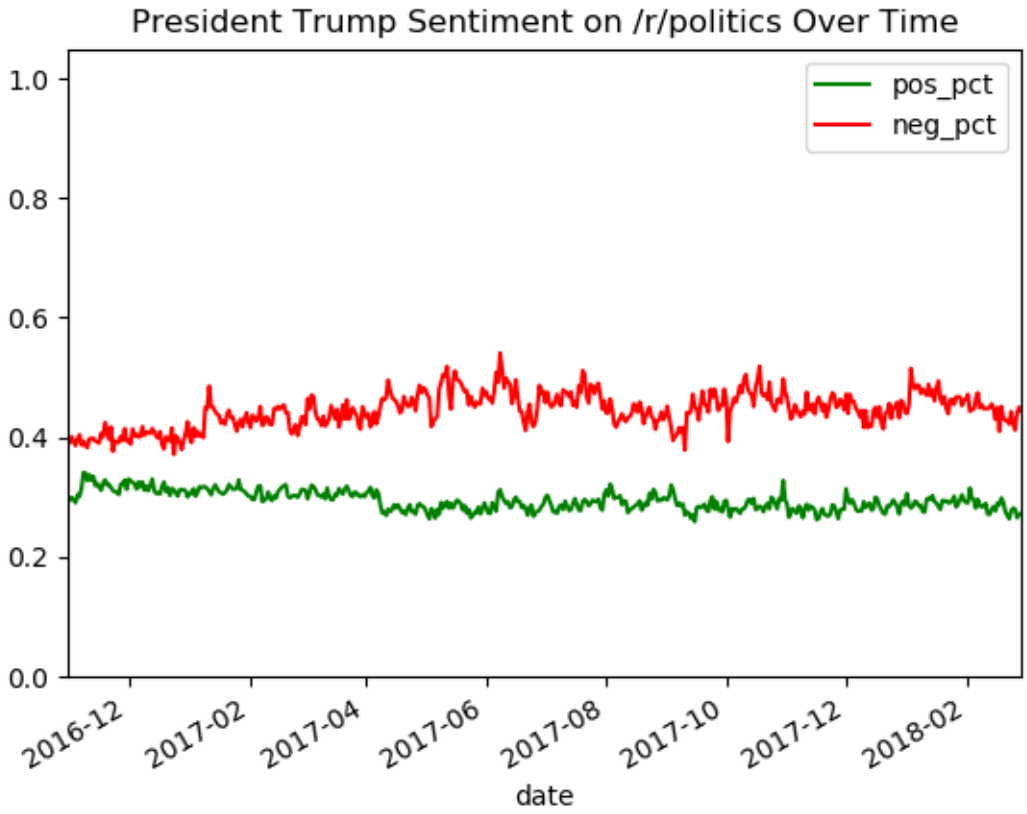
June 3, 2019

Wilson Jusuf
Student ID: 404997407
University of California, Los Angeles

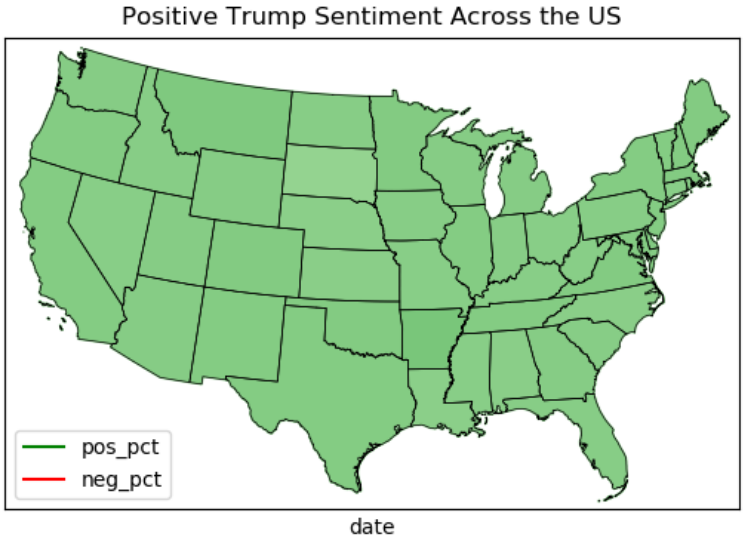
Hanif Leoputera Lim
Student ID: 504971751
University of California, Los Angeles

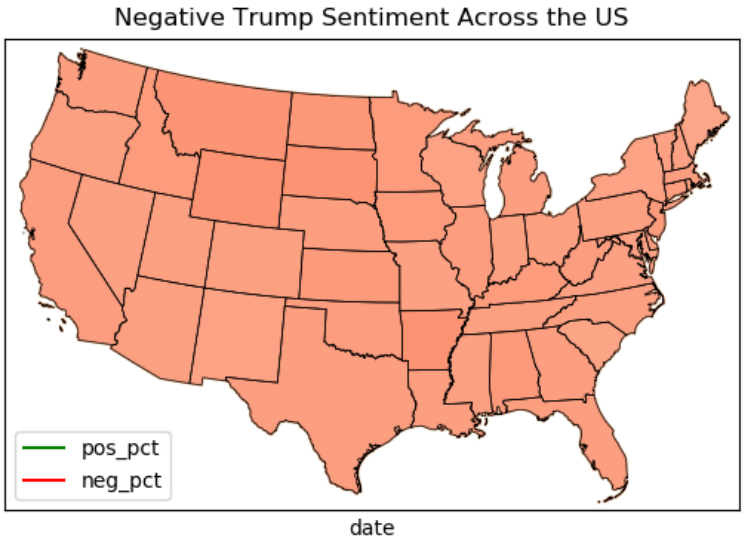
1 Findings

1.1 Sentiment Time series plot by day

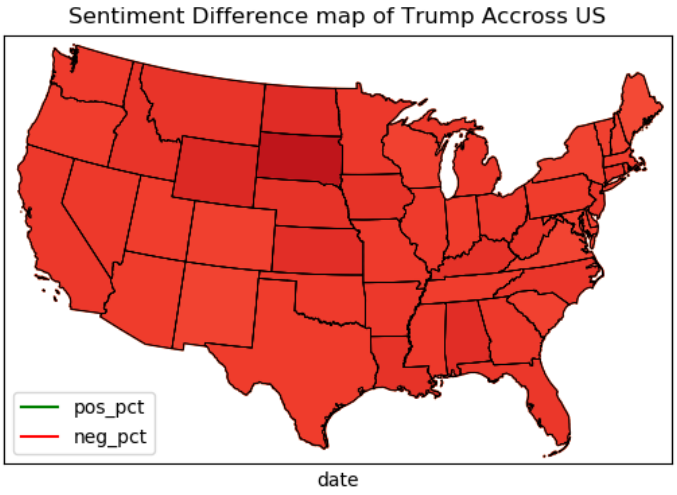


1.2 Sentiment by State





1.3 Sentiment Difference by State



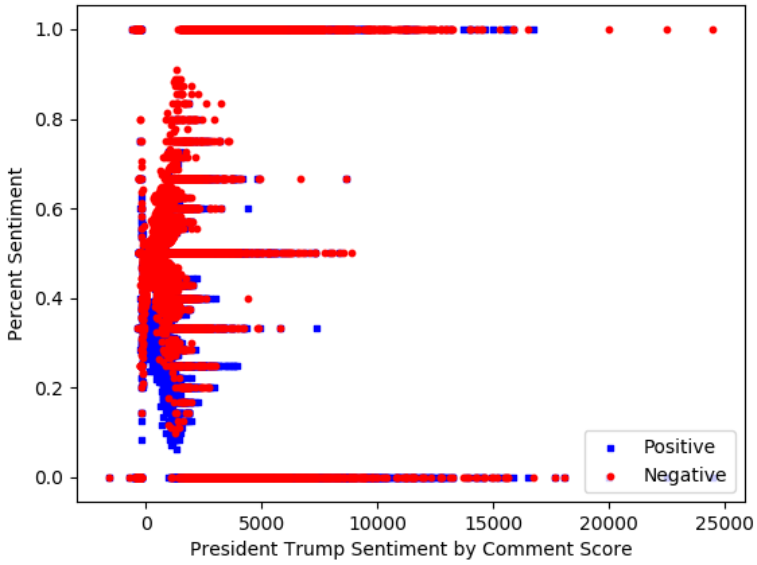
1.4 Top 10 Positive Stories

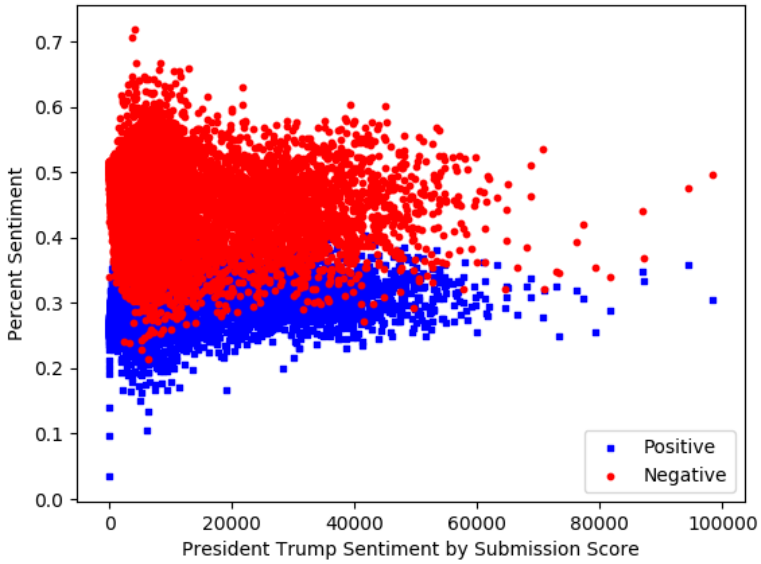
```
title,pct_pos
Sean Spicer finally gets to meet Pope Francis,1.0
"Breitbart under Bannon: Breitbarts comment section reflects alt-right, anti-Semitic
↪  language",1.0
"After Trump tweets, pressure grows for full Russia investigation",1.0
Confirmed: Donald Trump Says He Will Take $1 Salary as President,1.0
Is President Trumps Immigration Order a 'Muslim Ban'?,1.0
Kamala Harris says she'll co-sponsor single-payer health care bill,1.0
The Left Has Lost its Mind: Cory Booker to Testify Against Sessions,1.0
EU says to stick to Iran rapprochement despite Trump's criticism,1.0
"Tom Perez: Win or Lose, I Will Work to Unite the Democratic Party",1.0
Maxine Waters: Trump is the most deplorable person Ive ever met,1.0
```

1.5 Top 10 Negative Stories

```
title,pct_neg
"Trump's Government wants to punish people who speak out against him, especially if
↪  they're black women",1.0
"Mnuchin Asked for Tax Dollars to Fund His Honeymoon, Despite $300 Million Net
↪  Worth",1.0
Trump and North Korea war of words escalates,1.0
Trump's 'Loser' Speech in Bethlehem,1.0
Trump revives criticism of 'both sides' in Charlottesville,1.0
How Deutsche Bank Enabled A Dirty Offshore Bank To Move Dark Money,1.0
Trump again seizes on terror incident to call for travel ban,1.0
Trump says hes fairly close to deal on young immigrants,1.0
Judge questions tossing out Arpaio conviction after pardon,1.0
Trump repeats 'both sides' Charlottesville rhetoric,1.0
```

1.6 Scatterplot of Comment Score and Submission Score vs Sentiment





1.7 Summary of findings

Overall, the sentiment of trump was overall more negative in all states. The positive percentage was considerably lower than the negative percentage. However, the intensity of the negative sentiment varies by state.

Both negative and positive stories involve Trump in the title. It relates to either Trump or his administration.

Looking at the scatterplot, we see that the sentiment percentages converge to neutral (0.5) as the comment score increases. However, there are also extreme stories (sentiment of 1.0 (100%) or 0.0 (0%)) getting high comment scores.

For submission score, the sentiment weakly converges to around 0.4 as submission score increases. The submissions that are more 'neutral' therefore tend to have higher scores.

2 Questions

1. Take a look at `labeled_data.csv`. Write the functional dependencies implied by the data. Set $I =$ `Input_id`, $D =$ `labeldem`, $G =$ `labelgop`, $T =$ `labeldjt`.

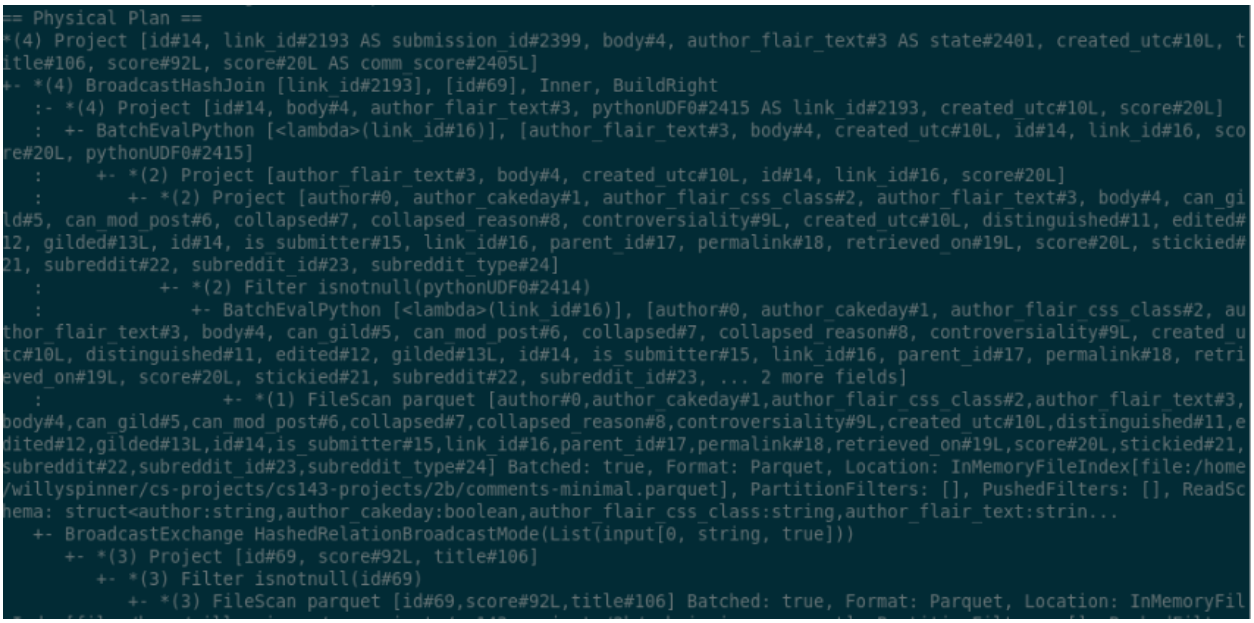
   The dependencies are:

   - $I \to D$
   - $I \to G$
   - $I \to T$

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?

   The data frame isnt normalized. There are redundancies like (`subreddit_id`, `subreddit`), so updating the subreddit would require us to update `subreddit_id` too. This could lead to insert/update integrity if the DBA isnt careful. To decompose it, we could break relation R into R2(`subreddit_id`, `subreddit`), while R1 will have everything in R except for `subreddit`. However, one reason why the previous schema was used is because joining two separate tables are expensive. Querying which subreddit a comment come from, is probably so common that these two tables are frequently joined, hence computationally wasteful.

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?



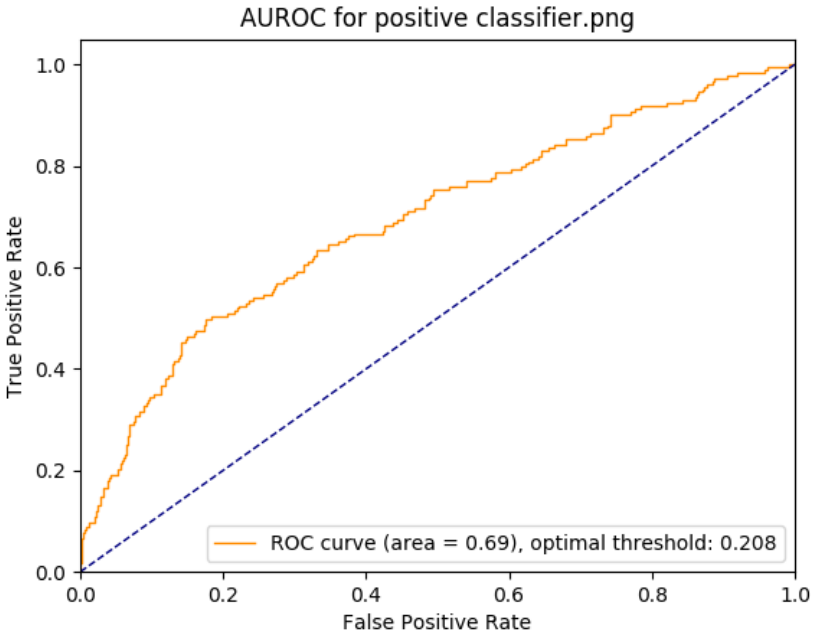
   Spark first sequentially scanned the parquet, select all the attributes in the relation and then select all the attributes that we wanted. Once projected, the smaller and filtered table will be broadcast hash joined on the attribute we specified. Once joined, the attributes that we want are selected again. In Spark, table will be transmitted through the network with broadcast hash joins. Transmitting a huge table is prohibitively slow. So, we feel this is the reason why spark seems to redundantly select only the attributes we need before it gets broadcasted. Spark is using broadcast hash join. Spark is usually ran across many machines i.e. RDD, thus it usually have to join across partitions in different machines. Thus, the reason why they use broadcast hash join.

3 Classifier

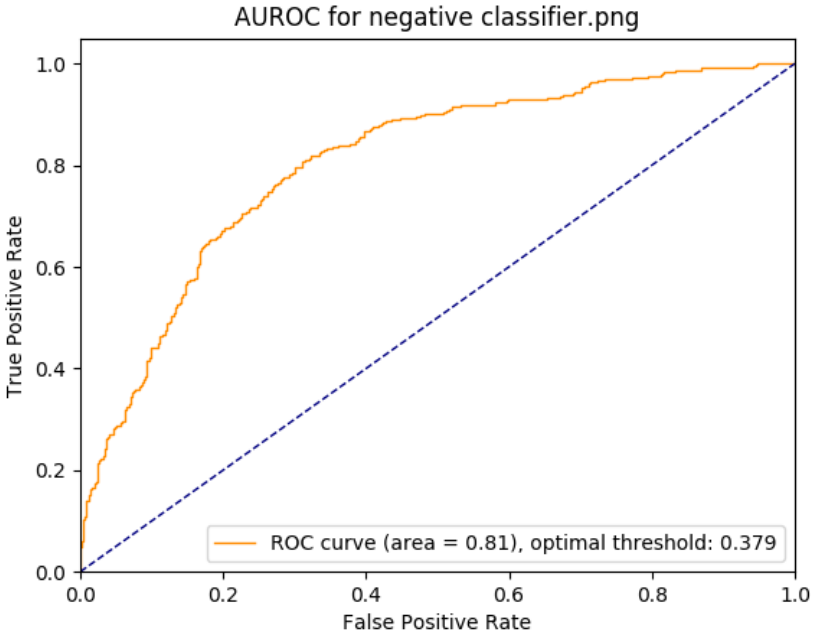
We changed the positive and negative classifiers' thresholds to their optimal values based on how close it is to the point (FPR $= 0,$ TPR $= 0$). Formally, we want the threshold T as the minimal euclidean distance:

$$\hat{T} = \underset{T}{\operatorname{argmin}} \sqrt{(\mathrm{FPR}_T)^2 + (1 - \mathrm{TPR}_T)^2} \tag{1}$$

Here are our classifier auROC graphs:



classifier.png



classifier.png