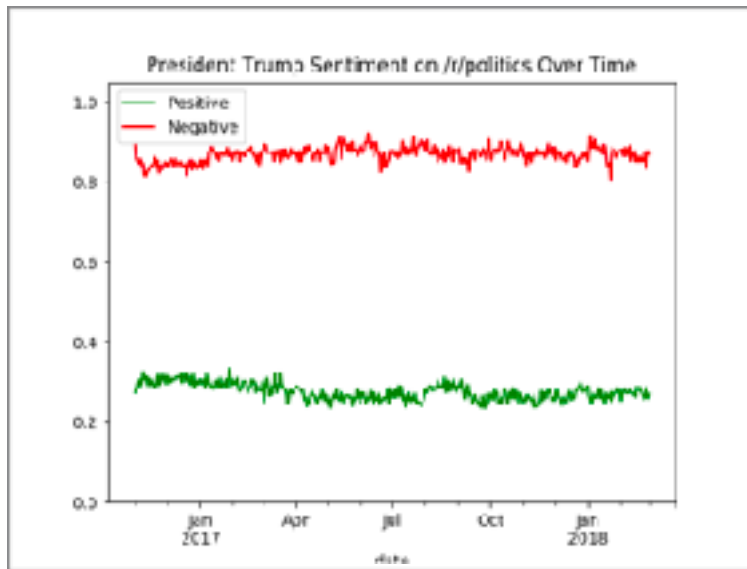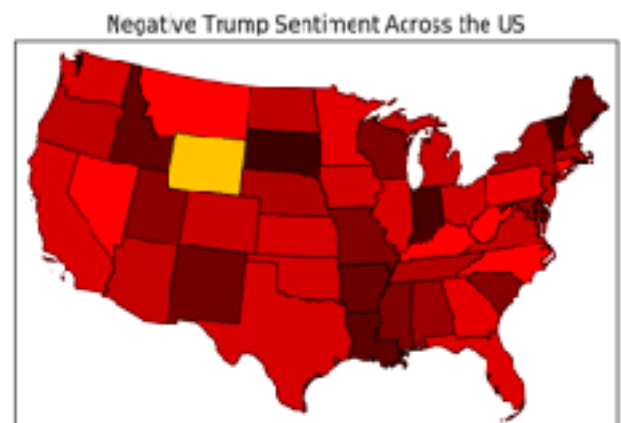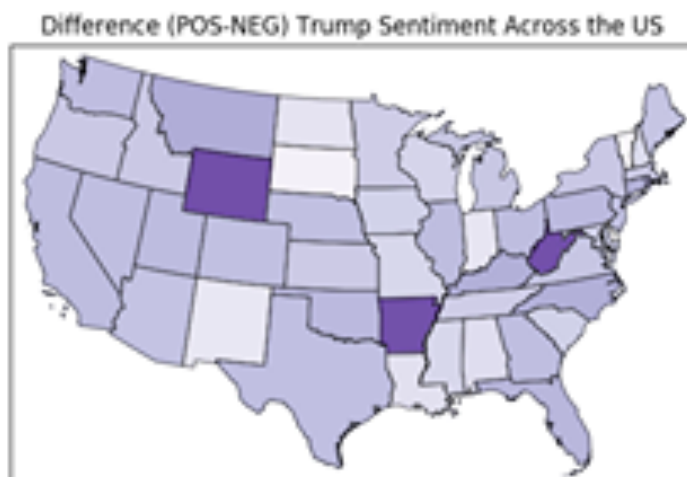## 1. Time Series Plot

2. Positive and Negative Sentiment Maps
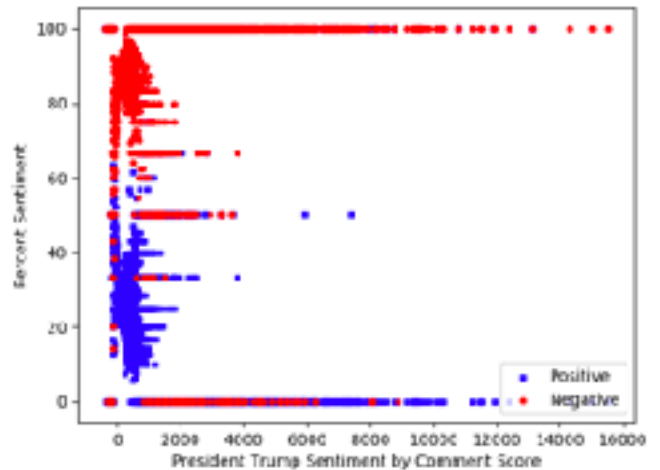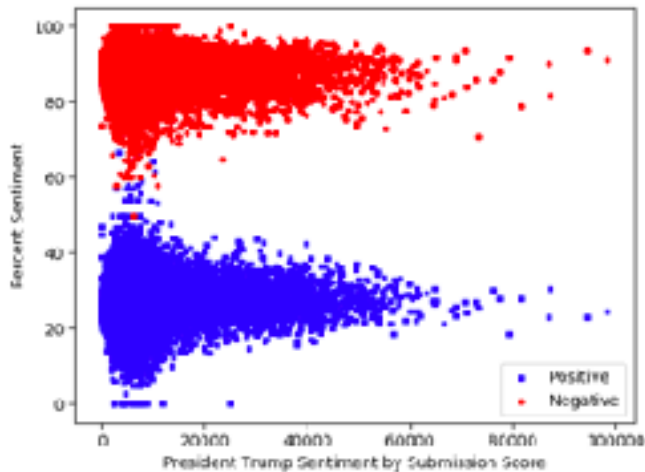

3. Difference (Positive minus Negative) Map

4. Most positive and most negative submissions (the lists contain the submission_id's)

```
+-------------+        +-------------+
|most_positive|        |most_negative|
+-------------+        +-------------+
|      5kyoao|          |      69euq8|
|      66esg4|          |      6ardm6|
|      5plni4|          |      69fa14|
|      6y8ul4|          |      66b1nk|
|      5w8kcd|          |      69fdar|
|      62bcl6|          |      69g98r|
|      655jy0|          |      6afhol|
|      639xea|          |      6ad6cp|
|      7paxu2|          |      6623yq|
|      5ncsql|          |      6adp7i|
+-------------+        +-------------+
```

5. Scatter Plots

6. Extra credit plots

We created the same plots for the other 2 labels: democratic and GOP, in order to compare the sentiment to that expressed towards Trump.

Democratic:

Negative Democratic Sentiment Across the US

Democratic Sentiment by Submission Score

Positive Democratic Sentiment Across the US
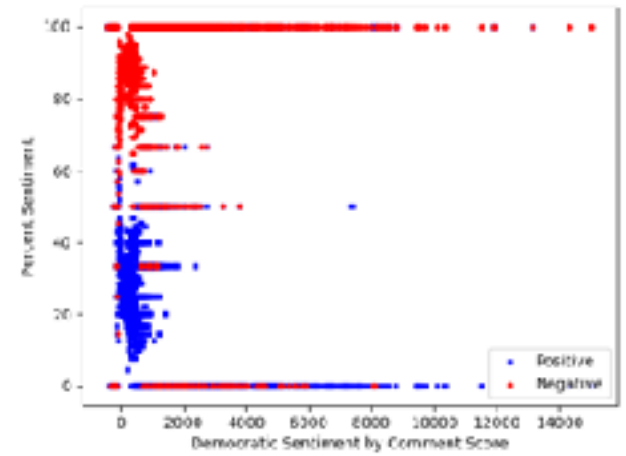

Democratic Sentiment by Comment Score

Difference (POS-NEG) Democratic Sentiment Across the US

Democratic Sentiment on r/politics Over Time
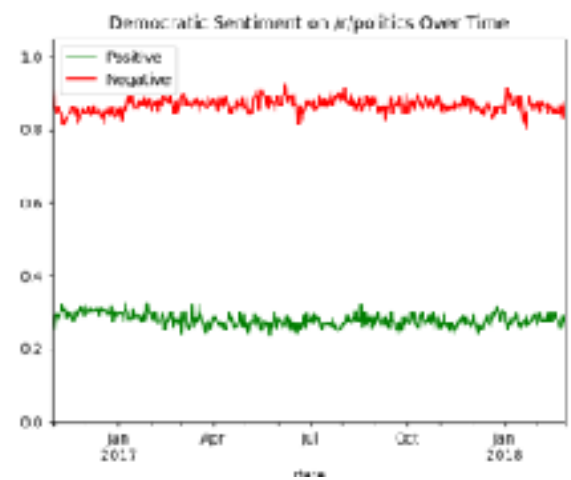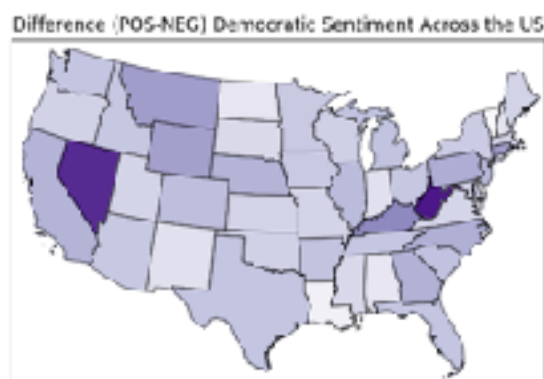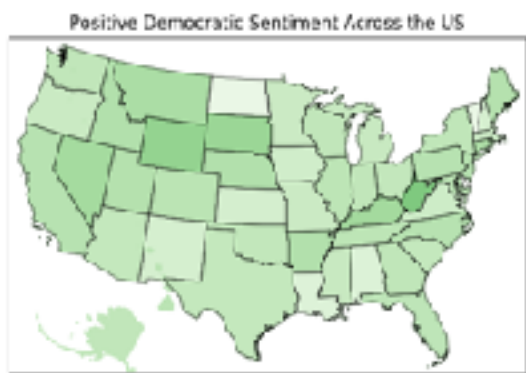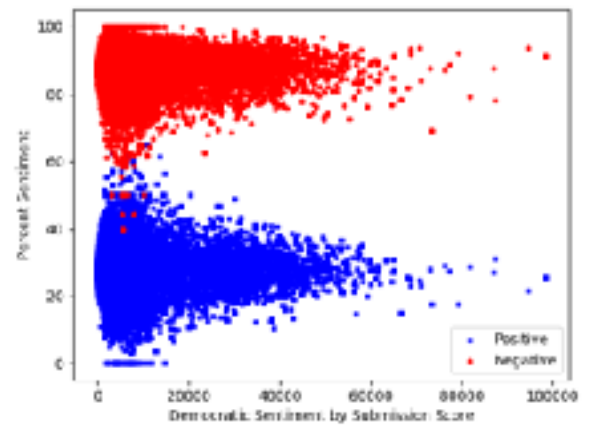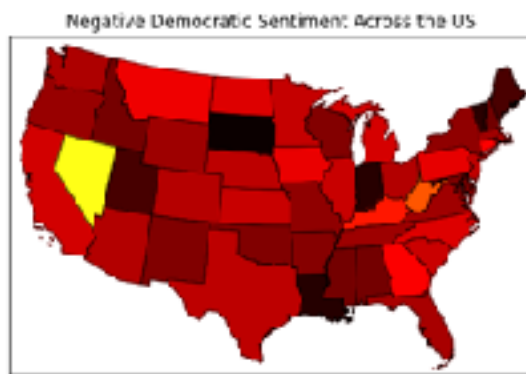
7. ROC curve

8. Paragraph summarizing findings

We notice that /r/politics seems to have an exceedingly negative opinion of President Trump from before he was elected to months after he was inaugurated. There is not one state whose Reddit users express more positive than negative sentiment about Trump. It is Reddit after all. By our model, Arkansas has the most positive things to say about Trump, which isn't surprising. Historically, Arkansas is as red a state as you can get, so our model seems to have gotten something right. On the other end of the spectrum, users from Delaware and Vermont, blue states, have the least positive sentiment in their Reddit comments. When we observe the sentiment towards the democratic party, the shades of quite a few of the states seem to invert. States that had more positive things to say about Trump don't have the same positive attitude towards the democratic party and vice versa. Overall, Reddit seems to be a forum for people to voice negative thoughts. This probably isn't because people are naturally pessimistic or angry, but rather because people like to challenge other people's opinions and stories and the natural sentiment that our model gleans from that is negative.

QUESTION 1:

Check out some rows in the labeled dataframe: labeled.show()

Functional Dependencies:

- Input.id ---> labeldem, labelgop, labeldjt

Which implies, by Armstrong axioms:

- Input.id ---> labeldem
- Input.id ---> labelgop
- Input.id ---> labeldjt

- Et al. by decomposition/association

QUESTION 2:

Checkout the schema for comments dataframe: comments.printSchema()

root
l-- author: string (nullable = true)
l-- author_cakeday: boolean (nullable = true)
l-- author_flair_css_class: string (nullable = true)

l-- author_flair_text: string (nullable = true)
l-- body: string (nullable = true)
l-- can_gild: boolean (nullable = true)
l-- can_mod_post: boolean (nullable = true)
…
Checkout a row in the dataframe: comments.show(n=1)

Is it normalized?

There are 3 distinct entities involved in the table: sub-reddit, user, comment

- The comment id is candidate key of entire relationship.
- User data like flair, cake etc. has partial dependency on author
- Subreddit data like sub name has partial dependency on the s_id

So, considering basic 2NF criteria:

- Table is not normalized.
- As a starter, can be decomposed into distinct comment, user and sub-reddit tables.
- Keyed by id, author and s-id attributes respectively.

Consider comment data:

- Candidate key is id + any trivial superset
- Functional Dependencies:
- id —> all attributes (by def of key)
- body —> link_id i.e. a transitive dependency that can be decomposed in 3NF

Why is given table not normalized?

- Typically, every rendering of a comment on a reddit page uses all the attributes in the non normalized table
- If normalized, we'll need to compute an expensive JOIN for each useful render of a comment
- So even though there is redundancy that can be removed, the data makes sense collectively - particularly on the front-end.