# Data Mining Reddit Comments

Sarah Barili
Sarah.Barili@colorado.edu

Kathryn Gray
Kathryn.Gray@colorado.edu

Roy DeJesus
Roy.DeJesus@colorado.edu

Jia Lin
Jia.Lin@colorado.edu

## 1. INTRODUCTION

Reddit is considered ”The Front Page of the Internet” by it’s users. It is driven by a strong, user-centered subculture that revolves around alternative knowledge and information discovery, as such it can be sourced as a constant stream of new information regarding a vast range of topics. There is significant potential to derive meaningful patterns and trends from this data, and apply it to problems concerning anything from natural disaster prediction to mental health diagnosis.

This project will revolve around Reddit, its information, and people that use it. We will be using this new found knowledge and the culture, the style, and the judgement of its people to lay the ground work that will support us and allow us to build the walls and the many floors of this project.

## 2. BODY

### 2.1 Motivation

The aim of this project is to find find meaningful patterns in Reddit User comment data, and apply it to to solving one or more of several proposed problems:
-Determining which words and phrases were used most in political sub-reddits
-Cross checking users posting in the political sub-reddit with other comments and frequently used sub-reddits
-Identifying which cities are most frequently mentioned by users posting in the political sub-reddit
-Identifying user sentiment on political topics
-Determining whether there is any correlation between users, what places they mentioned, and how those places voted

These steps will help determine whether we can identify political leanings using Reddit. This is a relevant question because the recent election polls did not accurately predict results. While we are not planning to look at multiple months, this method could be used in the future to discern whether this is a more accurate way to determine voting results.

### 2.2 Literature Survey

Many research projects followed from the publication of Reddit comment data. There has been work done on analyzing users, types of comments, linguistics, and comments on specific topics.

In ”Role of the Social Web in Behavioral Therapy” researchers analyzed mental illnesses through the user’s comments[4]. This was accomplished in a multi-step process. First, the language of ”self-disclosure” among reddit users is identified with text mining. Researchers mined the frequent words in three identified mental health-oriented sub-reddits, and compared these words to a psycholinguistic lexicon to determine which types of words appeared the most often (i.e. words that are ’emotionally expressive’). Using this, they built a statistical model to predict the factors that determine maximum social support on posts in these sub-reddits. Finally, this information was applied to identify the role of ’disinhibiton’ in the Reddit community due to analyze its effectiveness in getting users to disclose thier mental illness. In this way, Reddit posts could be proven useful to predict or identify mental illness in its users as a means to provide better assistance to these users.

There has also been research into the types of comments that users post. These researchers studied which comments had the most upvotes, based on when they were posted and how long the user had been on Reddit [6]. This paper also had research on how many comments a user had posted based on how long they had been a part of the Reddit community. They also researched how many upvotes and downvotes a comment got based both on how many upvotes and downvotes its parent got and how high in the comment tree the comment was.

In ”German Reddit Corpus”, researcher Adrien Barbaresi analyzed comment contents to build a linguistic corpus of German language based on its use in German sub-reddits[3]. The data was visualized to map the possible geographic locations of German-speaking users. He first utilized language detection software to store a words and phrases as ”tokens” in a dictionary structure. From this he was able to derive useful information, such as the frequency of German word usage in sub-reddits, and the most common nicknames used by authors in these posts. By comparing the dictionary of tokens to a lexicon of nouns that describes cities, and a geonames database that maps cities to specific locations, he was able to geographically locate the posts based on city name mentioned in comments.

In ”Analyzing Trump v. Clinton text data at Reddit”, Emily Gao analyzed the comments from the r/politics sub-reddit for July 2016 [5]. By identifying the most popular words and phrases in these comments, and finding correlation between the timestamp of the most frequent words used and significant happenings in the news, Gao was able to identify what reddit users were most concerned about in the months preceding the 2016 elections. First, she mined the dataset for the most frequently used tokens (words or phrases) used in the sub-reddit in July 2017. Second, she analyzed the frequency of each word according to the timestamp of the comment or post, and provided a useful visualization to identify certain periods during the month that these words seemed to spike in popularity. Using this data, she then identified the top 20 most popular comments (based on votes) corresponding to these days, and derived a collective user sentiment on these topics and words. For example, The word ”Hillary”, and the topic ”email servers” seemed to spike in popularity on July 5th, and mining the top comments for this day revealed collective user sentiment that could have been used to more accurately predict the results of the 2016 election that followed several months later.

## 2.3 Proposed Work

Before we can pre-process and analyze the data. We have to understand JSON. While JSON is a popular name in most computer science student’s minds; there are more to it than just load and write. Also most students may not know how to code using JSON. We begin by making sure every one can load and read the data. Understanding how to read the data comes first with this project, but more than that, understanding how to read it fast is key since our data set is 42GB with tens of millions of comments, each with its own fields. We can cut valuable time off compiling and run-time by ignore useless fields, and passing on un-related sub-reddits. Finding a fast search and sorting algorithm is required since, our current estimates of just reading the data file be a couple of hours.

We will also have to familiarize ourselves with python coding. While there are other languages that can import and export JSON files, python will easier to learn if needed because of the simple syntax. When coding with a big file like this, errors tend to appear. Python errors never pass silently unlike other programming languages, so using python will allow us to see and read why the program has crashed and fix it. Python has a vast standard library that will allow us to pull functions that will make this project go smoother.

Several steps will have to be taken to pre-process and analyze this data. Integration with another data set is not necessary, as the data is contained in one large format JSON file. Although, in the future, we may integrate this data set with other sets with information on election results, location data, and the like. Upon review of the JSON formatted data, it is adequately cleaned, and none of the fields needs to be deleted or normalized. The dataset needs to be significantly reduced. For this project, only comments from specific sub-reddits will be used, namely, r/politics and related sub-reddits. Once reduced, this data will be moved to a relational database system for easier analysis.

After pre-processing, the comments’ field ’body’ will be mined to build a linguistic ’corpus’, to identify significant words and phrases. This will include searching for names of people and places, patterns, and emoticons in the body of the comments. Data will be searched for users that post most frequently in the r/politics sub-reddit. After useful information is extracted from the data, a visualization of the geographic distribution of words and phrases based on commonly mentioned cities will be constructed.

Previous work on this topic used text mining techniques to analyze the frequency and use of the German language in various sub-reddits. The analysis of the r/politics sub-reddit was restricted to the month of July 2016, and focused on finding a correlation between the timestamp of a post and current events in the news. In this project, the focus will be on aggregate frequency of words and phrases in the r/politics and related sub-reddits using techniques employed to build the German linguistic corpus, rather than the correlation of word frequency to current events. This analysis could then be compared to previous work on the r/politics sub-reddit to compare trending topics before and after the election and concurrent inauguration.

## 2.4 Data Set

The data set being used was found on the reddit website[2]. It is approximately 42GB of user comments from the month of January 2017. The data is stored in JSON string format; each object represents a comment. There are 78,946,585 comments in the data set and each object has the following potentially useful fields: Body, edited, Author, sub-reddit, sub-reddit ID, Author Flair, Stickied, ID, parent ID, score, retrieved on, Controversiality, and link ID. The body contains the actual text of the comments, which will be used to mine for frequent words and phrases. The sub-reddit and sub-reddit ID will be useful for our initial data reduction. The Author field stores the username of the author of each comment. This could be used for tracking users to determine the types of comments likely to post. We may look at the score, which tracks the up-votes and down-votes, to determine how popular or unpopular an opinion is with the given population of the sub-reddit.

One thing to understand about reddit is that it’s a world wide Internet form. If we where to search by the English alphabet, there is a huge likelihood that it would return with tons and tons of errors. An example of this would be to look through a Chinese sub-reddit. The hanzi would alone be hard for any program to accept. In situations like this we would iterate over unicode codepoints, then encode them individually for output. This would most definitely increase compile time and run-time. Using the hanzi example as troubles that we would be intercepted by, we would have to write code that accepts other alphabets or other characters and either skip over all of them or somehow translate them and then run them through again.

The score would also be hard to look through. The score is a profile specific thing, where the user can either hide it or not let other people vote on their comment. To make sure that this problem doesn’t crash our program, we would have to carefully code the program to loop through the file and remember where there are comments that have no score field, then re-loop the file and track the comments that do have score.

The examples of trouble given above can apply to every field in every comment. Writing a program that can run through all of that and return no errors will be a huge undertaking. This is why we would be writing in python; with it’s many libraries, we can find translators for other languages. The problem with missing fields can be handled, because of the JSON formatted data. With JSON, we can specify our search result to only objects that have score fields, or other fields, which will severely decrease our compile time and run-time.

## 2.5 Evaluation Methods

We will not have to clean our data, since all the fields are filled with data that will work for us. Each object from the JSON is a single comment, so we do not have to deal with redundant data of any kind. We will need to reduce our data into a more manageable size. The first way is to make sure we are only looking at the comments written in English. Once we have those comments we will split on the sub-reddit or sub-reddit ID field, looking specifically at r/politics and related sub-reddits. Once we have this smaller data set, we will use sampling to determine common words and phrases we will be looking for. Although sampling may not catch all of the frequent words and phrases, we gain a good understanding of what we will be looking for in our complete data set. Once we go through and determine which tokens we are looking for, we can remove any comments that do not contain any of them. In this way, we can again reduce the number of items we are looking at. Once we have our reduced data set, we can find the frequency of any tokens found as well as determining the sentiment of the user's comment.

After this, we will again look at our whole data set to determine what other sub-reddits the users were posting, or subscribed to. This will give us more information about the users themselves. From here, we will look at the posts of these specific users and determine what places they mention. We may try to remove any data mentioning "travel" or places mentioned in traveling sub-reddits. We will need to do some normalization here since cities are more likely to be mentioned than towns, and cities could have different political views from the surrounding towns. We will integrate another data set with election data at this point and try to find correlations with user sentiment and how the area voted.

Reducing data is only half the battle. The other half is the sorting algorithm. There are tons of big name sorting algorithms out there, like: bucket sort, bubble sort, heapsort, quicksort, quick select, spaghetti sort, or radix sort, just to name a few. A first step is figure out what we are sorting. The score field seems to be the best, and easiest, thing to sort. We could add weights to words that we want to see and then sort the comments for ones that have most of the words we are looking for.

We want something fast because if our sorting algorithm's average run-time case is $O(n^3)$, then we input 73 million comments in, and for fun let's say it takes 1ms(a really slow computer) to look through one comment, then we'll be waiting for around 12 years for this program to finish. That is not good. We cannot accept any sorting algorithm's worst run-time case, not average run-time case, that is bigger than $O(n^2)$. Our current thought is to use quicksort, its worst average run-time case is $O(n^2)$, but it's average run-time case is $O(nlogn)$.

Quicksort is a comparison-sort, which means it can sort any items with a "less than" relation. To do this with comments, as stated previously, we can give a number weight to words that we're looking for, artificially give comments a number value. Quick sort is not a stable sort, meaning that the relative order of equal weighted items are not preserved. We don't find this troublesome since we are looking through all of the comments but this will tell us which comments may hold more of what we're looking for, and will look at those first.

Quicksort is a divide and conquer sorting algorithm. It divides the array into two smaller arrays, one with low elements, other with high elements. To do this, it picks a pivot. loop through the array and sets the element in the higher element array if it's bigger than the pivot; if it's lower, then it goes into the lower element array.

## 2.6 Tools

### 2.6.1 JSON

Our dataset is JSON formatted. JSON stands for JavaScript Object Notation. JSON is a very clean format; it already parses our data in sub-sections, which we call fields. Each field has the name of the field and what goes with that field, basically a collection of name/value pairs. A reddit comment goes into the "body" field, which is the main focus of this project.

### 2.6.2 Python

To take advantage of the JSON format, we decided to use Python as our main language. Python emphasizes code readability. It can express concepts in fewer lines of code than most other languages. Python features a dynamic type system, an automatic memory management and supports multiple programming paradigms. It also has a large and comprehensive standard library. These libraries contain various convenient JSON oriented libraries. We will be utilizing them to their full potential. Python is also very oriented for data analysis.

### 2.6.3 WEKA

WEKA was another tool that we employed. This allowed us to easily collect and organize text in all the Reddit comments to clean and understand our dataset. WEKA is also convenient for text data presentation.

### 2.6.4 Pandas Data Analysis Library

Some future tools we are looking to implement is Pandas Data Analysis Library. Pandas contains various Python specific tools for data analysis and data display with the convenience of easy cooperation between the Python scripts.

### 2.6.5 Database

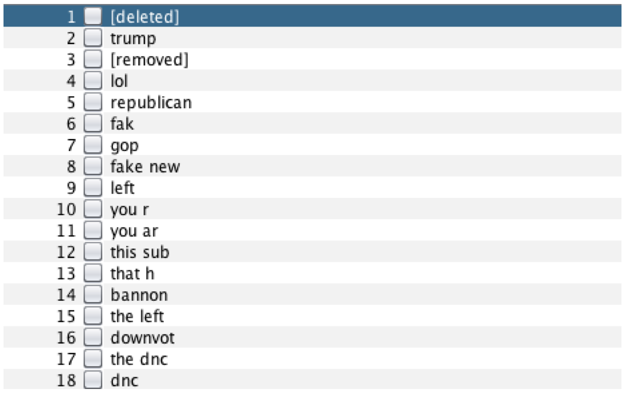
Another possible tool would be to insert the data set into a database or data warehouse such as MySQL or BigQuery. This would give us an easy, simple way to query our data for specific information.

## 2.7 Milestones

### 2.7.1 First Milestones

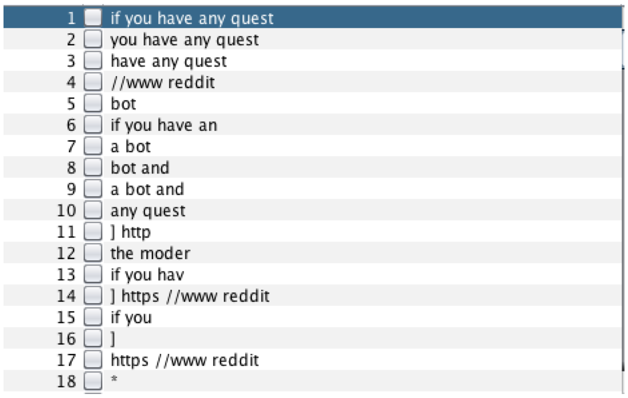
1. Data Integration and pre-processing
   reduce the JSON data to include only comments from the r/Politics sub-reddit. Transform the data to a mySQL or BigQuery database for analysis.

2. Process for the Data
   Information extraction of the new dataset. Create a corpus that contains the most common words and phrases used in comment bodies. Identify the most active users in the r/Politics sub-reddit based on the



1 □ [deleted]
2 □ trump
3 □ [removed]
4 □ lol
5 □ republican
6 □ fak
7 □ gop
8 □ fake new
9 □ left
10 □ you r
11 □ you ar
12 □ this sub
13 □ that h
14 □ bannon
15 □ the left
16 □ downvot
17 □ the dnc
18 □ dnc

**Figure 1: Splitting on comments that are suspicious**



1 □ if you have any quest
2 □ you have any quest
3 □ have any quest
4 □ //www reddit
5 □ bot
6 □ if you have an
7 □ a bot
8 □ bot and
9 □ a bot and
10 □ any quest
11 □ ] http
12 □ the moder
13 □ if you hav
14 □ ] https //www reddit
15 □ if you
16 □ ]
17 □ https //www reddit
18 □ *

**Figure 2: Splitting on comments that are suspicious**

number of comments and posts they aggregate. Identify the most popular comments based on the number of upvotes received. Search for frequently mentioned cities, and words associated with these cities.

3. Evaluation
Search for a correlation between most frequently used words and phrases and events in the month of January 2017. Use this and most popular comments to identify the most popular topics on the sub-reddit this months, and predict a collective user opinion about them. Map a geographic distribution of reddit comments based on content, by cross comparing with geographic location of mentioned cities, to identify a correlation between comment topic and location or demographic in the United States. Identify significant interactions between sub-reddits and additional user characteristics, by tracking the activity of the most active users in the r/Politics sub-reddit to other sub-reddits. Potential visualization of data to help better understand results.

### 2.7.2 *Milestones Achieved*

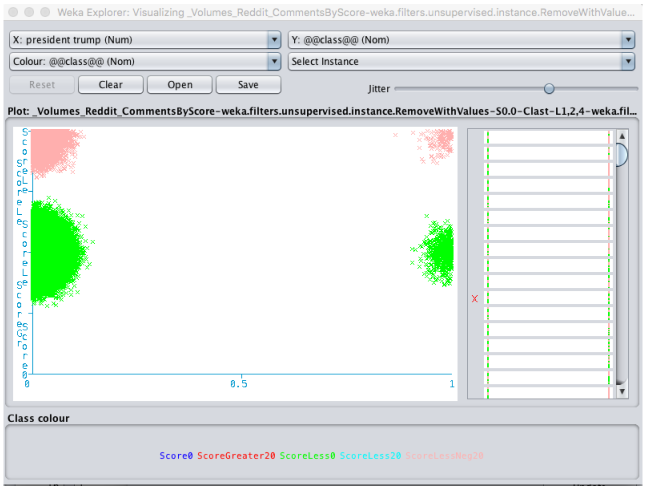
1. Data Pre-processing and Cleaning
Using a simple Python script and the JSON library, we managed to reduce our dataset to a sample size of nearly 100,000. The script failed to return results when ran through the entire dataset for all comments in the sub-reddit r/Politics. This may be due because of the large size of the data or the memory limitations. We overcame this problem by only inputting half the dataset during the parsing. A few other problems arise from this however. We must make sure the data is complete and does not lose its JSON formatting.

2. Data Processing
We utilized WEKA to text analyze the dataset and extract useful information. Figure 1 shows how we were able to extract the top words used in the r\Politics sub-reddit comments. The results were as expected considering the events that occurred in January 2017. Political words relating to the US presidency were among the top. Words such as "trump" and variations of "fake news" were all among the top words. Additionally, Reddit keywords "[deleted]" and "[removed]", which were irrelevant, were also among the frequent words used. These were easily removed however; WEKA has convenient features to remove such stem and stop words. Next we used WEKA to analyze the comments and score relationship. Image BLANK displays. It is split up by score and shows comments with a score between 1 to less than 20. Most comments score were among this range so we used this to show the comments with fringe amount of votes and compared them to each other. Using simple Python script, we were able to search the top authors in the r/Politics sub-reddit. Our results had many errors due to Reddit bots.

3. Evaluation
Search for a correlation between most frequently used words and phrases and events in the month of January 2017. Use this and most popular comments to identify the most popular topics on the sub-reddit this months, and predict a collective user opinion about them. Map a geographic distribution of reddit comments based on content, by cross comparing with geographic location of mentioned cities, to identify a correlation between comment topic and location or demographic in the United States. Identify significant interactions between sub-reddits and additional user characteristics, by tracking the activity of the most active users in the r/Politics sub-reddit to other sub-reddits. Potential visualization of data to help better understand results.

### 2.7.3 *Future Plans*

1. Data Cleaning
We plan to test a couple methods to correct for the bots that are skewing our data. First we plan to limit only to people who comment less than a realistic threshold. It would be difficult for users to comment more than 5,000 comments in a month. By doing this we could remove various irrelevant comments but, on the other hand, we could possibly remove good data as well.
Another possible plan is to remove those comments by hand using the stem words that WEKA retrieves for us which we deem useless. Figure 2 above is just an example of the text that we split on to remove such comments. Using this information we could manually remove certain comments that contain such patterned text. This may be really challenging however, due to


Weka Explorer: Visualizing _Volumes_Reddit_CommentsByScore-weka.filters.unsupervised.instance.RemoveWithValue...

X: president trump (Num)

Y: @@class@@ (Nom)

Colour: @@class@@ (Nom)

Plot: _Volumes_Reddit_CommentsByScore-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L1,2,4-weka.fil...

**Figure 3: Score When "president trump" was used**

the amount of data that wish to remove.

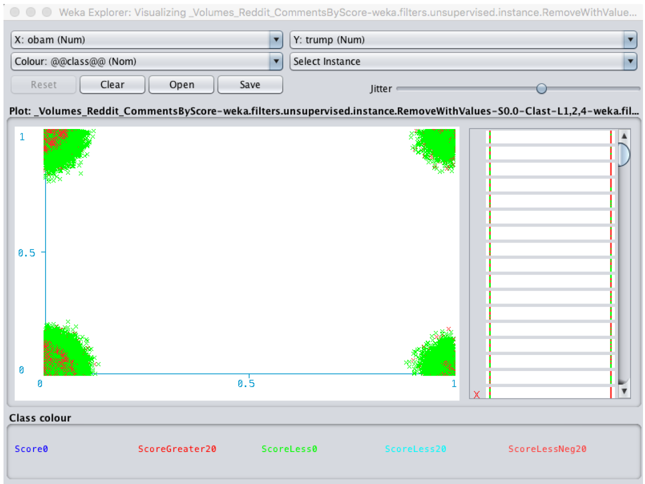
2. Data Storage and Display
   We plan to implement Pandas Data Analysis Library. This allows for easy, fast data manipulation. Pandas prides itself on being one of the quickest data Using this we can show correlation through the easy to implement graphs and charts. According to their website [1], Pandas handles the "majority of typical use cases in finance, statistics, social science, and many areas of engineering." First we must process all the data into a database. We plan on using MySQL since it's free and easy to use.

3. Further information One member of our group reached out to the r\Politics moderators in order to gain further research about how they handle unwanted comments or automated Reddit accounts. We will be interviewing one moderator in order to gain a better understanding of how the moderators deal with them. We hope this may help find a better way to clean our data.

## 3. CONCLUSIONS

Using WEKA we managed to see some interesting relationships in the r\Politics subreddit. Firstly, we looked at commonly used stem words and compared the score when they were used. Figure 3 above shows the relationship of the stem word "trump" and Reddit score. Because most comments reside between a score of zero to ten, we decided to only look at those comments scored below zero and above 20. The data shows that a negative score is more prevalent when "trump" or variations of that word is used in a comment.

WEKA also allowed us to compare the score when two objects were used. Again for in order to see some In figure 4 above, we looked at comments when "trump" and "obama" were used. As you can see when either of these stem words were used, a score less than zero could be predicted. However, looking at the graph clearly shows that comments were scored less than a zero more frequently when "obama" was mentioned than "trump".


Weka Explorer: Visualizing _Volumes_Reddit_CommentsByScore-weka.filters.unsupervised.instance.RemoveWithValue...

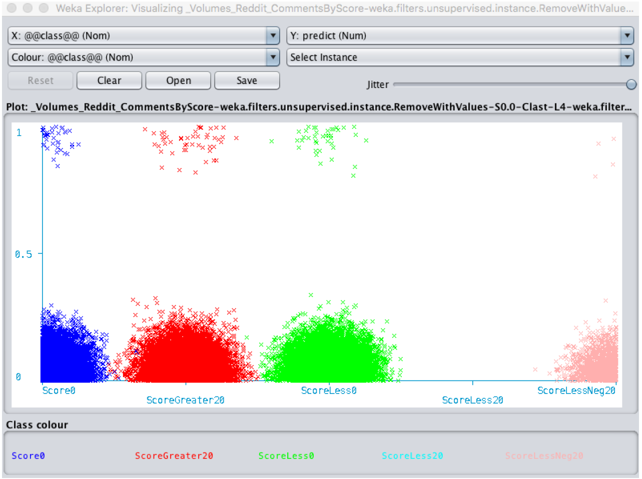
X: obam (Num)

Y: trump (Num)

Colour: @@class@@ (Nom)

Plot: _Volumes_Reddit_CommentsByScore-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L1,2,4-weka.fil...

**Figure 4: Score When "trump" and "obama" was used**


Weka Explorer: Visualizing _Volumes_Reddit_CommentsByScore-weka.filters.unsupervised.instance.RemoveWithValue...

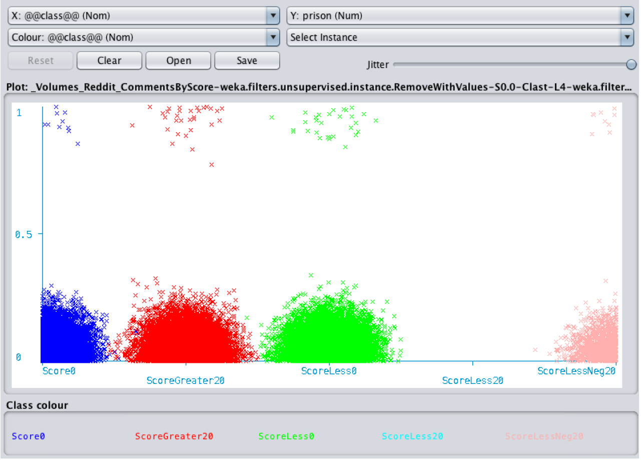
X: @@class@@ (Nom)

Y: predict (Num)

Colour: @@class@@ (Nom)

Plot: _Volumes_Reddit_CommentsByScore-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L4-weka.filter...

**Figure 5: Score When "predict" was used**


X: @@class@@ (Nom)

Y: prison (Num)

Colour: @@class@@ (Nom)

Plot: _Volumes_Reddit_CommentsByScore-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-Clast-L4-weka.filter...

**Figure 6: Score When "prison" was used**

We can look at specific non political words as well. Figure 5 and figure 6 show the scores of comments based on usage. A comment shows that the word was used if it is marked at the vertical point one; a zero if it does not appear in the comment. As can be observed in the graphs, “predict” seems to be most active in those comments with a score greater than twenty, while “prison” is a little more difficult to differentiate.

Through our mining of Reddit, we hope to gain a more complete view of political sentiment. This will be accomplished using data mining techniques and text analysis. Reddit data aligns perfectly with this as we can search through the political sub-reddits and have plenty of data to find meaningful results.

We plan to use these political sub-reddits to determine whether we can gain any useful knowledge about a user’s political ideas and approximate location based on what they post on Reddit. Reddit contains a diverse group of users from different political groups. This results in a large dataset that could be used later with future comment data from future elections and political events to determine whether we could better predict the outcome using this analysis.

## 4. REFERENCES

[1] pandas-docs, http://pandas.pydata.org/pandas-docs/stable/.

[2] Reddit january 2017 comments now available via torrent r/datasets, https://www.reddit.com/r/datasets/comments/5uftxe/reddit_january_2017_comments_now_available_via/.

[3] A. Barbaresi. Collection, Description, and Visualization of the German Reddit Corpus. In G. S. for Computational Linguistics & Language Technology, editor, *2nd Workshop on Natural Language Processing for Computer-Mediated Communication*, Proceedings of the 2nd Workshop on Natural Language Processing for Computer-Mediated Communication / Social Media, pages 7–11, Essen, Germany, Sept. 2015.

[4] M. De Choudhury and S. De. Mental health discourse on reddit: Self-disclosure, social support, and anonymity, http://ai2-s2-pdfs.s3.amazonaws.com/2db7/15a479c8961d3020fe906f7bedfa0311b937.pdf, 2014.

[5] E. Gao. Analyzing trump v. clinton text data at reddit, Jan 2017.

[6] T. Weninger. An exploration of submissions and discussions in social news: mining collective intelligence of reddit, Feb 2014.