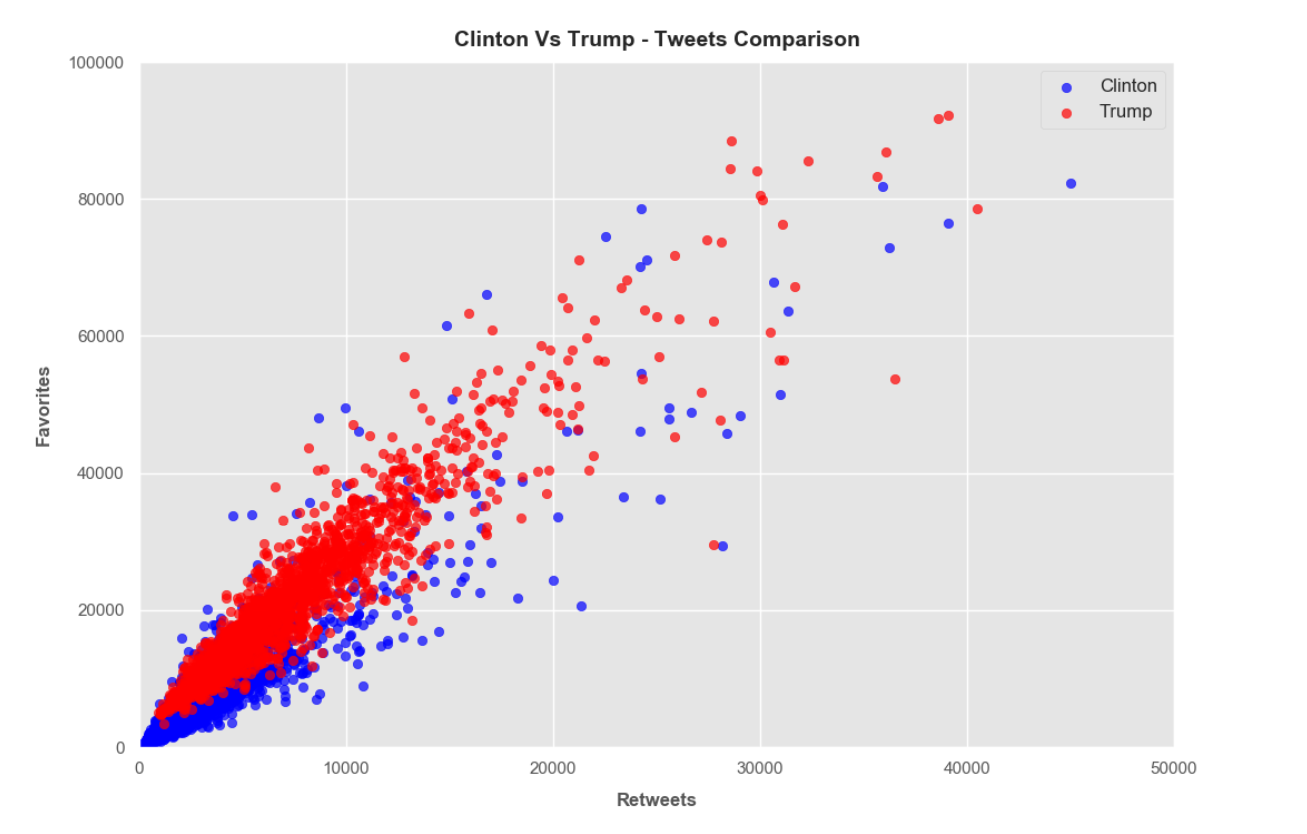
Today, November 8th 2019, exactly three years have passed since the 2016 US presidential election. At this hour back then, the people of the United States exercised their right to elect their president in what turned out to be the most polarizing US election in history. Had I done this twitter posting analysis of Trump and Clinton three years ago before the election, I could have predicted, with significant confidence, that Donald Trump would emerge as the winner.

The figure **17\_ScatterRetweetAndFavorite.png,** shows the comparison of retweets and favorites of the two candidates, as visualized in the scatter chart model.

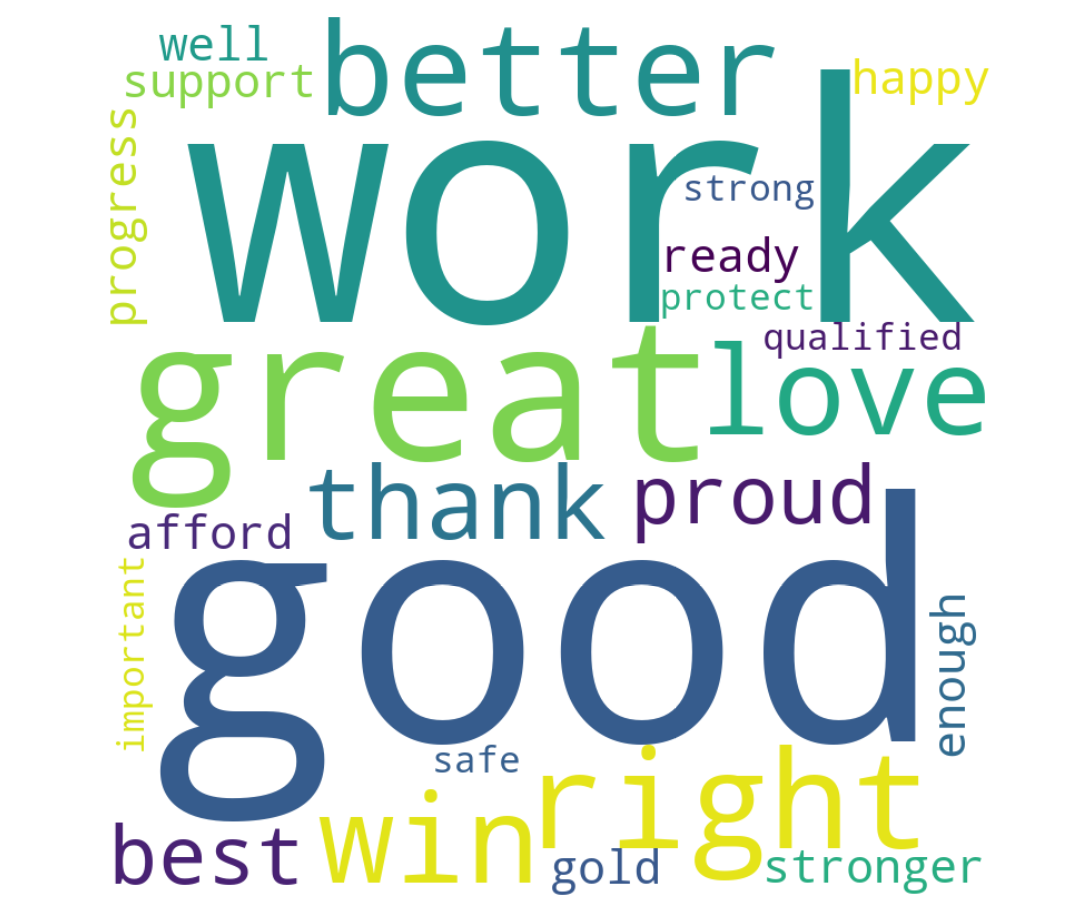
**Fig. 17\_ScatterRetweetAndFavorite.png**

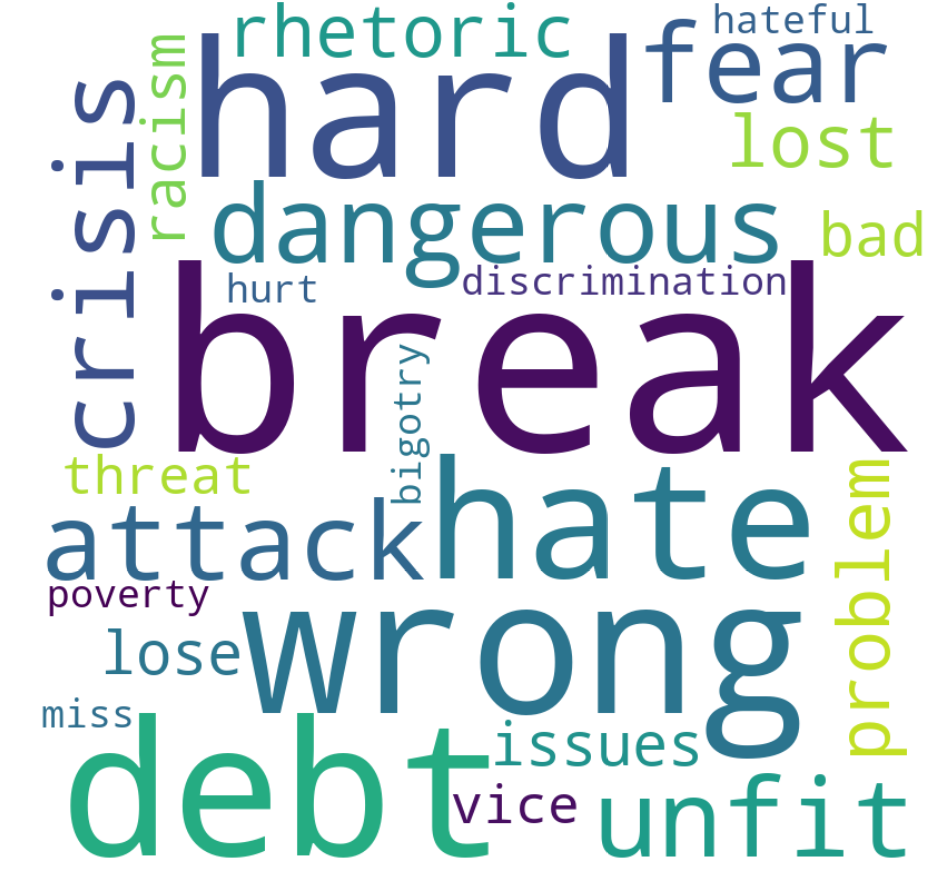


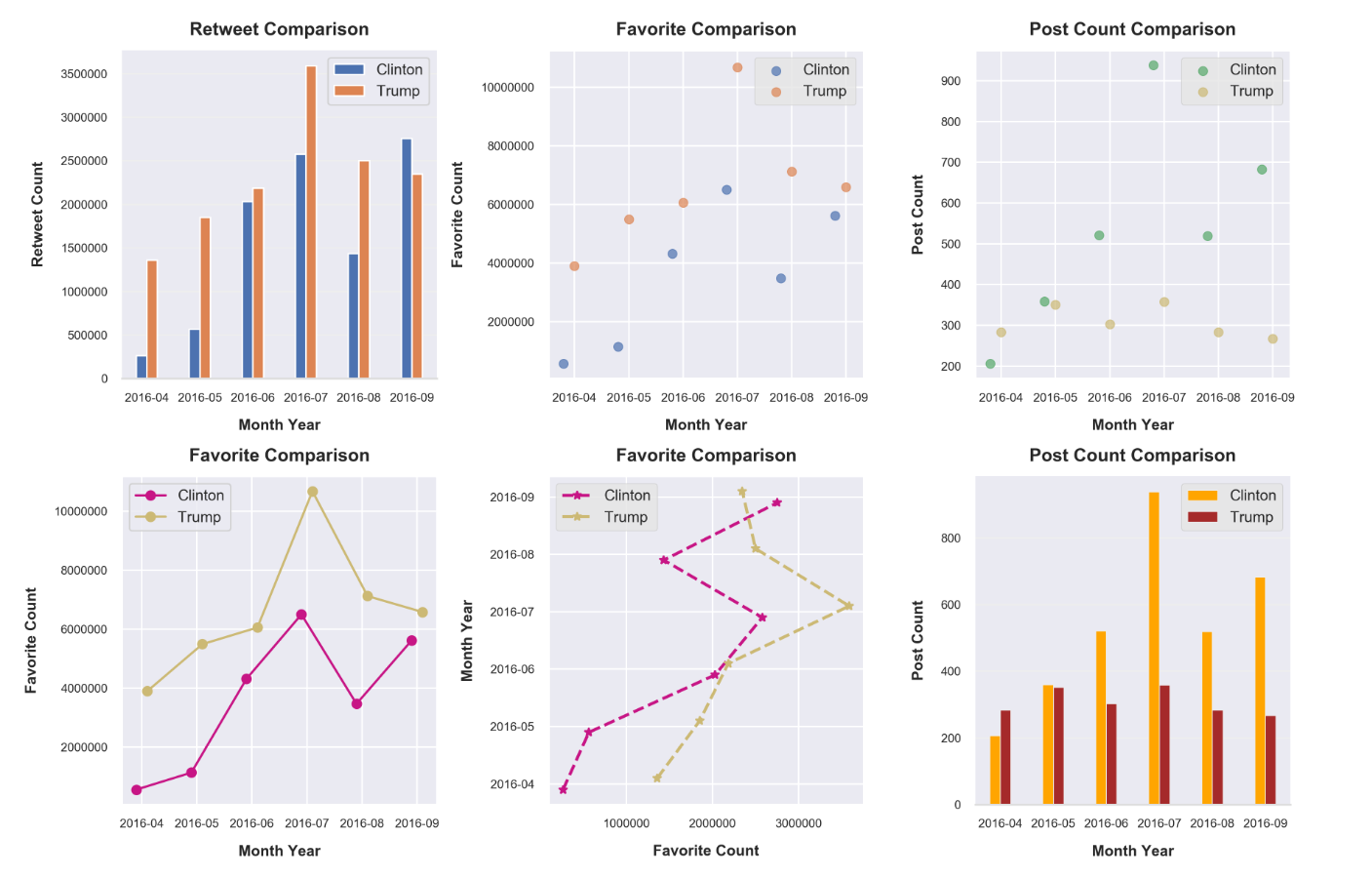
On this chart, one can see the enthusiasm of the Trump supporters in retweeting and liking his posts. Comparing his results to Hillary, the difference can easily be seen, with the scatter chart showing the “big league” discrepancy.

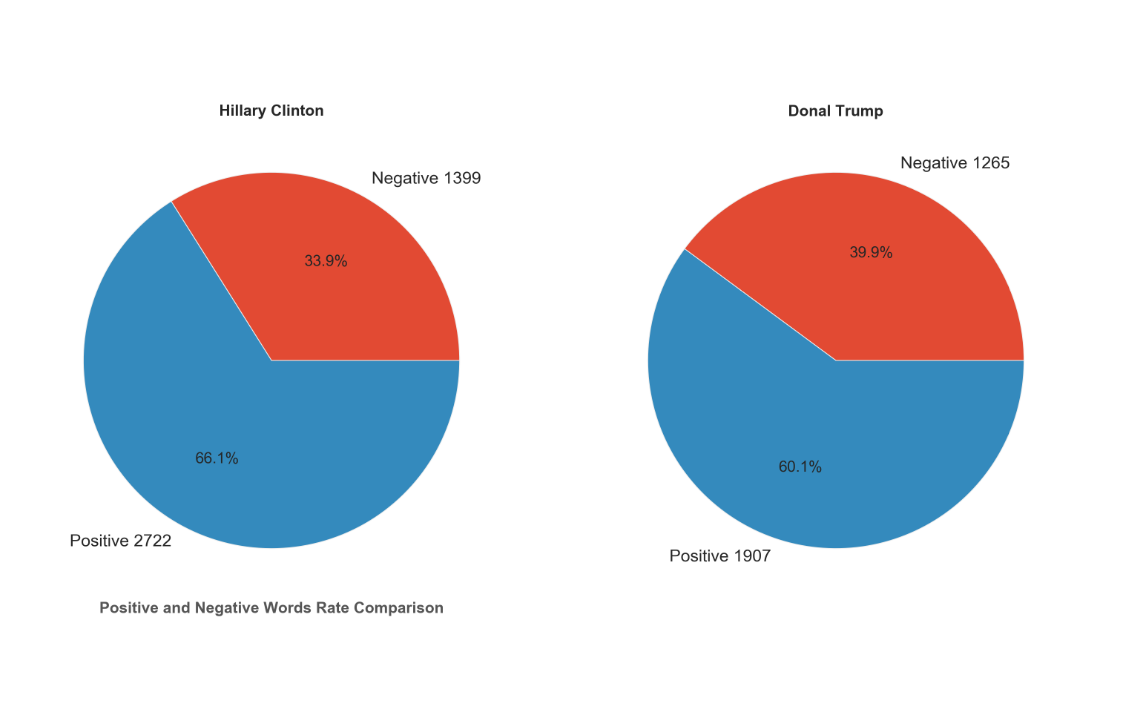
The other charts also demonstrated an easy read of the numbers that indicate that Trump had an underlying advantage. It was also interesting to see the two candidate’s use of certain words, positive or negative, and how they resonated with their respective followers.

**Hillary Clinton Positive Words and Negative Words**

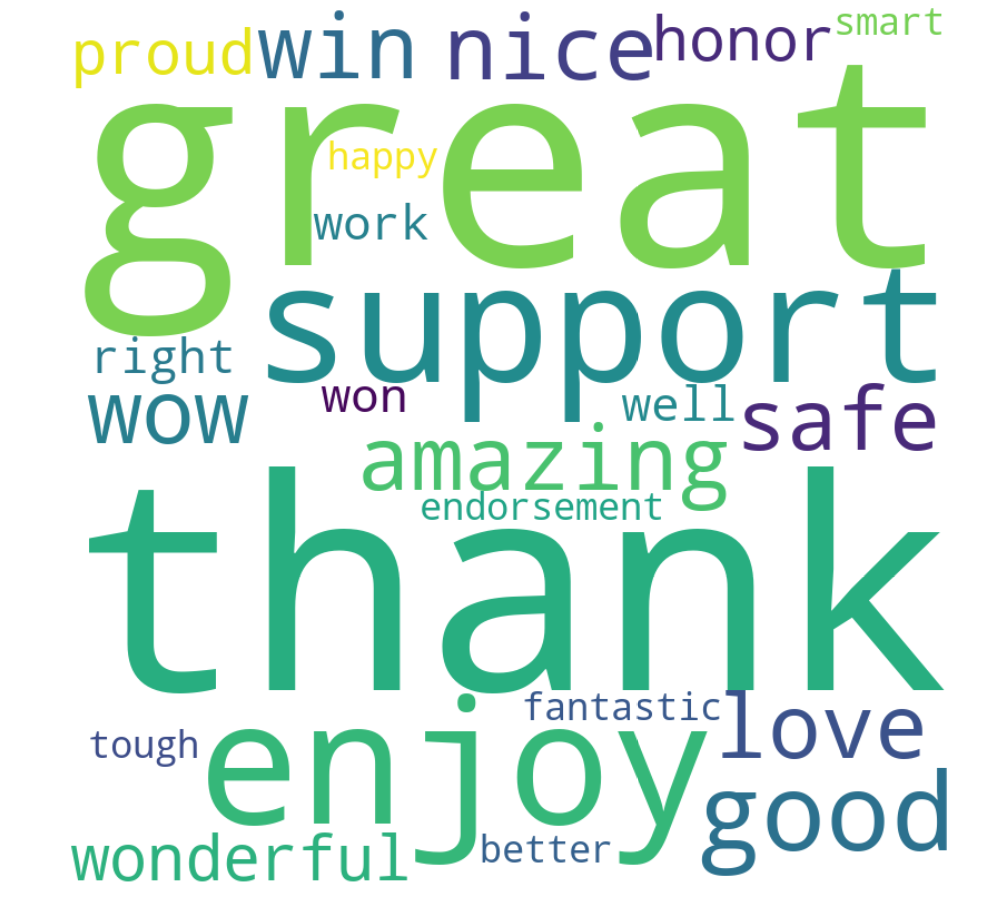




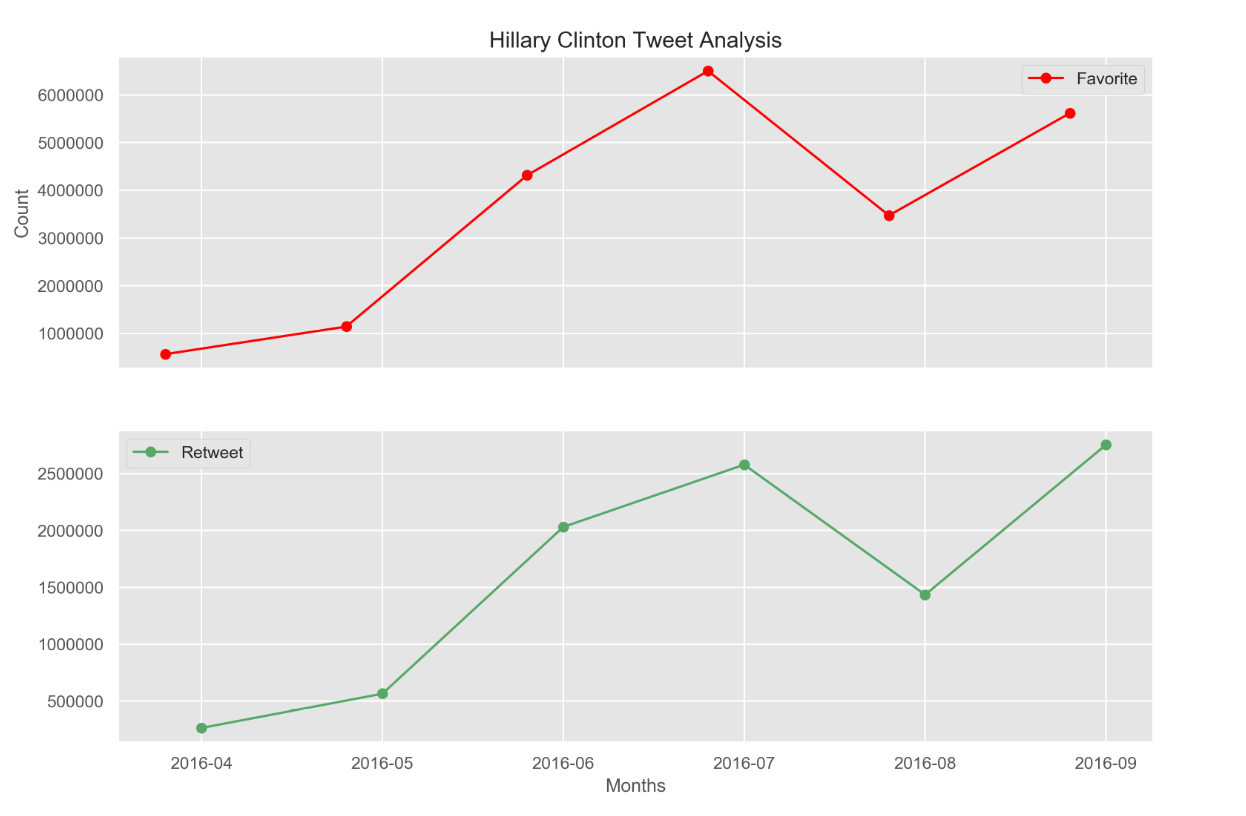


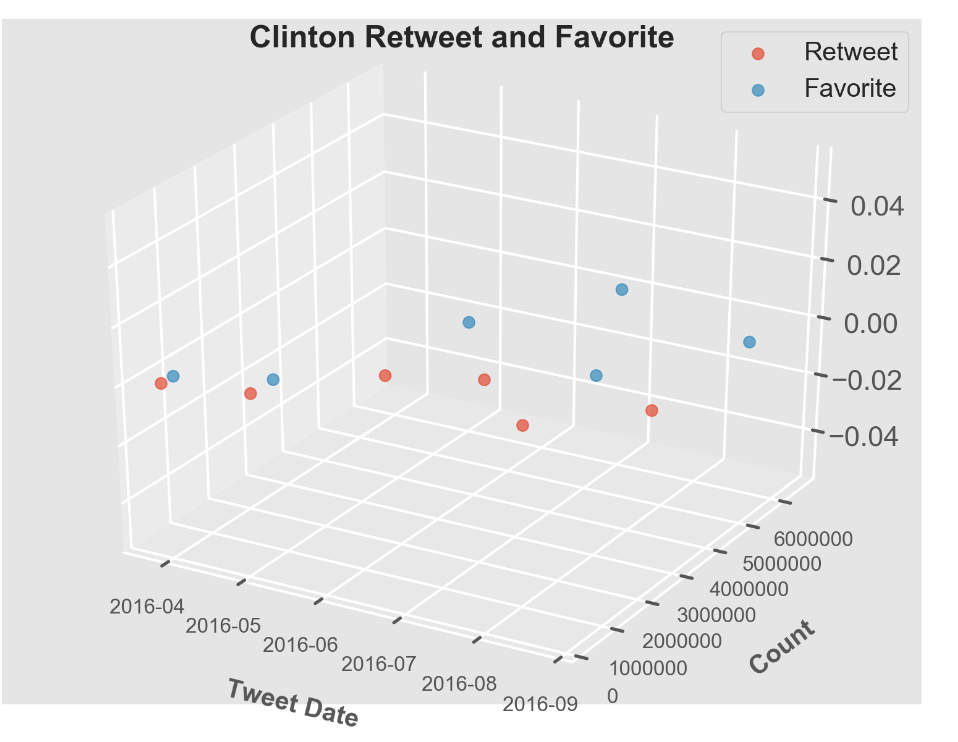


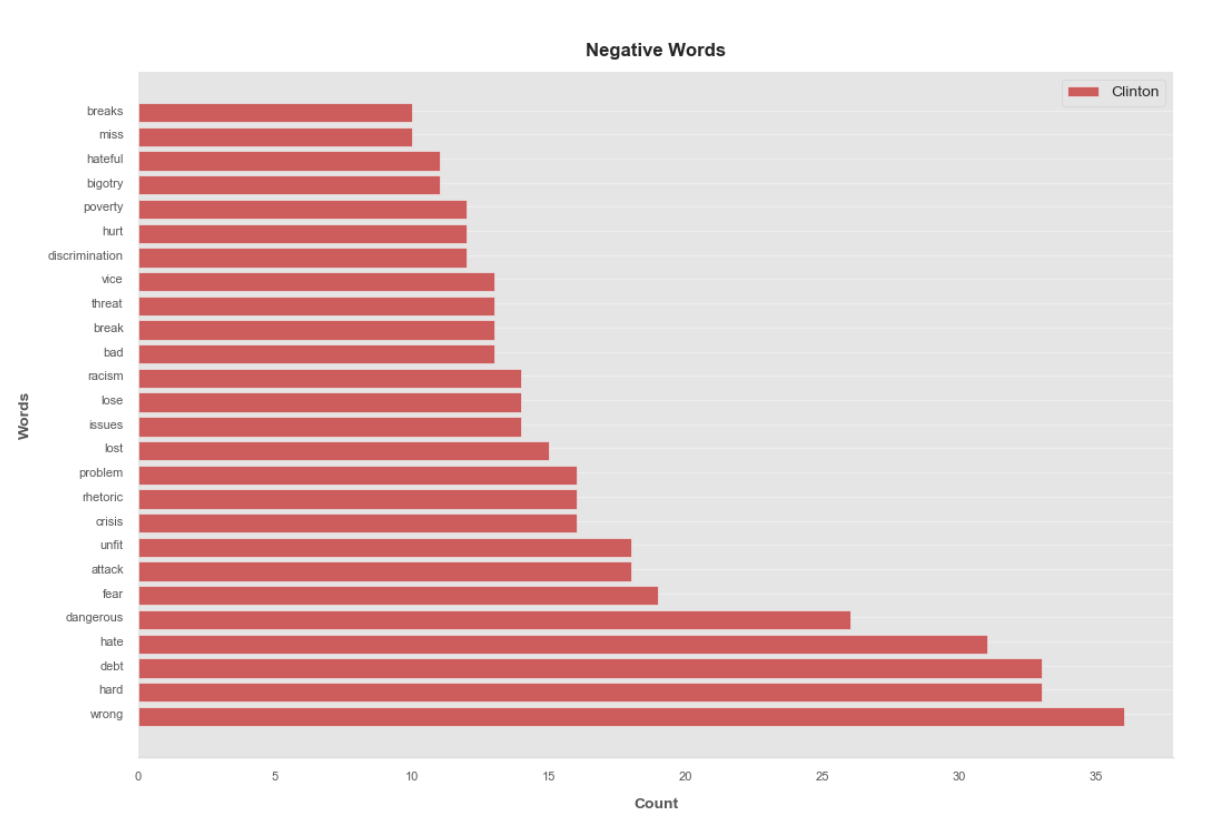
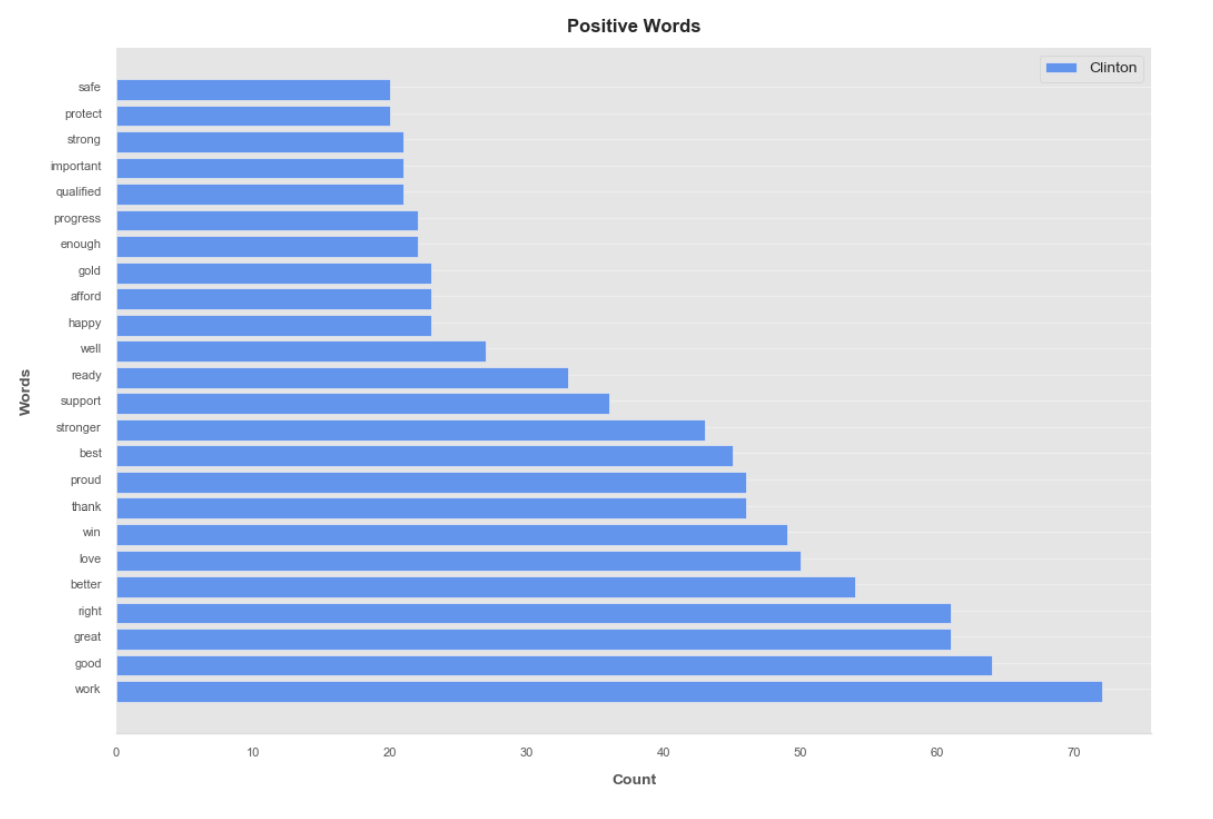
**Donald Trump Positive & Negative Words**

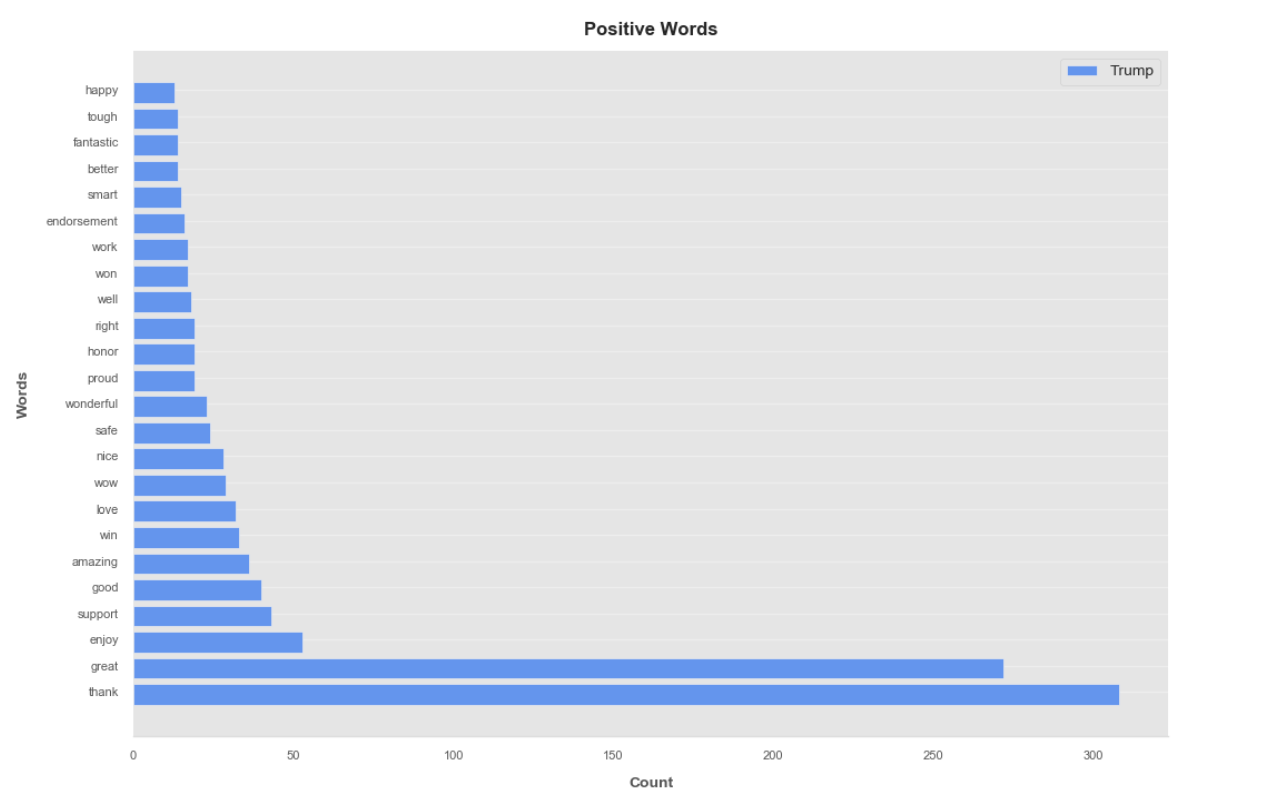


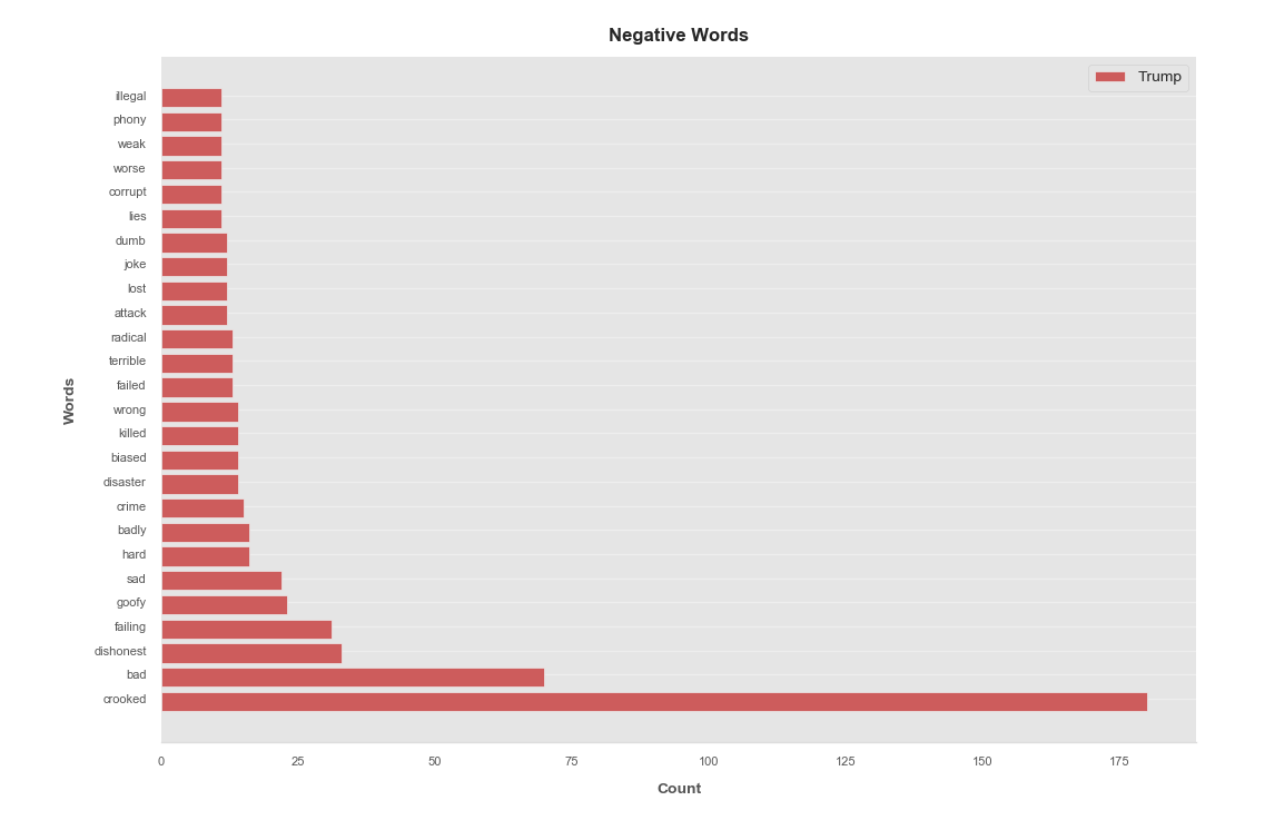












**Clinton Retweet Analysis – Logistic Regression**

**Summary of f1-score:**

|  |  |
| --- | --- |
| **Test Size** | **F1-score** |
| 0.25 | Between 0.66 to 0.69 |
| 0.35 | between 0.67 to 0.71 |
| 0.45 | Between 0.66 to 0.69 |

|  |
| --- |
| I have classified my retweet count as follows:   * Class 1 (1-1000) * Class 2 (1001-2000) * Class 3 (2001-5000) * Class 4 – above 5000   My Features are the following:   * Favorite Count, Hashtag Count, Mention Count, Weekday, Hour, URL Count and Session |
| **Below is at 0.35 Test Size**  **Accuracy score is 0.69**/1  Classification Report  precision recall f1-score support  1 0.74 0.75 0.74 271 ---total in class = 774  2 0.62 0.62 0.62 381 ----total in class = 1089  3 0.70 0.70 0.70 348 --- total in class = 994  4 0.81 0.77 0.79 130 --- total in class = 371  accuracy 0.69 1130  macro avg 0.72 0.71 0.71 1130  weighted avg 0.70 0.69 0.69 1130  Confusion Matrix  [[202 68 1 0]  [ 68 238 75 0]  [ 4 75 245 24]  [ 0 0 30 100]]  Overall f1-score  0.7135898312746344  This is consistently giving me and f1-score between 0.67 to 0.71 |

**Using KNeighbors**

Using this code below and by changing the n\_neighbors=1 to up t o 5, I noticed that the accuracy value changes:

**Clinton Retweet Analysis – KneighborClassifier**

Performing the KNeighborClassifier using the same dataset, I got the following results:

|  |  |
| --- | --- |
| **Test Size** | **Accuracy** |
| 0.25 | 0.7447335811648079 |
| 0.35 | 0.7053097345132744 |
| 0.45 | 0.7017906336088154 |

Using this code below and by changing the n\_neighbors=1 to up t o 5, I noticed that the accuracy value changes:

clf = neighbors.KNeighborsClassifier(n\_neighbors=1)

clf.fit(X\_train, y\_train)

accuracy = clf.score(X\_test, y\_test)

print(accuracy)

|  |  |
| --- | --- |
| N\_Neighbor = | 0.35 percent test size |
| Without | 0.7061946902654868 |
| 1 | 0.6707964601769911 |
| 2 | 0.6584070796460177 |
| 3 | 0.6973451327433628 |
| 4 | 0.7106194690265487 |
| 5 | 0.7230088495575221 |

For this test, I consider that the logistic regression is a good fit for my dataset since I would like to predict the level of retweet being as the classification of a tweet using features like “Favorite Count, Hashtag Count, Mention Count, Week Day, Hour and URL count”. One challenge though with what I have done is how to set up the correct classification. In this case, I used the retweet counts.