

Predicting the US Primary Election Results

Aditi Singh (110285096), Arjun Bora (110310134), Yunke Tian (109929662), Yinquan Hao (107989208.)

Motivation:

U.S. Presidential elections are not just important for the U.S. nationals, but also to the nationals of other countries. It becomes more important when the candidates have extreme ideas, strategies towards some particular issue, ethnicity and policy, like foreign policies.

This time it becomes even more exciting because of the fact that, if Hillary Clinton becomes the president, she would be the first women president; if Sanders become the president, he would be the first Jewish president and if Trump wins, he might be the first industrialist to become the president.

There can be lot of factors, like age, income, ethnicity, state, education level, sex that can affect the voting behavior of people. The more predictors we consider, the prediction becomes more accurate and challenging.

As the primaries elections are still going on, we have applied our model to the ongoing elections and verify our model. Thus, this is an interesting, easily verifiable on real results and this is why we were motivated to choose this project.

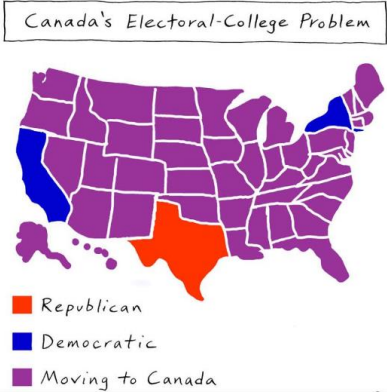
Approach:

Data Collection and Process:

We are collecting data from many different sources:

1. **Twitter API** – to get the latest tweets, using Python to download the twitter feeds using hash tags.

Task	Code	Challenge	Solution
Tweets collected for the US primary elections	<ul style="list-style-type: none">• Python code is written to find out the tweets collected over a period of time for a particular hash-tag value / keyword	<ul style="list-style-type: none">• Twitter API has a limitation for the # of tweets collected per API count	<ul style="list-style-type: none">• Ran the script multiple times as a foreground process. Found out a twitter archiver tool which collects the tweets for a given hash-tag and creates a backup to a Google drive account.
Cleaning up and categorizing tweets	<ul style="list-style-type: none">• Code was written to bucket the tweets based on the hash-tags and also the keyword search within the text	<ul style="list-style-type: none">• Multiple hash-tags + keywords present in the tweet. E.g. the user might use multiple hash-tags like #Clinton and #trump in the same tweet.• Multiple keywords exist in the tweet like e.g. Donald trump and also Hillary are spoken about in the same tweet	<ul style="list-style-type: none">• The tweet# would occur both in the trump and the Clinton dictionaries created.• If more than one keywords from this list: [Hillary Clinton, Bernie Sanders, Ted Cruz, Donald trump] have been found, then the importance of the score of the tweets would be shared by these keywords.• If more than one hash-tag has been found then the tweet value would be fully allocated to all hash-tags.• This is because hash-tags used more deliberately to point to a particular person.
Location mapping of tweets	<ul style="list-style-type: none">• Wrote a python script which scraped user information from the profile and collected the location information.• Wrote a python script to match the location using fuzzy logic library	<ul style="list-style-type: none">• Twitter has disabled geo-location information collection per tweet• Location script can access only 180 API calls to profile per call.• Fuzzy logic code can find the approximate match to a county and not the actual county mapping	<ul style="list-style-type: none">• Since people did not conform to any format for location entry, we could not map it correctly to a specific county• Found out the match using the max score based on the similarity matching using edit distance.

Sentiment Score	<ul style="list-style-type: none"> The NLTK library is a famous library which is used to find out the sentiment of a tweet. The code calculates the compound score of the tweet and then multiplies it with the number of re-tweets it has. 	<ul style="list-style-type: none"> Sentiment score is a very subjective score. Does not capture sarcasm, and subtlety. Many words have negative/positive connotations, but they are not found in the lexicographic dictionaries used by the classifier. 	<ul style="list-style-type: none"> Python script calculates the total sentiment score of the tweet by categorizing it based on the candidate and also by county. The sentiment score is weighted using the number of re-tweets it has, based on the number of '@' characters the corresponding tweet has. This would add weights to the tweets and good tweets (the ones which are most re-tweeted) get more weightage in the final output than the others
Popularity score	<ul style="list-style-type: none"> The code categorizes the tweet count based on the county Secondly to use weighted scores, we multiply the count with the #followers for each user. 	 <p>Canada's Electoral-College Problem</p> <p> ■ Republican ■ Democratic ■ Moving to Canada </p>	<ul style="list-style-type: none"> The tweets would be weighted by the person's #follower. This would guarantee that genuine tweets only make through.
Cumulative State-wise votes/score	<ul style="list-style-type: none"> Code is written which aggregates the scores/ votes whenever the argument of the candidate name is provided. 		

2. Kaggle datasets – 2016 US election dataset (US primaries data).

Task	Code
Standardization of data set	<ul style="list-style-type: none"> The X values – factors (all attributes from the demographics data file + sentiment score + popularity scores) are standardized.
Dimensionality Reduction using PCA with L1/ L2	<ul style="list-style-type: none"> There are more than 50 dimensions which are being considered. Thus using PCA we are reducing the number of dimensions to such a value which gives us the minimum MSE value.
Prediction using Multiple Linear Regression Model + 10 fold Cross Validation	<ul style="list-style-type: none"> Since the data is split county wise, we had decided to make 4 Multiple linear regression model. One for each candidate: Bernie Sanders, Hillary Clinton from the Democratic Party and Donald Trump, Ted Cruz from the Republican Party. Secondly, our script also includes 10 fold cross validation. This would create a test/training data set and thus would avoid data over-fitting. The beta values for the MR model is chosen based on the lowest value of the MSE.
Accuracy of Prediction	<ul style="list-style-type: none"> The (predicted value – the actual value of the final votes)/ (actual votes) all aggregated state-wise gives the accuracy. Challenge: There are many other factors which affect the final vote count. Since this is the primaries, different rules exist in different states regarding voting. E.g.
Final Election Results Prediction	<ul style="list-style-type: none"> The final results are shown using the electoral votes collected by the winning member of the respective party. Finally the candidate who wins the majority of the electoral votes wins.

Overview:

There are many factors that could influence the popularity of a candidate and the aim is to find out if there is some correlation between the demographic distribution of the country (age/education level/ race & ethnicity / income etc) and

the candidate popularity, and if yes, then include these dimensions to come up with a multiple linear regression model to predict the final number of votes which would be acquired by the contesting candidates.

Factors: The below mentioned factors would be used to predict the votes in the primary for each candidate.

Data Source	Attribute	Reason
Twitter	Popularity of Candidates = (#tweets) x (weight of individual tweet)	The tweets would be weighted by the person's #follower. This would guarantee that genuine tweets only make through.
Twitter	Weighted Sentiment score based on the #retweets	This would add weights to the tweets and good tweets (the ones which are most retweeted) get more weightage in the final output than the others
Kaggle (2016 US Elections)	County wise demographics	There are several factors which characterize a particular county. We would be running PCA on the set of factors to get some dimensions which characterize the counties best which have the minimum MSE values. These final dimensions would be used as factors in the multiple-linear regression model.

Further Analysis:

After getting the result for primaries, we want to predict the final election results.

Three Broad Categories: *Apart from the first phase of Probability/Discovery/Prediction for the primary elections we would also extend the model to incorporate the below mentioned points*

Probability: Based on this data, we calculate the number of free voters, who only voted in the final elections and not in the primary elections. We find the probability of a free voter to vote for the republican candidate and to vote for democratic candidate. Now using this probability and the vote count in primaries, we count the final election votes for both the candidates for every state.

Discovery: We find correlation between some of the factors and the final vote count of the winner of the primaries in a particular county.

Prediction: Applying this probability to our predicted primaries result, we can predict the number of votes for the final election, and thus predict Who Will be the president of the U.S.A. for 2016-2020.

Predicting Final Election results from primary results:

Our dataset is primary and final election result of 2008 elections. We chose the latest election result so that other political situations be nearly same to the current political situations. We found 2012 data not suitable, because at that time an incumbent president and this fact supported him and made the situation different than current situation.

Dataset primary_tofinal.csv contains 4 columns, # votes Obama won in primary, # votes all other democratic candidates won in primary, # votes McCain won in primary, # votes all other republican candidates won in primary, # votes Obama won in finals, # votes McCain won in finals.

Based on this data, we calculate the number of free voters, who only voted in the final elections and not in the primary elections. We find the probability of a free voter to vote for the republican candidate and to vote for democratic candidate. We store this probability for every state in a dict 'PROB_DEM'.

Now using this probability and the vote count in primaries, we count the final election votes for both the candidates for every state. We give all the electoral votes for a state to the candidate whose predicted votes are higher. The one with more number of electoral votes is our predicted candidate for the US Presidential Elections 2016.

Sources of errors:

We learnt later that US presidential elections are little more complicated than we thought. In some states, voting for primaries just do not take place, and party chooses it's candidate by 'Caucus'. Then there are free voters, and even different states have different type of free voters (close, open, semi close, semi open). These points affect the probability PROB_DEM used in Section II.

Secondly, the total number of votes won in each state does not determine the final win. It depends on the electoral votes (which we have incorporated) and the final delegates which each candidate can get. The delegates who decide to back-up the candidates may do so based on their wishes.

Results Figure:

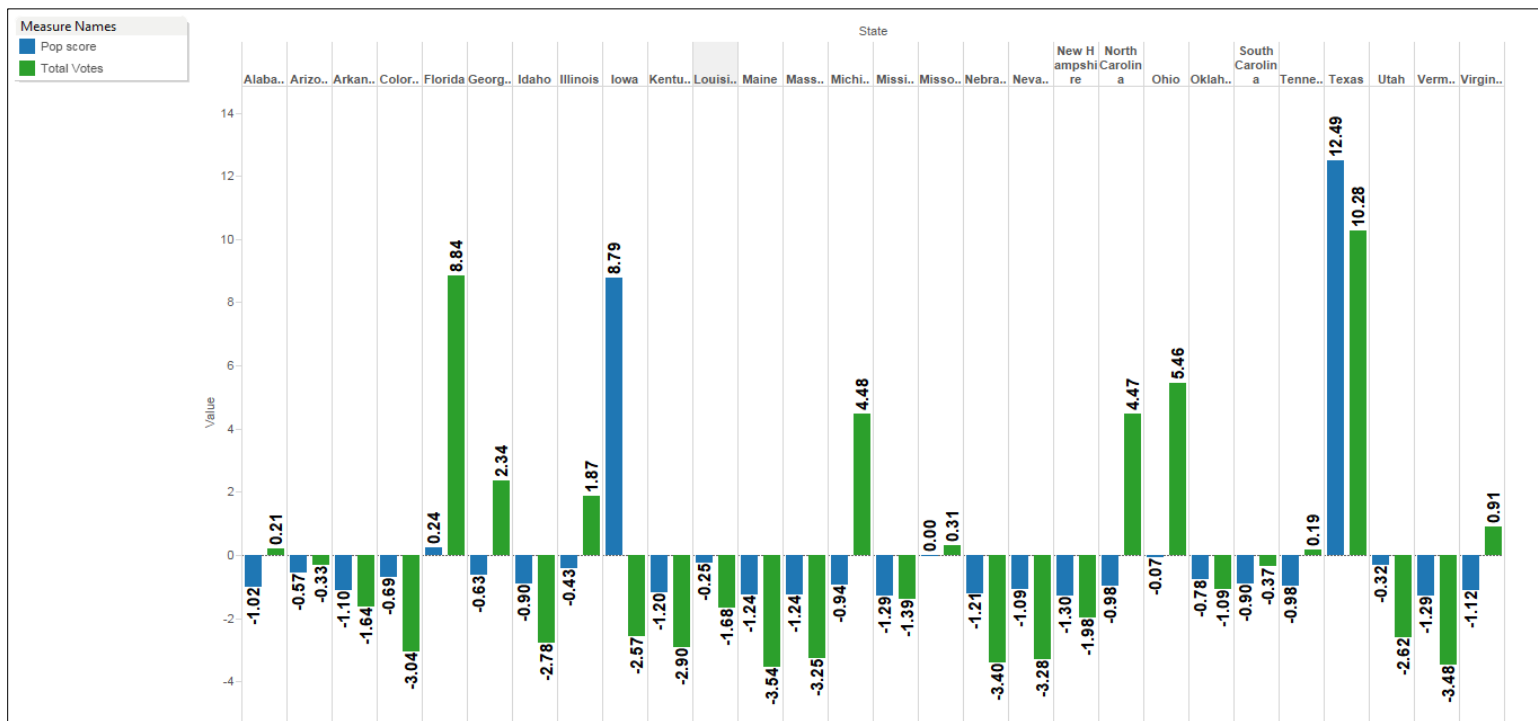


Fig1: Popularity Score from twitter (blue) vs Total Votes in actual primaries (kaggle) (green) over all states

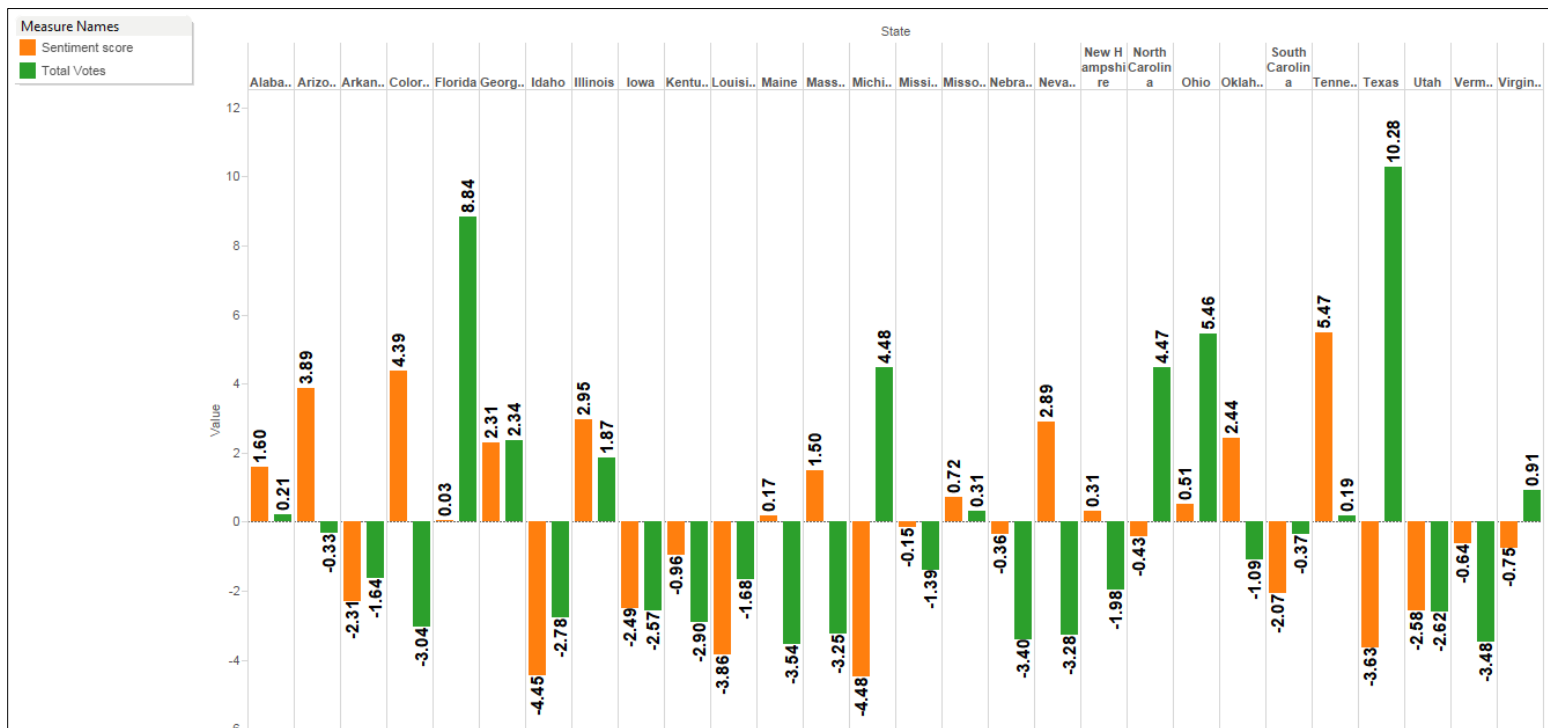


Fig2: Sentiment Score from twitter (orange) vs Total Votes in actual primaries (kaggle) (green) over all states

Calculating MSEs for candidates[Calculating the best components & then the MSE comes from the Regression model]:

Donald Trump	Ted Cruz	Hillary Clinton	Bernie Sanders
Best components: 47	Best components: 36	Best components: 51	Best components: 47
Best Test MSE:0.158267	Best Test MSE:0.158984	Best Test MSE:0.120325	Best Test MSE:0.242928

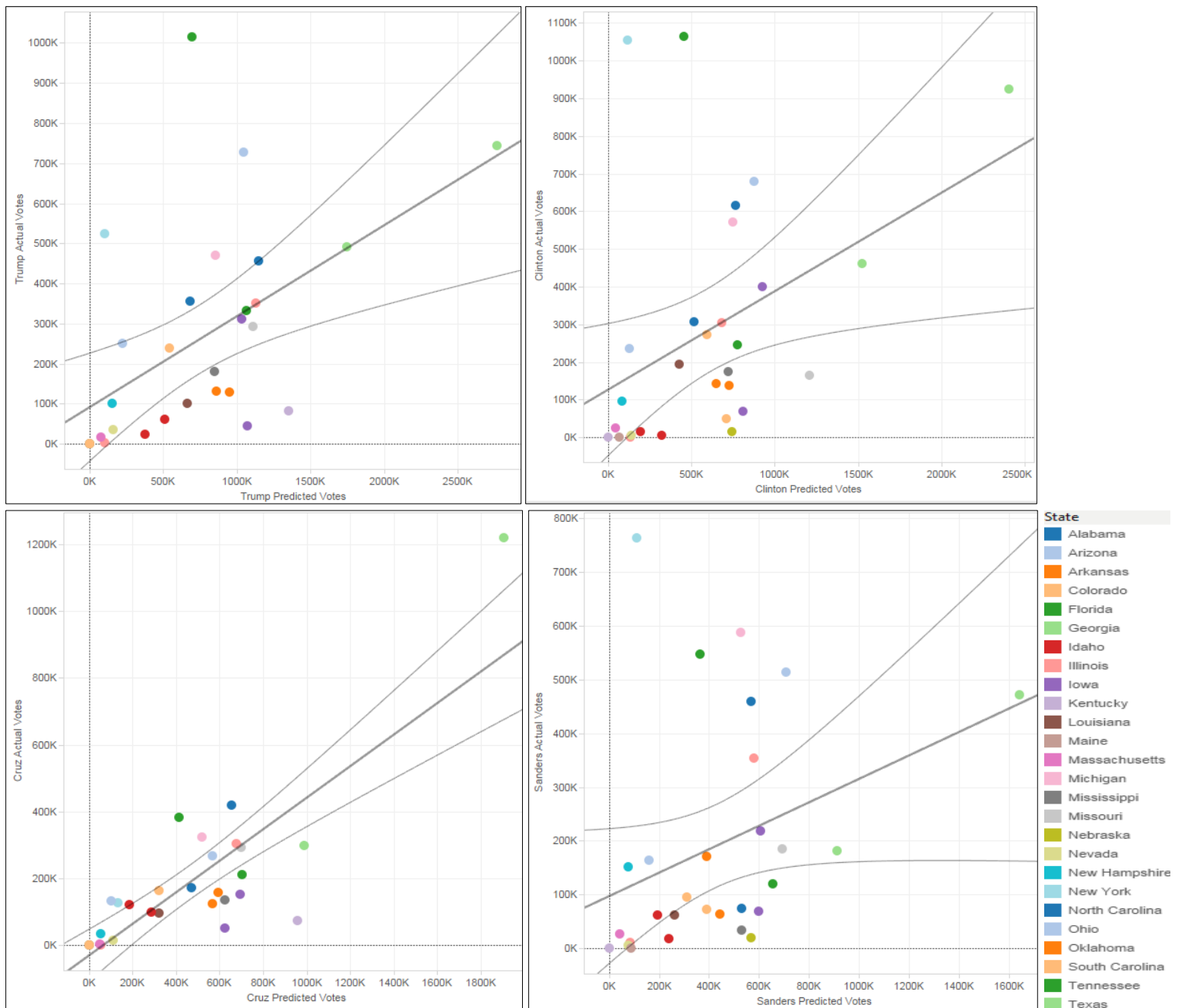
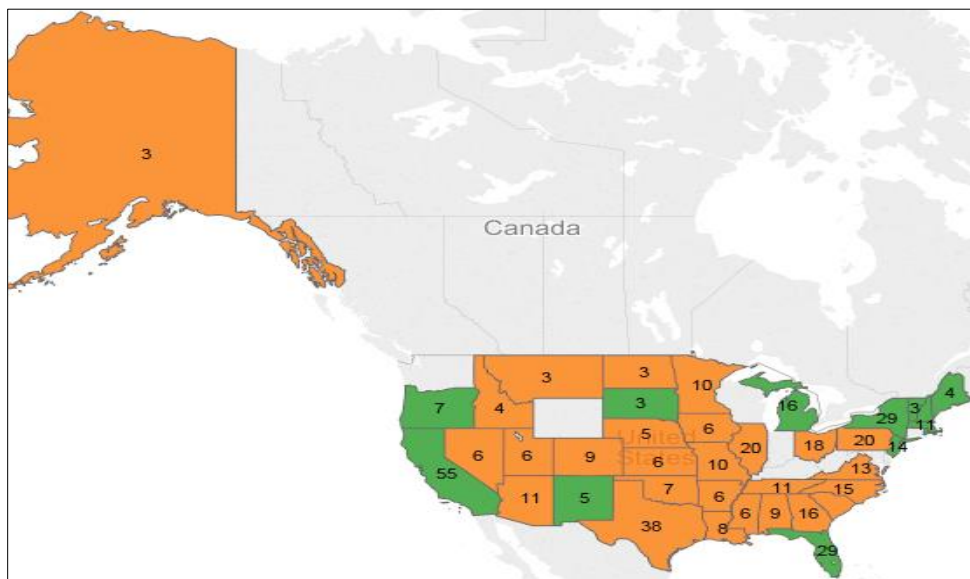


Fig3: Predicted Scores vs Actual candidate Scores over all states [A. Trump, B. Hillary, C. Cruz , D. Sanders].



Final Results:

Our model has predicted the distribution of states won by Hillary Clinton and Donald Trump.

By the total #of electoral votes won, the final Winner for the Final Elections, 2016 would be Hillary Clinton and not Donald Trump.

Trump Total : 188 (14 states) - Green

Hillary Total : 278 (27 states) - Orange

Congratulations Hillary Clinton !! 😊