Using Decision Trees to Predict Political Party Data

Shreya Patel *

May 2018

Abstract

This project explores the use of tree models in political science.

*Department of Economics, University of Oklahoma. E-mail address: shreyapatel@ou.edu

1 Introduction

Political data is widely used in the world of data science. Recently, an analytics firm was exposed for using private social media user data to classify the user by political views. This was done through Facebook data by Cambridge Analytica. This led to the question of how this is possible, which led to the creation of this project.

2 Literature Review

Analyses of political party affiliation are widely conducted by social scientists and statisticians for a multitude of reasons. This is extremely sought-after data because it can be used to predict elections more. President Obama's 2012 campaign is recognized for being the first presidential campaign to truly embrace and use data to it's potential. It is assumed that both Hillary Clinton and Donald Trump's 2016 campaigns used much more voter data than Obama's. Obama's campaign used much less data, but is still awarded the best use of data thanks to the lead data scientist, Rayid Ghani. His work in this campaign was simple and brilliant. Trump and Hillary's campaigns are credited with having more data than they knew what to do with, thanks to advances in technology via social media networks and more. This information is interesting in wake of the Cambridge Analytica and Facebook scandal. This raises the question of how hard it really is to predict a voter's political party affiliation. That is what this project seeks to study.

3 Data

The dataset used for this project are the Congressional Voting Records from 1984. This is available at the University of California Irvine's Machine Learning Repository, hosted on their website. This is a set of 135 rows of data. The first column classifies the affiliation of the candidate by one of two categories: Democrat or Republican. The next 16 columns mark a "y" or an "n" for yes or no votes to particular bills. The 16 columns are described as the following:

- 1. Handicapped-infants
- 2. Water-project-cost-sharing
- 3. Adoption-of-the-budget-resolution
- 4. Physician-fee-freeze
- 5. El-salvador-aid
- 6. Religious-groups-in-schools
- 7. Anti-satellite-test-ban
- 8. Aid-to-nicaraguan-contras
- 9. Mx-missle
- 10. Immigration
- 11. Synfuels-corporation-cutback
- 12. Education-spending
- 13. Superfund-right-to-sue
- 14. Crime
- 15. Duty-free-exports
- 16. Export-administration-act-south-africa

The Congressional Quarterly Almanac (CQA) collected the data. The categories originally listed by the CQA for each vote were from the following options:

- 1. Voted for
- 2. Paired for
- 3. Announced for
- 4. Voted against
- 5. Paired against
- 6. Announced against
- 7. Voted present
- 8. Voted present to avoid conflict of interest
- 9. Did not vote or otherwise make a position known

Options 1-3 from this list were categorized as yes, 4-6 was categorized as no, and 7-9 were categorized as unknown. Each row from columns 1-16 is almost all yes or no votes, with a few unknowns mixed in the dataset. Essentially, this dataset is organized categorically. The yes or no votes help associate certain votes with Democrat or Republican identities.

4 Methods

4.1 Decision Trees, Random Forests, and Politics

Decision trees and random forests models are very commonly used in politics. Tree models are easy to use and understand. They also require fewer assumptions on the side of the researcher than other model types. This allows political scientist, statisticians, and others studying political data to take a step back and allow the data to tell the story. Tree models also allow ease of use when working with extremely large data sets, which can be expected from voting data (Montgomery).

4.2 Decision Trees

Decision trees are fairly straightforward predictive models. These models aim to predict the value of the target variable by classifying input data. The visual representation of the tree typically starts with one node, or class, at the top that then stems to different leaf options underneath. There must be two or more options to classify the given node further. Decision trees are binary when there are only two options. Decision trees can learn by splitting the data set into pieces. For example, the rows of data fed into the tree can be split. One part can be used for training, and the other half for predicting accuracy. There are two different types of decision trees. One type is the classification tree – where the target variable has specific categorical identifications (Mitchell). The other kind is a regression tree – there the target variable has a non-categorical, continuous value. Both types of trees are used for this project (Magee).

4.3 Random Forests

Binary trees are straightforward to understand and use. However, they can also be too much so in training. Binary trees have a tendency to over fit around the training data. Some algorithms do better than others at correcting this. Random forests are implemented to reduce over fitting by randomizing binary trees. The random forest will is implemented to see if that is the case when comparing to the two binary trees (Mitchell).

4.4 Algorithms and Packages

All programming for this project was done in R. There were three distinct algorithms used for this project. Two of these are algorithms for implementing decision trees, while the third is used for random forest models. The first is the C5.0 algorithm. The R package was named C50. The C5.0 algorithm is a very well known method for creating binary trees. This model is simpler to use than the other, and also requires less data cleaning. The second is the CART algorithm. CART stands for Classification and Regression Trees. The R package used to implement this algorithm was rpart. This implementation creates a simple structure. It should be noted that both algorithms tend to over fit the model, as is common with binary trees. The third algorithm used was the ensemble algorithm to create the random forest. The package in R was used for this was ensembleR. This algorithm enables the use of random forests.

4.5 Process

As previously stated, this project was created in R using C50, rpart, and ensembleR packages. The three different models were created separately and their results compared at the end. The voting data file was used to train and predict outcomes. The file used has 435 rows of data. Every model used rows 1 to 300 as training data and 301 to 435 to predict outcomes. Confusion matrices were created from each model and used to find accuracy rates of each model. The accuracy rate is found using:

$$ErrorRate = \frac{FalsePositive + FalseNegative}{Total}$$

5 Findings

5.1 C5.0 Decision Tree

The decision tree categorized 71 of the 135 voters as Democrats and 61 out of the 135 voters as Republicans. 7 of the Democratic voters categorized were given false negatives and 1 of the Republican voters was given a false positive. 3 out of the 135 voters were categorized as unknown. Given this information, and excluding the 3 unknown, the findings for the C5.0 decision tree proved an error rate of 0.097560976. This gives an accuracy rate of 0.939393939. See Table 1 and Figure 1.

5.2 CART Decision Tree

The decision tree categorized 71 of the 135 voters as Democrats and 61 out of the 135 voters as Republicans. 0 of the Democratic voters categorized were given false negatives and 0 of the Republican voters was given a false positive. 3 out of the 135 voters were categorized as unknown. Given this information, and excluding the 3 unknown, the findings for the CART decision tree proved an error rate of 0. This gives an accuracy rate of 1.00. See Table 2 and Figure 2.

5.3 Ensemble Random Forest

The random forest categorized 184 of the 335 voters as Democrats and 142 out of the 335 voters as Republicans. 2 of the Democratic voters categorized were given false negatives and 0 of the Republican voters was given a false positive. 9 out of the 335 voters were categorized as unknown. Given this information, and excluding the 9 unknown, the findings for the CART decision tree proved an error rate of 0.006134969. This makes for an accuracy rate of 0.993865031. See Table 3.

6 Conclusion

The differences in the different models and algorithms are very apparent. The C5.0 decision tree and the CART gave error rates of 0.097560976 and 0, respectively. This can be explained by the fact that this is because the CART decision tree tries to correct over fitting, while C5.0 requires additional steps to include that. Therefore, CART gives a more accurate reading when working with non-training data. The ensemble random forest uses more variety because of the model, so it also provides a more accurate prediction than the C5.0 decision tree. This answers the original question: how hard is it really to predict political affiliation of users? Given the right tools and data, it is proven here that all it takes is a few lines of code.

final.bib [heading=final.bib, title=Whole bibliography]

7 Tables and Figures

Table 1

Confusion Table for C5.0 Decision Tree		
Y or N	Democrat	Republican
n y	TP = 70 FN = 7	FP = 1 TN = 54
Accuracy rate Error rate	0.939393939 0.097560976	(TP+TN) / total (FN+FP) /total

Table 2

Confusion Table for CART Decision Tree		
Y or N	Democrat	Republican
n y	TP = 71 FN = 0	FP = 0 TN = 61
Accuracy rate Error rate	1 0	(TP+TN) / total (FN+FP) /total

Table 3

Confusion Table for Ensemble Random Forest		
Y or N	Democrat	Republican
n y	TP = 184 FN = 2	FP = 0 TN = 140
Accuracy rate Error rate	0.993865031 0.006134969	(TP+TN) / total (FN+FP) /total

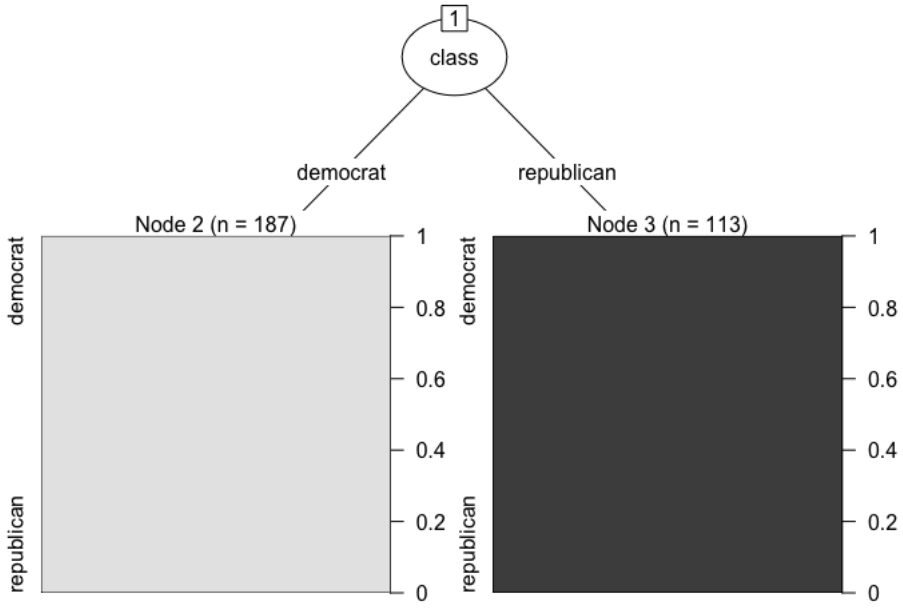

Figure 1

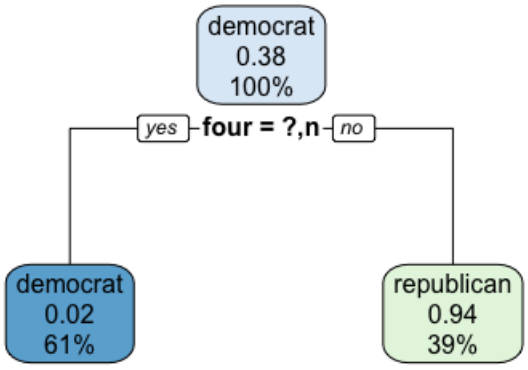

Figure 2