.

Introduction:

In the United States Congressional system there is a separation between the two parties. The two parties in the system are Republicans and Democrats. Each party has different stances on different issues which influences their congressional vote on different bills. In my work I will use different classifying models to extract information about congressmen and their voting style to determine how to classify them into their political party.

The models I'll be using to classify all the congressmen are the Nave Bayes Classifier model and the Regression Trees Classifier model. Both predict the party affiliation of the congressman, but they are very different in structure. In this research will be investigating the accuracy of their classification and draw conclusions on the models and determine what situations does one out performs the other.

I will base these models on descriptions found in David J. Hands work Nave Bayes in Top Ten Algorithms in Data Mining published in 2009 and Dan Strinbergs work in CART: Classification and Regression Trees.

Data Problem Definition:

To gather information about the congressmen and their votes we used a data set from the University of California Irvine Machine Learning Repository. The data set we used is called Congressional Voting Records Data Set. The data set is from 1984 and contains the voting records for sixteen major bill for that year and whether they voted for the bill (y), against the bill (n) or did not vote at all for the bill (?). The first column of the data set is the identification of the congressman. It states if the congressman is either republican (r) or democrat (d). The other columns are their vote on each of the bills. Here are the bills that are present in the data set in the order they appear in the data set:

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 Missile
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

This data along with my algorithms will allow us to predict the party affiliation of the congressmen and allow us to compare Nave Bayes and Regression Trees.

Model Definition

Nave Bayes is very similar to the traditional Bayes and fallows the fallowing format:

$$\mathrm{P}(party \mid votes) = \frac{\mathrm{P}(votes \mid party)\,\mathrm{P}(party)}{\mathrm{P}(votes)}$$

To compute the conditional probability of votes given the class you can find the individual conditional probabilities of each individual vote and that will be equivalent. You can calculate the $\mathrm{P}(vote_i \mid class)$ by making a sub data frame of the training data with just those of the party and then determine which ones voted whatever the test congressman voted for that bill and then divide that over the total amount of congressmen in that data frame. The $\mathrm{P}(party)$ is calculated by taking the training data and then finding the number of people that identify as apart of that party and divide that about the number of congressmen in training data.

$$= \frac{\prod_{i=1}^{votes} \mathrm{P}(vote_i \mid pary)\,\mathrm{P}(party)}{\mathrm{P}(votes)}$$

In Nave Bayes you are comparing the conditional probability that the congressman is Republican and to the conditional probability that the congressman is Democrat in both the probability votes is the same so you can just disregard the probability of the votes all together.

$$\mathrm{P}(party \mid votes) = \prod_{i=1}^{votes} \mathrm{P}(vote_i \mid party)\,\mathrm{P}(party)$$

All of these probabilities are found based on the training data set you give it. You then can give the the formula a new congressman and then compute the $\mathrm{P}(Republican \mid votes)$ and the $\mathrm{P}(Democrat \mid votes)$ and the one with the highest output is the party that the training congressman belongs too.

Regression Trees is the other classification technique I used. In Regression Tree classification is a binary recursive partitioning procedure. The beginning root is the whole training data and its children are a sub data frame that is slit based on the content of one of it contents. Where the right child is all the congressmen that votes yes to a certain bill and the left child is all the congressmen that vote no to that same bill. That process is recursively done until each child node is pure. Pure means that the data frame is just Republican or Democrat. The split is determined by a Gini. The Gini is calculated by first finding the purity.

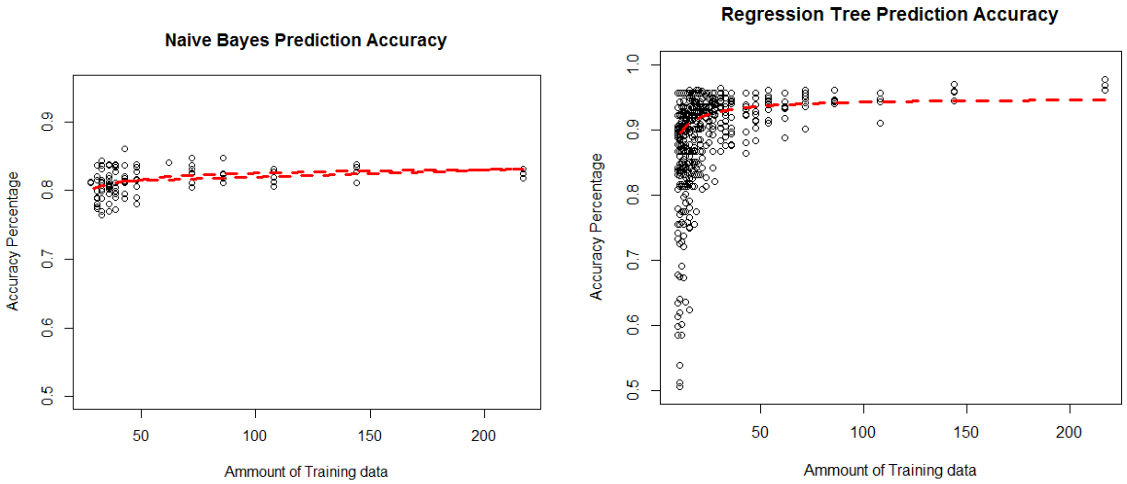
$$purity(dataframe) = 1 - (\mathrm{P}(dataframe))^2 - (1 - \mathrm{P}(dataframe))^2$$

$purity(data)$ is the how pure the data set is. $\mathrm{P}(vote_i)$ is the probability that you get a republican given the data frame.

$$gini(vote_i) = purity(root) - prurity(right) - purity(left)$$

$purity(root)$ is the purity of the root data frame. $purity(right)$ is the subset of all the congressman that voted yes on $vote_i$ and $purity(left)$ is the subset of all the congressman that voted no on $vote_i$. The Gini is then computed for every vote and then the official split vote is the vote with the highest Gini. This process is continued over and over again until the nodes are completely pure.

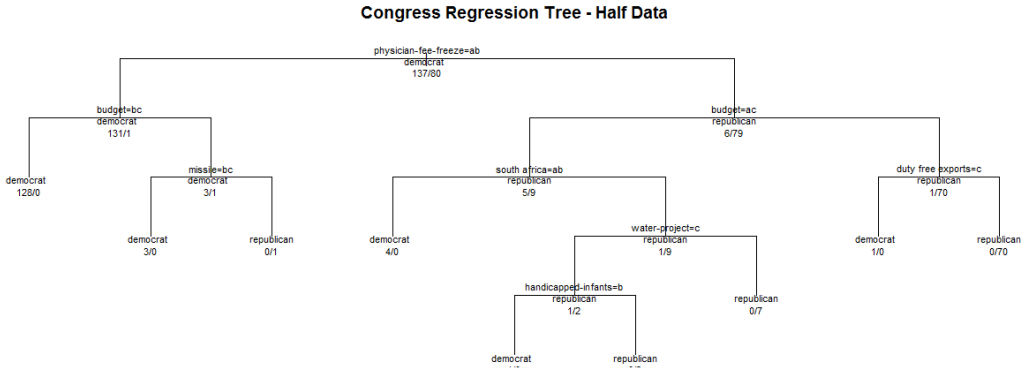
Evaluation and Results:



Black Dots are the actual testing data prediction accuracy . The red line is the line to model the relationship

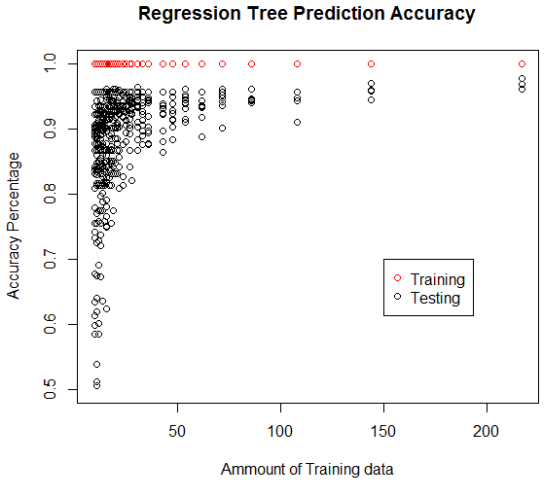
As you can see in the graphs above there is difference in the Accuracy of the two models. In Naive Bayes, there is a more constant prediction accuracy. No matter what the size of the training data the accuracy falls around 80%. For Regression Trees there is a drastic change in accuracy when the size of the training data is about 15 congressmen to when the size is about 250 congressmen. In both cases the accuracy tends to get higher as there the training data set gets bigger. But Regression Trees tend to out preform Naive Bayes in the end with the accuracy averaging around 97%. Not is all lost with Naive Bayes because it does out perform Regression Trees when the the training data is smaller

Something I did notice about Regression Trees is that there is a large variance in the accuracy when the data is smaller. This is a sign of over fitting. Over fitting is bad because it means that your are training your model on data to fit your training data and not the real world. So when you test new data on the model it tends to not do as well. An example of over fitting is below.



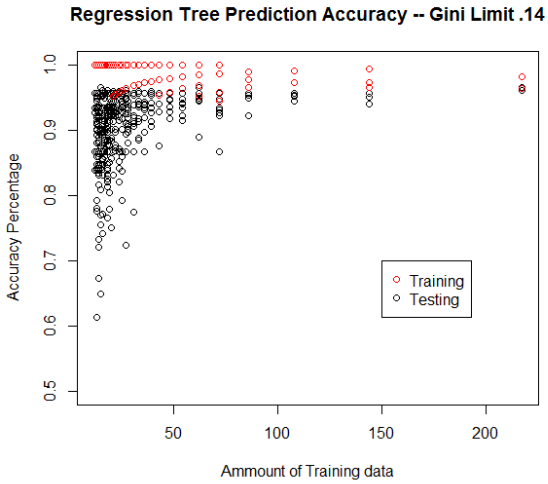
In the example above the the farthest right grandchild has one democrat and seventy republicans. It then further splits again. This is fitted to the training data and not the real world. If a new Congressman or Congresswoman came in and voted yes to the physician fee freeze and the budget is it highly likely that that person is republican. But if that person voted no for the duty free exports bill they will be deemed as Democrat. This is an example of over fitting

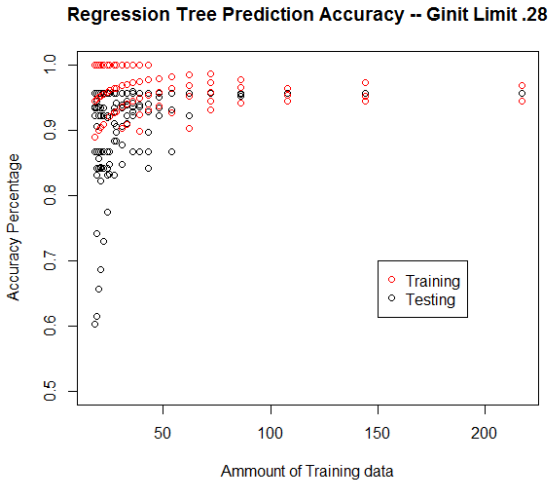
You can see over fitting in a graphical sense below. In the graph below the training data has an accuracy level of a constant 100%, while the testing data has a significantly lower average. This means that my Regression Tree Classification model is over fitting.



.

One way you can prevent over fitting is by stopping the splitting process when there is an n amount of one party in a node. That fixed n is usually something low, such as less than 10. You can also have a fixed Gini that serves as your stopping point. I choose the fixed Gini as my version of pruning. You can choose the other one and get similar outcomes.





As you see when I pruned the trees I the training and testing had similar accuracy. This also limited the amount of variance in the lower training set sizes.

Conclusion and Discussion

From our results we were able to accurately predict the party affiliation of a congressman based on their votes on bills. We were able to do this with two different models, Niave Bayes and Regression Trees. Both models were made to do the same thing and they were both built off the same data. Yet they had varying degrees of accuracy.Niave Bayes preforms at about 75% accuracy with a training set of about 20. As the training set grow the accuracy grows but plateaus at around 80%.

In Regression Trees, the accuracy starts smaller than Niave Bayes on small training set size of about 20. But as the training set grows the accuracy grows faster than Niave Bayes and then finally plateaus around 95%.

In the end I came to the conclusion that Naive Bayes does better when you have a smaller training set. And Regression Trees does better when you have a larger training set.