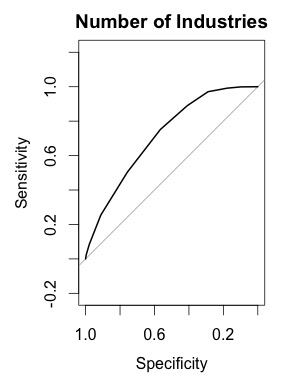
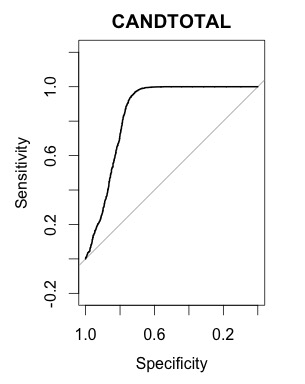
For the Parametric Statistical Tests, the dataset, “PoldataSPIndustriesStockData no outliers”, was used, the student t-test was run on one of the three hypothesis developed, while the logistic regression model was run on a possible linear relationship. In hindsight, it would have been interesting to run cross validation on the model, but, unfortunately, time was a factor in forgoing it. ROC curves were provided on both of the attributes tested by the logistic regression model.

The first Null hypothesis was tested was: the is no difference between the total contributions that an incumbent gets and one that a challenger gets. We used our Merged Data set without outliers. We preformed a **student t-test** (not pairwise) to test this null hypothesis. We got a p-value of < 2.2e-16 and given that that this a social science analysis, the threshold should be .05. The p-value crosses this threshold and is well in the rejection region, so the p-value of extremely significant. So we reject the null hypothesis in favor of the alternative which is that there is a difference. Given the mean of the two categories of contributions, it is clear that the incumbent has a higher amount of the contributions compared to the challenger. This is also verifiable when compared to the association rules.

The Second Null hypothesis that tested involved a **logistic regression model**. The idea behind the model is to predict who the winner will be based on the Total amount of money raised and the Number of supporting industries. From the confusion matrix, we know the following; The Precision of the model is 0.8125881, the recall is 0.8662994, is F-measure is 0.8385846. Below are the ROC plots of both variables. The accuracy against the training data is 0.8184791; however, the number seems low because the prediction should have better matched against the actual results. This may mean that the data might not best tested using a model that assuming normally distributed data.



From the ROC curved, it seems like the logistic model predicts with a surprising degree of accuracy. This could signify that the model maybe be over fitting. This is probably because the model is predicting the values based upon itself. As excepted, the total amount of money raised seems to have a bigger impact on accuracy as opposed to Number of Industries.

Confusion Matrix:

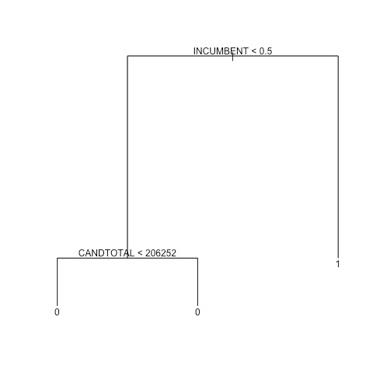
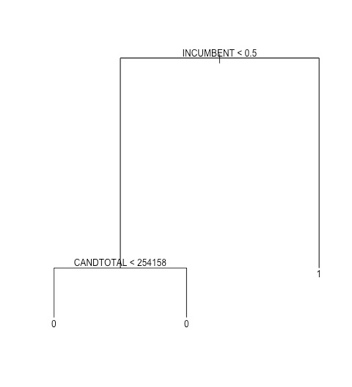
|  |  |  |
| --- | --- | --- |
|  | lose | win |
| Lose | 1273 | 399 |
| win | 267 | 1730 |

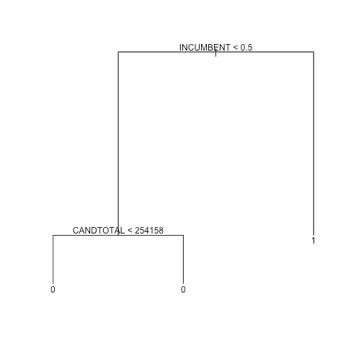
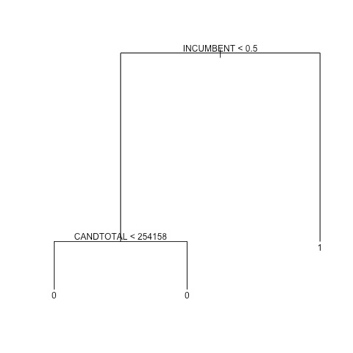
For the data driven Predictive models, we looked into the following **third hypothesis**, the idea was to see if the winner attribute of a candidate, a binary one, could be predicted looking at Candidate’s Total Industry Contributions, the number of supporting industries, and whether the candidate was an incumbent. In each of the three models, cross validation was run 5 times. The training set comprised of 80% of the data set whereas the test dataset comprised of 20% of the original dataset.

For the decision Tree, the follow occurred with the tree after in was pruned in terms of confusion matrices, accuracy of model, precision, recall, and the F-measure:

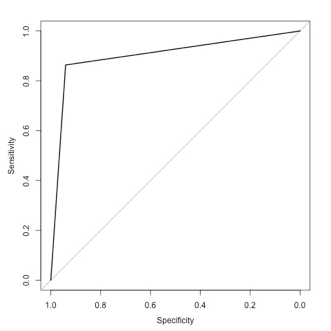
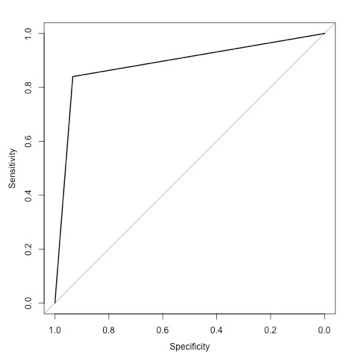
Please note for all confusion matrices, 0 denotes losing and 1 denotes winning.

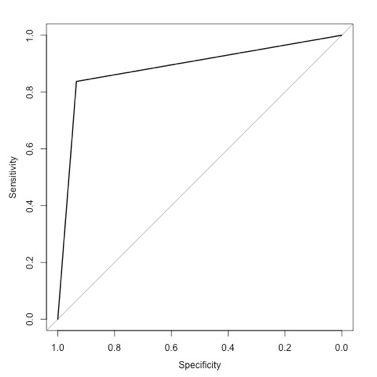
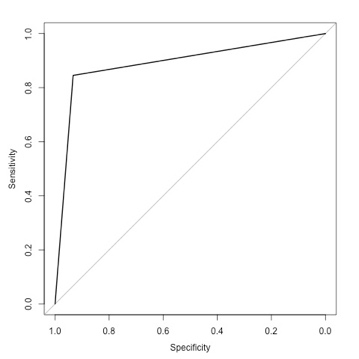
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4 | Iteration 5 |
| |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 314 | 70 | | 1 | 16 | 334 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 316 | 58 | | 1 | 14 | 346 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 313 | 60 | | 1 | 20 | 341 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 321 | 59 | | 1 | 17 | 337 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 332 | 52 | | 1 | 21 | 329 | |
| accuracy: 0.882834 | accuracy: 0.901907 | accuracy: 0.891008 | accuracy: 0.896458 | accuracy: 0.900545 |
| precision: 0.826733 | precision: 0.856436 | precision: 0.850374 | precision: 0.85101 | precision: 0.86352 |
| Recall: 0.954286 | Recall: 0.96111111 | Recall: 0.944598 | Recall: 0.9519774 | Recall: 0.94 |
| F-measure: 0.88594 | F-measure: 0.90576 | F-measure: 0.89501 | F-measure: 0.89867 | measure: 0.90013 |





Due to an issue in the for loop to run the iteration, I wasn’t able to get the tree structure. However, from the tree structure, it is obvious that the candidate total does not impact whether a challenger wins or not given the status of being an incumbent.

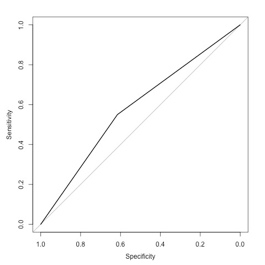
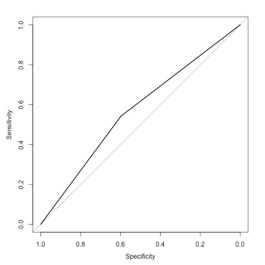
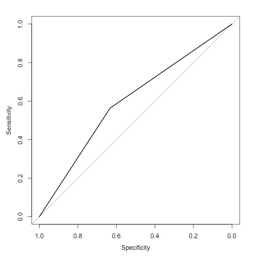
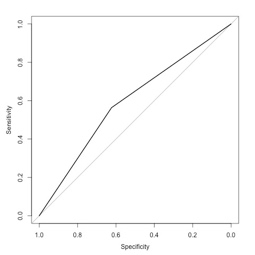


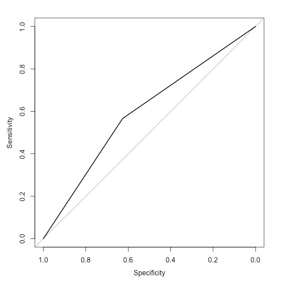


From the ROC curves, it seems like the model does a good job of predicting the candidate that will win. Interesting note is that the decision tree seems to associate winning the election with being an incumbent. The accuracy rates also seem to confirm this idea in addition to the tree structure and confusion matrix because they are close to what the winning rate for an incumbent is. Based on this model, it confirms a deeply believed theory in congressional politics, incumbents win. However, the model may not be the most useful because it’s precision is lower than the other categories.

For the KNN (lazy learner) algorithm, the k value was benchmarked for values between 1 to 20; the benchmark was based on finding the k value that resulted in the most accuracy. This could have been improved if, for each k, was tested approximately 1000 times, but given the size of the dataset, one time was enough. In the end, the K value that was used was 17.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4 | Iteration 5 |
| |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 256 | 16 | | 1 | 86 | 376 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 244 | 18 | | 1 | 80 | 392 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 247 | 14 | | 1 | 96 | 377 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 236 | 13 | | 1 | 89 | 396 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 229 | 20 | | 1 | 111 | 374 | |
| accuracy: 0.576294 | accuracy: 0.611716 | accuracy: 0.572207 | accuracy: 0.588556 | accuracy: 0.5722 |
| precision: 0.959184 | precision: 0.956098 | precision: 0.96419 | precision: 0.96822 | precision: 0.94924 |
| Recall: 0.81385281 | Recall: 0.830508 | Recall: 0.797040 | Recall: 0.816495 | Recall: 0.771134 |
| measure: 0.880562 | F-measure: 0.88889 | F-measure: 0.87269 | F-measure: 0.88591 | F-measure: 0.85097 |

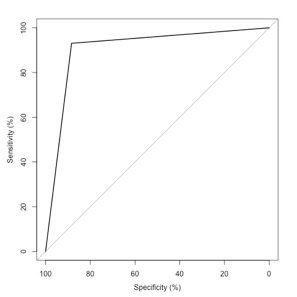
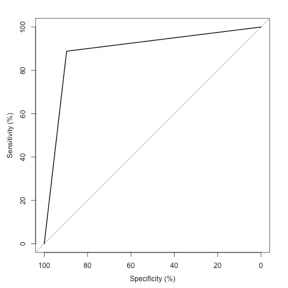


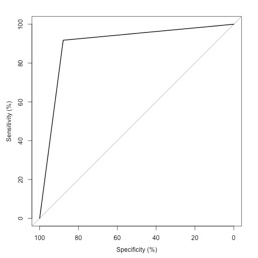
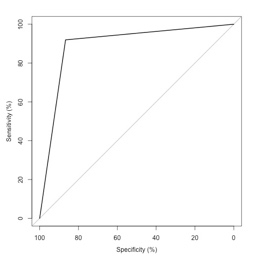
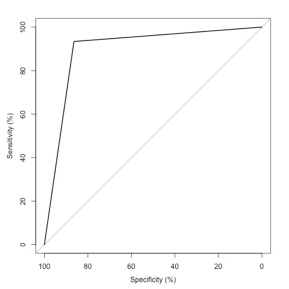


Based on the accuracy, this model doesn’t do a good job of determining accuracy; however, the precision is very good, meaning that the amount of true positives is very good. The downsize is that there is a significant of false positives that are detected. This model may not be the best for overall detection. The issue may be because of the incumbent value.

For the Naïve Bayes algorithm, the follow occurred in terms of confusion matrices, accuracy of model, precision, recall, and the F-measure:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4 | Iteration 5 |
| |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 333 | 54 | | 1 | 19 | 328 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 324 | 60 | | 1 | 18 | 332 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 327 | 58 | | 1 | 17 | 332 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 321 | 62 | | 1 | 15 | 336 | | |  |  |  | | --- | --- | --- | |  | 0 | 1 | | 0 | 328 | 49 | | 1 | 18 | 339 | |
| accuracy: 0.900545 | accuracy: 0.89373 | accuracy: 0.897820 | accuracy: 0.895095 | accuracy: 0.908719 |
| precision: 0.858639 | precision: 0.84694 | precision: 0.85128 | precision: 0.84422 | precision: 0.87371 |
| Recall: 0.94524496 | Recall: 0.9485714 | Recall: 0.9512893 | Recall: 0.95726496 | Recall: 0.9495798 |
| F-measure: 0.89986 | F-measure: 0.89487 | F-measure: 0.8985 | F-measure: 0.8972 | F-measure: 0.91007 |





From the results of the confusion matrix, the accuracy, precision, recall, and F-Statistics, are equal to or better than the other two models. The ROC curves cover a bigger area than than either KNN or Decision trees. The false negative numbers are significantly less as well, which is also evident by precision and recall.