**Advanced Classification**

**Purpose:** To build and test advanced classifiers and prescribe strategies

**Description:** Using data from 2010 Congressional elections, we intend to build a classifier that would predict the election’s outcome. The data set includes information about the campaign funds, social media (Twitter, Facebook, and YouTube) campaigns, and demographics (age, gender) of 941 candidates who were in race in the general elections for The 112th House of Representatives seats in The U.S. Congress.[[1]](#footnote-1)

<http://www.fec.gov/finance/disclosure/metadata/DataDictionaryWEBALL.shtml#search=%22trans_from_auth%22>

Step Wise -

1. Remove all of the observations with any missing values using function complete.cases()
2. Randomly assign 70% of the observations to train\_data and the remaining observations to test\_data (Refer to Module 6 for the code).
3. Use train\_data to build a random forest classifier with 10 trees. Use library(randomForest).
   1. OOB estimate of error rate – 9.5%
   2. variables R tried at each split **-** 5
   3. Now use 20 trees.
      1. OOB estimate of error rate - 8.77%
      2. variables R tried at each split **-**5
   4. Now use 30 trees.
      1. OOB estimate of error rate - 7.85%
      2. variables R tried at each split - 5
   5. Increase the number of trees in 10 increments (e.g. 40, 50, …). Using OOB error rate to evaluate your random forest classifier as to how many trees would you recommend.

*Results:*

*40 – 7.85%*

*50 - 6.46%*

*60 – 6.62%*

*70 – 6.62%*

*80 - 6.15%*

*90 – 6.31%*

*100 -6.77%*

*110 - 6.62%*

*120 – 6.31%*

*130 - 6.92%*

*140 - 6.31%*

*150 -6.77%*

*Looking at the OOB error to evaluate I would recommend 80 number of trees as it has the least OOB error.*

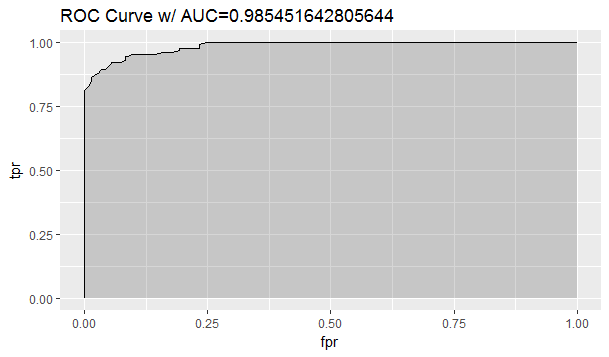
* 1. Use tuneRF() function to find the best value for mtry.

Replace n with the number of trees you recommended in 3.5. The recommended value for mtry - 4

* 1. Use your recommended number of trees and mtry value to build a new random forest classifier using train\_data. What is OOB estimate of error rate? 6.31%
  2. Use library(caret)[[2]](#footnote-2) and the code in Module 6 to create the confusion matrix for test\_data. Fill out the confusion matrix in below. Use “W” as the value of option positive in confusionMatrix() function.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual | | |
| Predicted |  | W | L |
| W | **125** | **11** |
| L | **8** | **135** |

* + 1. value of accuracy - 93.19%
    2. value of TPR - 93.98%
    3. value of FPR - 8.22%
  1. Calculate AUC and create the ROC curve.
     1. value of AUC- 0.98545
     2. ROC curve is below:

**

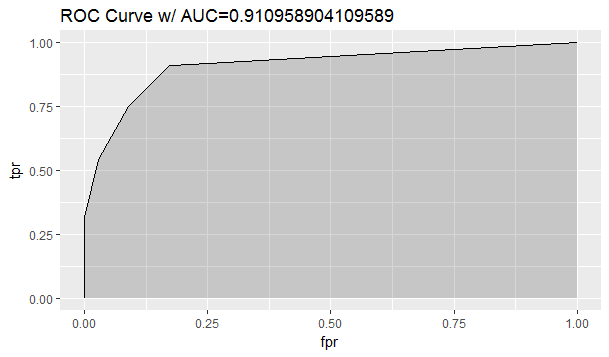
* 1. Use varImpPlot() to create the plot for variable importance.

Five important type of variable when we use MeanDecreaseAccuracy are:

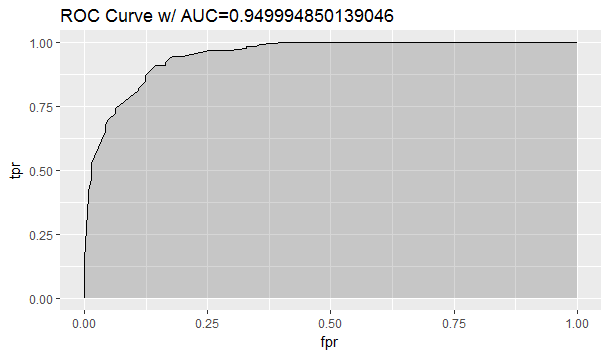
1. opp\_fund
2. coh\_cop
3. other\_pol\_cmte\_contrib
4. facebook
5. coh\_bop
6. Use library(nnet) and the code in Module 6 to build a neural network classifier.
   1. Use 5 hidden nodes in your ANN.
      1. input nodes in the ANN - 39
      2. weights in the ANN - 206
      3. Use library(caret) to create the confusion matrix for test\_data. Fill out the confusion matrix in below. Use “W” as the value of option positive in confusionMatrix() function.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual | | |
| Predicted |  | W | L |
| W | **100** | **13** |
| L | **33** | **133** |

* + 1. sensitivity – 75.19%
    2. specificity– 91.10%
    3. calculate AUC and create the ROC curve.
       1. of AUC - 0.91095
       2. the ROC curve is in the space below:

**

* 1. Increase the number of hidden nodes until you get the following error: “*Error in nnet.default(x, y, w, entropy = TRUE, ...): too many (1026) weights.”* Use the maximum number of hidden nodes that you can use to build your ANN classifier.
     1. maximum number of hidden nodes that we could use - 24
     2. calculate AUC and create the ROC curve.
        1. AUC - 0.94999
        2. ROC curve is in the space below:



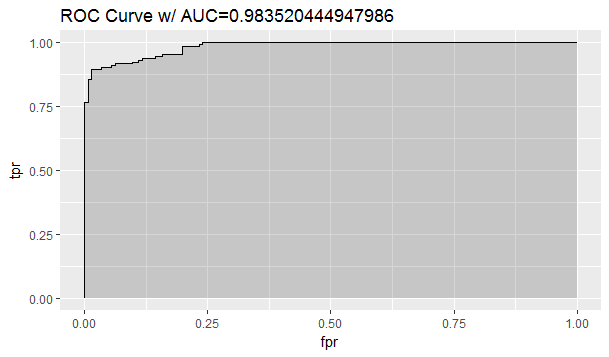
1. Use library(caret)[[3]](#footnote-3) to build a GBM classifier.
   1. Use function summary() to obtain the relative importance of the variables in the model. Compare these variables with the variables you obtained from random forest classifier. they are same in gbm and random forest

|  |  |
| --- | --- |
| **GBM** | **RF** |
| other\_pol\_cmte\_contrib | opp\_fund |
| coh\_cop | coh\_cop |
| opp\_fund | other\_pol\_cmte\_contrib |
| facebook1 | Facebook |
| coh\_bop | coh\_bop |

* 1. Use library(caret) create the confusion matrix for test\_data. Fill out the confusion matrix in below. Use “W” as the value of option positive in confusionMatrix() function.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual | | |
| Predicted |  | W | L |
| W | **122** | **9** |
| L | **11** | **137** |

* 1. calculate AUC and create the ROC curve.
     1. AUC - 0.983520
     2. the ROC curve is in the space below:

**

Though there is a close call between GBM and Random Forest, I would finally choose Random Forest as the AUC is greater than that of GBM and the sensitivity i.e. the TPR is higher. My understanding of the problem is to predict the outcome and building strategy around it so the True positive rates gives an impression of the model being accurately predicting wins. Hence Random Forest performs better in predicting True positives.

The buzz from the 2008 election motivated the candidates for political offices to employ social media campaigns to get their message across. Based on the analysis, I would you recommend sparing money and resources to create social media campaigns, among the three social media platforms (Facebook, Twitter, and YouTube), I would recommend Facebook as a media platform to invest in. As the below results suggests facebook bagged the highest number of wins. The models RF and GBM also points facebook as a one of the top five references variable to support their model.

gen\_election L W

facebook

0 481 206

1 13 229

gen\_election L W

twitter

0 480 250

1 14 185

gen\_election L W

youtube

0 482 209

1 12 226

Looking at the above analysis it can be said “Money Buys Political Power” as the major important variable having an impact on the model are:

|  |
| --- |
| **opp\_fund** – opportunity Fund |
| **coh\_cop** – Ending Cash |
| **other\_pol\_cmte\_contrib** -Contributions from Other Political Committees |
| **Facebook –** Facebook campaign |
| **coh\_bop** - Beginning Cash |

Higher the value of mean decrease accuracy or mean decrease gini score, higher the importance of the variable in the model. Hence the major variables showcase the funds required for the campaigns which contribute towards winning. The more the funds are the better the chances of winning the elections.

My Prescriptions for success based on my analysis to my candidate would be as follows:

1. Separate campaigns focused on raising funds may it be local or raising funds from other political committees. More funds leads to better campaigns leads to better awareness of the party among the voters
2. In terms of picking social media for campaigns, Facebook should be the one to invest on. It yields better results as seen in the past. It can play a pivotal role in winning the elections. Facebook channel can be useful motivating the youth for voting as its one of the medium most of the youth are connected to.

1. To read more about the general election, please refer to: <http://www.wwnorton.com/college/polisci/campaignsandelections/ch/09/outline.aspx>, and for information about the U.S. Congress, please refer to: <https://en.wikipedia.org/wiki/United_States_Congress> [↑](#footnote-ref-1)
2. You may get an error that will require you to use library(e1071). In this case, install this library using install.packages(“e1071”). [↑](#footnote-ref-2)
3. Please do NOT use library(gbm). [↑](#footnote-ref-3)