Stat 154 Final project (More than a corgi club)

Jiyoon Clover Jeong and Jin Kweon

Fall 2017

Our Mascotte

Intro

The final project involves with the following three questions:

1. Which variable does have to do with the income the most?
2. Which model is the best in terms of having less testing error rate?
3. Which model is the best in terms of having higher AUC?

Our goal:

Based on 15 variables on individual samples, perform prediction to determine whether a person makes over 50K a year

Our sub-goal:

Make the prediction error rate reasonably good enough and make the model not too much complicated

Data

The dimensions of training and testing sets are 32561 by 15 and 16281 by 15.

Two types of data: Continuous and Categorical

- Continuous variables: age, fnlwgt, education-num, capital-gain, capital-loss, hours-per-week,
- Categorical variables: workclass, education, marital-status, occupation, relationship, race, sex, native-country, income

EDA

Steps we do here:

1. Find missing value/NA
2. Take out the variables that are not interpretable
3. Find out continuous variables, and transform them into numeric if necessary
4. Find out categorical variables, and transform them into factor if necessary
5. Handling outliers
6. Do str and summary to find out data structures and fix if necessary
7. Discretizeing/Binning
8. Convert to dummy indicators

Highlights of EDA

1. Collinearity issues between “education” and “education.num”
2. Remove not-needed variable: “fnlwgt”
3. Find outliers data from “capital gain”
4. Dummify for interpretation purpose
5. Use missForest package to impute missing values

Plan

- Classification tree
- Bagged tree
- Random forest
- Boosted tree

Tuning Parameters

1. Classification tree: alpha for cost complexity pruning, classification for splitting tree
2. Bagged tree: number of trees (using tuning parameter alpha for cost complexity), depth of the tree/minimum size of nodes
3. Random forest: number of predictors considered, number of trees, depth of the tree/minimum size of nodes
4. Boosted tree: number of splits/interaction depth, shrinkage parameter, number of trees

Classification Tree Intro

Most famous non-parametric classification methods

- Why not Regression tree?

Ans: we try to predict a qualitative response: income (two categories)

- Package: rpart

Classification Tree Strengths

- Can be constructed in the presence of qualitative predictor variables without creating dummy variables
- Currently used widely and easy to interpret for non-expert
- Closer to human decision making
- Can be displayed graphically
- Can handle missing data
- Can handle multi-collinearity issue better than any other linear methods

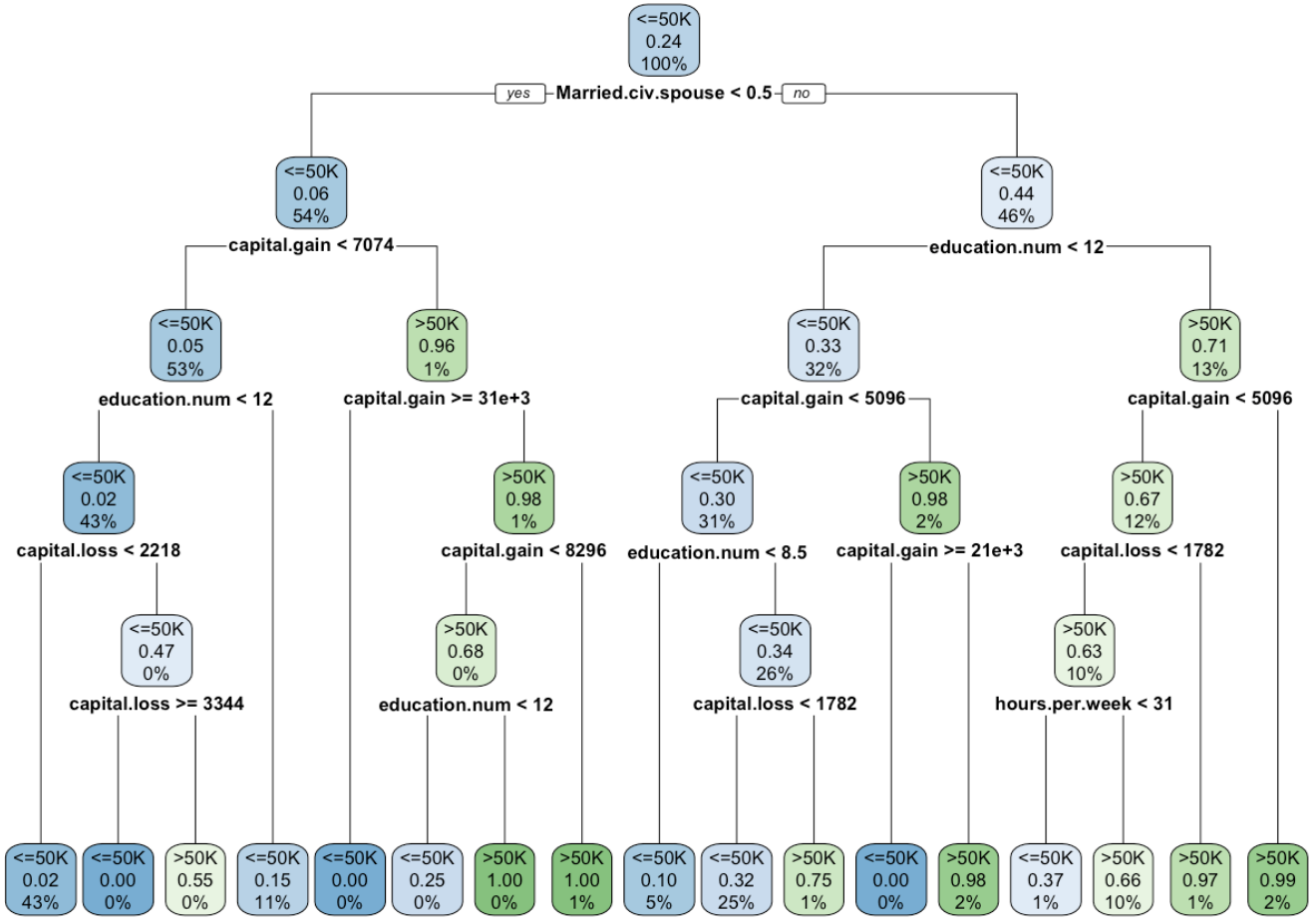
Classification Tree Weakness

- When the distribution is given such as linear or quadratic between classification outcome and predictors, it will perform worse than other classification methods
- Generally weaker predictive power
- Suffer from instability
- Easy to overfit
- Need large sample sizes

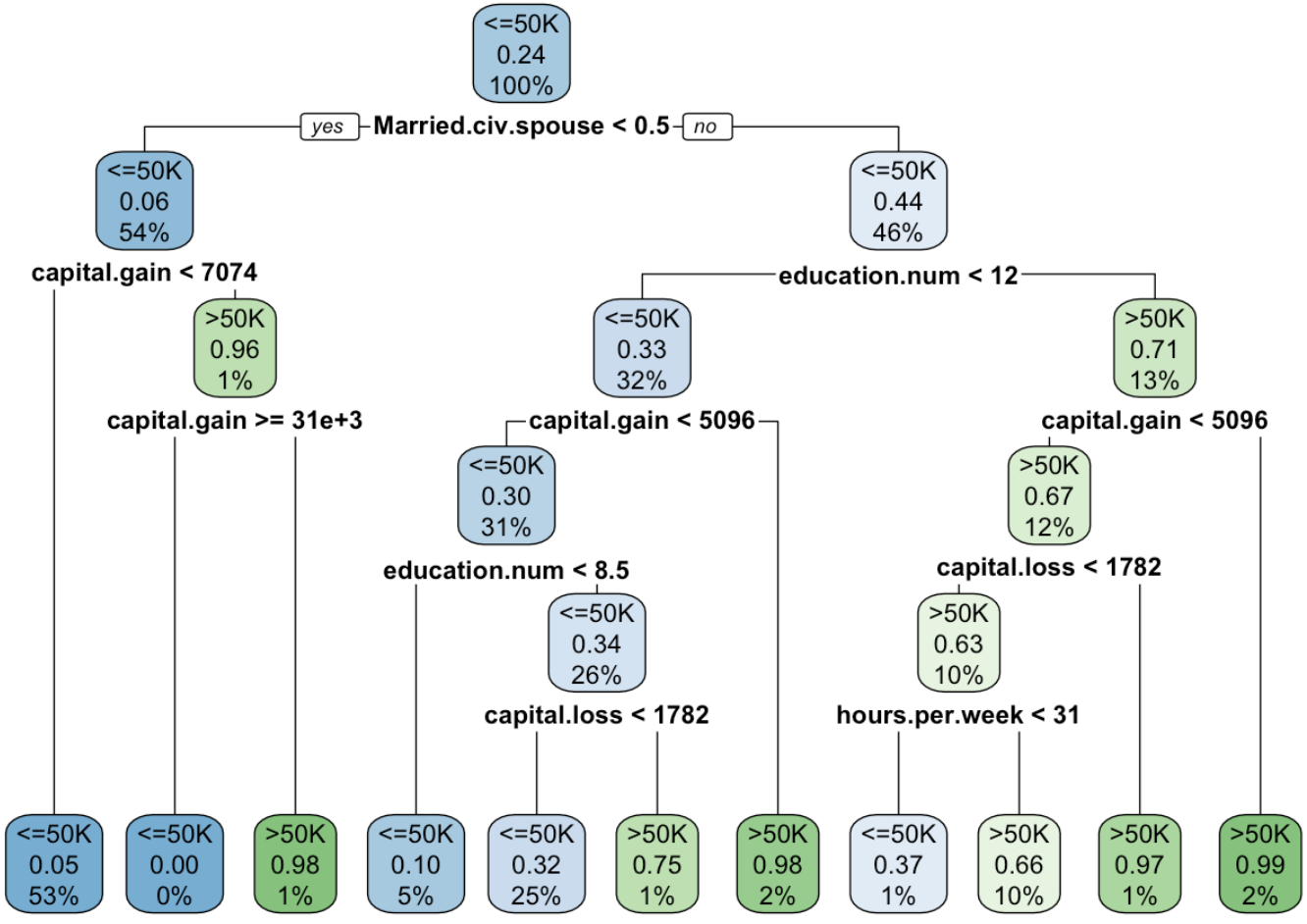
Classification Tree Tuning Parameters

- Alpha in cost-complexity pruning/weakest link pruning
- Minsplit (the minimum number of observation in a node for a split to take place)
- Minbucket (the minimum number of observation we can keep in terminal nodes).

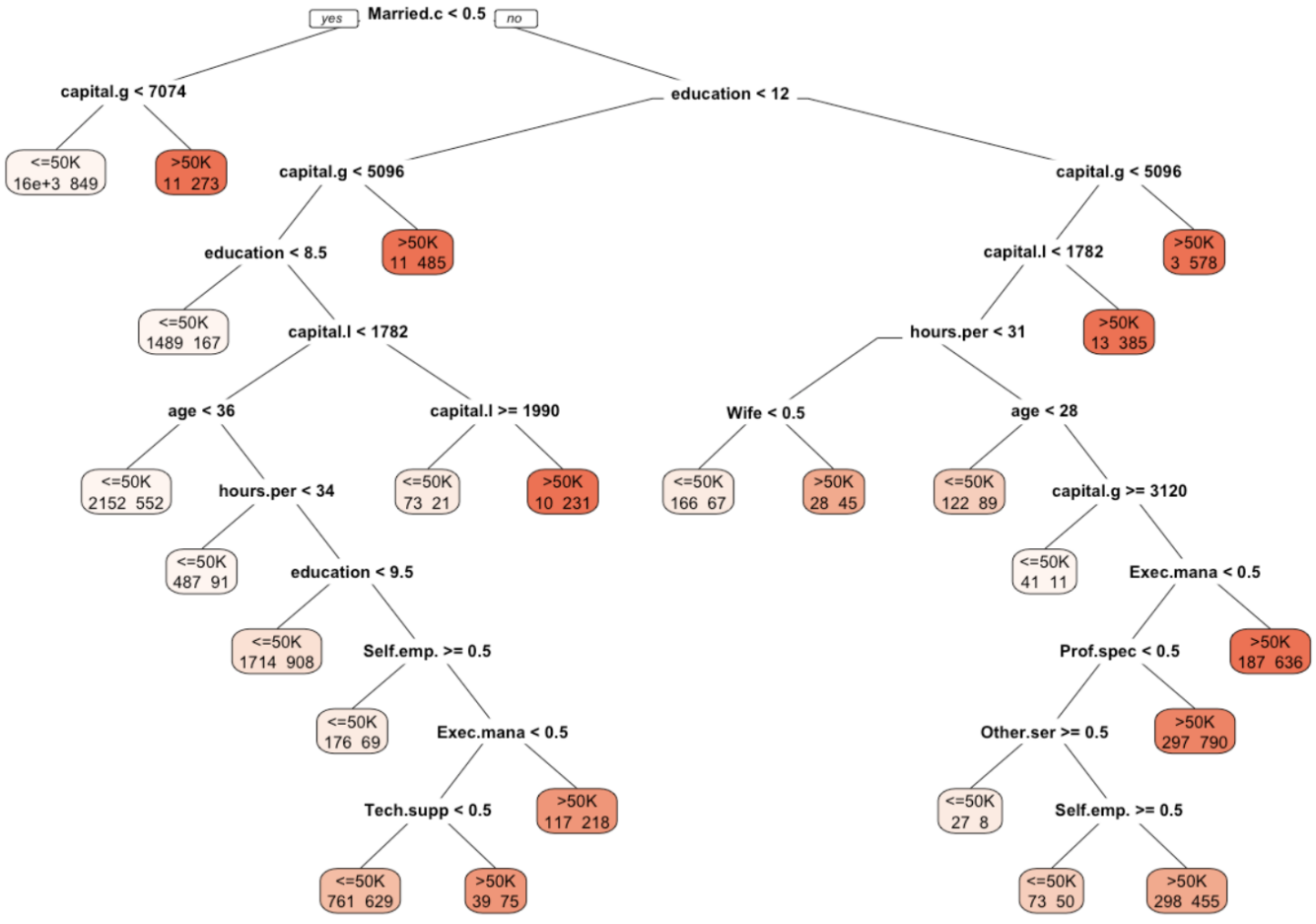
Classification Tree - unprunned model



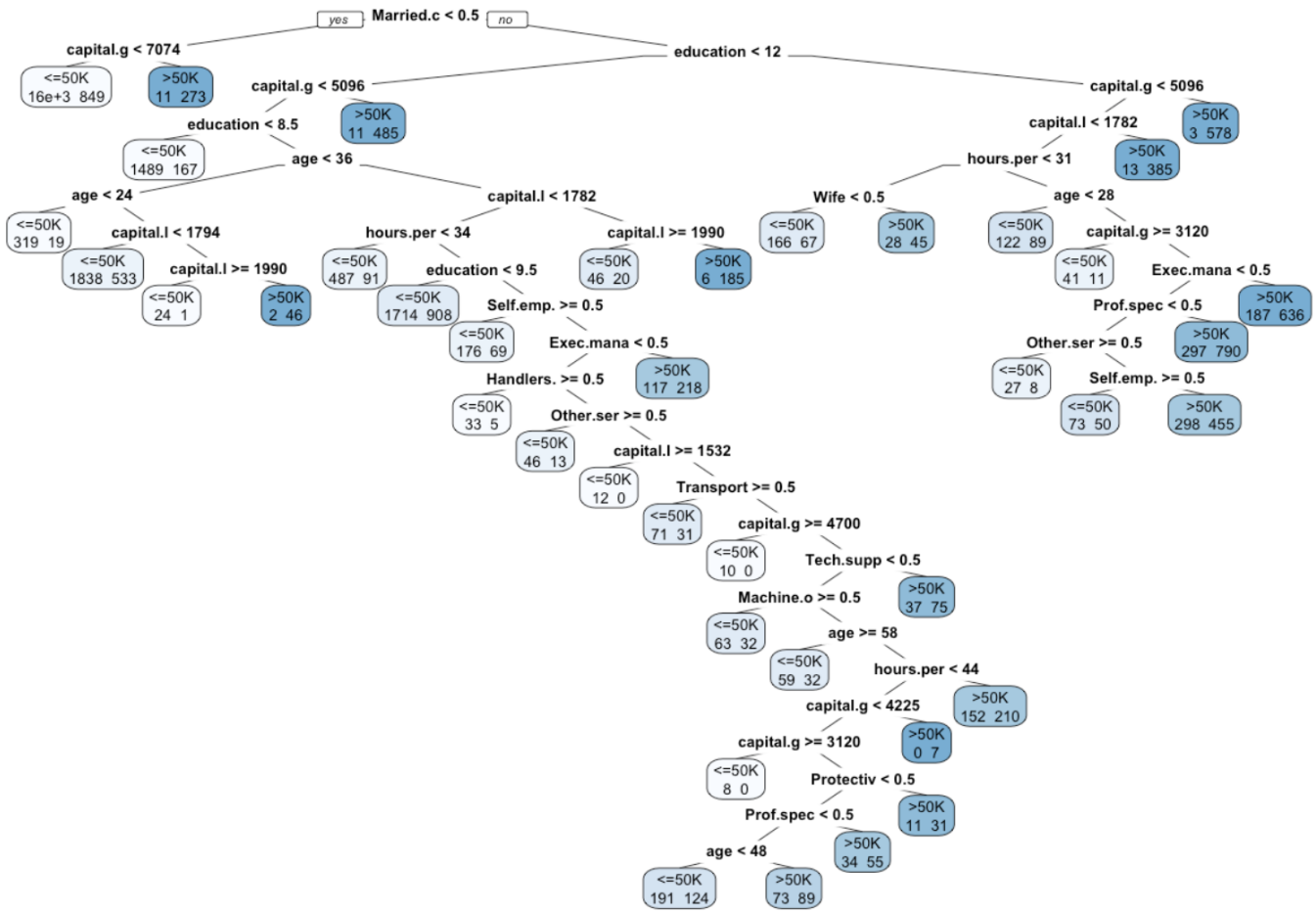
Classification Tree - tunned by the first method



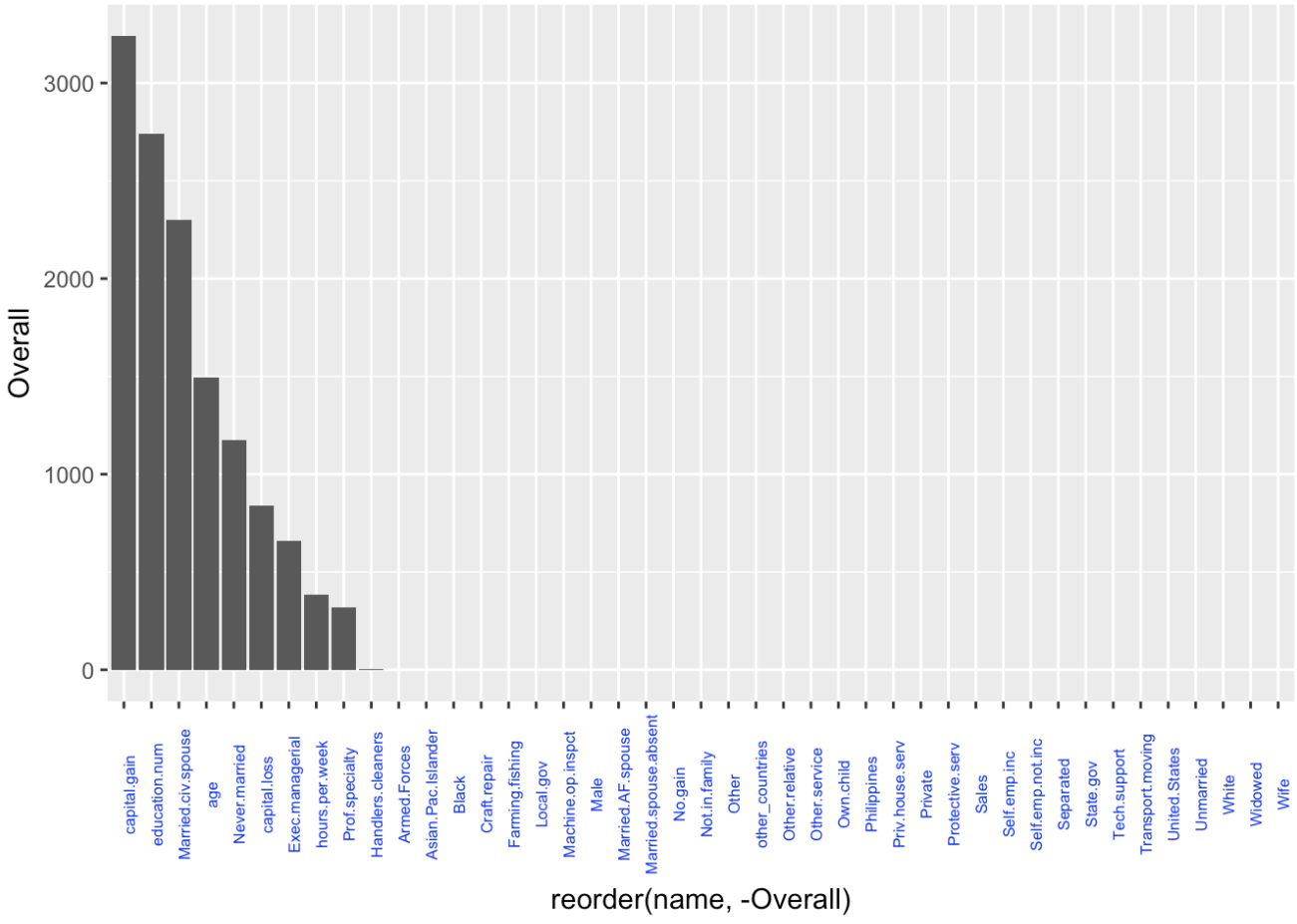
Classification Tree - tunned by the second method



Classification Tree - tunned by the third method

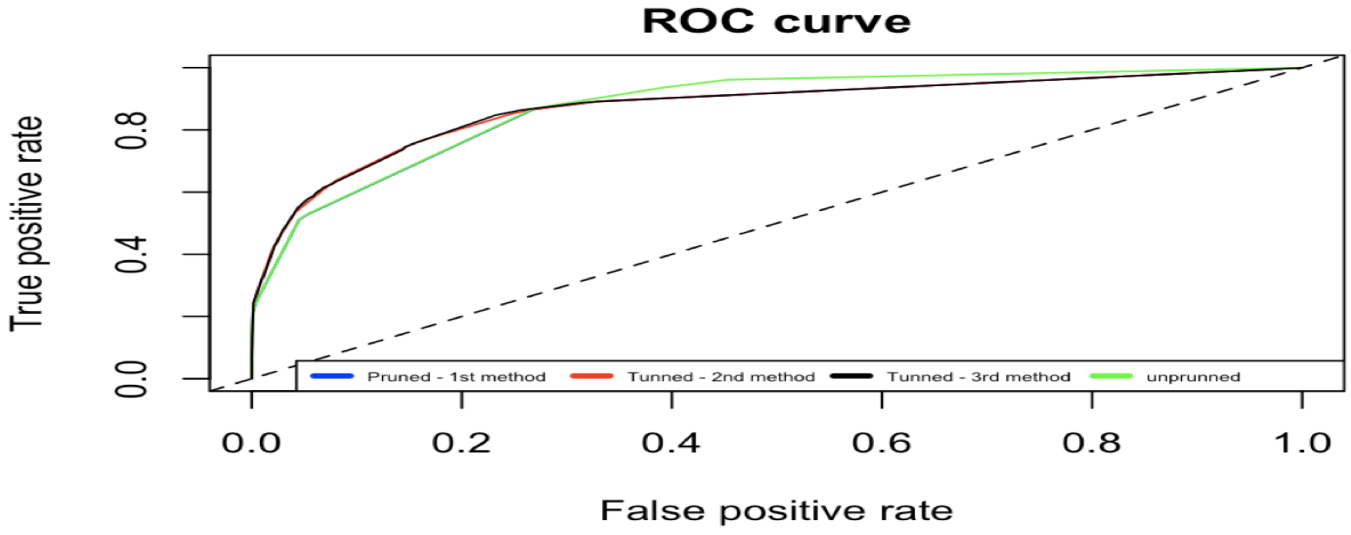


Classification Tree Varaible Importance



all models has similar variable importance statistics as this graph.

Classification ROC Testing Set



Blue : tunned by the first method

Red : tunned by the second method

Black : tunned by the third method

Green : unprunned tree

Bagged Tree Intro

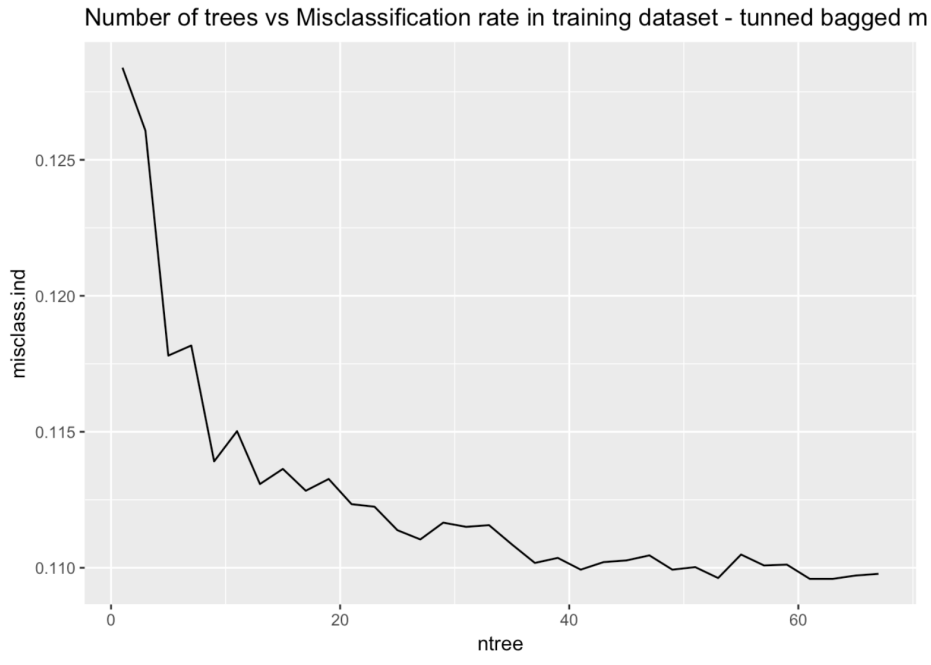
- Originally stemmed from bootstrapping method (compare with boosting)
- Reduce the one of the biggest weakness of decision trees: high variance
- One weaknees is that it is harder to interpret the model
- Special type of random forest when mtry is equal to the number of variables in the dataset
- Packages: randomForest and mlr

Bagged Tree Tuning Parameters

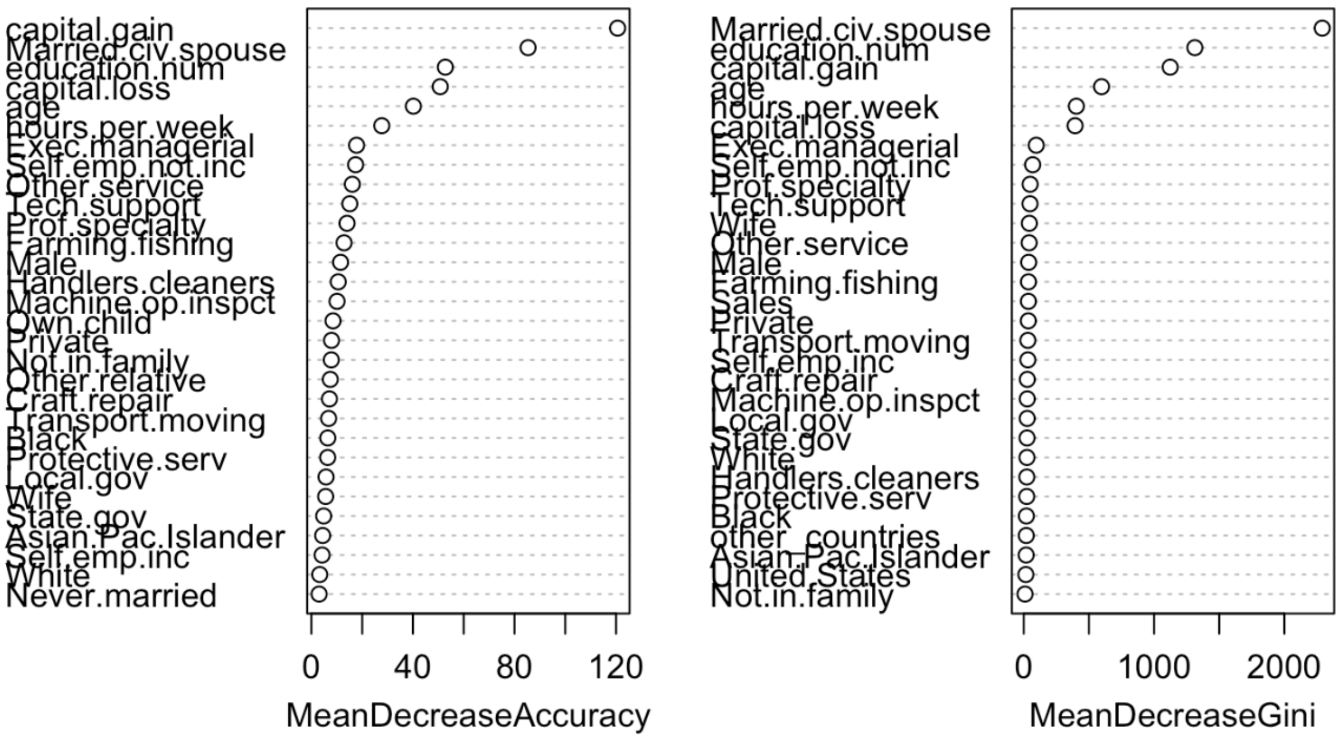
- Number of trees(ntree): Number of trees to grow. Large tree is more computationally expensive.
- Node size(nodesize): Minimum number of observations in the terminal nodes –> directly related to depth of the tree

Tunned Bagged Tree Model

```
##
## Call:
##  randomForest(formula = f, data = data, classwt = classwt, cutoff = cutoff,      ntree = 68L, mtry = 43, impor
tance = TRUE, nodesize = 43L)
##                Type of random forest: classification
##                      Number of trees: 68
## No. of variables tried at each split: 43
##
##         OOB estimate of  error rate: 13.66%
## Confusion matrix:
##       <=50K >50K class.error
## <=50K 22960 1760  0.07119741
## >50K   2666 5016  0.34704504
```

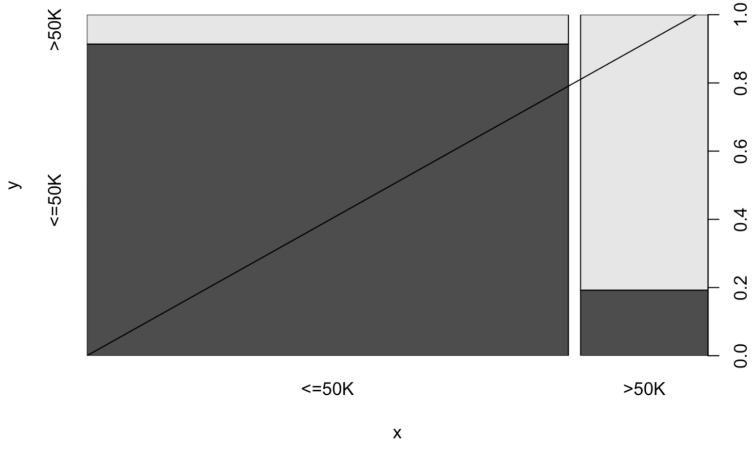


Tunned Bagged Tree Varaible Importance



Confusion Matrix and predicted values on train dataset of Tunned Model

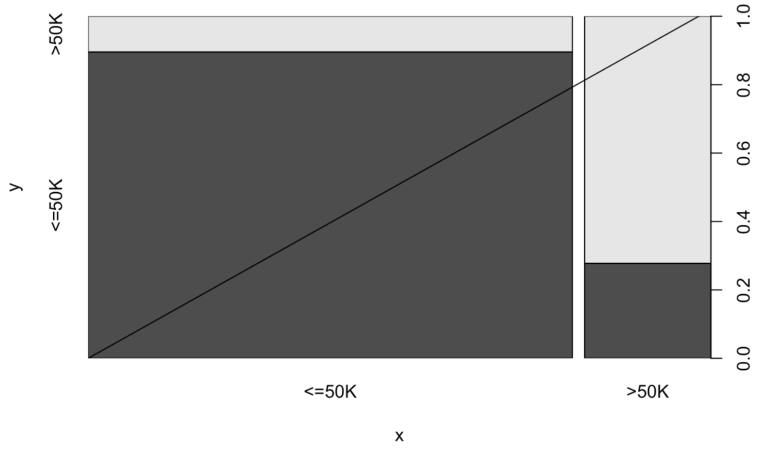
```
Confusion Matrix and Statistics

          Reference
Prediction <=50K  >50K
     <=50K 23414  2203
     >50K   1306  5479

               Accuracy : 0.8917
                 95% CI : (0.8883, 0.8951)
    No Information Rate : 0.7629
    P-Value [Acc > NIR] : < 0.00000000000000022
```



white : >50K, black : <=50K

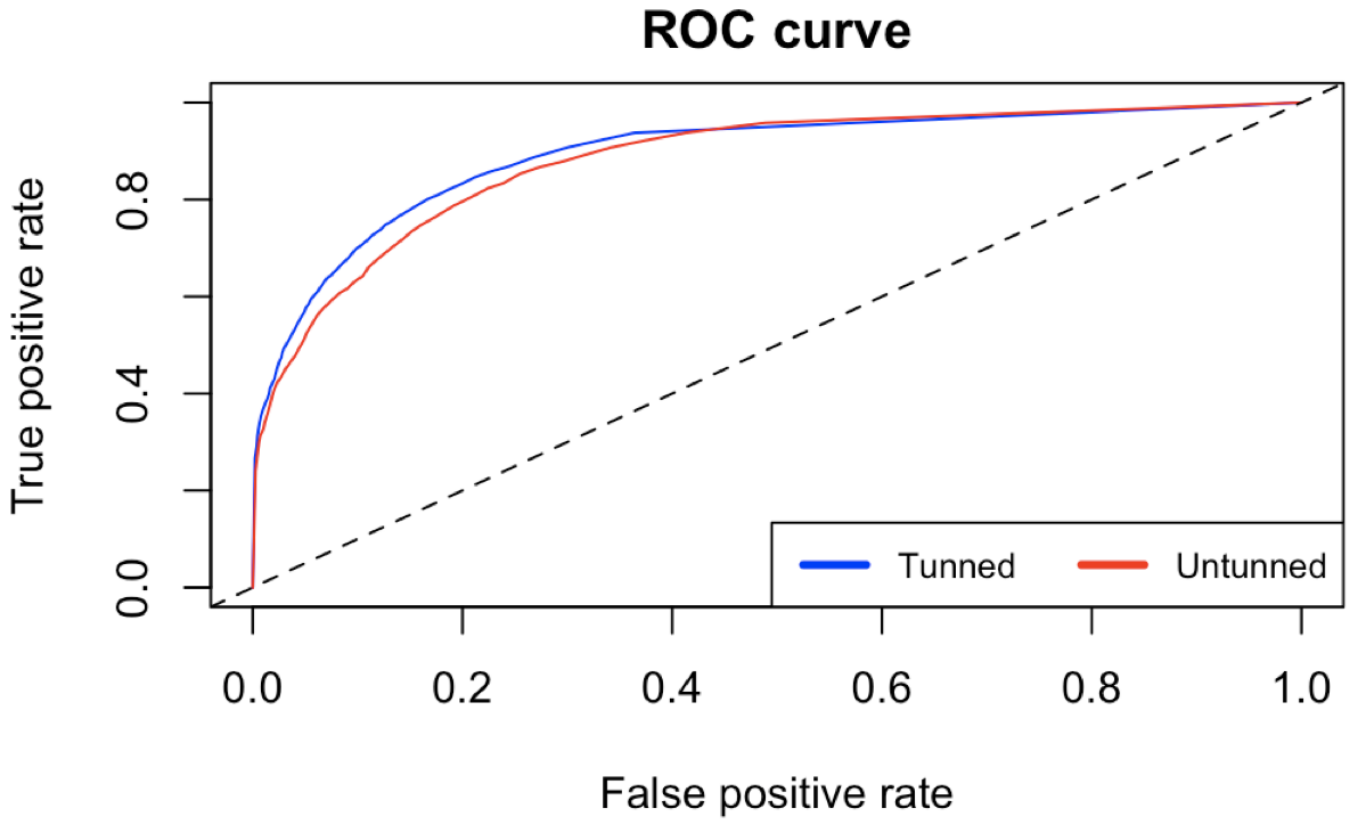
Confusion Matrix and predicted values on test dataset of Tunned Model

```
Confusion Matrix and Statistics

          Reference
Prediction <=50K  >50K
     <=50K 11504  1340
     >50K    931  2421

               Accuracy : 0.8598
                 95% CI : (0.8543, 0.8651)
    No Information Rate : 0.7678
    P-Value [Acc > NIR] : < 0.00000000000000022
```



white : >50K, black : <=50K

Bagged Tree ROC Testing Set



Random Forest Intro

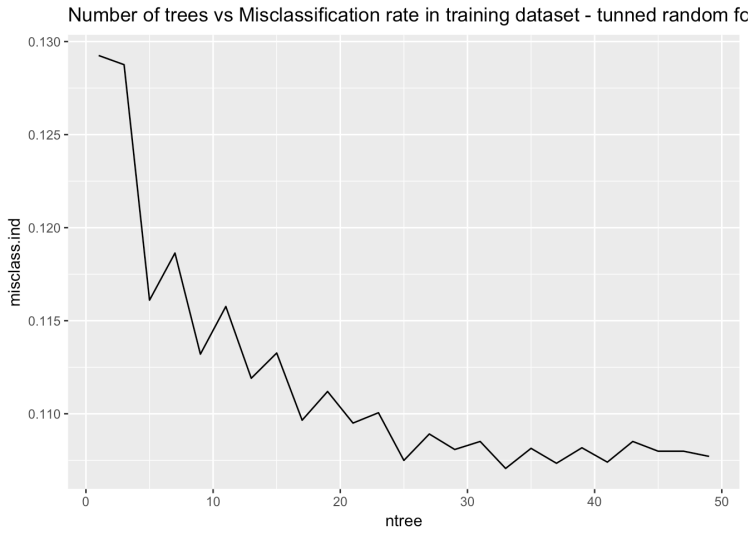
- Another method improving bagged trees by decorrelating the trees
- Only consider some subsets (tuning is needed) of variables out of all.
- If the subset size is p, where p is the number of variables from our data, it is the same as bagging tree.
- Packages: randomForest and mlr

Random Forest Tuning Parameters

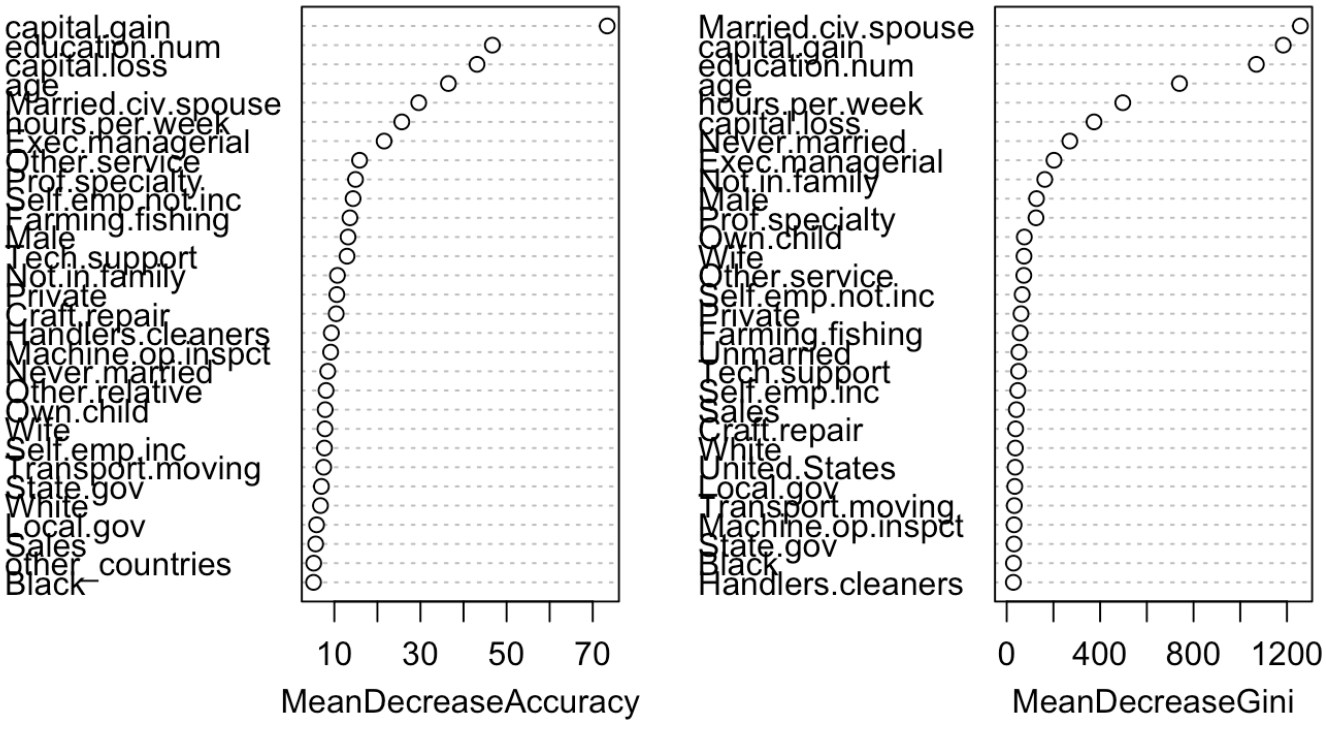
- Number of trees(ntree): Number of trees to grow. Large tree is more computationally expensive.
- Node size(nodesize): Minimum number of observations in the terminal nodes –> directly related to depth of the tree
- Number of variable used for each splitting(mtry) : Number of variables we should select at a node split. P/3 for regression and sqrt(p) for classification tree.

Tunned Random Forest Model

```
##
## Call:
##  randomForest(formula = f, data = data, classwt = classwt, cutoff = cutoff,      ntree = 79L, importance = TRU
E, mtry = 8L, nodesize = 14L)
##                Type of random forest: classification
##                      Number of trees: 79
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 13.47%
## Confusion matrix:
##       <=50K >50K class.error
## <=50K 23063 1657  0.06703074
## >50K   2709 4973  0.35264254
```

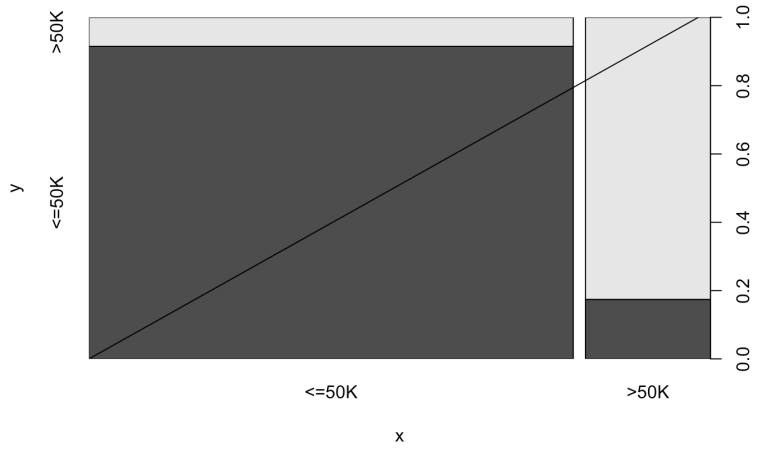


Tunned Random Forest Varaible Importance



Confusion Matrix and predicted values on train dataset of Tunned Model

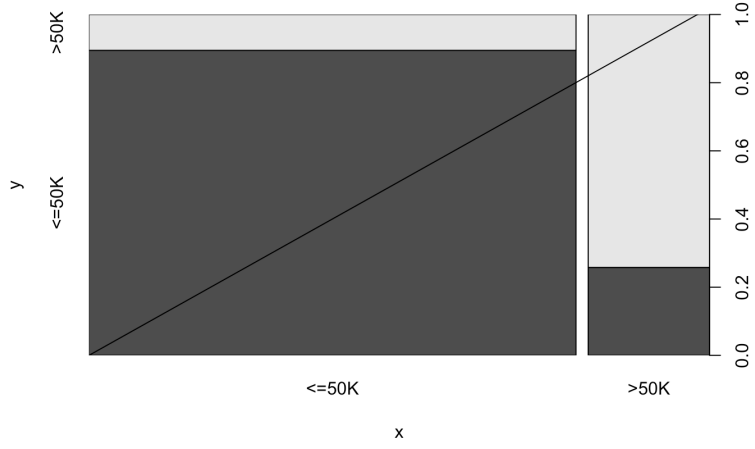
```
Confusion Matrix and Statistics

          Reference
Prediction <=50K  >50K
     <=50K 23561  2189
     >50K   1159  5493

               Accuracy : 0.8967
                 95% CI : (0.8933, 0.9)
    No Information Rate : 0.7629
    P-Value [Acc > NIR] : < 0.00000000000000022
```



white : >50K, black : <=50K

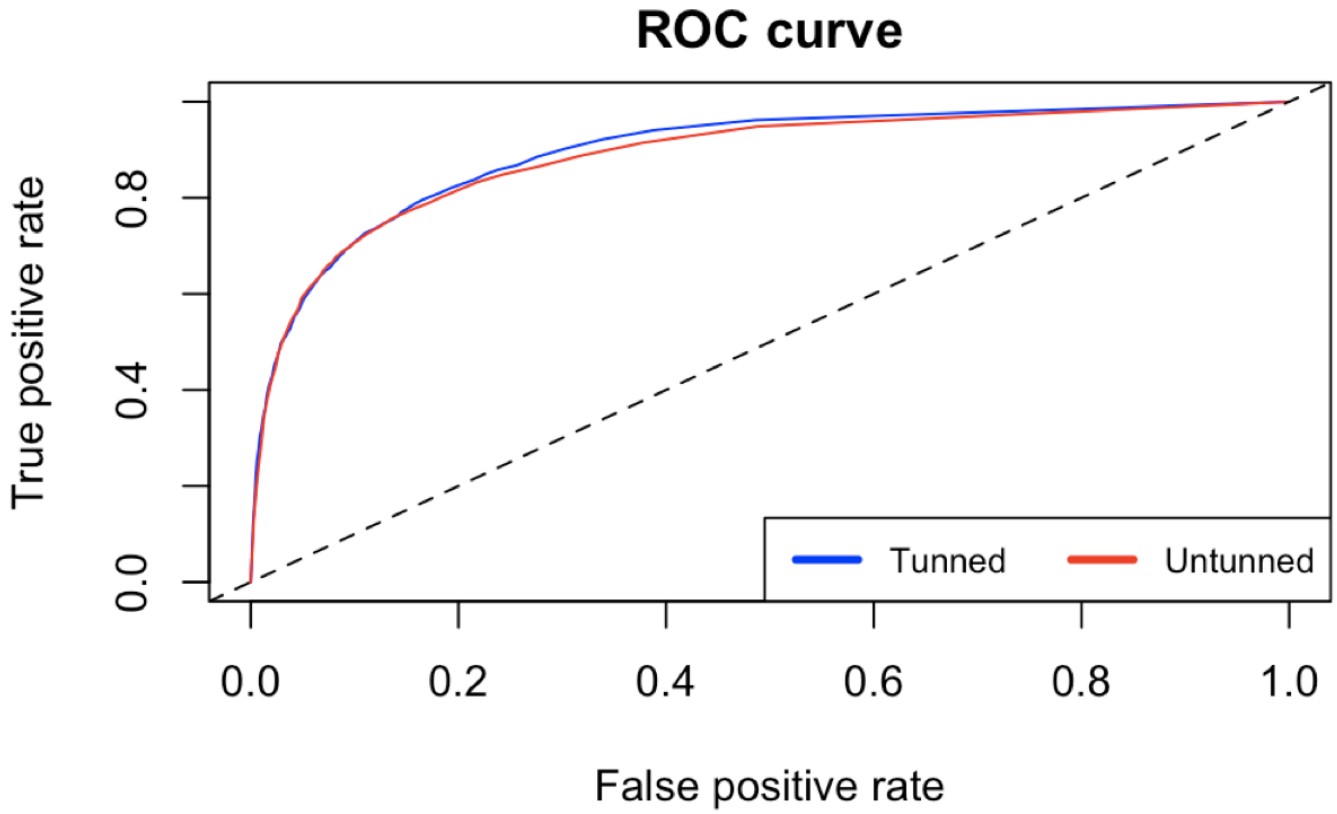
Confusion Matrix and predicted values on test dataset of Tunned Model

```
Confusion Matrix and Statistics

          Reference
Prediction <=50K  >50K
     <=50K 11601  1362
     >50K    834  2399

               Accuracy : 0.8644
                 95% CI : (0.859, 0.8696)
    No Information Rate : 0.7678
    P-Value [Acc > NIR] : < 0.00000000000000022
```



white : >50K, black : <=50K

Random Forest ROC Testing Set



Boosted Tree Intro

- Another method doing the similar way as bagging
- Grow sequentially based on information from previous trees
- Not use bootstrap sampling
- Helps lower the variance by making into weak learners by restricitng the tree depth.
- Packages: bst or gbm

Boosted Tree Tuning Parameter

- Interaction.depth: how slowly to improve/learn the model
- Shrinkage parameter: parameter to allow more and different shaped tree but slowing down the process
- Number of trees

Boosted Tree Models

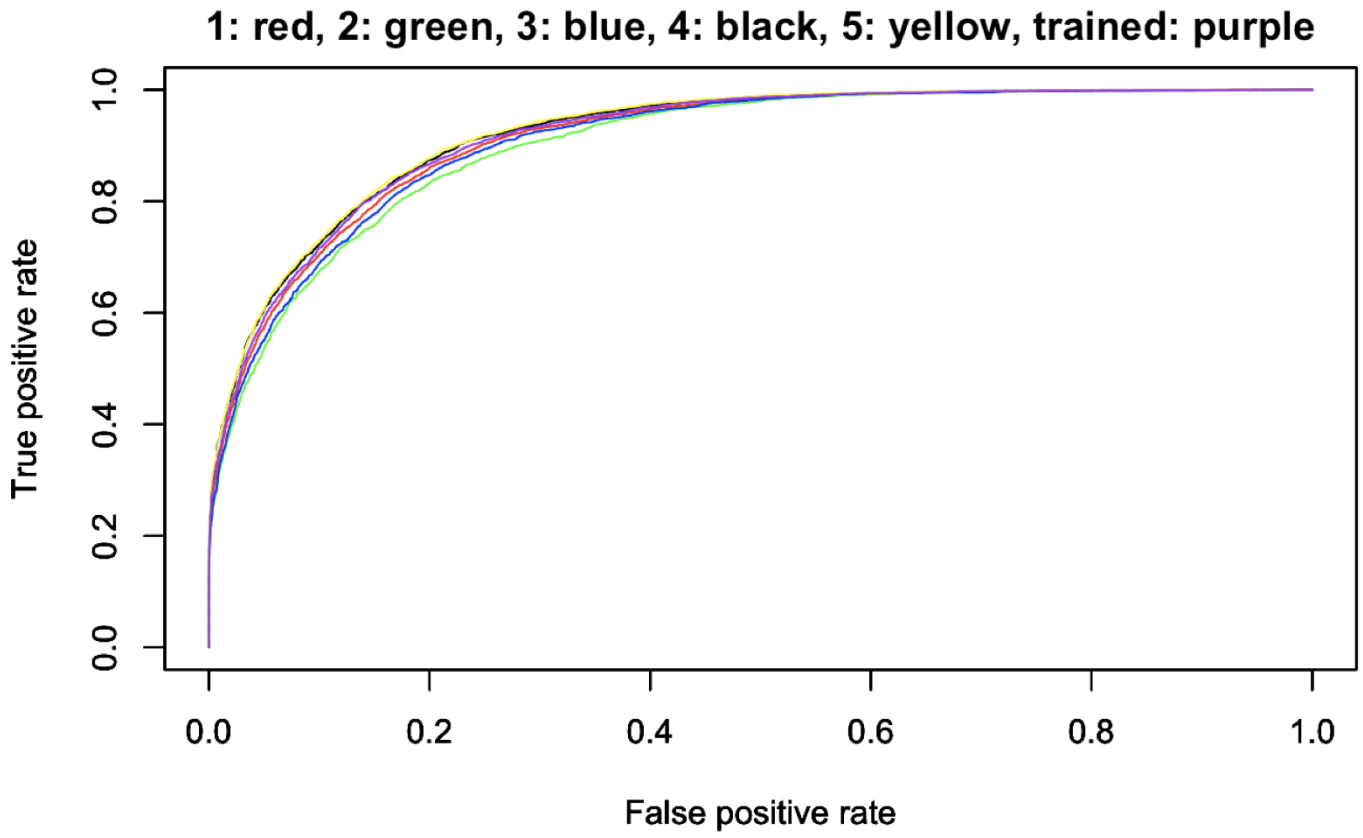
1. number of tree = 5000, interaction depth = 5, and shrinkage parameter = 0.001.
2. number of tree = 2000, interaction depth = 5, and shrinkage parameter = 0.001.
3. number of tree = 5000, interaction depth = 3, and shrinkage parameter = 0.001.
4. number of tree = 5000, interaction depth = 3, and shrinkage parameter = 0.2.
5. number of tree = 5000, interaction depth = 3, and shrinkage parameter = 0.1.
6. number of tree = 150, interaction depth = 3, and shrinkage parameter = 0.1 (Train function)

Boosted Tree Variable Importance

6 big important variables

- Married civ spouse
- education num
- age
- capital gain
- hours per week
- capital loss

Boosted Tree ROC Testing Set



Model Selection

- Two ways to do it

1. AUC: Data imbalance (76.5% of earning less than 50K & 23.5% of earning more than 50K)
2. Testing set accuracy rate: Our goal is to improve prediction accuracy

Model Selection

Model Selection		
	AUC	Test set Accuracy rate
Classification	0.87687	0.86089
Bagged	0.89425	0.86206
Random	0.89624	0.86608
Boosted	0.92319	0.87102

Additional Methods

- Model-based Cluster
- K-mean Cluster
- Hierarchial Cluster
- LDA
- QDA
- MDA

Conclusion

- Rank1: Boosted tree
- Rank2: Random forest
- Rank3: Bagged tree
- Rank4: Classification tree

1

[1] Please look at definition to learn more why we got this conclusion.

How it can be further developed

- Outliers handlings
- Perform PCA in EDA
- More tuning/training

Reference

http://rmarkdown.rstudio.com/authoring_pandoc_markdown.html#images