Census Income Data Set Classification

Members: Zhixin Li, Dodo Qian

Instructors: Gaston Sanchez, Johnny Hong

Introduction

The data set for this project is the Census Income Data Set donated by Ronny Kohavi and Barry Becker to the UCI Machine Learning Repository. It describes 15 variables on a sample of individuals from the US Census database. The prediction task is to determine whether a person makes over 50K a year using the rest 14 variables.

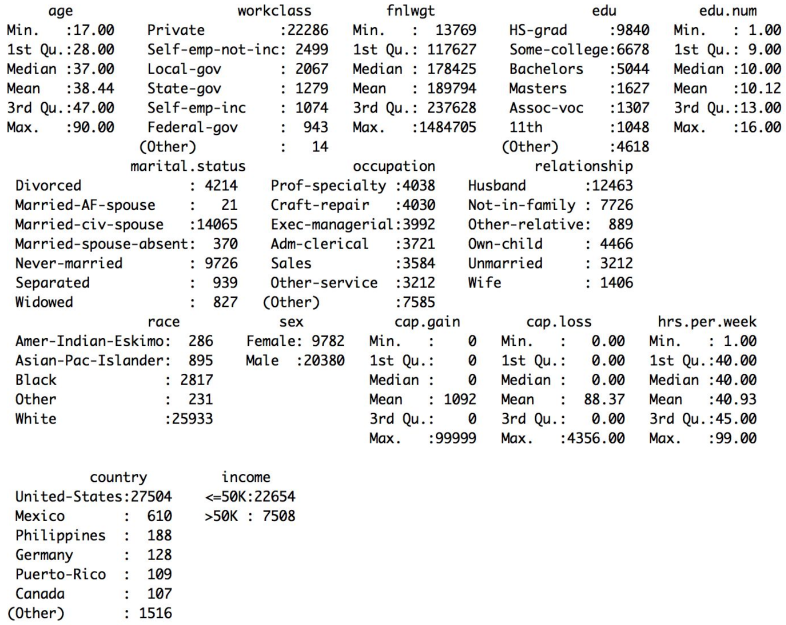
There will be four parts in this project report. In the first part, we analyze the data set in an exploratory way to understand basic properties of the data. We try to show the simple relationship between features and income variable visually so that we could simplify and regroup the data set in a cleaning way. In the second part, we analysis the data by applying different models. Keep in mind that our goal is to predict whether a person could make over 50K per a year based on the other variables. We used five different models (Classification Tree, Bagged Tree, Random Forest, Neural Networks, Support Vector Machine) to see which one gives us the best result based on the highest test accuracy rates and AUC values. In the last two parts, we select the best model based on the result and make conclusion.

Exploratory Data Analysis (EDA)

Basic Description and Cleaning of the Dataset

The Census Income data contains 48842 instances including unknown values, and contains 45222 if the unknown values are removed (30162 observations for train set, 15060 observations for test set). Since the proportion of the unknown values, which is a small number 7.41%, therefore, it is confident for us to say that dropping the unknown values will not affect our prediction.

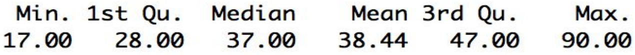
Among the whole data, both train set and test set contain 15 variables ("age", "workclass", "fnlwgt", "edu", "edu.num", "marital.status", "occupation", "relationship", "race", "sex", "cap.gain", "cap.loss", "hrs.per.week", "country", "income"), which represent different meanings. By applying common sense and distribution graphs, we are going to explain how and why we preprocess the datasets before we do any predictions. The following graphs and statistics pertain to the train set (census_train).



```
      age                   workclass          fnlwgt                  edu           edu.num     
 Min.   :17.00    Private         :22286   Min.   :  13769    HS-grad      :9840   Min.   : 1.00  
 1st Qu.:28.00    Self-emp-not-inc: 2499   1st Qu.: 117627    Some-college:6678   1st Qu.: 9.00  
 Median :37.00    Local-gov       : 2067   Median : 178425    Bachelors    :5044   Median :10.00  
 Mean   :38.44    State-gov       : 1279   Mean   : 189794    Masters      :1627   Mean   :10.12  
 3rd Qu.:47.00    Self-emp-inc    : 1074   3rd Qu.: 237628    Assoc-voc    :1307   3rd Qu.:13.00  
 Max.   :90.00    Federal-gov     :  943   Max.   :1484705    11th         :1048   Max.   :16.00  
                 (Other)          :   14                     (Other)      :4618                  
              marital.status            occupation            relationship  
  Divorced             : 4214    Prof-specialty :4038    Husband       :12463  
  Married-AF-spouse    :   21    Craft-repair   :4030    Not-in-family : 7726  
  Married-civ-spouse   :14065    Exec-managerial:3992    Other-relative:  889  
  Married-spouse-absent:  370    Adm-clerical   :3721    Own-child     : 4466  
  Never-married        : 9726    Sales          :3584    Unmarried     : 3212  
  Separated            :  939    Other-service  :3212    Wife          : 1406  
  Widowed              :  827   (Other)         :7585                         
                 race            sex             cap.gain        cap.loss        hrs.per.week  
  Amer-Indian-Eskimo:  286   Female: 9782   Min.   :    0   Min.   :   0.00   Min.   : 1.00  
  Asian-Pac-Islander:  895   Male  :20380   1st Qu.:    0   1st Qu.:   0.00   1st Qu.:40.00  
  Black             : 2817                  Median :    0   Median :   0.00   Median :40.00  
  Other             :  231                  Mean   : 1092   Mean   :  88.37   Mean   :40.93  
  White             :25933                  3rd Qu.:    0   3rd Qu.:   0.00   3rd Qu.:45.00  
                                            Max.   :99999   Max.   :4356.00   Max.   :99.00  

          country          income     
  United-States:27504    <=50K:22654  
  Mexico       :  610    >50K : 7508  
  Philippines  :  188                 
  Germany      :  128                 
  Puerto-Rico  :  109                 
  Canada       :  107                 
 (Other)       : 1516                 
```

Figure 1: summary of census_train income dataset

- **age**: the age of each individual (integer greater than 0)



```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  17.00   28.00   37.00   38.44   47.00   90.00 
```

Table 1: summary of “age” variable

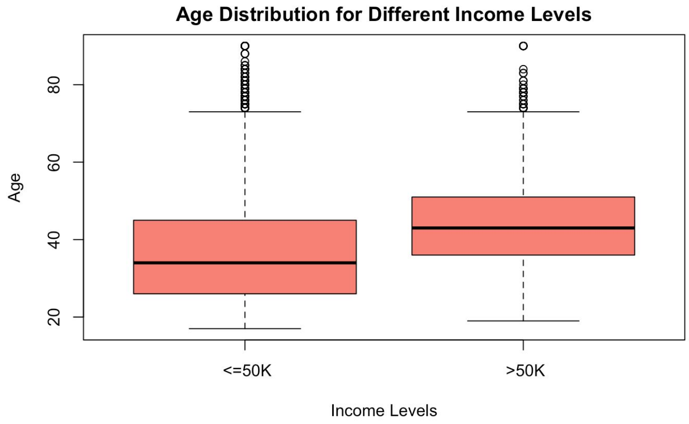


Figure 2: Age Distribution for Different Income Levels

From the plot, we can see that a majority of the working people are aged from 25 to 65. The average age of income <=50K is below 30-year-old while the average age of income >50K is above 42-year-old. This leads to the assumption that experience definitely matters the income level. Therefore, we'd better leave this variable and not to group it in this case.

- **workclass**: employment status of each individual ("Federal-gov", "Local-gov", "Never-worked", "Private", "Self-emp-inc", "Self-emp-not-inc", "State-gov", "Without-pay", total is 8 levels)

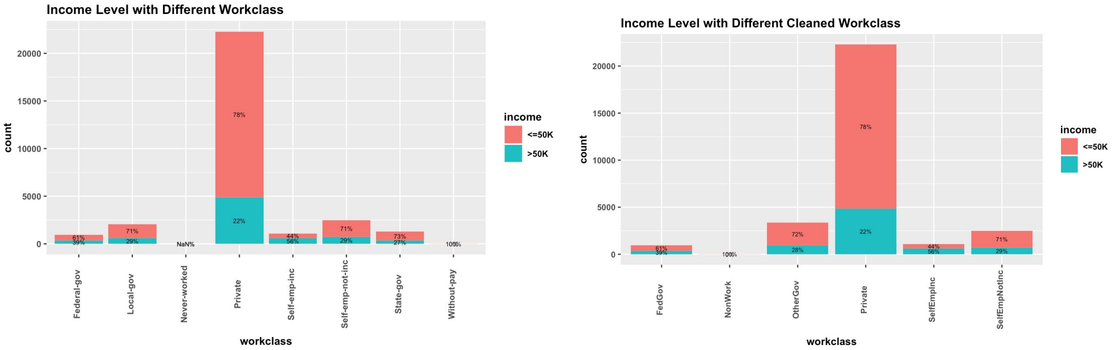


Figure 3: income level with different workclass before and after grouping

```
      FedGov       NonWork      OtherGov       Private    SelfEmpInc SelfEmpNotInc 
         943            14          3346         22286          1074          2499 
```

Table 2: summary of grouped "workclass" variable

From Figure 3, except federal government and self-emp-inc, the probability of making >50K are similar among all other work classes. Federal government is seen as the elite in public sector, and self-emp-inc means that individuals own their own company, so it is reasonable that both work classes earn more than 50K. We can see from the data that "Local-gov" and "State-gov" present similarly, so we can group them together as "OtherGov". We could also group "Never-worked" and "Without-pay" as a group since the proportions of them is equal to or nearly zero. After cleaning the “workclass” variable, we have the following:

- **fnlwgt**: final weight of each individual (continuous integer)

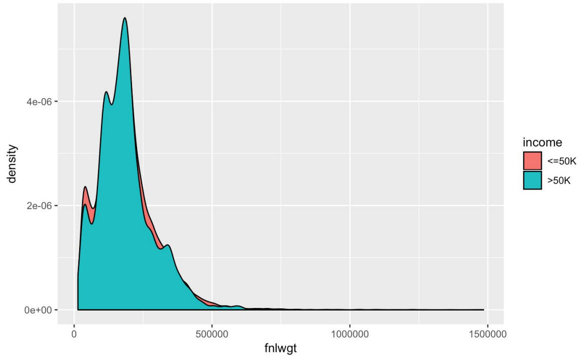
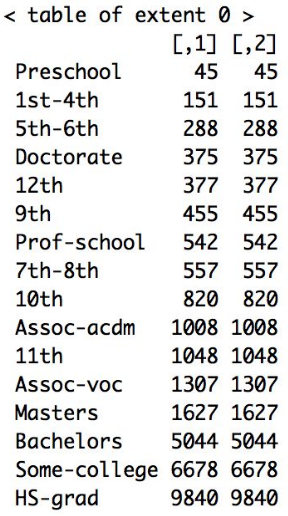


Figure 4: density of “fnlwgt” in two different income levels

We can see from the graph that most of the data from "fnlwgt" can separate into "<=50K" and ">50K" groups equally (two densities of different income level are almost overlapping each other). This means it does not matter for us to discard “fnlwgt” variable since it won't affect our prediction. Besides, for the simplified purpose, we also should discard this variable.

- **edu** / **edu.num**: the highest level of education achieved by an individual (“10th”, “11th”, “12th”, “1st-4th”, “5th-6th”, “7th-8th”, “9th”, “Assoc-acdm”, “Assoc-voc”, “Bachelors”, “Doctorate”, “HS-grad”, “Masters”, “Preschool”, “Prof-school”, “Some-college”) / the highest level of education achieved by an individual in numerical form (integer range from “1” to “16”)



```
< table of extent 0 >
             [,1] [,2]
 Preschool      45   45
 1st-4th       151  151
 5th-6th       288  288
 Doctorate     375  375
 12th          377  377
 9th           455  455
 Prof-school   542  542
 7th-8th       557  557
 10th          820  820
 Assoc-acdm   1008 1008
 11th         1048 1048
 Assoc-voc    1307 1307
 Masters      1627 1627
 Bachelors    5044 5044
 Some-college 6678 6678
 HS-grad      9840 9840
```

Table 3: comparison of “edu” and “edu.num” variables

We can see that both "edu" and "edu.num" show us the same information, and it is enough for us to select one of them to do the prediction. We are going to select "edu" variable.

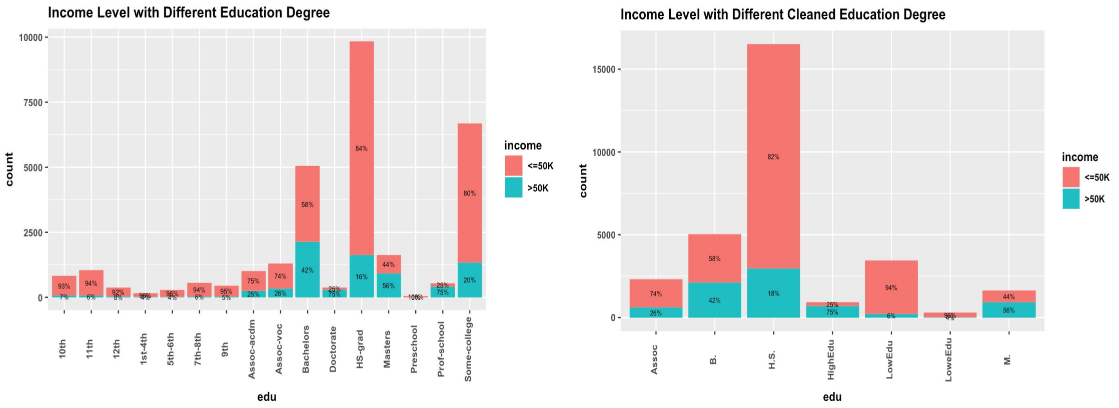


Figure 5: income level with different education degree before and after grouping

```
Assoc       B.     H.S.  HighEdu   LowEdu  LoweEdu       M. 
 2315     5044    16518      917     3453      288     1627 
```

Table 4: summary of grouped “edu” variable

From Figure 5, most people have at most a high school degree and a few have a doctorate degree, which is a fair representation of the society. It is easy to notice that the number of people of making greater than $50,000 a year increase as the years of education increases. For those who don’t have any forms of college education (less than or equal to 8 years of education), less than 10% have an annual income of greater than $50,000. While for those with doctorate degrees or go to professional school, nearly 3/4 makes greater than $50,000 a year. It is reasonable for us to group "10th", "11th", "12th", "1st-4th", "5th-6th", "7th-8th", "9th" and "Preschool" as "LowEdu" group, "Assoc-acdm" and "Assoc-voc" as "Assoc" group, "HS-grad" and "Some-college" as "H.S." group, "Bachelors" as "B." group, "Masters" as "M." group, and "Doctorate" and "Prof-school" as "HighEdu" group.

- **marital.status**: marital status of each individual (“Divorced”, “Married-AF-spouse”, “Married-civ-spouse”, “Married-spouse-absent”, “Never-married”, “Separated”, “Widowed”)
  - “Married-AF-spouse”: married to a Armed Forces spouse
  - “Married-civ-spouse”: married to a civilian spouse

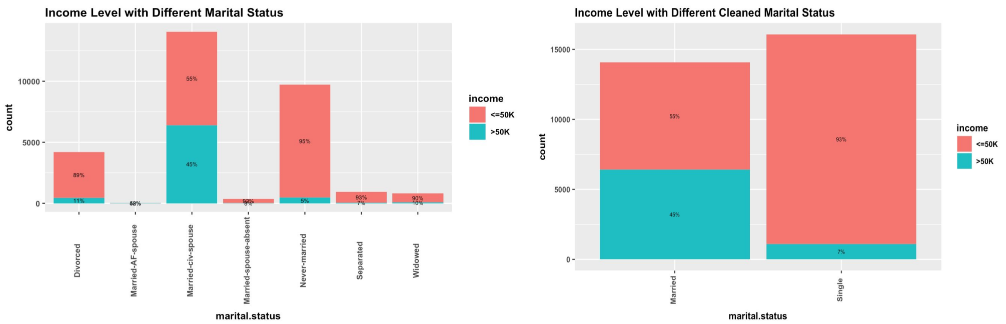


Figure 6: income level with different marital status before and after grouping

We can see "Divorced", "Married-spouse-absent", "Never-married", "Separated" and "Widowed" are very similar to each other in the distribution graph, and they are actually representing "Single" in the reality, so we classify them as a "Single" group. As for the "Married-AF-spouse" and "Married-civ-spouse", we are also see that they are very similar to each other, so we are going to group them together as "Married" group.

- **occupation**: occupation of each individual (“Adm-clerical”, “Armed-Forces”, “Craft-repair”, “Exec-managerial”, “Farming-fishing”, “Handlers-cleaners”, “Machine-op-inspct”, “ Other-service”, “Priv-house-serv”, “Prof-specialty”, “Protective-serv”, “Sales”, “Tech-support”, “Transport-moving”)

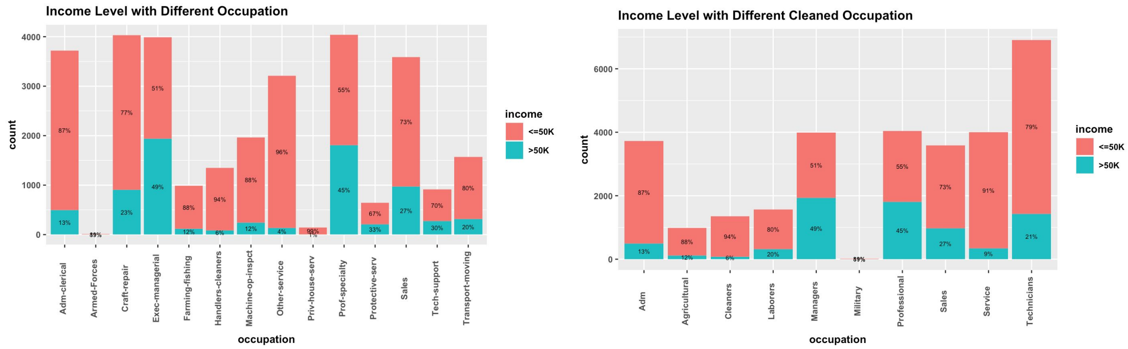


Figure 7: income level with different occupation before and after grouping

```
         Adm  Agricultural      Cleaners      Laborers      Managers      Military  Professional 
        3721           989          1350          1572          3992             9          4038 
       Sales       Service   Technicians 
        3584          3999          6908 
```

Table 5: summary of grouped “occupation” variable

From Figure 7, exec-management and prof-speciality have high probability earning more than 50K. But the probabilities for farming-fishing, handles-cleansers, and other service are greatly lower than the others. We grouped "Craft-repair", "Machine-op-inspct", and "Tech-support" as "Technicians" since technicians are people who either skilled in the technique of an art or craft or an expert in the practical application. Those three jobs satisfied the definition of technicians and their distributions are similar. "Other-service", "Priv-house-serv", and "Protective-serv" are grouped to

"Service" since they perform similar type of job. Other jobs are renamed for easy understanding.

- **relationship**: shows how individual is relative to others (“Husband”, “Not-in-family”, “Other-relative”, “Own-child”, “Unmarried”, “Wife”)

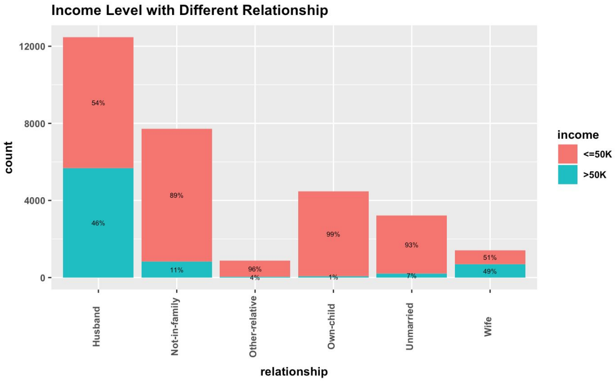


Figure 8: income level of different relationship

We can see from the figure that the distributions of “Husband” and “Wife” are similar, and they can be groupe into one group as “Married” based on sex. As for the rest of levels, we can see they are similar to each other, and can be grouped as “Single”. Since we have had two variables “marital.status” and “sex” could represent this “relationship” variable, so it’s better for us to discard this variable to simply the dataset.

- **race**: descriptions of each individual's race (“Amer-Indian-Eskimo”, “Asian-Pac-Islander”, “Black”, “Other”, “White”)

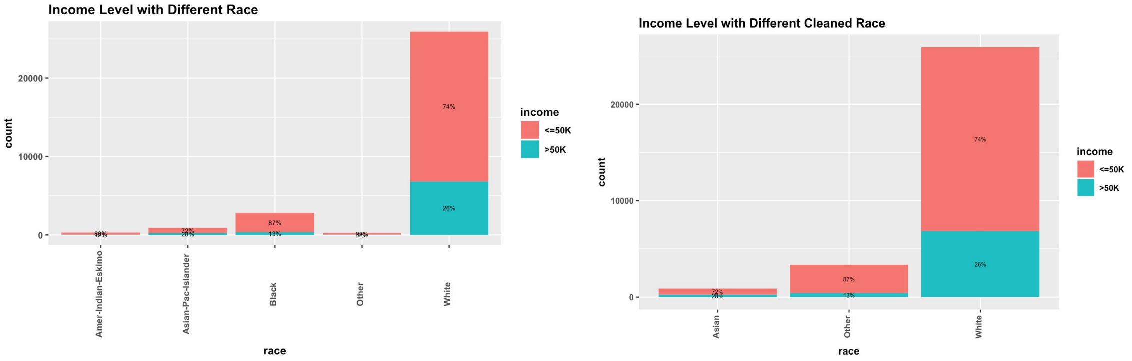


Figure 9: income level with different race before and after grouping

We can see the proportions of “Black”, “Amer-Indian-Eskimo” and “Other” are very similar to each other and they represent the minority group, so we decide to group them as a group “Other”.

- **sex**: the biological sex of each individual (“male”, “female”)

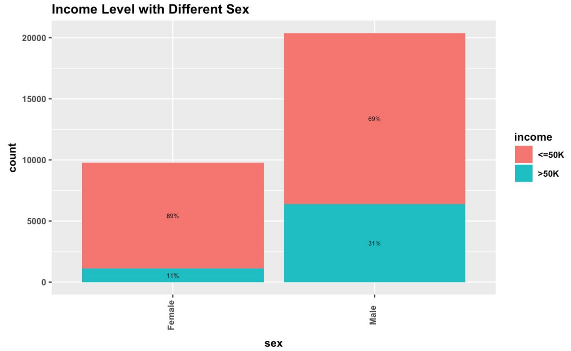


Figure 10: income level with different sex

The proportions are largely different from each other, this variable might be very important in the following prediction. Therefore, we are going to leave it untouched.

- **cap.gain** and **cap.loss**: capital gain and loss from financial investments for each individual (continuous variables)

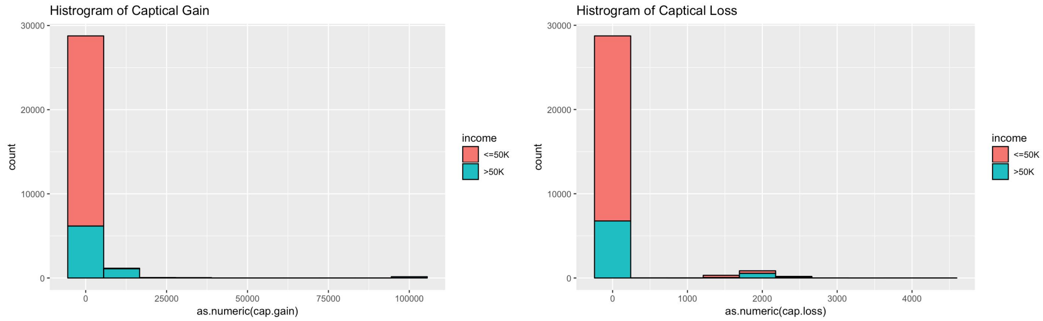


Figure 11: income level with cap.gain and cap.loss

We can see that both histograms show these two variables are highly screwed. Therefore, we are going to set there groups for these two variables: None, Low and High representing “0”, high gain/loss, and low gain/loss.

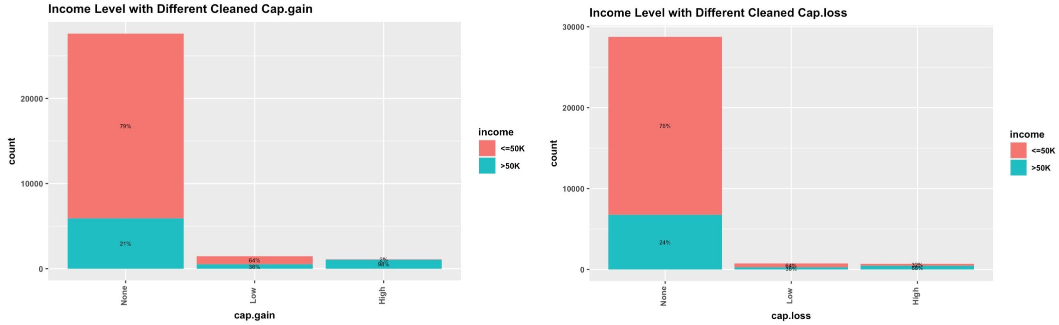


Figure 12: income level with different grouped cap.gain and cap.loss

- **hrs.per.week**: the hours of individual work per week (integer greater than 0)

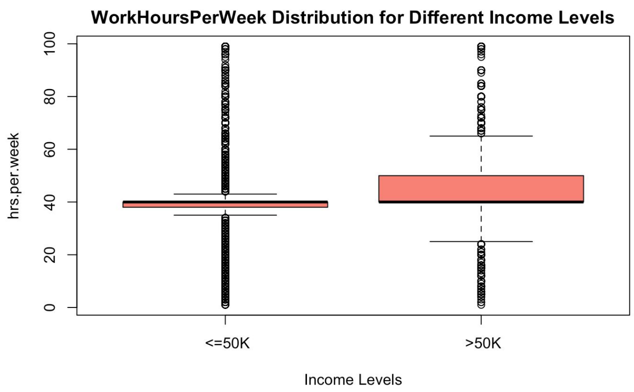


Figure 13: WorkHoursPerWeek distribution for different income levels

We can see from the graph that the majority of individuals are working 40 hour weeks which is expected as the societal norm, and more work hours results in more income. Besides, even there are many outliers in the graph, we yet decided to keep them all since we haven't been sure whether those outliers are relevant to the prediction or not.

- **country**: country region for each individual

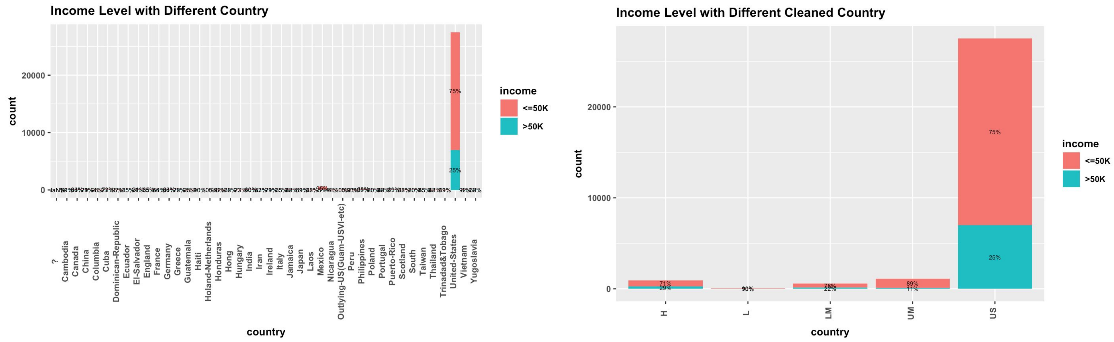


Figure 14: income level with different country before and after grouping

There are too many categories in the country variable, which makes it difficult to build trees. Since the goal is to predict the income, and the income level is varied among different countries, depending on the economic development, so we grouped the countries based on GNI per capita (dollar value of a country's final income in a year, divided by its population, reflects the average income of a country's citizens) in 2016, calculated using the World Bank Atlas method. The countries are grouped into low-income(L) economies($1,005 or less), lower middle-income(LM) economies($1,006 to $3,955), upper middle-income(UM) economies($3,956 to $12,235) and high-income(H) economies($12,236 or more). United States(US) has a large number of population, so it is separated out as a single category.

- income: there are two different levels of income variable (">=50K", "<50K"), we replaced ">=50K" as "1" and "<50K" as "0".

After the preprocessing, we will get a cleaned dataset containing 11 variables (“income” is taken as y variable; “fnlwgt”, “edu.num” and “relationship” are dropped). In the following, we will use different ways to build predicting models and select the best model by comparing their AUC area.

Analysis and Methods (model building)

Three different kinds of decision trees were mainly used in the prediction: *classification tree*, *bagged tree* and *random forest*. And two more additional methods will also be applied: *Support Vector Machine* and *Neural Network*.

Classification Tree

Classification tree is used to predict a qualitative response based on recursive partitioning of the independent variables. The partitioning will divide the space into smaller and smaller regions till possibly reaching homogeneous regions. The observation is predicted to a specific class by using majority rule, namely, most commonly occurring class. When building a classification tree, Gini Index, cross-entropy or classification error rate can be used to assess the quality of splits and prune the tree. Pruning is needed to prevent overgrown tree using the training data since a very large tree can overfit the training data.

In R, *rpart()* from “rpart” package and *prune()* are used to build the classification tree and prune the tree.

We first built the tree with all 11 variables. In order to find the minimum cross-validated error, we used *printcp()* and *plotcp()* to display the cp(complex parameter) table and visualize cross-validation results. We selected the complex parameter that corresponds with the minimum CV error (here cp=0.010000), and placed it into *prune()* function. The pruned model turned out to have the same tree structure as unpruned model, and it was applied to the training set to calculate the training accuracy, which is **83.33%**.

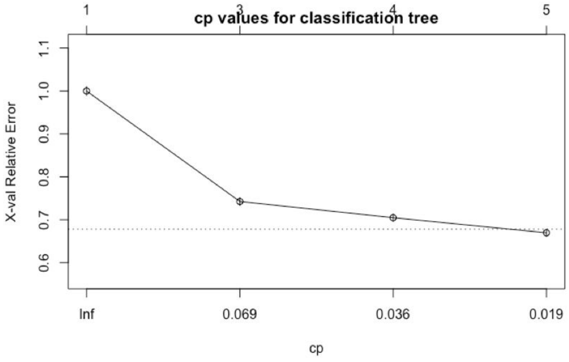


Figure 15: Cp values and its CV-error rate

To find the important variables, we applied *varImp()* on the model, and “cap.gain”, “edu”, “marital.status”, “occupation”, “age” and “hrs.per.week” are the important variables in decreasing order (Figure 17).

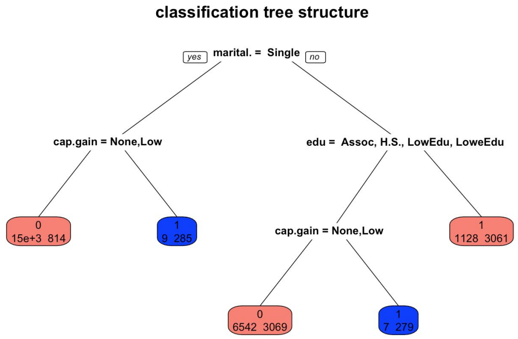


Figure 16: classification tree from *rpart()*

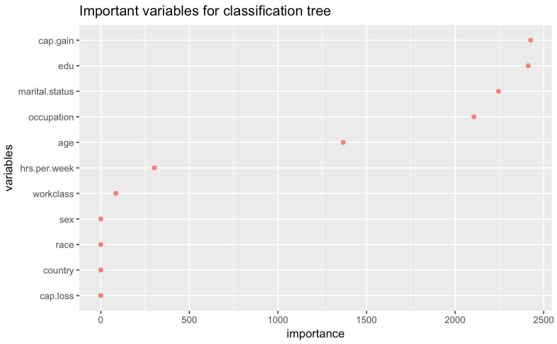


Figure 17: important variables for classification tree

Besides using the classic cross-validation method to prune the classification tree, we also try a different way to prune it. It basically looking for the optimal cp value based on the accuracy training rate first, and then plot the classification trees applying two different splitting ways as criterion (“information” and “gini”) based on the optimal tuning parameter.

Before training the census_train dataset, we use function *trainControl()* (setting *method = “repeatedcv”* to do the repeated cross-validation) to get the tuning parameter (cp value), and store it as “trctrl”. In the following step, we use function *train()* and set *trControl = trctrl, split = "information"* to train the training dataset. From here, we can get Figure 18, which shows us the optimal tuning parameter (cp value). Form the graph, the tuning parameter which gives us the highest accuracy will be the optimal tuning parameter, we can look for the highest accuracy and the corresponding cp value is the optimal complexity parameter. After that, we used function *prp()* to plot the classification tree (optimal complexity parameter (cp = 0.001331913)).

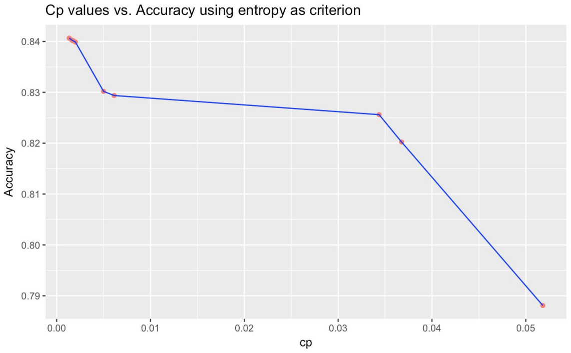


Figure 18: Cp values vs. training accuracy using entropy

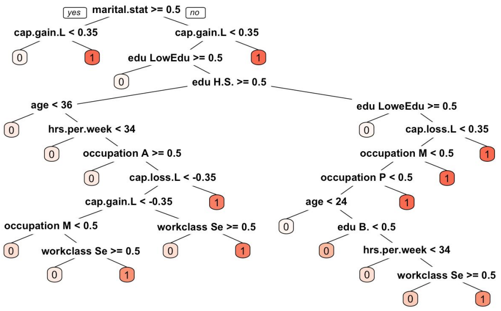


Figure 19: classification by using entropy as criterion

We also using the same method to get another classification tree except setting split = “gini” this time. We can see from the plot that the optimal complexity parameter is cp = 0.001198721.

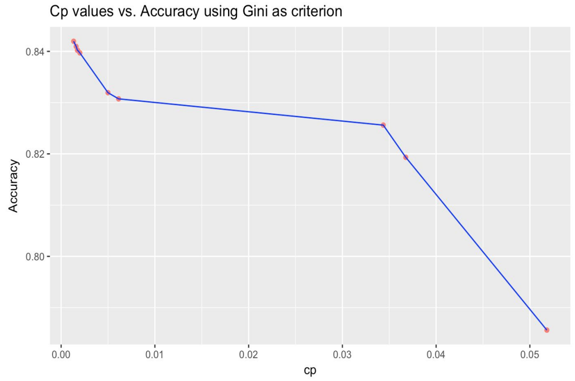


Figure 20: Cp values vs. training accuracy using Gini index

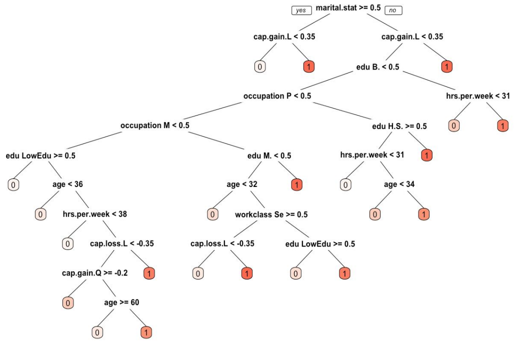
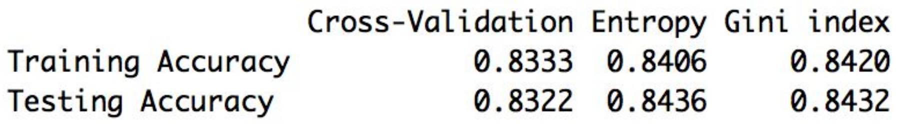


Figure 21: classification by using Gini index as criterion



```
                  Cross-Validation Entropy Gini index
Training Accuracy           0.8333  0.8406     0.8420
Testing Accuracy            0.8322  0.8436     0.8432
```

Table 6: three prune methods all based on CV error

The training and testing accuracies are very similar for the above three methods, since ROC curve can only be obtained from Cross-Validation method, so we decided to choose it as our best model.

Bagged tree

Bagging, or bootstrap aggregating, can lower the variance of classification tree by taking repeated samples from the training data set, building a separate prediction model for each training set and averaging the resulting predictions. At each split, all the predictors are considered The observations that do not use to fit the model are called Out-of-bag(OOB) observations, which can be used to estimate the test error for the bagged model. The interpretation of a collection of bagged trees is more difficult than a single tree, but it improves the accuracy. Generally, pruning will hurt performance of bagged trees, so we chose not to prune it. Another reason is the number of predictors is fixed here, so don't need to tune this parameter.

In R, *randomForest()* from "randomForest" package is used with *mtry = total # of predictors*.

We built trees with *randomForest()* with mtry=11(the total number of variables), and decided to tune Maxnodes for tree pruning by manually doing a 5-fold CV on census_train data set for different maxnodes values (we chose 5-fold instead of 10-fold to reduce computing time). From Figure 22, the minimum CV error achieves at Maxnodes=48, so we add this parameter to form the pruned model, which has a training accuracy **84.33%**.

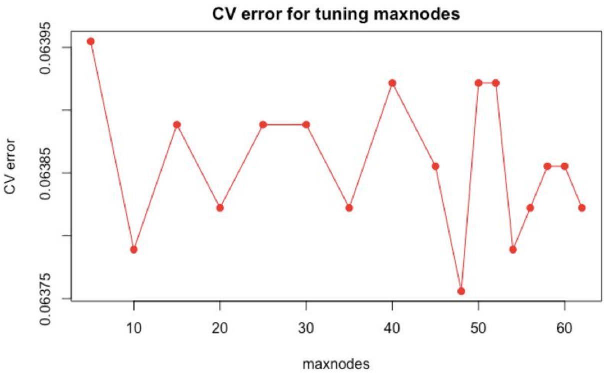


Figure 22: CV error for tuning Maxnodes

From Figure 23, the Mean Decrease Accuracy shows how much the model accuracy decreases if we drop that variable. The Mean Decrease Gini is a measure of variable importance based on the Gini impurity index used for the calculation of splits in trees. The important variables are these two methods are slightly different: Accuracy--“cap.gain”, “marital.status”, “edu”, “occupation”, “age” and “cap.loss”; Gini--“marital.status”, “age”, “edu”, “hrs.per.week”, “cap.gain” and “occupation”.

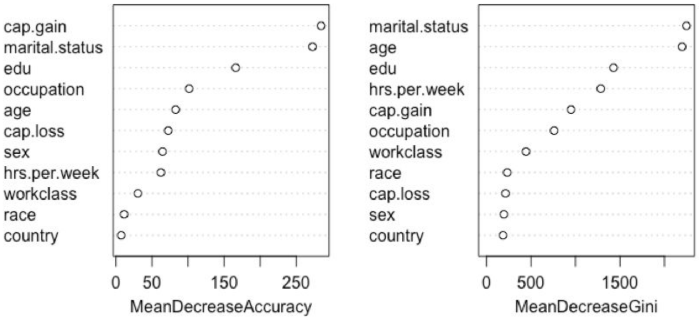


Figure 23: Important variables for bagged tree

Random Forest

Random forest is an improvement over bagged tree by a simple tweak that decorrelates the sub-trees. It changes the algorithm that variables searching is limited to a random sample of m variables out of total p variables. We typically choose m to be around square root of p. Some parameters can be used to prune the tree: mtry(number of variables considered at each split) and ntree(number of trees to grow).

In R, *randomForest()* from “randomForest” package is used to build the tree.

We used *randomForest()* with default mtry=3 to build the tree. Consider the size of this data set, ntree does not have a great influence on the performance of the trees. This can be seen from Figure 24, the OOB error settles when ntree=100, so we use ntree=200. Too many trees does not degrade prediction performance, only computational cost. This is because random forest achieves a lower test error solely by variance reduction. Therefore increasing the number of trees won't have any effect on the bias of your model.

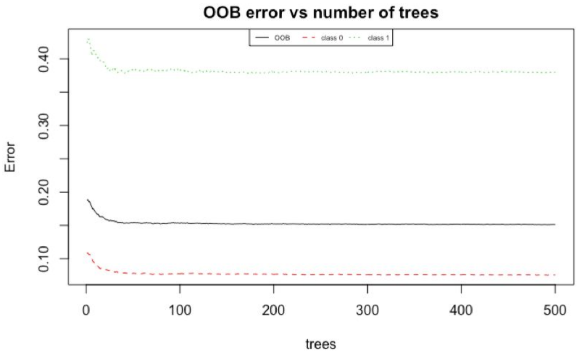


Figure 24: OOB error for different ntree values (black line)

We decided to prune the tree with parameter mtry. Two different approaches are used to tune mtry: *tuneRF()* and manually dp a 5-fold CV. *tuneRF()* is a function that uses OOB error to find the optimal mtry parameter starting with the default value of mtry, and it gives mtry=3 as the optimal option (Figure 25). After performing a manual 5-fold CV on census_train data set, the minimum CV error corresponds with mtry=3. Both methods show that mtry=3 is the best (Figure 26), which is the same as the default value in *randomForest()*, so our original model is the best one for random forest, and the training accuracy is 90.92%.

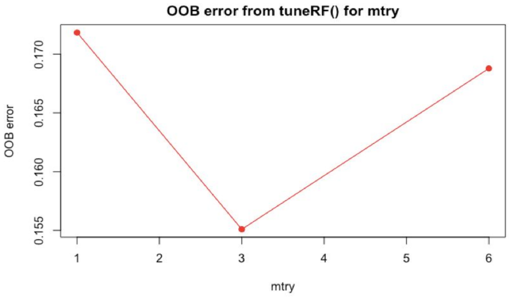


Figure 25: OOB error for mtry

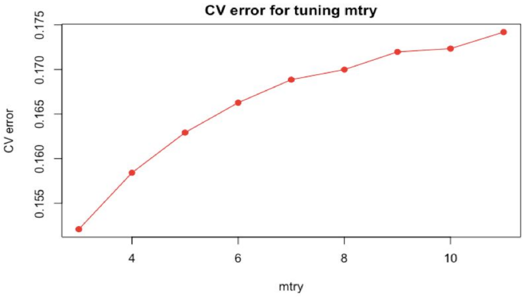


Figure 26: CV error for mtry

The important variables based on two methods are slightly different: Accuracy--“cap.gain”, “marital.status”, “edu”, “occupation”, “age” and “cap.loss”; Gini--“marital.status”, “age”, “edu”, “cap.gain”, “occupation” and “hrs.per.week” (Figure 27).

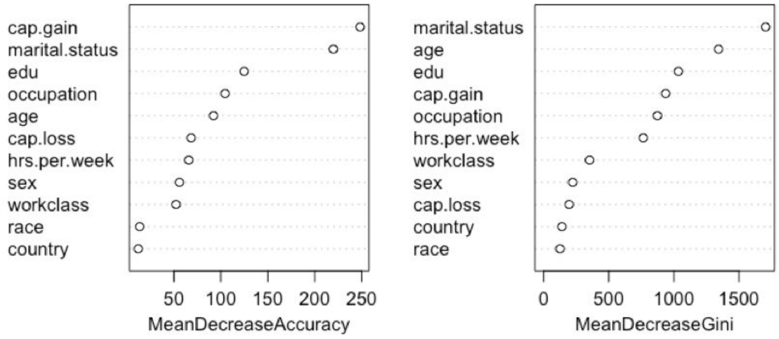


Figure 27: Important variables for random forest

Support Vector Machine

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. It is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyperplane that differentiate the two classes very well. Support Vectors are simply the coordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyperplane/ line). In this project, we used function *ksvn()* from package kernlab to fulfill our goal.

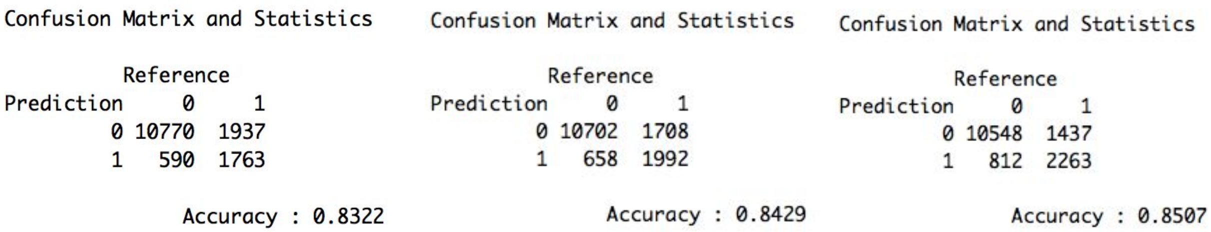
Neural Network

"Neural Network" (NN), is a learning algorithm that is inspired by the structure and functional aspects of biological neural networks. Computations are structured in terms of an interconnected group of artificial neurons, processing information using a connectionist approach to computation. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs, to find patterns in data, or to capture the statistical structure in an unknown joint probability distribution between observed variables. We also used it to do the classification in this project. *nnet()* function from nnet package works well in this case.

Model Selection

The best model in each method are compared based on training accuracy, test accuracy and AUC (mostly rely on AUC since AUC is generated by different thresholds).

From the confusion matrix (Table 7&8), the correct prediction of “0”(income<=50K) are similar among all the models. The correct prediction of “1”(income>50K) counts more towards the difference in accuracy. From Table 9, test accuracies are lower than the training accuracies, and this is reasonable since the model is built based on the training set, so it should have a higher accuracy for training set. Random forest has the highest accuracy for both training and test sets. AUC(area under the curve) measures how TPR and FPR trade off, and it is commonly used as an evaluation of all possible thresholds in binary classifiers. Instead, test accuracy is calculated based on a specific threshold. AUC is more reliable than the test accuracy especially when we have imbalanced data set, which may lead to misleading results by just computing the accuracy based on the majority rule. The AUC for random forest and SVM are the highest, implying that those two models have higher predictive power. The sensitivity and specificity are computed by using “1”(income>50K) as the positive class. Specificity are similar among three models, but the sensitivity for random forest is higher than the rest two. This observation is consistent with the confusion matrix such that the true positive events . The results in Table 9 are visualized in Figure 28 and 29. We can see that all models are very similar (mostly differ by less than 5% for each measure), and random forest slightly surpass the other models.

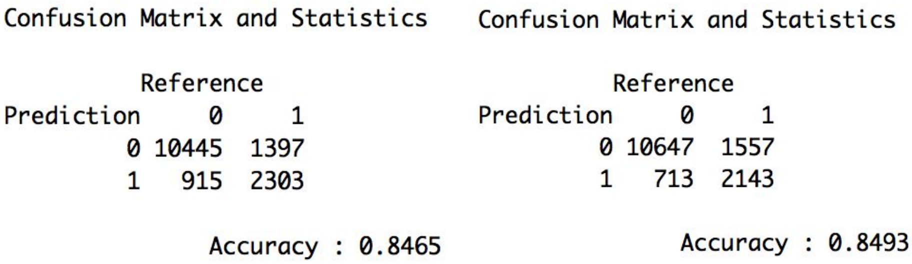


```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10770  1937
         1   590  1763

               Accuracy : 0.8322
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10702  1708
         1   658  1992

               Accuracy : 0.8429
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10548  1437
         1   812  2263

               Accuracy : 0.8507
```

Table 7: confusion matrix and test accuracy rate resulted from Classification Tree, Bagged Tree and Random Forest (from left to right)



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10445  1397
         1   915  2303

               Accuracy : 0.8465
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10647  1557
         1   713  2143

               Accuracy : 0.8493
```

Table 8: confusion matrix and test accuracy rate resulted from Neural Network and Support Vector Machine

	Classification tree	Bagged tree	Random forest	Support vector machine	Neutral network
Training accuracy	83.33%	84.33%	90.92%	NA	NA
Test accuracy	83.22%	84.29%	85.07%	84.93%	84.65%
AUC	83.77%	84.77%	88.54%	88.92%	77.09%
Sensitivity (TPR)	47.65%	53.84%	61.59%	NA	NA
Specificity (TNR)	94.81%	94.21%	92.57%	NA	NA

Table 9: statistical summary for all classifiers (NAs mean values are not computed due to limited time)

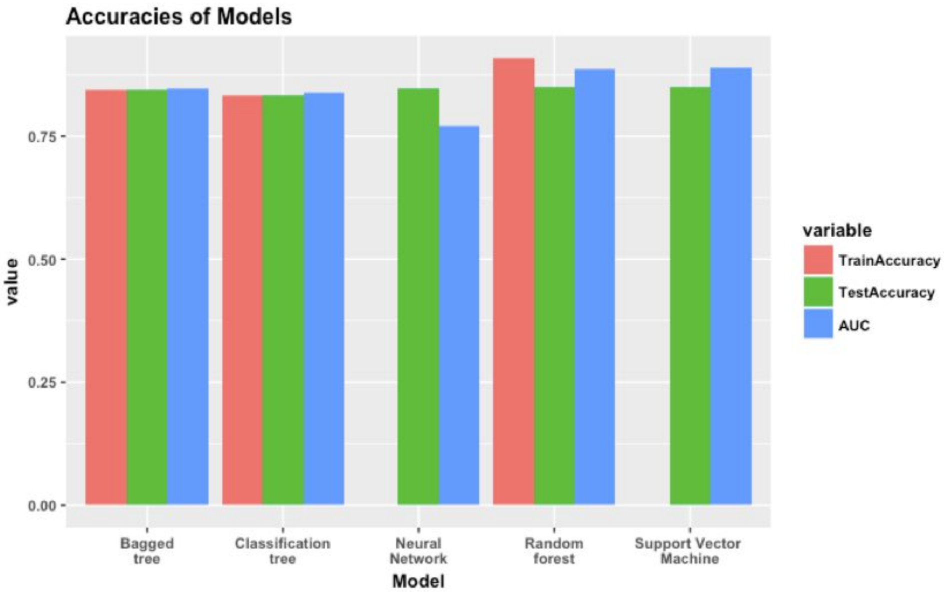


Figure 28. Accuracies and AUC for all models

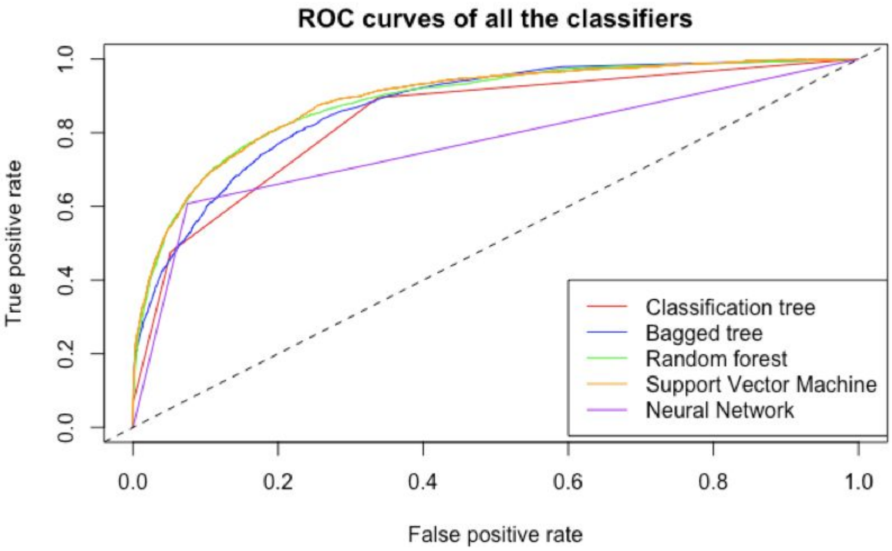


Figure 29. ROC curves for all classifiers

Conclusion

Through the comparison of all the models, mainly classification tree, bagged tree and random forest, we found that random forest perform the best, namely, having the best prediction for income >50K. It has highest training accuracy, test accuracy and AUC. But overall, all the models have very similar outcomes. When pruning the trees, CV error is more reliable than OOB error, so we generally use CV method to tune the parameters. The important variables for all classifiers are very similar, mainly includes “cap.gain”, “marital.status”, “edu”, “occupation”, “age”, “hrs.per.week” and “cap.loss”. Compared to “race” and “country”, these features actually make more sense to affect the income level. When your income is high, you can make investment or sell real estate properties, so you will have “cap.gain” and “cap.loss”. When you are married or become parents, you need to make more money to support the family. If you have a higher education, you may get better occupation which can have a higher income. Old people and young adults are generally not in the workforce, so they have less income than middle-aged people. If you are hardworking and work more hours per week, you can get more paid. To build a classification tree, only three features are used (“marital.status”, “edu” and “cap.gain”), and these features rank very top for bagged tree and random forest. Therefore, it concludes that “marital.status”, “edu” and “cap.gain” are the most important features in classification.

The EDA part can be done in many different ways. For a limited time, we did not try other grouping methods. Furthur work can be done to group the features in different ways. For example, group “country” in terms of continents or group “age” into several ranges. Besides, boosting can also be performed for classification.