Classification Model employing 1994 Census database.

By,
Naga Ramesh Kamisetti

Sohan Gudipati

Mani Kanth Kasula

Venkata Naga Siva Kalyan Mannava

Table of Contents

1. Executive Summary

This report performs classification on a census data to find out the important factors that affect a citizen's income. The data in question is taken from UC IRVINE Machine Learning Repository named "Adult" data set which in turn takes the data from 1994 census. The main concern of the study is to find out the factors that influence the income of a person. Several factors that seem unrelated exert a surprisingly significant influence on the income. Finding these correlations, as well as analysing the reasons behind is the main theme of the study. We also aim to find how affective the classification techniques are and why?

The data is first cleaned by removing missing data and noise. Logistic regression was applied and it was observed that further modification of data is required. As such, data was trimmed and clustered to group the similar behaving data. Then advanced models like CART and Random Forests are performed on the data. Several other models were applied which didn't work due to a variety of reasons. Neural Networks, a highly advanced method new to us was studied and applied. ROC technique was used to pick the best model based on the accuracy.

We found out that several factors like relationship, education and capital gain are decisive in deciding a person's income. Low capital gain also adversely affects the income. We also found out that any education below college has the same impact on income (negative). Also, several occupations like army, farming-fishing, handlers-cleaners etc. have low incomes compared to the occupations like exec-managerial. Apart from these, we also found that logistic regression works the best. Advanced techniques like Neural Networks worked poorly. Models like QDA didn't even work.

The results made us conclude that income disparity is a real and serious issue. Education and relationship are important factors as higher education like college often ensures a comfortable career and a person who has a comfortable income is more likely to get married. Also, the fact the QDA didn't work which made us conclude that the data is linearly distributed and not quadratically. Advanced methods need not necessarily produce great results. Often, the best model is determined by the data.

Based on results, we recommend that suitable policies be made to ensure more young citizens opt for higher education. Considering the fact that several occupations like farming and army fall under low income category, efforts are to be made in the form of welfare schemes to ensure a decent life for these people. After all, it is only fair to reward the efforts of these people, who ensure that we are safe and well fed.

2. **Introduction** [1] [2]

This project evaluates the prospects of data mining of demographic data based out of 1994 census database, for the purpose of creating one of more classification models. These models should be capable of accurately predicting individuals whose salary exceeds a specific value.

The dataset used for analysis has been taken from University of California Irvine Data Repository namely "Adult" dataset. It comprises of data samples from all the 51 states and several other countries and encompasses all the possible demographics such as age, education and current employment stage.

The classification model developed as a part of this project aims at classifying individuals with salaries exceeding 50K US dollars. The report includes background work done during the process of data cleaning, clustering relevant data and description of interesting or useful patterns which were found while performing several classifiers in R environment.

This report will describe the work carried out during the iterative process of data preparation, modelling and evaluation including data formatting, consistency or other quality issues, opportunities for useless instance or attribute removal and the approaches taken to solving issues with instances affected by noise, outliers or missing values.

3. Literature Review

The following papers were found to be informational and so were reviewed to obtain valuable addition to our project. Each of them had specific conclusions and several discussions which have helped us gain an understanding of the problem statement and analyse the tools and opportunities available at hand.

[1] M. Hoffman, "Predicting Earning Potential on Adult Dataset," Institute of Technology Blanchardstown, 2011.

[2] Ritika, "Research on Data Mining Classification," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 4, no. 4, pp. 329-332, 2014.

[3] Z. Zheng, "A Benchmark for Classifier Learning," The University of Sydney, Sydney, 1993.

[4] C.-Y. J. Peng, K. L. Lee and G. M.Ingersoll, "An Introduction to Logistic Regression Analysis and Reporting," *Journal of Educational Research,* vol. Vol.96(No.1), no. September/October, 2002.

[5] V. Sampath, A. Flagel and C. Figueroa, *A Logistic Regression Model to Predict Freshmen Enrolments.*

[6] M. Fernandez-Delgado, E. Cernandas, S. Barro and D. Amorim, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems," *Journal of Machine Learning Research 15,* pp. 3133-3181, 2014.

[7] R. Caruana and A. Niculescu-Mizil, "An Empirical Comparision of Supervised Learning Algorithms," in *International Conference on Machine Learning*, Pittsburgh,PA, 2006.

[8] S. Rosset, "Model Selection via the AUC," in *International Conference on Machine Learning*, Banuff, 2004.

The first study was on **Business Intelligence and Data Mining Applications Project: Predicting earning potential on Adult Dataset. Institute of Technology, Blachardstown.** This is done by Veera Gudipati.

This paper gave us a clear picture on how to start working on a given dataset. There were challenges associated with data Interpretation such as removing missing data or incomplete data.

The study in this paper initially concentrated on analysing various properties of each variables in the dataset. The steps followed in this report are well structured and as follows:

- Step 1: Define Data mining objective.
- Step 2: Project plan.
- Step 3: Data Understanding, which includes sections such as Data description, Data exploration & Verification of Data quality.
- Step 4: This is primarily data construction.
- Step 5: Final step being modelling.

Our project follows the above guidelines, but however employs different classifiers when compared with this article. This paper included illustrations of box plots for each attribute, suggesting that each of these may have predictive significance.

Rule Induction Classifier was applied to the training dataset which achieved an accuracy 82.85%, but the disadvantage is the long run time which was almost 3 days. Hence this was not recommended. And also, they have concluded that country, education, survey weight & marital status did not affect the classification model.

The second paper to be reviewed is; **an empirical comparison of Supervised Learning Algorithms**. This was done by Naga Ramesh. This paper gave us an overview of methodology

which involves comparison of analysis between ten supervised learning methods: SVM, neural nets, logistic regression, decision trees, bagged trees, boosted trees & boosted stumps.

This paper also discussed several performance metrics namely: Accuracy, F-Score, Lift, ROC Area, Average precision, Precision/Recall break-even point, squared error & cross entropy.

And also, the authors of the paper have divided the above mentioned 8 performance metrics into 3 groups: threshold metrics, ordering/rank metrics and probability metrics:

- For the threshold metrics, it is not important how close a prediction is to a given threshold, only if it is above or below the threshold. The metrics which come under this category are: Accuracy, Lift and F-Score.
- Coming to the second category are the Ordering/rank metrics. These metrics depend only on the ordering of the cases and not the predicted values. The metrics which are Ordering/rank type are area under the ROC curve, average precision (APR), and precision/recall break-even point (BEP).
- Third type are the probability metrics, which are Square Error (RMS) and cross entropy (MXE). These interpret the predicted value of each case as the conditional probability of that class being in the positive class.

The authors have commented that this field has made substantial progress in the last decade.

Third paper, **An Introduction to Logistic Regression Analysis & Reporting** gives an idea of scope of analysis using logistic regression. The central mathematical concept that underlies

logistic regression is the logit which is basically the natural logarithm of an odds ratio. This paper has been studied by Veera Gudipati.

One of the main conclusion from this paper is that Logistic Regression is well suited for describing and testing hypothesis about relationships between a categorical outcome variable and one or more categorical or continuous predictor variables.

A two-predictor logistic model has been fitted to the data. The analysis was done by the Logistic procedure in SAS version 8.

Criteria used for overall model evaluation: Any logistic model if it shows a better fit to the data when compared to the intercept-only model. This is baseline since it contains no predictors.

Next step is the validation step. The degree to which predicted probabilities agree with actual outcomes is expressed as either a measure of association or a classification model.

The authors have also provided recommendations about information to include in the assessments. Few of them are as follows:

- An overall evaluation of the logistic model.
- Statistical tests on individual predictors.
- Goodness-of-fit statistics.
- An assessment of the predicted probabilities.

Fourth paper, **Research on Data Mining Classification** has been studied by Kalyan Mannava. This research deals with problems regarding data mining, which involves cleaning the enormous amount of information received and properly organizing it.

Experts in Big Data field agree that data mining is the most challenging part of any analysis work. In all the major fields, gigantic amounts of raw information are received from all the sources. Most of this information is 'Noise', data with no real meaning or use. As it is virtually impossible to study all the data and pick the relevant portions, several techniques are developed to make the job easier.

One of the most attractive techniques is classification-which essentially involves finding rules that categorize the data into disjoint groups. There are several classification models like LDA, QDA, KNN, Decision Trees etc., while these techniques are often employed to perform analysis, they can also be employed in cleaning the data. Now we have even more advanced models like Neural Networks, Genetic programming.

Neural Networks essentially mimic human brain and learn from experience, just like humans do. In Genetic programming, each solution is coded into a chromosome which has the capability to mutate/combine with other chromosomes based on their fitness value. This method is inspired by evolution and needless to say, highly effective. Another interesting model is Ant colony, in which the techniques employed by Ants to trace the shortest route back to the colony are employed for data mining.

Thus, we have a vast arsenal of models at our disposal to effectively deal with data mining. However, caution needs to be asserted in utilizing these models, because, these models are only as good as the data available. Good data is always the first requirement for good data exploration. If good data is available, then the best technique can be chosen. Most of the times this is a trial and error process-finding the best technique. This is really the place where a data-scientist needs to execute discretion and find the best technique or rather a combination of available techniques to obtain the optimum data mining.

The fifth research paper; **Model Selection via the AUC** has been reviewed by Naga Ramesh. This research presents a statistical analysis of AUC as an evaluation criterion for classification scoring models.

It introduces concepts about ROC analysis & the AUC, which is a one-number measure of a model's discrimination performance. The paper discussed how ROC has been used to discuss relationship between sensitivity and specificity for a 2-class classification model when applied to a test data.

The authors have quoted several other research papers which prove the fact that in recent years, there is an upcoming interest in using AUC as an evaluation measure in Data Mining & Machine learning communities.

The basic aim of the paper is:

- To increase understanding of this evaluation measure and its advantages.
- To present a new tool for its analysis.

The methodology used is to compare the AUC scores of two models when using the same evaluation or test set. And in the final step using AUC for a completely new scenario and proving that it still may be a better model than empirical error rate.

Following is the description of the method implemented:

- Significance testing of the difference between AUC scores.
    - Exact moments
    - Assumptions and calculations
    - Empirical Evaluation of methods from AUC analysis
- Using AUC to evaluate Classification Performance.
    - Simulations and real data experiments

The authors have concluded with a few unanswered questions such as Can we define more rigorously and clearly situations where AUC is preferable? And where the misclassification error rate is preferable?

Sixth paper, **Do we need 100s of classifiers to solve real world problems?** Has been reviewed by Mani Kanth Kasula. The summary is as follows:

With hundreds of classifiers available today, it becomes quite a task to decide the right one to use on a data set to obtain the best possible results. This paper aims to address the issue of finding the right classifiers by ranking them on the basis of average accuracy and also calculating the probability of achieving the best accuracy. Based on the ranking, parallel random forest was found to be the best classifier and random forest to be the best family of classifiers.

When performing a classification, we come across different types of bias entering the process. Some of them arise due to the criterion used to collect and prepare the data set and also the type of data available. Also, it is practically impossible to determine the best possible accuracy that can be achieved by using a certain classifier. To overcome these difficulties, an exhaustive evaluation of 179 classifiers, arising from 17 different classifier families, were performed on 121 data sets taken from the UCI repository leading to a total of 21,659 combinations. Sometimes, errors during classification can arise due the complexity of the data set. However, in this work, it was assumed that any errors that might arise are due the limitations of the classifiers. A four-fold cross validation was performed on the all the data sets to address the issues of bias entering due the selection of training and testing sets and the average of the observations was recorded.

The accuracy of the performance of all the classifiers on each data set was tabulated and the average was calculated. Based on these averages, a ranking was given in the decreasing order

of the accuracy. Also, the best classifier for each data set was recorded to calculate the probability of achieving the maximum accuracy.

One interesting finding was that the random forest method, despite being a relatively old method, works better than many other recent approaches that were developed. The best results were achieved by parallel random forest with an average of 82.0% and a high of 94.1% of the maximum accuracy. In the top 20 (Based on the ranking), there are 5 classifiers from the random forest family, 5 from Support vector machines, 4 from neural networks and 4 ensembles, which again proves the reliability of random forest methods. Extreme learning machine with multiple kernels had the highest probability of achieving the maximum accuracy at 13.2% with svm_c coming second at 10.7%. We realize that despite having so many classifiers, the probability of achieving the maximum accuracy is very far from 100%.

Seventh paper, **A Benchmark for Classifier Learning**, has been studied and reviewed by Mani Kanth Kasula.

Even though we have many algorithms for classifier learning, it would be of great help to have a standard benchmark for comparison of the same. Some classifiers work better on datasets with a particular dataset pattern and type. This paper proposes sixteen dimensions to describe classification tasks to form a benchmark using real world data and synthetic datasets. Any data set for a classification problem has three aspects viz. form of the attributes, form of the instances, and the form of the classes. In accordance with aspects, the proposed dimensions can be grouped as the following.

1	Regarding Attribute	1. Type of attributes
		2. Number of attributes
		3. Number of different nominal attribute values
		4. Number of irrelevant attributes.
2	Regarding Instances	1. Dataset size
		2. Dataset density
		3. Level of noise in attribute values
		4. Level of noise in class memberships
		5. Frequency of missing attribute values.
3	Regarding Classes	1. Number of classes
		2. Default accuracy
		3. Entropy
		4. Predictive accuracy
		5. Relative accuracy
		6. Average information score
		7. Relative information score

To perform a benchmark study, thirty one real world and synthetic datasets from the UCI repository and the ones mentioned in three of the Proceedings of the International Workshop/Conference on machine learning were taken and all the proposed sixteen dimensions were defined for each data set and tabulated. It was observed that the behavior of classifiers were highly dependent on the dimensions of the dataset. For example, C4.5, owing to its capability of tolerating irrelevant attributes, performed better than IB1 on domains involving irrelevant attributes while the performance of IB1 was found to be much better than C4.5 on domains having continuous attributes because of its treatment of the same.

We learn that using such a benchmark helps us define the dimensions of a given dataset and decide the algorithm that might work better.

Eighth paper, **A Logistic Regression Model to predict Freshman Enrolments** has been studied and reviewed by Kalyan Mannava.

Every year, university admissions committees face challenges in estimating the enrolment due to uncertainty of human factors and choices. The office of admissions at George Mason University faces a daunting task of meeting the enrolment target while maintaining the quality of intake in terms of their credentials. This paper seeks to present the step by step procedure of creating a logistic regression model in SAS® to predict whether a student will enroll or not after being admitted at George Mason University. It gives a demonstration of how a classification technique can be used to solve real world problems.

Using high school GPA (0-4.00), SAT scores (0-1600), sex (male/female), race (white/ Hispanic / black /Asian /other), residency (In-State/Out-State), distance from college (in miles) as predictors, the enrolment indicator (Yes/No) was estimated using historical data of the same. Race and sex had data recorded in the character format and they were therefore recoded to the numeric format. GPA and SAT scores were found to be fairly normal but the plot for distance was significantly deviating from normality. This problem was treated by using a log transform. Outliers were determined using the PROC STANDARD procedure in SAS® and missing data in the dataset was accounted for and tagged using recoding to avoid deleting those rows.

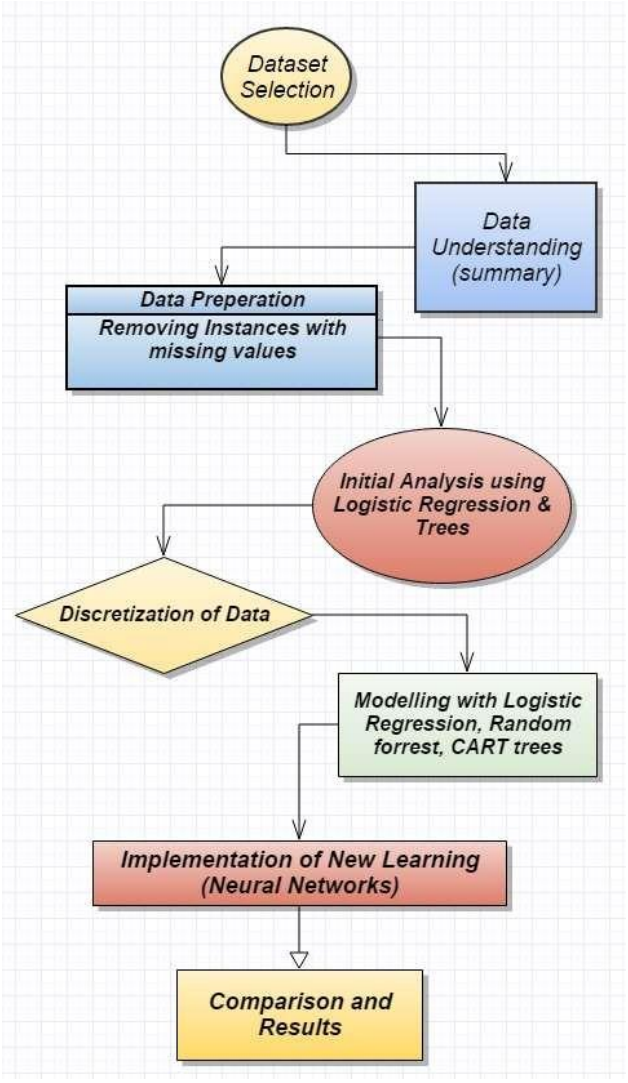
A backward selection method was used with PROC LOGISTIC function to model the data. Only terms up to the second order of interaction were considered because interpretation of higher order terms is very complex. The predictive power (C), specificity and sensitivity were calculated from the classification results obtained.

From the historical data, it was observed that only 35%- 40% of the admitted students enrolled at the university and therefore the cut-off value for classification was set at 0.35 instead of the standard 0.5.

Here, we learnt the importance of cut-off value, deciding a certain value and its interpretation according to the problem under consideration. This paper gave a detailed approach of how to address a practical problem and the treatment of available data. Some of the learnings on how to treat data before applying classification techniques were employed in the current project.

4. Project Approach

The following flowchart gives an idea about the project timeline which starts after the literature review:



The new learning is implementation of neural network classification.

Neural networks is a classifier technique inspired from the human brain structure. It consists of three layer viz. input layer, activation function (hidden layer) and the output layer. The input layer, as the name says inputs historical data into the hidden layer. The hidden layer consists of complex mathematical functions that modify input data and makes predictions. The output layer gives the final predictions.

Neural networks package can be installed in R by using the following code.

install.packages("neuralnet")

Neural networks can detect complex nonlinear relationships and easily detects all possible interactions among the attributes. However, it has its own cons. Using neural networks can lead to a very high computational load and also there is a very high probability of overfitting the data.

Neural networks are often used for studying behaviour and control in animals and machines, pattern recognition, and data compression.

http://www.dummies.com/how-to/content/how-predictive-analysis-neural-networks-work.html
http://www.jclinepi.com/article/S0895-4356(96)00002-9/abstract?cc=y=
http://turing.iimas.unam.mx/~cgg/cogs/doc/FCS-ANN-tutorial.htm

5. Implementation Details

5.1 Data Understanding

The first step of the project was dataset selection. Adult dataset is based on 1994 census data, and is sourced from UCI Machine Learning Repository. It contains two parts namely training data “adult.data” and testing data “adult.test”. It has a total of forty nine thousand instances. The last column is the income field which has been converted into a binary value denoting whether the salary is <50K USD or >50K USD.

76% of the records come under the class label of <50K USD as income. There are fourteen attributes in all which give demographic data, few examples are education, gender, country etc.

A total of 4262 instances have missing data. 1836 entries in Employment class, 1843 in Occupation and 583 entries in Country.

Summary of the data set is as follows (this reflects the refined dataset):

```
> summary(pd)
      age                type_employer        education
 Min.   :17.00   Private         :22286   HS-grad   :9840
 1st Qu.:28.00   Self-emp-not-inc: 2499   highschool:8103
 Median :37.00   Local-gov       : 2067   Bachelors :5044
 Mean   :38.44   State-gov       : 1279   <=10th    :2316
 3rd Qu.:47.00   Self-emp-inc    : 1074   Associates:2315
 Max.   :90.00   Federal-gov     :  943   Masters   :1627
                 (Other)         :   14   (Other)   : 917

                marital              occupation
Divorced              : 4214   Prof-specialty :4038
Married-AF-spouse     :   21   Craft-repair   :4030
Married-civ-spouse    :14065   Exec-managerial:3992
Married-spouse-absent:  370   Adm-clerical   :3721
Never-married         : 9726   Sales          :3584
Separated             :  939   Other-service  :3212
Widowed               :  827   (Other)        :7585

       relationship                    race           sex
Husband       :12463   Amer-Indian-Eskimo:  286   Female: 9782
Not-in-family : 7726   Asian-Pac-Islander:  895   Male  :20380
Other-relative:  889   Black             : 2817
Own-child     : 4466   Other             :  231
Unmarried     : 3212   White             :25933
Wife          : 1406

capital_gain capital_loss  hr_per_week      country
None:27624   None:28735   Min.   : 1.00   US          :27504
Low : 1448   Low :  734   1st Qu.:40.00   Mexico      :  610
High: 1090   High:  693   Median :40.00   Other       :  542
                          Mean   :40.93   Philippines:  188
                          3rd Qu.:45.00   Germany     :  128
                          Max.   :99.00   Puerto-Rico:  109
                                          (Other)     : 1081
```

5.2 Data Cleaning

After performing initial analysis (code included in Appendix) using logistic regression and trees, it is found to be necessary to filter the dataset. All the missing data is accounted to be present in 3 attributes namely Employer type, Occupation and Country. Before this, we had to add labels for each columns for the dataset to make it easy for viewing and modelling.

The following code removes the missing values. It simply means the dataset (pd) is modified and now will not include entries which equal to a question mark "?".

```
pd<-pd[pd$type_employer!="?",]
pd<-pd[pd$occupation!="?",]
pd<-pd[pd$country!="?",]
```

In fact, there are two ways of dealing with missing data. We can either chose to guess the value of the missing entry based on the previous and the next entry or we can delete. Since the quantity of missing data is low compared to the total number of instances, we have chosen to delete the missing values in entirety.

Another challenge encountered during initial analysis using trees was the high number of classes in country attribute. Initially there were 42 different classes, most of the data represents United States while some countries have less than 50 instances. All such entries have been changed to "Other". This has been enforced with the help of following code.

```
> tree.pdffinal=tree(income~.,pdffinal)
Error in tree(income ~ ., pdffinal) :
  factor predictors must have at most 32 levels
> pdfreduced=pdffinal
```

```
pd$country[pd$country=="Columbia"]="Columbia"
pd$country[pd$country=="China"]="China"
pd$country[pd$country=="Canada"]="Canada"
pd$country[pd$country=="Cambodia"]="Other"
pd$country[pd$country=="Ecuador"]="Other"
pd$country[pd$country=="France"]="Other"
pd$country[pd$country=="Greece"]="Other"
pd$country[pd$country=="Haiti"]="Other"
pd$country[pd$country=="Holand-Netherlands"]="Other"
pd$country[pd$country=="Honduras"]="Other"
pd$country[pd$country=="Hong"]="Other"
pd$country[pd$country=="Hungary"]="Other"
pd$country[pd$country=="Iran"]="Other"
pd$country[pd$country=="Ireland"]="Other"
pd$country[pd$country=="Laos"]="Other"
pd$country[pd$country=="Nicaragua"]="Other"
pd$country[pd$country=="Outlying-US(Guam-USVI-etc)"]="Other"
pd$country[pd$country=="Peru"]="Other"
pd$country[pd$country=="Poland"]="Other"
pd$country[pd$country=="Portugal"]="Other"
pd$country[pd$country=="Scotland"]="Other"
pd$country[pd$country=="Taiwan"]="Other"
pd$country[pd$country=="Thailand"]="Other"
pd$country[pd$country=="Trinadad&Tobago"]="Other"
pd$country[pd$country=="Yugoslavia"]="Other"
pd$country=as.factor(pd$country)
```

Initial analysis using trees have shown us that data specific to education has been shown under single category. To avoid this, we have edited the data and classified into <10th, high school etc. This has been enforced using the following code.

```
pd$education=as.character(pd$education)
pd$education = gsub("^10th","<=10th",pd$education)
pd$education = gsub("^11th","highschool",pd$education)
pd$education = gsub("^12th","highschool",pd$education)
pd$education = gsub("^1st-4th","<=10th",pd$education)
pd$education = gsub("^5th-6th","<=10th",pd$education)
pd$education = gsub("^7th-8th","<=10th",pd$education)
pd$education = gsub("^9th","<=10th",pd$education)
pd$education = gsub("^Assoc-acdm","Associates",pd$education)
pd$education = gsub("^Assoc-voc","Associates",pd$education)
pd$education = gsub("^Bachelors","Bachelors",pd$education)
pd$education = gsub("^Doctorate","Doctorate",pd$education)
pd$education = gsub("^HS-Grad","HS-Grad",pd$education)
pd$education = gsub("^Masters","Masters",pd$education)
pd$education = gsub("^Preschool","<=10th",pd$education)
pd$education = gsub("^Prof-school","Prof-School",pd$education)
pd$education = gsub("^Some-college","highschool",pd$education)
pd$education=as.factor(pd$education)
```

This has been done to ensure the data is well prepared for modelling stage.

5.3 Model Implementation

The following packages were installed and library function has been called.

```
#installing required packages
install.packages("randomForest")
install.packages("ROCR")
install.packages("nnet")
install.packages("rpart")
install.packages("MASS")

library(randomForest)
library(ROCR)
library(nnet)
library(rpart)
library(MASS)
library(tree)
```

The following classification techniques have been employed.

```
# Logistic Regression and its ROC analysis
glm.fit = glm(income ~.-relationship, family = "binomial",data = pd)
glm.preds = predict(glm.fit,newdata=pdtest,type="response")
glm.pred = prediction(glm.preds,pdtest$income)
glm.perf = performance(glm.pred,"tpr","fpr")
```

```
# Random Forest
bestmtry <- tuneRF(pd[-13],pd$income, ntreeTry=100, stepFactor=1.5,improve=0.01, trace=TRUE, plot=TRUE, dobest=FALSE)
rf.fit <-randomForest(income~.,data=pd, mtry=2, ntree=1000, keep.forest=TRUE, importance=TRUE,test=pdtest)
rf.preds = predict(rf.fit,type="prob",newdata=pdtest)[,2]
rf.pred = prediction(rf.preds, pdtest$income)
rf.perf = performance(rf.pred,"tpr","fpr")
```

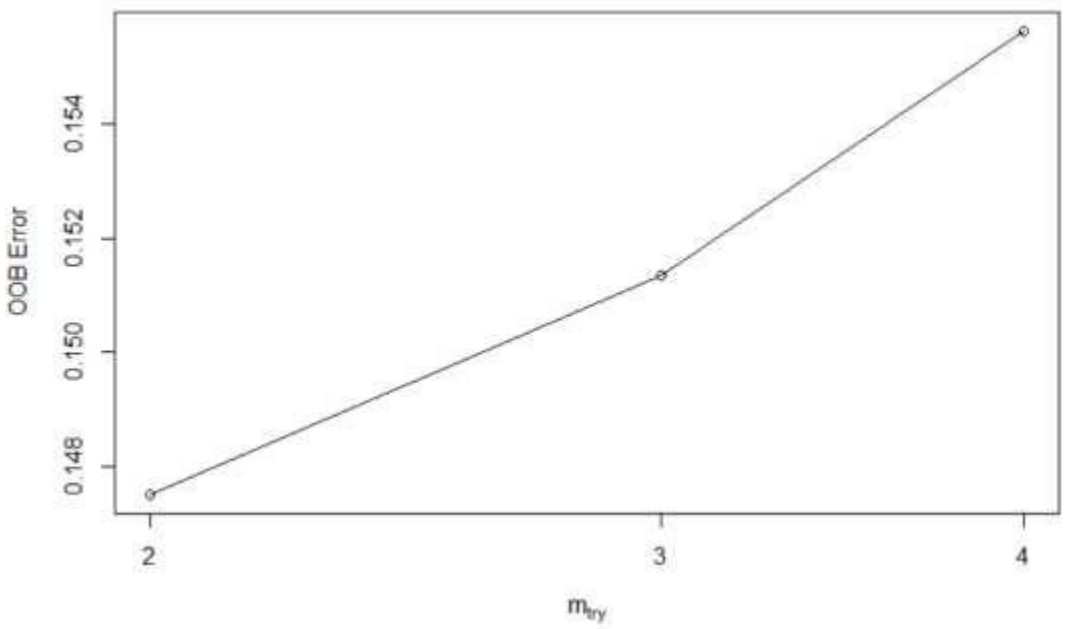
```
# CART Trees
mycontrol = rpart.control(cp = 0, xval = 10)
tree.fit = rpart(income~., method = "class",data = pd, control = mycontrol)
tree.fit$cptable
tree.cptarg = sqrt(tree.fit$cptable[8,1]*tree.fit$cptable[9,1])
tree.prune = prune(tree.fit,cp=tree.cptarg)
tree.preds = predict(tree.prune,newdata=pdtest,type="prob")[,2]
tree.pred = prediction(tree.preds,pdtest$income)
tree.perf = performance(tree.pred,"tpr","fpr")
```

```
# Neural Network
nnet.fit = nnet(income~.,data=pd,size=2,maxit=10000,decay=.001)
nnet.preds = predict(nnet.fit,newdata=pdtest,type="raw")
nnet.pred = prediction(nnet.preds,pdtest$income)
nnet.perf = performance(nnet.pred,"tpr","fpr")
```

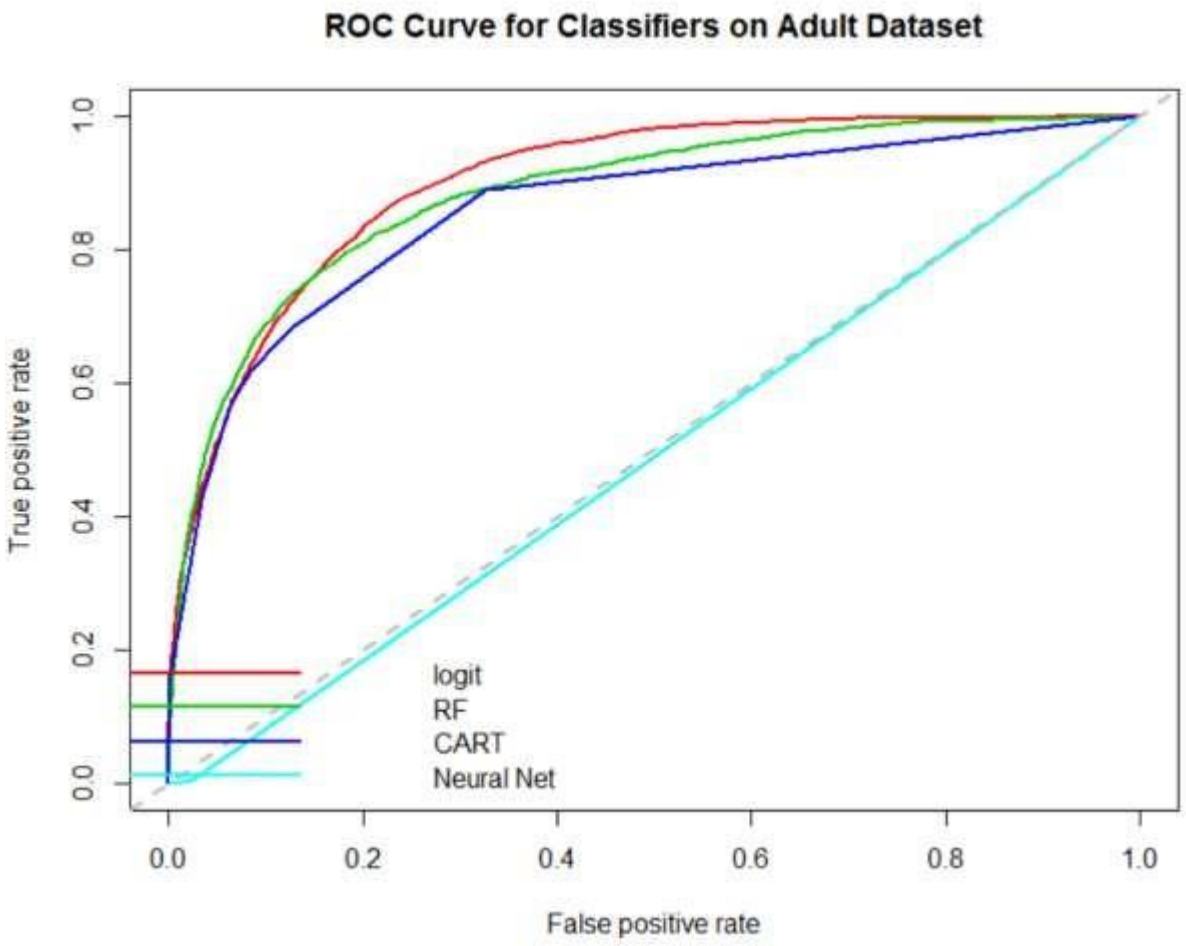
```
# Plotting ROC Curves p
plot(glm.perf,col=2,lwd=2,main="ROC Curve for Classifiers on Adult Dataset")
plot(rf.perf,col=3,lwd=2,add=T)
plot(tree.perf,lwd=2,col=4,add=T)
plot(nnet.perf,lwd=2,col=5,add=T)
abline(a=0,b=1,lwd=2,lty=2,col="gray")
legend("bottomright",col=c(2:5),lwd=2,legend=c("logit","RF","CART","Neural Net"),bty='n')
```

```
#tree
library(tree)
attach(pd)
tree.pd=tree(income~.,pd)
summary(tree.pd)
tree.pred=predict(tree.pd,pdtest,type="class")
table(tree.pred,pdtest$income)
plot(tree.pd)
text(tree.pd,pretty=0)
```

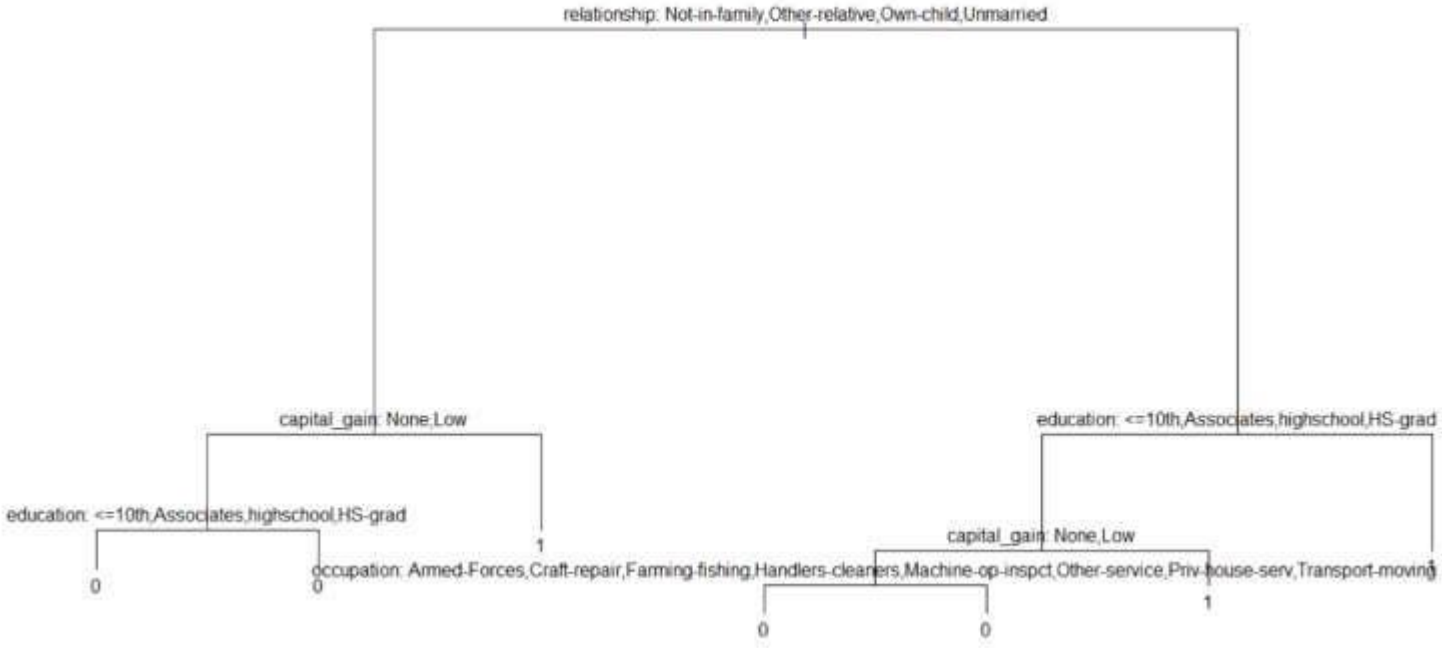
6. Results



The least OOB Error is obtained for m(try) value of 2. Hence in the code for random forest classifier this value is taken.



The ROC curve gives the accuracy of prediction- accuracy is directly proportional to the area under the curve. Logistic Regression and random Forests have the highest accuracy, closely followed by CART. At lower False positive rates all three methods perform equally and at mid range RF dominates. At high False positive rates though, Logistic Regression works the best. And regarding Neural Networks, it seems to be slightly below the y=x line, which rates its performance as poor.



It can be seen from the above tree that the leftmost leaf has 0 on both sides. That means the two terminal nodes have the same value. While this corresponding split does not decrease the classification error rate, it increases the node purity. One way of interpreting it is, if the test data falls on to the region given by right hand leaf, then it is certain that it belongs to low income category where as if it belongs to region given by left node then it is probably a low income category. This split increases the Gini index and cross-entropy.

The CART shows us several interesting results. Most of the low income categories fall under Not-in-family, Own child and Unmarried sections. Capital gains seem to be affecting the income positively. Below college, all education categories basically impact the income in the same way (low income category). Several occupations like armed forces, crafts, farming-fishing and other services all fall under the low income category. Overall married-Highly educated-high capital gain seem to positively affect the income while a negation of these categories almost always fall under low income.

7. Conclusion

The results obtained yield some interesting conclusions.

- Income disparity is real and serious. Education seems to be the most decisive factor. Also, any education below college seems to have the same impact on the income. Hence efforts must be made to ensure more students opt for higher studies.
- Several occupations like army, farming and handlers fall under the low income categories. While this problem cannot be solved easily, efforts must be made to ensure they lead a decent life. This can be done in form of subsidies and welfare schemes.
- Relationship status seems to affect the income. This may actually be a reverse implication; people with high incomes tend to get married. This actually shows in other way too. People who are unmarried, Not in family or own a child tend to be in low income group. Child welfare schemes might be a relief.
- Capital gains seem to be affecting the income- hardly a surprise.
- Several categories of a variable under certain circumstances (when taken in combination with particular categories of other variables) seem to have the same affect on the income. This led us to cluster the data. This also showed that economic disparity is not as varied as the variables themselves. It seems to follow a broader, simpler pattern
- The most advanced models need not necessarily work the best. Neural Networks performed poorly and QDA fell flat when tried. Simple method like Logistic Regression worked the best. Also the data seems to be linearly distributed- which explains the failure of QDA. We can conclude that the model to be used is determined by the data being worked upon.

8. References

[1] M. Hoffman, "Predicting Earning Potential on Adult Dataset," Institute of Technology Blanchardstown, 2011.

[2] Ritika, "Research on Data Mining Classification," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 4, no. 4, pp. 329-332, 2014.

[3] Z. Zheng, "A Benchmark for Classifier Learning," The University of Sydney, Sydney, 1993.

[4] C.-Y. J. Peng, K. L. Lee and G. M.Ingersoll, "An Introduction to Logistic Regression Analysis and Reporting," *Journal of Educational Research,* vol. Vol.96(No.1), no. September/October, 2002.

[5] V. Sampath, A. Flagel and C. Figueroa, *A Logistic Regression Model to Predict Freshmen Enrolments.*

[6] M. Fernandez-Delgado, E. Cernandas, S. Barro and D. Amorim, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems," *Journal of Machine Learning Research 15,* pp. 3133-3181, 2014.

[7] R. Caruana and A. Niculescu-Mizil, "An Empirical Comparision of Supervised Learning Algorithms," in *International Conference on Machine Learning*, Pittsburgh,PA, 2006.

[8] S. Rosset, "Model Selection via the AUC," in *International Conference on Machine Learning*, Banuff, 2004.

9. Appendix

Code for Training data Discretization:

```
#the dataframe is downloaded from the website and read in to R using read.table command
pd=read.table("https://archive.ics.uci.edu/ml/machine-learning-
databases/adult/adult.data",sep=",",header=F,col.names=c("age", "type_employer",
"fnlwgt", "education","education_num","marital", "occupation", "relationship",
"race","sex","capital_gain", "capital_loss", "hr_per_week","country",
"income"),fill=FALSE,strip.white=T)
#to get the names of the variables
names(pd)
#to get the dimensions of the data frame
dim(pd)
#finding out the class of a variable using lapply()
lapply(pd,class)
#findout the missing values in a column using table(pd$variablename)
#oserved that there are rows with "?" in type_employer, occupation and country.
#removemissingvalues
pd<-pd[pd$type_employer!="?",]
pd<-pd[pd$occupation!="?",]
pd<-pd[pd$country!="?",]
dim(pd)
```

```
#changing the income values <=50K to "0" and >50K to "1"
pd$income=ifelse(pd$income=="<=50K",0,1)
#class probability for "0" is 0.7510775
#class probability for "1" is 0.2489225
#removing the variable education_num and fnlwgt
pd[["education_num"]]=NULL
#removed this variable because it reperesents same data as education variable
#and intial analysis showed that it cluster the analysis.
pd[["fnlwgt"]]=NULL
summary(pd)
dim(pd)
pd$education=as.character(pd$education)
pd$education = gsub("^10th","<=10th",pd$education)
pd$education = gsub("^11th","highschool",pd$education)
pd$education = gsub("^12th","highschool",pd$education)
pd$education = gsub("^1st-4th","<=10th",pd$education)
pd$education = gsub("^5th-6th","<=10th",pd$education)
pd$education = gsub("^7th-8th","<=10th",pd$education)
pd$education = gsub("^9th","<=10th",pd$education)
pd$education = gsub("^Assoc-acdm","Associates",pd$education)
pd$education = gsub("^Assoc-voc","Associates",pd$education)
pd$education = gsub("^Bachelors","Bachelors",pd$education)
pd$education = gsub("^Doctorate","Doctorate",pd$education)
pd$education = gsub("^HS-Grad","HS-Grad",pd$education)
pd$education = gsub("^Masters","Masters",pd$education)
pd$education = gsub("^Preschool","<=10th",pd$education)
pd$education = gsub("^Prof-school","Prof-School",pd$education)
pd$education = gsub("^Some-college","highschool",pd$education)
pd$education=as.factor(pd$education)
summary(pd$education)
#country is by default taken as factor.
#to modify its content first it is converted to character type
pd$country=as.character(pd$country)
pd$country[pd$country=="United-States"]="US"
pd$country[pd$country=="Vietnam"]="Vietnam"
pd$country[pd$country=="South"]="South"
pd$country[pd$country=="Puerto-Rico"]="Puerto-Rico"
pd$country[pd$country=="Philippines"]="Philippines"
pd$country[pd$country=="Mexico"]="Mexico"
pd$country[pd$country=="Japan"]="Japan"
pd$country[pd$country=="Jamaica"]="Jamaica"
pd$country[pd$country=="Italy"]="Italy"
pd$country[pd$country=="India"]="India"
pd$country[pd$country=="Guatemala"]="Guatemala"
pd$country[pd$country=="Germany"]="Germany"
pd$country[pd$country=="England"]="England"
pd$country[pd$country=="El-Salvador"]="El-Salvador"
pd$country[pd$country=="Cuba"]="Cuba"
pd$country[pd$country=="Dominican-Republic"]="Dominican-Republic"
pd$country[pd$country=="Columbia"]="Columbia"
```

```
pd$country[pd$country=="China"]="China"
pd$country[pd$country=="Canada"]="Canada"
pd$country[pd$country=="Cambodia"]="Other"
pd$country[pd$country=="Ecuador"]="Other"
pd$country[pd$country=="France"]="Other"
pd$country[pd$country=="Greece"]="Other"
pd$country[pd$country=="Haiti"]="Other"
pd$country[pd$country=="Holand-Netherlands"]="Other"
pd$country[pd$country=="Honduras"]="Other"
pd$country[pd$country=="Hong"]="Other"
pd$country[pd$country=="Hungary"]="Other"
pd$country[pd$country=="Iran"]="Other"
pd$country[pd$country=="Ireland"]="Other"
pd$country[pd$country=="Laos"]="Other"
pd$country[pd$country=="Nicaragua"]="Other"
pd$country[pd$country=="Outlying-US(Guam-USVI-etc)"]="Other"
pd$country[pd$country=="Peru"]="Other"
pd$country[pd$country=="Poland"]="Other"
pd$country[pd$country=="Portugal"]="Other"
pd$country[pd$country=="Scotland"]="Other"
pd$country[pd$country=="Taiwan"]="Other"
pd$country[pd$country=="Thailand"]="Other"
pd$country[pd$country=="Trinadad&Tobago"]="Other"
pd$country[pd$country=="Yugoslavia"]="Other"
pd$country=as.factor(pd$country)
#percent of males=67.56846
#percent of females=32.43154
#changing capital gain in to three factors none,low and high.
pd[["capital_gain"]] <- ordered(cut(pd$capital_gain,c(-Inf,
0,median(pd[["capital_gain"]][pd[["capital_gain"]] >0]),Inf)),labels = c("None", "Low", "High"))
pd[["capital_loss"]] <- ordered(cut(pd$capital_loss,c(-Inf,
0,median(pd[["capital_loss"]][pd[["capital_loss"]] >0]),Inf)),labels = c("None", "Low", "High"))
pd$income=as.factor(pd$income)
```

Code for Test Data Discretization:

Here we use file.choose() command and load the test data file “adult.test” into R environment.

```
pdtest=read.csv(file.choose(),sep=",",header=F,col.names=c("age", "type_employer",
"fnlwgt", "education","education_num","marital", "occupation", "relationship",
"race","sex","capital_gain", "capital_loss", "hr_per_week","country",
"income"),fill=FALSE,strip.white=T)
names(pdtest)
dim(pdtest)
lapply(pdtest,class)
pdtest<-pdtest[pdtest$type_employer!="?",]
pdtest<-pdtest[pdtest$occupation!="?",]
pdtest<-pdtest[pdtest$country!="?",]
dim(pdtest)
```

```
pdtest$income=ifelse(pdtest$income=="<=50K.",0,1)
pdtest[["education_num"]]=NULL
pdtest[["fnlwgt"]]=NULL
summary(pdtest)
dim(pdtest)
pdtest$education=as.character(pdtest$education)
pdtest$education = gsub("^10th","<=10th",pdtest$education)
pdtest$education = gsub("^11th","highschool",pdtest$education)
pdtest$education = gsub("^12th","highschool",pdtest$education)
pdtest$education = gsub("^1st-4th","<=10th",pdtest$education)
pdtest$education = gsub("^5th-6th","<=10th",pdtest$education)
pdtest$education = gsub("^7th-8th","<=10th",pdtest$education)
pdtest$education = gsub("^9th","<=10th",pdtest$education)
pdtest$education = gsub("^Assoc-acdm","Associates",pdtest$education)
pdtest$education = gsub("^Assoc-voc","Associates",pdtest$education)
pdtest$education = gsub("^Bachelors","Bachelors",pdtest$education)
pdtest$education = gsub("^Doctorate","Doctorate",pdtest$education)
pdtest$education = gsub("^HS-Grad","HS-Grad",pdtest$education)
pdtest$education = gsub("^Masters","Masters",pdtest$education)
pdtest$education = gsub("^Preschool","<=10th",pdtest$education)
pdtest$education = gsub("^Prof-school","Prof-School",pdtest$education)
pdtest$education = gsub("^Some-college","highschool",pdtest$education)
pdtest$education=as.factor(pdtest$education)
summary(pdtest$education)
#country is by default taken as factor.
#to modify its content first it is converted to character type
pdtest$country=as.character(pdtest$country)
pdtest$country[pdtest$country=="United-States"]="US"
pdtest$country[pdtest$country=="Vietnam"]="Vietnam"
pdtest$country[pdtest$country=="South"]="South"
pdtest$country[pdtest$country=="Puerto-Rico"]="Puerto-Rico"
pdtest$country[pdtest$country=="Philippines"]="Philippines"
pdtest$country[pdtest$country=="Mexico"]="Mexico"
pdtest$country[pdtest$country=="Japan"]="Japan"
pdtest$country[pdtest$country=="Jamaica"]="Jamaica"
pdtest$country[pdtest$country=="Italy"]="Italy"
pdtest$country[pdtest$country=="India"]="India"
pdtest$country[pdtest$country=="Guatemala"]="Guatemala"
pdtest$country[pdtest$country=="Germany"]="Germany"
pdtest$country[pdtest$country=="England"]="England"
pdtest$country[pdtest$country=="El-Salvador"]="El-Salvador"
pdtest$country[pdtest$country=="Cuba"]="Cuba"
pdtest$country[pdtest$country=="Dominican-Republic"]="Dominican-Republic"
pdtest$country[pdtest$country=="Columbia"]="Columbia"
pdtest$country[pdtest$country=="China"]="China"
pdtest$country[pdtest$country=="Canada"]="Canada"
pdtest$country[pdtest$country=="Cambodia"]="Other"
pdtest$country[pdtest$country=="Ecuador"]="Other"
pdtest$country[pdtest$country=="France"]="Other"
pdtest$country[pdtest$country=="Greece"]="Other"
```

```
pdtest$country[pdtest$country=="Haiti"]="Other"
pdtest$country[pdtest$country=="Holand-Netherlands"]="Other"
pdtest$country[pdtest$country=="Honduras"]="Other"
pdtest$country[pdtest$country=="Hong"]="Other"
pdtest$country[pdtest$country=="Hungary"]="Other"
pdtest$country[pdtest$country=="Iran"]="Other"
pdtest$country[pdtest$country=="Ireland"]="Other"
pdtest$country[pdtest$country=="Laos"]="Other"
pdtest$country[pdtest$country=="Nicaragua"]="Other"
pdtest$country[pdtest$country=="Outlying-US(Guam-USVI-etc)"]="Other"
pdtest$country[pdtest$country=="Peru"]="Other"
pdtest$country[pdtest$country=="Poland"]="Other"
pdtest$country[pdtest$country=="Portugal"]="Other"
pdtest$country[pdtest$country=="Scotland"]="Other"
pdtest$country[pdtest$country=="Taiwan"]="Other"
pdtest$country[pdtest$country=="Thailand"]="Other"
pdtest$country[pdtest$country=="Trinadad&Tobago"]="Other"
pdtest$country[pdtest$country=="Yugoslavia"]="Other"
pdtest$country=as.factor(pdtest$country)
pdtest[["capital_gain"]] <- ordered(cut(pdtest$capital_gain,c(-Inf,
0,median(pdtest[["capital_gain"]][pdtest[["capital_gain"]] >0]),Inf)),labels = c("None", "Low",
"High"))
pdtest[["capital_loss"]] <- ordered(cut(pdtest$capital_loss,c(-Inf,
0,median(pdtest[["capital_loss"]][pdtest[["capital_loss"]] >0]),Inf)),labels = c("None", "Low",
"High"))
pdtest$income=as.factor(pdtest$income)
```

Model file:

```
#logistic regression code
glm.fit=glm(income~.,data=pd,family="binomial")
summary(glm.fit)
glm.probs=predict(glm.fit,pdtest,type="response")
glm.pred=rep("0",length(pdtest$income))
glm.pred[glm.probs>0.5]="1"
table(glm.pred,pdtest$income)
dim(pdtest)
mean(glm.pred!=pdtest$income)

#installing required packages
install.packages("randomForest")
install.packages("ROCR")
install.packages("nnet")
install.packages("rpart")
install.packages("MASS")

library(randomForest)
library(ROCR)
```

```
library(nnet)
library(rpart)
library(MASS)
library(tree)

# Logistic Regression and its ROC analysis
glm.fit = glm(income ~.-relationship, family = "binomial",data = pd)
glm.preds = predict(glm.fit,newdata=pdtest,type="response")
glm.pred = prediction(glm.preds,pdtest$income)
glm.perf = performance(glm.pred,"tpr","fpr")

# Random Forest
bestmtry <- tuneRF(pd[-13],pd$income, ntreeTry=100, stepFactor=1.5,improve=0.01,
trace=TRUE, plot=TRUE, dobest=FALSE)
rf.fit <-randomForest(income~.,data=pd, mtry=2, ntree=1000, keep.forest=TRUE,
importance=TRUE,test=pdtest)
rf.preds = predict(rf.fit,type="prob",newdata=pdtest)[,2]
rf.pred = prediction(rf.preds, pdtest$income)
rf.perf = performance(rf.pred,"tpr","fpr")


# CART Trees
mycontrol = rpart.control(cp = 0, xval = 10)
tree.fit = rpart(income~., method = "class",data = pd, control = mycontrol)
tree.fit$cptable
tree.cptarg = sqrt(tree.fit$cptable[8,1]*tree.fit$cptable[9,1])
tree.prune = prune(tree.fit,cp=tree.cptarg)
tree.preds = predict(tree.prune,newdata=pdtest,type="prob")[,2]
tree.pred = prediction(tree.preds,pdtest$income)
tree.perf = performance(tree.pred,"tpr","fpr")

# Neural Network
nnet.fit = nnet(income~.,data=pd,size=2,maxit=10000,decay=.001)
nnet.preds = predict(nnet.fit,newdata=pdtest,type="raw")
nnet.pred = prediction(nnet.preds,pdtest$income)
nnet.perf = performance(nnet.pred,"tpr","fpr")

# Plotting ROC Curves p
plot(glm.perf,col=2,lwd=2,main="ROC Curve for Classifiers on Adult Dataset")
plot(rf.perf,col=3,lwd=2,add=T)
plot(tree.perf,lwd=2,col=4,add=T)
plot(nnet.perf,lwd=2,col=5,add=T)
abline(a=0,b=1,lwd=2,lty=2,col="gray")
legend("bottomright",col=c(2:5),lwd=2,legend=c("logit","RF","CART","Neural Net"),bty='n')

#tree
library(tree)
attach(pd)
tree.pd=tree(income~.,pd)
summary(tree.pd)
```

```
tree.pred=predict(tree.pd,pdtest,type="class")
table(tree.pred,pdtest$income)
plot(tree.pd)
text(tree.pd,pretty=0)
```