Final Project: Adult Income in the U.S.

Ali Malenchik

February 23, 2020

# Data Import

#set working directory  
setwd("/Users/alimalenchik/Documents/Grad\_School/DSC\_520/Final\_Project/Salary\_Prediction")  
  
#store CSV files into data frames (remove whitespace)  
test\_df <- read.csv("adult-test.csv", skip=1, header=FALSE, strip.white=TRUE) #ignore first line having a comment in the csv  
train\_df <- read.csv("adult-train.csv", header=FALSE, strip.white=TRUE)

# Data Cleansing & Standardization

**Add column names to the data frames**

#Add column names  
colnames(train\_df) <- c("Age","Workclass","fnlwgt","Education","Education\_number","Marital\_status","Occupation","Relationship","Race","Gender","Capital\_gain","Capital\_loss","Hours\_per\_week","Native\_country","Salary")  
  
colnames(test\_df) <- c("Age","Workclass","fnlwgt","Education","Education\_number","Marital\_status","Occupation","Relationship","Race","Gender","Capital\_gain","Capital\_loss","Hours\_per\_week","Native\_country","Salary")

**Filter out records having missing data**

Since the dataset is relatively large, we will eliminate any rows containing missing values in order to reduce error in the analysis.

#filter out any rows containing missing values ("?") and remove levels with 0 records from factor variables  
train\_df.clean <- train\_df %>% filter(rowSums(train\_df=="?") == 0) %>% droplevels()  
test\_df.clean <- test\_df %>% filter(rowSums(test\_df=="?") == 0) %>% droplevels()

2399 records have been removed from the training data set.

1221 records have been removed from the test data set.

**Modify Native Country Variable**

I updated Native\_country to a binary variable “Native\_country.USA” indicating whether the person is native to the U.S. or not. I believe making this modification will allow for a more accurate model and there will be more meaningful relationships with the predictor.

#add new variable Native\_country.USA  
train\_df.clean <- train\_df.clean %>% mutate(Native\_country.USA=as.factor(ifelse(Native\_country=="United-States", "USA", "Other")))  
  
test\_df.clean <- test\_df.clean %>% mutate(Native\_country.USA=as.factor(ifelse(Native\_country=="United-States", "USA", "Other")))

**Fix Salary Variable**

Upon summarizing the data I noticed one issue: The values for Salary bracket differ between the test and train dataset.

#compare Salary variables  
summary(train\_df.clean$Salary)

## <=50K >50K   
## 22654 7508

summary(test\_df.clean$Salary)

## <=50K. >50K.   
## 11360 3700

In order to accurately predict the Salary bracket and enable us to compare the prediction against the test dataset, the values need to match. I will strip the “.” from the Salary attribute.

#remove "." from test\_df.clean$Salary  
test\_df.clean$Salary<-gsub("\\.","",test\_df.clean$Salary) %>% as.factor  
  
#compare Salary variables again  
summary(train\_df.clean$Salary)

## <=50K >50K   
## 22654 7508

summary(test\_df.clean$Salary)

## <=50K >50K   
## 11360 3700

**Remove unnecessary attributes**

The following fields will be removed as they are either difficult to measure or not relevant to predicting Salary bracket: fnlwgt, Marital\_status, Relationship, Capital\_gain, Capital\_loss, Hours\_per\_week. I also removed Education since we will use “Education\_number” in its place, along with Native\_country as we will use “Native\_country.USA” instead.

#select only relevant fields  
train\_df.clean <- train\_df.clean %>% select("Salary","Age","Workclass","Education\_number","Occupation","Race","Gender","Native\_country.USA")  
   
test\_df.clean <- test\_df.clean %>% select("Salary","Age","Workclass","Education\_number","Occupation","Race","Gender","Native\_country.USA")

# Exploratory Analysis

**Explore the clean training data**

#view the clean structure  
str(train\_df.clean)

## 'data.frame': 30162 obs. of 8 variables:  
## $ Salary : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 2 2 ...  
## $ Age : int 39 50 38 53 28 37 49 52 31 42 ...  
## $ Workclass : Factor w/ 7 levels "Federal-gov",..: 6 5 3 3 3 3 3 5 3 3 ...  
## $ Education\_number : int 13 13 9 7 13 14 5 9 14 13 ...  
## $ Occupation : Factor w/ 14 levels "Adm-clerical",..: 1 4 6 6 10 4 8 4 10 4 ...  
## $ Race : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 1 2 1 2 ...  
## $ Native\_country.USA: Factor w/ 2 levels "Other","USA": 2 2 2 2 1 2 1 2 2 2 ...

#summarize the data  
summary(train\_df.clean)

## Salary Age Workclass Education\_number  
## <=50K:22654 Min. :17.00 Federal-gov : 943 Min. : 1.00   
## >50K : 7508 1st Qu.:28.00 Local-gov : 2067 1st Qu.: 9.00   
## Median :37.00 Private :22286 Median :10.00   
## Mean :38.44 Self-emp-inc : 1074 Mean :10.12   
## 3rd Qu.:47.00 Self-emp-not-inc: 2499 3rd Qu.:13.00   
## Max. :90.00 State-gov : 1279 Max. :16.00   
## Without-pay : 14   
## Occupation Race Gender   
## Prof-specialty :4038 Amer-Indian-Eskimo: 286 Female: 9782   
## Craft-repair :4030 Asian-Pac-Islander: 895 Male :20380   
## Exec-managerial:3992 Black : 2817   
## Adm-clerical :3721 Other : 231   
## Sales :3584 White :25933   
## Other-service :3212   
## (Other) :7585   
## Native\_country.USA  
## Other: 2658   
## USA :27504   
##   
##   
##   
##   
##

#view the proportions for Salary bracket  
train\_df.clean$Salary %>% table %>% prop.table

## .  
## <=50K >50K   
## 0.7510775 0.2489225

test\_df.clean$Salary %>% table %>% prop.table

## .  
## <=50K >50K   
## 0.7543161 0.2456839

Based on the proportion tables we can see that approximately 75% of people in both data sets have income <=50K; the dataset is imbalanced.

**Determine significance of each variable in relation to Salary**

#Perform chi-squared test for categorical variables  
chisq.out <- train\_df.clean %>% select("Workclass","Occupation","Race","Gender","Native\_country.USA") %>% map(~chisq.test(.x, train\_df.clean$Salary))  
  
#perform correlation test for numeric variables  
cor.test.out <- train\_df.clean %>% select("Age","Education\_number") %>% map(~cor.test(.x, as.numeric(train\_df.clean$Salary)))  
  
#store attributes and their p-values in a data frame  
attribute\_name <- c("Workclass", "Occupation", "Race", "Gender", "Native\_country.USA", "Age", "Education\_number")  
p\_value <- c(chisq.out$Workclass$p.value, chisq.out$Occupation$p.value, chisq.out$Race$p.value, chisq.out$Gender$p.value, chisq.out$Native\_country.USA$p.value, cor.test.out$Age$p.value, cor.test.out$Education\_number$p.value)  
kable(data.frame(attribute\_name, p\_value), digits = 20, caption="P Value of Each Attribute in relation to Salary Bracket")

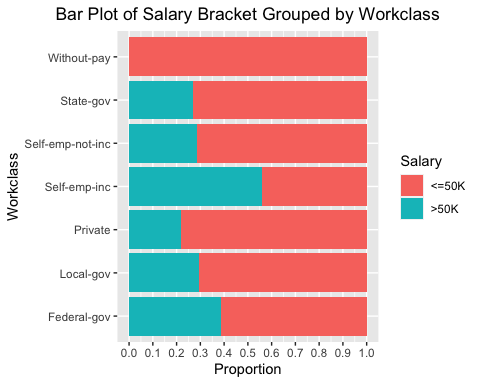
P Value of Each Attribute in relation to Salary Bracket

|  |  |
| --- | --- |
| attribute\_name | p\_value |
| Workclass | 0.00000e+00 |
| Occupation | 0.00000e+00 |
| Race | 0.00000e+00 |
| Gender | 0.00000e+00 |
| Native\_country.USA | 3.42935e-12 |
| Age | 0.00000e+00 |
| Education\_number | 0.00000e+00 |

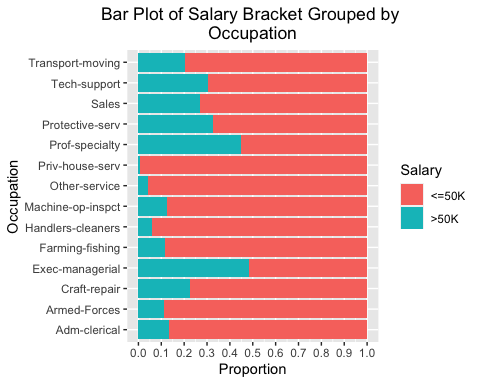
Results: Since all variables have a p-Value less than the significance level of 0.05, each of the variables is significantly associated with Salary bracket.

**Visualize Variable Relationships with Salary Bracket**

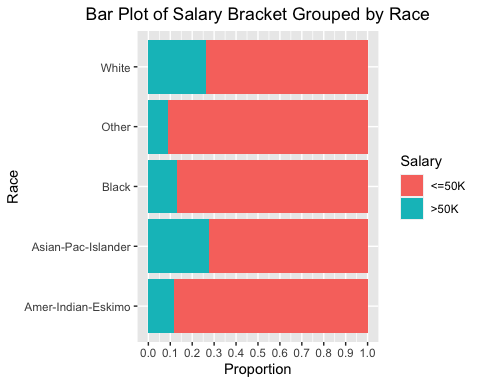
#common elements present in all plots  
my\_theme <- list(  
 geom\_bar(position="fill"), #bar plot  
 ylab("Proportion"), #add y axis label  
 scale\_y\_continuous(breaks = seq(0, 1, .1), limits = c(0, 1)), #update y axis tick marks  
 coord\_flip(), #flip x & y axis for readability of labels  
 theme(plot.title = element\_text(hjust = 0.5)) #center title  
 )  
  
ggplot(train\_df.clean,aes(x=Workclass,fill=Salary)) + ggtitle("Bar Plot of Salary Bracket Grouped by Workclass") + my\_theme



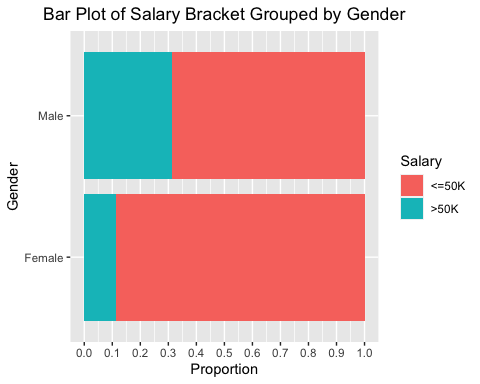
ggplot(train\_df.clean,aes(x=Occupation,fill=Salary)) + ggtitle("Bar Plot of Salary Bracket Grouped by \nOccupation") + my\_theme



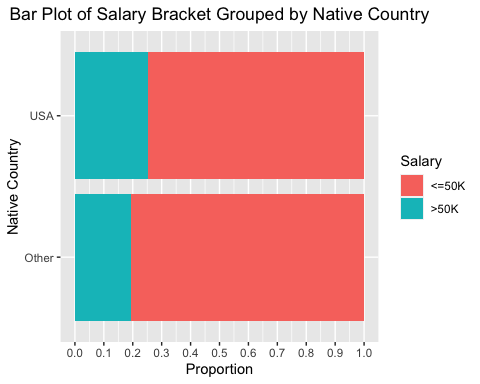
ggplot(train\_df.clean,aes(x=Race,fill=Salary)) + ggtitle("Bar Plot of Salary Bracket Grouped by Race") + my\_theme



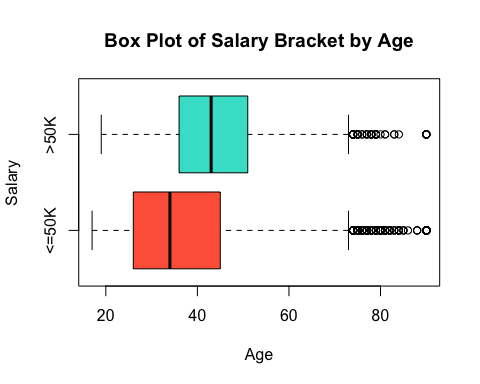
ggplot(train\_df.clean,aes(x=Gender,fill=Salary)) + ggtitle("Bar Plot of Salary Bracket Grouped by Gender") + my\_theme



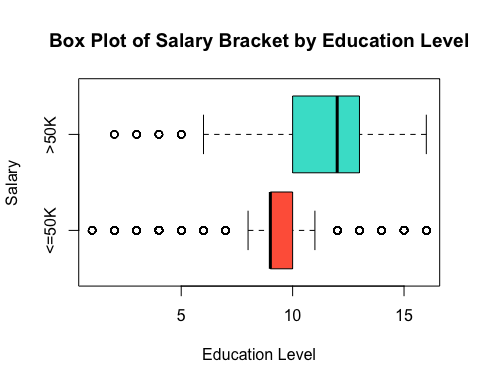
ggplot(train\_df.clean,aes(x=Native\_country.USA,fill=Salary)) + ggtitle("Bar Plot of Salary Bracket Grouped by Native Country") + xlab("Native Country") + my\_theme



boxplot(Age~Salary,data=train\_df.clean, horizontal=TRUE, col=c("tomato","turquoise"), main="Box Plot of Salary Bracket by Age")



boxplot(Education\_number~Salary,data=train\_df.clean, horizontal=TRUE, col=c("tomato","turquoise"), main="Box Plot of Salary Bracket by Education Level", xlab="Education Level")



# Model 1: Generalized Linear Model

**Step 1. Create the model**

#>update Salary to numeric  
train\_df.glm <- train\_df.clean  
train\_df.glm$Salary <- as.numeric(train\_df.clean$Salary == ">50K") #50K will correspond to 1, <=50K will correspond to 0  
  
test\_df.glm <- test\_df.clean  
test\_df.glm$Salary <- as.numeric(test\_df.clean$Salary == ">50K") #50K will correspond to 1, <=50K will correspond to 0  
  
#create the model  
mdl.glm <- glm(Salary ~ ., family="binomial", data=train\_df.glm)

**Step 3. Explore the model**

#summarize model  
summary(mdl.glm)

##   
## Call:  
## glm(formula = Salary ~ ., family = "binomial", data = train\_df.glm)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4651 -0.6709 -0.3810 -0.0441 3.2353   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.295849 0.247261 -29.507 < 2e-16 \*\*\*  
## Age 0.042813 0.001293 33.103 < 2e-16 \*\*\*  
## WorkclassLocal-gov -0.518285 0.097108 -5.337 9.44e-08 \*\*\*  
## WorkclassPrivate -0.380958 0.081164 -4.694 2.68e-06 \*\*\*  
## WorkclassSelf-emp-inc 0.142467 0.107061 1.331 0.18329   
## WorkclassSelf-emp-not-inc -0.652942 0.095547 -6.834 8.27e-12 \*\*\*  
## WorkclassState-gov -0.775987 0.107753 -7.202 5.95e-13 \*\*\*  
## WorkclassWithout-pay -11.287428 78.018198 -0.145 0.88497   
## Education\_number 0.281166 0.008238 34.129 < 2e-16 \*\*\*  
## OccupationArmed-Forces -1.031487 1.125508 -0.916 0.35942   
## OccupationCraft-repair 0.224088 0.069193 3.239 0.00120 \*\*   
## OccupationExec-managerial 0.994976 0.064750 15.366 < 2e-16 \*\*\*  
## OccupationFarming-fishing -0.644416 0.121902 -5.286 1.25e-07 \*\*\*  
## OccupationHandlers-cleaners -0.869508 0.129834 -6.697 2.13e-11 \*\*\*  
## OccupationMachine-op-inspct -0.115755 0.091066 -1.271 0.20369   
## OccupationOther-service -1.078163 0.106569 -10.117 < 2e-16 \*\*\*  
## OccupationPriv-house-serv -2.489757 1.021758 -2.437 0.01482 \*   
## OccupationProf-specialty 0.653563 0.067090 9.742 < 2e-16 \*\*\*  
## OccupationProtective-serv 0.674816 0.109698 6.152 7.67e-10 \*\*\*  
## OccupationSales 0.387116 0.069524 5.568 2.58e-08 \*\*\*  
## OccupationTech-support 0.559440 0.095152 5.879 4.12e-09 \*\*\*  
## OccupationTransport-moving 0.156353 0.087132 1.794 0.07274 .   
## RaceAsian-Pac-Islander 0.484501 0.220217 2.200 0.02780 \*   
## RaceBlack 0.192304 0.206115 0.933 0.35082   
## RaceOther -0.129503 0.323924 -0.400 0.68931   
## RaceWhite 0.545605 0.197201 2.767 0.00566 \*\*   
## GenderMale 1.282173 0.041260 31.075 < 2e-16 \*\*\*  
## Native\_country.USAUSA 0.130236 0.068754 1.894 0.05820 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 33851 on 30161 degrees of freedom  
## Residual deviance: 25726 on 30134 degrees of freedom  
## AIC: 25782  
##   
## Number of Fisher Scoring iterations: 11

Based on the results of the model, some aspects of each variable seem to be significant predictors of Salary bracket. The negative coefficients indicate that the characteristic is less likely to have a Salary greater than 50K.

The residual deviance is a significant decrease from the null deviance, meaning including the independent variables improved the model.

**Step 4. Calculate Accuracy of the Model**

#calculate probability for every observation  
test\_df.glm$prob <- predict(mdl.glm, test\_df.glm, type = "response")  
  
#transform probabilities into successes and failures (1’s and 0’s) with a threshold of .5  
test\_df.glm <- test\_df.glm %>% mutate(pred = 1\*(prob > .5) + 0)  
  
#compare Salary vs pred  
test\_df.glm <- test\_df.glm %>% mutate(accurate = 1\*(pred == Salary))  
  
#compute accuracy  
accuracy.glm <- round(((sum(test\_df.glm$accurate)/nrow(test\_df.glm))\*100),2)

The accuracy of the generalized linear model is 79.97%.

# Model 2: Random Forest Classifier

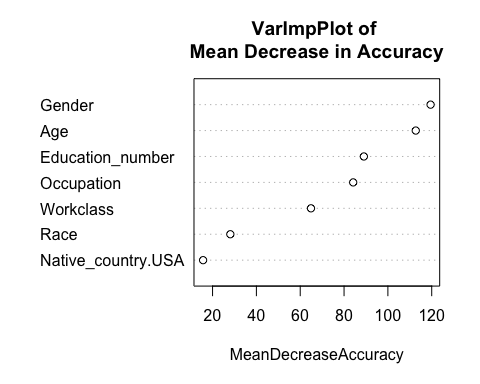
**Step 1. Create the model**

#set seed for reproduceability  
set.seed(51)  
  
#train the model  
(mdl.rf <- randomForest(Salary ~ ., data=train\_df.clean, importance=TRUE, ntree=500))

##   
## Call:  
## randomForest(formula = Salary ~ ., data = train\_df.clean, importance = TRUE, ntree = 500)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 2  
##   
## OOB estimate of error rate: 18.94%  
## Confusion matrix:  
## <=50K >50K class.error  
## <=50K 20895 1759 0.07764633  
## >50K 3954 3554 0.52663825

**Step 2. Explore the model**

#view importance of variables  
varImpPlot(mdl.rf,type=1,main="VarImpPlot of \nMean Decrease in Accuracy")



A high decrease in accuracy is expected for very predictive variables. We can see that removing Gender or Age from the model would result in a large decrease in accuracy, whereas removing Race or Native\_country.USA would be less impactful.

**Step 3. Calculate Accuracy of the Model**

(confmat.rf <- predict(mdl.rf, test\_df.clean) #make predictions  
 %>% table(test\_df.clean[,1]) #create table of true values vs predictions  
 %>% confusionMatrix) #run confusion matrix

## Confusion Matrix and Statistics  
##   
##   
## . <=50K >50K  
## <=50K 10441 1969  
## >50K 919 1731  
##   
## Accuracy : 0.8082   
## 95% CI : (0.8019, 0.8145)  
## No Information Rate : 0.7543   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.4279   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9191   
## Specificity : 0.4678   
## Pos Pred Value : 0.8413   
## Neg Pred Value : 0.6532   
## Prevalence : 0.7543   
## Detection Rate : 0.6933   
## Detection Prevalence : 0.8240   
## Balanced Accuracy : 0.6935   
##   
## 'Positive' Class : <=50K   
##

#compute accuracy  
accuracy.rf <- round(((confmat.rf$overall[['Accuracy']])\*100),2)

The accuracy of the random forest model is 80.82%.

# Model 3: Classification Tree using RPart

**Step 1. Create the model**

mdl.rpart <- rpart(Salary ~ ., data=train\_df.clean, method = 'class', minsplit = 5, cp=-1) #cp=-1 to fully grow the tree  
  
#calculate cp of smallest tree that minimizes prediction error  
bestcp <- mdl.rpart$cptable[which.min(mdl.rpart$cptable[,"xerror"]),"CP"]  
  
#prune tree  
mdl.rpart <- prune(mdl.rpart, bestcp)

**Step 2. Explore the model**

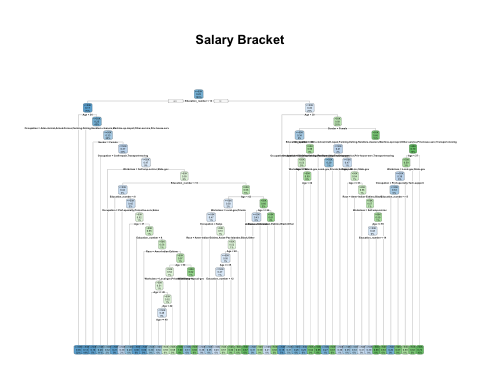
#view cross validation results  
printcp(mdl.rpart)

##   
## Classification tree:  
## rpart(formula = Salary ~ ., data = train\_df.clean, method = "class",   
## minsplit = 5, cp = -1)  
##   
## Variables actually used in tree construction:  
## [1] Age Education\_number Gender Occupation   
## [5] Race Workclass   
##   
## Root node error: 7508/30162 = 0.24892  
##   
## n= 30162   
##   
## CP nsplit rel error xerror xstd  
## 1 0.06346564 0 1.00000 1.00000 0.0100018  
## 2 0.01704848 3 0.80874 0.81007 0.0092811  
## 3 0.00381815 4 0.79169 0.79382 0.0092107  
## 4 0.00359616 10 0.76878 0.78330 0.0091644  
## 5 0.00193127 11 0.76518 0.76958 0.0091030  
## 6 0.00159830 13 0.76132 0.77104 0.0091097  
## 7 0.00119872 18 0.75333 0.77118 0.0091103  
## 8 0.00113213 19 0.75213 0.76971 0.0091037  
## 9 0.00093234 21 0.74987 0.77025 0.0091061  
## 10 0.00066596 25 0.74614 0.77038 0.0091067  
## 11 0.00059936 32 0.74134 0.76958 0.0091030  
## 12 0.00053277 35 0.73948 0.76851 0.0090982  
## 13 0.00044397 41 0.73628 0.77051 0.0091073  
## 14 0.00043287 47 0.73362 0.76518 0.0090832

#view variable importance  
data.frame(importance = mdl.rpart$variable.importance) %>% kable

|  |  |
| --- | --- |
|  | importance |
| Education\_number | 1295.664449 |
| Age | 904.286820 |
| Occupation | 815.621322 |
| Gender | 382.352837 |
| Workclass | 77.387929 |
| Race | 25.536061 |
| Native\_country.USA | 5.775353 |

#plot the model  
rpart.plot(mdl.rpart, main = "Salary Bracket")



**Step 3. Calculate Accuracy of the Model**

(confmat.rpart <- predict(mdl.rpart, type="class",newdata=test\_df.clean) #make predictions  
 %>% table(test\_df.clean[,1]) #create table of true values vs predictions  
 %>% confusionMatrix) #run confusion matrix

## Confusion Matrix and Statistics  
##   
##   
## . <=50K >50K  
## <=50K 10438 1978  
## >50K 922 1722  
##   
## Accuracy : 0.8074   
## 95% CI : (0.801, 0.8137)  
## No Information Rate : 0.7543   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.4252   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9188   
## Specificity : 0.4654   
## Pos Pred Value : 0.8407   
## Neg Pred Value : 0.6513   
## Prevalence : 0.7543   
## Detection Rate : 0.6931   
## Detection Prevalence : 0.8244   
## Balanced Accuracy : 0.6921   
##   
## 'Positive' Class : <=50K   
##

#compute accuracy  
accuracy.rpart <- round(((confmat.rpart$overall[['Accuracy']])\*100),2)

The accuracy of the model is 80.74%.

# Compare Model Accuracies

(data.frame("Model" = c("GLM","Random Forest","RPART"), "Accuracy" = c(accuracy.glm, accuracy.rf, accuracy.rpart)) %>% kable)

|  |  |
| --- | --- |
| Model | Accuracy |
| GLM | 79.97 |
| Random Forest | 80.82 |
| RPART | 80.74 |