Prediction Model for Income Classification

Under $50000 vs Over $50000

steve dubois

4/22/2019

ABSTRACT

The prominent inequality of wealth and income is a huge concern especially in the United States. The likelihood of diminishing poverty is one valid reason to reduce the world's surging level of economic inequality. The principle of universal moral equality ensures sustainable development and improves the economic stability of a nation. Governments in different countries have been trying their best to address this problem and provide an optimal solution. The aim here to show the usage of machine learning techniques in providing a solution to the income equality problem. The UCI Adult Dataset has been used for the purpose. Specifically, several machine learning classification models have been compared to predict whether a person's yearly income in the US falls in the income category of either greater than 50K dollars or less/equal to 50K dollars category based on a certain set of attributes. So, what_Y(>50, <=50) is predicted given (X1, X2, X3,...Xn), where Y is an income level, and X is a statistic feature of an individual.

LIBRRARIES USED - R PACKAGES

```
library(knitr)
library(ggvis)
library(ISLR)
library(e1071)
library(gmodels)
library(tidyverse)
library(tidyr)
library(dplyr)
library(readr)
library(ggplot2)
library(randomForest)
library(caret)
library(data.table)
library(gbm)
library(rpart)
library(rpart.plot)
library(plotly)
library(ggvis)
library(neuralnet)
library(nnet)
library(MASS)
library(devtools)
install_github('araastat/reprtree', force = TRUE)
```

```
##
##
   checking for file ‘/private/var/folders/xc/v9gh8m7j65n8yj0gq2_m5v8r0000gn/T/RtmprN0kgm/remotes4b8542

v  checking for file ‘/private/var/folders/xc/v9gh8m7j65n8yj0gq2_m5v8r0000gn/T/RtmprN0kgm/remotes4b8542
```

```
##

-  preparing 'reprtree':
##

   checking DESCRIPTION meta-information ...

v  checking DESCRIPTION meta-information
##

-  checking for LF line-endings in source and make files and shell scripts
##

-  checking for empty or unneeded directories
##

-  building 'reprtree_0.6.tar.gz'
##


##
```

```
  library(reprtree)
  library(rattle)
  library(NeuralNetTools)
```

LOADING CENSUS DATA

```
url.train <- "http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"
url.test <- "http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test"
url.names <- "http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names"
download.file(url.train, destfile = "adult_train.csv")
download.file(url.test, destfile = "adult_test.csv")
download.file(url.names, destfile = "adult_names.txt")


# Read the training and test data into memory
train <- read.csv("adult_train.csv", header = FALSE)

# The test data has an unnecessary first line that messes stuff up, this fixes that problem
all_content <- readLines("adult_test.csv")
skip_first <- all_content[-1]
test <- read.csv(textConnection(skip_first), header = FALSE)
```

Initializing headers

```
feature <- c("Age",
             "Work_Class",
             "Final_Weight",
             "Education",
             "Education_Num",
```

```
            "Marital_Status",
            "Occupation",
            "Relationship",
            "Race",
            "Sex",
            "Capital_Gain",
            "Capital_Loss",
            "Hours_Per_Week",
            "Native_Country",
            "IncomeCLASS")
```

```
##
##  TRUE
## 32561
```

```
##
## Under_50K  More_50K
##        76        24
```

```
## [1] 0
```

```
## [1] 0
```

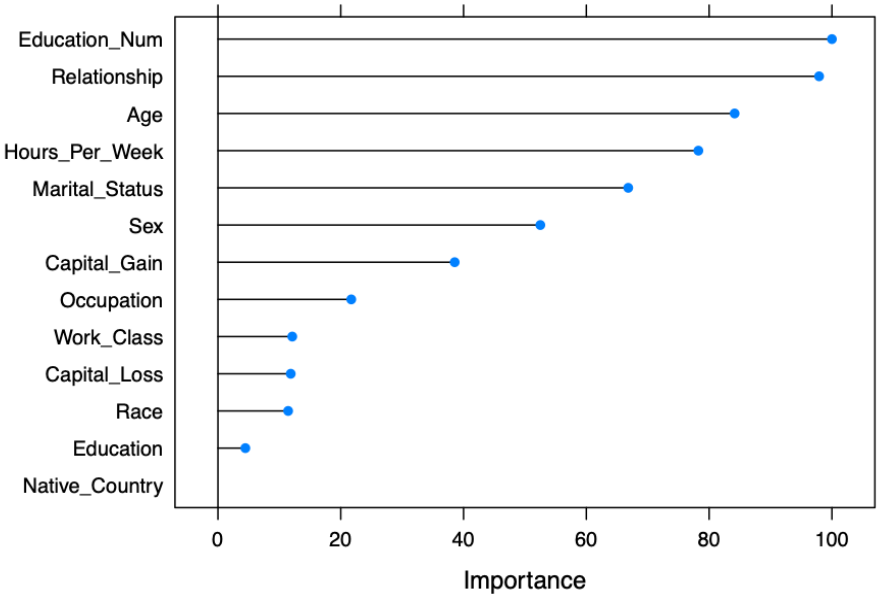
LINEAR DISCRIMINANT ANALYSIS

Identification the Signigicant Demographic Feaures

```
set.seed(1414)
model.lda <- train(IncomeCLASS ~ .,
                   data = train,
                   method = "lda")
```

RANKING FEATURE SIGNIFICANCE TO INCOME CLASSS

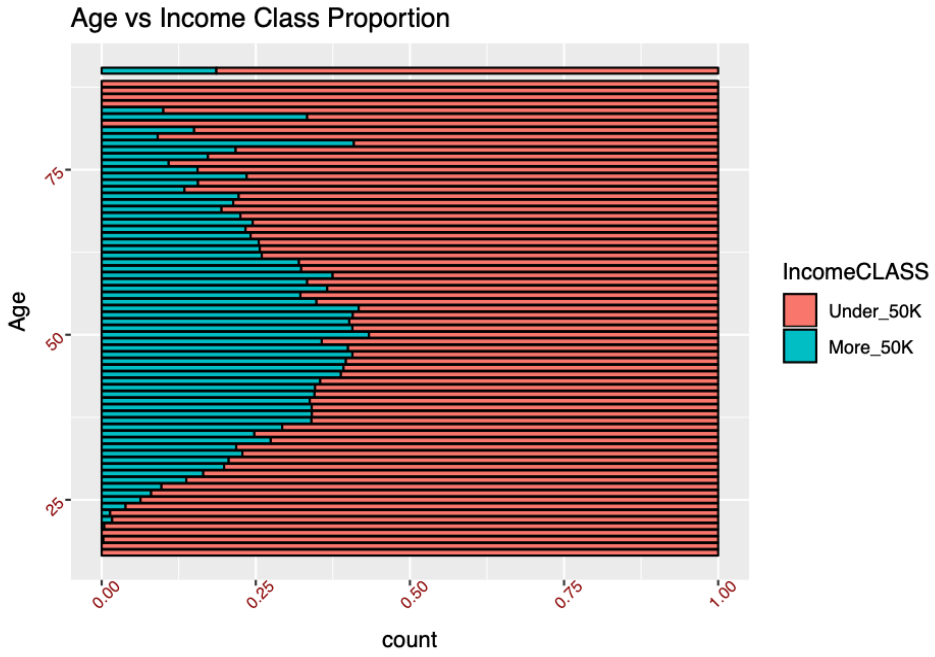
```
plot(varImp(model.lda))
```



SETTING CATEGORICAL; FEATURES

EXPLORATORY DATA ANALYSIS USING GGPLOT

```
P <- ggplot(train,aes(x = Age, fill = IncomeCLASS)) + geom_bar(position = "fill", color = "black") + co
P1 <- P + labs(title = "Age vs Income Class Proportion")
P1
```

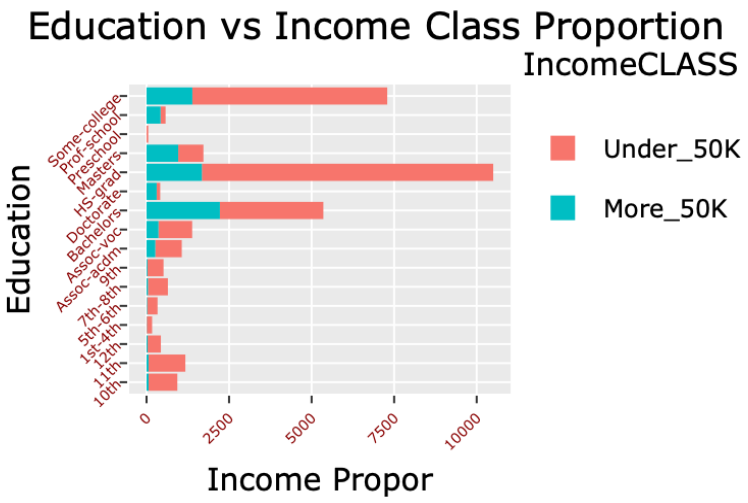



```
gg <- train %>% ggvis(x = ~Age) %>% layer_histograms(fill = "green")
gg
```

Renderer: SVG | Canvas

Download

```
Q <- ggplot(train,aes(x = Education, fill = IncomeCLASS)) + geom_bar() + ylab("Income Propor") +  coord_
Q1 <- Q + labs(title = "Education vs Income Class Proportion") + theme(axis.text = element_text(colour =
ggplotly(Q1 = ggplot2::last_plot())
```

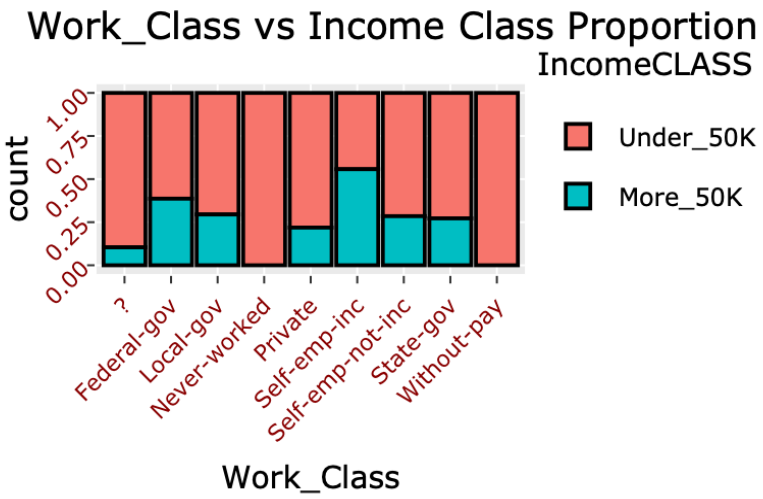



```
train %>% ggvis(x = ~Age) %>% layer_histograms() %>% layer_paths( strokeWidth := 1, stroke := "red")
```

Renderer: SVG | Canvas

Download

```
R <- train %>% ggplot(aes(x = Work_Class, fill = IncomeCLASS)) + geom_bar(position = "fill", color = "b
R1 <- R + labs(title = "Work_Class vs Income Class Proportion")
ggplotly(R1 = ggplot2::last_plot())
```



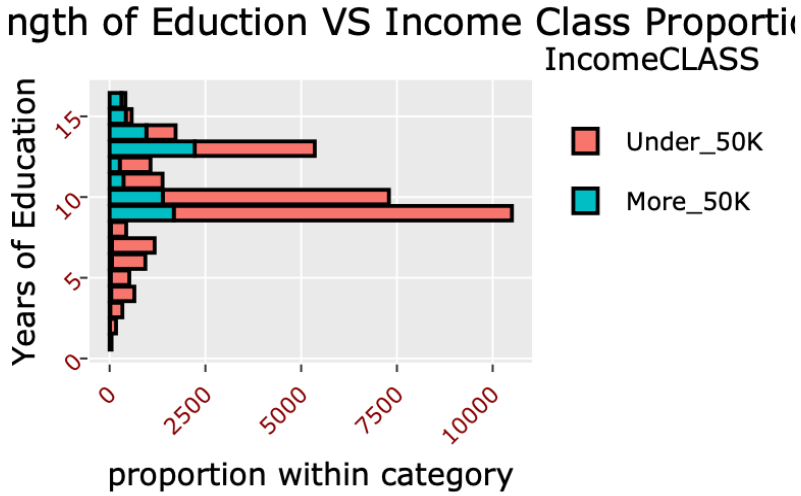
We find that the people employed in private companies have more people with income above 50k and Self Employed people having a higher proportion of people with income greater than 50k.

```
hh <- train %>% ggvis(x = ~Education_Num) %>% layer_histograms(fill = "green")
hh
```

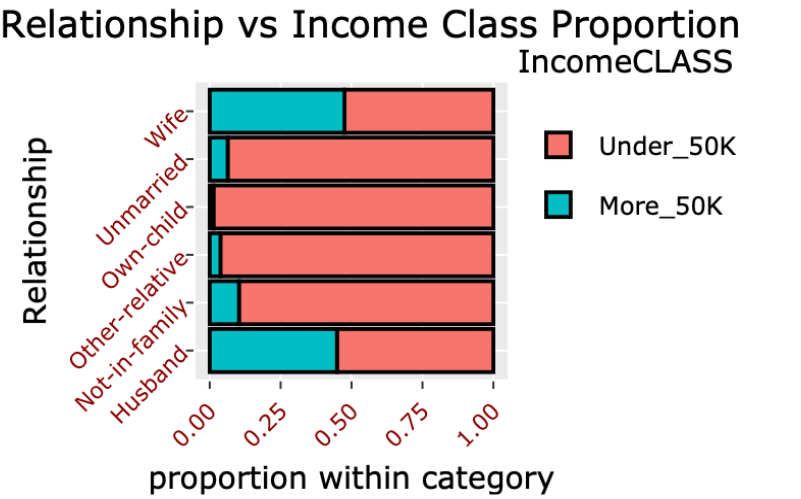
Renderer: SVG | Canvas

Download

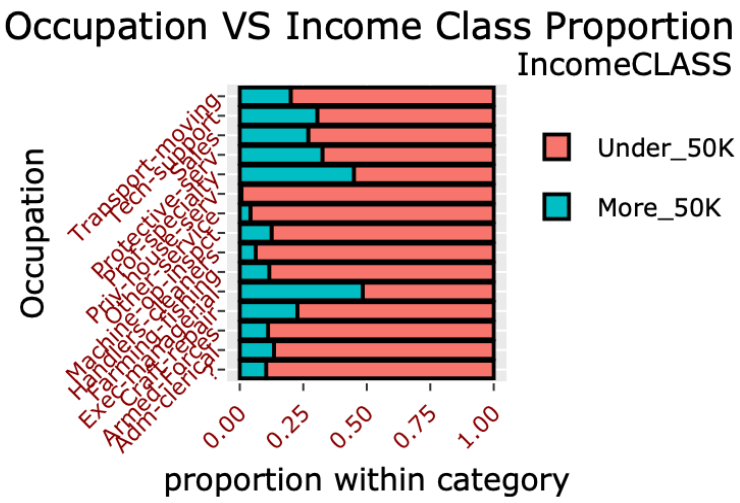
```
S <- ggplot(train,aes(x = Education_Num, fill = IncomeCLASS)) + xlab("Years of Education") + ylab("prop
S1 <- S + labs(title = "Length of Eduction VS Income Class Proportion")
ggplotly(S1 = ggplot2::last_plot())
```




```
T <- ggplot(train,aes(x = Relationship, fill = IncomeCLASS)) + ggtitle("Relationship VS Income Class Pr
T1 <- T + labs(title = "Relationship vs Income Class Proportion")
ggplotly(T1 = ggplot2::last_plot())
```




```
jj <- ggplot(train,aes(x = Occupation, fill = IncomeCLASS)) + ggtitle("Occupation VS Income Class Propo
ggplotly(jj = ggplot2::last_plot())
```



PERFORMANCE METRICS for PREDICTIVE POWER DETERMINATION & MODEL SELECTION:

Accuracy Statistic: The higher, the better. (0:1)

Accuracy defined more simply: # of correction predictions / total predictions

Accuracy defined in terms of true positives/negatives: TN + FN / # SAMPLES

Kappa Statistic: The higher, the better. (0:1)

NAIVE BAYES MODEL

```
#train Naive Bayes
set.seed(32323)
model_Naive <- naiveBayes(IncomeCLASS ~ ., data = train)
pred_Nb <- predict(model_Naive, test)
levels(train$IncomeCLASS) <- c("Under_50K", "More_50K")
levels(test$IncomeCLASS) <- c("Under_50K", "More_50K")
levels(pred_Nb) <- c("Under_50K", "More_50K")
confusionMatrix(pred_Nb, test$IncomeCLASS)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Under_50K More_50K
##   Under_50K     11567     1978
##   More_50K        868     1868
##
```

```
##                Accuracy : 0.8252
##                  95% CI : (0.8193, 0.831)
##     No Information Rate : 0.7638
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4619
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9302
##             Specificity : 0.4857
##          Pos Pred Value : 0.8540
##          Neg Pred Value : 0.6827
##              Prevalence : 0.7638
##          Detection Rate : 0.7105
##    Detection Prevalence : 0.8320
##       Balanced Accuracy : 0.7079
##
##        'Positive' Class : Under_50K
##
```

```
CrossTable(pred_Nb, test$IncomeCLASS)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  16281
##
##
##              | test$IncomeCLASS
##      pred_Nb | Under_50K |  More_50K | Row Total |
## -------------|-----------|-----------|-----------|
##    Under_50K |     11567 |      1978 |     13545 |
##              |   144.270 |   466.457 |           |
##              |     0.854 |     0.146 |     0.832 |
##              |     0.930 |     0.514 |           |
##              |     0.710 |     0.121 |           |
## -------------|-----------|-----------|-----------|
##     More_50K |       868 |      1868 |      2736 |
##              |   714.229 |  2309.267 |           |
##              |     0.317 |     0.683 |     0.168 |
##              |     0.070 |     0.486 |           |
##              |     0.053 |     0.115 |           |
## -------------|-----------|-----------|-----------|
## Column Total |     12435 |      3846 |     16281 |
##              |     0.764 |     0.236 |           |
## -------------|-----------|-----------|-----------|
```

```
##
##
```

```
summary(pred_Nb)
```

```
## Under_50K  More_50K
##     13545      2736
```

```
cm_Nb <- data.frame(confusionMatrix(pred_Nb, test$IncomeCLASS)[3])
kable(cm_Nb)
```

	overall
Accuracy	0.8251950
Kappa	0.4619397
AccuracyLower	0.8192724
AccuracyUpper	0.8310012
AccuracyNull	0.7637737
AccuracyPValue	0.0000000
McnemarPValue	0.0000000

TRAIN THE RPART DECISION TREE MODEL

```
# rpart decision tree
set.seed(32323)
model_part <- caret::train(IncomeCLASS ~ ., data = train, method = "rpart")

pred_rpart <- predict(model_part, test, type = "raw")

confusionMatrix(pred_rpart, test$IncomeCLASS)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Under_50K More_50K
##   Under_50K     11808     2042
##   More_50K        627     1804
##
##                Accuracy : 0.8361
##                  95% CI : (0.8303, 0.8417)
##     No Information Rate : 0.7638
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4796
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9496
##             Specificity : 0.4691
##          Pos Pred Value : 0.8526
##          Neg Pred Value : 0.7421
##              Prevalence : 0.7638
##          Detection Rate : 0.7253
##    Detection Prevalence : 0.8507
##       Balanced Accuracy : 0.7093
```

```
##
##        'Positive' Class : Under_50K
##
```

```
cm_rpart <- data.frame(confusionMatrix(pred_rpart, test$IncomeCLASS)[3])
CrossTable(pred_rpart, test$IncomeCLASS)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  16281
##
##
##              | test$IncomeCLASS
##   pred_rpart | Under_50K |  More_50K | Row Total |
## -------------|-----------|-----------|-----------|
##    Under_50K |     11808 |      2042 |     13850 |
##              |   142.958 |   462.215 |           |
##              |     0.853 |     0.147 |     0.851 |
##              |     0.950 |     0.531 |           |
##              |     0.725 |     0.125 |           |
## -------------|-----------|-----------|-----------|
##     More_50K |       627 |      1804 |      2431 |
##              |   814.465 |  2633.353 |           |
##              |     0.258 |     0.742 |     0.149 |
##              |     0.050 |     0.469 |           |
##              |     0.039 |     0.111 |           |
## -------------|-----------|-----------|-----------|
## Column Total |     12435 |      3846 |     16281 |
##              |     0.764 |     0.236 |           |
## -------------|-----------|-----------|-----------|
##
##
```

```
summary(pred_rpart)
```

```
## Under_50K  More_50K
##     13850      2431
```

```
kable(cm_rpart)
```

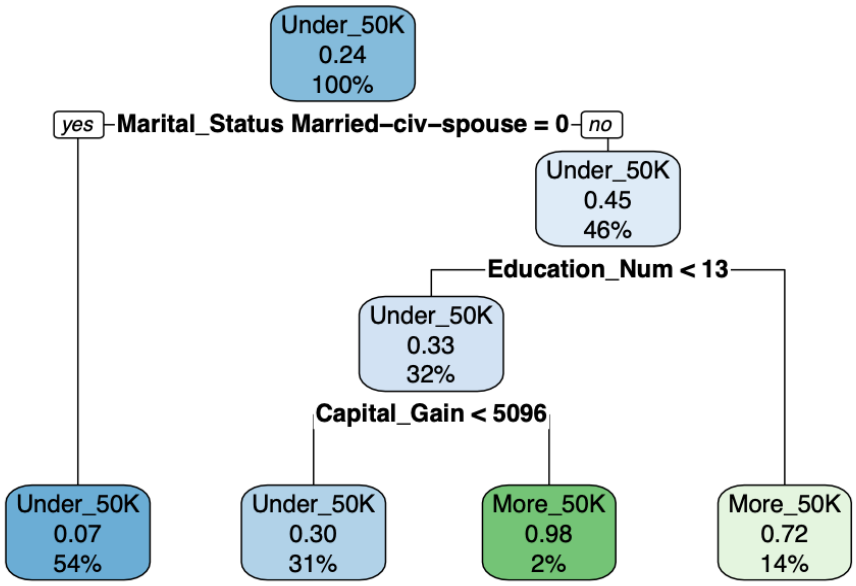
	overall
Accuracy	0.8360666
Kappa	0.4795717
AccuracyLower	0.8302892
AccuracyUpper	0.8417236

	overall
AccuracyNull	0.7637737
AccuracyPValue	0.0000000
McnemarPValue	0.0000000

```
accuracy <- sum(pred_rpart == test$IncomeCLASS)/length(test$IncomeCLASS)
print(accuracy)
```

```
## [1] 0.8360666
```

```
rpart.plot(model_part$finalModel)
```



TRAIN THE RANDOM FOREST MODEL

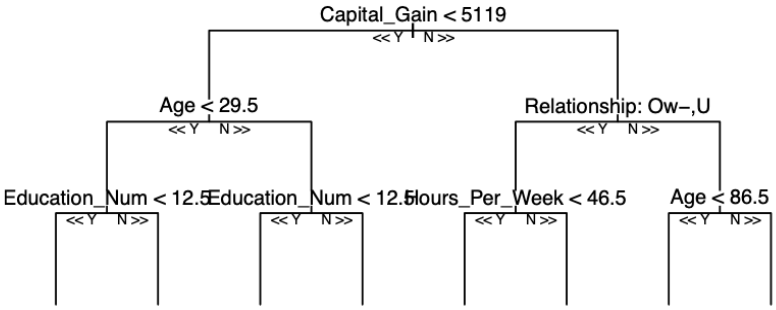
```
set.seed(32323)
model.rf <- randomForest(IncomeCLASS ~ ., data = train, ntree = 500, importance = TRUE, mtry = 2, do.tr
```

```
## ntree      OOB      1      2
##   100:  13.40%  5.93% 36.95%
##   200:  13.35%  5.86% 36.99%
##   300:  13.30%  5.81% 36.88%
##   400:  13.31%  5.87% 36.79%
##   500:  13.34%  5.94% 36.67%
```

```
pred.rf <- predict(model.rf, test)
summary(pred.rf)
```

```
## Under_50K  More_50K
##     13107      3174
```

```
reprtree:::plot.getTree(model.rf, k = 3, depth = 4)
```




```
confusionMatrix(test$IncomeCLASS, pred.rf)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Under_50K More_50K
##   Under_50K     11670      765
##   More_50K       1437     2409
##
##                Accuracy : 0.8648
##                  95% CI : (0.8594, 0.87)
##     No Information Rate : 0.805
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6011
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8904
##             Specificity : 0.7590
##          Pos Pred Value : 0.9385
##          Neg Pred Value : 0.6264
##              Prevalence : 0.8050
##          Detection Rate : 0.7168
##    Detection Prevalence : 0.7638
##       Balanced Accuracy : 0.8247
##
##        'Positive' Class : Under_50K
##
```

```
cm.rf <- data.frame(confusionMatrix(pred.rf, test$IncomeCLASS)[3])
kable(cm.rf)
```

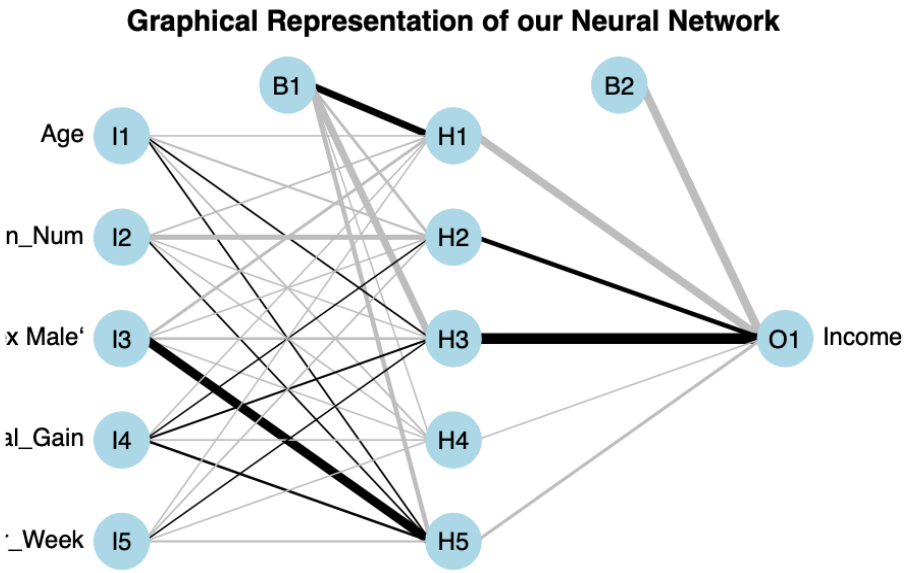
	overall
Accuracy	0.8647503
Kappa	0.6011184
AccuracyLower	0.8594013
AccuracyUpper	0.8699687
AccuracyNull	0.7637737
AccuracyPValue	0.0000000

	overall
McnemarPValue	0.0000000

NEURAL NETWORKS

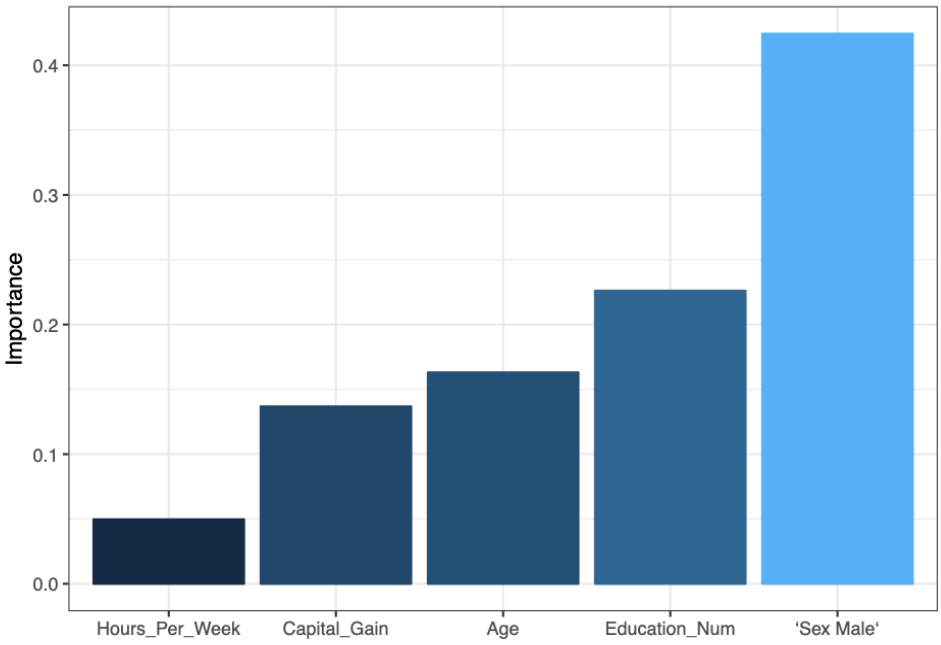
```
set.seed(32323)
keeps <- c("Education_Num",
           "Age",
           "Hours_Per_Week",
           "Sex",
           "Capital_Gain",
           "IncomeCLASS")

train.reduced <- train[,which(names(train) %in% keeps)]
test.reduced <- test[,which(names(test) %in% keeps)]
start <- proc.time()[3]
model.nn <- train(IncomeCLASS ~ .,
                  data = train.reduced,
                  method = "nnet")
```

```
par(mar = c(1, 1, 1, 1))
plotnet(model.nn$finalModel, y_names = "IncomeCLASS")
title("Graphical Representation of our Neural Network")
```




```
predictions <- predict(model.nn, test.reduced[,1:5])
accuracy <- sum(predictions == test.reduced[,6])/length(test.reduced[,6])
print(accuracy)
```

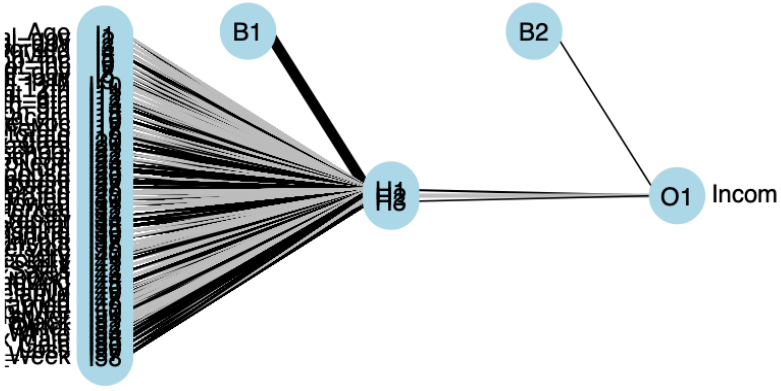
```r
cm.neural <- data.frame(confusionMatrix(predictions, test.reduced$IncomeCLASS)[3])
kable(cm.neural)

garson(model.nn$finalModel)
```




```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Under_50K More_50K
##   Under_50K     11062     1373
##   More_50K       1165     2681
##
##                Accuracy : 0.8441
##                  95% CI : (0.8384, 0.8497)
##     No Information Rate : 0.751
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5759
##  Mcnemar's Test P-Value : 3.976e-05
##
##             Sensitivity : 0.9047
##             Specificity : 0.6613
##          Pos Pred Value : 0.8896
##          Neg Pred Value : 0.6971
##              Prevalence : 0.7510
##          Detection Rate : 0.6794
##    Detection Prevalence : 0.7638
```

```
##        Balanced Accuracy : 0.7830
##
##         'Positive' Class : Under_50K
##
```

	overall
Accuracy	0.8441128
Kappa	0.5759170
AccuracyLower	0.8384486
AccuracyUpper	0.8496537
AccuracyNull	0.7637737
AccuracyPValue	0.0000000
McnemarPValue	0.0000398



	overall
Accuracy	0.8441128
Kappa	0.5759170
AccuracyLower	0.8384486
AccuracyUpper	0.8496537
AccuracyNull	0.7637737
AccuracyPValue	0.0000000
McnemarPValue	0.0000398

MODEL COMPARISON

NAIVE BAYES

```
kable(cm_Nb)
```

	overall
Accuracy	0.8251950
Kappa	0.4619397

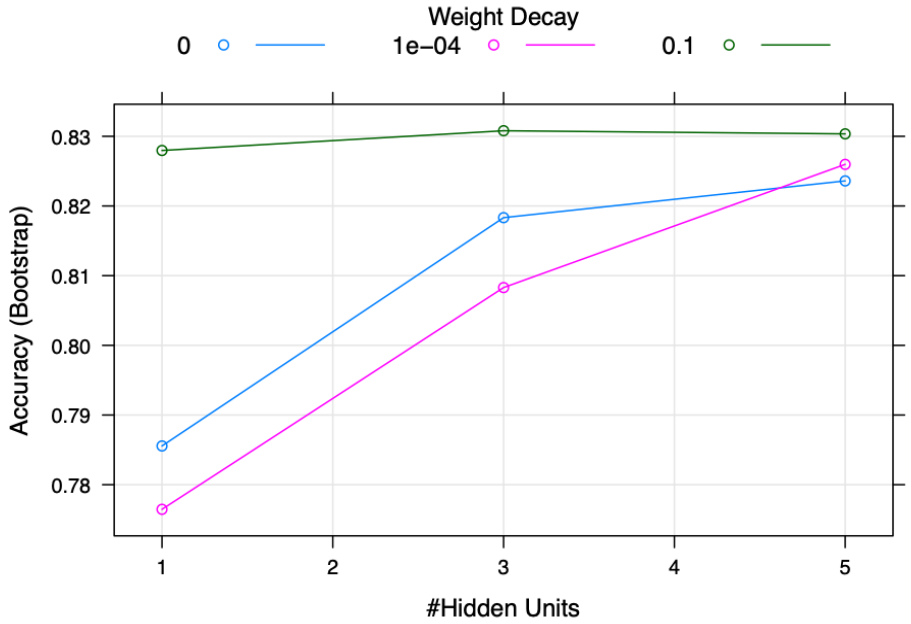
	overall
AccuracyLower	0.8192724
AccuracyUpper	0.8310012
AccuracyNull	0.7637737
AccuracyPValue	0.0000000
McnemarPValue	0.0000000

RPART

```
kable(cm_rpart)
```

	overall
Accuracy	0.8360666
Kappa	0.4795717
AccuracyLower	0.8302892
AccuracyUpper	0.8417236
AccuracyNull	0.7637737
AccuracyPValue	0.0000000
McnemarPValue	0.0000000

NEURAL NETWORK



NEURAL MODEL >>> model_neural_AA <- train(IncomeCLASS ~ ., data = train, method = "nnet", trControl =

RANDOM FOREST

```
kable(cm.rf)
```

	overall
Accuracy	0.8647503
Kappa	0.6011184
AccuracyLower	0.8594013
AccuracyUpper	0.8699687
AccuracyNull	0.7637737
AccuracyPValue	0.0000000
McnemarPValue	0.0000000

CONCLUSIONS:

In this binary context, and its 86% accuracy, the random forest model performed best to classify income levels. If the complexity or dimensionality were greater, I would suspect neural network model would close the accuracy gap.

The analysis confirmed (and quantified) what is considered common sense:

Age, education, occupation, and marital status (or relationship kind) are good for predicting income (above a certain threshold).

(1) if a person earns more than $50000 he is very likely to be a married man with large number of years of education;
(2) single parents, younger than 25 years, who studied less than 10 years, and were never-married make less than $50000.

INFERENCES

About 46% of the people are in a relationship called “Husband” or “Wife” which is then further classified based on Education Level where nearly 14% who earn above $50 K have the education of Bachelors, Prof-school, Masters and Doctorate.

The other education levels have income predominantly below $50 k with just 2% having salaries above $50k who also have capital gains greater than $5096

With respect to other relationships, only 1% have income above $50 k and with capital gains greater than $7074.

In the relationship of Education and Number of People Earning > 50 k and separated by Work Class. We find that Bachelors graduates working in Private companies have a higher number of people earning above 50 k.

In the relationship of Average hours per week with respect to gender and separated by Work Class and we find that Males typically work more hours per week on Average across all work classes.

In the relationship of marital status and income levels separated by Work Class, the majority of the people in Married with Civilian spouse have an income greater than 50 k and majorly in the private sector.

A takeaway in looking at impact of occupation, capital gain and capital loss on the income; we find that Executives at Managerial Level have more people with income greater than 50k and Professional Specialty has more capital gains.