# Census Analysis

Andrew Jensen

2/26/2020

# PREAMBLE

Census income https://archive.ics.uci.edu/ml/datasets/Census+Income

## Preparation

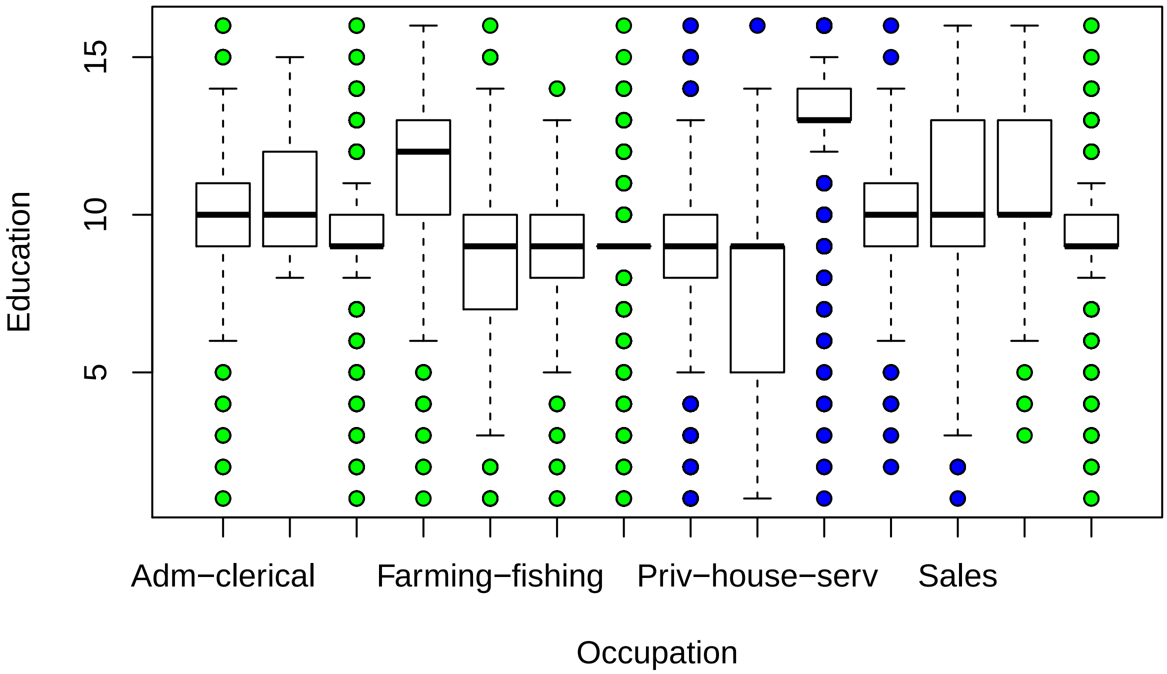
- Recombine test and train data, clean empty lines.
- Quote wrap qualitative data and remove nasty characters with python script.

**Column info**
**age:** continuous.
**workclass:** Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
**fnlwgt:** continuous.
**education:** Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
**education-num:** continuous.
**marital-status:** Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
**occupation:** Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
**relationship:** Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
**race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.**
sex: Female, Male.
**capital-gain:** continuous.
**capital-loss:** continuous.
**hours-per-week:** continuous.
**native-country:** United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

# dont forget to set working directory

```r
# see python script for the few csv modifications
census <- read.table(file = './adult_prepped.csv', header = TRUE, sep = ',')

# sample data to train and test sets
```

```
ct <- sample(nrow(census), nrow(census) * 0.8, replace = FALSE)
ctrain <- census[ct,]
ctest <- census[-ct,]

# Plot Occupation by age while looking for target which is probable income
plot(census$occupation, census$education.num, xlab="Occupation", ylab="Education", pch=21, bg=c('green'
```




Notice the relationship between occupation and probable income

## Logistic regression

```
library(ROCR)
```

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
glm0 <- glm(prob.income~education+hours.per.week+age+workclass+marital.status*relationship, data = ctrai

# probabilities, predictions, and accuracy of new model
probs <- predict.glm(glm0, newdata=ctest, type = "response")
```

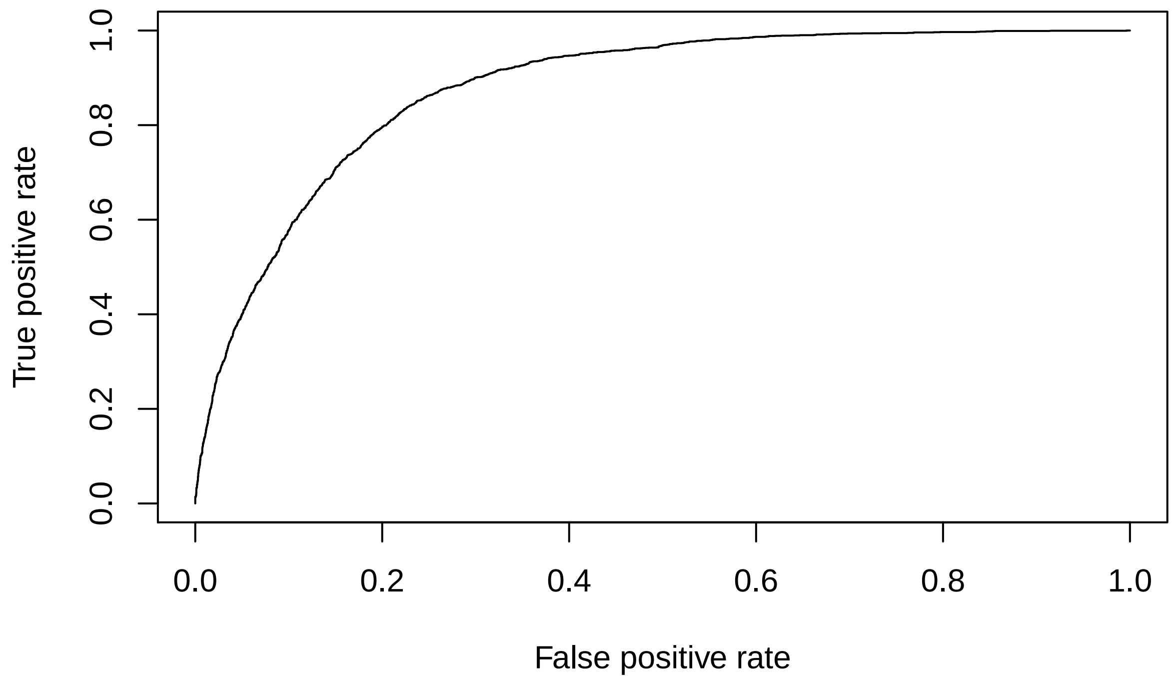
```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
pr <- prediction(probs, ctest$prob.income) # specific to performance
pred <- ifelse(probs>0.5, 2, 1)
```

```r
# prep data for confusion matrix
facprob <- factor(as.integer(ctest$prob.income))
facpred <- factor(pred, levels = 1:2)

# TPR = sensitivity, FPR=specificity
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]

# setup and use confusion matrix
# summary(glm0) very verbose
table(pred, ctest$prob.income)
```

```
##
## pred <=50K >50K
##    1  6391 1113
##    2   543 1133
```

```r
plot(prf)
```




```r
auc
```

```
## [1] 0.8771494
```

## Naive Bayes

```r
library(e1071)
nb0 <- naiveBayes(prob.income~.-capital.gain-capital.loss, data = ctrain)
summary(nb0)
```

```
##         Length Class  Mode
## apriori  2     table  numeric
## tables  13     -none- list
```

```
## levels     2     -none- character
## isnumeric 13     -none- logical
## call       4     -none- call
```

```
# create predictions from NB model
#raw <- predict(nb0, newdata=ctest, type="raw")
pred2 <- predict(nb0, newdata=ctest, type="class")

# print classifier statistics on NB model
library(caret)        # grab mlbench
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

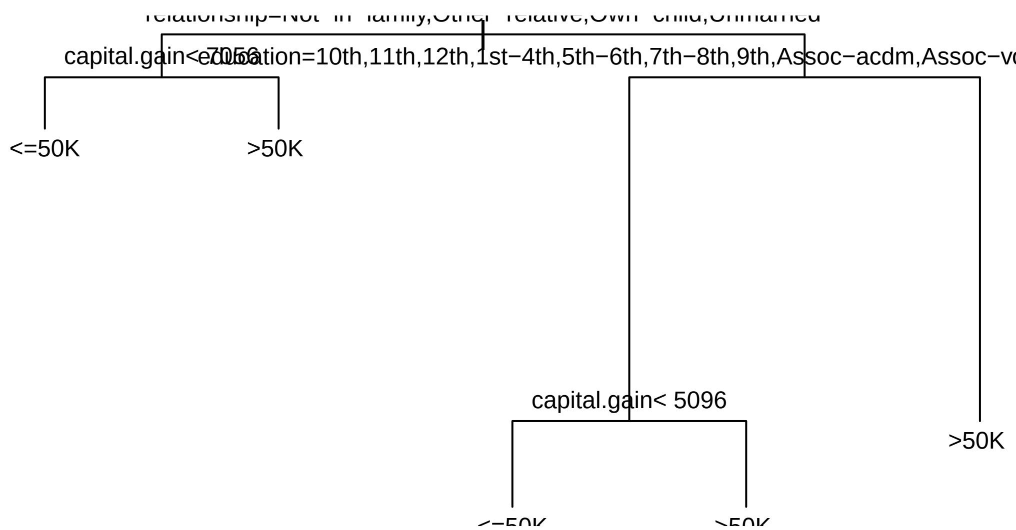
```
facpreds <- factor(as.integer(pred2), levels = 1:2)
facpreds[is.na(facpreds)] <- 2
factarg <- factor(as.integer(ctest$prob.income), levels = 1:2)
factarg[is.na(factarg)] <- 2

# confusion matrix for all the things
confusionMatrix(facpreds, factarg, positive = '2')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2
##          1 6262  592
##          2 1207 1708
##
##                Accuracy : 0.8158
##                  95% CI : (0.808, 0.8235)
##     No Information Rate : 0.7646
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5318
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7426
##             Specificity : 0.8384
##          Pos Pred Value : 0.5859
##          Neg Pred Value : 0.9136
##              Prevalence : 0.2354
##          Detection Rate : 0.1748
##    Detection Prevalence : 0.2984
##       Balanced Accuracy : 0.7905
##
##        'Positive' Class : 2
##
```

# Decision Tree

```
library(rpart)
tree_cen <- rpart(prob.income~., data=census, method = 'class')
```

```r
plot(tree_cen)
text(tree_cen, cex=0.75, pretty=1)
```




```r
#summary(tree_cen)

tree_pruned <- prune.rpart(tree_cen, cp = 0.7)
# plot(tree_pruned)
# text(tree_pruned, cex=0.75, pretty=1)
summary(tree_pruned)
```

```
## Call:
## rpart(formula = prob.income ~ ., data = census, method = "class")
##   n= 48842
##
##    CP nsplit rel error xerror        xstd
## 1 0.7      0         1      1 0.008067898
##
## Node number 1: 48842 observations
##   predicted class=<=50K  expected loss=0.2392818  P(node) =1
##     class counts: 37155 11687
##    probabilities: 0.761 0.239
```

```r
pred_cen <- predict(tree_cen, newdata=ctest, type="class")
pred_pruned <- predict(tree_pruned, newdata=ctest, type="class")

print("First tree")
```

```
## [1] "First tree"
```

```r
table(pred_cen, ctest$prob.income)
```

```
##
## pred_cen <=50K >50K
##    <=50K  7094 1165
##    >50K    375 1135
```

```r
print(paste("Accuracy: ", mean(pred_cen==ctest$prob.income)))
```

```
## [1] "Accuracy:  0.842358480908998"
```

```r
print("Pruned tree")
```

```
## [1] "Pruned tree"
```

```r
table(pred_pruned, ctest$prob.income)
```

```
##
## pred_pruned <=50K >50K
##       <=50K  7469 2300
##       >50K      0    0
```

```r
print(paste("Accuracy: ", mean(pred_pruned==ctest$prob.income)))
```

```
## [1] "Accuracy:  0.76456136759136"
```

**Help: first condition is relationship**

# RESULTS

## Algorithms ranked

1. Logistic Regression - Accuracy:0.8723435

- Predictors tweaked for accuracy first.
- Summary emphasized the predictors that went on to make better models.
- Ended up producing the most accurate model.

2. Decision Tree - Accuracy:0.842563210154571

- Simplest implementation worked best for this algorithm
- Reemphasized the importance of predictors in logit summary
- More positives and less false negative than logit
- Pruning didnt help the fit at all and made it more inaccurate

3. Naive Bayes - Accuracy:0.8161

- More time consuming to implement (factoring model statistics warranted data replacement)
- Alot more True negatives while suffering every other instance in the table
- Worked better with more predictors
- Maybe it was just be but it was very tempremental about what it would allow for a formula

## Analysis

Its interesting to see how much of an impact relationships make on probable income, as well as how unnecessary capital gain and lose are for creating an accurate model. Education and occupation made the biggest impact and were most relevent to each model, so boost those two things in life and you could make more money.