Classification

By Spencer Gray & Michael Stinnett

2022-09-25

Data Source: Adults (https://archive.ics.uci.edu/ml/datasets/Adult)

Linear models for classification work to try and prediction what “class” or “group” an observation falls into. This is a qualitative feature of the observation in question. Generally the linear models do this by making a classification decision based on a linear combination of the other given data in the observation.

For my example I am going to do a simple binary classification. I have found a nice set of census data with a target group being if the adult person makes more than 50k or less than or equal to 50k USD a year. In general, a classification model does not have to be binary and could have as many groups as it want.

In general, the main strength of classification is its own weakness. If there are discernible groups, then classification is great. If there is not, then the algorithms will plot an observation into a camp even it doesn’t fit.

I have combined both the test and training data given from UCI into one csv to get a better 80/20 split

Load the Data

```
df <- read.csv("adult.csv")
```

I am only interested in a few metrics of the census data. Namely age, education level with higher number being more educated, race, sex, hours worked per week, and the target income. I broke the data down this way to make it simpler for data cleaning and demonstration purposes.

Clean the data

```
df <- df[,c(1,5,9,10,13,15)]
df$race <- factor(df$race)
df$sex <- factor(df$sex)
df$Income <- factor(df$Income)
```

Create the 80/20 split

Divide into train and test

```
set.seed(1234)
i <- sample(1:nrow(df), 0.80*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Data Exploration

A few data basic r data exploration functions on the training data set

```
names(train)
```

```
## [1] "age"            "education.num"  "race"           "sex"
## [5] "hours.per.week" "Income"
```

```
dim(train)
```

```
## [1] 39073     6
```

```
summary(train)
```

```
##       age        education.num                   race           sex
##  Min.   :17.00   Min.   : 1.00    Amer-Indian-Eskimo:  375    Female:12859
##  1st Qu.:28.00   1st Qu.: 9.00    Asian-Pac-Islander: 1227    Male  :26214
##  Median :37.00   Median :10.00    Black             : 3766
##  Mean   :38.63   Mean   :10.08    Other             :  331
##  3rd Qu.:48.00   3rd Qu.:12.00    White             :33374
##  Max.   :90.00   Max.   :16.00
##  hours.per.week     Income
##  Min.   : 1.00    <=50K:29686
##  1st Qu.:40.00    >50K : 9387
##  Median :40.00
##  Mean   :40.41
##  3rd Qu.:45.00
##  Max.   :99.00
```

```
str(train)
```

```
## 'data.frame':    39073 obs. of  6 variables:
##  $ age           : int  76 34 44 44 50 36 17 26 43 25 ...
##  $ education.num : int  5 11 10 10 14 13 6 13 10 14 ...
##  $ race          : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 5 5 5 5 5 3 5 ...
##  $ sex           : Factor w/ 2 levels " Female"," Male": 2 2 2 2 2 2 2 1 1 2 ...
##  $ hours.per.week: int  40 50 35 40 40 65 20 45 40 20 ...
##  $ Income        : Factor w/ 2 levels " <=50K"," >50K": 1 1 2 1 2 2 1 1 1 1 ...
```

```
head(train)
```

	age	education.num	race	sex	hours.per.week	Income
	<int>	<int>	<fct>	<fct>	<int>	<fct>
40784	76	5	White	Male	40	<=50K
40854	34	11	White	Male	50	<=50K
41964	44	10	White	Male	35	>50K

	age	education.num	race	sex	hours.per.week	Income
	<int>	<int>	<fct>	<fct>	<int>	<fct>
15241	44	10	White	Male	40	<=50K
33702	50	14	White	Male	40	>50K
35716	36	13	White	Male	65	>50K

6 rows

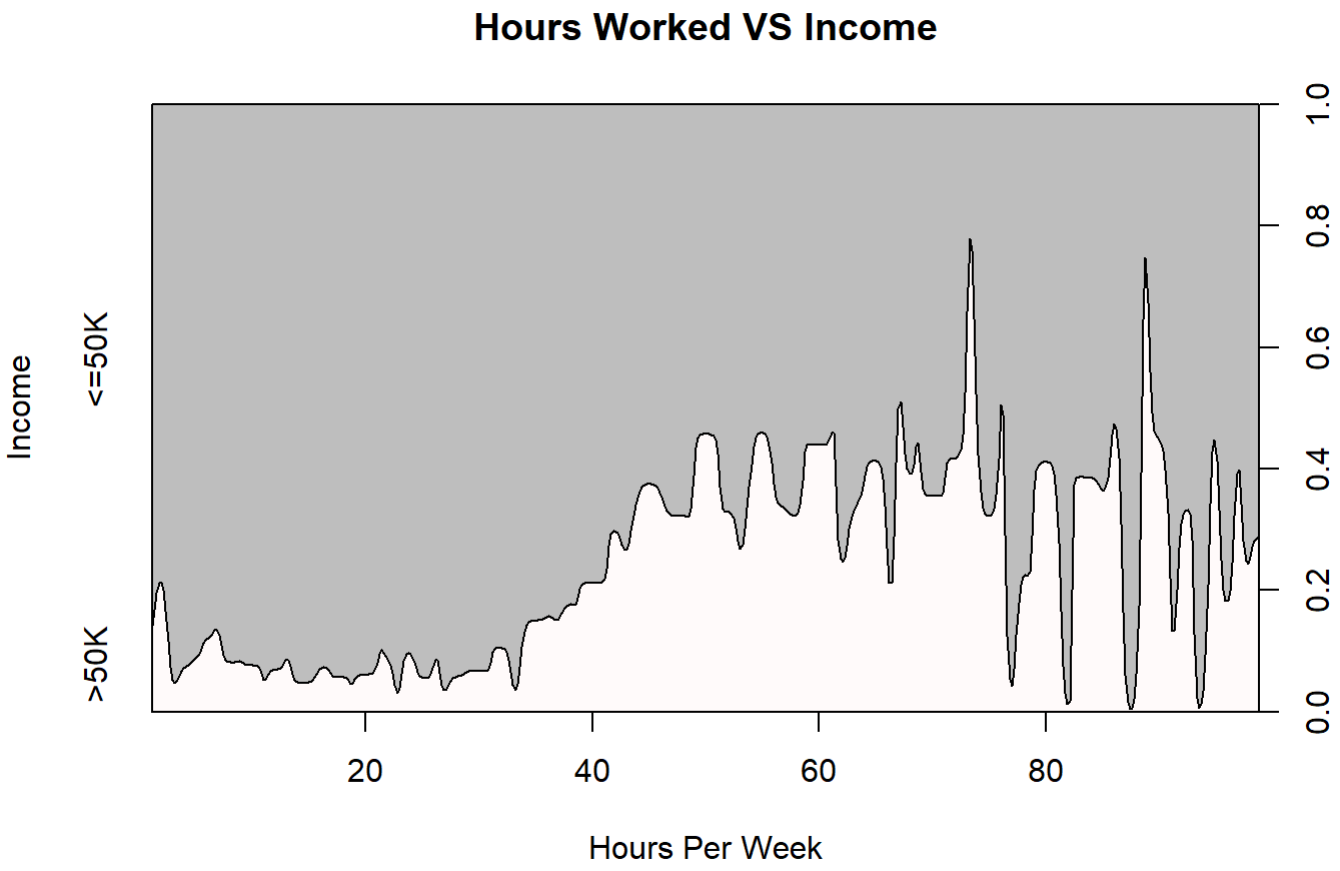
Informative Graphs

It seems 30 to 55ish seems to be the most wealthy.

```
cdplot(train$age, train$Income, col=c("snow", "gray"),
       main="Age Vs Income", xlab = "Age", ylab = "Income")
```

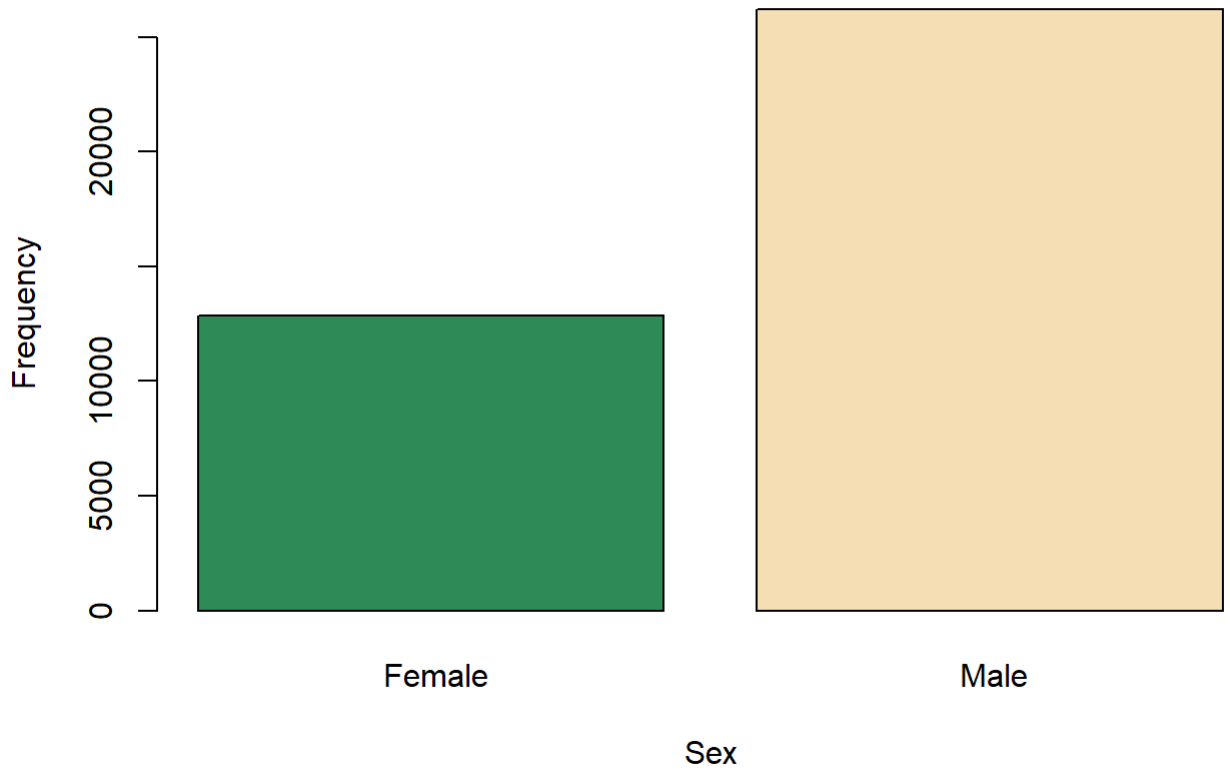
I found this comparison fascinating there is a point where over working on hours does actually always mean wealthier in general. Contrary to “grind” culture in america.

```
cdplot(train$hours.per.week, train$Income, col=c("snow", "gray"),
       main="Hours Worked VS Income", xlab="Hours Per Week", ylab = "Income")
```



More men were surveyed in this data by far. Meaning there will be bias toward men, making demonstrating the wage gap a little harder.

```
barplot(table(train$sex), xlab="Sex", ylab="Frequency",
col=c("seagreen","wheat","sienna3"))
```

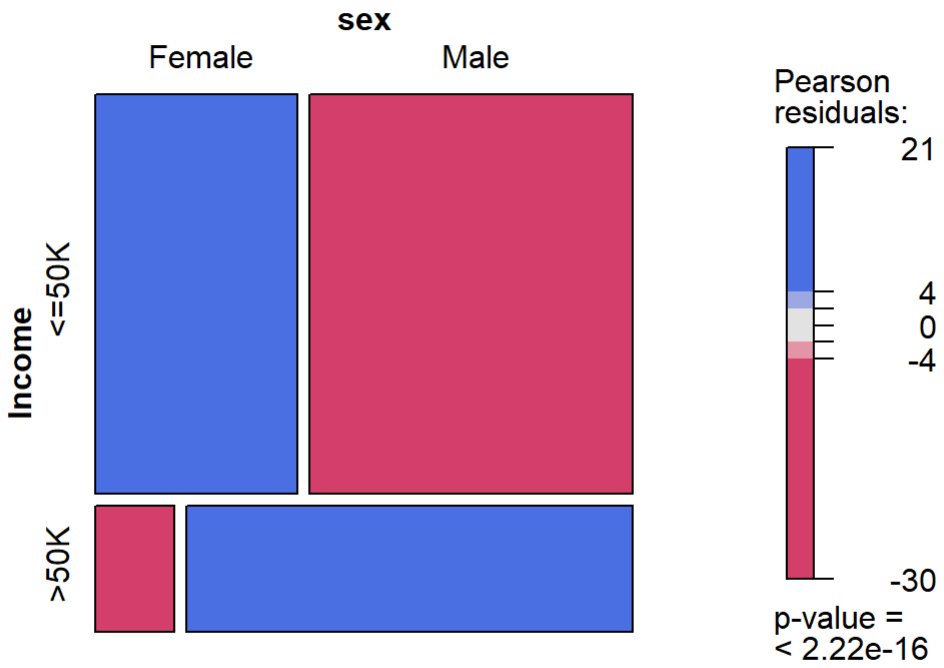


However, it is still apparent that more men end up wealthier than women.

```
library(vcd)
```

```
## Loading required package: grid
```

```
mosaic(table(train[,c(6,4)]), shade=TRUE, legend=TRUE)
```



Build a logistic regression model

```
glm1 <- glm(Income~., data=train, family="binomial")
summary(glm1)
```

```
##
## Call:
## glm(formula = Income ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0382  -0.6701  -0.4079  -0.1119   3.2163
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -9.546238   0.198233 -48.157  < 2e-16 ***
## age                      0.044721   0.001082  41.340  < 2e-16 ***
## education.num            0.354889   0.006100  58.183  < 2e-16 ***
## race Asian-Pac-Islander  0.466014   0.185392   2.514  0.01195 *
## race Black               0.064270   0.178292   0.360  0.71849
## race Other              -0.042410   0.262920  -0.161  0.87185
## race White               0.522025   0.170418   3.063  0.00219 **
## sex Male                 1.127186   0.034499  32.673  < 2e-16 ***
## hours.per.week           0.035444   0.001176  30.135  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43087  on 39072  degrees of freedom
## Residual deviance: 33408  on 39064  degrees of freedom
## AIC: 33426
##
## Number of Fisher Scoring iterations: 5
```

According to our summary here. R thinks that several of the predictors were important. The low p values on age, education, race white, sex male, and hours.per.week signify that. The coefficients are all in log odds here. We have a lower Residual deviance than null deviance. Meaning that our model as a whole is better than just the intercept. The AIC would be used if we made several logisitic models to compare them to each other. The lower the AIC the better.

Build a naive Bayes model

```
library(e1071)
nb1 <- naiveBayes(Income~., data=train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     <=50K      >50K
## 0.7597574 0.2402426
##
## Conditional probabilities:
##         age
## Y            [,1]     [,2]
##    <=50K 36.86260 14.07342
##    >50K  44.23426 10.55619
##
##         education.num
## Y             [,1]     [,2]
##    <=50K  9.596679 2.438159
##    >50K  11.613082 2.377596
##
##         race
## Y         Amer-Indian-Eskimo  Asian-Pac-Islander       Black       Other
##    <=50K         0.011082665         0.029845719 0.111466685 0.009971030
##    >50K          0.004900394         0.036326835 0.048684351 0.003728561
##         race
## Y              White
##    <=50K 0.837633902
##    >50K  0.906359859
##
##         sex
## Y           Female      Male
##    <=50K 0.3853668 0.6146332
##    >50K  0.1511665 0.8488335
##
##         hours.per.week
## Y            [,1]     [,2]
##    <=50K 38.80425 12.30955
##    >50K  45.50826 11.13540
```

For Naive Bayes we see each predictors conditional probability of being a person making more than >50k or <=50k. For the factor predictors we get each level's conditional probability. For the integer predictors we get the mean and standard deviation for each class. I think it is important to note that male has a higher conditional probability to make more than 50k. White and Asian-Pac-Islander also have that quality. On average, the higher the education level the more likely you are to make more than 50k. On average, the age of 44 is when a person makes more than 50k a year.

Test on logistic regression

```
logProbs <- predict(glm1, newdata=test, type="response")
logPred <- ifelse(logProbs>0.5, 2, 1)
```

```
library(caret)
```

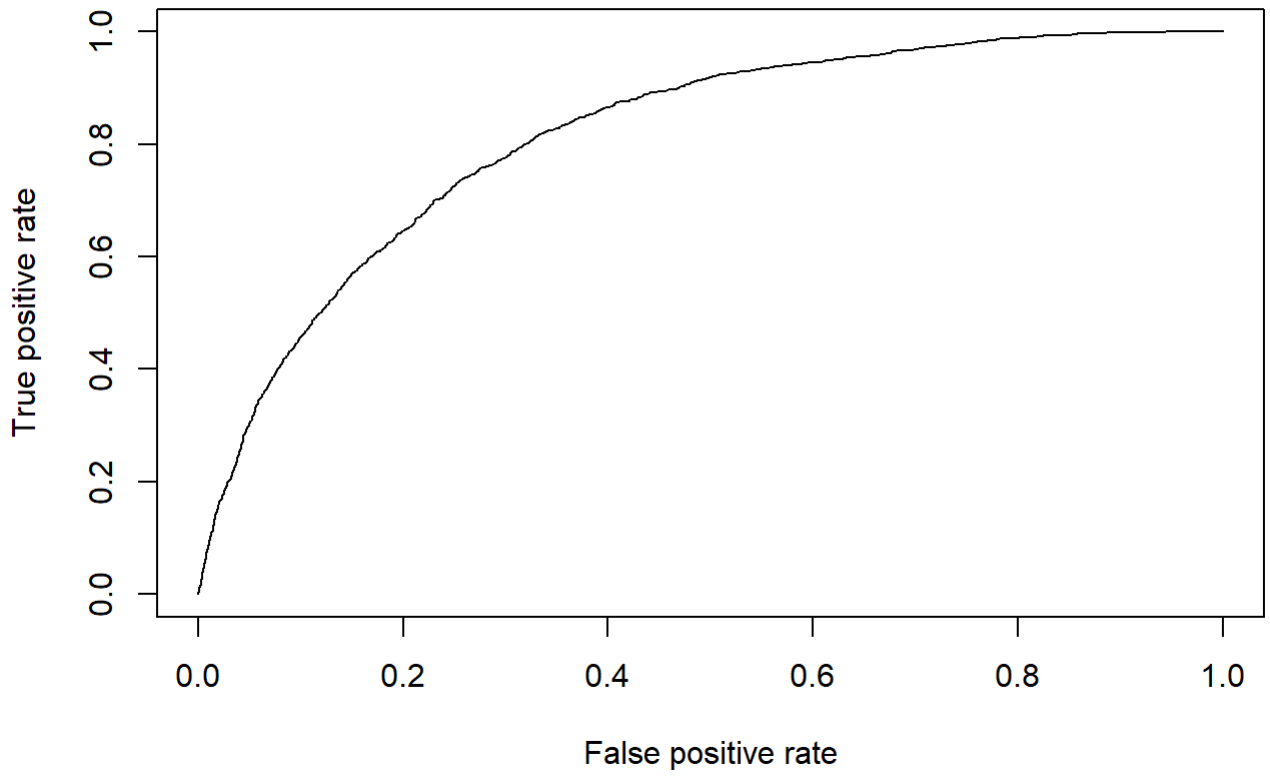
```
## Loading required package: ggplot2
```

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

```
confusionMatrix(as.factor(logPred), reference=as.factor(as.integer(test$Income)))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2
##          1 6946 1436
##          2  523  864
##
##                Accuracy : 0.7995
##                  95% CI : (0.7914, 0.8074)
##     No Information Rate : 0.7646
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3543
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9300
##             Specificity : 0.3757
##          Pos Pred Value : 0.8287
##          Neg Pred Value : 0.6229
##              Prevalence : 0.7646
##          Detection Rate : 0.7110
##    Detection Prevalence : 0.8580
##       Balanced Accuracy : 0.6528
##
##        'Positive' Class : 1
##
```

```
library(ROCR)
logPr <- prediction(logProbs, test$Income)
logPrf <- performance(logPr, measure = "tpr", x.measure = "fpr")
plot(logPrf)
```




```
logAuc <- performance(logPr, measure = "auc")
logAuc <- logAuc@y.values[[1]]
logAuc
```

```
## [1] 0.8145207
```

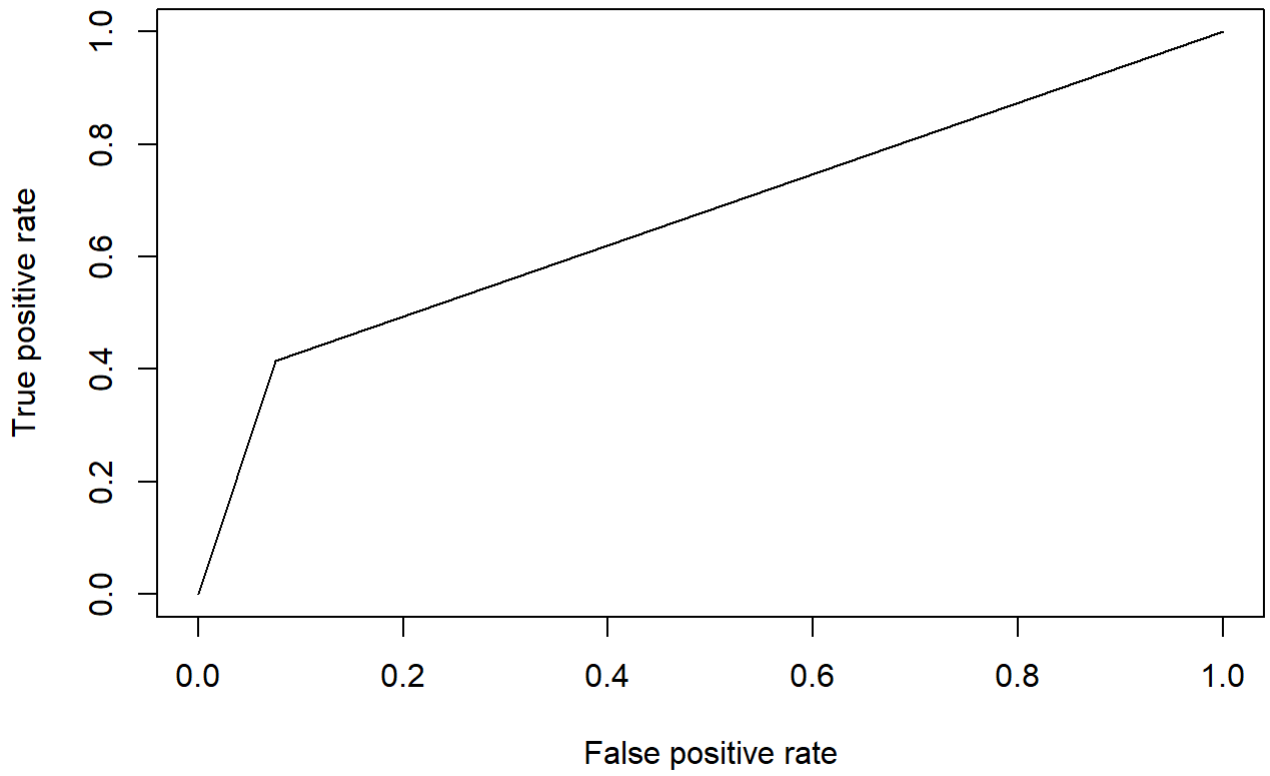
Test on naive bayes

```
bayesPred <- predict(nb1, newdata=test, type="class")
```

```
library(caret)
confusionMatrix(as.factor(as.integer(bayesPred)), reference=as.factor(as.integer(test$Income)))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2
##          1 6906 1344
##          2  563  956
##
##                Accuracy : 0.8048
##                  95% CI : (0.7968, 0.8126)
##     No Information Rate : 0.7646
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3856
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9246
##             Specificity : 0.4157
##          Pos Pred Value : 0.8371
##          Neg Pred Value : 0.6294
##              Prevalence : 0.7646
##          Detection Rate : 0.7069
##    Detection Prevalence : 0.8445
##       Balanced Accuracy : 0.6701
##
##        'Positive' Class : 1
##
```

```
library(ROCR)
bayesPr <- prediction(as.integer(bayesPred), as.integer(test$Income))
bayesPrf <- performance(bayesPr, measure = "tpr", x.measure = "fpr")
plot(bayesPrf)
```




```
bayesAuc <- performance(bayesPr, measure = "auc")
bayesAuc <- bayesAuc@y.values[[1]]
bayesAuc
```

```
## [1] 0.670137
```

Comparing the metrics

The most prevalent metric here is that accuracy is slightly higher on the Naive Bayes than the Logistic Regression. Naive Bayes has slightly lower sensitivity and and a marginally higher specificity. Meaning it predicted more true negatives and and a little less true positives.

Interestingly, Naive Bayes has a very sharp trade off of false positives to true positives. Whereas, Logistic regression has a nice curve. This is shown of by the ROC.

I think Naive Bayes preformed better in this case because the data set wasn't super huge. Also, naive Bayes works well when the data is biased, and as we demonstrated comparing male and female participants, this data set is biased.

Strengths and Weaknesses of Naive bayes and Linear Regression

Naive Bayes

Strengths:
-works well on small data
-easy to implement
-easy to interpret
-handles high dimensions well

Weaknesses:
-May be out preformed by other classifiers on larger data sets
-Guesses are made up for values not in the training set
-Nonindependent predictors inhibit the performance

Linear Regression

Strengths:
-Separates linearly separable classes
-Is computationally inexpensive
-Has good probabilities to work with

Weakness:
-Prone to underfitting. If there are more complicated decision boundaries, it will likely not find it unless given really good training data.

Benefits and draw backs of each metric used

The confusion matrix is used to show off the amount of correct and incorrect predictions. From this matrix the accuracy, sensitivity, specificity are measured. These help quantify the extent to which each class was classified correctly. However, this does not account for a correct prediction by pure chance.

Cohen’s kappa attempts to adjust for the correct prediction by pure chance. The main draw back here is the results of cohen’s kappa are not always agreed upon.

ROC and AUC visualize the performance of the algorithm. These show off the give and take of true positive rates to false positive rates. The ROC is the line that shows this and the area under that line is the AUC. These only show a limited give and take.