ML 4375 Project 2 - Classification

Supratik Pochampally

Abstract

For my classification dataset, I chose the Census Income dataset from the UCI Machine Learning Repository. The purpose of the dataset is to use attributes such as age, education level, and others to predict if a person's income exceeds $50k/yr, with attributes collected from census data.

Dataset

Let's start by reading in the dataset and printing the number of rows and column names:

```
# Read in the .csv file of the data set
df <- read.csv("CensusIncome.csv", header = TRUE)
# Print number of rows
print(paste("Number of rows:", nrow(df)))
```

```
## [1] "Number of rows: 32561"
```

```
# Print attributes
names(df)
```

```
##  [1] "age"            "workclass"      "fnlwgt"         "education"
##  [5] "education.num"  "marital.status" "occupation"     "relationship"
##  [9] "race"           "sex"            "capital.gain"   "capital.loss"
## [13] "hours.per.week" "native.country" "income"
```

We see that the data set is large, with 32,561 rows and many columns, some being very ambiguously named.

Data cleaning

Link to source: https://archive.ics.uci.edu/ml/datasets/Adult

Because the dataset is messy and needs cleaning, we will identify what predictors we want to use to predict our target column, being the "income" column.

The "fnlwgt" attribute has no documentation in the description of the dataset, so we want to remove it in case it is a non-predictive column that messes with our models in the future. We also want to deal with the "native.country" column, since there are many different countries a person can be from. Because the United States is likely the most common country, let's see how many instances in the data-frame have the native-country of "United-States":

```
# Find number of instances of "United States"
numUS <- length(which(df$native.country == " United-States"))
print(paste("Number of instances with United States as native.country:", numUS))
```

```
## [1] "Number of instances with United States as native.country: 29170"
```

Because this is a large majority of our dataset, making it very imbalanced, we can remove this column altogether.Also, the "capital.loss" and "capital.gain" columns do not have description, and are both very ambiguous, which we also want to remove. Furthermore, the "marital.status" and "relationship" columns represent similar things, so we can remove "relationship" altogether. Lastly, the "education" and "education.num" column have the same data, but the "education.num" column assigns numerical values to the education level. Thus, we will remove the "education" column.

```
df <- df[-c(3, 4, 8, 11, 12, 14)]
names(df)
```

```
## [1] "age"            "workclass"      "education.num"  "marital.status"
## [5] "occupation"     "race"           "sex"            "hours.per.week"
## [9] "income"
```

Let's also check if there are any NA or NaN values in the columns of the dataset:

```
colSums(is.na(df))
```

```
##            age      workclass  education.num marital.status     occupation
##              0              0              0              0              0
##           race            sex hours.per.week         income
##              0              0              0              0
```

Luckily we have no NA or NaN values, so we can proceed without having to replace anything with the mean of the column.

Lastly, let's factorize some of our discrete, categorical predictors that are currently characters.

```
df$workclass <- as.factor(df$workclass)
df$marital.status <- as.factor(df$marital.status)
df$occupation <- as.factor(df$occupation)
df$race <- as.factor(df$race)
df$sex <- as.factor(df$sex)
df$income <- as.factor(df$income)
```

Data exploration

R functions

Let's use some R functions for data exploration.

```
# Print the first 6 rows
head(df)
```

```
##   age         workclass education.num      marital.status         occupation
## 1  39         State-gov            13       Never-married       Adm-clerical
## 2  50  Self-emp-not-inc            13  Married-civ-spouse    Exec-managerial
## 3  38           Private             9            Divorced  Handlers-cleaners
## 4  53           Private             7  Married-civ-spouse  Handlers-cleaners
## 5  28           Private            13  Married-civ-spouse     Prof-specialty
## 6  37           Private            14  Married-civ-spouse    Exec-managerial
##     race     sex hours.per.week income
## 1  White    Male             40  <=50K
## 2  White    Male             13  <=50K
## 3  White    Male             40  <=50K
## 4  Black    Male             40  <=50K
## 5  Black  Female             40  <=50K
## 6  White  Female             40  <=50K
```

```
# Display the internal structure of the data frame
str(df)
```

```
## 'data.frame':    32561 obs. of  9 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass     : Factor w/ 9 levels " ?"," Federal-gov",..: 8 7 5 5 5 5 5 7 5 5 ...
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital.status: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
##  $ occupation    : Factor w/ 15 levels " ?"," Adm-clerical",..: 2 5 7 7 11 5 9 5 11 5 ...
##  $ race          : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
##  $ sex           : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
##  $ hours.per.week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ income        : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
```

```
# Print summary of dataset
summary(df)
```

```
##       age                     workclass     education.num
##  Min.   :17.00    Private         :22696   Min.   : 1.00
##  1st Qu.:28.00    Self-emp-not-inc: 2541   1st Qu.: 9.00
##  Median :37.00    Local-gov       : 2093   Median :10.00
##  Mean   :38.58    ?               : 1836   Mean   :10.08
##  3rd Qu.:48.00    State-gov       : 1298   3rd Qu.:12.00
##  Max.   :90.00    Self-emp-inc    : 1116   Max.   :16.00
##                  (Other)          :  981
##                 marital.status             occupation
##   Divorced              : 4443    Prof-specialty :4140
##   Married-AF-spouse     :   23    Craft-repair   :4099
##   Married-civ-spouse    :14976    Exec-managerial:4066
##   Married-spouse-absent:  418    Adm-clerical   :3770
##   Never-married         :10683    Sales          :3650
##   Separated             : 1025    Other-service  :3295
##   Widowed               :  993   (Other)         :9541
##                   race            sex        hours.per.week     income
##   Amer-Indian-Eskimo:  311    Female:10771   Min.   : 1.00    <=50K:24720
##   Asian-Pac-Islander: 1039    Male  :21790   1st Qu.:40.00    >50K : 7841
##   Black             : 3124                   Median :40.00
##   Other             :  271                   Mean   :40.44
```

```
##   White             :27816                   3rd Qu.:45.00
##                                              Max.   :99.00
##
```

```
# Print the average age
print(paste("Average age:", mean(df$age)))
```

```
## [1] "Average age: 38.5816467553208"
```

```
# Print the average education level
print(paste("Average education level:", mean(df$education.num)))
```

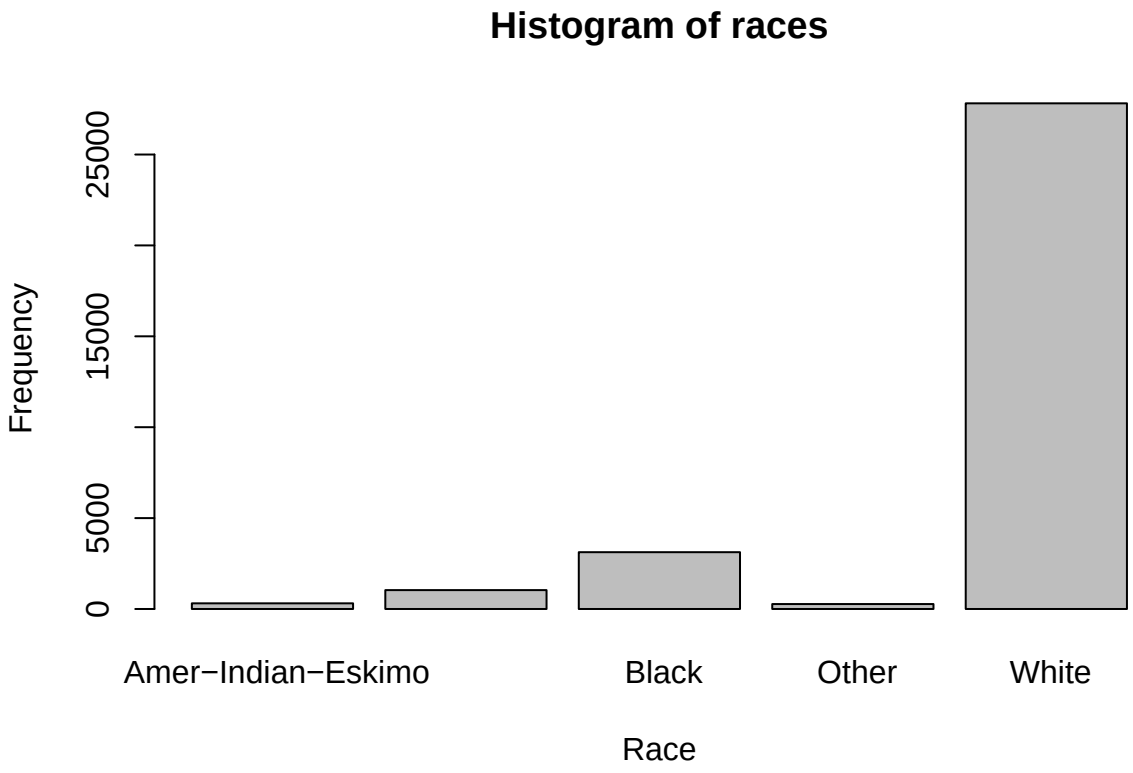
```
## [1] "Average education level: 10.0806793403151"
```

Although the education.num average doesn't mean much by itself, we know from the description of the data set that 10.081 is approximately some-college education.

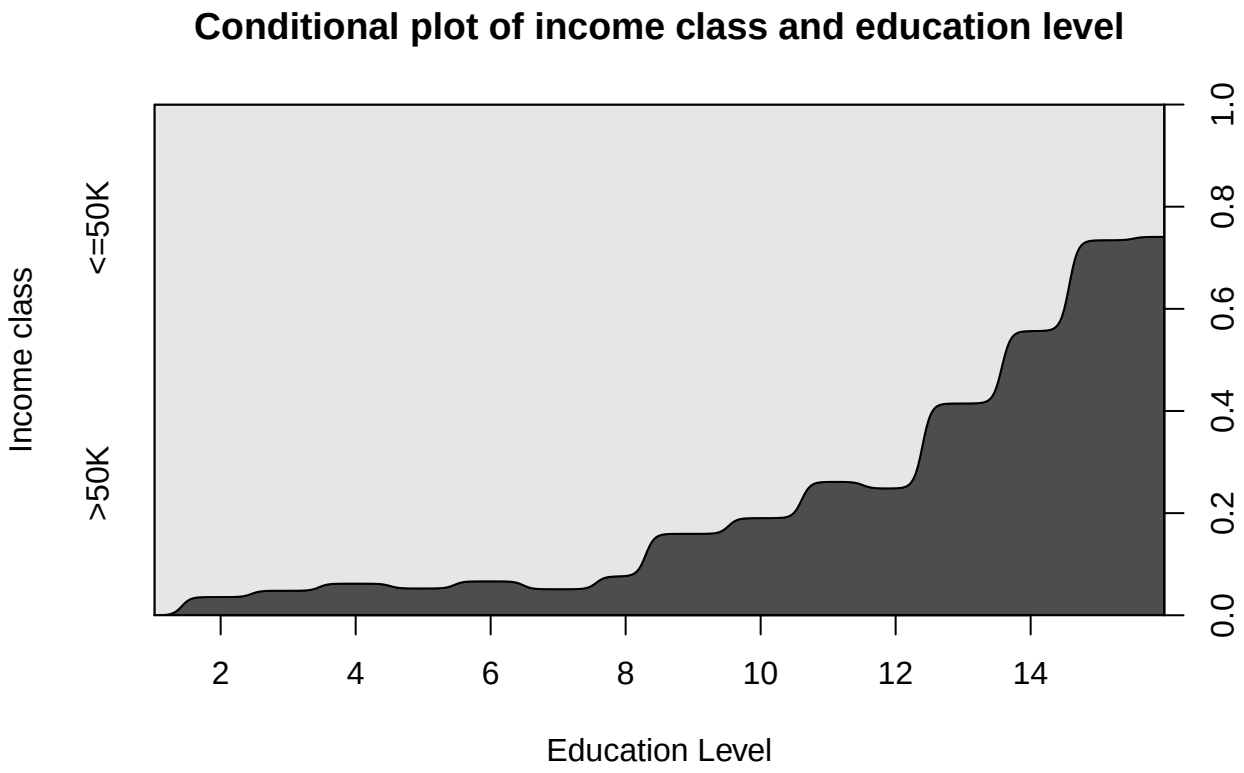
R graphs

Now let's create some informative R graphs for data exploration.

```
# Histogram of race
barplot(table(df$race), main = "Histogram of races", xlab = "Race", ylab = "Frequency")
```



```
# Conditional plot of education level and income
cdplot(df$education.num, df$income, main = "Conditional plot of income class and education level", xlab
```



Based on the graphs, we can see a much higher frequency of the the rows are White, followed by Black. This imbalance may prove to be an issue when building the models. Furthermore, as I thought, there is a clear correlation between education level and income.

ML algorithms

We will attempt to run 3 classification algorithms over this dataset - Logistic Regression, Naive Bayes, and Decision Tree classification. Before running each algorithm, let's discuss feature selection.

Feature selection

The features in the dataset that are used to determine whether or not a person has an income above or below $50K are age, workclass, education.num, marital.status, occupation, race, sex, and hours.per.week. The reasons for eliminating other attributes was discussed in the data cleaning section above. We will now be discussing why these features were selected. Logically, hours worked per week, occupation, and workclass would likely have some correlation with income. These attributes deal directly with the amount someone works as well as their specialization of work, such which involves the occupation they work in as well as the specific workclass that occupation falls under. Education level similarly deals with how educated a person is, which would determine how skilled they are. Skilled labor obviously entails a higher income than unskilled labor, so this would also likely be an significant feature. Age somewhat coincides with the same reasons as

education level, as work experience will come with age and will usually increase job prospects and therefore income. Marital status is important in many countries, as you are usually taxed based on your marital status. Lastly, race and sex can help us explore the softer side of some factors that can affect income. This involves gender and racial inequality with things such as wage gaps between people.

We can now start implementing our classification algorithms. Let's begin by splitting our data into train and test sets of 75% and 25% respectively.

```
# Set seed to ensure the same split of training and testing sets
set.seed(1234)
# Split the data
i <- sample(1:nrow(df), nrow(df) * 0.75, replace = FALSE)
train <- df[i, ]
test <- df[-i, ]
```

Code to run logistic regression

Now we can run logistic regression over our predictors.

```
# Run logistic regression
glm1 <- glm(income~., data = train, family = "binomial")
# Print summary of model
summary(glm1)
```

```
##
## Call:
## glm(formula = income ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7532  -0.5528  -0.2348  -0.0541   3.1731
##
## Coefficients: (1 not defined because of singularities)
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -9.457346   0.314669 -30.055  < 2e-16 ***
## age                                0.029958   0.001740  17.220  < 2e-16 ***
## workclass Federal-gov              1.077993   0.166447   6.476 9.39e-11 ***
## workclass Local-gov                0.453074   0.151695   2.987  0.00282 **
## workclass Never-worked            -9.400845 182.267610  -0.052  0.95887
## workclass Private                  0.626070   0.135449   4.622 3.80e-06 ***
## workclass Self-emp-inc             0.890491   0.161521   5.513 3.52e-08 ***
## workclass Self-emp-not-inc         0.158024   0.148603   1.063  0.28761
## workclass State-gov                0.283525   0.164594   1.723  0.08497 .
## workclass Without-pay            -10.795833 149.978045  -0.072  0.94262
## education.num                      0.300996   0.009971  30.186  < 2e-16 ***
## marital.status Married-AF-spouse   2.492349   0.515665   4.833 1.34e-06 ***
## marital.status Married-civ-spouse  2.043124   0.069676  29.323  < 2e-16 ***
## marital.status Married-spouse-absent -0.115808   0.233207  -0.497  0.61948
## marital.status Never-married      -0.520874   0.086331  -6.033 1.60e-09 ***
## marital.status Separated          -0.148791   0.171029  -0.870  0.38431
## marital.status Widowed            -0.102381   0.159964  -0.640  0.52216
## occupation Adm-clerical            0.105038   0.107270   0.979  0.32748
## occupation Armed-Forces           -0.733913   1.298322  -0.565  0.57188
```

```
## occupation Craft-repair                0.130142   0.093553   1.391  0.16419
## occupation Exec-managerial             0.863079   0.095704   9.018  < 2e-16 ***
## occupation Farming-fishing            -1.080921   0.161635  -6.687 2.27e-11 ***
## occupation Handlers-cleaners          -0.766117   0.168021  -4.560 5.12e-06 ***
## occupation Machine-op-inspct          -0.199659   0.116469  -1.714  0.08648 .
## occupation Other-service              -0.847058   0.136311  -6.214 5.16e-10 ***
## occupation Priv-house-serv            -2.702046   1.189139  -2.272  0.02307 *
## occupation Prof-specialty              0.626076   0.101615   6.161 7.22e-10 ***
## occupation Protective-serv             0.617408   0.143888   4.291 1.78e-05 ***
## occupation Sales                       0.319656   0.098542   3.244  0.00118 **
## occupation Tech-support                0.655462   0.131878   4.970 6.69e-07 ***
## occupation Transport-moving                  NA         NA      NA       NA
## race Asian-Pac-Islander                0.395880   0.267497   1.480  0.13889
## race Black                             0.454757   0.255765   1.778  0.07540 .
## race Other                             0.041110   0.366858   0.112  0.91078
## race White                             0.508402   0.245404   2.072  0.03829 *
## sex Male                               0.125662   0.056023   2.243  0.02489 *
## hours.per.week                         0.030948   0.001763  17.549  < 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: 26878  on 24419  degrees of freedom
## Residual deviance: 17298  on 24384  degrees of freedom
## AIC: 17370
##
## Number of Fisher Scoring iterations: 12
```

Looking at the summary of the model, we can conclude that age, workclass, education level, occupation, and hours per week are very significantly correlated to income. martial.status is slightly correlated, especially between married and never married, and race and sex are not significant.

Now let's test the model over the testing set.

```
# Assign probabilities and predictions based on the model
probs1 <- predict(glm1, newdata = test, type = "response")
```

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

```
pred1 <- ifelse(probs1 > 0.5, ">50K", "<=50K")
# Print the confusion matrix of the predictions vs. the test set
table(pred1, test$income)
```

```
##
## pred1    <=50K  >50K
##   <=50K   5691   917
##   >50K     454  1079
```

Looking at the table, we see that we got 5691 true positives, 1079 true negatives, 917 false positives, and 454 false negatives.

Logistic regression metrics

These predictions compute to be the following metrics:

```
# Calculate and print the accuracy
acc1 <- (5691 + 1079) / (5691 + 1079 + 917 + 454)
print(paste("Accuracy:", acc1))
```

```
## [1] "Accuracy: 0.83159317037219"
```

```
# Calculate and print the sensitivity
sensitivity1 <- (5691) / (5691 + 454)
print(paste("Sensitivity:", sensitivity1))
```

```
## [1] "Sensitivity: 0.926118795768918"
```

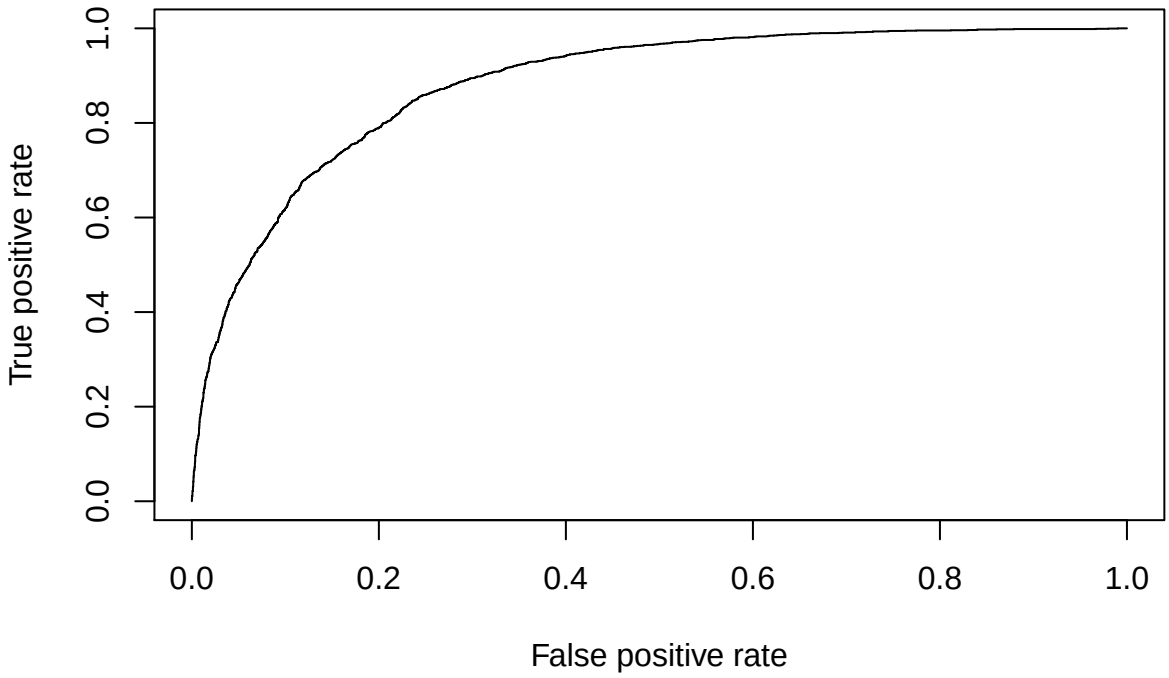
```
# Calculate and print the specificity
specificity1 <- (1079) / (1079 + 917)
print(paste("Specificity:", specificity1))
```

```
## [1] "Specificity: 0.540581162324649"
```

We see that accuracy and sensitivity are good, but the specificity is lacking. This means we are overestimating incomes of certain people.

We can also use an ROC curve and AUC value to observe the performance.

```
# Plot the ROC curve
library(ROCR)
pr1 <- prediction(probs1, test$income)
prf1 <- performance(pr1, measure = "tpr", x.measure = "fpr")
plot(prf1)
```




```
# Calculate and print the AUC value
auc1 <- performance(pr1, measure = "auc")
auc1 <- auc1@y.values[[1]]
print(paste("AUC:", auc1))
```

```
## [1] "AUC: 0.882468476415785"
```

We see that the ROC curve shoots upwards and to the right with a little space left on the top left. We also have a fairly high AUC value, which means our predictive value is decent.

Code for Naive Bayes

Next we will try Naive Bayes using the same training and testing set from before so we get the most accurate comparison.

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.4
```

```
nb1 <- naiveBayes(income~., data = train)
summary(nb1)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
## tables    8      -none- list
## levels    2      -none- character
## isnumeric 8      -none- logical
## call      4      -none- call
```

```
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##    <=50K     >50K
## 0.760647 0.239353
##
## Conditional probabilities:
##         age
## Y            [,1]     [,2]
##    <=50K 36.80861 14.06201
##    >50K  44.22618 10.43956
##
##         workclass
## Y                    ?  Federal-gov    Local-gov  Never-worked      Private
##    <=50K 0.0676177658 0.0231493943 0.0593808883  0.0003230148 0.7172005384
##    >50K  0.0244653550 0.0484174508 0.0802395210  0.0000000000 0.6337040205
##         workclass
## Y         Self-emp-inc  Self-emp-not-inc    State-gov  Without-pay
##    <=50K  0.0199730821      0.0743472409 0.0375235532 0.0004845222
##    >50K   0.0807527802      0.0881094953 0.0443113772 0.0000000000
##
##         education.num
## Y             [,1]     [,2]
##    <=50K  9.602584 2.454872
##    >50K  11.620701 2.350823
##
##         marital.status
## Y            Divorced  Married-AF-spouse  Married-civ-spouse
##    <=50K 0.1590847914       0.0005921938        0.3361507402
##    >50K  0.0612489307       0.0013686912        0.8544054748
##         marital.status
## Y         Married-spouse-absent  Never-married    Separated      Widowed
##    <=50K           0.0159892328   0.4140511440 0.0372543742 0.0368775236
##    >50K            0.0044482464   0.0602224123 0.0082121471 0.0100940975
##
##         occupation
## Y                    ?  Adm-clerical  Armed-Forces  Craft-repair
##    <=50K 0.0679407806  0.1315208614  0.0003768506  0.1290444145
##    >50K  0.0244653550  0.0663815227  0.0001710864  0.1209580838
##         occupation
```

```
## Y         Exec-managerial  Farming-fishing  Handlers-cleaners
##    <=50K     0.0841453567     0.0351547779       0.0511978466
##    >50K      0.2502994012     0.0124893071       0.0097519247
##         occupation
## Y         Machine-op-inspct  Other-service  Priv-house-serv  Prof-specialty
##    <=50K       0.0708479139   0.1279676985     0.0057065949    0.0929205922
##    >50K        0.0335329341   0.0176218991     0.0001710864    0.2338751069
##         occupation
## Y         Protective-serv        Sales  Tech-support  Transport-moving
##    <=50K     0.0176043069 0.1085868102  0.0254104980      0.0515746972
##    >50K      0.0277159966 0.1252352438  0.0355859709      0.0417450813
##
##         race
## Y         Amer-Indian-Eskimo  Asian-Pac-Islander       Black       Other
##    <=50K         0.011467026         0.030524899 0.110309556 0.009851952
##    >50K          0.004106074         0.035072712 0.052181352 0.003592814
##         race
## Y              White
##    <=50K 0.837846568
##    >50K  0.905047049
##
##         sex
## Y           Female      Male
##    <=50K 0.3849798 0.6150202
##    >50K  0.1527802 0.8472198
##
##         hours.per.week
## Y            [,1]     [,2]
##    <=50K 38.84011 12.29228
##    >50K  45.44756 10.95034
```

From the summary of the classifier and the probabilities themselves, we can conclude that our dataset is overall quite imbalanced, as 76.1% of the observations have an income >=50K while the remaining 23.9% have an income <50K. We see a clear difference in the average level of education between low and high income, with a difference of two education levels between each of the classes. The average age of high income is 44.23, while the average age of low income is 36.81. The probability of high income when your married to a civilian spouse is 85.44%, with a probability of only 0.137% if married to a spouse in the Armed Forces. There is also a significant disparity of probability of high income between male and female sex, with a probability of 84.7% over 15.3%. Lastly, the average hours worked per week of someone with high income is 45.45 hours, while the average hours worked per week of someone with low income is 38.84 hours.

Now let's test the model over the testing set.

```
pred2 <- predict(nb1, newdata = test, type = "class")
table(pred2, test$income)
```

```
##
## pred2     <=50K  >50K
##    <=50K   5362   683
##    >50K     783  1313
```

Looking at the table, we see that we got 5362 true positives, 1313 true negatives, 683 false positives, and 783 false negatives.

Naive Bayes metrics

These predictions compute to be the following metrics:

```
acc2 <- (5362 + 1313) / (5362 + 1313 + 683 + 783)
print(paste("Accuracy:", acc2))
```

```
## [1] "Accuracy: 0.819923842279818"
```

```
# Calculate and print the sensitivity
sensitivity2 <- (5362) / (5362 + 783)
print(paste("Sensitivity:", sensitivity2))
```

```
## [1] "Sensitivity: 0.872579332790887"
```

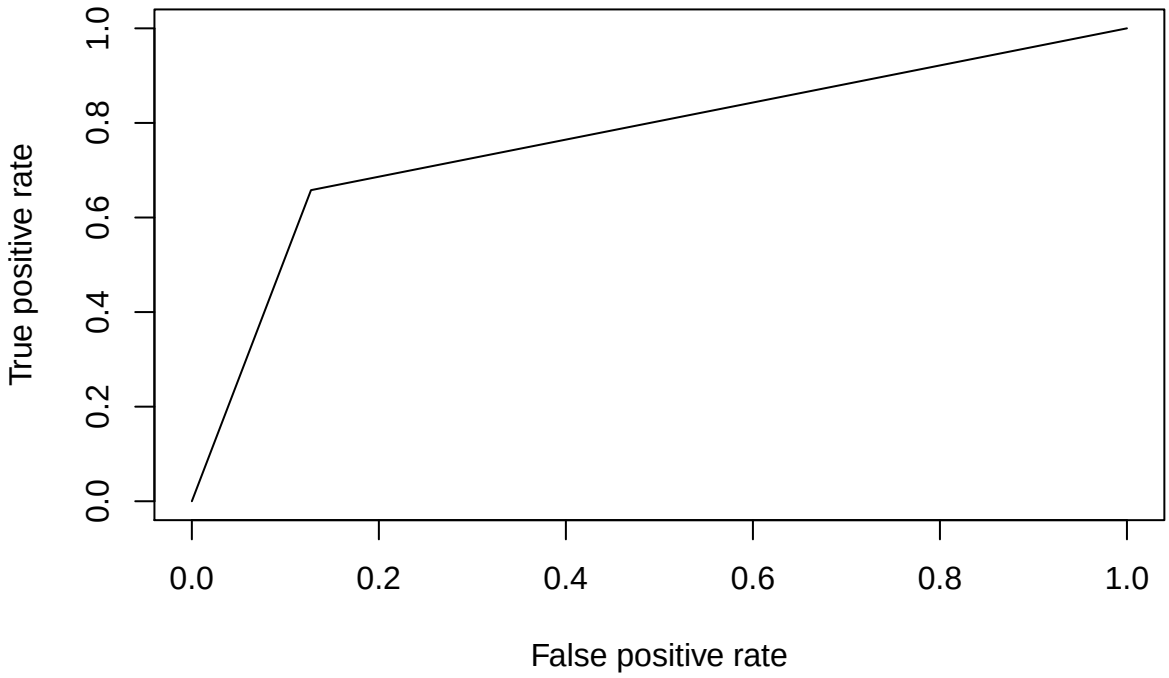
```
# Calculate and print the specificity
specificity2 <- (1313) / (1313 + 683)
print(paste("Specificity:", specificity2))
```

```
## [1] "Specificity: 0.657815631262525"
```

We see that accuracy and sensitivity are once again decent, but this time the specificity is also a little better. This means we are overestimating incomes of certain people slightly less than we did with logistic regression.

We can also use an ROC curve and AUC value to observe the performance.

```
# Plot the ROC curve
predvec1 <- ifelse(as.character(pred2) == " >50K", 1, 0)
realvec1 <- ifelse(as.character(test$income) == " >50K", 1, 0)
pr2 <- prediction(predvec1, realvec1)
prf2 <- performance(pr2, measure = "tpr", x.measure = "fpr")
plot(prf2)
```




```
# Calculate and print the AUC value
auc2 <- performance(pr2, measure = "auc")
auc2 <- auc2@y.values[[1]]
print(paste("AUC:", auc2))
```

```
## [1] "AUC: 0.765197482026706"
```

We see that the ROC curve shoots upwards and to the right in a straighter with slightly more space left on the top left. We also have a fairly lower AUC value compared to that of logistic regression, which means our predictive value is not as good.

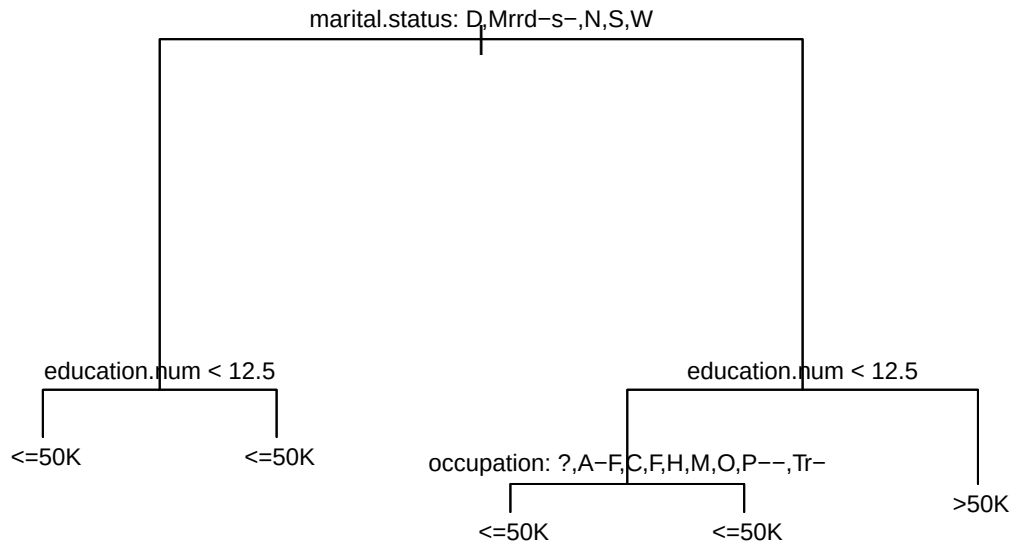
Code for Decision Tree

Lastly, we will try Decision Tree classification once again using the same training and testing set from before so we get the most accurate comparison.

```
# Run decision tree classification
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.0.4
```

```r
tree1 <- tree(income~., data = train)
# Plot decision tree
plot(tree1)
text(tree1, cex = 0.75, pretty = 1)
```



Because our data is fairly complex, the actual decision tree diagram is fairly unclear and hard to read. Instead, we can try interpreting the decision tree outline:

```r
# Print decision tree outline
tree1
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 24420 26880  <=50K ( 0.76065 0.23935 )
##    2) marital.status:  Divorced, Married-spouse-absent, Never-married, Separated, Widowed 13163  626
##      4) education.num < 12.5 10461  2951  <=50K ( 0.96817 0.03183 ) *
##      5) education.num > 12.5 2702  2618  <=50K ( 0.81125 0.18875 ) *
##    3) marital.status:  Married-AF-spouse, Married-civ-spouse 11257 15470  <=50K ( 0.55565 0.44435 )
##      6) education.num < 12.5 7863  9980  <=50K ( 0.66934 0.33066 )
##       12) occupation:  ?, Armed-Forces, Craft-repair, Farming-fishing, Handlers-cleaners, Machine-op
##       13) occupation:  Adm-clerical, Exec-managerial, Prof-specialty, Protective-serv, Sales, Tech-su
##      7) education.num > 12.5 3394  4101  >50K ( 0.29228 0.70772 ) *
```

The output above is slightly unclear, so I copy-pasted it down below:

---

node), split, n, deviance, yval, (yprob) * denotes terminal node

1) root 24420 26880 <=50K (0.76065 0.23935)
  2) marital.status: Divorced, Married-spouse-absent, Never-married, Separated, Widowed 13163 6264 <=50K (0.93596 0.06404)
    4) education.num < 12.5 10461 2951 <=50K (0.96817 0.03183) *
    5) education.num > 12.5 2702 2618 <=50K (0.81125 0.18875) *
  3) marital.status: Married-AF-spouse, Married-civ-spouse 11257 15470 <=50K (0.55565 0.44435)
    6) education.num < 12.5 7863 9980 <=50K (0.66934 0.33066)
      12) occupation: ?, Armed-Forces, Craft-repair, Farming-fishing, Handlers-cleaners, Machine-op-inspct, Other-service, Priv-house-serv, Transport-moving 4835 5383 <=50K (0.75512 0.24488) *
      13) occupation: Adm-clerical, Exec-managerial, Prof-specialty, Protective-serv, Sales, Tech-support 3028 4185 <=50K (0.53236 0.46764) *
    7) education.num > 12.5 3394 4101 >50K (0.29228 0.70772) *

As seen in the decision tree outline, splits education level have the most significant probabilities, followed my splits in occupation and lastly splits in marital status.

Let's now test the model over the testing set.

```
pred3 <- predict(tree1, newdata = test, type = "class")
table(pred3, test$income)
```

```
##
## pred3     <=50K  >50K
##    <=50K   5882  1180
##    >50K     263   816
```

Looking at the table, we see that we got 5882 true positives, 816 true negatives, 1180 false positives, and 263 false negatives.

Decision Tree metrics

These predictions compute to be the following metrics:

```
acc3 <- (5882 + 816) / (5882 + 816 + 1180 + 263)
print(paste("Accuracy:", acc3))
```

```
## [1] "Accuracy: 0.822749048028498"
```

```
# Calculate and print the sensitivity
sensitivity3 <- (5882) / (5882 + 263)
print(paste("Sensitivity:", sensitivity3))
```

```
## [1] "Sensitivity: 0.95720097640358"
```

```r
# Calculate and print the specificity
specificity3 <- (816) / (816 + 1180)
print(paste("Specificity:", specificity3))
```

```
## [1] "Specificity: 0.408817635270541"
```

We see that accuracy and sensitivity are once again decent, but this time the specificity is even worse. This means we are overestimating incomes of certain people the most out of all three models.

We can try pruning our tree to see if our performance improves:

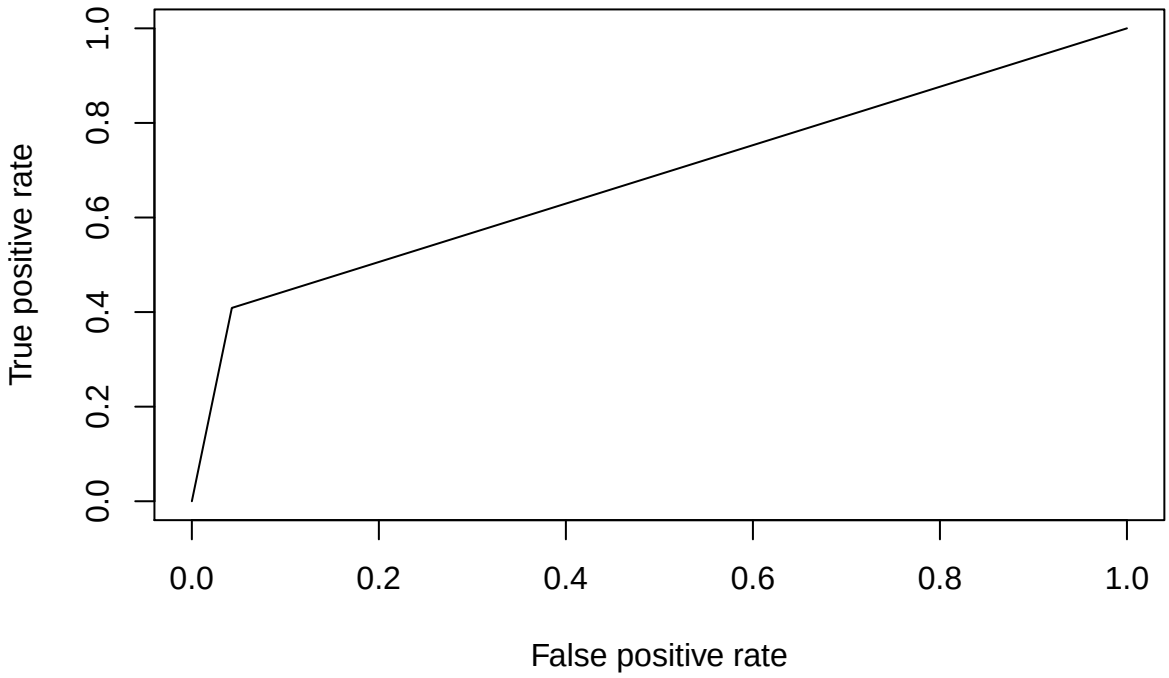
```r
tree1_pruned <- prune.tree(tree1, best=5)
pred4 <- predict(tree1_pruned, newdata = test, type = "class")
table(pred4, test$income)
```

```
##
## pred4     <=50K  >50K
##    <=50K   5882  1180
##    >50K     263   816
```

This didn't change our predictions at all, so we can continue analyzing the original tree.

We can use an ROC curve and AUC value to observe the performance.

```r
# Plot the ROC curve
predvec2 <- ifelse(as.character(pred3) == " >50K", 1, 0)
realvec2 <- ifelse(as.character(test$income) == " >50K", 1, 0)
pr3 <- prediction(predvec2, realvec2)
prf3 <- performance(pr3, measure = "tpr", x.measure = "fpr")
plot(prf3)
```




```
# Calculate and print the AUC value
auc3 <- performance(pr3, measure = "auc")
auc3 <- auc3@y.values[[1]]
print(paste("AUC:", auc3))
```

```
## [1] "AUC: 0.683009305837061"
```

We see that the ROC curve shoots upwards and to the right in a straighter with even more space left on the top left. We also have the lowest AUC value of all three models, which means our predictive value is the worst of the three.

Results analysis

When analyzing the results of the algorithms, it is important to take into account all the metrics that we calculated earlier. These being accuracy, sensitivity, specificity, ROC curve, and AUC value.

Performance and rankings

Logistic regression showed to have the highest accuracy of the three models of 0.832 followed by 0.820 for Naive Bayes and 0.823, but had a worse sensitivity than the decision tree algorithm. However, this came at a cost since decision tree had the lowest specificity of 0.41. Regardless of it's high sensitivity, it's low specificity shows that the decision tree algorithm is only good at predicting when someone has an income <=50K. Logistic regression also had the highest AUC or area under curve value of 0.88, followed by Naive

Bayes at 0.77 and then decision tree at 0.68. This is further depicted by each algorithm's ROC curve, where logistic regression's shoots up farther than Naive Bayes's and decision tree's ROC curves. Based on all these metrics, I believe the following is the correct rankings for the three algorithms:

1) Logistic Regression
2) Naive Bayes
3) Decision Tree

Logistic regression is clearly the best performing algorithm because it has both the best accuracy and best predictive value given it's AUC and ROC curve. I believe that Naive Bayes performs better than decision tree because it has a better accuracy. Although decision tree has a better sensitivity than both logistic regression and Naive Bayes and has a slightly better accuracy than Naive Bayes, it's specificity is lacking to the point where it is only good at predicting one class of the target, which is why Naive Bayes performs better than it overall as seen by it's better AUC value than decision tree.

Why Logistic Regression was the best

Let's compare logistic regression with the Naive Bayes and decision tree algorithms to see why it may have performed better in our data set.

Logistic Regression vs. Naive Bayes

Logistic regression is a discriminative classifier, meaning that it directly estimates the parameters of P(Y|X). Naive Bayes is a generative classifier, meaning that it directly estimates parameters or P(Y) and P(X|Y). Naive Bayes also has the naive assumption, which assumes that each predictor is independent of each other. If this assumption holds true, which it generally doesn't, then logistic regression and Naive Bayes over the same data set as the training set approaches infinity would converge towards very similar classifiers. In this dataset, we see some predictors that could be not independent, such as workclass and occupation. This would somewhat hurt the performance of Naive Bayes, which could suggest why it had worse performance than logistic regression. Furthermore, Naive Bayes will generally do better with smaller data sets, while logistic regression will improve in performance as the size of the data set increases. Because the size of our data set was so big, this could suggest why logistic regression performed better than Naive Bayes.

Logistic Regression vs. Decision Tree

Decision tree classification is very prone to overfitting. Even after pruning the data to help generalize it, we saw no improved performance, further instating the fact that the classifier was likely heavily overfitted. Although the assignment asked for no Random Forest/Boosting, this may have helped overcome the high variance that we see with most decision tree classifiers. Since we were restricuted from doing so, however, we were left with a poor performing decision tree classifier compared to our logistic regression classifier.

Big picture

Our classifiers were able to to learn about some factors that may contribute to being high or low income. Specifically, we were able to see the impact of age, workclass, education level, marital status, occupation, race, sex, and hours per week on income level. from our models, we specifically saw that age, education level, sex, and hours-per-week had the largest impact on income. It was interesting to see that sex had a much larger impact on income than race did. This information is useful because we now have a better idea of gender inequality and it's actual statistical impact on a person's income. Education level and hours-per-week were pretty obvious indicators of a person's income, so this information was not as useful. We also saw that

white people had a much higher probability of being in the higher income bracket, but this could be due to a much larger number of observations being from white people. In the big picture, we can use these factors to help balance out the factors that provide an unfair increase in income, such as things like sex. We can also observe how marital-status affects income, and explore ideas of whether or not we think it's fair to tax based on marital status.