Classification: Logistic Regression, kNN, Decision Trees

Haniyyah Hamid, Jered Hightower, Sai Gonuguntla

10/8/2022

Logistic Regression

Data set used: https://www.kaggle.com/datasets/lodetomasi1995/income-classification?datasetId=149550&language=null

Importing data

```
data1 <- read.csv("income_dataset.csv")
str(data1)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass     : chr  " State-gov" " Self-emp-not-inc" " Private" " Private" ...
##  $ fnlwgt        : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ education     : chr  " Bachelors" " Bachelors" " HS-grad" " 11th" ...
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital.status: chr  " Never-married" " Married-civ-spouse" " Divorced" " Married-civ-spouse" ...
##  $ occupation    : chr  " Adm-clerical" " Exec-managerial" " Handlers-cleaners" " Handlers-cleaners" ...
##  $ relationship  : chr  " Not-in-family" " Husband" " Not-in-family" " Husband" ...
##  $ race          : chr  " White" " White" " White" " Black" ...
##  $ sex           : chr  " Male" " Male" " Male" " Male" ...
##  $ capital.gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital.loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ native.country: chr  " United-States" " United-States" " United-States" " United-States" ...
##  $ income        : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

Data cleaning

We will remove unnecessary columns. We want the education.num, hours.per.week, age, and income columns. Income will be the target.

```
df <- data1[,c(1, 5, 13, 15)]
df$income <- factor(df$income)
str(df)
```

```
## 'data.frame':    32561 obs. of  4 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ 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 ...
```

```
head(df)
```

```
##   age education.num hours.per.week income
## 1  39            13             40  <=50K
## 2  50            13             13  <=50K
## 3  38             9             40  <=50K
## 4  53             7             40  <=50K
## 5  28            13             40  <=50K
## 6  37            14             40  <=50K
```

Handle missing values

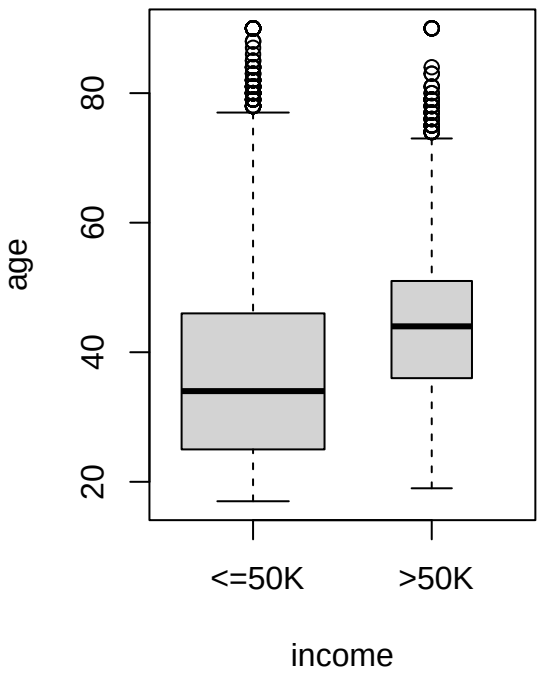
Checking to see if there are any missing data within the data frame, which there aren't.

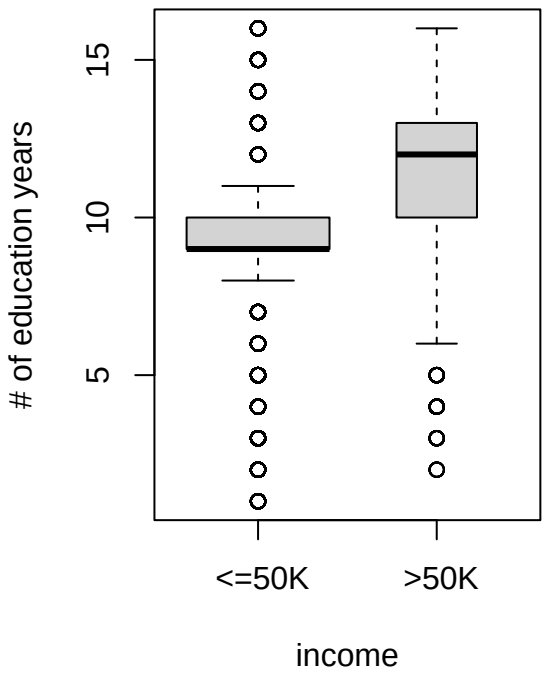
```
sapply(df, function(x) sum(is.na(x)==TRUE))
```

```
##            age  education.num hours.per.week         income
##              0              0              0              0
```

Plotting data

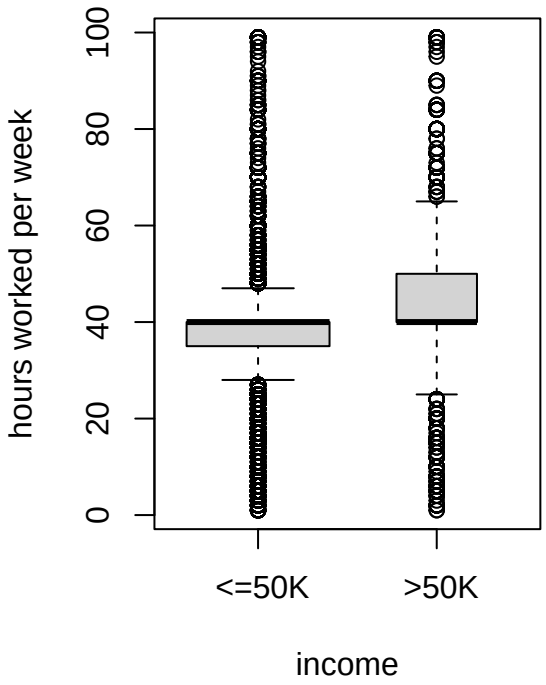
```
par(mfrow=c(1,2))
plot(df$income, df$age, xlab="income", ylab="age", varwidth=TRUE)
plot(df$income, df$education.num, xlab="income", ylab="# of education years", varwidth=TRUE)
```






```
plot(df$income, df$hours.per.week, xlab="income", ylab="hours worked per week", varwidth=TRUE)
summary(df)
```

```
##       age         education.num   hours.per.week     income
##  Min.   :17.00   Min.   : 1.00   Min.   : 1.00    <=50K:24720
##  1st Qu.:28.00   1st Qu.: 9.00   1st Qu.:40.00    >50K : 7841
##  Median :37.00   Median :10.00   Median :40.00
##  Mean   :38.58   Mean   :10.08   Mean   :40.44
##  3rd Qu.:48.00   3rd Qu.:12.00   3rd Qu.:45.00
##  Max.   :90.00   Max.   :16.00   Max.   :99.00
```



We see that with the predictor age, the median age for a person who makes <=50k is about 35 and about 43 for a person who makes >50k. With the predictor being # of years educated (how many years spent getting an education), the median years for a person who makes <=50k is about 9 (HS grad) and about 13 (bachelors) for a person who makes >50k. With the predictor # of hours worked per week, the median # of hours is about 40 for someone who makes <50k, and this is the same median for someone for makes >50k. However the third quartile of the box plot for someone who makes >50k is much larger than that of someone who makes <=50k.

Train and test

80/20 train and test

```
set.seed(1234)
i <- sample(1:nrow(df), 0.8*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

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
## -2.8259  -0.6987  -0.4412  -0.1360   2.9416
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -8.542165   0.123558  -69.14   <2e-16 ***
## age             0.046440   0.001298   35.79   <2e-16 ***
## education.num   0.346724   0.007254   47.80   <2e-16 ***
## hours.per.week  0.043185   0.001425   30.30   <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: 28706  on 26047  degrees of freedom
## Residual deviance: 23144  on 26044  degrees of freedom
## AIC: 23152
##
## Number of Fisher Scoring iterations: 5
```

We see that the P value on all the predictors indicate that they are very good predictors for the target. Age has the least standard error (0.001298) of the 3 predictors, while the number of education years has the highest (0.007254).

Evaluate on the test set

```
probs <- predict(glm1, newdata=test, type="response")
pred <- ifelse(probs>0.5, " >50K", " <=50K")
acc <- mean(pred==test$income)
print(paste("accuracy = ", acc))
```

```
## [1] "accuracy =  0.788423153692615"
```

```
table(pred, test$income)
```

```
##
## pred      <=50K  >50K
##    <=50K   4621  1077
##    >50K     301   514
```

Confusion matrix

```
library(caret)
```

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

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

```
confusionMatrix(as.factor(pred), reference=test$income)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  <=50K  >50K
##      <=50K   4621  1077
##      >50K     301   514
##
##                Accuracy : 0.7884
##                  95% CI : (0.7783, 0.7983)
##     No Information Rate : 0.7557
##     P-Value [Acc > NIR] : 2.408e-10
##
##                   Kappa : 0.3137
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9388
##             Specificity : 0.3231
##          Pos Pred Value : 0.8110
##          Neg Pred Value : 0.6307
##              Prevalence : 0.7557
##          Detection Rate : 0.7095
##    Detection Prevalence : 0.8749
##       Balanced Accuracy : 0.6310
##
##        'Positive' Class :  <=50K
##
```

Analysis of the Logistic Regression model on the dataset

We find that the accuracy is 0.7884, which implies that the logistic regression model accurate enough to predict future values of the dataset. We find that the sensitivity was 0.9388, showing that the true positive rate is quite high and accurate at predicting true results. We also find that the specificity was 0.3231, showing that the true negative rate is quite low and not as accurate at predicting false results. Therefore, based off these results logistic regression can show that age, number of education years, and number of hours worked per week are great predictors for determining if a person makes <=50k or >50k. The model itself is not entirely but at least decently accurate at predicting future results.

kNN Classification

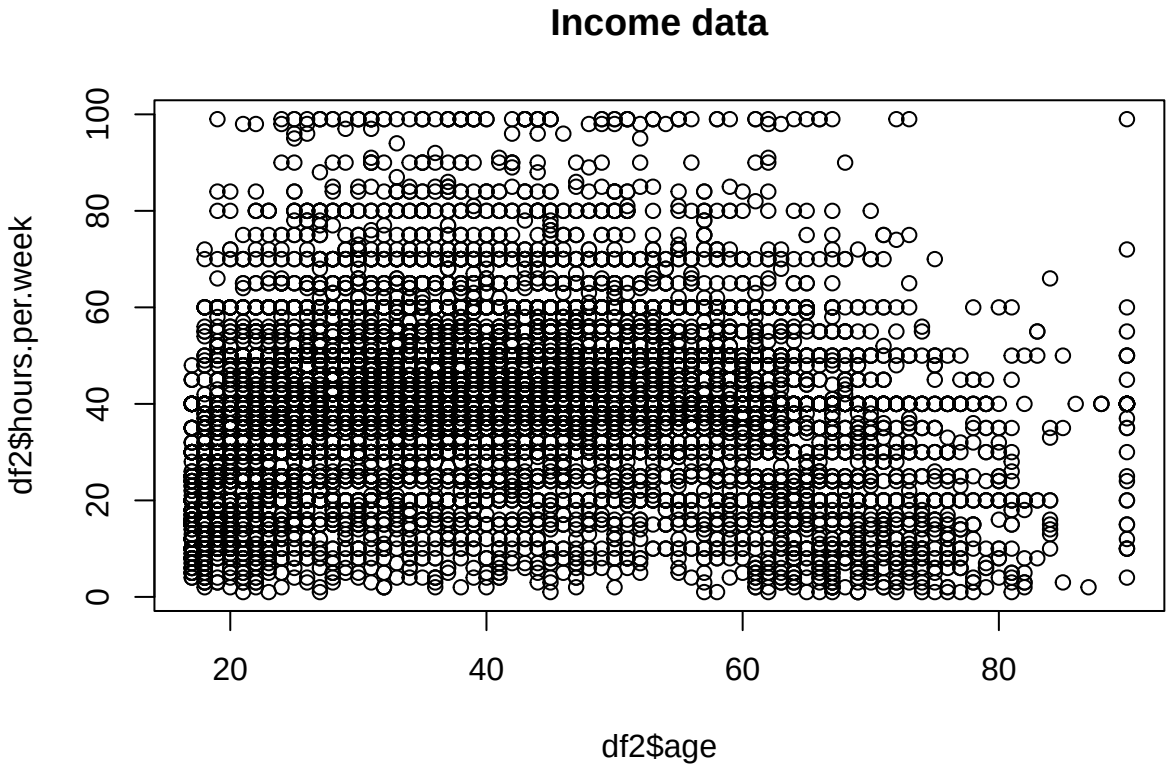
Read in data in a new dataframe

```
df2 <- data1[,c(1, 5, 13, 15)]
str(df2)
```

```
## 'data.frame':    32561 obs. of  4 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ hours.per.week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ income        : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

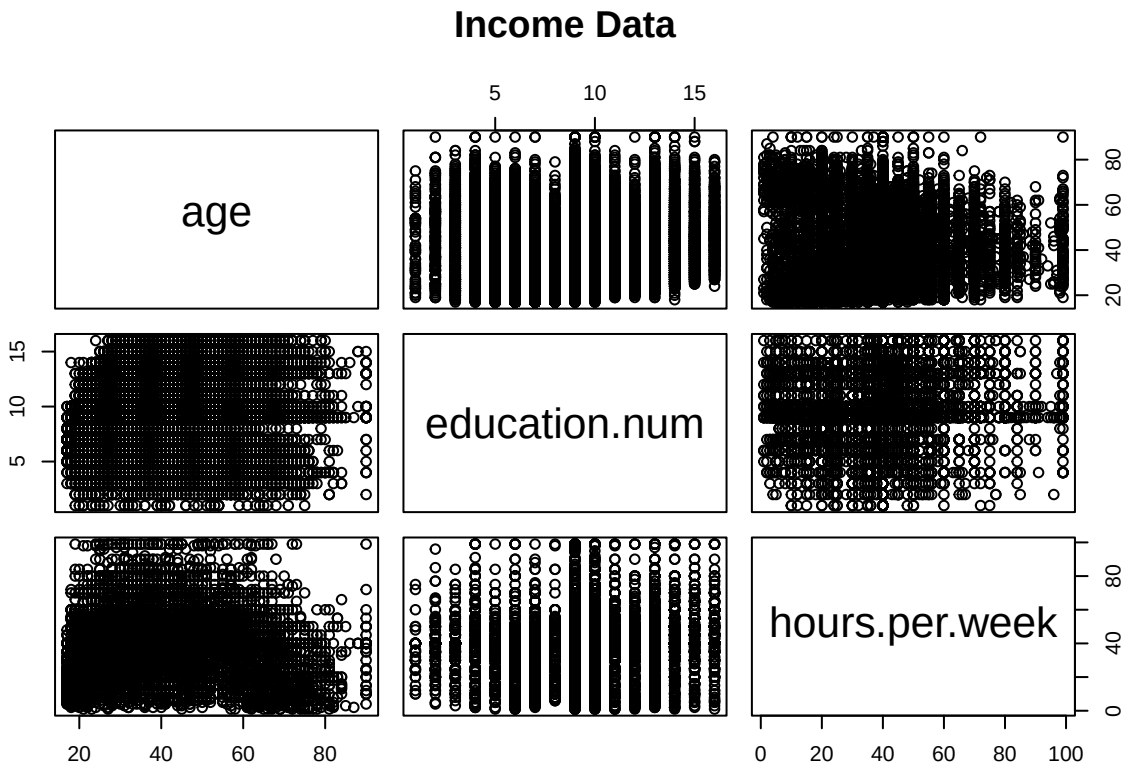
Plotting

```
plot(df2$age, df2$hours.per.week, pch=21, bg=c("red", "blue") [unclass(df2$income)], main="Income data"
```



Pair scatter plots

```
pairs(df2[1:3], main="Income Data", pch=21, bg=c("red", "green3")[unclass(df2$income)])
```



Divide into train/test sets

```
set.seed(1958)
ind <- sample(2, nrow(df), replace=TRUE, prob=c(0.67, 0.33))
df2.train <- df2[ind==1, 1:3]
df2.test <- df2[ind==2, 1:3]
df2.trainLabels <- df2[ind==1, 4]
df2.testLabels <- df2[ind==2, 4]
```

Classify

```
library(class)
df2_pred <- knn(train=df2.train, test=df2.test, cl=df2.trainLabels, k=3)
```

Compute accuracy

```
results <- df2_pred == df2.testLabels
acc <- length(which(results==TRUE)) / length(results)
print(paste("accuracy = ", acc))
```

```
## [1] "accuracy =  0.77592936802974"
```

```
table(results, df2_pred)
```

```
##        df2_pred
## results  <=50K  >50K
##   FALSE   1603   808
##   TRUE    7347  1002
```

With kNN classification we find that we get an accuracy of about 0.78. Meaning that with knn clustering, we find that we are able to classify the results fairly accurately.

Decision Trees

Read in data in a new dataframe

```
df3 <- data1[,c(1, 5, 13, 15)]
str(df3)
```

```
## 'data.frame':    32561 obs. of  4 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ hours.per.week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ income        : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

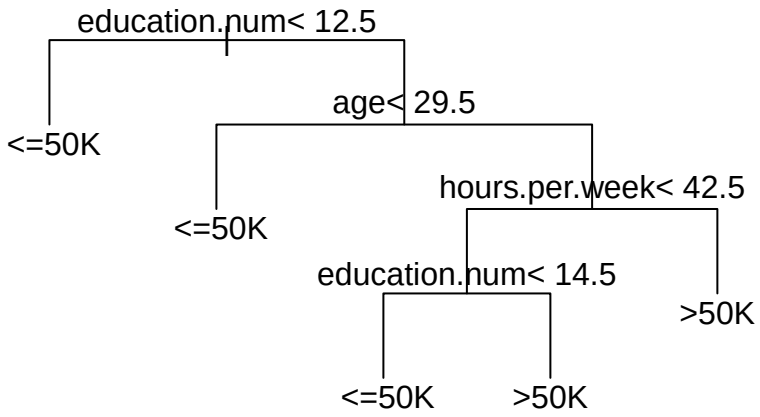
Using rpart

```
library(rpart)
tree1 <- rpart(df3$income~., data=df3, method="class")
tree1
```

```
## n= 32561
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 32561 7841  <=50K (0.7591904 0.2408096)
##    2) education.num< 12.5 24494 3932  <=50K (0.8394709 0.1605291) *
##    3) education.num>=12.5 8067 3909  <=50K (0.5154332 0.4845668)
##      6) age< 29.5 1617  232  <=50K (0.8565244 0.1434756) *
##      7) age>=29.5 6450 2773  >50K (0.4299225 0.5700775)
##       14) hours.per.week< 42.5 3501 1667  <=50K (0.5238503 0.4761497)
##         28) education.num< 14.5 3103 1399  <=50K (0.5491460 0.4508540) *
##         29) education.num>=14.5 398  130  >50K (0.3266332 0.6733668) *
##       15) hours.per.week>=42.5 2949  939  >50K (0.3184130 0.6815870) *
```

Plotting the rpart tree

```
plot(tree1, uniform=TRUE, margin=0.2)
text(tree1)
```



Using tree() package with training data

```
library(tree)
set.seed(1958)
j <- sample(1:nrow(df), 0.8*nrow(df), replace=FALSE)
train <- df3[j,]
test <- df3[-j, ]
tree2 <- tree(as.factor(income)~., data=train)
tree_pred <- predict(tree2, newdata=test, type="class")
table(tree_pred, test$income)
```

```
##
## tree_pred  <=50K  >50K
##     <=50K   4386   825
##     >50K     570   732
```

```
mean(tree_pred==test$income)
```

```
## [1] 0.785813
```

We find that with making a tree using training data and then evaluating it on test data, the accuracy of the decision tree was ~0.79. This value is very similar to the accuracy found when using the logistic regression model to predict the results.