Adult Salary Classification

Raghuvir Reddy

2/21/2022

#Fit classification models to predict whether person makes over 50K a year.

In this report I will use logistic regression, Linear Discriminate Analysis, and K-Nearest Neighbors to classify income.

Data: Census Income data set (https://archive.ics.uci.edu/ml/ datasets/census+income)

```
#Read in dataset
adult <- read.csv(file="/Users/raghu/RStudio Projects/Adult Salary/adult.csv", na.strings = c("?"), stri
head(adult)
```

```
##   age workclass fnlwgt    education educational.num     marital.status
## 1  25   Private 226802         11th               7      Never-married
## 2  38   Private  89814      HS-grad               9 Married-civ-spouse
## 3  28 Local-gov 336951   Assoc-acdm              12 Married-civ-spouse
## 4  44   Private 160323 Some-college              10 Married-civ-spouse
## 5  18      <NA> 103497 Some-college              10      Never-married
## 6  34   Private 198693         10th               6      Never-married
##          occupation  relationship  race gender capital.gain capital.loss
## 1 Machine-op-inspct     Own-child Black   Male            0            0
## 2   Farming-fishing       Husband White   Male            0            0
## 3   Protective-serv       Husband White   Male            0            0
## 4 Machine-op-inspct       Husband Black   Male         7688            0
## 5              <NA>     Own-child White Female            0            0
## 6     Other-service Not-in-family White   Male            0            0
##   hours.per.week native.country income
## 1             40  United-States  <=50K
## 2             50  United-States  <=50K
## 3             40  United-States   >50K
## 4             40  United-States   >50K
## 5             30  United-States  <=50K
## 6             30  United-States  <=50K
```

```
# colnames
colnames(adult)
```

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

```r
# checking dimensions
dim(adult)
```

```
## [1] 48842    15
```

```r
# checking structure
str(adult)
```

```
## 'data.frame':    48842 obs. of  15 variables:
##  $ age            : int  25 38 28 44 18 34 29 63 24 55 ...
##  $ workclass      : Factor w/ 8 levels "Federal-gov",..: 4 4 2 4 NA 4 NA 6 4 4 ...
##  $ fnlwgt         : int  226802 89814 336951 160323 103497 198693 227026 104626 369667 104996 ...
##  $ education      : Factor w/ 16 levels "10th","11th",..: 2 12 8 16 16 1 12 15 16 6 ...
##  $ educational.num: int  7 9 12 10 10 6 9 15 10 4 ...
##  $ marital.status : Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5 3 3 3 5 5 5 3 5 3 ...
##  $ occupation     : Factor w/ 14 levels "Adm-clerical",..: 7 5 11 7 NA 8 NA 10 8 3 ...
##  $ relationship   : Factor w/ 6 levels "Husband","Not-in-family",..: 4 1 1 1 4 2 5 1 5 1 ...
##  $ race           : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 3 5 5 3 5 5 3 5 5 5 ...
##  $ gender         : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 2 2 1 2 ...
##  $ capital.gain   : int  0 0 0 7688 0 0 0 3103 0 0 ...
##  $ capital.loss   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week : int  40 50 40 40 30 30 40 32 40 10 ...
##  $ native.country : Factor w/ 41 levels "Cambodia","Canada",..: 39 39 39 39 39 39 39 39 39 39 ...
##  $ income         : Factor w/ 2 levels "<=50K",">50K": 1 1 2 2 1 1 1 2 1 1 ...
```

```r
# Checking missing values
colSums(is.na(adult))
```

```
##             age       workclass          fnlwgt       education educational.num
##               0            2799               0               0               0
##  marital.status      occupation    relationship            race          gender
##               0            2809               0               0               0
##    capital.gain    capital.loss  hours.per.week  native.country          income
##               0               0               0             857               0
```

```r
# dropping missing values
adult <- na.omit(adult)
```

Exploratory Data Analysis

```r
# proportion of income earned
prop <- ((table(adult$income)))
prop
```

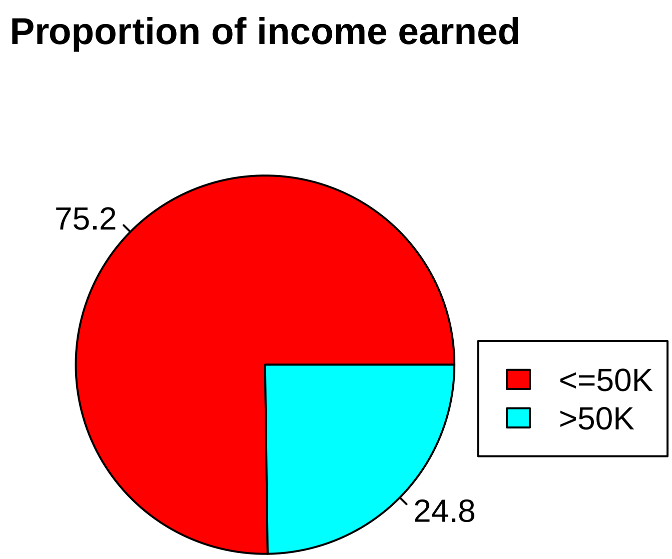
```
##
## <=50K  >50K
## 34014 11208
```

```r
piepercent<- round(100*prop/sum(prop), 1)
piepercent
```

```
##
## <=50K  >50K
##  75.2  24.8
```

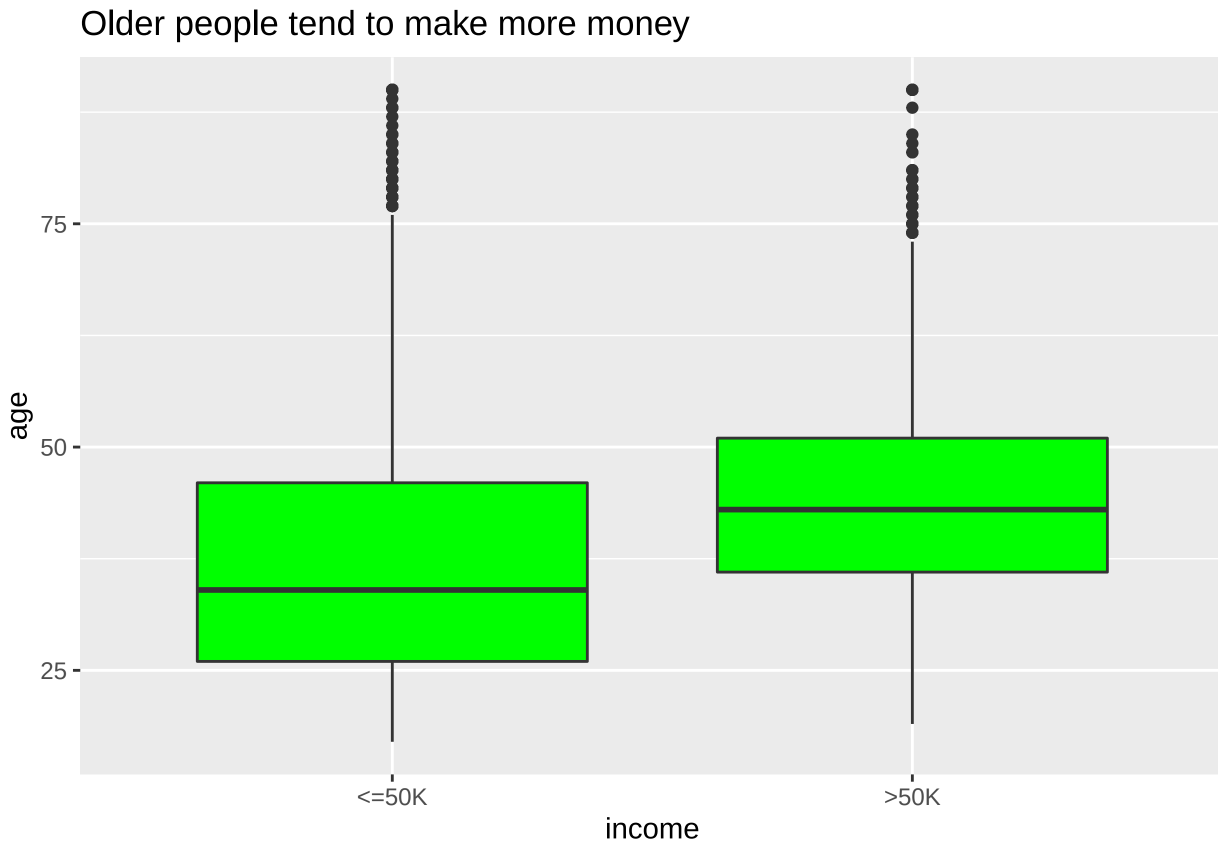
75.2% of sample has income <=50K.

```r
# Pie chart of income
pie(prop, labels = piepercent, main = "Proportion of income earned", col = rainbow(length(prop)))
legend(.9, .1, c("<=50K",">50K"), fill = rainbow(length(prop)))
```

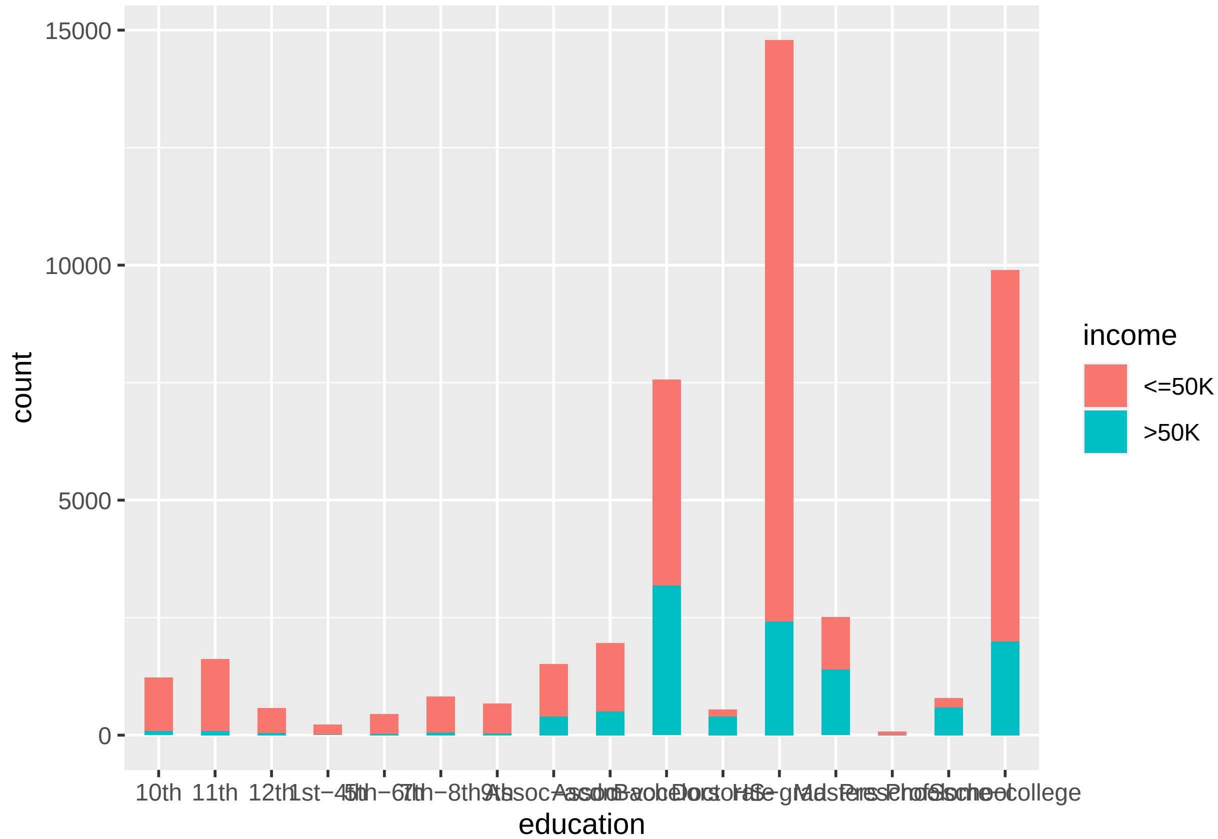



```r
library(ggplot2) #ggplot2 Functions
```

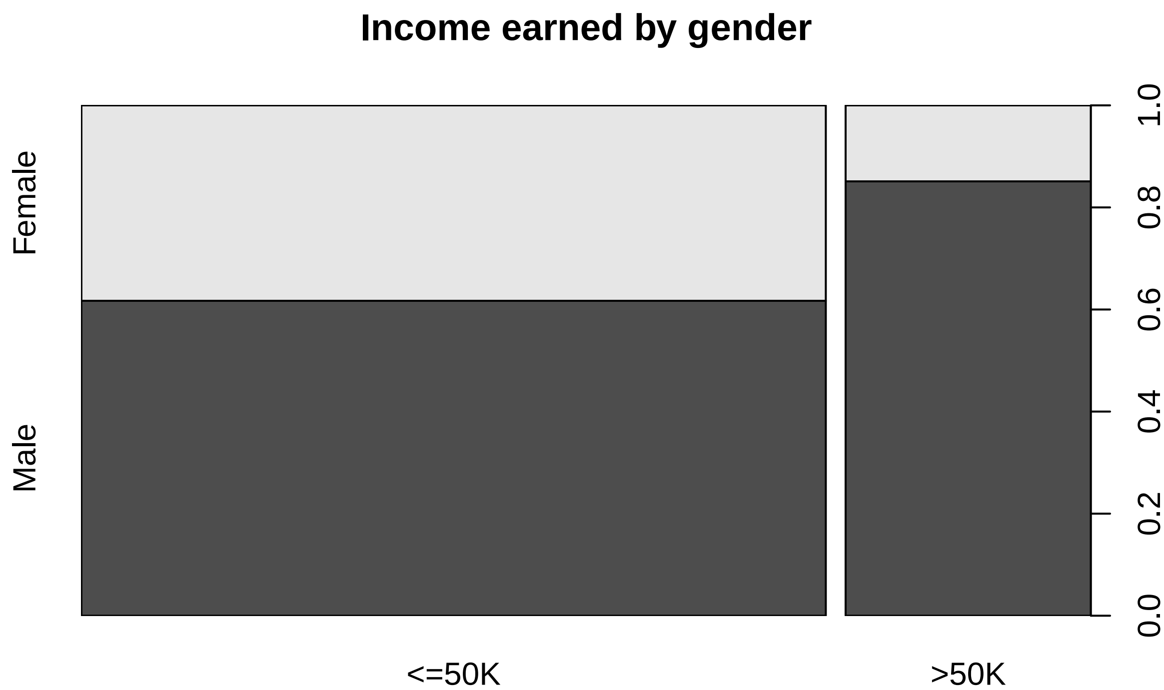
```r
# EDA - age and income
ggplot(adult, aes(x=income, y=age)) +  geom_boxplot(fill='green') + ggtitle("Older people tend to make r
```




```
# education and income level
library(ggplot2)
ggplot(data=adult, aes(x=education,fill = income)) + stat_count(width = 0.5)
```




```
colors <- c("red", "blue")
plot((adult$income), adult$gender, main = "Income earned by gender",xlab = "", ylab = "")
```



Data modeling / wrangling

The dataset used in this study has fourteen independent variables and one dependent variable i.e. income. Intuitively, we can say that age, workclass, education, occupation and hours.per.week would be significant in predicting the income. Relationship, race and sex are also going to have strong predictive power.

Removal of Features

Opted to not use the features: 'fnlwgt', 'relationships', 'education', and 'capitalGains/Loss'. These features either were not useful for our analysis or had too many outliers. 'education_num' preferred over 'education'.

```
data.frame(colnames(adult)) #Returns column index numbers in table format
```

```
##    colnames.adult.
## 1              age
## 2        workclass
## 3           fnlwgt
## 4        education
## 5  educational.num
## 6   marital.status
## 7       occupation
## 8     relationship
## 9             race
## 10          gender
## 11    capital.gain
```

```
## 12    capital.loss
## 13  hours.per.week
## 14  native.country
## 15          income
```

Drop cols: 3, 4, 11, 12,

```
# dropping fnlwgt, education, capital.gain, capital.loss
df <- adult[,-c(3,4,11,12)]
```

```
# revaluing income
library(plyr)
plyr::revalue(df$income, c("<=50K" = "0", ">50K" = "1")) -> df$income
```

Splitting df into training & testing

```
# 0.7 training set, 0.3 test set
set.seed(1)
train_id <- sample(1:nrow(df), nrow(df)*0.7 , replace=F)
training <- df[train_id,]
testing <- df[-train_id,]
```

Logistic Regression

```
# fit model and predict
logit_model <- glm(income ~., data = training, family = binomial)
logit_model_pred <- predict(logit_model, newdata=testing, type="response")
set.seed(1)
logit_model_pred <- ifelse(logit_model_pred > 0.5, 1, 0)
```

```
# confusion matrix
table(logit_model_pred , testing$income)
```

```
##
## logit_model_pred    0    1
##                0 9392 1477
##                1  797 1901
```

```
# log error
mean(logit_model_pred != testing$income)
```

```
## [1] 0.1676126
```

16.8% miss classification error in logistic regression model.

Linear Discriminant Analysis

```
str(training)
```

```
## 'data.frame':    31655 obs. of  11 variables:
##  $ age            : int  29 44 30 20 29 65 46 42 43 23 ...
##  $ workclass      : Factor w/ 8 levels "Federal-gov",..: 7 4 4 2 4 4 6 4 4 4 ...
##  $ educational.num: int  14 13 9 10 10 9 11 14 10 10 ...
##  $ marital.status : Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 3 3 3 5 1 7 3 3 3 5 ...
##  $ occupation     : Factor w/ 14 levels "Adm-clerical",..: 10 10 8 1 12 1 4 10 14 8 ...
##  $ relationship   : Factor w/ 6 levels "Husband","Not-in-family",..: 1 1 3 4 5 5 1 1 1 3 ...
##  $ race           : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 2 5 5 5 5 5 5 5 5 5 ...
##  $ gender         : Factor w/ 2 levels "Female","Male": 2 2 1 1 1 1 2 2 2 1 ...
##  $ hours.per.week : int  20 45 40 20 54 40 40 40 45 25 ...
##  $ native.country : Factor w/ 41 levels "Cambodia","Canada",..: 36 39 39 39 39 39 39 39 39 39 ...
##  $ income         : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 2 2 1 1 ...
```

Since LDA assumes that the observations within each class come from a normal distribution and thus expects predictors to be continuous variables, we only select "age", "education-num", and "hours-per-week" in this model.

```
# fit model
library(MASS)
lda_income_fit <- lda(income ~ age + educational.num + hours.per.week, family = binomial(logit), data =
lda_income_fit
```

```
## Call:
## lda(income ~ age + educational.num + hours.per.week, data = training,
##     family = binomial(logit))
##
## Prior probabilities of groups:
##         0         1
## 0.7526457 0.2473543
##
## Group means:
##        age educational.num hours.per.week
## 0 36.76180        9.638447       39.37834
## 1 44.02465       11.603576       45.66922
##
## Coefficients of linear discriminants:
##                        LD1
## age             0.04074195
## educational.num 0.30325207
## hours.per.week  0.03446943
```

```
# predict
lda_pred_income <-  predict(lda_income_fit, testing)$class

# confusion matrix
table(lda_pred_income, testing$income)
```

```
##
## lda_pred_income    0    1
##               0 9525 2312
##               1  664 1066
```

```
# error rate of lda model
mean(lda_pred_income != testing$income)
```

```
## [1] 0.2193558
```

22% error rate in classification.

K-Nearest Neighbors (KNN)

KNN models can also handle categorical variables, but this requires us to convert categorical variables to m-1 (m=levels of the categorical variables) dummy variables with value equals to 1 or 0. One hot encoding can be used to overcome this.

```
# Create new dataframe with dummy variables using one hot encoding
library(mltools)
library(data.table)
df_knn <- one_hot(as.data.table(df))
```

```
# head
head(df_knn)
```

```
##    age workclass_Federal-gov workclass_Local-gov workclass_Never-worked
## 1:  25                     0                   0                      0
## 2:  38                     0                   0                      0
## 3:  28                     0                   1                      0
## 4:  44                     0                   0                      0
## 5:  34                     0                   0                      0
## 6:  63                     0                   0                      0
##    workclass_Private workclass_Self-emp-inc workclass_Self-emp-not-inc
## 1:                 1                      0                          0
## 2:                 1                      0                          0
## 3:                 0                      0                          0
## 4:                 1                      0                          0
## 5:                 1                      0                          0
## 6:                 0                      0                          1
##    workclass_State-gov workclass_Without-pay educational.num
## 1:                   0                     0               7
## 2:                   0                     0               9
## 3:                   0                     0              12
## 4:                   0                     0              10
## 5:                   0                     0               6
## 6:                   0                     0              15
##    marital.status_Divorced marital.status_Married-AF-spouse
## 1:                       0                                0
## 2:                       0                                0
## 3:                       0                                0
```

```
## 4:                      0                                0
## 5:                      0                                0
## 6:                      0                                0
##    marital.status_Married-civ-spouse marital.status_Married-spouse-absent
## 1:                                 0                                    0
## 2:                                 1                                    0
## 3:                                 1                                    0
## 4:                                 1                                    0
## 5:                                 0                                    0
## 6:                                 1                                    0
##    marital.status_Never-married marital.status_Separated marital.status_Widowed
## 1:                            1                        0                      0
## 2:                            0                        0                      0
## 3:                            0                        0                      0
## 4:                            0                        0                      0
## 5:                            1                        0                      0
## 6:                            0                        0                      0
##    occupation_Adm-clerical occupation_Armed-Forces occupation_Craft-repair
## 1:                       0                       0                       0
## 2:                       0                       0                       0
## 3:                       0                       0                       0
## 4:                       0                       0                       0
## 5:                       0                       0                       0
## 6:                       0                       0                       0
##    occupation_Exec-managerial occupation_Farming-fishing
## 1:                          0                          0
## 2:                          0                          1
## 3:                          0                          0
## 4:                          0                          0
## 5:                          0                          0
## 6:                          0                          0
##    occupation_Handlers-cleaners occupation_Machine-op-inspct
## 1:                            0                            1
## 2:                            0                            0
## 3:                            0                            0
## 4:                            0                            1
## 5:                            0                            0
## 6:                            0                            0
##    occupation_Other-service occupation_Priv-house-serv
## 1:                        0                          0
## 2:                        0                          0
## 3:                        0                          0
## 4:                        0                          0
## 5:                        1                          0
## 6:                        0                          0
##    occupation_Prof-specialty occupation_Protective-serv occupation_Sales
## 1:                         0                          0                0
## 2:                         0                          0                0
## 3:                         0                          1                0
## 4:                         0                          0                0
## 5:                         0                          0                0
## 6:                         1                          0                0
##    occupation_Tech-support occupation_Transport-moving relationship_Husband
## 1:                       0                           0                    0
```

```
## 2:                      0                      0                     1
## 3:                      0                      0                     1
## 4:                      0                      0                     1
## 5:                      0                      0                     0
## 6:                      0                      0                     1
##    relationship_Not-in-family relationship_Other-relative
## 1:                          0                           0
## 2:                          0                           0
## 3:                          0                           0
## 4:                          0                           0
## 5:                          1                           0
## 6:                          0                           0
##    relationship_Own-child relationship_Unmarried relationship_Wife
## 1:                      1                      0                 0
## 2:                      0                      0                 0
## 3:                      0                      0                 0
## 4:                      0                      0                 0
## 5:                      0                      0                 0
## 6:                      0                      0                 0
##    race_Amer-Indian-Eskimo race_Asian-Pac-Islander race_Black race_Other
## 1:                       0                       0          1          0
## 2:                       0                       0          0          0
## 3:                       0                       0          0          0
## 4:                       0                       0          1          0
## 5:                       0                       0          0          0
## 6:                       0                       0          0          0
##    race_White gender_Female gender_Male hours.per.week native.country_Cambodia
## 1:          0             0           1             40                       0
## 2:          1             0           1             50                       0
## 3:          1             0           1             40                       0
## 4:          0             0           1             40                       0
## 5:          1             0           1             30                       0
## 6:          1             0           1             32                       0
##    native.country_Canada native.country_China native.country_Columbia
## 1:                     0                    0                       0
## 2:                     0                    0                       0
## 3:                     0                    0                       0
## 4:                     0                    0                       0
## 5:                     0                    0                       0
## 6:                     0                    0                       0
##    native.country_Cuba native.country_Dominican-Republic native.country_Ecuador
## 1:                   0                                0                      0
## 2:                   0                                0                      0
## 3:                   0                                0                      0
## 4:                   0                                0                      0
## 5:                   0                                0                      0
## 6:                   0                                0                      0
##    native.country_El-Salvador native.country_England native.country_France
## 1:                          0                      0                     0
## 2:                          0                      0                     0
## 3:                          0                      0                     0
## 4:                          0                      0                     0
## 5:                          0                      0                     0
## 6:                          0                      0                     0
```

```
##    native.country_Germany native.country_Greece native.country_Guatemala
## 1:                      0                     0                        0
## 2:                      0                     0                        0
## 3:                      0                     0                        0
## 4:                      0                     0                        0
## 5:                      0                     0                        0
## 6:                      0                     0                        0
##    native.country_Haiti native.country_Holand-Netherlands
## 1:                    0                                 0
## 2:                    0                                 0
## 3:                    0                                 0
## 4:                    0                                 0
## 5:                    0                                 0
## 6:                    0                                 0
##    native.country_Honduras native.country_Hong native.country_Hungary
## 1:                       0                   0                      0
## 2:                       0                   0                      0
## 3:                       0                   0                      0
## 4:                       0                   0                      0
## 5:                       0                   0                      0
## 6:                       0                   0                      0
##    native.country_India native.country_Iran native.country_Ireland
## 1:                    0                   0                      0
## 2:                    0                   0                      0
## 3:                    0                   0                      0
## 4:                    0                   0                      0
## 5:                    0                   0                      0
## 6:                    0                   0                      0
##    native.country_Italy native.country_Jamaica native.country_Japan
## 1:                    0                      0                    0
## 2:                    0                      0                    0
## 3:                    0                      0                    0
## 4:                    0                      0                    0
## 5:                    0                      0                    0
## 6:                    0                      0                    0
##    native.country_Laos native.country_Mexico native.country_Nicaragua
## 1:                   0                     0                        0
## 2:                   0                     0                        0
## 3:                   0                     0                        0
## 4:                   0                     0                        0
## 5:                   0                     0                        0
## 6:                   0                     0                        0
##    native.country_Outlying-US(Guam-USVI-etc) native.country_Peru
## 1:                                         0                   0
## 2:                                         0                   0
## 3:                                         0                   0
## 4:                                         0                   0
## 5:                                         0                   0
## 6:                                         0                   0
##    native.country_Philippines native.country_Poland native.country_Portugal
## 1:                          0                     0                       0
## 2:                          0                     0                       0
## 3:                          0                     0                       0
## 4:                          0                     0                       0
```

```
## 5:                          0                     0                       0
## 6:                          0                     0                       0
##    native.country_Puerto-Rico native.country_Scotland native.country_South
## 1:                          0                       0                    0
## 2:                          0                       0                    0
## 3:                          0                       0                    0
## 4:                          0                       0                    0
## 5:                          0                       0                    0
## 6:                          0                       0                    0
##    native.country_Taiwan native.country_Thailand native.country_Trinadad&Tobago
## 1:                     0                       0                              0
## 2:                     0                       0                              0
## 3:                     0                       0                              0
## 4:                     0                       0                              0
## 5:                     0                       0                              0
## 6:                     0                       0                              0
##    native.country_United-States native.country_Vietnam
## 1:                            1                      0
## 2:                            1                      0
## 3:                            1                      0
## 4:                            1                      0
## 5:                            1                      0
## 6:                            1                      0
##    native.country_Yugoslavia income_0 income_1
## 1:                         0        1        0
## 2:                         0        1        0
## 3:                         0        0        1
## 4:                         0        0        1
## 5:                         0        1        0
## 6:                         0        0        1
```

```r
# 0.7 training set, 0.3 test set
set.seed(1)
trainid <- sample(1:nrow(df_knn), nrow(df_knn)*0.7 , replace=F)
knn.train <- df_knn[trainid,]
knn.test <- df_knn[-trainid,]
```

```r
# convert to data frames
knn.train <- as.data.frame(knn.train)
knn.test <- as.data.frame(knn.test)
label = as.data.frame(knn.train$income_1)
```

```r
# Train a knn classifier and change k accordingly. k =1
library(class)
knn_pred <- knn(knn.train, knn.test, knn.train$income_1, k=1)
table(knn_pred, knn.test$income_1)
```

```
##
## knn_pred    0    1
##        0 9934  365
##        1  255 3013
```

```r
mean(knn_pred != knn.test$income_1)
```

```
## [1] 0.04569912
```

Error of only 4.6%.

```r
# k = 5
knn_pred <- knn(knn.train, knn.test, knn.train$income_1, k=5)
table(knn_pred, knn.test$income_1)
```

```
##
## knn_pred    0    1
##        0 9985  425
##        1  204 2953
```

```r
mean(knn_pred != knn.test$income_1)
```

```
## [1] 0.0463625
```

Error is around 4.6%.

```r
# k = 10
knn_pred <- knn(knn.train, knn.test, knn.train$income_1, k=10)
# confusion matrix
table(knn_pred, knn.test$income_1)
```

```
##
## knn_pred    0    1
##        0 9977  502
##        1  212 2876
```

```r
# error for k = 10
mean(knn_pred != knn.test$income_1)
```

```
## [1] 0.0526277
```

Error increased from 4.6% to 5.4% when k increased to 10.

```r
# k = 30
knn_pred <- knn(knn.train, knn.test, knn.train$income_1, k=30)
# confusion matrix
table(knn_pred, knn.test$income_1)
```

```
##
## knn_pred    0    1
##        0 9887  657
##        1  302 2721
```

```
# error for k = 30
mean(knn_pred != knn.test$income_1)
```

```
## [1] 0.07068622
```

Error is 7.1% for k = 30.

```
# k = 50
knn_pred <- knn(knn.train, knn.test, knn.train$income_1, k=50)
# confusion matrix
table(knn_pred, knn.test$income_1)
```

```
##
## knn_pred    0    1
##        0 9824  771
##        1  365 2607
```

```
# error for k = 50
mean(knn_pred != knn.test$income_1)
```

```
## [1] 0.08373259
```

Error is 8.3%.

```
# k = 100
knn_pred <- knn(knn.train, knn.test, knn.train$income_1, k=100)
# confusion matrix
table(knn_pred, knn.test$income_1)
```

```
##
## knn_pred    0    1
##        0 9738  946
##        1  451 2432
```

```
# error for k = 100
mean(knn_pred != knn.test$income_1)
```

```
## [1] 0.1029704
```

Error is 10.3%.

Ideal k value is either 1 or 5.

Summary of results

Linear Discriminant Analysis (LDA)

- LDA was worst performing method to classify income with the highest error or misclassification rate of 22%.

- LDA classification relies on Bayes theorem and attempstto solve P(X = x | Y = y) i.e for a given value of x what is the probability of Y?
- LDA makes strong assumptions: Predictors must be normal, X distributions for different classes must be far apart, no multicollinearity, and no outliers.
- In practice, it is very difficult to implement these assumptions and this might explain high error values for lda.

Logistic regression

- This model performed better than LDA having an error rate of 16.8%
- It is a good alternative to LDA to predict binary variables as it makes fewer 'strong' assumptions and is less sensitive to not normal data, outliers, and multicollinearity.
- It uses the log function to estimate probability of outcome occuring.

K-Nearest Neighbors

- Best performing model having an error rate of 4.6% for (k = 1, k = 5).
- Ideal for multiclass problems.
- K-NN is a Non-parametric algorithm i.e it doesn't make any assumption about underlying data or its distribution.
- However, KNN is sensitive to outliers and is computationally slow