Part 2 Classification

The dataset used can be found here: https://archive.ics.uci.edu/ml/datasets/Adult (https://archive.ics.uci.edu/ml/datasets/Adult)

```
library(e1071)

adultsData <- read.csv("adult.data", header=FALSE)
names(adultsData) <- c("age", "workclass", "fnlwgt", "education", "education_num", "marital_status", "occupation", "relationship", "race", "sex", "capital_gain", "capital_loss", "hours_per_week", "native_country", "income")

sapply(adultsData, function(x) sum(is.na(x)))
```

```
##            age      workclass         fnlwgt      education  education_num
##              0              0              0              0              0
## marital_status     occupation   relationship           race            sex
##              0              0              0              0              0
##   capital_gain   capital_loss hours_per_week native_country         income
##              0              0              0              0              0
```

```
adultsData <- adultsData[(complete.cases(adultsData)),]
sum(is.na(adultsData))
```

```
## [1] 0
```

```
adultsData$workclass <- factor(adultsData$workclass)
adultsData$education <- factor(adultsData$education)
adultsData$marital_status <- factor(adultsData$marital_status)
adultsData$occupation <- factor(adultsData$occupation)
adultsData$relationship <- factor(adultsData$relationship)
adultsData$race <- factor(adultsData$race)
adultsData$sex <- factor(adultsData$sex)
adultsData$native_country <- factor(adultsData$native_country)
adultsData$income <- factor(adultsData$income)

set.seed(12345)

spec <- c(train=.6, test=.2, validate=.2)
i <- sample(cut(1:nrow(adultsData), nrow(adultsData)*cumsum(c(0,spec)), labels=names(spec)))
train <- adultsData[i=="train",]
test <- adultsData[i=="test",]
vald <- adultsData[i=="validate",]

print("Correlation between age and income: ")
```

```
## [1] "Correlation between age and income: "
```

```
cor(train$age, as.numeric(train$income))
```

```
## [1] 0.2294612
```

```
print("Correlation between capital gain and income: ")
```

```
## [1] "Correlation between capital gain and income: "
```

```
cor(train$capital_gain, as.numeric(train$income))
```

```
## [1] 0.2246008
```

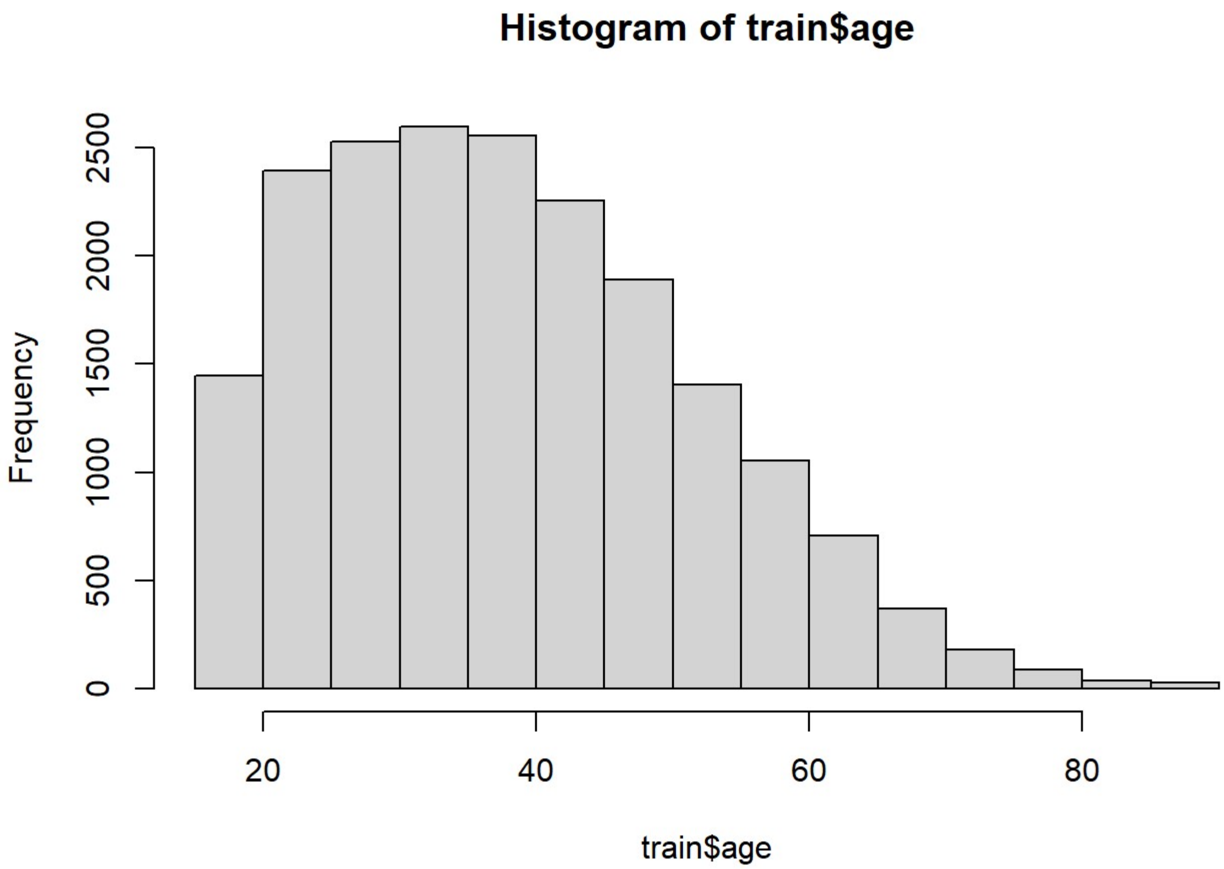
```
print("Correlation between capital loss and income: ")
```

```
## [1] "Correlation between capital loss and income: "
```

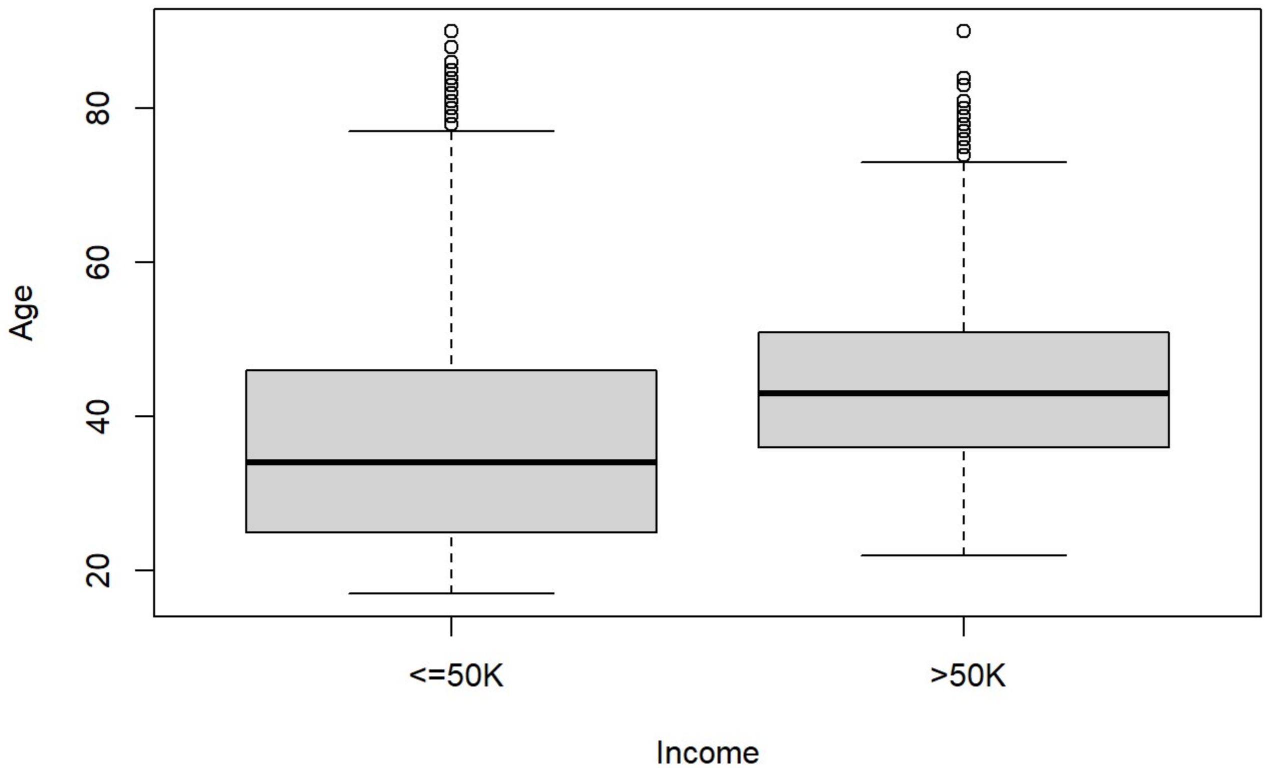
```
cor(train$capital_loss, as.numeric(train$income))
```

```
## [1] 0.1403765
```

```
hist(train$age)
```




```
boxplot(train$age~train$income, xlab="Income", ylab="Age")
```




```
svm1 <- svm(income~., data = train, kernel = "linear", cost = 10, scale = TRUE)
summary(svm1)
```

```
##
## Call:
## svm(formula = income ~ ., data = train, kernel = "linear", cost = 10,
##     scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##
## Number of Support Vectors:  6757
##
##  ( 3388 3369 )
##
##
## Number of Classes:  2
##
## Levels:
##   <=50K  >50K
```

```
pred1 <- predict(svm1, newdata = test)
svm1Tab <- table(pred1, test$income)
svm1Acc <- (sum(diag(svm1Tab))/sum(svm1Tab))

print("Linear Kernel: ")
```

```
## [1] "Linear Kernel: "
```

```
print("Accuracy: ")
```

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

```
svm1Acc
```

```
## [1] 0.8533477
```

```
print("Error rate: ")
```

```
## [1] "Error rate: "
```

```
(1 - svm1Acc)
```

```
## [1] 0.1466523
```

```
print("Sensititvity: ")
```

```
## [1] "Sensititvity: "
```

```
(svm1Tab[1,1] /(svm1Tab[1,1] + svm1Tab[2,1]))
```

```
## [1] 0.9425147
```

```
print("Specificity: ")
```

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

```
(svm1Tab[2,2] /(svm1Tab[2,2] + svm1Tab[1,2]))
```

```
## [1] 0.5770925
```

The accuracy for the linear kernel is reasonable.

```
svm2 <- svm(income~., data = train, kernel = "polynomial", cost = 10, scale = TRUE)
summary(svm2)
```

```
##
## Call:
## svm(formula = income ~ ., data = train, kernel = "polynomial", cost = 10,
##     scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  10
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  8252
##
##  ( 4157 4095 )
##
##
## Number of Classes:  2
##
## Levels:
##   <=50K  >50K
```

```
pred2 <- predict(svm2, newdata = test)
svm2Tab <- table(pred2, test$income)
svm2Acc <- (sum(diag(svm2Tab))/sum(svm2Tab))

print("Polynomial Kernel: ")
```

```
## [1] "Polynomial Kernel: "
```

```
print("Accuracy: ")
```

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

```
svm2Acc
```

```
## [1] 0.8309275
```

```
print("Error rate: ")
```

```
## [1] "Error rate: "
```

```
(1 - svm2Acc)
```

```
## [1] 0.1690725
```

```
print("Sensititvity: ")
```

```
## [1] "Sensititvity: "
```

```
(svm2Tab[1,1] /(svm2Tab[1,1] + svm2Tab[2,1]))
```

```
## [1] 0.9717652
```

```
print("Specificity: ")
```

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

```
(svm2Tab[2,2] /(svm2Tab[2,2] + svm2Tab[1,2]))
```

```
## [1] 0.3945878
```

The accuracy for the polynomial kernel is slightly lower than the linear kernel.

```
tune_svm1 <- tune(svm, income~., data=vald, kernel="polynomial",
                  ranges=list(cost=c(0.001, 0.01, 0.1, 1, 5, 10, 100)))

summary(tune_svm1)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   100
##
## - best performance: 0.156765
##
## - Detailed performance results:
##    cost     error dispersion
## 1 1e-03 0.2375307 0.01598046
## 2 1e-02 0.2333852 0.01662042
## 3 1e-01 0.2333852 0.01662042
## 4 1e+00 0.2214090 0.01568808
## 5 5e+00 0.2109664 0.01404275
## 6 1e+01 0.1945367 0.01364307
## 7 1e+02 0.1567650 0.01175288
```

```
pred_tune1 <- predict(tune_svm1$best.model, newdata = test)
tuneTab <- table(pred_tune1, test$income)
tuneAcc <- (sum(diag(tuneTab))/sum(tuneTab))

print("Tuned Polynomial Kernel: ")
```

```
## [1] "Tuned Polynomial Kernel: "
```

```
print("Accuracy: ")
```

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

```
tuneAcc
```

```
## [1] 0.840602
```

```
print("Error rate: ")
```

```
## [1] "Error rate: "
```

```
(1 - tuneAcc)
```

```
## [1] 0.159398
```

```
print("Sensititvity: ")
```

```
## [1] "Sensititvity: "
```

```
(tuneTab[1,1] /(tuneTab[1,1] + tuneTab[2,1]))
```

```
## [1] 0.9563274
```

```
print("Specificity: ")
```

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

```
(tuneTab[2,2] /(tuneTab[2,2] + tuneTab[1,2]))
```

```
## [1] 0.4820642
```

After tuning the model for the polynomial kernel the accuracy is higher than before but still lower than the linear kernel.
This could be due to the data set being not conducive to being divided polynomially.

```
svm3 <- svm(income~., data = train, kernel = "radial", cost = 10, gamma = 1, scale = TRUE)
summary(svm3)
```

```
##
## Call:
## svm(formula = income ~ ., data = train, kernel = "radial", cost = 10,
##     gamma = 1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##
## Number of Support Vectors:  14473
##
##  ( 10425 4048 )
##
##
## Number of Classes:  2
##
## Levels:
##   <=50K  >50K
```

```
pred3 <- predict(svm3, newdata = test)
svm3Tab <- table(pred3, test$income)
svm3Acc <- (sum(diag(svm3Tab))/sum(svm3Tab))

print("Radial Kernel: ")
```

```
## [1] "Radial Kernel: "
```

```
print("Accuracy: ")
```

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

```
svm3Acc
```

```
## [1] 0.7979115
```

```
print("Error rate: ")
```

```
## [1] "Error rate: "
```

```
(1 - svm3Acc)
```

```
## [1] 0.2020885
```

```
print("Sensititvity: ")
```

```
## [1] "Sensititvity: "
```

```
(svm3Tab[1,1] /(svm3Tab[1,1] + svm3Tab[2,1]))
```

```
## [1] 0.9236238
```

```
print("Specificity: ")
```

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

```
(svm3Tab[2,2] /(svm3Tab[2,2] + svm3Tab[1,2]))
```

```
## [1] 0.408433
```

The accuracy of the radial kernel is significantly lower than the other two kernels. This could perhaps be due to a poor gamma value.