```r
data <- read.csv("~/Downloads/UniversalBank.csv")
head(data)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
## 5             0                  0          0      0          1
## 6             0                  0          0      1          0
```

```r
clean_data <- subset(data, select = -c(ID, ZIP.Code))
clean_data$Personal.Loan <- as.factor(clean_data$Personal.Loan)
str(clean_data)
```

```
## 'data.frame':    5000 obs. of  12 variables:
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : int  1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal.Loan     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
```

```r
set.seed(123)
library(caret)
```

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

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

```r
train_index <- createDataPartition(clean_data$Personal.Loan, p = 0.7, list = FALSE)
train_data <- clean_data[train_index, ]
test_data <- clean_data[-train_index, ]
```

```r
library(class)
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
```

```r
train_X <- as.data.frame(lapply(train_data[, -which(names(train_data) == "Personal.Loan")], normalize))
test_X  <- as.data.frame(lapply(test_data[, -which(names(test_data) == "Personal.Loan")], normalize))

train_Y <- train_data$Personal.Loan
test_Y  <- test_data$Personal.Loan

library(class)
knn_pred <- knn(train = train_X, test = test_X, cl = train_Y, k = 5)

table(Predicted = knn_pred, Actual = test_Y)
```

```
##          Actual
## Predicted    0    1
##         0 1352   46
##         1    4   98
```

```r
conf_matrix <- table(Predicted = knn_pred, Actual = test_Y)

accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
sensitivity <- conf_matrix["1", "1"] / sum(conf_matrix[, "1"])
specificity <- conf_matrix["0", "0"] / sum(conf_matrix[, "0"])

accuracy
```

```
## [1] 0.9666667
```

```r
sensitivity
```

```
## [1] 0.6805556
```

```r
specificity
```

```
## [1] 0.9970501
```

```r
ks <- c(1, 3, 5, 7, 15)

results <- data.frame(k = ks, Accuracy = NA, Sensitivity = NA, Specificity = NA)

for (i in seq_along(ks)) {
  k_val <- ks[i]
  knn_pred <- knn(train = train_X, test = test_X, cl = train_Y, k = k_val)
  conf_matrix <- table(Predicted = knn_pred, Actual = test_Y)

  acc <- sum(diag(conf_matrix)) / sum(conf_matrix)
  sens <- conf_matrix["1", "1"] / sum(conf_matrix[, "1"])
  spec <- conf_matrix["0", "0"] / sum(conf_matrix[, "0"])

  results[i, ] <- c(k_val, acc, sens, spec)
}

results
```

```
##    k  Accuracy Sensitivity Specificity
## 1  1 0.9693333   0.8125000   0.9859882
## 2  3 0.9720000   0.7708333   0.9933628
## 3  5 0.9666667   0.6805556   0.9970501
## 4  7 0.9620000   0.6319444   0.9970501
## 5 15 0.9526667   0.5277778   0.9977876
```

```
misclassified <- test_data[knn_pred != test_Y, ]
nrow(misclassified)
```

```
## [1] 71
```

```
head(misclassified)
```

```
##     Age Experience Income Family CCAvg Education Mortgage Personal.Loan
## 48   37         12    194      4   0.2         3      211             1
## 79   54         30    133      2   2.6         3        0             1
## 188  46         21    159      3   1.9         3      315             1
## 322  44         20    101      3   4.4         2       82             1
## 325  56         30    158      4   6.1         1        0             1
## 366  57         32    174      1   6.8         2      466             1
##     Securities.Account CD.Account Online CreditCard
## 48                   1          1      1          1
## 79                   0          0      0          0
## 188                  0          0      1          0
## 322                  0          0      0          0
## 325                  0          0      0          0
## 366                  0          0      1          0
```

```
summary(misclassified)
```

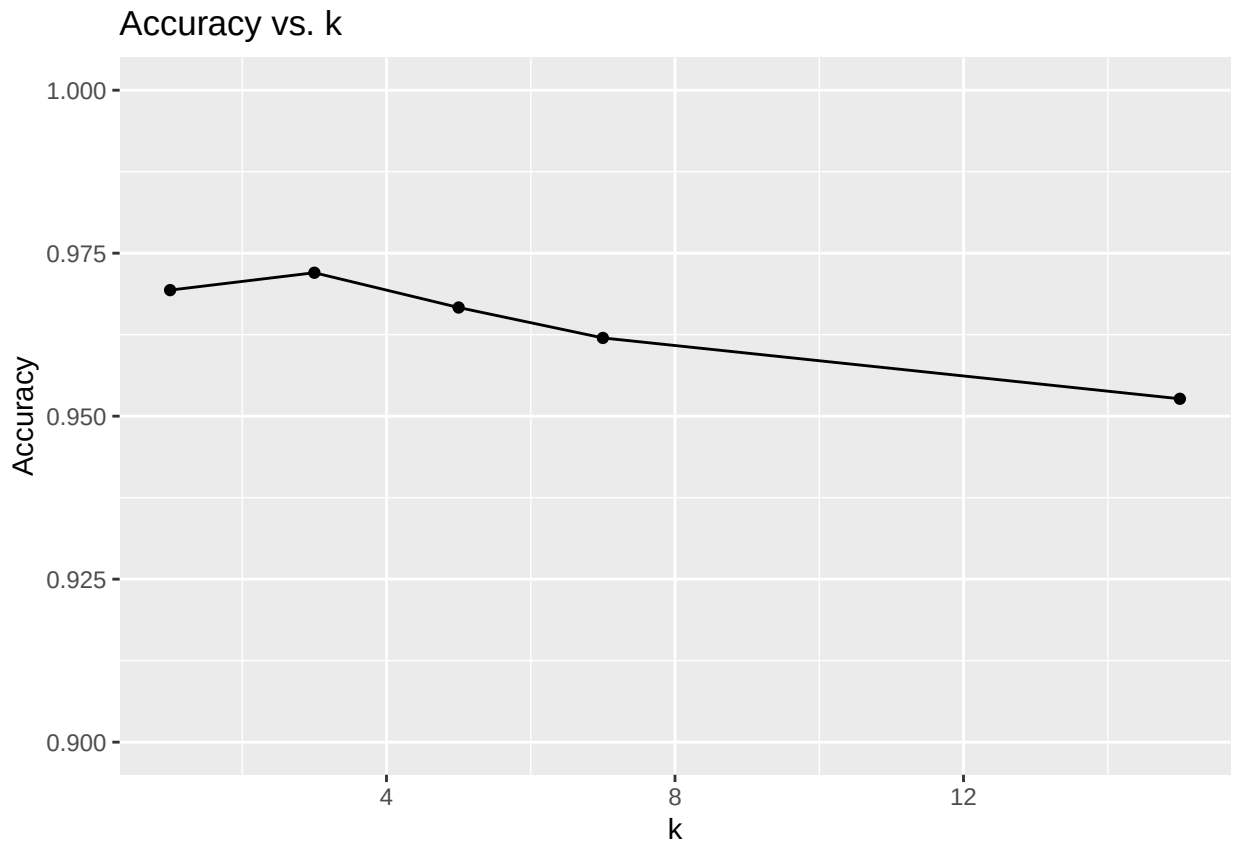
```
##       Age          Experience        Income          Family
##  Min.   :26.0   Min.   : 0.00   Min.   : 21.0   Min.   :1.000
##  1st Qu.:33.0   1st Qu.: 8.00   1st Qu.:115.0   1st Qu.:2.000
##  Median :44.0   Median :20.00   Median :133.0   Median :3.000
##  Mean   :43.9   Mean   :18.73   Mean   :132.3   Mean   :2.577
##  3rd Qu.:54.0   3rd Qu.:30.00   3rd Qu.:151.5   3rd Qu.:3.000
##  Max.   :65.0   Max.   :41.00   Max.   :201.0   Max.   :4.000
##      CCAvg          Education        Mortgage      Personal.Loan
##  Min.   : 0.100   Min.   :1.000   Min.   :  0.00   0: 3
##  1st Qu.: 2.300   1st Qu.:1.000   1st Qu.:  0.00   1:68
##  Median : 3.330   Median :2.000   Median :  0.00
##  Mean   : 3.572   Mean   :1.986   Mean   : 79.93
##  3rd Qu.: 4.800   3rd Qu.:3.000   3rd Qu.:119.50
##  Max.   :10.000   Max.   :3.000   Max.   :466.00
##  Securities.Account   CD.Account         Online         CreditCard
##  Min.   :0.0000     Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000     Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.2254     Mean   :0.2394   Mean   :0.4085   Mean   :0.3662
##  3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000     Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

```r
prop.table(table(misclassified$Personal.Loan))
```

```
##
##          0          1
## 0.04225352 0.95774648
```

```r
library(ggplot2)

ggplot(results, aes(x = k, y = Accuracy)) +
  geom_line() +
  geom_point() +
  ylim(0.9, 1) +
  labs(title = "Accuracy vs. k", x = "k", y = "Accuracy")
```




```r
set.seed(123)
train_index <- createDataPartition(clean_data$Personal.Loan, p = 0.5, list = FALSE)
train_data <- clean_data[train_index, ]
remaining_data <- clean_data[-train_index, ]

val_index <- createDataPartition(remaining_data$Personal.Loan, p = 0.6, list = FALSE)
val_data <- remaining_data[val_index, ]
test_data <- remaining_data[-val_index, ]

normalize <- function(x) {
```

```
  return((x - min(x)) / (max(x) - min(x)))
}

train_X <- as.data.frame(lapply(train_data[, -which(names(train_data) == "Personal.Loan")], normalize))
val_X <- as.data.frame(lapply(val_data[, -which(names(val_data) == "Personal.Loan")], normalize))
test_X <- as.data.frame(lapply(test_data[, -which(names(test_data) == "Personal.Loan")], normalize))

train_Y <- train_data$Personal.Loan
val_Y <- val_data$Personal.Loan
test_Y <- test_data$Personal.Loan

knn_train <- knn(train = train_X, test = train_X, cl = train_Y, k = 3)
knn_val <- knn(train = train_X, test = val_X, cl = train_Y, k = 3)
knn_test <- knn(train = train_X, test = test_X, cl = train_Y, k = 3)

conf_train <- table(Predicted = knn_train, Actual = train_Y)
conf_val <- table(Predicted = knn_val, Actual = val_Y)
conf_test <- table(Predicted = knn_test, Actual = test_Y)

conf_train
```

```
##          Actual
## Predicted    0    1
##         0 2255   54
##         1    5  186
```

```
conf_val
```

```
##          Actual
## Predicted    0    1
##         0 1349   47
##         1    7   97
```

```
conf_test
```

```
##          Actual
## Predicted   0   1
##         0 898  37
##         1   6  59
```