# Building KNN Function

*Nouman Riaz*

*October 25, 2016*

## The objective is to implement KNN at 'Default' dataset from ISLR package.

**First we will load the dataset from ISLR package.**

```
require(ISLR)
```

```
## Loading required package: ISLR
```

```
data("Default")
summary(Default)
```

```
##  default    student       balance           income
##  No :9667   No :7056   Min.   :   0.0   Min.   :  772
##  Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340
##                        Median : 823.6   Median :34553
##                        Mean   : 835.4   Mean   :33517
##                        3rd Qu.:1166.3   3rd Qu.:43808
##                        Max.   :2654.3   Max.   :73554
```

**Now we will replace the labels by 'Yes' as '1' and 'No' as '0'**

```
Default$default <- as.factor(gsub("Yes","1",Default$default))
Default$default <- as.factor(gsub("No","0",Default$default))
```

**To split the dataset as 70% training and 30% testing, we will use 'set.seed' function to keep the reproducibility.**

```
smp_size <- floor(0.70 * nrow(Default))

set.seed(123)
train_ind <- sample(seq_len(nrow(Default)), size = smp_size)

train <- Default[train_ind, ]
test <- Default[-train_ind, ]
```

**We now need to introduce new variables for new labels as per our k-values.**

```
test$nlabel_k7 <- as.factor(c(1,0)) #results for k=7
test$nlabel_k11 <- as.factor(c(1,0)) #results for k=11
test$nlabel_k15 <- as.factor(c(1,0)) #results for k=15
```

**We will now find new labels using KNN technique. That is, first find nearest k-points for the test datapoints and then assign new label as per voting.**

**a) For k=7**

```
for(i in 1:nrow(test))
{
  #First we need to calculate distance for each test data point.
  dist <- sqrt((test$income[i]-train$income)^2+(test$balance[i]-train$balance)^2)

  ind <- which(dist %in% sort(dist)[1:7]) #finding index of k-nearest points
  labels <- train$default[ind] #Getting labels of nearest points
  test$nlabel_k7[i] <- labels[which.max(labels)] #Selecting the top from voting
}

#Now we will calculate error

error_k7 <- sum(as.numeric(test$default!=test$nlabel_k7))/nrow(test)
cat(error_k7)
```

```
## 0.102
```

**b) For k=11**

```
for(i in 1:nrow(test))
{
  #First we need to calculate distance for each test data point.
  dist <- sqrt((test$income[i]-train$income)^2+(test$balance[i]-train$balance)^2)

  ind <- which(dist %in% sort(dist)[1:11]) #finding index of k-nearest points
  labels <- train$default[ind] #Getting labels of nearest points
  test$nlabel_k11[i] <- labels[which.max(labels)] #Selecting the top from voting
}

#Now we will calculate error

error_k11 <- sum(as.numeric(test$default!=test$nlabel_k11))/nrow(test)
cat(error_k11)
```

```
## 0.1266667
```

**c) For k=15**

```
for(i in 1:nrow(test))
{
  #First we need to calculate distance for each test data point.
  dist <- sqrt((test$income[i]-train$income)^2+(test$balance[i]-train$balance)^2)

  ind <- which(dist %in% sort(dist)[1:15]) #finding index of k-nearest points
  labels <- train$default[ind] #Getting labels of nearest points
  test$nlabel_k11[i] <- labels[which.max(labels)] #Selecting the top from voting
}

#Now we will calculate error

error_k15 <- sum(as.numeric(test$default!=test$nlabel_k15))/nrow(test)
cat(error_k15)
```

```
## 0.4956667
```