R Notebook

Step - 1 - collecting the data -

The dataset used in the question is the wine dataset downloaded from Machine Learning repository of UC Irvine. The 13 features are - (dontated by Riccardo Leardi, riclea ‘@’ anchem.unige.it ) 1) Alcohol 2) Malic acid 3) Ash 4) Alcalinity of ash 5) Magnesium 6) Total phenols 7) Flavanoids 8) Nonflavanoid phenols 9) Proanthocyanins 10)Color intensity 11)Hue 12)OD280/OD315 of diluted wines 13)Proline

The data is downloaded from UC Irvine machine learning repository as follows -

dataurl <- "https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data"  
download.file(url = dataurl, destfile = "wine.data")  
wine\_df <- read.csv("wine.data", header = FALSE)

Step - 2 - Exploring and preparing data

The data is saved in wine\_df as seen in Step - I. Next, we will take a look at the structure of the data using str function() as follows -

str(wine\_df)

## 'data.frame': 178 obs. of 14 variables:  
## $ V1 : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ V2 : num 14.2 13.2 13.2 14.4 13.2 ...  
## $ V3 : num 1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...  
## $ V4 : num 2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...  
## $ V5 : num 15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...  
## $ V6 : int 127 100 101 113 118 112 96 121 97 98 ...  
## $ V7 : num 2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...  
## $ V8 : num 3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...  
## $ V9 : num 0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...  
## $ V10: num 2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...  
## $ V11: num 5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...  
## $ V12: num 1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...  
## $ V13: num 3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...  
## $ V14: int 1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...

The attribute with the class label is at index 1. It consists of 3 values 1, 2 & 3. These class labels are going to be predicted by the K-NN model.Since it is nominal variable we will analyze using table() as follows -

table(wine\_df$V1)

##   
## 1 2 3   
## 59 71 48

We can see that we have 59 records for class one, 71 records for classs two and 48 records for class three.

Many R machine learning classifiers require that the target feature is coded as a factor, so we will need to recode the diagnosis variable.

wine\_df$V1 <- factor(wine\_df$V1, levels = c(1,2,3),  
 labels = c("One","Two","Three"))

Next, We would like to display the results in percentages for V1 column. We will do this by multiplying the proportions by 100, then using the round() function as follows -

round(prop.table(table(wine\_df$V1)) \* 100, digits = 1)

##   
## One Two Three   
## 33.1 39.9 27.0

We have 33.1 % records of class one, 39.9% records of class two and 27% records for class three.

To take a closer look at the other features we will use Summary function on the three of these features as follows -

summary(wine\_df[c("V3", "V4", "V14")])

## V3 V4 V14   
## Min. :0.740 Min. :1.360 Min. : 278.0   
## 1st Qu.:1.603 1st Qu.:2.210 1st Qu.: 500.5   
## Median :1.865 Median :2.360 Median : 673.5   
## Mean :2.336 Mean :2.367 Mean : 746.9   
## 3rd Qu.:3.083 3rd Qu.:2.558 3rd Qu.: 985.0   
## Max. :5.800 Max. :3.230 Max. :1680.0

Looking at these results, it is pretty clear we need to apply normalization as V3 and V4 have very small range however V14 ranges from 278 to 1680. As a result, the impact of V14 is going to be much larger than V3 and V4.

The class variable or V1 is in particular order i.e. all the 1s are followed by 2’s which are then followed by 3’s. We need to shuffle the data in random manner to make sure that all the classes are selected randomly and have equal probability of getting selected. The code to do is as follows -

wine\_df <- wine\_df[sample(nrow(wine\_df)),]

#### Transformation - normalizing numeric data

A function named normalize() will be created to normalize the features. This function will take a vector x of numeric values, and for each value in x, it will subtract the minimum value in x and will divide it by the range of values in x and finally it will return the vector. The code for this function is as follows -

normalize <- function(x) {  
return ((x - min(x)) / (max(x) - min(x)))}

The lapply() function takes a list and applies a specified function to each list element. As a data frame is a list of equal-length vectors, we will be using lapply() to apply normalize() to each numerical feature in the data frame. The final step is to convert the list returned by lapply() to a data frame, using the as.data.frame() function. The code to do that is as follows -

wine\_dfn <- as.data.frame(lapply(wine\_df[2:14], normalize))

To verify the transformation is done correctly, we will check summary of the one of the features as follows -

summary(wine\_dfn$V14)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.1587 0.2821 0.3344 0.5043 1.0000

We can see that the numbers are between 0 and 1. Hence the data is said to be normalized.

#### Data Preparation - creating training and test datasets

To see how well our learner performs on a dataset of unlabeled data we will be dividing data into two portions - training dataset that will be used to build k-nn model and a test dataset that will be used to estimate the predictive accuracy of the model. The first 130 records will be used for the training dataset and the remaining 48 to simulate new records

wine\_train <- wine\_dfn[1:130, ]  
wine\_test <- wine\_dfn[131:178, ]

The target variable which is V1, will be stored in factor vectors, split between the training and test datasets as follows -

wine\_train\_labels <- wine\_df[1:130, 1]  
wine\_test\_labels <- wine\_df[131:178, 1]

Step - 3 - training a model on the data

For training and classification using knn() we need to pass four parameters to the function. The first parameter is training dataframe, second parameter is testing dataframe, third parameter is a factor vector with the class for each row in the training data and k is an integer that represent the nearest neighbors. To classify the wine dataset following code is used:

library(class)

## Warning: package 'class' was built under R version 3.4.4

wine\_test\_pred <- knn(train = wine\_train, test = wine\_test, cl = wine\_train\_labels, k = 10)

The above code will return a factor vector of predicted labels for each of the examples in the test dataset, which are being assigned to wine\_test\_pred.

Step - 4 - evaluating model performance

To evaluate how well the predicted classes in the wine\_test\_pred vector matches up with the known values in the wine\_test\_labels\_vector, Crosstable() function is used, with all the unnecessary chi-square values from the output removed as follows :

library(gmodels)

## Warning: package 'gmodels' was built under R version 3.4.4

CrossTable(x = wine\_test\_labels, y = wine\_test\_pred, prop.chisq = FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 48   
##   
##   
## | wine\_test\_pred   
## wine\_test\_labels | One | Two | Three | Row Total |   
## -----------------|-----------|-----------|-----------|-----------|  
## One | 19 | 0 | 0 | 19 |   
## | 1.000 | 0.000 | 0.000 | 0.396 |   
## | 0.905 | 0.000 | 0.000 | |   
## | 0.396 | 0.000 | 0.000 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Two | 2 | 16 | 0 | 18 |   
## | 0.111 | 0.889 | 0.000 | 0.375 |   
## | 0.095 | 1.000 | 0.000 | |   
## | 0.042 | 0.333 | 0.000 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Three | 0 | 0 | 11 | 11 |   
## | 0.000 | 0.000 | 1.000 | 0.229 |   
## | 0.000 | 0.000 | 1.000 | |   
## | 0.000 | 0.000 | 0.229 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Column Total | 21 | 16 | 11 | 48 |   
## | 0.438 | 0.333 | 0.229 | |   
## -----------------|-----------|-----------|-----------|-----------|  
##   
##

From the Crosstable, we can see that other than for class Two, the k-nn model shows pretty good accuracy in predicting the classes one and three. There is only one misclassification for class two. The accuracy rate is 100% for class one and two and 95% for class Two.

Step - 5 - Improving model performance

To improve the performance, first an alternative method for rescaling our numeric features is used. Instead of using normalize(), scale() function will be used. which by default, rescales values using the z-score standardization. Following code is used for that :

wine\_z <- as.data.frame(scale(wine\_df[-1]))

The code will rescale all the features except for V1 and will save the result in wine\_z data frame. To verify the transformation was successfully applied we check one of the features as follows:

summary(wine\_z$V2)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -2.42739 -0.78603 0.06083 0.00000 0.83378 2.25341

We can see that the mean of the standardized variable is 0, which should be the case. The range is also fairly compact.

Next, we wil follow the same steps from above of dividing the data into training and test data sets, and then classify the test instances using the knn() functions. The results same as above will be validated using crosstable() function. The code to do this is as follows:

wine\_train <- wine\_z[1:130, ]  
wine\_test <- wine\_z[131:178, ]  
wine\_train\_labels <- wine\_df[1:130, 1]  
wine\_test\_labels <- wine\_df[131:178, 1]  
wine\_test\_pred <- knn(train = wine\_train, test = wine\_test, cl = wine\_train\_labels, k = 10)  
CrossTable(x = wine\_test\_labels, y = wine\_test\_pred, prop.chisq = FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 48   
##   
##   
## | wine\_test\_pred   
## wine\_test\_labels | One | Two | Three | Row Total |   
## -----------------|-----------|-----------|-----------|-----------|  
## One | 19 | 0 | 0 | 19 |   
## | 1.000 | 0.000 | 0.000 | 0.396 |   
## | 0.950 | 0.000 | 0.000 | |   
## | 0.396 | 0.000 | 0.000 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Two | 1 | 16 | 1 | 18 |   
## | 0.056 | 0.889 | 0.056 | 0.375 |   
## | 0.050 | 1.000 | 0.083 | |   
## | 0.021 | 0.333 | 0.021 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Three | 0 | 0 | 11 | 11 |   
## | 0.000 | 0.000 | 1.000 | 0.229 |   
## | 0.000 | 0.000 | 0.917 | |   
## | 0.000 | 0.000 | 0.229 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Column Total | 20 | 16 | 12 | 48 |   
## | 0.417 | 0.333 | 0.250 | |   
## -----------------|-----------|-----------|-----------|-----------|  
##   
##

We do not see any change in the model after changing the transformation method to z-score transformation to normalize()

Secondly, we try to improve the performance of the model by varying the different values of k as follows:

wine\_train <- wine\_dfn[1:130, ]  
wine\_test <- wine\_dfn[131:178, ]  
  
wine\_test\_pred <- knn(train = wine\_train, test = wine\_test, cl = wine\_train\_labels, k=5)  
CrossTable(x = wine\_test\_labels, y = wine\_test\_pred, prop.chisq=FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 48   
##   
##   
## | wine\_test\_pred   
## wine\_test\_labels | One | Two | Three | Row Total |   
## -----------------|-----------|-----------|-----------|-----------|  
## One | 19 | 0 | 0 | 19 |   
## | 1.000 | 0.000 | 0.000 | 0.396 |   
## | 0.950 | 0.000 | 0.000 | |   
## | 0.396 | 0.000 | 0.000 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Two | 1 | 16 | 1 | 18 |   
## | 0.056 | 0.889 | 0.056 | 0.375 |   
## | 0.050 | 1.000 | 0.083 | |   
## | 0.021 | 0.333 | 0.021 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Three | 0 | 0 | 11 | 11 |   
## | 0.000 | 0.000 | 1.000 | 0.229 |   
## | 0.000 | 0.000 | 0.917 | |   
## | 0.000 | 0.000 | 0.229 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Column Total | 20 | 16 | 12 | 48 |   
## | 0.417 | 0.333 | 0.250 | |   
## -----------------|-----------|-----------|-----------|-----------|  
##   
##

wine\_test\_pred <- knn(train = wine\_train, test = wine\_test, cl = wine\_train\_labels, k=11)  
CrossTable(x = wine\_test\_labels, y = wine\_test\_pred, prop.chisq=FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 48   
##   
##   
## | wine\_test\_pred   
## wine\_test\_labels | One | Two | Three | Row Total |   
## -----------------|-----------|-----------|-----------|-----------|  
## One | 19 | 0 | 0 | 19 |   
## | 1.000 | 0.000 | 0.000 | 0.396 |   
## | 0.950 | 0.000 | 0.000 | |   
## | 0.396 | 0.000 | 0.000 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Two | 1 | 16 | 1 | 18 |   
## | 0.056 | 0.889 | 0.056 | 0.375 |   
## | 0.050 | 1.000 | 0.083 | |   
## | 0.021 | 0.333 | 0.021 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Three | 0 | 0 | 11 | 11 |   
## | 0.000 | 0.000 | 1.000 | 0.229 |   
## | 0.000 | 0.000 | 0.917 | |   
## | 0.000 | 0.000 | 0.229 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Column Total | 20 | 16 | 12 | 48 |   
## | 0.417 | 0.333 | 0.250 | |   
## -----------------|-----------|-----------|-----------|-----------|  
##   
##

wine\_test\_pred <- knn(train = wine\_train, test = wine\_test, cl = wine\_train\_labels, k=15)  
CrossTable(x = wine\_test\_labels, y = wine\_test\_pred, prop.chisq=FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 48   
##   
##   
## | wine\_test\_pred   
## wine\_test\_labels | One | Two | Three | Row Total |   
## -----------------|-----------|-----------|-----------|-----------|  
## One | 19 | 0 | 0 | 19 |   
## | 1.000 | 0.000 | 0.000 | 0.396 |   
## | 0.950 | 0.000 | 0.000 | |   
## | 0.396 | 0.000 | 0.000 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Two | 1 | 17 | 0 | 18 |   
## | 0.056 | 0.944 | 0.000 | 0.375 |   
## | 0.050 | 1.000 | 0.000 | |   
## | 0.021 | 0.354 | 0.000 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Three | 0 | 0 | 11 | 11 |   
## | 0.000 | 0.000 | 1.000 | 0.229 |   
## | 0.000 | 0.000 | 1.000 | |   
## | 0.000 | 0.000 | 0.229 | |   
## -----------------|-----------|-----------|-----------|-----------|  
## Column Total | 20 | 17 | 11 | 48 |   
## | 0.417 | 0.354 | 0.229 | |   
## -----------------|-----------|-----------|-----------|-----------|  
##   
##

From the above results, we can see that the results for k = 5 are little less accurate than k = 11 and k = 15. Overall k-nn model has very high accuracy rate.