K- nearest neighbours implementation in R

In this lecture

- Case study
  - Problem statement
- Solve the case study using R
  - Read the data from a ".csv" file
  - Understand the data
  - knn() function
  - Interpret the results

Key points from previous lecture

- knn is primarily used as a classification algorithm
- It is supervised learning algorithm
  - Data is labelled
- Non-parametric method
- No explicit training phase is involved
- Lazy learning algorithm
- Notion of distance is needed
- Majority voting method ✓

Automotive Service company: a case study

Automotive Service Study: Problem statement

An automotive service chain is launching its new grand service station this weekend. They offer to service a wide variety of cars. The current capacity of the station is to check 315 cars thoroughly per day.

As an inaugural offer, they claim to freely check all cars that arrive on their launch day, and report whether they need servicing or not!

Unexpectedly, they get 450 cars. The service men won't work longer than the working hours but the data analysts have to!

Can you save the day for the new service station?

How can a data scientist save a day for them?

- He has been a data set which contains some attributes of car that can be easily measured and wont require much time and a conclusion that if service is needed for that or not. -"serviceTrainData.csv" ✓
- Now for the cars they cannot check in detail, they measure those attributes- "serviceTestData.csv" ✓
- Use knn classification technique to classify the cars they cannot test manually and say whether service is needed or not .

Solution to case study using R

Getting things ready

- Setting working directory, clearing variables in the workspace
- Installing or loading required packages

```
##### knn Implementation in R #########
# Set the working directory as the directory which contains the
data files
# setwd("Path of the directory with data files")
rm(list=ls()) # to clear the environment
# install.packages("caret",dependencies = TRUE)
# install.packages("class",dependencies = TRUE)
library(caret) # for confusionMatrix
library(class) # for knn
```

Reading the data

- Data for this case study is provided to you in files with names "serviceTrainData.csv ", "serviceTestData.csv"
- To read the data from a ".csv" file we use `read.csv()` function

read.csv()

Reads a file in table format and creates a data frame from it

SYNTAX

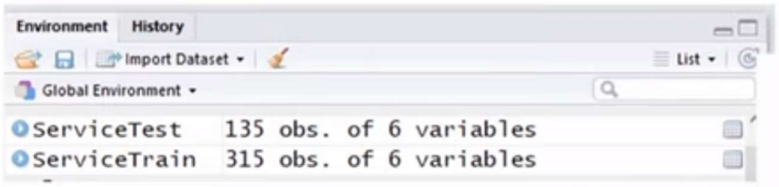
```
read.csv(file,row.names)
```

file	the name of the file which the data are to be read from. Each row of the table appears as one line of the file.
row.names	a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names.

Reading the data

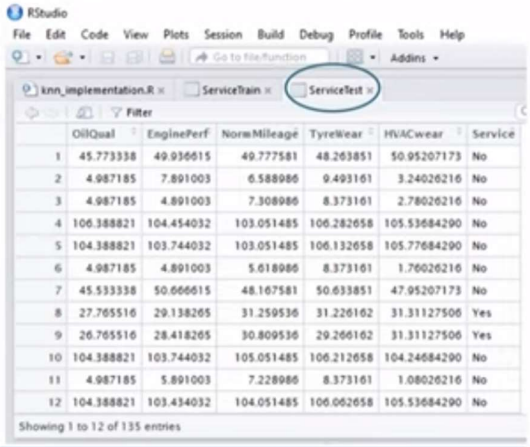
- Data for this case study is provided to you in files with names "serviceTrainData.csv ", "serviceTestData.csv"

```
#Reading the data
ServiceTrain <- read.csv("serviceTrainData.csv")
ServiceTest <- read.csv("serviceTestData.csv")
```



Viewing the data

```
View(ServiceTest)
```



	OilQual	EnginePerf	NormMileage	TyreWear	HVACwear	Service
1	45.773338	49.936615	49.777581	48.263851	50.95207173	No
2	4.987185	7.891003	6.588986	9.493161	3.24026216	No
3	4.987185	4.891003	7.308986	8.373161	2.78026216	No
4	106.388821	104.454032	103.051485	106.282658	105.53684290	No
5	104.388821	103.744032	103.051485	106.132658	105.77684290	No
6	4.987185	4.891003	5.618986	8.373161	1.76026216	No
7	45.533338	50.666615	48.167581	50.633851	47.95207173	No
8	27.765516	29.138265	31.259536	31.226162	31.31127506	Yes
9	26.765516	28.418265	30.809536	29.266162	31.31127506	Yes
10	104.388821	103.744032	105.051485	106.212658	104.24684290	No
11	4.987185	5.891003	7.228986	8.373161	1.08026216	No
12	104.388821	103.434032	104.051485	106.062658	105.53684290	No

Showing 1 to 12 of 135 entries

Understanding the data

- ServiceTrain contains 315 observations of 6 variables
- ServiceTest contains 135 observations of 6 variables
- The variables are: OilQual, Engineperf, NormMileage, TyreWear, HVACwear and Service
  - First five columns are the details about the car and last column is the label which says whether a service is needed or not

Structure of the data

- Structure of data
  - Variables and their data types
- `str()`

Compactly display the internal structure of an R object

SYNTAX

```
str(object)
```

object	any R object about which you want to have some information.

Structure of ServiceTrain

```
> str(ServiceTrain)
'data.frame':   315 obs. of  6 variables:
 $ OilQual    : num  103.4 26.8 62.4 45.5 104.4 ...
 $ EnginePerf : num  103.5 26.2 63.7 49.9 103.3 ...
 $ NormMileage: num  103.1 31.3 59.7 48.8 103.1 ...
 $ TyreWear   : num  106.2 29.2 64.7 48.1 105.8 ...
 $ HVACwear   : num  105.7 31.3 58.6 48 106.5 ...
 $ Service    : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 1 1 1
```

Structure of ServiceTest

```
> str(ServiceTest)
'data.frame':   135 obs. of  6 variables:
 $ OilQual    : num  45.77 4.99 4.99 106.39 104.39 ...
 $ EnginePerf : num  49.94 7.89 4.89 104.45 103.74 ...
 $ NormMileage: num  49.78 6.59 7.31 103.05 103.05 ...
 $ TyreWear   : num  48.26 9.49 8.37 106.28 106.13 ...
 $ HVACwear   : num  50.95 3.24 2.78 105.54 105.78 ...
 $ Service    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 2
```

Summary of the data

- Summary of data
  - The function invokes particular methods which depend on the class of the first argument.
- summary()

Summary gives a 5 point summary for numeric attributes in the data

SYNTAX

```
summary(object)
```

object	any R object about which you want to have some information.

Summary of ServiceTrain

```
> summary(ServiceTrain)
    OilQual          EnginePerf
 Min.   :  0.9872   Min.   :  1.891
 1st Qu.: 26.7655   1st Qu.: 27.418
 Median : 59.6633   Median : 59.741
 Mean   : 59.6493   Mean   : 60.306
 3rd Qu.:104.3888   3rd Qu.:103.744
 Max.   :106.4288   Max.   :105.744
  NormMileage        TyreWear
 Min.   :  3.359   Min.   :  6.213
 1st Qu.: 31.260   1st Qu.: 29.036
 Median : 57.221   Median : 60.304
 Mean   : 60.297   Mean   : 61.759
 3rd Qu.:103.051   3rd Qu.:106.173
 Max.   :105.051   Max.   :108.173
    HVACwear      Service
 Min.   : -1.72   No :232
 1st Qu.: 31.34   Yes: 83
 Median : 60.62
 Mean   : 60.39
 3rd Qu.:105.54
 Max.   :107.54
```

Summary of ServiceTest

```
> summary(ServiceTest)
    OilQual          EnginePerf
 Min.   :  2.597   Min.   :  1.891
 1st Qu.: 26.696   1st Qu.: 27.418
 Median : 61.023   Median : 61.501
 Mean   : 58.629   Mean   : 59.077
 3rd Qu.:104.229   3rd Qu.:103.744
 Max.   :106.389   Max.   :105.744
  NormMileage        TyreWear
 Min.   :  3.589   Min.   :  6.143
 1st Qu.: 31.260   1st Qu.: 28.901
 Median : 59.351   Median : 61.304
 Mean   : 59.118   Mean   : 60.864
 3rd Qu.:103.051   3rd Qu.:106.173
 Max.   :105.051   Max.   :108.173
    HVACwear       Service
 Min.   : -1.72   No :99
 1st Qu.: 31.31   Yes:36
 Median : 62.62
 Mean   : 58.99
 3rd Qu.:105.33
 Max.   :105.83
```

Implementation of k-nearest neighbours : knn()

```
knn(train, test, cl, k = 1)
```

Arguments

train	matrix or data frame of training set cases.
test	matrix or data frame of test set cases. A vector will be interpreted as a row vector for a single case.
cl	factor of true classifications of training set
k	number of neighbours considered.

Applying knn algorithm on data

```
# Applying k-NN algorithm
# K Nearest neighbour is a lazy algorithm and can do prediction directly with the testing
dataset, command "knn", accepts training and testing datasets the class variable of interest
i.e outcome categorical variable is provided for the parameter "cl". parameter "k" is to
specify the number of nearest neighbours required.

predictedknn <- knn(train = ServiceTrain[,-6],
                    test = ServiceTest[,-6],
                    cl = ServiceTrain$Service,
                    k  = 3)
```

- ServiceTrain[,-6] gives information in ServiceTrain except the last column
- ServiceTest[,-6] gives information in ServiceTestexcept the last column
- ServiceTrain$Service gives the last column of training data as a classification factor to the algorithm

Results: predicted classes

- "predictedknn" is the output from the algorithm, which has a categorical variable "Yes" or "No", indicating whether service is needed or not for each case in Test data

```
> # printing the information in predictedknn
> predictedknn
  [1] No  No  No  No  No  No  No  Yes Yes No  No
 [12] No  No  No  No  No  Yes No  No  Yes Yes No
 [23] Yes No  No  No  No  No  No  No  No  No  No
 [34] No  No  Yes No  No  No  No  No  Yes No  Yes
 [45] No  No  No  Yes Yes No  Yes No  Yes No  No
 [56] No  No  No  No  No  No  No  No  No  Yes Yes
 [67] Yes No  Yes No  No  Yes No  No  No  No  No
 [78] No  Yes Yes Yes Yes No  Yes No  No  Yes Yes
 [89] Yes No  No  No  Yes No  Yes No  No  No  No
[100] No  No  No  No  No  No  Yes No  No  No  No
[111] No  Yes No  Yes No  Yes Yes No  Yes No  No
[122] No  No  No  Yes No  No  No  No  No  No  No
[133] Yes No  No
Levels: No Yes
```

Results: generating confusion matrix manually

```
# Command to develop and print a confusion matrix
conf_matrix = table(predictedknn,ServiceTest[,6])

predictedknn No Yes
         No  99   0
         Yes  0  36
```

```
# Ameasure of accuracy is calculated by summing the true positives and true negatives and dividing them by total number of samples
knn_accuracy = sum(diag(conf_matrix))/nrow(ServiceTest)

> knn_accuracy
[1] 1
```

Results

```
# confusionMatrix command shown below used from caret package

COnF_Matrix <-confusionMatrix(data = predictedknn,ServiceTest$Service)
```

```
> COnF_Matrix

Confusion Matrix and Statistics

          Reference
Prediction No Yes
       No  99   0
       Yes  0  36

               Accuracy : 1
                 95% CI : (0.973, 1)
    No Information Rate : 0.7333
    P-Value [Acc > NIR] : < 2.2e-16
```

Results

```
# confusionMatrix command shown below used from caret package

COnF_Matrix <-confusionMatrix(data = predictedknn,ServiceTest$Service)
```

```
 Kappa : 1
 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.7333
         Detection Rate : 0.7333
   Detection Prevalence : 0.7333
      Balanced Accuracy : 1.0000

       'Positive' Class : No
```

Conclusion

- read.csv() can be used to read data from .csv files
- str() function gives data types of each attribute in the given R-object
- summary() provides a summary of R-objects
- K-nearest neighbors is supervised learning technique –needs labelled data
- In R knn algorithm can be implemented using knn()