DataVizA Tutorial: k Nearest Neighbour Classification: Solutions

Department of Econometrics and Business Statistics, Monash University

Tutorial 9

Wine Data

These questions are based on the problem from last weeks tutorial

1. Carry out kNN classification using all data in *ExistingWines.rds* in the training set and predict the data in *NewWines.rds*. Let $k = 1$

```
#Use tidyverse
library(dplyr)
library(class)
#Load in data
ExistingWines<-readRDS('ExistingWines.rds')
#Split x and y variables and standardise
old_x<-select(ExistingWines,-BestMarket)%>%scale()

#Keep mean and standard deviation
mean_old_x<-attr(old_x,"scaled:center")
std_old_x<-attr(old_x,"scaled:scale")

old_y<-pull(ExistingWines,BestMarket)

#Load in New Wine Data

new_x<-readRDS('NewWines.rds')%>%
  scale(center = mean_old_x,scale = std_old_x)


yhat_k1<-knn(old_x,new_x,old_y,k=1)
```

2. What are the predictions for the first ten wines in the *NewWines.rds*

```
#The first ten predictions can be checked using the head function

head(yhat_k1,10)
```

```
##  [1] Australia Australia Australia Australia Australia Australia Australia
##  [8] Australia Australia Australia
## Levels: Australia Europe Japan
```

3. Repeat the analyis with $k = 5$

```
#Same code as before different k argument
yhat_k5<-knn(old_x,new_x,old_y,k=5)
head(yhat_k5,10)
```

```
##  [1] Australia Australia Australia Australia Australia Australia Australia
##  [8] Australia Australia Australia
## Levels: Australia Europe Japan
```

4. What are the predicted probabilities for the first ten wines in the *NewWines.rds* when k=5

```r
#Same as before but set prob to T
yhat_k5<-knn(old_x,new_x,old_y,k=5,prob = T)
#Use attr function to get probabilities
head(attr(yhat_k5,"prob"),10)
```

```
##  [1] 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.8 1.0 1.0
```

```r
#Notice that only the probability of the highest probabilty class is available.
```

5. Split the data in *ExistingWines.rds* into a training sample (of roughly 70%) and a test sample (of roughly 30%). Hint `runif(125)<0.7` will create a vector of length 125 where each element is either TRUE with probability 70% and false with probability 30%. You may also want to use the `ifelse` function

```r
#Create an indicator that determines whether it is training or test sample.
ind<-ifelse(runif(125)<0.7,"Training Sample","Test Sample")

train_y<-old_y[ind=="Training Sample"]
test_y<-old_y[ind=="Test Sample"]

train_x<-old_x[ind=="Training Sample",]
test_x<-old_x[ind=="Test Sample",]
```

6. Is $k = 1$ a better choice than $k = 5$ according to the misclassification rate?

```r
yhat_k1<-knn(train_x,test_x,train_y,k=1)
mean(yhat_k1!=test_y)
```

```
## [1] 0.1025641
```

```r
yhat_k5<-knn(train_x,test_x,train_y,k=5)
mean(yhat_k5!=test_y)
```

```
## [1] 0.07692308
```

```r
#For this particular example k=5 has a lower missclassification rate
```