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
old_x<-select(ExistingWines,-BestMarket)
old_y<-pull(ExistingWines,BestMarket)

#Load in New Wine Data
new_x<-readRDS('NewWines.rds')
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 Japan     Australia
##  [8] Japan     Europe    Australia
## Levels: Australia Europe Japan
```

3. Repeat the analyis with $k = 11$

```
#Same code as before different k argument
yhat_k11<-knn(old_x,new_x,old_y,k=11)
head(yhat_k11,10)
```

```
##  [1] Australia Australia Australia Australia Australia Japan     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=11

```
#Same as before but set prob to T
yhat_k11<-knn(old_x,new_x,old_y,k=11,prob = T)
#Use attr function to get probabilities
head(attr(yhat_k11,"prob"),10)
```

```
##  [1] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.4545455 0.9090909
##  [8] 0.4545455 0.4545455 0.8181818
```

```
#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

```
#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 $k = 11$ according to the misclassification rate?

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

```
## [1] 0.2307692
```

```
yhat_k11<-knn(train_x,test_x,train_y,k=11)
mean(yhat_k11!=test_y)
```

```
## [1] 0.3333333
```

```
#For this particular example k=1 has a lower missclassification rate
```