Lab session: Knn

Dr. Orsdemir

Instructions:

- Data files can be found at this link

Questions:

1. Universities often rely on a high school student's GPA and scores on the SAT or ACT for the college admission decisions. Consider the data for 120 applicants on college admissions (Admit equals 1 if admitted, 0 otherwise) along with the student's GPA and SAT scores.
- Perform KNN analysis to estimate a classification model for college admission decisions in the Admit_Data worksheet. What is the optimal value of k if you want to maximize accuracy? Use k-fold cross-validation with 10 folds. Test the number of neighbors from 1 to 15.
- Predict the outcome of new applications in the Admit_Score worksheet.

Guide

- Use caret package to build a KNN model.

```
# load caret library
library(???)

# read data into R
dta_admit <- read.csv(???)

# which cross-validation method
# you will use and with how many
# folds. Indicate them here.
myCtrl <- trainControl(method=???,
                       number=???)

# We need to create a dataframe to
# tell train function which k values
# we want to test. The line below will
# create a dataframe with column name .k
```

```
# and values changing from 1 to 15
myGrid <- expand.grid(.k=c(1:???))
# You can also do
# myGrid <- data.frame(.k=c(1:???))
# Both will work.

set.seed(1) # for reproducibility

# Build the model here using
# train function
knn.mod <- train(as.factor(???)~.,
                 data=???,
                 method=???,
                 trControl=???,
                 tuneGrid=???,
                 preProc=c(???,???))
# Best k in terms of accuracy
# is given in the output below
knn.mod
# When you use knn.mod to
# predict the new observations
# R will use the best k automatically.
```

- Classify the observations in the new data.

```
dta_admit_score <- read.csv(???)


# you do not need to scale the data (i.e., standardize)
# because it will be done automatically in this workflow.
# In particular, knn.mod holds this information and
# passes it to
# predict. As a result
# Predict knows that newdata must be scaled.
# And scales (standardizes) it automatically

knn_prob <- predict(knn.mod,
                    newdata = ???,
                    type = ???)
# when the model is fitted with caret
# type= "prob" not "response" as in glm!!

# predictions
knn_class <- ifelse(???>???,???,???)
```

```
knn_class
```

2. **Predicting Boston Housing Prices.** The file boston_housing.csv contains information collected by the U.S. Bureau of the Census concerning housing in the area of Boston, Massachusetts. The data set includes information on 506 census housing tracts in the Boston area. The goal is to predict the median house price in new tracts based on information such as crime rate,pollution and number of rooms. The data set contains 12 predictors, and the response is the median house price (MEDV). (Hint: Note that your outcome variable is numerical not categorical. KNN works with both types of outcome variables.)

A. Partition the data into two sets by allocating 90% of data to one and the rest to the other. The first set will serve as training and validation sets (90% of data, partition 1) and the second set will serve as the test set (10% of data, partition 2).

```
dta_boston <- read.csv(???)
head(dta_boston)

boston_train_valid_index <- sample(???,
                                   size=???*0.90,
                                   replace=F)

boston_train_valid <- dta_boston[???,]
boston_test <- dta_boston[???,]
```

B. Fit a KNN model using K-fold cross validation with 10 folds to partition 1. Do not forget to standardize the variables.

```
library(caret)
myCtrl <- trainControl(method=???,
                       number=???)

myGrid <- expand.grid(.k=c(1:15))

set.seed(1)

knn_boston <- train(as.numeric(???)~., #outcome is as.numeric!!
                    data=???,
                    method=???,
                    trControl=???,
                    tuneGrid=???,
                    preProc=???)
# Best k in terms of RMSE
# note that because the outcome is
# numerical. The accuracy metric is
# RMSE
```

```
knn_boston
```

C. Using this model, predict the MEDV in partition 2. What is the RMSE?

```
# Predict the values in the test set.

predicted_prices <- predict(???,
        newdata=???,
        type='raw') # We are predicting a numerical variable

forecast::accuracy(???,boston_test$MEDV)
```

3. Law enforcement agencies monitor social media sites on a regular basis, as a way to identify and assess potential crimes and terrorism activities. For example, certain keywords on Facebook pages are tracked, and the data are compiled into a data mining model to determine whether or not the Facebook page is a potential threat. Officer Matthew Osorio is assigned to explore data mining techniques that can be used for this purpose. He starts by experimenting with KNN algorithms to monitor and assess social media sites with war-related terms as well as suspicious keywords. He collects a data set with 300 observations, a portion of which is shown in the accompanying table. Each record in the data set includes the following variables: Threat (1 if yes, 0 otherwise), the number of suspicious words (WarTerms and Keywords), and the number of hyperlinks to or mentioning of suspicious sites. (Hint: The workflow is same as the first question.)

- Perform KNN analysis on the data set. What is the optimal value of k?
- What is the misclassification rate for the optimal k?