STA6714-19Spring 0001-Homework for Chapter 7

Mithun Mohanraj

7.3 Predicting Housing Median Prices. The file BostonHousing.csv contains information on over 500 census tracts in Boston, where for each tract multiple variables are recorded. The last column (CAT.MEDV) was derived from MEDV, such that it obtains the value 1 if MEDV > 30 and 0 otherwise. Consider the goal of predicting the median value (MEDV) of a tract, given the information in the first 12 columns.Partition the data into training (60%) and validation (40%) sets.

Data pre-processing:

```
#removing the cat.medv variable

Boston=read.csv("BostonHousing.csv")

ndata <- Boston[c(1:13)]


#splitting the data into training and validation

smp_size <- floor(0.6* nrow(ndata))

train_idx <- sample(seq_len(nrow(ndata)), size = smp_size)

trX_boston = ndata[train_idx, c(1:12)]

teX_boston = ndata[-train_idx,c(1:12) ]

trY_boston = ndata[train_idx, "MEDV"]

teY_boston = ndata[-train_idx,"MEDV" ]


#scaling the predictor variables using apply function

trX_boston=as.data.frame(apply(trX_boston[,], 2, function(x) (x -
min(x))/(max(x)-min(x))))

teX_boston=as.data.frame(apply(teX_boston[,], 2, function(x) (x -
min(x))/(max(x)-min(x))))
```

a, Perform a k-NN prediction with all 12 predictors (ignore the CAT.MEDV column), trying values of k from 1 to 5. Make sure to normalize the data, and choose function knn() from package class rather than package FNN. To make sure R is using the class package (when both packages are loaded), use class::knn(). What is the best k? What does it mean?

R code for knn prediction:

```
library(FNN)

pred <- knn.reg(trX_boston, teX_boston, trY_boston, k = 3)$pred

actual = teY_boston

predicted = pred

rmse = function(actual, predicted) {

sqrt(mean((actual - predicted) ^ 2))

}

rmsvalues=rmse(actual = actual,predicted = pred)

rmsvalues
```

Results:

RMSE values for k =1 to k=5

```
For k =1 , RMSE =  5.805845
For k =2 , RMSE =  4.586361
For k =3 , RMSE =  4.204205
For k =4 , RMSE =  4.208656
For k =5 , RMSE =  4.606749
```

Best K:
From the results , it is clear that RMSE decreases when k increases until k =3 and increases again .Initially , model was over-fitting and the best fit is when k =3

The optimal k value for the data is 3.

B, Predict the MEDV for a tract with the following information, using the best k:

best k =3 is used for prediction

new data has to be normalized before performing predictions

R code for prediction:

```
library(caret)

norm.values <- preProcess(treX_boston[,], method=c("center", "scale"))
newdf <- data.frame(
CRIM=0.2,
ZN=0,INDUS=7,CHAS=0,NOX=0.538,RM=6,AGE=62,DIS=4.7,RAD=4,TAX=307,PT
RATIO=21,LSTAT=10)
newdf<- predict(norm.values, newdf)
predicted<- FNN::knn.reg(trX_boston, newdf, trY_boston, k = 3)$pred
predicted
```

result:

the predicted MEDV value for the data is `21.16667`

C,If we used the above k-NN algorithm to score the training data, what would be the error of the training set?

To score the training data , we just make predictions on training data and calculate the training RMSE

Using the best k =3 ,
R code :

```
pred <- knn.reg(trX_boston, trX_boston, trY_boston, k = 3)$pred
actual = trY_boston
predicted = pred
rmse = function(actual, predicted) {
sqrt(mean((actual - predicted) ^ 2))
}
rmsvalues=rmse(actual = actual,predicted = pred)
rmsvalues
```

Result :

Training RMSE is obtained as 3.051922

D,Why is the validation data error overly optimistic compared to the error rate when applying this k-NN predictor to new data?

The best model we chose is based on the performance of model on validation data . the validation data error is overly optimistic compared to the error rate we get when this model is applied on the new data for prediction.

E,If the purpose is to predict MEDV for several thousands of new tracts, what would be the disadvantage of using k-NN prediction? List the operations that the algorithm goes through in order to produce each prediction.

The main disadvantage of using k-nn prediction for large data is that it takes a lot of time to give out the prediction results because for each new data ,we have to compute its distance from the entire training data and find predictions based on its neighbours.

KNN PREDICTION ALGORITHM:

1,For each prediction , compute the distance between the new data and trained data

2, find k nearest neighbours based on the distances , in the order of increasing distance value.

3, calculate the weighted average response value of all neighbours as the predicted value for the new data , where nearest neighbours are given more weightage.