STAT 6340 Mini Project 1 Report

Randy Suarez Rodes

2/3/2021

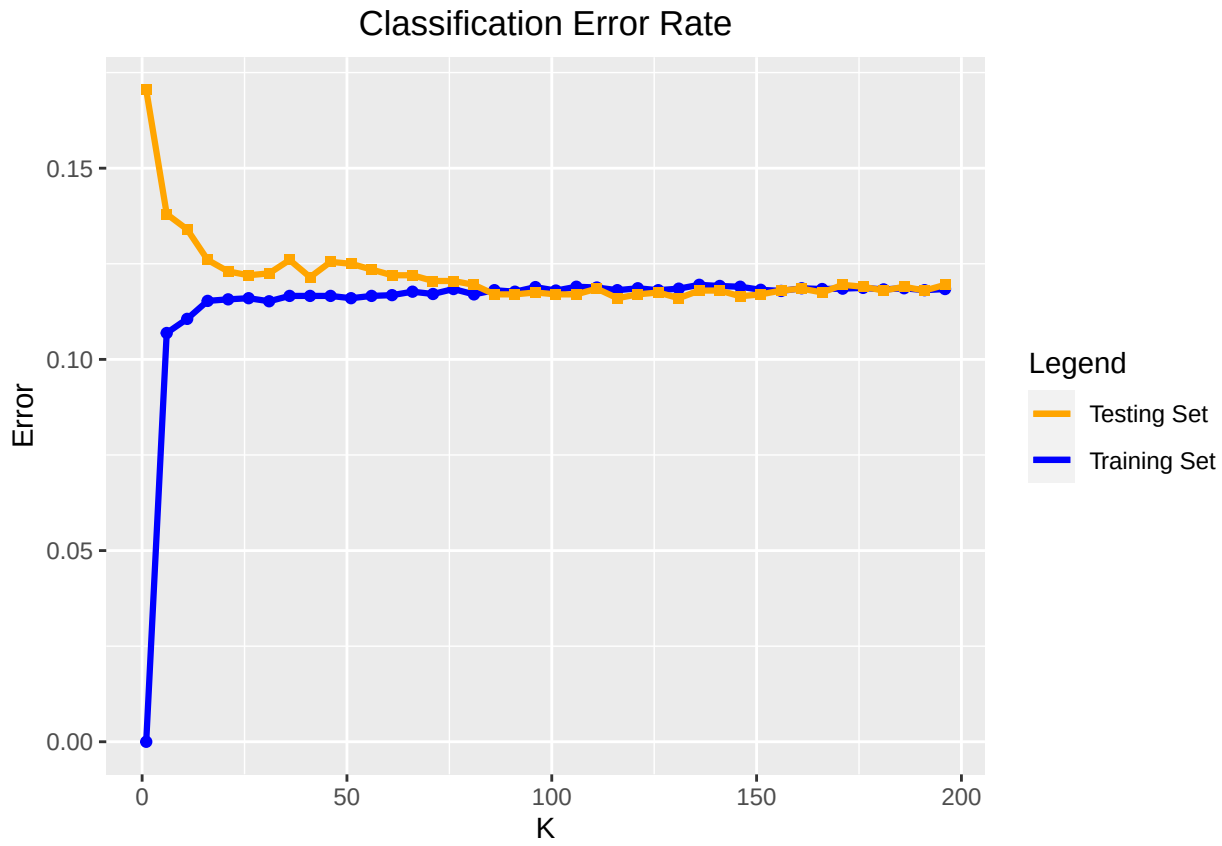
1 Answers

Experiment 1

Question 1.a

We have fit KNN for both training and testing set for 40 values of k ranging between 1, 5, . . . ,196.(See Code section).

Question 1.b



Question 1.c

Question 1.d

Experiment 2

Question 2.a

Question 2.b

Question 2.c

2 Code

```
library(ggplot2) #Used for graphics an visual representations
library(class) #Used for KNN models
library(caret) #Used for confusion matrix

classification_error_rate=function(ypred,ytrue)
{
  mean(ypred!=ytrue)
}

set.seed(8467) #Fixing a seed to replicate results in case of a tie on KNN

graph_colors=c("blue","orange")
graph_legend=c("Training Set","Testing Set")


#Experiment 1

#Value of k for experiment 1
topK=200
kvals=seq(1,topK,5)


#Reading training and testing data set
trn=read.csv("1-training_data.csv", stringsAsFactors = TRUE)
tst=read.csv("1-test_data.csv", stringsAsFactors = TRUE)

#Saving training and testing labels
trn_y=trn$y
tst_y=tst$y

#Dropping the classes to use the training and data sets on the knn function.
trn$y=NULL
tst$y=NULL

#Dataframe to track the errors
Error_df=data.frame(k=kvals,k_rate=1/kvals,trn_Error=kvals,tst_Error=kvals)

#Question 1.a
for(i in 1:length(kvals))
{
  #Fitting KNN for training data
  trn_pred=knn(trn,trn,cl=trn_y,k=kvals[i])
  Error_df$trn_Error[i]=classification_error_rate(trn_pred,trn_y)

  #Fitting KNN for testing data
  tst_pred=knn(trn,tst,cl=trn_y,k=kvals[i])
  Error_df$tst_Error[i]=classification_error_rate(tst_pred,tst_y)

}

#Question 1.b
```

```
g=ggplot(data=Error_df, aes(x=k,y=trn_Error))+
  geom_line(aes(y=trn_Error,color=graph_legend[1]),size=1.1)+
  geom_point(color="blue",shape=19)+
  geom_line(aes(y=tst_Error,color=graph_legend[2]),size=1.1)+
  geom_point(x=Error_df$k,y=Error_df$tst_Error,color="orange",shape=15)+
  scale_color_manual("Legend",values = c("Training Set"="blue","Testing Set"="orange"))+
  labs(title="Classification Error Rate",x="K",y="Error")+
  theme(plot.title=element_text(hjust=0.5))
print(g)


g1=ggplot(data=Error_df, aes(x=k_rate,y=trn_Error))+
  geom_line(aes(y=trn_Error,color=graph_legend[1]),size=1.1)+
  geom_point(color="blue",shape=19)+
  geom_line(aes(y=tst_Error,color=graph_legend[2]),size=1.1)+
  geom_point(x=Error_df$k,y=Error_df$tst_Error,color="orange",shape=15)+
  scale_color_manual("Legend",values = c("Training Set"="blue","Testing Set"="orange"))+
  labs(title="Classification Error Rate",x="1/K",y="Error")+
  theme(plot.title=element_text(hjust=0.5))
print(g1)


#Question 1.c

ind_optimalK=which.min(Error_df$tst_Error)

Error_df[ind_optimalK,]
optimalK=Error_df$k[ind_optimalK]

#Question 1.d

x1=seq(min(trn[,1]),max(trn[,1]),length.out=100)
x2=seq(min(trn[,1]),max(trn[,1]),length.out=100)
grid <- expand.grid(x=x1, y=x2)

bestK=knn(trn,grid,cl=trn_y,k=optimalK,prob = TRUE )
prob <- attr(bestK, "prob")
prob = ifelse(bestK=="yes", prob, 1-prob)
prob_matrix = matrix(prob, length(x1), length(x2))

#plot(trn, pch="o", cex=1.2, col=ifelse(bestK=="yes", "blue", "orange"))
plot(trn, col=ifelse(trn_y=="yes", "blue", "orange"),main=paste("Desicion boundary for Training data K=
contour(x1,x2,prob_matrix,levels=0.5, labels="", xlab="", ylab="", lwd=2, add = TRUE)


contour_df=data.frame(x1=x1,x2=x2,prob_matrix=prob_matrix)

trn$col=factor(ifelse(trn_y=="yes",graph_colors[1], graph_colors[2]))

g3=ggplot(trn,aes(x=x.1,y=x.2,color=col))+geom_point()+geom_contour()
```