knn_CV.R

Magilan

Mon Oct 08 16:27:47 2018

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(sp)
library(class)

# Data Input

data <- read.csv("C:/Users/Magilan/Desktop/ML_project/austin_weather.csv",header = TRUE)
data1=na.omit(data,invert=FALSE)
attach(data1)

# Scalling the Data

standardized.X=scale(data1[,-c(1,20,21,22)])

# Data Partitioning

index <- createDataPartition(Rain, p = 0.7, list = FALSE)
train.X=standardized.X[index,]
test.X=standardized.X[-index,]
train.Y=Rain[index]
test.Y=Rain[-index]

# Knn Model

knn.pred=knn(train.X,test.X,train.Y,k=1)
head(data.frame(knn.pred,test.Y))
```

```
##   knn.pred test.Y
## 1      yes    yes
## 2       no     no
## 3       no    yes
## 4       no     no
## 5       no     no
## 6      yes    yes
```

```
confusionMatrix(knn.pred,test.Y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  227  58
##        yes  30  75
##
##                Accuracy : 0.7744
##                  95% CI : (0.7296, 0.8149)
##     No Information Rate : 0.659
##     P-Value [Acc > NIR] : 4.51e-07
##
##                   Kappa : 0.4711
##  Mcnemar's Test P-Value : 0.003999
##
##             Sensitivity : 0.8833
##             Specificity : 0.5639
##          Pos Pred Value : 0.7965
##          Neg Pred Value : 0.7143
##              Prevalence : 0.6590
##          Detection Rate : 0.5821
##    Detection Prevalence : 0.7308
##       Balanced Accuracy : 0.7236
##
##        'Positive' Class : no
##
```

```
knn.pred1=knn(train.X,test.X,train.Y,k=2)
confusionMatrix(knn.pred1,test.Y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  221  53
##        yes  36  80
##
##                Accuracy : 0.7718
##                  95% CI : (0.7269, 0.8125)
##     No Information Rate : 0.659
##     P-Value [Acc > NIR] : 8.062e-07
##
##                   Kappa : 0.4761
##  Mcnemar's Test P-Value : 0.08989
##
##             Sensitivity : 0.8599
##             Specificity : 0.6015
##          Pos Pred Value : 0.8066
##          Neg Pred Value : 0.6897
##              Prevalence : 0.6590
##          Detection Rate : 0.5667
##    Detection Prevalence : 0.7026
##       Balanced Accuracy : 0.7307
##
##        'Positive' Class : no
##
```

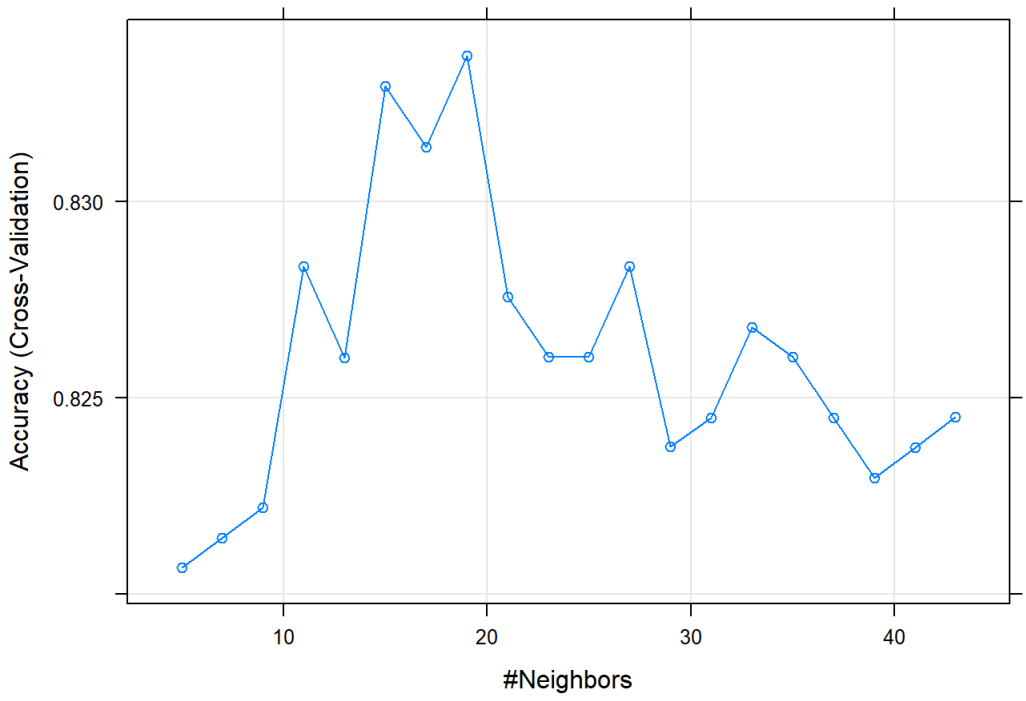
```
knn.pred2=knn(train.X,test.X,train.Y,k=100)
confusionMatrix(knn.pred2,test.Y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  238  66
##        yes  19  67
##
##                Accuracy : 0.7821
##                  95% CI : (0.7377, 0.822)
##     No Information Rate : 0.659
##     P-Value [Acc > NIR] : 7.248e-08
##
##                   Kappa : 0.4699
##  Mcnemar's Test P-Value : 6.057e-07
##
##             Sensitivity : 0.9261
##             Specificity : 0.5038
##          Pos Pred Value : 0.7829
##          Neg Pred Value : 0.7791
##              Prevalence : 0.6590
##          Detection Rate : 0.6103
##    Detection Prevalence : 0.7795
##       Balanced Accuracy : 0.7149
##
##        'Positive' Class : no
##
```

```
# Cross Validation to find the value of K with highest Accuracy

tr=cbind(standardized.X,Rain)

model <- train(
  Rain ~., data = data1[,-c(1,20,22)], method = "knn",
  trControl = trainControl("cv", number = 10),
  preProcess = c("center","scale"),
  tuneLength = 20
)
plot(model)
```




```
k=model$bestTune
k
```

```
##    k
## 8 19
```

```
knn.pred3=knn(train.X,test.X,train.Y,k= model$bestTune)
confusionMatrix(knn.pred3,test.Y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  237  54
##        yes  20  79
##
##                Accuracy : 0.8103
##                  95% CI : (0.7678, 0.848)
##     No Information Rate : 0.659
##     P-Value [Acc > NIR] : 2.799e-11
##
##                   Kappa : 0.5501
##  Mcnemar's Test P-Value : 0.000125
##
##             Sensitivity : 0.9222
##             Specificity : 0.5940
##          Pos Pred Value : 0.8144
##          Neg Pred Value : 0.7980
##              Prevalence : 0.6590
##          Detection Rate : 0.6077
##    Detection Prevalence : 0.7462
##       Balanced Accuracy : 0.7581
##
##        'Positive' Class : no
##
```