Iris\_KNN

Nabil Momin

2024-06-10

library(corrgram)  
library(corrplot)

## corrplot 0.92 loaded

library(caTools)  
library(Amelia)

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.8.2, built: 2024-04-10)  
## ## Copyright (C) 2005-2024 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

#### Getting the data  
  
library(ISLR)  
  
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

####  
var(iris[,1])

## [1] 0.6856935

var(iris[,2])

## [1] 0.1899794

#### Making standardization  
  
library(class)  
  
standard.iris <- scale(iris[1:4])  
  
## Binding both standard.iris and species   
  
final.iris <- cbind(standard.iris,iris[5])  
  
View(final.iris)  
  
## Testing variance, it should now show us 1 for every one   
  
var(standard.iris[,1])

## [1] 1

var(standard.iris[,2])

## [1] 1

### Making test and train  
  
sample <- sample.split(final.iris,SplitRatio = 0.7)  
train <- subset(final.iris,sample == TRUE)  
test <- subset(final.iris,sample==FALSE)  
  
#### KNN Model  
  
model.species <- knn(train[1:4],test[1:4],train$Species,k=1)  
  
print(model.species)

## [1] setosa setosa setosa setosa setosa setosa   
## [7] setosa setosa setosa setosa setosa setosa   
## [13] setosa setosa setosa setosa setosa setosa   
## [19] setosa setosa versicolor versicolor versicolor versicolor  
## [25] versicolor versicolor versicolor versicolor virginica virginica   
## [31] versicolor versicolor versicolor versicolor versicolor versicolor  
## [37] versicolor versicolor versicolor versicolor virginica virginica   
## [43] virginica virginica virginica virginica virginica virginica   
## [49] virginica virginica virginica virginica virginica virginica   
## [55] virginica virginica virginica virginica virginica virginica   
## Levels: setosa versicolor virginica

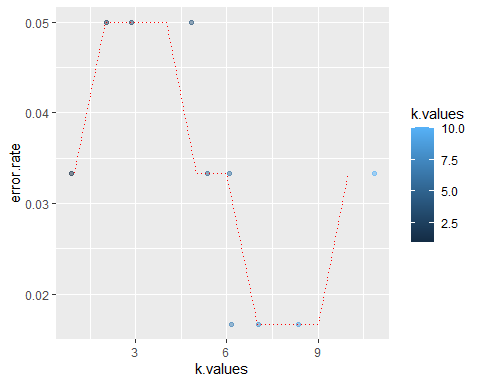
#### Mean error  
  
meanerror <- mean(test$Species != model.species)  
  
print(meanerror)

## [1] 0.03333333

### Elbow method or graph to see the k values and when it stabilizes  
  
model.species <- NULL  
error.rate <- NULL  
  
for (i in 1:10){  
 model.species <- knn(train[1:4],test[1:4],train$Species,k=i)  
 error.rate[i] <- mean(test$Species != model.species)  
}  
  
k.values <- 1:10  
  
df <- data.frame(error.rate,k.values)  
  
print(df)

## error.rate k.values  
## 1 0.03333333 1  
## 2 0.05000000 2  
## 3 0.05000000 3  
## 4 0.05000000 4  
## 5 0.03333333 5  
## 6 0.03333333 6  
## 7 0.01666667 7  
## 8 0.01666667 8  
## 9 0.01666667 9  
## 10 0.03333333 10

ggplot(df,aes(k.values,error.rate)) + geom\_point(position=position\_jitter(w=1, h=0),aes(color=k.values),alpha=0.5) + geom\_line(lty='dotted',color='red')



## although the error value starts high but as the k value increases the error rate goes down significantly  
## also this data set is too small to really implement elbow method