HW 2 Student

IZ Raad

10/17/2023

This homework is meant to illustrate the methods of classification algorithms as well as their potential pitfalls. In class, we demonstrated K-Nearest-Neighbors using the `iris` dataset. Today I will give you a different subset of this same data, and you will train a KNN classifier.

```
set.seed(123)
library(class)

df <- data(iris)

normal <-function(x) {
  (x-min(x))/(max(x)-min(x))
}

iris_norm <- as.data.frame(lapply(iris[,c(1,2,3,4)], normal))

subset <- c(1:45, 58, 60:70, 82, 94, 110:150)
iris_train <- iris_norm[subset,]
iris_test <- iris_norm[-subset,]

iris_target_category <- iris[subset,5]
iris_test_category <- iris[-subset,5]
```

1

Above, I have given you a training-testing partition. Train the KNN with $K = 5$ on the training data and use this to classify the 50 test observations. Once you have classified the test observations, create a contingency table – like we did in class – to evaluate which observations your algorithm is misclassifying.

```
pr <- knn(iris_train, iris_test, cl = iris_target_category, k = 5)
tab <- table(pr, iris_test_category)
tab
```

```
##             iris_test_category
## pr           setosa versicolor virginica
##   setosa          5          0         0
##   versicolor      0         25         0
##   virginica       0         11         9
```

```
accuracy <- function(x){
  sum(diag(x))/(sum(rowSums(x)))*100}
accuracy(tab)
```

```
## [1] 78
```

2

Discuss your results. If you have done this correctly, you should have a classification error rate that is roughly 20% higher than what we observed in class. Why is this the case? In particular run a summary of the `iris_test_category` as well as `iris_target_category` and discuss how this plays a role in your answer.

This k-nearest neighbors classification misclassified 11 versicolors as virginicas, leaving us with a 78% accuracy. Let's see why this is.

```
summary(iris_test_category)
```

```
##     setosa versicolor  virginica
##          5         36          9
```

```
summary(iris_target_category)
```

```
##     setosa versicolor  virginica
##         45         14         41
```

We notice that versicolors are greatly underrepresented in iris_target_category, and most of them have ended up in the testing set. Because we are using a k-nearest neighbors algorithm, underrepresenting one category may result in testing data having less possible neighbors of that category. Any bordering case will be more likely to default to the class with more data in it, as this will hold the bulk of the neighbors. Thus, this underrepresented class is more likely to be misclassified.

3

Build a github repository to store your homework assignments. Share the link in this file.

Click Here for GitHub