HW 2

Daniella Perez

09/26/2024

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.

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.

```
#normalizing the numeric columns
normal <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

#applyinfg normalization- data is now scaled between 0 and 1.
iris_norm <- as.data.frame(lapply(iris[, c(1:4)], normal))


subset <- c(1:45, 58, 60:70, 82, 94, 110:150)

#training and test sets for the features and the target category (species)
iris_train <- iris_norm[subset, ]
iris_test <- iris_norm[-subset, ]

#category (species)
iris_target_category <- iris[subset, 5]  # Training labels
iris_test_category <- iris[-subset, 5]   # Test labels

#k-NN with k=5
knn_pred <- knn(train = iris_train, test = iris_test, cl = iris_target_category, k = 5)

#contingency table
contingency_table <- table(Predicted = knn_pred, Actual = iris_test_category)
print(contingency_table)
```

```
##             Actual
## Predicted    setosa versicolor virginica
##   setosa          5          0         0
##   versicolor      0         25         0
##   virginica       0         11         9
```

QUESTION2: 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.

calculating the classification rate:

```
correct_classifications <- sum(knn_pred == iris_test_category)
total_observations <- length(iris_test_category)
accuracy <- correct_classifications / total_observations
print(accuracy)
```

```
## [1] 0.78
```

```
summary(iris_test_category)
```

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

```
summary(iris_target_category)
```

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

The decrease in accuracy is caused by an imbalance between the training and test sets. The training set has many examples of setosa and virginica, but fewer versicolor samples, while the test set contains a larger number of versicolor observations. This imbalance leads the KNN model to misclassify versicolor more frequently, resulting in about a 20% lower accuracy. The unequal representation of classes impacts the model's ability to generalize effectively across all species.

QUESTION3: Choice of K can also influence this classifier. Why would choosing $K = 6$ not be advisable for this data?

Choosing $K = 6$ can result in ties during classification and is not suitable for a dataset with three classes. Opting for an odd, indivisible K helps prevent ties and enhances decision-making within the KNN algorithm. We rather pick an odd K to avoid ambiguity. We want K to be a value that is indivisible.

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