STOR 390: HOMEWORK 2

Riley Harper

February 8, 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.

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.

```r
set.seed(123)
#STUDENT INPUT
k <- 5
knnmod <- knn(train = iris_train, test = iris_test, cl = iris_target_category, k = k)
continTable <- table(Predicted = knnmod, Actual = iris_test_category)
continTable
```

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

```r
classifErrorRate <- 1 - sum(diag(continTable)) / sum(continTable)
classifErrorRate
```

```
## [1] 0.22
```

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.

```r
print("Test Category:")
```

```
## [1] "Test Category:"
```

```r
summary(iris_test_category)
```

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

```r
print("Train Category:")
```

```
## [1] "Train Category:"
```

```r
summary(iris_target_category)
```

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

Of the species selected for the training set, only 14 were of the 'versicolor' class-which represented 72% of the testing set. Clearly, the training set is unrepresentative of the population as a whole and, in this example, had led to biased results. In fact, the model likely overfit to the 'setosa' and 'virginica' classes due to their higher representation in the training data. Overfitting is a phenomena that occurs when a model 'learns' the features and noise in the training data to a level that it negatively impacts the performance of the model on new data. In this case, since the 'versicolor' class is underrepresented in the training set, the model may not have learned to classify it as effectively as the other two classes.

To address this issue and improve the model's ability to generalize, cross-validation is a technique which can be employed. Cross-validation involves systematically partitioning the dataset into several subsets and training the model multiple times, each time using a different subset as the test set and the remaining data as the training set. This process helps in ensuring that the model is tested on a representative sample of the entire dataset, thereby giving a more accurate estimate of its performance on unseen data.There are various forms of cross-validation, such as k-fold and stratified cross-validation. The choice of method depends on the context of the data set and the data setand the data scientist's judgment. For instance, stratified cross-validation is particularly effective in maintaining proportional representation of each class across all folds, which is crucial in scenarios like ours where class imbalance is a concern.

3

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

https://github.com/rmharp/STOR-390