Izaak King

Professor Eleish

Data Science

November 4, 2025

“Lab 5”

For this lab, I will train 2 SVM classifiers to predict the type of wine using a subset of Flavanoids, Total phenols, Color Intensity, and Alcohol. This subset includes the two highest variable contributors from both component 1 and component 2 from our PCA in lab 4. One SVM will use a linear kernel and one will use a polynomial kernel. Following this, I will train another classifier from kNN and see how it performs based on the features, comparing models with precision, recall and F1.

I. SVM classifiers

We first start by training an SVM classifier to predict the type of wine. We use train the model using a sample of 80% of our data. We will then output performance scores when predicting it’s own train data, as well as on the other 20% of the data, the test data. We will start by using tune.svm to find optimal cost and gamma values

1. SVM with linear kernel

   a. Tuning SVM

```
> tuned.svm0 <- tune.svm(
+   Type ~ Flavanoids + `Total phenols` + `Color Intensity` + Alcohol,
+   data = wine,
+   kernel = 'linear',
+   cost = Cost.range
+ )
> tuned.svm0$best.parameters
  cost
1    1
```

Thus, we will train our model with cost = 1

   b. Performance scores for training data

```
> accuracy
[1] 0.9577465

> data.frame(precision, recall, `
  precision    recall        f1
1 0.9347826 0.9347826 0.9347826
2 0.9473684 0.9473684 0.9473684
3 1.0000000 1.0000000 1.0000000
```

   c. Performance scores for test data

```
> accuracy
[1] 0.9444444

> data.frame(precision, recall, `
  precision    recall        f1
1 0.9285714 1.0000000 0.9629630
2 0.9285714 0.9285714 0.9285714
3 1.0000000 0.8888889 0.9411765
```

2. SVM with polynomial kernel

a. Tuning SVM

```
> tuned.svm1 <- tune.svm(
+   Type ~ Flavanoids + `Total phenols` + `Color Intensity` + Alcohol,
+   data = wine,
+   kernel = 'polynomial',
+   gamma = gamma.range,
+   cost = Cost.range
+ )
> tuned.svm1$best.parameters
    gamma cost
102   0.2    2
```

Thus, we will train our model with gamma = 0.2, and cost = 2

b. Performance scores for training data

```
> accuracy
[1] 0.9225352
```

```
> data.frame(precision, recall, f1)
  precision    recall        f1
1 0.9302326 0.8695652 0.8988764
2 0.8750000 0.9824561 0.9256198
3 1.0000000 0.8974359 0.9459459
```

c. Performance scores for test data

```
> accuracy
[1] 0.9166667
```

```
> data.frame(precision, recall, f1)
  precision    recall        f1
1 0.9285714 1.0000000 0.9629630
2 0.8666667 0.9285714 0.8965517
3 1.0000000 0.7777778 0.8750000
```

II. kNN classifier

We will now train a kNN model to compare performance. This will be trained with the same subset input variables, and k = 3 for each wine type. We will train with the same sample of 80%, and test with the same 20%

a. Performance scores for test data

```
> accuracy
[1] 0.9166667

> data.frame(precision, recall, f1)
  precision    recall        f1
1 0.9285714 1.0000000 0.9629630
2 0.9230769 0.8571429 0.8888889
3 0.8888889 0.8888889 0.8888889
```

III. Model Comparison

Based on accuracy, precision, recall and f1 scores our SVM with the linear kernel was the best performing model for this wine dataset. It performed better than both the other SVM and kNN model. It's also interesting to note that the SVM with the polynomial kernel performed with nearly the same accuracy as the kNN, with precision and f1 scores better, but recall worse than the kNN model. Overall, it's clear that our SVM approach was more effective than the kNN model.