Iris Classification Project

Arjun Niranjan

06/10/2020

Introduction

The purpose of this project is to analyse a subset of Fisher's Iris dataset and to train a classifier using Python's Sci-Kit Learn module to determine whether a flower's species is Setosa or Versicolor based on some of its physical properties. The following code chunk displays the features we will be using to classify by, followed by the two classes:

```
import pandas as pd
import numpy as np
from sklearn import svm

iris = pd.read_csv('iris.csv')
iris = iris[(iris['Species']!='virginica')]
iris.columns[1:4]
```

```
## Index(['Sepal.Length', 'Sepal.Width', 'Petal.Length'], dtype='object')
```

```
set(list(iris['Species']))
```

```
## {'versicolor', 'setosa'}
```

Building a basic classifier function

The following code chunk takes a random selection of n labelled flowers as the training data. The remaining flowers are left as the test batch. The function's output is the proportion of predicted labels that are correct.

```
def classify(n):
    X_train = iris.sample(n)
    Y_train = X_train['Species']
    X_test = pd.concat([X_train, iris]).drop_duplicates(keep=False)
    Y_test = X_test['Species']
    X_train = X_train.drop(['Species'], axis=1)
    X_test = X_test.drop(['Species'], axis=1)
    clf = svm.SVC(kernel='linear')
    clf.fit(X_train,Y_train)
    Y_pred = clf.predict(X_test)
    result=[Y_test.iloc[i]==Y_pred[i] for i in range(100 - n)]
    accuracy = sum(result)/(100-n)
    return accuracy

classify(50)
```

```
## 1.0
```

The ouput above tells us how well a training set of size 50 predicted the species of 50 unlabelled flowers; a value of one showing a perfect prediction and a value of zero shows a wholely incorrect prediction.

Further Analysis

We can investigate how much training data it takes to build a fairlt accurate classifier. If a smaller training set still trains a good classifier, then we can infer that there is a lot of difference between the two classes. We can use Monte Carlo simulation on different values of n to see how accuracy varies with training set size. The code chunk below tests classifiers of sizes 30, 20, 10, 5 and 2 10000 times each. Then the average success rates are printed as output. The try-except statement ensures we get at least one of each species.

```
sizes = [30,20,10,5,2]
for n in sizes:
    outs = np.empty(10000)
    for i in range(10000):
        try:
            outs[i] = classify(n)
        except ValueError:
            i = i-1
            continue
    accuracy = sum(outs)/10000
    print(f"Training set of size {n} gives our classifier a {accuracy} success rate")
```

```
## Training set of size 30 gives our classifier a 0.9914471428571764 success rate
## Training set of size 20 gives our classifier a 0.9817712500000078 success rate
## Training set of size 10 gives our classifier a 0.9565611111111332 success rate
## Training set of size 5 gives our classifier a 0.9310354605263054 success rate
## Training set of size 2 gives our classifier a 0.45442653061223165 success rate
```

This tells us that the two species of flower are so different that just taking a training set of 5 flowers is still quite accurate with a success rate over 90%.

Evaluation

We have learned from this project that Setosa and Versicolor flowers have significant difference in a combination of sepal length, width and petal length. The differences are so substantial that a classifier trained on just 5 data points has a fairly high success rate when determining the species of the remaining 95 flowers in the dataset.