**CAB320 Assignment 2: Machine Learning**

*Declan Kemp (n10263047), Callum McNeilage (n10482652)*

Introduction

Machine learning is a constantly growing field of computer science with a broad range of potential applications in a variety of fields. This report focuses on the use of machine learning for classification problems, specifically the classification of tumours based on various physical characteristics.

Methodology

Several different classifiers were implemented in Python using the Scikit-Learn library, specifically Decision Tree, Nearest Neighbours, Support Vector Machine, and Neural Network. For all classifiers, a similar process for training and testing is used. The data is extracted from the csv file and divided into an array of classification data (X) and an array of class labels (y). These are then randomly divided into a training set and a testing set, with 30% being reserved for testing. The classifier is then trained using 5-fold cross validation to tune a specific hyperparameter across a provided range of values, and finally run on the reserved testing set. The results of this final run are then printed to the console.

Decision Trees are a comparatively simple way to train a machine learning classifier. They examine the data and derive a series of branching questions which eventually lead to one of the class labels. Decision Trees are easier to follow the computer’s “thinking” than most other classifiers but are at substantial risk of creating overly complex trees which match the training data very well but do not generalise. This is called overfitting, and one of several ways to mitigate this is by modifying the minimum number of samples required to form a leaf node (the end points of the tree with a class label). This code tests minimum leaf sizes between 1 and 10.

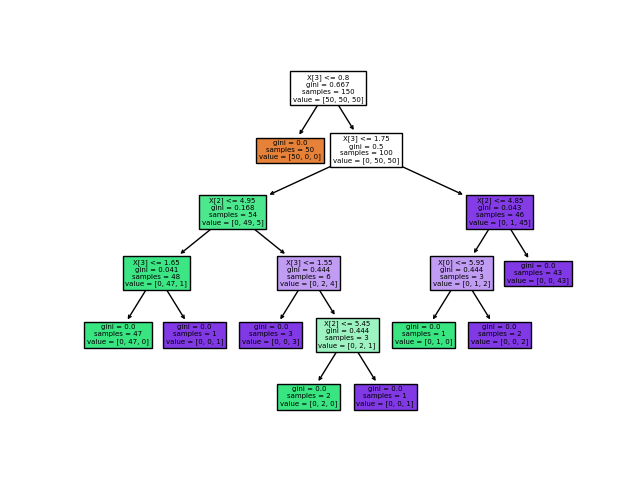


Figure 1: An example Decision Tree

The principle behind Nearest Neighbours classifiers is to simply find which of the training data points is closest to the example being tested, treating each attribute an example has (each physical characteristic of a tumour, in this case) as a separate dimension. To smooth the data and avoid overfitting, this is expanded to be the most common result of the nearest k neighbours, where k is a positive integer. Since this classification problem is binary in nature (every tumour is one of malignant and benign, there are no other potential class labels), it is recommended to use an odd number for k to avoid ties between the two categories. This code is tested on all the odd numbers between 1 and 19 inclusive.

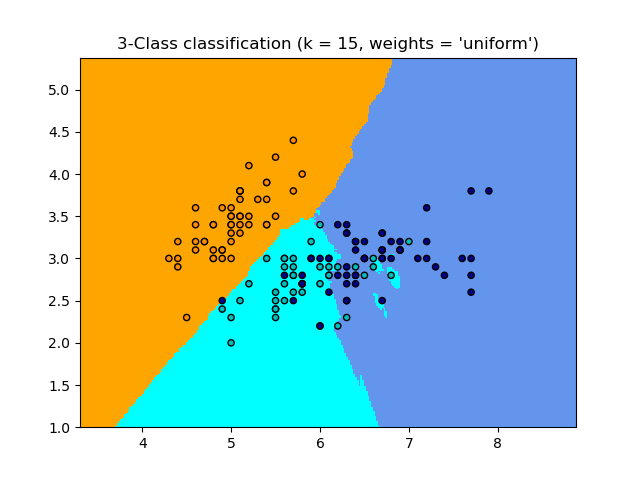


Figure 2: An example Nearest Neighbours classification

Support Vector Machines work by treating all the datapoints as higher-dimensional vectors and creating hyperplanes with one fewer dimension to separate them by class. The hyper-parameter modified here is C, a regularisation parameter in the formula used to determine where the hyperplanes should sit. Unlike the other hyper-parameters tested here, C is most usefully scaled as orders of magnitude, so the values selected are the first to fifth powers of ten (1, 10, …, 100 000).

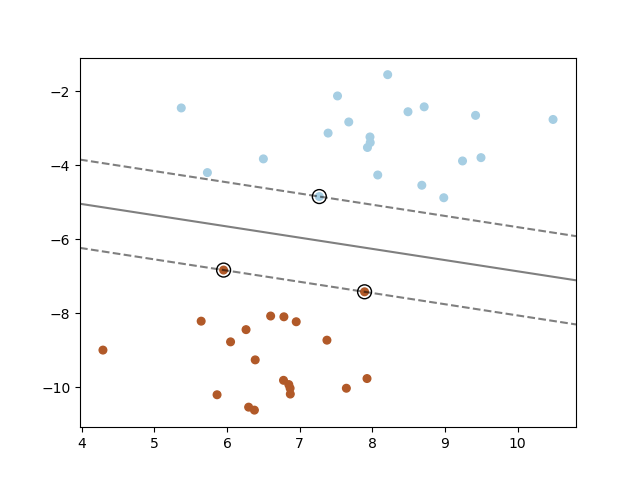


Figure 3: An example Support Vector Machine Classifier

Neural Network classifiers are comparatively different to the other classifiers. They are trained through many repeated iterations over the data and depend on hidden layers of neurons making understanding their choices much harder. They can often create more accurate and useful classifiers, but at the cost of a greatly increased computation time. The specific Neural Network used here is Multi-layer Perceptron, which consists of several layers of binary classifiers. The hyper-parameter being tuned here is the number of hidden neurons, which are arranged in three layers, each layer with a multiple of five – [5, 5, 5], [10, 10, 10] and so on until [25, 25, 25].



Figure 4: An example Neural Network

Results

After running each classifier 50 times, the results were as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| *Table 1:* | Accuracy | Hyper-parameter | Time taken per test (s) |
| Decision Tree | 92.6% | S = 5 | 0.203 |
| Nearest Neighbours | 93.3% | K = 8 | 0.279 |
| Support Vector Machine | 95.2% | C = 10000 | 0.370 |
| Neural Network | 97.0% | N = 42 | 20.2 |

Note that the hyper-parameter listed here is the average parameter used across the 50 tests – S is the minimum leaf size, K is the number of neighbours considered (only odd numbers were tested, but the mean fell closest to 8), and N is the total number of hidden neurons ([15, 15, 15] would be listed as 45, for instance). Since C scales by orders of magnitude, the mean is not a sensible measure to use, so the mean of the logarithms (base 10) was used instead, then converted back to a true value for C.

It can be seen above that all classifiers perform well, assigning class labels with more than 90% accuracy. The classifiers which perform better take more time to do so, with fairly consistent and gradual shifts within the Decision Tree, Nearest Neighbours, and Support Vector Machine but a substantial time increase for the Neural Network, on the order of 100x as long to compute.

Conclusion

In general, it would be reasonable to conclude that the Neural Network is not sufficiently better than the other classifiers to justify the massively increased computation cost – less than 2% increased accuracy for 55x the computation time. However, cancer diagnosis is an extremely sensitive area, and even slight gains are likely worth the increase in computation time. That said, 97% accuracy is still too low for such a risk-filled classification, so a machine learning classifier capable of aiding or replacing oncologists would have to be substantially more deliberately designed than these highly generic classifiers. A large number of additional factors, such as the location of the tumour in the body, medical history of the patient, various carcinogenic risk factors, and the true rates of various cancers within the overall population would all need to be considered, as well as training the classifier on a much larger data-set and with more specific class labels. Regardless, machine learning is a powerful tool and will widespread in future medical diagnosis.

References

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011