COMP3308 Assignment 2

490182143

Aim

The aim for this study was to observe the performance of k-Nearest Neighbours and Naive Bayes classification algorithms, both written natively in Python without the support of any external libraries. The aim was to create a fair comparison, achieved by eliminating any variables that may cause any unfair advantage, such as the language used, external libraries, dataset and the same method for testing. This study is important as it highlights my ability in being able to compose algorithms from scratch by capturing their mathematical essence and being able to see them perform on real world data and obtain impressive accuracies, and be very competent against pre-built classifiers as found on Weka.

Data

The dataset is the Pima Indians Diabetes dataset from the National Institute of Diabetes and Digestive and Kidney Diseases. It contains 8 attributes of the human body:

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)

The ninth variable in the dataset is the diagnosis of the patient, with “yes” meaning they were diagnosed with diabetes and “no” meaning they weren’t. A second dataset was created from the first, using a correlation-based feature selection process. The principle behind CFS is that it evaluates several permutations of subsets of features in the dataset to see which one performs the best, with its performance being ranked by its correlation with the classification but inversely ranking the correlation to each other. The Best First Search method was used for searching the subsets for evaluation.

Results

WEKA

	ZeroR	1R	1NN	5NN	NB	DT	MLP	SVM	RF
NO CFS	65.1042%	70.8333%	67.8385%	74.4792%	75.1302%	72.0052%	75.3906%	76.3021%	74.8698%
CFS	65.1042%	70.8333%	69.0104%	74.4792%	76.3021%	73.3073%	75.7813%	76.6927%	75.9115%

MY CLASSIFIER

	My1NN	My5NN	MyNB
No CFS	77.0031%	89.0045%	91.1566%
CFS	78.2558%	87.0116%	91.4517%

The classifiers written in Python appear to have a higher accuracy than the Weka pre-built classifiers by a statistically significant amount. Naive Bayes appears to perform better than kNN in Weka as well as in Python, however it should be noted that kNN is not a linear classifier, as opposed to Naive Bayes, which is. Since the Pima dataset is relatively small the differences in compilation time aren't big but it would scale as datasets increase. There is a stark increase in accuracy when increasing the k value from 1 to 5, observed both in the Python classifier as well as on Weka. Although their difference CFS doesn't appear to have a large difference in the accuracy of the classifiers, with My5NN surprisingly seeing a drop of ~2% in the accuracy with CFS.

Conclusion

kNN and Naive Bayes are both robust classification algorithms for datasets such as the Pima dataset, SVM used in Weka is also a viable option for datasets such as this. kNN may not be as favourable of a choice if the dataset is larger and/or has more dimensions. It would be worthwhile to explore the accuracy of SVM on Python.

Reflection

This assignment and study offered me the opportunity to build two of the most popular classifiers that are often used with the aid of an external library, and being able to build its counterpart natively on Python using just the theoretical knowledge from the lectures offers me a sense of accomplishment in being able to have a look under the hood and build a pipeline for a real world dataset and deriving accurate predictions. There were several complexities in terms of understanding the flow of data and processing it at each step were fun challenges in helping me learn how to build a practical and useful implementation of a machine learning algorithm.