Introduction

Aim:

The aim of this study is to evaluate and compare the accuracy of my implementations of a Naïve Bayes (NB) and k-Nearest Neighbour (k-NN) classifiers against Weka's ZeroR, 1R, k-NN, NB, Decision Tree (DT), Multi-Layer Perceptron (MLP), Support Vector Machine (SVM) and Random Forest (RF) algorithms using 10-fold stratified cross validation with and without correlation-based feature selection (CFS) on the Pima Indian Diabetes dataset.

Importance:

The goal of any classifier is to correctly classify new data. Accuracy, the proportion of correctly classified examples, is the measure of performance used to evaluate this. Evaluation allows for the comparison of the performance of different classifiers and also may be used to estimate the performance of classifiers in predicting the presence of diabetes in other Pima Indian patients.

Data

Dataset:

The dataset is taken from the Pima Indians Diabetes Database. The variable investigated is whether a patient from a population near Phoenix, Arizona, USA, shows signs of diabetes according to World Health Organisation criteria (i.e the 2-hour postload plasma glucose was at least 200mg/dl). All 768 patients are females at least 21 years old of Pima Indian heritage. The attributes measured were:

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)
9. Class variable ("yes" or "no")

Attribute Selection:

The classifiers were also evaluated on the dataset after CFS was applied. Attribute selection is based on the idea that data contains attributes that are either redundant or irrelevant so can be removed without loss of information. The hypothesis behind CFS is that "Good subsets of features contain features that are highly correlated with the class and uncorrelated with each other". This means the attributes selected after CFS are all good at predicting the class but are also uncorrelated to each other which avoids redundancy of information. This also avoids the curse of dimensionality where high dimensional data leads to overfitting and high computational costs.

The attributes selected by CFS were:

1. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
2. 2-Hour serum insulin (mu U/ml)
3. Body mass index (weight in kg/(height in m)^2)
4. Diabetes pedigree function
5. Age (years)

The class variable ("yes" or "no") is also necessarily required so is kept in the dataset.

Results and Discussion:

Results:

	ZeroR	1R	1NN	5NN	NB	DT	MLP	SVM	RF
No Feature Selection	65.10%	70.83%	67.84%	74.48%	75.13%	71.75%	75.39%	76.30%	74.87%
CFS	65.10%	70.83%	69.01%	74.48%	76.30%	73.31%	75.78%	76.69%	75.91%

	My1NN	My5NN	MyNB
No Feature Selection	68.87%	75.52%	74.61%
CFS	68.62%	74.61%	76.31%

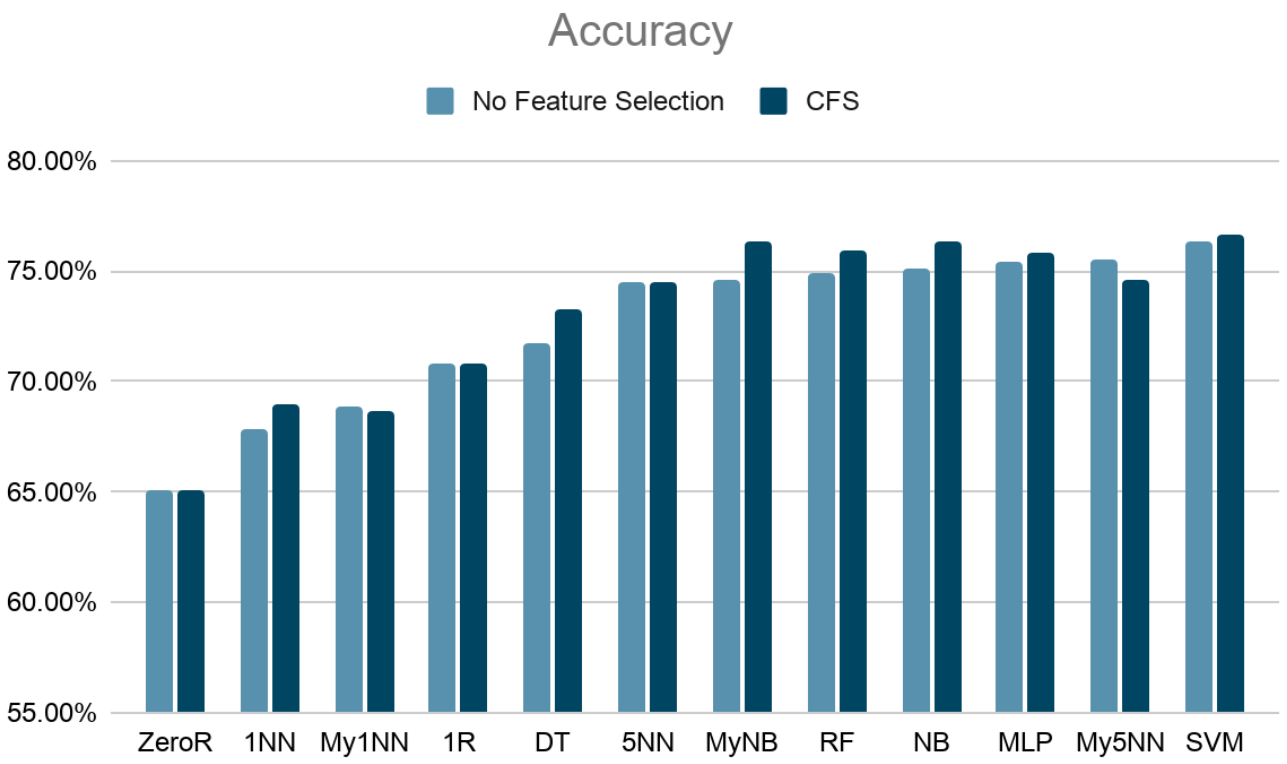
Discussion:

Each classifier was evaluated based on its accuracy. However, accuracy on training data is not a good indicator of performance so test data is required. In general, more training data results in a better classifier and more test data results in a better accuracy estimate. With limited data, there is a tradeoff between the two. 10-fold stratified cross validation addresses this issue by splitting the data into 10 equal subsets, with one subset serving as the test data and the remaining as the training data. The classifier is built 10 times with a different subset as the test data each time, and the average accuracy is taken. Subsets are stratified, such that each class is represented in approximately equal proportions in each.

From the results, my 1NN and 5NN classifiers performed slightly better than Weka's with no feature selection, whereas my NB classifier performed slightly worse. This difference is likely due to the evaluation method as both mine and Weka's implementations should be identical or close enough that it would have no significant difference. The difference likely lies in the stratification of the subsets which has some uncertainty around it. Nonetheless, the difference in performance is negligible. Interestingly, Weka's 1NN performed slightly better than mine on the CFS dataset but slightly worse on the NB. Again this is likely due to the stratification method and is also an insignificant difference.

In general, Weka's classifiers performed similarly or slightly better on the CFS dataset than the dataset with no feature selection applied in terms of accuracy using the method described above. The ZeroR, 1R and 5NN classifiers saw no improvement from the CFS dataset, whereas the 1NN, NB, DT, MLP, SVM and RF classifiers saw a small improvement, with DT seeing the highest improvement at 1.56% increase. For my own implementations of the k-NN and NB classifiers, the CFS dataset performed slightly worse for both the 1NN and 5NN classifiers but better for the NB classifier. k-NN performs well in low dimensions which could possibly explain the lack of improvement in the CFS dataset.

Based on the accuracy performance alone, CFS appears to have only slight benefits. However, CFS also has the benefit of reducing the amount of data required for learning, more compact and easier to understand learned knowledge, and reduced training time (Hall, 1999). Therefore, in general, CFS is considered successful if data dimensionality, the number of attributes, is reduced while the accuracy of the classifier improves or remains the same (Hall, 1999). From the results and previous section, CFS reduced the dimensionality of the dataset while maintaining similar accuracy. Therefore, it can be considered successful in this case.



The zeroR algorithm is the simplest classification method by predicting the majority class. It can therefore be taken as the baseline performance as it has no predictive power. As can be seen it performed the worst out of all the classifiers at 65.10% accuracy. 1R is the next simplest algorithm and is computationally cheap. In problems where one attribute is sufficient to determine the class 1R performs well but in this case, 1R does not perform well as no one attribute can determine the presence of diabetes.

Next, the k-NN is a lazy learning algorithm meaning a classifier is not built until a new example needs to be classified. The advantages of this is faster training but at the cost of slower classification and large memory requirements as all training examples need to be stored. While simple, k-NN tends to work well in practice but is sensitive to irrelevant attributes. Both Weka's and my implementation of 1NN performed poorly, likely due to noise in the training data resulting in wrong classifications. In general, there is benefit to increasing k up to a certain point which improves its robustness to noise in the data. This can be seen in the better performance of both Weka's and my implementation of 5NN both of which performed similarly to more complex algorithms like MLP and SVM.

Eager learning algorithms such as 1R, DT, NB, neural networks like MLP, and SVM are the opposite of lazy learning algorithms. These algorithms take longer to train but are faster at classification of new examples. DT is more complicated than 1R but is still an efficient and easy to implement algorithm. The cost of building the tree in DT is O($mn\log n$) for n instances and m attributes. It is also easier to understand for humans than neural networks and SVM. However, from the results it performs poorly in terms of accuracy, only beating out ZeroR, 1NN and 1R.

RF is an ensemble of classifiers that builds on DT by bagging trees using randomly selected attributes to form predictions and classify new examples using majority voting. This minimises the effects of errors that are unique to each DT and therefore improves the overall accuracy as can be seen in the results. RF typically outperforms a single DT but requires each individual DT to perform well and have low correlation to each other. However it clearly requires more computational power than a single DT.

NB is a statistical classifier based on Bayes theorem. It is simple due to the assumption of independence of attributes which is unlikely to be true. However, it is very efficient, with a time complexity of O(*mn*) for *m* training examples and *n* attributes. Estimating conditional probabilities also reduces the impact of noise making NB more robust to noise in the dataset. Despite being simple and requiring assumptions that likely do not hold, it performs quite well, close to the best performer SVM in terms of accuracy.

MLP is a neural network made up of layers of artificial neurons called perceptrons. Theoretically they can approximate any function well. In practice however, this is dependent on starting conditions and the number of hidden layers and neurons. Too few results in underfitting, a neural network that is unable to learn, whereas too many results in overfitting. Weka's MLP uses backpropagation, an algorithm using the steepest gradient descent method for minimising the mean square error to increase the learning rate of the MLP. Neural networks like MLP are more complex and less efficient than simpler classifiers but yet for this dataset performed similarly to simpler classifiers like NB and 5NN.

SVM is a classifier that creates an optimal hyperplane based on the training data to classify new examples. SVM performs well when there is a clear separation of classes and also generalises well as it avoids overfitting. SVM is a relatively efficient algorithm that can learn non-linear boundaries like MLP but it performed the best in terms of accuracy on both the CFS and original dataset.

Conclusion:
In summary, 10-fold stratified cross validation is an effective way of evaluating different classifiers. Application of CFS on these algorithms was also successful in terms of offering benefits such as generally improved accuracy and efficiency but it should be noted that CFS may fail to select relevant features when features are highly correlated (Hall, 1999). Based on the results of this study, SVM performed the best out of all methods tested, but k-NN, NB, RF and MLP all performed comparably all of which ranging between the 74-76% accuracy mark, making them potentially useful predictors of diabetes in 21 years old female patients of Pima Indian heritage.

An interesting area for future work would be to determine which of the best performing classifiers offer the most practical benefits in predicting diabetes. As each algorithm has its advantages and disadvantages, with some being much simpler and efficient, identifying the simplest classifier that can classify new patients with a reasonable degree of accuracy would help reduce cost in an implementation of these classifiers. In general, some problems are simple enough to be solved with less complex classifiers. Another area for future work could be to identify other attributes in diabetic patients that could be better predictors of diabetes. By comparing the accuracy achieved in those datasets with the one in this study, one would be able to determine the predictive power of different attributes to create a better model for predicting the presence of diabetes in patients.

Reflection:
The most important aspect of this study for me was that it not only allowed me to apply the knowledge learnt in the unit through implementation of my own algorithms but also to apply it in a scenario that is similar to real world applications of machine learning. It gave me a greater appreciation of the value of data science and work involved in creating machine learning models for real world applications and the potential benefits of machine learning and AI in general. It also helped me to reinforce my understanding of the theory behind the efficiency of each of the classifier algorithms through comparing each of them. Overall, I found it to be a uniquely rewarding study that gave me insight into the purpose behind AI.

References

Hall, M.A. (1999). *Correlation-based Feature Selection for Machine Learning* (Doctoral thesis, The University of Waikato, New Zealand). Retrieved from https://www.cs.waikato.ac.nz/~mhall/thesis.pdf