Report

SID:480133780, 480378222

Aim

The aim is to compare the performance of machine learning algorithms in a prediction task. The task is to predict the likelihood of diabetes in female Pima peoples. This study highlights different model's performance in terms of accuracy and uses this performance to explain their possible application in medicine. The paper gives an introduction on how Machine Learning and AI could be applies in the medical field. Furthermore, we are going to compare our python implementation of classifiers with Weka's built-in classifiers.

Dataset

The dataset used is the Pima Indians Diabetes data (pima.csv) that provides a sample size of 768 people who shared information relevant to diabetes and whether they have diabetes or not (500 do have, 267 do not have). The source and owner of the data is the National Institute of Diabetes and Digestive and Kidney Diseases. The data has been modified by replacing missing values with average values for the purpose of the assignment.

All patients are female and at least 21 years old, and they all provided the following information:

- Number of times they were pregnant
- Plasma glucose concentration
- Blood pressure
- Skin fold thickness
- Insulin they received
- BMI
- Diabetes pedigree function
- Age
- Whether they have diabetes or not

With this dataset, the aim is to predict diabetes outcome in individuals with the above information. Apart from that, this dataset was modified with Correlation-based feature selection using the method Best-First Search in Weka software. With that, 6 variables were kept:

- Plasma glucose concentration
- Insulin they received
- BMI
- Diabetes pedigree function
- Age
- Whether they have diabetes or not

During our work, we worked with both datasets and compared results.

Result and Discussion

Comparison between classifiers (Weka)

	ZeroR	1R	1NN	5NN	NB	DT	MLP	SVM	RF
No Feature Selection	65.1042	70.8333	67.8385	74.4792	75.1302	71.7448	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

Python Implementation

	My 1NN	My 5NN	My NB
No Feature Selection	68.2347	75.1281	75.2604
Feature selection	68.3526	75.3913	76.0416

Overall, the SVM algorithm performed the best with 76-77% accuracy. While the ZeroR algorithm performed the worst.

ZeroR - 1R

Since there are 500 non-diabetes and 268 diabetes the ZeroR algorithm label every value as non-diabetes as it is the majority class. However, if we were testing on a different dataset, this accuracy could easily be worse which means that the classifier is not reliable.

Although 1R algorithm performed better, it only uses 1 feature which realistically will not provide the best prediction of diabetes. There are many factors that together cause diabetes and relying one 1 factor ignores too much information to be considered a good approach. On top of that, the selection of 1 attribute is not perfect if we work with continuous values (real numbers) instead of nominal values. That is because the feature selection is based on the best frequency table in terms of prediction that the 1R algorithm creates for every feature.

1NN and 5NN

Our implementation did about as well as weka's for 1NN and 5NN. Overall the accuracy is quite high for both; mainly due to normalising the data so that all values lie between 0 and 1. This makes the distance calculations not outweigh each other for different classes which reduces the likelihood of overfitting to a particular feature based on the metric used to measure it; e.g. features like age will always have a difference of at least one where as Plasma Glucose Concentration can have differences at a far smaller. By normalising, the differences can be equated so that the metric doesn't mislead the distance function. This does create some issues though, because it effectively equates the importance of all the features which is most likely not the case; as some features will be of greater importance than others.

Feature selection alleviates this problem a little bit but still has the problem of the selected features having different importance which our implementation will ignore. In general, KNNs computation time can be quite high for larger datasets but by using feature selection KNN models can be built and tested quite efficiently if being used at a larger scale.

NB

While 1R classifies based on 1 feature, NB uses all features, so the accuracy is significantly higher as expected. However, we do not know if the features we are using here are independent from each other and that could be an issue. On top of that we assumed that all variables here are equally important. For example, BMI could be more important than age or blood pressure and glucose concentration could be related to one another. Hence, we may not want to use this algorithm in practice. Note that accuracy of the Weka implementation is equal to the Python implementation. Considering the time taken to write the Python code, it is a better idea to only use Weka for performance as it takes less time (just like with KNN).

RF-DT

When it comes to Decision Tree and Random Forest, RF is almost like an extension of DT. It is fair to expect that RF performs better as it uses multiple trees to make prediction, but it seems like the difference between DT and RF in accuracy is around 1-2%. Considering that the computation time of RF is longer than for the DT but the result is similar, we might only stick with the DT algorithm. If more features were added perhaps the RF would perform better as there would be more features to bag and create a more diverse forest.

MLP

The MLP model did very well in terms of accuracy and compared to most neural nets is quite simple too. The only work comes from configuring starting weights and number of hidden layers which is automated by Weka. MLP did quite well but there may be better methods in the neural net family as its possible for our MLP to have reached a local optimum in terms of reducing the error instead of the global optimum. More testing should be done (especially out of sample) before it should be used as we do not know if this MLP is overfitted to the training data; meaning it won't handle new data points well.

SVM

SVM had the highest accuracy with or without feature selection. SVM can generate non-linear decision boundaries by leveraging the kernel function making it computationally efficient. Because it tries to maximise the margin between points it can lead to a very effective classifier which is reflected in the results. Maximising the margin of the decision boundary makes it far easier to classify the fringe cases then a classifier like KNN as to do so in KNN you need to expand your neighbours but in doing so you expand the region of search which can lead to including points which lie to far away from the new data point; swinging the vote and hence the prediction. Due to its ability to deal with fringe cases it is likely for SVM to generalise to new data however this was not tested. SVM is also quite efficient in terms of training being much faster than MLP and in terms of cost and performance the trade-off is quite good due to its high performance.

Effect of Feature Selection

As we can see in the table, using the CFS dataset, Weka gave us an overall better accuracy for almost every algorithm than for the original dataset where we did not select features. That could be due to the fact that there were many unnecessary variables included in the original dataset that make noises for the classification algorithms. Also, working with less variable reduces the computational time of our program. So, removing them helps the performance overall.

Reflection

Student 1:

One of the most interesting and new experience during the assignment was definitely related to Weka. I have never used it before, but found it convenient to use. I would suggest it to people who are interested in AI but do not have experience in coding. Overall, it was an enjoyable assignment that was challenging and interesting at the same time as one of the few occasions where we really dig into some advanced algorithms.

Student 2:

For me the most important thing I learnt was the ease of implementation for some of these methods. KNN does not take a lot of work to get going and its predictive power is quite good considering the amount of work you do. I found it really interesting to see how it compared to wekas performance as it only did a bit worse which was nice to see but I'm interested to see in how you can make it even better (maybe experiment with different distance metrics). It was also good to see all these different methods we've learnt about being put in practise as it lets you test your understanding while also giving you an insight into how their strengths and weaknesses translate into the real world or with real data.

Conclusion

Our results show that some classifiers outperform others however this is by no means a comprehensive comparison of all the classifiers we used as each has its own set of strengths and weaknesses which translates to performance for different kinds of data or prediction problems. What our report does show is that for data with minimal noise, some class imbalance (500 vs 267) and a relatively small set of features; SVM can outperform unsupervised methods like KNN or supervised methods like MLP etc as well as rule-based algorithms like 0R and 1R. It's important to note that this can easily change based on the balance in the dataset or the performance metric being used.

Furthermore, because the performance is only measured for in sample performance, we cannot make any comment on their performance in practise, although there in sample performance is a good indicator of what to expect. It's also important to consider the computational cost of running these algorithms as in practise they may not be suitable due to how slow or expensive they are – it's important to strike a balance by consulting with domain experts on the resources available, the performance needed etc.

In our case, SVM seemed to perform the best after removing some attributes. Although its performance was the highest there is still a large room for improvement and because the

classifier could potentially affect a person's life (if used for prognostics or screening) it is paramount that it can be as accurate and reliable as possible.

Going forward, expanding the dataset would be a good first step, as right now it only looks at female patients and only adults. Expanding the set of features or doing feature selection with a domain expert may also lead to better success in terms of prediction. Currently, none of the models have an output to explain the relationship between features (except for 1R) which makes its practical use limited as well.