**Aim**

Our aim is to train a given dataset with K-Nearest Neighbour and the Naïve Bayes algorithms to predict weather a person might have diabetes or not given his medical profile. Moreover, evaluate a given dataset using the stratified cross validation method and comparing to the performance of other classifiers on the same dataset using the software Weka. Finally, evaluate the effect of feature selection in the performance of all the classification methods using a particular Correlation-based Feature Selection method (CFS) from Weka.

This problem is important to bestow the prediction of diabetes to start the treatment in the early stages of the disease making easier to handle its effects and hence increasing the chances of the patient to have a normal life.

**Data**

The dataset used to train and testthe algorithm was donated by Vincent Sigillito from the National Institute of Diabetes and Digestive and Kidney Diseases to our supervisors and modified for the purpose of this assignment. The dataset contains eight attributes of data taken from a group of 768 women with at least 21 years old of Pima Indian heritage. Each data also indicates weatherthe patient has diabetesor not.The eight attributes are:

1. Number of times pregnant;

2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test;

3. Diastolic blood pressure (mmHg);

4. Triceps skin fold thickness (mm);

5. 2-Hour serum insulin (mu U/ml);

6. Body mass index (weight/height in kg/m2);

7. Diabetes pedigree function;

8. Age (years).

For testing the results, we used the software Weka to make a Correlation-based Feature Selection (CFS) to take a subset of attributes that best represent the dataset, considering how good the attributes are at predicting the class and how much they correlate with the other attribute. The attributes selected were:

2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test

5. 2-Hour serum insulin (mu U/ml)

6. Body mass index (weight/height in kg/m2)

7. Diabetes pedigree function

8. Age (years)

**Results and Discussion**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ZeroR [%] | 1R [%] | 1NN [%] | 5NN [%] | NB [%] | DT [%] | MLP [%] |
| No feature selection | 65.1042 | 70.8333 | 67.8385 | 74.4792 | 75.1302 | 71.857 | 75.3906 |
| CFS | 65.1042 | 70.8333 | 69.0104 | 74.4792 | 76.3021 | 73.3073 | 75.7813 |

Accuracy table for the classifiers tested in Weka

|  |  |  |  |
| --- | --- | --- | --- |
|  | My1NN [%] | My5NN [%] | MyNB [%] |
| No feature selection | 69.2708 | 76.3021 | 79.0364 |
| CFS | 67.8385 | 74.8698 | 79.1667 |

Accuracy table for the classifiers that we generated

It is expected that the performance gets better with the complexity of the classifier, which is exactly what happened at Weka since we can consider 1NN and 1R to have roughly the same complexity, even though we expected that 1NN would be better than 1R. However, Decision Trees had a completely disappointing performance, i.e. its performance was roughly better than 1R even though its complexity is as high as MLP itself. Comparing only with ZeroR, we can see that the 5NN, NB and MLP classifiers had a nice and acceptable performance.

About our implemented algorithms, they also had increasing performances with increasing complexity, and all of them were surprisingly better than weka’s, if we compare them with the same type of classification and by comparing best performances as well. This may be due to the 10 folds cross validation, once different training sets result in different testing results. We used a totally stratified 10 folds cross validation, so all the classes have same probability in each fold as well as in the total data set. We cannot say for sure if weka stratifies its folds, but it is reasonable to assume that it does not. Comparing only with ZeroR we can assume that our results were very satisfying for this particular problem.

Considering the performance changes using CFS, the features selected seem to be intuitively better to use as attributes to solve the problem, and were supposed to increase the performance of the classifiers. All of the Weka classifiers had slightly better performances or kept with the same performance. However, in our implemented algorithms, the performances for 1NN and 5NN were worse and the performance for NB was just 0.1% better. Therefore, we can conclude that the CFS was not a good practice for our algorithm, but was for the classifiers of Weka. This may be because we generated the CFS set using the program Weka, and this CFS function in Weka might have a correlation to how its algorithms run.

**Conclusion**

By the end of these experiments, we could see that algorithms that are more complex may have better results for classification, although that is not true for all the algorithms. We can also say that for this particular problem, the CFS approach had a negative influence in our classifiers and a positive influence for the classifiers from Weka. Overall, the findings were good and we can say that our classifiers are satisfactory and reliable, if we consider a comparison with the ZeroR algorithm. Although if we think that we are dealing with a classification problem that involves life risk, 79.1667% of accuracy may not be the best of results.

To improve this experiment we propose the following future work:

* Have a better understanding of the CFS tool as to why it didn’t help much our algorithms;
* Acquire more data for better classification;
* Implementation of other classifiers like the MLP.

With special consideration on the last one, because our algorithms had far better performance than Weka’s we can infer that implementing our own MLP (or similar) we would be able to achieve even better results.

**Reflection**

Throughout this assignment, we achieved a better understanding of the implications of a couple of classifying algorithms, specifically Naïve Bayes and K Nearest Neighbour. We also had a grasp into a type of feature selection tool, and how it relates with the classifiers. By running our codes several times during the experiment, we also had an idea of the influence that stratification has in the classification. The most interesting thing we learned was that algorithms of the same type implemented differently could have completely different results. Moreover, we also learned about the range of applicability of these classifying algorithms, and also that for each problem a different classifier may have a different – seldom better- rank in the performance rating than other more complex classifiers.