COMP3308 Assignment 1 Report

Sam Turner *[312130678] and* ***James Cooper-Stanbury*** *[312154402]*

1. Aim

The purpose of this study was to build classifiers that where able to predict whether a person has diabetes or not based on a number of attributes collected from the publicly available Pima Indians Diabetes database.

This is important for a number of medical reasons. Primarily, it is useful to be able to predict whether or not someone has diabetes without having to do invasive testing.

2. Data

Data Set Used

The data set used was the Pima Indians Diabetes data that is publicly available from the UCI Machine Learning Repository at http://archive.ics.uci.edu/ml/. In its raw state it contains 768 instances described with 8 numeric attributes. Where each attribute is correlates to a patients personal characteristics and test measurements. Each patient is of Pima Indian Heritage.

The attributes are as follows:

- Number of times pregnant
- Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Diastolic blood pressure (mm Hg)
- Triceps skin fold thickness (mm)
- 2-Hour serum insulin (mu U/ml)
- Body mass index (weight in kg/(height in m)^2)
- Diabetes pedigree function
- Age (years)
- Class variable (0 or 1)

Data Preparation

To preprocess the data, a number of steps where taken:

- Add headers for each column at the top of the file.
- Change the class names from `0` and `1` to `class0` and `class1` respectively.
- Remove instances using list wise deletion where at least one attribute is invalid. Refer to *Appendix 1 - Dealing with Missing Values* for our methodology and justification.
- The attributes have then been normalised so all values are in the range $0 \leq x \leq 1$

Attribute Selection

The Correlation Based Feature Selection (CFS) method was used to generate a CSV with a reduced number of attributes. CFS operates under the assumption that there is redundant or irrelevant fields in the data. Kohavi and John formalise the definition:

Definition: A feature V_i is said to be relevant if there exists some v_i and c for which $p(V_i = v_i) > 0$ such that:

$$p(C = c|V_i = v_i) \neq p(C = c)$$

The simplest way to select a feature subset is to test each possible subset of values to find the one that minimises the error rate, but obviously, this is an exhaustive search of the space and is not optimal. There are three main categories of feature selection algorithms: *wrappers*, *filters* and *embedded* methods where the method chosen is heavily influenced by the metric used.

Wrapper Algorithms

Wrapper algorithms use a predictive model to score feature subsets. Each new subset is used to train a model which is tested on a hold-out set. The error rate on a given subset gives a score for that subset. Wrapper methods are computationally expensive as they train a new model for each subset.

Filter Algorithms

Filter algorithms use a proxy measure instead of the error rate to score a feature subset. Filter methods are chosen because they are fast to compute. Though faster, they often produce a feature set which is not tuned to a specific type of predictive model.

The attributes selected as the 'best' by Weka's CFS where:

- *Plasma Glucose Concentration*
- *Body Mass Index*
- *Diabetes Pedigree Function*
- *Age (years)*

3. Results and Discussion

Results

Accuracy on Test Set [%]

	ZeroR	1R	1-NN	5-NN	NB	DT	MLP
No feature selection	66.73%	75.75%	72.37%	76.32%	76.13%	73.87%	75.56%
CFS	66.86%	74.76%	71.37%	74.39%	77.97%	76.84%	78.53%

	My1-NN	My5-NN	My-NB
No feature selection	72.21%	75.94%	75.55%
CFS	71.74%	74.76%	77.97%

Discussion

In our testing, Weka's *ZeroR*, *Naïve Bayes*, *DT* and *MLP* where all faster when using the data generated by the CFS. For our own implementations, *Naïve Bayes* was faster using the CFS data. In general, Weka's implementations where faster or equal to our own. Though in all cases, this variance was less than 1%.

Feature selection techniques such as CFS provide three main benefits such as *improved model interpretability*, *shorter training times* and *enhanced generalisation* due to a reduced number of attributes and therefore, reduced overfitting. As a result, the algorithms that performed better are those that are more heavily affected by overfitting. The algorithms that performed worse on the CFS data where 1R, 1NN and 5NN. This is interesting as the accuracy of KNN can be significantly worsened by the presence noisy or irrelevant features, which CFS is expected to remove. It makes sense that the higher the number of relevant features, the more accurate the classifier in the case of KNN. Therefore, perhaps the features removed by Weka's CFS did not introduce significant noise into the data set.

Overall, the features chosen by the CFS seemed fairly intuitive. Features like BMI and age are all basic, and easily calculated features that often tied to various diseases (i.e. risk of cancer increases as you age, people with higher BMI have a higher risk of diabetes).

4. Conclusions

On our data set, NB was not any more accurate than KNN, however it was significantly faster to compute. Once we ran CFS, our NB suddenly became a lot more accurate than the KNN while the KNN lowered in accuracy. It is clear that NB is a much more efficient algorithm, and in our case, more accurate.

Future work could include trying a number of different attribute selection methods when removing missing data to more accurately decide on the best method.

5. Reflection

The most important thing that was learned from this assignment was the importance of adhering to a process when analysing the data. There were a number of steps involved in transforming the original data into a form that was usable for this assignment. If a process was not followed, it would be easy to use data that is in a wrong form for testing thus leading to incorrect results and conclusions. Our process involved having a folder for each step of the assignment clearly labelled and documented with its function and purpose.

6. How to Run

For in depth instructions and documentation on the methodology followed to complete this assignment please use `README.md`, located in the root directory of this assignment. To run the classifiers:

1. The classifiers folder contains a `classifiers.py` script that contains code for both our K-Nearest Neighbour (KNN) and Naïve Bayes (NB) implementations.
2. By default, this will run the NB algorithm with 10 folds. Run `python classifiers.py -h` for more information on arguments that the program will accept.

7. Bibliography & Links

- a. Kohavi and G. John. Wrappers for feature subset selection. Artificial Intelligence, special issue on relevance, 97(1–2):273–324, 1996.
- http://www.utexas.edu/cola/centers/prc/_files/cs/Missing-Data.pdf
- http://en.wikipedia.org/wiki/Feature_selection

Appendix 1 - Dealing with Missing Data

When deciding how to deal with missing attributes in the data we first had to first see if we could learn anything about the distribution of missing data. When choosing a method to deal with missing data, it is important to understand whether the values are missing at random or not. We could not find any correlation between missing values and any other attributes in the data set,

therefore, we believe the data is missing completely at random.

We considered a number of different methods such as *deletion methods*, *single imputation methods* and *model based* methods. In the end, we decided making use of the *listwise deletion* method is the most appropriate for the data set. We chose this method for its simplicity and the fact that it doesn't result in biased estimates or weaken variance for data that is missing completely at random.

We decided to delete the rows where data was missing from any column for attributes where it didn't make sense to have a zero attribute. The attributes we decided on where:

- Plasma Glucose Concentration
- Diastolic Blood Pressure
- Tricep Skin Fold Thickness
- Body Mass Index (BMI)

If we had more time, the best approach would have been to try a variety of different methods and compare the outcomes.