COMP3308 Assignment 1

Diabetes Diagnosis Using Data Mining Methods on Pima Indian Data Set

[redacted]

University of Sydney

Contents

1. Aim

Data mining is the process of selecting, exploring and modeling large amounts of data[1]. Diabetes is common disease that impose a great amount of money on the affected people and society. I try to build a model using data mining techniques based on the data set, such that I can predict if a particular person with specific details (personal and medical) has diabetes or not.

2. Data

2.1. Data Set

Data set is a publicly available at [2]. it includes personal and medical examination data for people with Pima Indian background. In particular, all patients here are females at least 21 years old of Pima Indian heritage [2]. There are 768 instances, each with 9 attributes. Attributes are:

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")

There are 2 classes, "yes" and "no" for positive diabetes and negative diabetes respectively, with each having 500 and 268 instances respectively. There are not any missing values in the data set.

2.2. Normalization

Normalization has been carried out on the data set for better analysis using Weka [3], and result has been saved as comma-separated value format in "pima.csv' file. Values for each attribute has been normalized to be between 0 to1 except for class value.

2.3. Attribute Selection

For comparison purposes, I also did feature selection on normalized data set using Weka. New data set with selected features has been saved in "pima-CFS.csv" file. To carry out the feature selection, I used "CfsSubseteval" as evaluator and "BestFirst" algorithm as the search method in Weka. Both evaluator and search methods was performed with default options.

Selected features were:

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

with the original feature's number in parentheses.

3. Result and Discussion:

3.1. Result from Weka

	Zero R	1R	1NN	5NN	NB	DT	MLP
No Feature Selection	65.1042%	70.8333%	67.8385%	74.4792%	75.1302%	71.875%	75.3906%
CFS	65.1042%	70.8333%	69.0104	74.4792%	76.3021%	73.3073%	75.7813%

Table 1 – Accuracy results in percentage from Weka

I got slightly different result each time I ran my implemented classifiers, however the differences is not much. It might be due to the floating point precision and rounding implementation. Another reason is that each time I run the program, it will generate a new folds csv file as folds are randomly generated. This will create slight differences to Weka as they will run the algorithms using different folds.

CFS feature selection interpretation:
From the features selected by Weka, we can see that Weka chosen features that have different values. For example it didn't select attribute 1 (Number of times pregnant). The reason is that, there are many same values for that one, it goes the same for other attributes which they didn't get selected.

Weka Results analysis:

- overall we can see the highest accuracy is from MLP classifier and NB with close value in second position. However in CFS, NB beats MLP and we have the highest accuracy rate in CFS using NB.
- **Rule based classifiers:** Rule based classifiers have the same accuracy rate in both data sets. Zero R model just consists of a single value: the most common class, building the model with 65.10% accuracy rate. Since the class value stay same in both data sets. Same goes with One R. One R selected "plasma glucose concentration" attribute in both data sets and it gave us 70.83% accuracy rate.
- **KNN:** We have a better result for 5NN rather than 1NN in both data sets. Reason is that it can check with more neighbors to build the model, so we get more accurate result using 5NN rather than 1NN. Having said that, it's better to use odd numbers rather than even numbers in this classifier. We have a better accuracy rate for feature selected data set though.
- **NB:** accuracy rate in CFS data set is slightly better than the original data set. In Naive Bayes, we calculate probabilities of features with independence assumption between them. The reason we have a slightly better rate on CFS is that we don't have those features with same values in most cases. As a result probabilities for those feature didn't have much effect on our accuracy and by removing them we got a better result.
- **DT:** we have a better accuracy rate in CFS. Reason is that we eliminate those features with same value or similar range. So tree classifier, needed to branch less than the tree in original data set, and we got a slightly better result.
- **MLP:** we have almost the same result in both data sets. Because considering what feature have been removed, they didn't have much effect on neural network (nodes) or values here.

3.2. “MyProgram” Results Analysis:

Results from “MyProgram”

	My1NN	My5NN	MyNB
No Feature Selection	67.4504%	74.8632%	74.8769%
CFS	69.2840%	75.0909%	76.8181%

Table 2 – Accuracy results in percentage from "MyProgram"

My implemented classifiers has almost the same result with Weka's result. There are a bit differences in accuracy rate, that might be because of different implementation of cross-validation between mine and Weka's version, or it might just be the floating point values rounding procedure. I got a better result in feature selected data set.

4. Conclusion

provided data set has very informative data on diagnosing diabetes. They can be used by a medical staff to predict if a person has a high risk of having diabetes or not. We use CFS (Correlation-based feature selection) to reduce our data set and focus on more relevant attributes. As a result we got a slightly better accuracy rate in CFS data set. Overall, for a better model training, it's better to have bigger data with more instances so that we can build a more efficient model.

5. Reflection:

I learned more about artificial intelligence and how it can help to solve or improve our problems. This problems might look trivial, however the opportunities for further work, has no limit. I also learned how to implement a classifier, prepare the data sets, and the procedure on how to perform an experiment, how to analysis the result either in a right way or wrong way.

6. Instruction on how to run the program

I used java to implement my classifiers. There are 4 classes and one main class. Each classifier has their own class.
To run the program normally (after compiling)

```
java MyProgram {path to training data set} {path to test data set} {classifier name}
```

for example: java MyProgram pima.csv example.csv NB

to run the program to get the statistics:

```
java MyProgram {path to training data set} {path to test data set} {classifier name} {path to data set}
```

for example: java MyProgram pima.csv example.csv NB pima.csv.

This will generate the folds csv file. You need to run the program for normal data set and feature selected data set to get the result.

7. References

[1]. Arwa et.al 2014, Using Prediction methods in data mining for diabetes diagnosis.
[2]. https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes
[3].Hall, M.et.al 2009, The WEKA Data Mining Software

8. List of Tables

9. List of Figures

there are no figures in this paper