

Assignment2

Xiunan Fang
45020566

Aim

The aim of this assignment is to implement the Naïve Bayes and Decision Tree algorithms and evaluate different classifier using Weka. The effect of feature selection using CFS method from Weka is about to investigated. *Machine Learning (ML)* is the area of AI that is concerned with writing computer programs that can learn from examples. *ML* is the core of AI and practice of those algorithms is very beneficial.

Data

The dataset I use for this assignment is the modified Pima Indians Diabetes Database. There are 8 attributes with 768 instances, 2 classes – ‘yes’ or ‘no’.

8 attributes are:

1. Number of times pregnant
2. Plasma glucose concentration
3. Diastolic blood pressure
4. Triceps skin fold thickness
5. 2-Hour serum insulin
6. Body mass index
7. Diabetes pedigree function
8. Age

Correlation-based feature selection (CFS) is an algorithm for selecting a subset from the complete feature set. It can identify and screen irrelevant, redundant, and noisy features. In most cases, we can get equal or better accuracy using the reduced feature set selected by CFS than using the complete feature set.

By using weka, the attribute selection output of ‘pima.csv’ is shown in Figure 1 below:

```

Search Method:
  Best first.
  Start set: no attributes
  Search direction: forward
  Stale search after 5 node expansions
  Total number of subsets evaluated: 38
  Merit of best subset found:    0.173

Attribute Subset Evaluator (supervised, Class (nominal): 9 class):
  CFS Subset Evaluator
  Including locally predictive attributes

Selected attributes: 2,5,6,7,8 : 5
  plasma_glucose_concentration
  2-Hour_serum_insulin
  body_mass_index
  diabetes_pedigree_function
  age

```

Figure 1 attribute selection output by Weka

We can see that the selected attributes are plasma_glucose_concentration, 2-Hour_serum_insulin, body_mass_index, diabetes_pedigree_function and age.

Similarly, we can apply CFS to the discretized data, the attribute selection output of 'pima-discretised.csv' is the same as the numerical data.

Results and discussion

We can use Weka to evaluate and compare different classifiers. By selecting 10-fold stratified cross validation and running the following algorithms: ZeroR, 1R, k-Nearest Neighbor, Naïve Bayes, Decision Tree, Multi-Layer Perceptron and Support Vector Machine, the accordingly accuracy for different classifiers is shown in Table below.

Classifier accuracy

Numeric Data	ZeroR	1R	1NN	5NN	NB	MLP	SVM	MyNB
No feature selection	65.10%	70.83%	67.84%	74.48%	75.13%	75.39%	76.30%	74.28%
CFS	65.10%	70.83%	69.01%	74.48%	76.30%	75.58%	76.69%	75.71%

Table 1 Classifier accuracy using numeric data

Nominal Data	DT unpruned	DT pruned	MyDT
No feature selection	75.00%	75.39%	71.71%

CFS	79.43%	79.43%	77.08%
-----	--------	--------	--------

Table 2 Classifier accuracy using nominal data

DT diagrams

The DT diagram for each of the three DT classifiers-Weka DT-pruned, Weka DT-unpruned, myDT is shown below. As my DT tree is very large., the diagram is shown as a text-based diagram

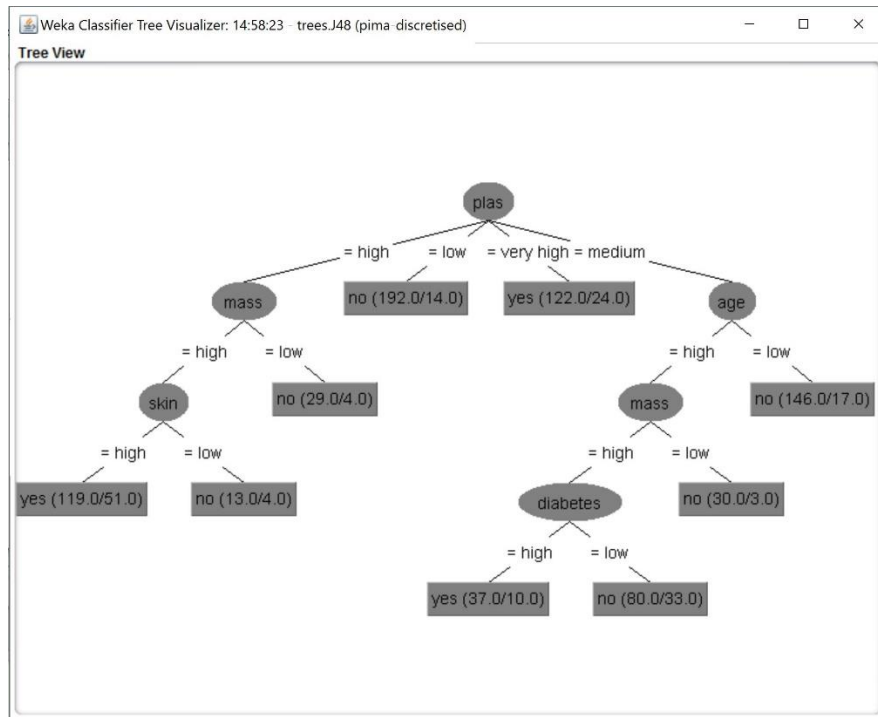


Figure 2 Tree view of DT_pruned

Figure 3 Tree view of DT_unpruned

And the txt below is the DT tree I build.

[illegible]

```

'low': {'skin': {'high': {'insu': {'high': {'pedi': {'high': 'no',
'low': {'age': {'high': {'pres': {
'high': {'preg': {'high': 'no',
'low': 'no'}},
'low': 'no',
'medium': 'no',
'very high': 'no'}},
'low': {'pres': {
'high': 'no',
'low': {'preg': {
'high': 'yes',
'low': 'yes'}},
'medium': 'yes',
'very high': 'yes'}}}}}}},
'low': {'pedi': {'high': 'yes', 'low': 'no'}}},
'low': 'no'}}},
'low': {'mass': {'high': {'insu': {
'high': {'age': {'high': {'pedi': {'high': {'pres': {'high': {'preg': {'high': {'skin': {'high': 'yes',
'low': 'no'},
'low': {'skin': {'high': 'no',
'low': 'yes'},
'low': 'yes',
'medium': 'yes',
'very high': 'yes'}}},
'low': {'skin': {'high': {'preg': {'high': {'pres': {'high': 'no',
'low': 'no'},
'low': {'pres': {'high': 'no',
'low': 'no'}}},
'low': 'no'}}},
'low': {'pres': {'high': 'no',
'low': {'skin': {'high': {'pedi': {'high': 'no',
'low': {'preg': {'high': 'no',
'low': 'no'},
'medium': 'no',
'very high': 'no'}},
'low': 'no',
'medium': 'no',
'very high': 'no'}}},
'low': 'no',
'medium': 'no',
'very high': 'no'}}},
'low': {'pres': {'high': {'age': {'high': 'no',
'low': {'skin': {'high': {'pedi': {'high': 'yes',
'low': {'preg': {'high': 'no',
'low': 'no'},
'medium': 'no',
'very high': 'no'}},
'low': 'no',
'medium': 'no',
'very high': 'no'}}},
'low': 'no',
'medium': 'no',
'very high': 'no'}}},
'medium': {'age': {'high': {'mass': {'high': {'pedi': {'high': {'preg': {'high': 'yes',
'low': {'skin': {'high': {'pres': {
'high': {'insu': {'high': 'yes',
'low': 'yes'}},
'low': {'insu': {'high': 'yes',
'low': 'yes'}}},
'low': 'yes',
'medium': 'yes',
'very high': 'yes'}}},
'low': {'insu': {'high': {'pres': {
'high': {'preg': {'high': {'skin': {'high': 'no',
'low': 'no'},
'low': {'skin': {'high': 'no',
'low': 'yes'}}},
'low': {'skin': {'high': {
'preg': {'high': 'yes', 'low': 'yes'}},
'low': {'preg': {'high': 'no',
'low': 'no'},
'medium': 'no',
'very high': 'no'}},
'medium': 'no',
'very high': 'no'}},
'low': 'no',
'very high': 'no'}}},
'low': {'pres': {'high': {'preg': {'high': 'no',
'low': {'pedi': {'high': 'no',
'low': {'pedi': {
'high': {
'insu': {
'high': 'no',
'low': 'no'}}},
'low': {'preg': {'high': 'yes',
'low': {'skin': {'high': 'no',
'low': {'insu': {
'high': {'pedi': {
'high': 'no',
'low': 'no'},
'low': 'no'}}},
'medium': 'no',
'very high': 'no'}}},
'low': {'mass': {'high': {'skin': {
'high': {'preg': {'high': {'pres': {'high': {'insu': {'high': {'pedi': {'high': 'yes',
'low': 'yes'}},
'low': 'yes'}},
'low': {'pedi': {'high': {'pres': {'high': {'insu': {'high': 'no',
'low': 'no'},
'low': {'insu': {'high': 'yes',
'low': 'yes'}}},
'low': {'pres': {
'high': {'insu': {'high': 'no', 'low': 'no'},
'low': {
'insu': {'high': 'no', 'low': 'no'}}}}},
'low': {'pedi': {'high': {'pres': {'high': 'no',
'low': {'insu': {
'high': {'preg': {'high': 'no', 'low': 'no'},
'low': 'no',

```

```

        'medium': 'no',
        'very high': 'no']],
        'medium': 'no',
        'very high': 'no']],
        'low': 'no'))],
        'low': {'pedi': {'high': {'insu': {'high': 'no',
            'low': {'pres': {'high': 'no',
                'low': {'preg': {'high': 'yes',
                    'high': 'yes',
                    'low': {'skin': {'high': 'yes',
                        'low': 'yes'}}}},
                    'medium': 'yes',
                    'very high': 'yes'}}}},
                'low': 'no'}})]},
        'very high': {'insu': {'high': {'mass': {'high': {'preg': {'high': {'pedi': {'high': 'yes',
            'low': {'pres': {'high': {'skin': {'high': {'age': {'high': 'yes',
                'low': 'yes'}}}},
                'low': 'yes'}}},
                'low': {'skin': {'high': {'age': {'age': {'high': 'yes',
                    'low': 'yes'}}}},
                    'low': 'yes'}})]},
            'low': {'age': {'high': {'pedi': {'high': {'skin': {'high': {'pres': {'high': 'yes',
                'low': 'yes'}}},
                'low': 'yes'}}},
                'low': 'yes',
                'medium': 'yes',
                'very high': 'yes'}}},
            'low': {'pres': {'high': {'skin': {'high': {'age': {'high': 'yes',
                'low': 'yes'}}},
                'low': {'skin': {'high': 'yes',
                    'low': 'yes'}}}})},
            'low': {'pedi': {'high': 'yes',
                'low': {'skin': {'high': {'pres': {'high': 'yes',
                    'low': 'yes'}}},
                'low': {'pres': {'high': 'yes',
                    'low': 'no'}}},
                'medium': 'yes',
                'very high': 'yes'}})]},
        'low': {'age': {'high': {'skin': {'high': {'preg': {'high': {'pedi': {'high': 'yes',
            'low': 'yes'}}},
            'low': 'yes',
            'medium': 'yes',
            'very high': 'yes'}}},
            'low': {'pres': {'high': {'pedi': {'high': 'yes', 'low': 'yes'}}},
            'low': {'pedi': {'high': 'yes', 'low': 'yes'}})]},
            'low': {'pres': {'high': {'skin': {'high': 'no',
                'low': {'preg': {'high': 'yes',
                    'low': 'yes'}}},
                'low': {'pedi': {'high': 'yes',
                    'low': 'yes'}}}},
                'medium': 'yes',
                'very high': 'yes'}}},
            'low': 'no',
            'medium': 'yes',
            'very high': 'yes'}})]},
        'low': {'pedi': {'high': 'yes', 'low': 'no'}})]}

```

Discussion

When using the numeric training data, it is clear that ZeroR classifier has the lowest accuracy rate. And the performance of 1R classifier, 1NN classifier is not that good as well. The reason of the low accuracy lies in the simplicity of the algorithm, but thanks to that the time taken to build model is very short.

We can see that the accuracy improves a lot when using 5NN comparing to 1NN, which confirms the fact that K-Nearest Neighbor is very sensitive to the value of k.

Naïve Bayes classifier has a good accuracy and relatively simple approach, which makes it very effective. Multi-Layer Perceptron is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. It is shown that the accuracy of Naïve Bayes and Multi-Layer Perceptron are similar, and by using feature selection, NB even outperforms MLP. However, MLP is a very time consuming method.

Support vector machine is supervised learning model with associated learning algorithm that analyze data used for classification and regression analysis. It has the highest accuracy among all the classifier.

The Naïve Bayes classifier I build has the accuracy of 74.28% using my implementation of 10-fold stratified cross-validation, which is a little lower than the NB classifier built by Weka but still reasonable. Figure 4 and Figure 5 below show the according accuracy for 10 folds of NB.

accForNB - Dictionary (10 elements)

Key	Type	Size	
0	float	1	0.7272727272727273
1	float	1	0.7402597402597403
2	float	1	0.7489177489177489
3	float	1	0.7435064935064936
4	float	1	0.7350649350649351
5	float	1	0.7402597402597403
6	float	1	0.7421150278293135
7	float	1	0.7483766233766234
8	float	1	0.75
9	float	1	0.7526041666666666

Figure 4 10-fold accuracy for NB

accForNB_CFS - Dictionary (10 elements)

Key	Type	Size	
0	float	1	0.7532467532467533
1	float	1	0.7597402597402597
2	float	1	0.7705627705627706
3	float	1	0.7532467532467533
4	float	1	0.7428571428571429
5	float	1	0.7532467532467533
6	float	1	0.7606679035250464
7	float	1	0.7613636363636364
8	float	1	0.7557803468208093
9	float	1	0.7604166666666666

Figure 5 10-fold accuracy for NB-CFS

When comparing accuracy between no feature selection data and CFS data, classifiers basically have better accuracy using CFS data except for zeroR, oneR and 5NN.

Comparing performance of the Weka's DT unpruned classifier and DT pruned classifier, the performance of pruned DT classifier is a little higher but not significantly. Because pruning has the benefit of avoiding overfitting. My Decision Tree classifier has lower accuracy than Weka's one, but the accuracy improves a lot when using CFS data.

Figure 6 and Figure 7 below show the according accuracy for 10 folds of DT.

accForNB - Dictionary (10 elements)

Key	Type	Size	
0	float	1	0.7272727272727273
1	float	1	0.7337662337662337
2	float	1	0.7272727272727273
3	float	1	0.6948051948051948
4	float	1	0.6909090909090909
5	float	1	0.696969696969697
6	float	1	0.7142857142857143
7	float	1	0.724025974025974
8	float	1	0.7283236994219653
9	float	1	0.7330729166666666

Figure 6 10-fold accuracy for DT

accForNB_CFS - Dictionary (10 elements)

Key	Type	Size	
0	float	1	0.7532467532467533
1	float	1	0.7597402597402597
2	float	1	0.7705627705627706
3	float	1	0.7532467532467533
4	float	1	0.7428571428571429
5	float	1	0.7532467532467533
6	float	1	0.7606679035250464
7	float	1	0.7613636363636364
8	float	1	0.7557803468208093
9	float	1	0.7604166666666666

Figure 7 10-fold accuracy for DT-CFS

CONCLUSION

From what have been don, we can draw the conclusion that CFS can improve the performance of the classifier in most cases. 10-fold Cross Validation can be used to compare the performance of different classifier. Decision Tree as a more completed method has better performance than Naïve Bayes. But Naïve Bayes as a classifier shows its good ability in classification, and even outperforms more sophisticated learning methods sometimes, which inspires me that we should always try the simple method first!

Suggest future work:

As we only build the simplest Decision Tree without any pruning, there might be some overfitting problem. And as shown in the Discussion part, the accuracy of myNB and myDT cannot outperform the Naïve Bayes and Decision Tree classifier build using Weka. Further work can focus on the improvement on the accuracy of the classifier. Besides, the comparison of different classifier is not that specific, more evaluation work can be done such as the t-paired significance test, and other performance measures such ass recall, precision and F1 are suggested.

Reflection

In this assignment, I implement the Naïve Bayes and Decision Tree algorithms and use the Pima Indian Diabetes dataset to evaluate them. During the whole process, I learned a lot about the logic of naïve bayes and decision tree and have a better understanding of machine learning in artificial intelligence. Since I work on this assignment alone, I also learned to have a better time management. I think I benefit a lot from this assignment.