ASSIGNMENT 2: CLASSIFICATION

Lab-report

460452492
460452481

Introduction

As solving classification problems is making significant progress in artificial intelligent area in recent years, it is necessary for every artificial intelligent learner to be aware of two of the introductory supervised classifiers: K-Nearest Neighbor classifier and Naïve Bayes classifier.

The aim of this study is to understand and implement the basic processing of K-Nearest Neighbor algorithms and Naïve Bayes algorithms on a real dataset. In this study, the Pima Indian Diabetes dataset, which would be introduced in detail in the next section, is used as the training dataset for these two classify algorithms to solve a real-life problem, that is to predict if the example with given attributes shows signs of diabetes or not.

the 10-fold stratified cross-validation would be implemented to evaluate the performance of these two classifiers as well as other classifiers on the same dataset. Furthermore, by implementing Correlation-based Feature Selection(CFS) from Weka to select feature on the same dataset, the influence of feature selection on different classifiers would be researched and discussed to give some main findings and future work.

As for the importance of this study, although extensive research in all aspects of diabetes (diagnosis, therapy, etc.) has led to the generation of huge amounts of data, they do not provide any kind of analysis, interpretation or extraction of knowledge. Considering the remarkable characters of machine learning techniques, it is believed that applying classifiers-based machine learning in diabetes research would be a key approach to utilizing large volumes of available diabetes-related data for extracting knowledge. With valuable knowledge extracted from diabetes research with respect to prediction and diagnosis, classifiers-based machine learning model could help to improve the efficacy of these techniques in diabetes treatment such as analysis symptoms, early detection of diseases, prevention of medical errors and medical description.

Data

Data set

According to *pima-Indians-diabetes.names*, The dataset used for this study is the Pima Indian Diabetes dataset from National Institute of Diabetes and Digestive and Kidney Diseases, it contains 768 instances described by 8 numeric attributes and 2 classes - yes and no.
(the 6 groups of original data set with heading is shown in *table1*).

Table1: original dataset with heading

Attribute Group	01	02	03	04	05	06	07	08	class
1	6	148	72	35	155	33.6	0.627	50	yes
2	1	85	66	29	155	26.6	0.351	31	no
3	8	183	64	29	155	23.3	0.672	32	yes
4	1	89	66	23	94	28.1	0.167	21	no
5	0	137	40	35	168	43.1	2.288	33	yes
6	5	116	74	29	155	25.6	0.201	30	no

the original data set has eight attributes which are all numeric values, for each attribute:
attribute 01: number of times pregnant;
attribute 02: plasma glucose concentration a 2 hours in an oral glucose tolerance test;
attribute 03: diastolic blood pressure;
attribute 04: triceps skin fold thickness;
attribute 05: 2-Hour serum insulin;
attribute 06: body mass index;
attribute 07: diabetes pedigree function and
attribute 08: age. All these attributes are factors of diabetes.

With comparison of the eight attributes the original data, it is shown that some of the attributes are in the magnitude of 1 while some of the attributes are above 100. Since different attributes are measured on different scales, when calculating the distance between 2 examples with the KNN algorithm, the effect of the attributes with the smaller scale will be less significant than those with the larger. Consequently, it is necessary to normalize the values of each attribute to make sure they are in the range [0,1]. After normalization, we can eliminate the influence of the different scaled attributes to the algorithm. Table 2 shows the

normalized data by using Weka function.

Table2: normalized dataset with heading

Attribute Group	01	02	03	04	05	06	07	08	class
1	0.352941	0.670968	0.489796	0.304348	0.169471	0.314928	0.234415	0.483333	yes
2	0.058824	0.264516	0.428571	0.23913	0.169471	0.171779	0.116567	0.166667	no
3	0.470588	0.896774	0.408163	0.23913	0.169471	0.104294	0.253629	0.183333	yes
4	0.058824	0.290323	0.428571	0.173913	0.096154	0.202454	0.038002	0	no
5	0	0.6	0.163265	0.304348	0.185096	0.509202	0.943638	0.2	yes
6	0.294118	0.464516	0.510204	0.23913	0.169471	0.151329	0.052519	0.15	no

Attribute selection

Considering overfitting may arise with high dimensional data for all classifiers, feature selection should be applied to reduce dimensionality in the dataset. In this study, CFS(Correlation-based feature selection) function from Weka is utilized as the method for selecting the best subsets by ranking the accuracy of using individual features to predict the class as well as the need to be uncorrelated with other attributes.
After applying CFS in the Weka, the attribute 02, attribute 05, attribute 06, attribute 07 and attribute 08 are selected as the best subsets of features(as shown in table3).

Table3: normalized dataset after CFS

Attribute Group	02	05	06	07	08	class
1	0.670968	0.169471	0.314928	0.234415	0.483333	yes
2	0.264516	0.169471	0.171779	0.116567	0.166667	no
3	0.896774	0.169471	0.104294	0.253629	0.183333	yes
4	0.290323	0.096154	0.202454	0.038002	0	no
5	0.6	0.185096	0.509202	0.943638	0.2	yes
6	0.464516	0.169471	0.151329	0.052519	0.15	no

Results and discussion

Accuracy of Different Algorithms in Weka

Results

After implementing different algorithms listed as following in Weka, the accuracy results of these eight different algorithms, e.g. ZeroR, 1R, 1NN,

3NN, NB, DT, MLP and SVM, without feature selection and with correlation-based feature selection feature are presented in table 4.

Table4: accuracy results of weka

	ZeroR	1R	1NN	3NN	NB	DT	MLP	SVM
No feature selection	65.1042%	70.8333%	67.8385%	72.6563%	75.1302%	71.875%	75.3906%	76.3021%
CFS	65.1042%	70.8333%	69.0104%	73.3073%	76.3021%	71.875%	75.7813%	76.6927%

Discussion

Comparing the accuracy of these eight classifiers with and without feature selection in Figure 1, it shows that all the classifiers implementing CFS have the higher or at least the same accuracy as all the classifiers which use the original data.

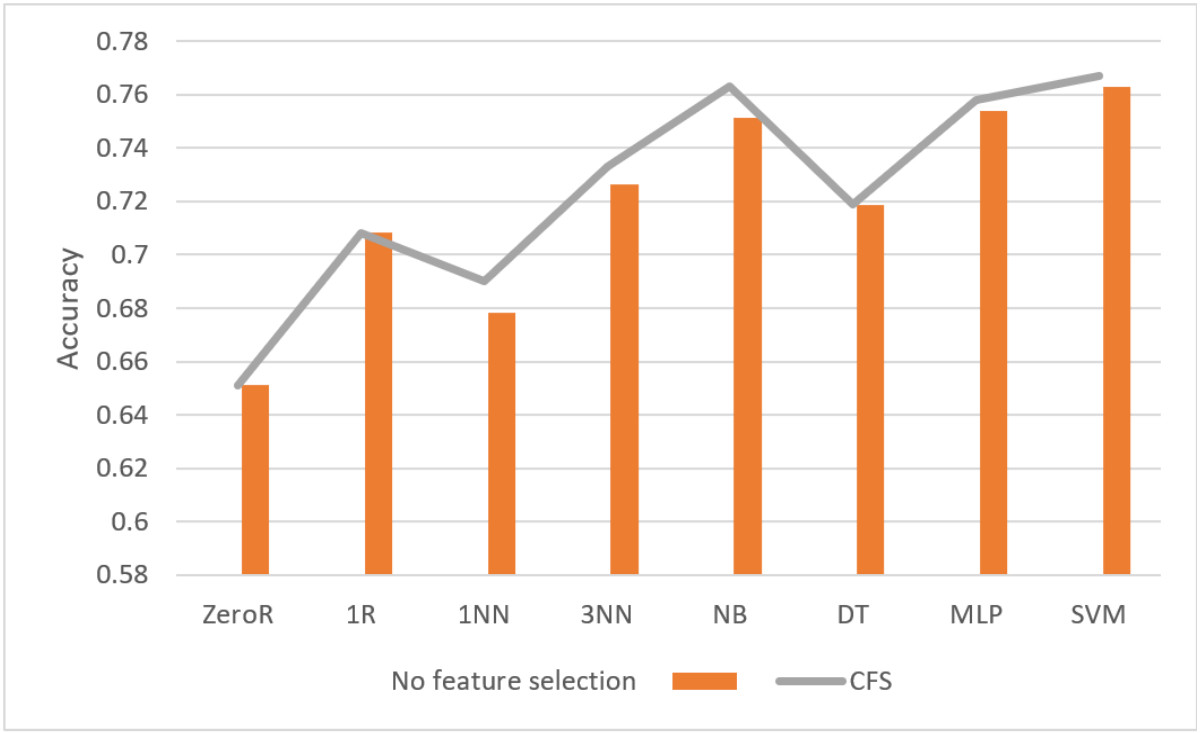

Figure 1: accuracy results of weka

As shown in the Figure 1, the accuracy of ZeroR, 1R and DT algorithms stays the same compared with CFS processed data, which indicates that there is no difference in reducing dimensionality for those classifiers. This phenomenon is mainly because of the mechanism of these three algorithms to build a classifier. As the ZeroR classifier simply uses the majority class category as the predictor, in this case, only the class data is selected to build the classifier by ZeroR algorithm.

The 1R algorithm only utilizes a single attribute with the highest accuracy to build the classifier. In this case, because CFS is used to pick up the attributes which have the better performance, this means CFS will not influence the final accuracy.

DT algorithm uses the entropy which measures the purity of a set of examples with respect to their class to choose the attribute until all the testing data have been classified. In this case, the accuracy of DT algorithm is not influenced by CFS may because that the DT algorithm has fully classified all data before using the 5^{th} attribute.

For classifier 1NN, 3NN, NB, MLP and SVM, the accuracy of using CFS processed data is slightly higher than using original data. Considering the mechanism of these algorithms, k-Nearest Neighbour algorithm is to use the k nearest training data example to identify new examples by remembering all the training data, while Naïve Bayes Algorithm uses all attributes and allows them to make contributions to the decision that are equally important and independent of one another. Based on this, in 1NN, 3NN and NB algorithms, they need to process all the attributes equally, therefore larger values of attributes would cause the noise on the classification, since high dimensional data causes problems for all classifiers - overfitting. Because CFS selects the better performance attributes, these two algorithms will be less influenced by the bad behavior attributes. However, the accuracy of 3NN seems to be higher than 1NN, because 3NN algorithm compares the 3 nearest training data to classify the new data while 1NN algorithm only uses 1 nearest training data to classify the new data.

For MLP algorithm, too many attributes may cause noise in the training examples which could lead to overfitting as well. With CFS, MLP algorithm could reduce the number of attributes, which is considered to increase the accuracy of MLP classifier.

For SVM algorithm, the large number of attributes will increase the computation difficulty and also add noise to the classifier. Using CFS processed data can eliminate the attributes with bad behavior to improve the performance of SVM classifier.

Accuracy of self-coding

Result

The accuracy results of 1NN, 3NN and Native Bayes algorithms with and without feature selection are presented in table 5.

Table 5: accuracy results of self-coding

	My1NN	My3NN	MyNB
No feature selection	68.3527%	73.6910%	75.2614%
CFS	68.2348%	73.9576%	76.1705%

To evaluate the performance of the classifiers mentioned above, our program implement the 10-fold stratified cross-validation.
To generate the the 10-fold stratified data based on given training data in required format, run the following command:

```
python pima.csv write_stratification
```

To reproduce the results from our code, run the following commands and the results will be printed.

for accuracy results of 1NN without CFS:

```
python MyClassifier.py pima.csv cross_validate 1NN
```

for accuracy results of 1NN with CFS:

```
python MyClassifier.py pima-CFS.csv cross_validate 1NN
```

for accuracy results of 3NN without CFS:

```
python MyClassifier.py pima.csv cross_validate 3NN
```

for accuracy results of 3NN with CFS:

```
python MyClassifier.py pima-CFS.csv cross_validate 3NN
```

for accuracy results of NB without CFS:

```
python MyClassifier.py pima.csv cross_validate NB
```

for accuracy results of NB with CFS

```
python MyClassifier.py pima-CFS.csv cross_validate NB
```

Discussion

In figure 2, it is indicated that the accuracy of My3NN and MyNB with CFS is higher than the accuracy of these two algorithms without CFS. However, the accuracy of My1NN with CFS is slightly less than My1NN without CFS.

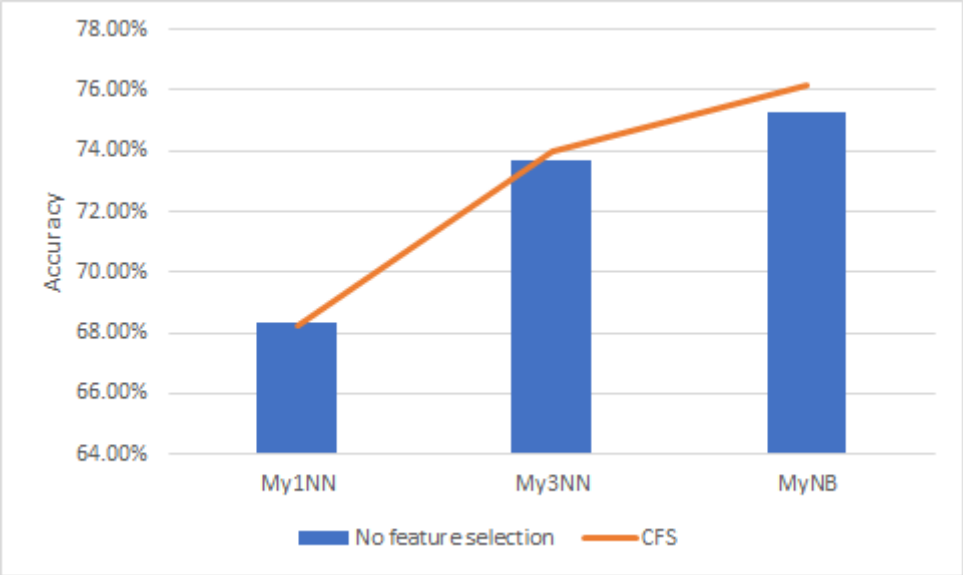

Figure 2 : accuracy results of self-coding

For My1NN, as it is shown that the accuracy is decreased by 0.1179%. This tiny difference may be caused by several reasons as following:

- the training data is not stratified and shuffle enough. As we only ensure that each class is represented with approximately equal proportions in both data sets, there might be a need to shuffle each fold to reduce the effect of random variation in choosing the folds. In addition, considering the training data set is small(instances are less than 1000), high computational cost could be avoided, so maybe the number of folds(in this case, it's 10) could be larger to build an accurate classifier as there would be random sampling.
- As this unexpected result arises from 1NN algorithm, there might be a possibility that k value(in this case, it's 1) is too small to build an accurate classifier, and this could reflect in the evaluation procedure as well.
- Last but the least, the crucial problem may be the low dimension of training data, there are only 8 attributes in this study and it is just a

little bit higher compared to 6, as 6 is considered as the threshold to define low and high dimensions. So the size of attributes could still be too small to indicate improvements in accuracy by using CFS.

For My3NN and MyNB classifiers, because of using CFS which reduces the number of attributes by eliminating the bad behavior data, the accuracy is increased by applying CFS.

With comparing the results of My1NN and My3NN, the difference between these two algorithms is that 1NN uses the one nearest training sample to classify the test sample class, but 3NN uses three nearest training ample to classify the test sample class. The reason for why CFS improves the accuracy of My3NN rather than My1NN is stated as above.

Conclusion

By applying CFS, the attributes are selected based on the accuracy of individual predictive ability of each attributes, subsets of features that are highly correlated with the class while having low intercorrelation are preferred. It can avoid the noise of the training data and can also reduce the time of computation.

The algorithms like ZeroR and 1R which only utilize one attribute or class to build classifier will not be influenced by CFS. However, the other algorithms which need to remember all the attributes, for example, 1NN, 3NN, NB, DT, MLP, and SVM, applying CFS to select attributes will increase the accuracy.

In conclusion, after comparing all the accuracy results in the algorithms mentioned above, it is believed that CFS would be a key approach to improving the accuracy when applying machine learning techniques for training data with high dimensions. Also, come back to the topic of this study, we believe that machine learning techniques would play a more and more important role in medical field with its accuracy superiority compared to inefficient manual diagnosis, in this case, diabetes prediction and diagnosis is a significant convincing example.

Reflection

With writing the code of KNN and NB algorithms, we get the chance to implement the knowledge of algorithms into practice. In the Weka part, it also practices our ability to use software for analysing data which can help us in further develop proficiency in software development.

The another important thing we learnt from this assignment is that the accuracy can not only be improved by using better algorithms, but also by preprocessing data in a right way, like normalization and CFS. In addition, we also train ourselves to work as a team with the ability to communicate and report writing, it is important to learn how to work as a team member for our future career.

Reference

[1]Ioannis Kavakiotis,a,b,∗ Olga Tsave,c Athanasios Salifoglou,c Nicos Maglaveras,b,d Ioannis Vlahavas,a and Ioanna Chouvardab,d (2017 Jan 8). *Machine Learning and Data Mining Methods in Diabetes Research.* Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5257026/

[2]Frank and Hall. COMP3308/3608, Lecture 5 ARTIFICIAL INTELLIGENCE *Introduction to Machine Learning.K-Nearest Neighbor. Rule-Based Algorithms: 1R*