COMP3308 Assignment 2 Report

Shoudi Huang
shua9875@uni.sydney.edu.au
500478204

Lehan Yang
lyan3310@uni.sydney.edu.au
500136245

1 Aim

Our study investigated different algorithms as classifiers to classify the presence or absence of diabetes on the Pima Indian Diabetes dataset. For the models, we use weka software to concisely compare the classification performance of several traditional machine learning models by accuracy, True Positive Rate and Precision metrics. And we also evaluated and compared the performance of KNN and Naive Bayesian classifiers that implemented from scratch. The advantages and disadvantages of using CFS for feature selection are also discussed, and the classification performance of each model is compared before and after using feature selection. Thus, the actual performance of various models and data enhancement methods on this dataset is briefly analyzed to provide empirical support for future data for model selection.

2 Data

2.1 Dataset

The "Pima Indian Diabetes" data set was modified by replacing missing values with averages and changing class to nominal values in March 2015. This data set contains 8 numeric attributes, 1 class attribute, and 768 instances. Among them, all numeric attributes are personal medical and physiological data that may be potentially related to diabetes, and the class attribute is whether the sample is tested positive for diabetes (Table 1). In addition, all instances in this data set were female patients at least 21 years old, of which 500 were negative for diabetes, and 268 were positive.

Attribute	Mean	SD
1. Number of times pregnant	3.8	3.4
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test	121.7	30.4
3. Diastolic blood pressure (mm Hg)	72.4	12.1
4. Triceps skin fold thickness (mm)	29.1	8.8
5. 2-Hour serum insulin (mu U/ml)	155.3	85.0
6. Body mass index (weight in kg/(height in m)^2)	32.5	6.9
7. Diabetes pedigree function	0.5	0.3
8. Age (years)	33.2	11.8
9. Class variable ("yes" or "no")		

Table 1: "Pima Indian Diabetes" Dataset

2.2 Attribute Selection

Traditional machine learning relies more on the quality of data compared to deep learning, and the higher the quality of data, the higher the theoretical performance tends to be. And modern deep-depth models tend to be more robust and more resistant to perturbations in the data. Feature

selection is the process of removing features that have smaller correlation to the class, and after than the features can drive up the performance of the model.

In our study, we chose Correlation-based Feature Selection [1], which considers that the smaller the correlation between features and features, the better the differentiation. The larger the correlation between features and classes, the better, and the easier it is to map features to classes.

We have the heuristic equation:

$$\text{Merit}_{\ s} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{1}$$

Where the S is the feature subset with k features. The $\overline{r_{cf}}$ is the averaged feature to class correlation, while $\overline{r_{ff}}$ is the inter-feature correlation. Then here we use best first search to search the highest heuristic combinations of features.

In the experiment to perform the CFS, we used the weka software, and got the following attributes in Table 2 as our new dataset.

Selected Attribute
1. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
2. 2-Hour serum insulin (mu U/ml)
3. Body mass index (weight in kg/(height in m)^2)
4. Diabetes pedigree function
5. Age (years)
6. Class variable (”yes” or ”no”)

Table 2: ”Pima Indian Diabetes” Dataset After CFS

3 Results and discussion

3.1 Result

Performance of Machine Learning Models in ”weka”

			ZeroR	1R	1NN	5NN	NB	DT	MLP	SVM	RF
No Feature Selection	Accuracy		65.1%	70.8%	67.8%	74.5%	75.1%	71.7%	75.4%	76.3%	74.9%
	Yes	TPR	0.0%	47.4%	54.5%	60.1%	60.4%	63.4%	65.7%	53.0%	62.7%
		Precision	None	60.5%	53.9%	64.4%	65.6%	58.8%	64.5%	71.7%	64.4%
	No	TPR	100.0%	83.4%	75.0%	82.2%	83.0%	76.2%	80.6%	88.8%	81.4%
		Precision	65.1%	74.7%	75.5%	79.3%	79.7%	79.5%	81.4%	77.9%	80.3%
CFS	Accuracy		65.1%	70.8%	69.0%	74.5%	76.3%	73.3%	75.8%	76.7%	75.9%
	Yes	TPR	0.0%	47.4%	55.6%	59.7%	57.1%	63.8%	64.2%	53.0%	61.9%
		Precision	None	60.5%	55.6%	64.5%	69.5%	61.3%	65.6%	72.8%	66.7%
	No	TPR	100.0%	83.4%	76.2%	82.4%	86.6%	78.4%	82.0%	89.4%	83.4%
		Precision	65.1%	74.7%	76.2%	79.2%	79.0%	80.2%	81.0%	78.0%	80.3%

Table 3: Performance of Machine Learning Models in ”weka”. Here we donate TPR for the True Positive Rate, Yes stands for Test Positive for diabetes, and No stands for Test Negative for diabetes

Performance of Self-built Machine Learning Models

			My1NN	My5NN	MyNB
No Feature Selection	Accuracy		68.4%	75.4%	75.3%
	Yes	TPR	55.6%	62.3%	60.8%
		Precision	54.6%	65.5%	65.9%
	No	TPR	75.2%	82.4%	83.0%
		Precision	76.0%	80.3%	79.9%
CFS	Accuracy		68.2%	75.1%	76.0%
	Yes	TPR	54.9%	62.0%	56.0%
		Precision	54.4%	65.1%	69.7%
	No	TPR	75.4%	82.2%	86.8%
		Precision	75.7%	80.1%	78.7%

Table 4: Machine Learning Models Performance. Here we donate TPR for the True Positive Rate, Yes stands for Test Positive for diabetes, and No stands for Test Negative for diabetes

3.2 Discussion

In Table 1, the means and standard deviations of numeric attributes in the "Pima Indian Diabetes" data set are pretty different. Therefore, normalization of all numeric attributes is necessary during data pre-processing. That will prevent a significant reduction in model accuracy due to inconsistent effects of features on target variables when building machine learning models for model prediction and classification based on distance metrics. That includes models such as Support Vector Machine (SVM), K Nearest Neighbors (KNN), and Multilayer Perceptron (MLP). Therefore, as shown in Table 3, the accuracy of SVM, KNN with k = 5, and MLP models are relatively high, reaching 76.3%, 74.5%, and 75.4%, respectively.

In Table 3 and Table 4, the True Negative Rate for the "Yes" class is higher for all classifiers created based on different machine learning models than the True Positive Rate for the "No" class. This feature is maintained in the self-built machine learning models or the machine learning models in the "weka" software, or using a data set with or without CFS. In addition, the precision of class "Yes "for all classifiers is higher than that of class "No". This indicates that the probability of confidence for a machine learning model based on a "Pima Indian Diabetes" data set to make a "test negative for diabetes" prediction is usually higher than that for a "test positive for diabetes" prediction.

The performance of all classifiers was analyzed by stratified 10-folds cross-validation. Stratified 10-fold cross-validation ensures that the proportion of target variable (Class Variable) instances in each fold is the same as that of the whole data set. Therefore, stratified 10-folds cross-validation is an appropriate performance analysis method for classifiers based on the unbalanced data set of "Pima Indian diabetes". It allows obtaining as much information about classifier performance as possible in limited data. Therefore, it ensures that all classifiers' performance data in Table 3 and Table 4 show their performance as much as possible.

While cross-validation brings data comprehensiveness, it also brings higher computational resource consumption. Compared with the traditional training set validation set partitioning, cross-validation will have an exponential increase in computational effort. This feature allows it to perform on small datasets and small models with relatively limited computational resources. But, when applied to very large datasets and very large deep models, the performance gain from cross-validation may not be so cost effective.

3.3 Effect of the Feature Selection

From the Table 3, we compare the performance of each model algorithm in the weka software using the same cross-validation setup, without feature selection and with CFS. we can see that all the algorithms performed in the weka software have Accuracy is improved or maintained, ZeroR and 1R and 5NN remain unchanged, while the rest of the models are improved. For the accuracy, we can see that all the models except ZeroR and 1R get a "Yes" class accuracy improvement after applying CFS. However, for the "No" class, a significant proportion of the models maintain the

same or slightly decrease the precision. This indicates that the characterization ability of "Yes" class increases significantly after we apply CFS on this dataset.

In our implemented KNN and Naive Bayes models, as Table 4, we found that the accuracy of 1NN and 5NN decreased slightly after using CFS, while there was a moderate increase in Naive Bayes. When subdivided into two classes, in the "Yes" class, the precision of both 1NN and 5NN decreases, but the precision of Naive Bayes increases significantly. In the "No" class, the precision of all three models decreases slightly.

Since the difference between our results and the values in weka is not too large, we consider it to be within the acceptable range, so our results are consistent with the results in weka. This means that the effect of CFS may not be too obvious when applied to KNN, but it is significant when applied to Naive Bayes.

3.4 Classifiers Comparison

In Table 3 and Table 4, the difference in accuracy, TPR, and precision between the My1NN classifier and the 1NN classifier remains below 1%, whether the data set is CFS. The classifier My5NN and 5NN also accord with this feature. Therefore, the performance of the self-built KNN machine learning model is not much different from that of the KNN machine learning model created with "Weka" software when k = 1 and K = 5. In addition, among all classifiers, the performance of the 5NN, NB, MLP, SVM, RF, My5NN, and MyNB classifiers is the best. The accuracy of these classifiers at about 75%. The TPR and precision of these classifiers' "yes" class is about 65%, except the TPR of the "yes" class of SVM is 53%, and the TPR and precision of the "no" class is about 80%. In addition, the performance of the DT classifier is also considerable. Its accuracy can maintain 71.7% when the data set has not been CFS and 73.3% when the data set has been CFS. The accuracy of the 1R classifier can reach 70.8%, but the TPR of the "yes" class is only 47.7%. The performance of the zeroR, 1NN and My1NN classifiers is the worst among all classifiers, with the accuracy of 65.1%, 67.8%, and 68.4%, respectively. Among them, the accuracy of the 1NN improved after the data set has been CFS, reaching 69%.

4 Conclusion

In our exploration of a range of traditional machine learning methods on the Pima dataset, we found that SVM, MLP methods benefit from stronger learning characterization and classification capabilities, as well as stronger model capacity, with relatively high performance. In contrast, the relatively shallow model like Zero R, 1R, and DT models are limited by the complexity of the data and do not show good results. KNN also rises to a good level of model performance with increasing K within a certain range. The performance of Naive Bayes is at a moderate level. In another dimension, the data is downscaled by feature selection, increasing the correlation between the classes and the features, and removing unnecessary attributes, making it easier for the model to learn from the data. Most of our models maintain the current performance or have a corresponding increase through CFS.

For future work, we can go further in the direction of model complexity and data dimensionality. For the models, we can increase the model complexity, improve the model representation and learning ability, explore underfitting and overfitting, and find the direction of optimization. For data, other more modern feature selection methods can be tried, and also can introduce prior knowledge for better feature selection.

5 Reflection

In this study, we obtained the experience of building two corresponding classifiers from starch by using the KNN and NB machine learning models' algorithms. In addition, through the analysis of the performance of different machine learning models, we have a basic understanding of their performance differences. Linking the performance differences of different machine learning models with the algorithms behind them deepens our understanding of the algorithms of those machine learning models.

In addition to the theoretical framework, we also need to pay attention to the details on implementation. For example, when processing the dataset, we need to know that the testset does not contain a label while the trainset does, so we need to be extra careful with the column. Our team's implementation spent a lot of time on finding bugs in testset slicing.

References

[1] Mark A Hall. Correlation-based feature selection of discrete and numeric class machine learning. 2000.