INTRODUCTION TO ARTIFICIAL INTELLIGENCE
COMP3308

ASSIGNMENT 2

BY
LAVA SHRESTHA
SID: 460277116

CONTENTS

1 Introduction

1.1 Introduction to Machine Learning

Machine Learning is considered to be one of the divisions of the Artificial intelligence in which computer programs learn from the examples, domain knowledge and user's feedback and without any human support or interference[1]. There are predominantly 3 kinds of Machine learning that are listed below:

i. Supervised
ii. Unsupervised
iii. Reinforcement

Firstly, in the Supervised learning, we are provided with the labelled data. Then the classifier will summarize that data and predicts the class for new unlabelled data. As supervised learning needs data with an already known label, those data are used to train the algorithms.

Whereas in unsupervised learning, we don't have any labels for given data. The algorithms explore the data and group the data examples into the finite number of clusters in such a way that each cluster are similar to each other in some way.

Reinforcement learning is the process that an agent or system learns independently from trial and error. Every action is performed to maximize the rewards or to get the optimal result[2]. These kinds of learning are mostly suited for control system application.

This report will emphasis on supervised learning. Some of the supervised machine learning classifiers has implemented with its accuracy and a brief explanation and comparison.

1.2 Aim

The primary purpose of this assignment was for us to understand and apply some of the machine-learning algorithm for the given dataset and how it can be useful in predicting diseases like diabetes. Additionally, aim to competently analyze and understand the information that we obtained from applying all those classifiers and want to know when, how and where should the appropriate classifier be used.

As part of the assignment, I used python to code and implement two classifiers (K Nearest Neighbour and Naïve Bayes). Additionally, tested the correctness of those classifiers by implementing 10-fold stratified cross-validation. Our purpose was not only to implement the classifier but also test its accuracy, its importance and how it can be helpful in predicting diabetes in people.

Furthermore, the objective was to proficiently learn WEKA, an open-source Machine Learning (ML) package, codded in Java and was created at the University of Waikato in New Zealand (http://www.cs.waikato.ac.nz/ml/weka/index.html). During this assignment we used WEKA was used for various purposes. It helped to normalize the given data and also helped to select the good subset for that data. Good subsets are those subsets that are extremely associated with the class. Various other classifiers were also applied to that dataset and respective accuracy was calculated. All the results including their comparison are provided within this report.

1.3 Importance

The studies of this classifier are very important for Machine learning. This helps us to predict the class, which can help greatly in the complex decision-making process. As we know the accuracy of each of the classifier, we will know how much we can rely on the information given by them. Comparing the accuracy of the classifier can be used to identify which classifiers are better in particular dataset. Like for this assignment, we will know how all the classifier will help us to determine the chances of having diabetes with some accuracy, provided some details of that patient.

2 Data

2.1 Data Set

That dataset was collected from the patients from the Pima Indian heritage background. They were all females with a minimum age of 21.

The National Institute of Diabetes and Digestive and Kidney Diseases own this dataset. This dataset was provided by, Vincent Sigillito (vgs@aplcen.apl.jhu.edu) Research Center, RMI Group Leader on 9 May 1990. But for the purpose of this assignment, some of the data were modified for consistency. The data that were missed was substituted by its attribute's averages.

The datasets contain a total of 768 instances. For each instances, it contains 8 attributes, which accounts for the information of the patients. They are:

i. Number of times pregnant
ii. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
iii. Diastolic blood pressure (mm Hg)
iv. Triceps skin fold thickness (mm)
v. 2-Hour serum insulin (mu U/ml)
vi. Body mass index (weight in kg/(height in m)^2)
vii. Diabetes pedigree function
viii. Age (years)

Furthermore, for each instance, there were class attributes included in it that were yes or no. If the patients are tested positive for diabetes they were places on the class 'YES' whereas 'NO' represent they weren't. In our dataset we had 500 instances that had class attribute as no and rest, 268 was in class yes.

2.2 CFS Method

CFS (Correlation-based features) is the method in which helped me to recognize the subsets, which are highly related to the class attributes and plays a major part in finding the class. We used Weka to discover that subset. Weka provided us with those subsets that are listed below:

i. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
ii. 2-Hour serum insulin (mu U/ml)
iii. Body mass index
iv. Diabetes pedigree function
v. Age

2.3 Normalisation of Data

The dataset that we were given was not normalised. It is very important to normalise any data before we use any classifier. There were 8 different attributes and those were measured in different scales. If the datasets are not normalised the value of smaller scale will be less significant compared to large. Weka was used to normalise those data. After the normalisation, the value of all the dataset will be between 0 and 1. All these normalised values will have equal importance while determining the class.

3 Results and Discussion

3.1 Accuracy Result

Table 1: Weka Accuracy Results in %

	ZeroR	1R	1NN	5NN	NB	DT	MLP	SVM	RF
No feature Selection	65.104	70.833	67.838	74.479	75.130	71.744	75.390	76.302	74.869
CFS	65.104	70.833	69.010	74.479	76.302	73.307	75.781	76.692	75.911

Table 2: My Implementation Accuracy Results in %

	MY1NN	MY5NN	MYNB
No feature Selection	68.872	75.526	74.605
CFS	68.621	74.607	76.307

The above table shows the accuracy of all the classifier that we implemented. Table 1 results are obtained by Weka and Table 2 results are obtained by my implementation of the classifier. To obtain the accuracy percentage we divide our data into 10 folds and each fold will have around an equal number of the data. Furthermore, the ratio of yes and no are approximately equal for each fold. Each time 9 fold is used to train the classifier and the 1 fold is used for testing. This is done 10 times using 9 different fold as testing and remaining one fold for training. Each time, we find the proportion of the examples, which are accurately categorised. Finally, we average the accuracy of each fold and those results are listed above.

3.2 Weka Classifier Evaluation

ZeroR was the least accurate among all the classifier whereas SVM was highest accurate classifier when no feature was selected. ZeroR picks the class value that has the most observed in the dataset[3]. This classifier is very simple but very useful, as we can use this for baseline performance for all other classifiers. Baseline performance is important for us to know how good any classifiers are. Any classifier that's accuracy is below ZeroR's accuracy can be considered as impracticable.

1R was another Classifier used with the accuracy of 70.833. 1R classifier generates the 1 rule, based on a single attribute and predicts the class based on that[4]. The best rule is selected by calculating the accuracy on all the attributes and one with the best accuracy is selected. Weka selected attributes no 2) to based its rule on. Following criteria were used by Weka in determining the class.

Table 3:Plasma glucose concentration a 2 hours in an oral glucose tolerance test (no feature Selection)

Attribute value	Predicted Class
< 0.454839	No
< 0.4612905	Yes
< 0.519355	No
< 0.5258065	Yes
< 0.5387095	No
< 0.545161	Yes
< 0.5774195	No
< 0.5967739999999999	Yes
< 0.6419355	No
< 0.7	Yes
< 0.7129030000000001	No
>= 0.7129030000000001	Yes

The 1R prediction was better than ZeroR. There was an increase of 5.72% increase. As we can see that 1R selects the best attribute to determine the class. We can conclude that attributes are dominant in deciding the class.

1NN and 5NN are part of K Nearest Neighbour. In Nearest Neighbour, for an unlabelled example, it finds the nearest distance to training data example and returns the class associated with it. We used the Euclidean distance formula to calculate the distance. For 1NN it just looks in 1 Nearest neighbour and returns the class associated with it whereas in 5NN it looks into 5 nearest neighbour and returns the class that has to occur more in those 5. As we can see the accuracy of the classifier has increased vastly from 1NN to 5NN. Both of these classifiers predicted better than ZeroR. 1NN has an accuracy of 67.838% whereas 5NN has an accuracy of 74.479. We can conclude that it's better to check the class with more neighbours. 5NN also performed better than 1R with 3.64%. We should only use these algorithms for the low dimension (up to 6) and cannot be trusted for high dimension[5]. Using of this classifier in high dimension can lead to over-fitting. This will give higher accuracy on training examples but might not work with new data. In other words, Classifier will memorise the training dataset and produce higher accuracy for it but won't work for new examples.

The next classifier used was Naïve Bayes. This classifier assumes that all the attributes in the dataset are independent of each other and evenly significant. It predicts its class based on that assumption. We should only use this when we know that all the attributes are equally important. The accuracy of this classifier was better with 75.130%, in comparison to most of the ZeroR, 1R, and KNN. This may be because of the assumption that Naïve Bayes makes for the dataset.

Decision Tree (DT) classifier accuracy was 71.744%. It only performed better than ZeroR, 1R, and 1NN, which don't look that appealing. The data in this classifier are represented in the tree. Leaf node contains the class and all other nodes contain an attribute's values. When predicting the class we start from the root and go down to the leaf until we get the class value[6]. Weka assigned Glucose concentration to be its root of the decision tree and built its decision tree from there.

Support Vector Machine (SVM) was the other classifier used. The accuracy of this classifier was highest among all other classifiers. Training points' subset is used during this process[7]. This classifier works by identifying a hyper plane that separates new examples[8]. This classifier gets more complex and takes a lot of time when we have a huge amount of data.

Finally, Random Forest (RF) classifier was used. The accuracy was 74.869% which was better than most of the classifier except NB, MLP, and SVM. The way this classifier work is by building numerous decision trees. It then merges those trees to get the most correct prediction of the new examples. Furthermore, adds extra unpredictability by searching top features with the arbitrary subset of features[9]. This may have lead to better prediction as we can see in our table.

3.3 My Classifier Evaluation

The first Classifier that I implemented was K Nearest Neighbour. I tested the accuracy in 1 and 5 neighbours. The accuracy of 1NN was 68.872 whereas 5NN was 75.526. Accuracy improved drastically by 6.654%. This is a big improvement. We can conclude that 5NN is better classifier than 1NN.

Besides that, I implemented the Naïve Bayes classifier. This was less accurate than 5NN. This could be because of the property of Naïve Bayes take into consideration that all the attributes are significant and independent of each other whereas 5NN don't.

Table 4: Results of Statistical test of Comparing Classifier without Feature Selection

Comparing Classifier	Confidence Interval (Z) at 95%
My1NN and MY5NN	= 0.691 ± 2.26 * 1.338
MY1NN and MYNB	= 0.059 ± 2.26 * 1.487
My5NN and MYNB	= 0.032 ± 2.26 * 1.205

Table 5: Results of Statistical test of Comparing Classifier with Feature Selection

Comparing Classifier	Confidence Interval (Z) at 95%
My1NN and MY5NN	= 0.059 ± 2.26 * 1.472
MY1NN and MYNB	= 0.077 ± 2.26 * 1.944
My5NN and MYNB	= 0.035 ± 2.26 * 1.428

Table 4 and Table 5 shows us the results that confidence interval when we compare 2 classifiers. All of the Confidence Interval values in those tables don't contain 0 in their interval, which says that differences are statistically significant as mentioned in our lecture slides[10].

3.4 Comparison of My Classifier with Weka's

My implementation of 1NN and 5NN performed better than that of Weka by 1.034 and 1.047 respectively. Whereas Naïve Bayes performed poorly by 0.525 when no feature was selected. Weka does stratified cross-validation by default. The reason that difference could be the way Weka works. It works by running the algorithm total of 11 times. It works by running 10 times for the cross-validation fold and final one on the whole dataset[11].

When Correlation-based feature was selected Weka performed better for 1NN while its 5NN and NB accuracy was marginally less accurate in comparison to my implementation of those classifiers.

3.5 Effects of Feature Selection

Weka was used for getting the new dataset from original dataset by Correlation-based feature (CFS) method. We used that new dataset for and get the accuracy on all the classifier. Those new datasets only had only 5 attributes plus their respective class.

For Weka, it almost performed better or the same on all the classifier. The highest improvement was in the Decision tree (DT), which improved by 1.563. Furthermore, Naïve Bayes performed better by 1.172%. It could be because of all the attributes on original dataset were not equally important in predicting the class of the dataset.

While using the CFS method for my implementation, Naïve Bayes improved favourably by 1.702. 1NN and 5NN accuracy decreased marginally by 0.251% and 0.919 respectively.

4 Conclusions and Future Work

4.1 Conclusion

There was some difference in the performance between the classifier. Mean and Standard deviation of Weka and my classifier are shown in the table below.

Table 6: Weka's Classifier Performance

	Mean	Standard Deviation
No feature Selection	72.41	3.86
CFS	73.05	3.97

Table 7: My Classifier Performance

	Mean	Standard Deviation
No feature Selection	73.001	3.61
CFS	73.18	4.04

ZeroR performed poorly and should not be only used as a classifier in machine learning. While it gives us a very good understanding of the threshold on how other classifiers should perform. The best classifier was SVM who performed 11.58% better than ZeroR. In conclusion, we should always start with a simple classifier to understand our data set and move to a more complex one. We should also not rely on one classifier and should try more classifier depending on time and memory.

Furthermore, from my findings, we can conclude that the CFS method for selecting the best subset is rewarding. All the attributes that we have in the dataset will not be necessary for the prediction of the class. Besides less number of attributes takes less memory and speed up the process.

Machine learning is a very important aspect. It can help us to make a decision, which can be expensive and delicate. Like in our data set of it can help doctors and medical practitioner to predict the chances of having diabetes with 73.05%(average accuracy of all classifier with CFS of Weka) accuracy. Machine learning can also be used in different other fields like financial services, health care, government, transportation, etc.

4.2 Suggestion for Future Work

Machine learning is very useful for the future. Use of machine learning has helped a lot of business, government, and healthcare so far. As we have seen in our case the accuracy of the best classifier was around 76%, this cannot be used in some of the critical decision-making processes. Study on the existing classifier should be for so that the accuracy can be improved.

My study showed that there was an improvement when Correlation-based feature selection was used for selecting the subset of the original dataset. Further research should be done in the feature selection technique. This can be very rewarding.

During my study, I also found that the Random Forest (RF) showed better results than Decision Tree (DT) and Random Forest worked by building multiple decision trees. Further study can be done in combining more than one classifier and coming up with better ones with higher accuracy.

5 Reflection

This assignment has challenged me and helped me learned a lot. This assignment has made the topic of machine learning more interesting and fascinating to me. I have managed to learn a great deal. Using supervised machine learning classifier on real-world data has given me experience that can be used in the real world and industry.

Furthermore, this assignment has helped me improved my python programming skills especially with using the 2-dimensional list. Additionally, I got more confidence and understanding in using open source software like Weka. Even though Weka didn't have the great graphics interface, it was fast and had a lot of classifiers that we can choose from. Their documentation was great and support was available online effortlessly.

6 Reference

[1] Irena Koprinska, Comp3308 Lecture Slides, Week 5, 2019

[2] Marr, B. (2018). *Artificial Intelligence: What's The Difference Between Deep Learning And Reinforcement Learning?*. [online] Forbes.com. Available at: https://www.forbes.com/sites/bernardmarr/2018/10/22/artificial-intelligence-whats-the-difference-between-deep-learning-and-reinforcement-learning/#6c1e9a9c271e

[3] Brownlee, J. (2016). *How To Estimate A Baseline Performance For Your Machine Learning Models in Weka*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/estimate-baseline-performance-machine-learning-models-weka/

[4] Irena Koprinska, Comp3308 Lecture Slides, Week 5, 2019

[5] Irena Koprinska, Comp3308 Lecture Slides, Week 5, 2019

[6] Irena Koprinska, Comp3308 Lecture Slides, Week 7, 2019

[7] Aylien, N. (2016). *Support Vector Machines: A Simple Explanation*. [online] Kdnuggets.com. Available at: https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html

[8] Patel, S. (2017). *Chapter 2 : SVM (Support Vector Machine) — Theory*. [online] Medium. Available at: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

[9] Donges, N. (2018). *The Random Forest Algorithm*. [online] Towards Data Science. Available at: https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

[10] Irena Koprinska, Comp3308 Lecture Slides, Week 6b, 2019

[11] Irena Koprinska, Comp3308 Lecture Slides, Week 6b, 2019