Classification of Bio-Medical Dataset

CSE 572 - Data Mining

By
Kavinya Rajendran (1209255628)
Thara Sridhar (1209386760)

Table of Contents

1. Introduction

The goal of this project is to classify a dataset with unknown class attributes, using a set of data with known class labels. Since known class labels are used, a supervised algorithm such as classification can be used to perform the task.

1.1. Dataset & Type of Attributes

We have been provided with a bio-medical dataset which contains features extracted from the image of an eye and predicts whether signs of presence of a diseases is present or not. There are 19 attributes to determine the class label.

- All attributes (1,2,..,19) of the data set except class label are considered to be **Numeric**.
- Class labels are considered to be **Nominal**.

1.2. WEKA

Waikato Environment for Knowledge Analysis (Weka) is a machine learning free software. It is a graphical user interface workbench containing collection of tools for data analysis, predictive modeling and visualization. It can also be connected to databases. With WEKA pre-processing, feature selection, training and testing datasets are made easy.

1.3. Classification

Classification is a supervised learning method, where a set of correctly classified class labels are available for the data. This can be used to train a model which can then be used to predict the labels for a future instance. It is the process of identifying the class labels for unknown dataset based on a model developed by dataset for which class labels are available. The algorithm that implements the classification process is called a classifier.

1.3.1 Conjunctive Rule

A rule consists of antecedents "AND"-ed together and the class label for classification. The consequent is distribution of available classes or the mean of the numeric value in the dataset. This class in weka uses single conjunctive rule learner that can predict numeric and nominal class labels. If a test instance is not covered by the rule, then it predicts using the default class distribution of data not covered by the rule in the training data. This learner selects an antecedent by computing the Information Gain of each antecedent and prunes the generated rule using Reduced Error Pruning (REP) or simple pre-pruning based on the number of antecedents. For classification, the information of one antecedent is the weighted average of the entropies of both the data covered and not covered by the rule. In pruning, weighted average of the accuracy rates on the pruning data is used for classification.

1.3.2 Multiclass Classifier

Multiclass or multinominal classification is the method of classifying instances into 2 or more classes. A binary class can be turned into a multinominal classifier. This is a meta-classifier to

handle multi-class datasets with 2-class classifier capable of applying error correcting output codes for increased accuracy.

1.3.3 Bagging

An ensemble method used to reduce variance and can do classification depending on the base learner. It creates separate samples of the training dataset and creates a classifier for each sample. The results of these multiple classifiers are then combined based on average or a voting method. The model is used when accuracy is critical, we don't need to understand the model, training cost is not critical or when there is a need to optimize a special or combination of error metrics.

1.3.4 Random tree

It constructs a tree that considers k randomly chosen attributes at each node. It does not perform pruning. Weka has an option of allowing estimation of class probabilities based on a hold out set.

1.3.5 Decision Table

Decision tables like decision trees or neural networks are classification models for prediction. It consists of a hierarchical table in which each entry in a higher level table gets broken down by values of a pair of additional attributes to form another table. It is a class in weka to build and use simple decision table majority classifier.

1.3.6. Naive Bayes

Naive Bayes is a simple probabilistic classifier based on the Bayes' theorem with strong independence assumptions between the features. It is highly scalable requiring number of parameters linear in number of variables in a learning problem. In weka it is a class for Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data, for this reason this is not a updateable classifier.

1.3.7. Random Forest

Random forest is a general technique of random decision forests that is an ensemble learning method for classification, that operate by constructing a multitude of decision trees at training time and outputs the class that is a mode of the classes for a classification. The selection of the random subset of features is a random subspace method. It constructs a forest of random trees.

1.3.8. Logistic Regression

It is a model where the dependent variables are categorical. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function which is the cumulative logistic distribution. In weka it is a class of building and using a multinomial logistics regression model with a ridge estimator. The original algorithm is modified to deal with instance weights

2. Submission 1 - Random Forest Classification (80% split on training data) without pre-processing

Our overall approach was to test if increase in accuracy for training dataset resulted in increasing accuracy in testing dataset. On splitting the training dataset into training and testing, we would be getting an accuracy. We used multiple submissions to test if the increase in accuracy leads to increase in the actual test data results.

2.1. Pre-processing

- Since the data was encrypted, we didn't perform any filters on the dataset provided.
- We didn't perform any data cleaning like removing instances or removing irrelevant attributes.

2.2. Classification Algorithm

1. For each of testing and training data, create its corresponding arff formatted files [1].
2. Build a model by defining its algorithm and percentage split on the training data and observe its accuracy.
3. Save the model built in previous step.
4. Load the saved model and apply the testing dataset to obtain the predicted class labels.

2.3. Experiments

We developed a few classifications models using the below mentioned algorithms with 66% (default split on WEKA) and 80% (5-fold cross validation) split on the given training dataset. The observations are provided in table 2.1 and graphically representation is shown in figure 2.1.

Classification Algorithm	Accuracy	
	With 66% Split	With 80% Split
Logistic	75.72%	76.09%
Random Forest	70.29%	72.28%
Naive Bayes	60.06%	59.78%
Decision Table	62.62%	63.04%
Random Tree	65.81%	63.59%
Bagging	70.61%	70.11%
Multiclass Classifier	75.72%	76.09%
Conjunctive Rule	61.98%	59.24%

Table 2.1 : Accuracy of Classification Algorithms, without preprocessing

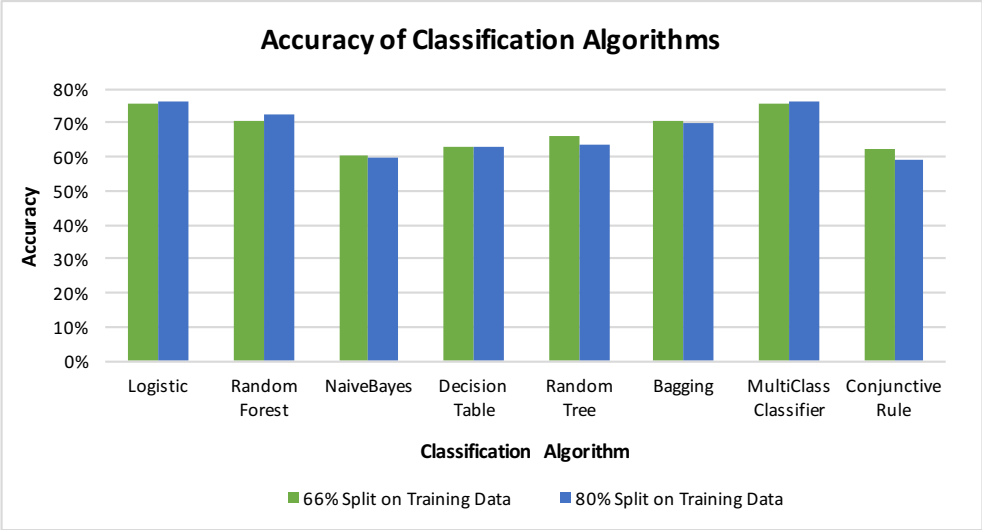

Figure 2.1 : Accuracy of Classification Algorithms, without preprocessing.

2.4. Observations

- Logistic and Multiclass Classifier use the same approach and yields the best accuracy compared to other approaches.
- Random forest provides the second highest highest accuracy.
- In many cases 5-fold validation results were better comparatively.

2.5. Training Label

We used the random forest algorithm with 5-fold cross validation to obtain the labels for the training data, using WEKA. The accuracy in this submission was used to compare against accuracy in submission two and conclude if increase in accuracy of training data actually increased the accuracy in testing data.

3. Submission 2 – Logistic/Multiclass Classifier (80% split on training data) with pre-processing

3.1. Feature Selection

WEKA provides many ranker algorithms that ranks the attributes based on its impact on the label. **We have named all the attributes as a sequence of number starting from 1 (1,2,3,..,18)** and worked on various ranker algorithm to obtain the ranking of the attributes**.**

Ranker Algorithm	Rank of Attributes
ChiSquaredAttributeEval	5,13,16,12,2,9,8,3,17,14,10,15,11,6,1,19,4,7,18
FilteredAttributeEval	5,13,9,16,12,2,8,3,17,14,10,11,15,6,1,19,4,7,18
InfoGainAttributeEval	5,13,9,16,12,2,8,3,17,14,10,11,15,6,1,19,4,7,18
SymmetricalUncertAttributeEval	5,9,13,2,12,16,8,3,14,11,10,17,15,6,1,19,7,4,18

Table 3.1 : Ranking results of various ranker algorithms using WEKA 3.6

Based on our observation, we decided to remove the least ranked attributes from our model as a part of pre-processing. But, we were uncertain of how many such attributes are to be removed. This led us to doing a number of experiments by removing different number of attributes and testing their accuracy.

3.2. Pre-Processing

- We used WEKA ranker algorithm to preprocess our data by considering only high ranked attributes for our model. The low ranked attributes are thus removed from the training data as a part of pre-processing.
- We didn't remove any instances.

3.3. Classification Algorithm

1. For each of testing and training data, create its corresponding arff formatted files [1].
2. Use WEKA ranker algorithms to rank the attributes in training dataset.
3. Remove few least ranked attributes as a part of pre-processing.
4. Build a model using logistic (multiclass classifier) with percentage split on the training data and observe its accuracy.
5. Save the model built in previous step.
6. Load the saved model and apply the testing dataset to obtain the predicted class labels.

3.4. Experiments

As logistic (multiclass classifier) gave us the best accuracy in our previous experiment, we decided on applying the algorithm on our next set of experiments. Hence, for this submission we compared the accuracy of logistic (multiclass classifier) on providing different feature set.

Attribute that are removed during pre-processing	**Accuracy for Percentage Split on Training Data**				
	60%	**66%**	**70%**	**75%**	**80%**
None	76.63%	75.72%	76.09%	76.96%	76.09%
18	76.90%	75.72%	76.09%	76.96%	76.09%
18, 7	76.90%	75.72%	76.09%	76.52%	75.54%
18, 7, 4	**76.63%**	**76.68%**	**76.45%**	**76.52%**	**76.09%**
18, 7, 4, 19	74.46%	75.72%	76.09%	75.65%	76.63%
18, 7, 4, 19, 1	76.09%	76.36%	76.81%	76.52%	77.17%
18, 7, 4, 19, 1, 6	**73.37%**	**74.44%**	**74.64%**	**76.09%**	**75%**
18, 7, 4, 19, 1, 6, 15	73.64%	74.76%	75%	76.52%	75%

Table 2.2 : Accuracy of Logistic model with different preprocessing and split on training data

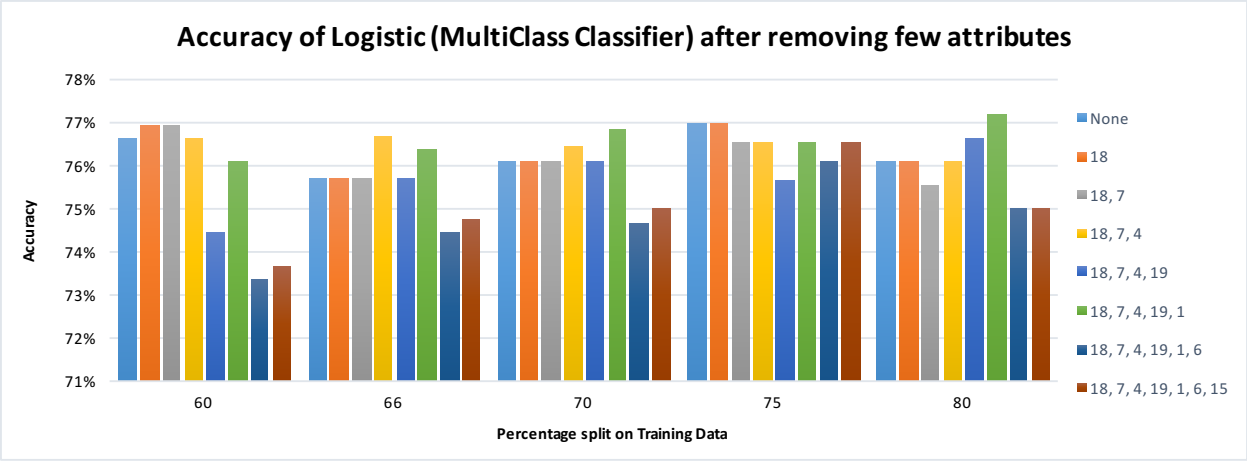

Figure 2.1 : Accuracy of Logistic model with different preprocessing and split on training data

3.5. Observations

- Building logistic model on any pre-processed training data provides accuracy that is higher than random forest (previous submission)
- On removing too many attributes, the percentage of accuracy drops.
- The training dataset on removing the attributes 18, 7 and 4 yields the best accuracy, which is consistent over all split of training data.

3.6. Training Label

We built a model by first pre-processing it on removing the 6 least ranked attributes from the training data and applying logistic classification using 5-fold cross validation. We used this model to obtain the training label.

4. Final Submission - Logistic/Multiclass Classifier (74% split on training data) with pre-processing

4.1. Pre-Processing

From the observations of our previous experiment, we decided make further experiments with minimum preprocessing that removes the attributes 18, 7 and 4. This is because, with that processed dataset, the accuracy remained almost consistent with every percentage split on the training data.

- Pre-processing involved removing attributes 18, 7 and 4 along with some additional attribute. We decided on removing 18, 7 and 4 because its accuracy is more and consistent throughout the different split.
- We didn't remove any instances.

4.2. Classification Algorithm

1. For each of testing and training data, create its corresponding arff formatted files [1].
2. Remove attributes 18, 7 and 4 from the training dataset as a part of pre-processing.
3. Remove additional attribute to test accuracy.
4. Build a model using logistic (multiclass classifier) with percentage split on the training data and observe its accuracy.
5. Save the model built in previous step.
6. Load the saved model and apply the testing dataset to obtain the predicted class labels.

4.3. Experiment

To take it a step further, we removed random attributes (on top of removing 18, 7 and 4) and applied logistic (multiclass classifier) for various percentage split on training data and observed its accuracy.

Attribute that are removed during pre-processing	Accuracy for Percentage Split on Training Data			
	66%	70%	74%	80%
18, 7, 4	76.68%	76.45%	77.41%	76.09%
18, 7, 4, 3	76.04%	75.36%	75.31%	76.09%
18, 7, 4, 5	76.36%	77.54%	78.24%	77.72%
18, 7, 4, 6	76.04%	76.09%	76.57%	76.63%
18, 7, 4, 10	76.63%	77.00%	78.66%	77.17%
18, 7, 4, 15	76.63%	77.64%	78.66%	77.72%
18, 7, 4, 17	77.00%	77.54%	78.66%	77.72%

Table 4.1 : Accuracy of Logistic model with different preprocessing and split on training data

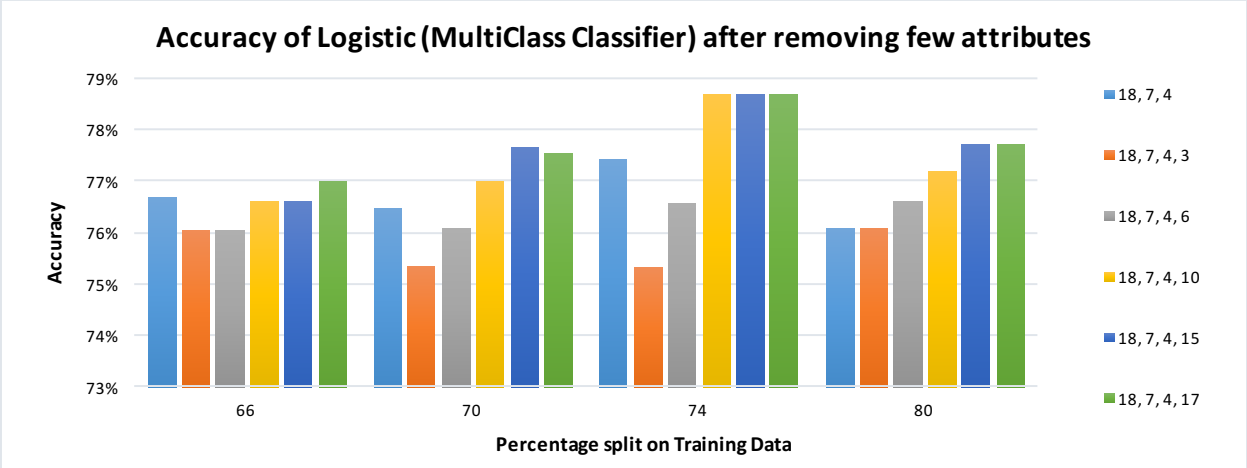

Figure 4.1 : Accuracy of Logistic model with different preprocessing and split on training data

4.4. Observations

- Accuracy remains within the range of 76% and 79%, indicating that it is still better than random forest model.
- The best results were obtained on removing attributes 10, 15 or 17 along with 18, 7 and 4.

4.5. Training Label

With submission 1 and 2, we understood that the accuracy difference with the training data follows a similar pattern with the accuracy in the test data. Hence, we chose the feature set with model and split percentage for which we got the maximum accuracy.

We thus pre-processed the training data set and removed the attributes 18, 7, 4 and 15 and passed it to our classification algorithm. Our classification model used the logistic (multiclass classifier) with 74% split on the training data to build our model. On applying the model to the testing data set, we obtained the test labels.

5. Conclusion

For the given bio-medical encrypted dataset, we performed various classification algorithms and tested its accuracy based on pre-processing the data. We also made use of ranker algorithms to make decisions on which attribute is to be removed as a part of pre-processing. As a result, we made lot of comparisons between and within a particular classification algorithm and provided the best solution as a part of submission.

6. References

[1] http://ikuz.eu/csv2arff/
[2] https://en.wikipedia.org/wiki
[3] http://wiki.pentaho.com/display/DATAMINING
[4] http://www.cs.columbia.edu/~wfan/software.htm
[5] http://dl.acm.org/citation.cfm?id=721218
[6] http://weka.sourceforge.net/doc.dev/weka/classifiers
[7]http://www.dbs.ifi.lmu.de/~zimek/diplomathesis/implementations/EHNDs/doc/weka/classifiers/rules/DecisionTable.html
[8] http://weka.8497.n7.nabble.com/Random-Tree-td17230.html
[9] http://stackoverflow.com/questions/30654126/multi-class-classification-in-weka
[10] http://www.quansun.com/bagging_es/