Homework 4: Introduction to Data Mining (CSE 5243)

Ashwini Joshi
10/23/2017

Introduction:

The report aims to compare and analyse different types of classifiers on Wine Dataset. Multiple classification algorithms are run with different set of combinations of the data attributes and other parameters in order to find out which suits the best for the given dataset. Exploratory Data Analysis and Data Transformation is done in the first two sections. Actual Model Building and Evaluation is included in the last two sections.

Section 1: Preliminary Data Analysis

This section analyses the given dataset in order to find interesting patterns and relationships between different fields.It is done using 'summary' function in RStudio. Since all the variables in the dataset are continuous except for quality and class, Table 1.1 summarizes mean, standard deviation, median, maximum value, minimum value and correlation with class attribute for each.

Attribute	Mean	Standard Deviation	Median	Minimum	Maximum	Correlation with class attribute
fx_acidity	8.32	1.741	7.90	4.6	15.90	0.09
vol_acidity	0.528	0.179	0.52	0.12	1.58	0.321
citric_acid	0.271	0.195	0.26	0	1	-0.159
resid_sugar	2.539	1.41	2.20	0.9	15.50	0.002
chlorides	0.087	0.047	0.08	0.012	0.611	0.109
free_sulf_d	15.875	10.46	14.00	1	72	0.061
tot_sulf_d	46.468	32.895	38.00	6	289	0.231
density	0.997	0.002	0.99	0.99	1.004	0.159
pH	3.311	0.154	3.31	2.74	4.01	0.003
sulph	0.658	0.170	0.62	0.33	2	-0.218
alcohol	10.423	1.066	10.20	8.4	14.9	-0.434

Table 1.1: Summary Statistics for Wine Dataset

Table 1.1 shows that alcohol has the strongest relation with the class variable. Also, sulph and citric_acid have negative correlation. Table 1.2 shows some additional correlations between different attributes with significantly high value. It is useful for Feature Subset Selection since correlated attributes can be removed in that stage.

Attributes	Correlation
(density, alcohol)	-0.496
(free_sulf_d, tot_sulf_d)	0.667
(fx_acid, pH)	-0.682

Table 1.2: Correlations between different features

Figure 1.1 shows histograms for some attributes. These graphs are plotted by importing the dataset in RStudio. It is observed that the distribution is positively skewed for all of them, hence most of the outliers lie on the right side (large side). For the field alcohol some outliers also lie on left side.

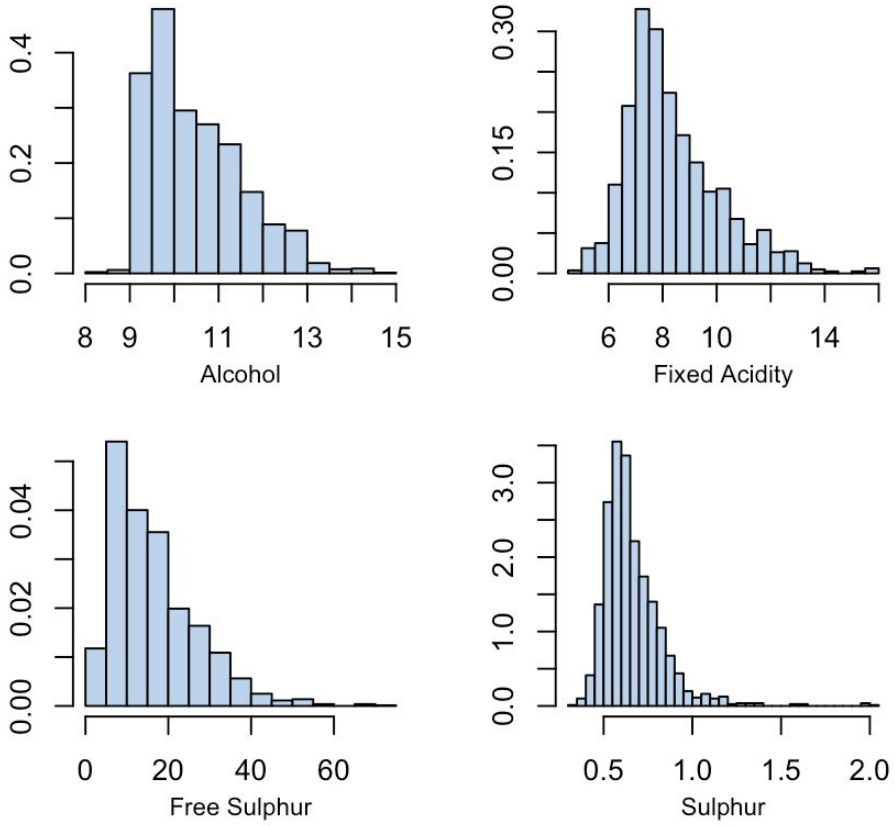

Figure 1.1: Histogram for attributes

Attribute quality is discretized into Low and High class. If its value is less than or equal to 5 then it is classified as Low, otherwise High. Figure 1.2 shows distribution for both quality and class variable.

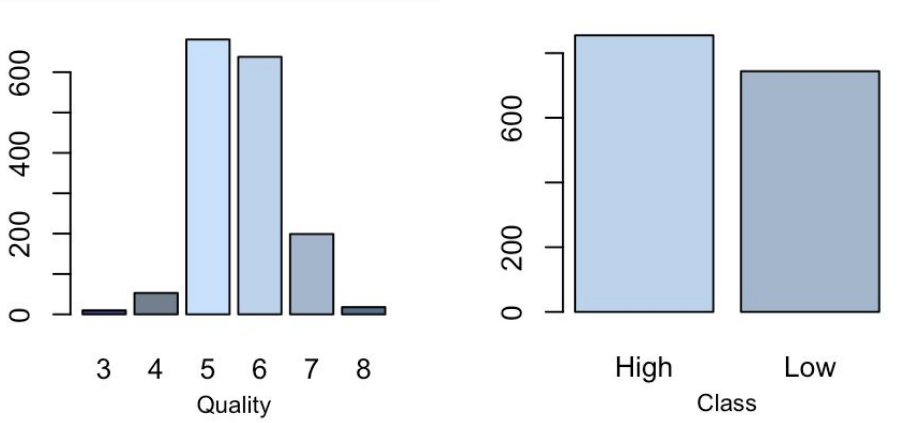

Figure 1.2: Distribution for Quality and Class attribute

Section 2: Data Transformation

This section discusses different types of data transformation techniques used on the dataset in order to make it suitable for various classifier models.

Outliers: As discussed in the previous section, some attributes like alcohol, fixed acidity, sulphur content have outliers. These outliers are removed in order to test whether it improves the performance of the classifiers or not. It is observed that, performance is significantly improved. For example, in decision tree classifier, accuracy before and after removing is 71.70% and 74.60% respectively. Hence, outliers and extreme values are removed using filters in Weka. After removal, size of the training dataset is reduced to 1435 from 1599 instances.

Feature Subset Selection: Correlations in Table 1.2 show that some of the variables are correlated to each other which results in lower accuracy. In order to check, Feature Subset Selection is done using a package in Weka called 'select attributes'. For each classifier, it gives the best set of features by removing unnecessary attributes which do not contribute toward predicting the value of the class variable. However, it is seen that there is not much improvement in accuracy even after removing some attributes except for Naive Bayes classifier. Detailed description is mentioned for each classifier in the Model Development (Section 3).

Handling Types of Attributes: Attribute 'ID' is removed since it is not helpful in determining the value of the class. Also, quality is categorized into low and high, so it is also not considered during building the classification model. Classifier models are built and run using inbuilt packages hence there is no need to normalize the attributes because these issues can be handled by controlling parameters while training each model. Also, binarization is not necessary since all the fields are continuous variables.

Section 3: Model Development

This sections talks about approach taken for training each model. Training dataset is divided into two parts. Two third part is used for training and remaining one third is used for testing the model built. Partition is done in this way because it keeps enough records for both training the model as well testing the output.

A. **Decision Tree:** This is built using a package called as 'J48' in Weka. Since some attributes are correlated, classifier is trained using different set of attributes. Selection is done using 'Greedy Stepwise' package in 'Select Attributes' in Weka which finds the best attribute at given stage. Attributes - pH, density and free_sulf_d are removed using the results obtained from the package. But, accuracy is reduced from 75.20% to 73.97%. Hence, all the attributes are considered for this classifier. Output is compared by setting 'binary splits' as true or false which decides whether to split tree with multiple splits or two splits. Also, parameter 'unpruned' is varied to see if pruning improves the output. It is seen that using binary splits and pruning set to true, maximum value of accuracy is obtained. Out of 488, it classifies 367 instances correctly giving an accuracy of 75.20%.

B. **Rule Based Classifier:** Rule based classifier is designed using package 'JRip' in Weka with all the attributes taken into consideration. It uses RIPPER algorithm where initial rule set is built first by growing it at every stage and then pruned with minimizing error at each stage. 'Folds' parameter with default value 3 decides the total number of stages in the algorithm where one fold is reserved for pruning and rest are used for used for growing rule set. It is kept as default because it is sufficient for pruning the rule set for maximum error reduction. Parameters 'use pruning' and 'check error rate' are set to true which allows to prune the tree by checking error rate at each fold. It gives an accuracy of 75.61% with 369 instances classified correctly and 119 instances misclassified out of 488.

C. **Naive Bayes Classifier:** It is implemented using 'Naive Bayes' package in Weka. This algorithm is straightforward since it directly computes the posterior probabilities for the instances using prior probabilities in the training dataset. 'NumDecimalplaces' is set to 2 which determines the precision of the calculated probabilities. There is a parameter called 'unsupervised Discretization' which is set to true to convert numeric attributes to nominal using supervised discretization since it is improving the results. Naive Bayes classifier is very sensitive to correlation between attributes because it assumes that the variables are independent. Hence, feature subset selection is used for this classifier using 'select attributes' package in Weka. Algorithm is run only using these attributes - vol_acidity, fee_sulf_d, tot_sulf_d, sulph and alcohol. Accuracy is improved from 74.18% to 77.25% after discretizing attributes and feature subset selection with 377 instances classified correctly out of 488.

D. **Artificial Neural Network:** This model is built using a package in Weka called 'MultiLayerPerceptron' considering all attributes. It builds a multilayer perceptron using a default learning rate of 0.3. It is changed to 0.5 to keep a balance between previous weights and current effect of iteration. However, 0.3 gives better output which implies that the effect of previous weights should be slightly more as compared to current iteration. Parameter 'momentum' decides the momentum applied at each level which is set to 0.2. Parameter 'Hidden Layers' decides the number of levels in the perceptron. It can take one of the four values - number of attributes, number of classes, number of attributes + number of classes, (number of attributes + number of classes) / 2. The fourth value gives optimized output since it keeps a balance between too less and too many number of levels. Accuracy is not improved after feature subset selection hence all the attributes are considered. This model classifies 375 instances correctly out of 488 with an accuracy of 76.84%.

E. **Support Vector Machine:** Classification model is trained using package 'SMO' in Weka with all the attributes because there is no improvement in accuracy with selecting only a set of features. Filter 'normalize traindata' is set to true to normalize the attributes. Tolerance factor is set to 0.001 which is the default value. It determines tolerance in error till a particular distance from the hyperplane created during building SVM. The model is built using Logistic Regression as calibrator since it gives the maximum accuracy. The output can be changed by changing the parameter 'build calibration models' which set to true in this case since it gives posterior probabilities which are needed to build the ROC curve. It correctly classifies 363 records out of 488 with an accuracy of 74.38%.

F. **Ensemble Classifier:** Two models were developed in this method - bagging and Random Forest. But, bagging gives an accuracy of 75.61% which is not very different from those obtained with single classifier models mentioned above. Hence, Random Forest is used in ensemble classifier. It is built using package 'RandomForest' in Weka. Parameter 'bagsize percent' gives percentage of size of the records as a percentage of training dataset at each iterations. Number of iterations can be manipulated using 'num Iterations' with default value 100 which is set to 400 since it provides a balance between time required to build the model and accuracy of the output. It increases accuracy from 79% to 81.56%. If 'print classifiers' is set to true, it divides the output in multiple sections with single classifier so that output of each can be analysed independently. Field 'maxDepth' can be used to adjust the maximum depth of the tree. It is set to 0 which means unlimited so that classifier can figure out the appropriate value depending upon the data passed to it. All variables are considered because no improvement in performance is observed after selecting a subset of features. It classifies 398 rows correctly with 90 misclassified instances giving an accuracy of 81.56%.

Section 4: Model Evaluation

This section evaluates the results of the models developed in the previous section. Classifier outputs from Weka are used to calculate statistics and plotting ROC curves in Python. Scikit package - 'sklearn.metrics' is used for building confusion matrix and ROC curve. Actual ROC curves are plotted using 'matplotlib' in Python. Performance is measured using parameters - accuracy, F1-score, confusion matrix and ROC curve.

Table 3.1 shows accuracy, F1-score and time taken to build the model using train split and time taken to test it on the test split for each classifier. Time taken to build the model is 0 seconds for Bayes classifier since it only calculates posterior probabilities using training dataset for prior probabilities rather than actually building a model.

Classifier	Accuracy	F1-score	Time Taken to Build the Model (In seconds)	Time Taken to Test the Model (In Seconds)
Decision Tree	75.20%	0.7835	0.02	0.04
Rule Based Classifier	75.82%	0.7808	0.15	0.02
Naive Bayes Classifier	77.25%	0.7970	0	0.02
Artificial Neural Network	76.84%	0.7903	1.13	0.02
Support Vector Machine	74.38%	0.7714	0.04	0.02
Ensemble (RandomForest)	81.55%	0.8398	1.19	0.24

Table 4.1: Accuracy, F1-score and Time Taken to Train and Test the model for all classifiers

Table 4.2 to 4.4 show confusion matrix with 'High' as positive class for each of the classifiers mentioned above.

	Actual Class (				Actual Class		
Predicted Class	Decision Tree	Class + (High)	Class - (Low)	Predicted Class	Rule Based Classifier	Class + (High)	Class - (Low)
	Class + (High)	TP (+/+) 219	FP (-/+) 58		Class + (High)	TP(+/+) 212	FP(-/+) 65
	Class - (Low)	FN (+/-) 63	TN (-/-) 148		Class - (Low)	FN (+/-) 54	TN(-/-) 157

Table 4.2: Confusion Matrix for Decision Tree and Rule Based Classifier

	Actual Class				Actual Class		
Predicted Class	Naive Bayes Classifier	Class + (High)	Class - (Low)	Predicted Class	MultiLayer Perceptron	Class + (High)	Class - (Low)
	Class + (High)	TP (+/+) 218	FP (-/+) 59		Class + (High)	TP(+/+) 213	FP(-/+) 64
	Class - (Low)	FN (+/-) 52	TN (-/-) 159		Class - (Low)	FN (+/-) 49	TN(-/-) 162

Table 4.3: Confusion Matrix for Naive Bayes and MultiLayer Perceptron

	Actual Class				Actual Class		
Predicted Class	Support Vector Machine	Class + (High)	Class - (Low)	Predicted Class	Random Forest	Class + (High)	Class - (Low)
	Class + (High)	TP (+/+) 211	FP (-/+) 66		Class + (High)	TP(+/+) 236	FP(-/+) 41
	Class - (Low)	FN (+/-) 59	TN (-/-) 152		Class - (Low)	FN (+/-) 49	TN(-/-) 162

Table 4.4: Confusion Matrix for Support Vector Machine and Random Forest

Figure 4.1 to 4.3 show Receiver Operating Characteristics Curve for all the classifiers implemented with 'High' as positive class:

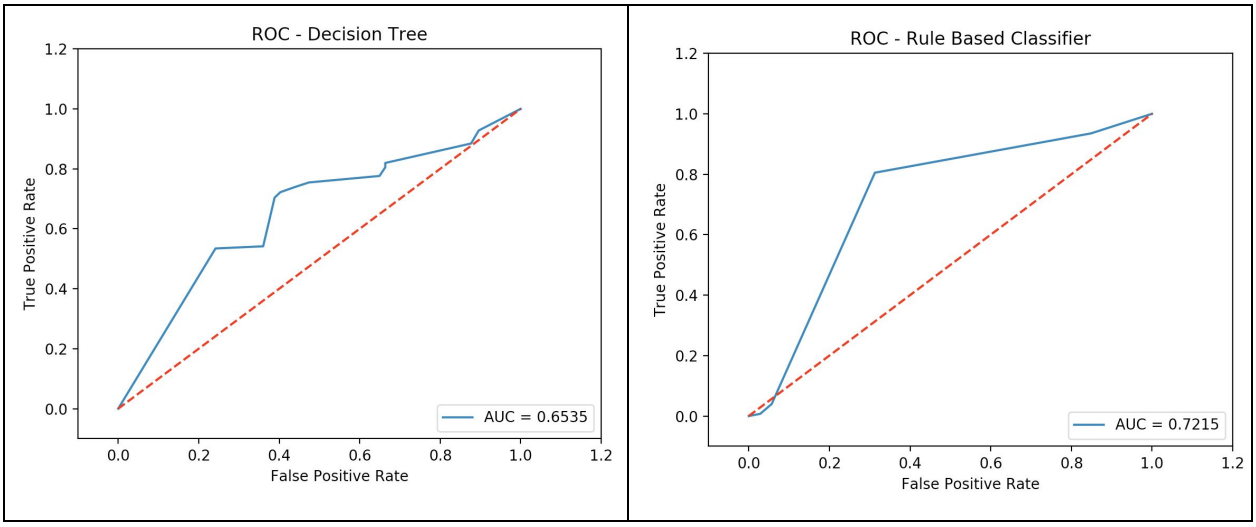

Figure 4.1: ROC Curve for Decision Tree and Rule Based Classifier

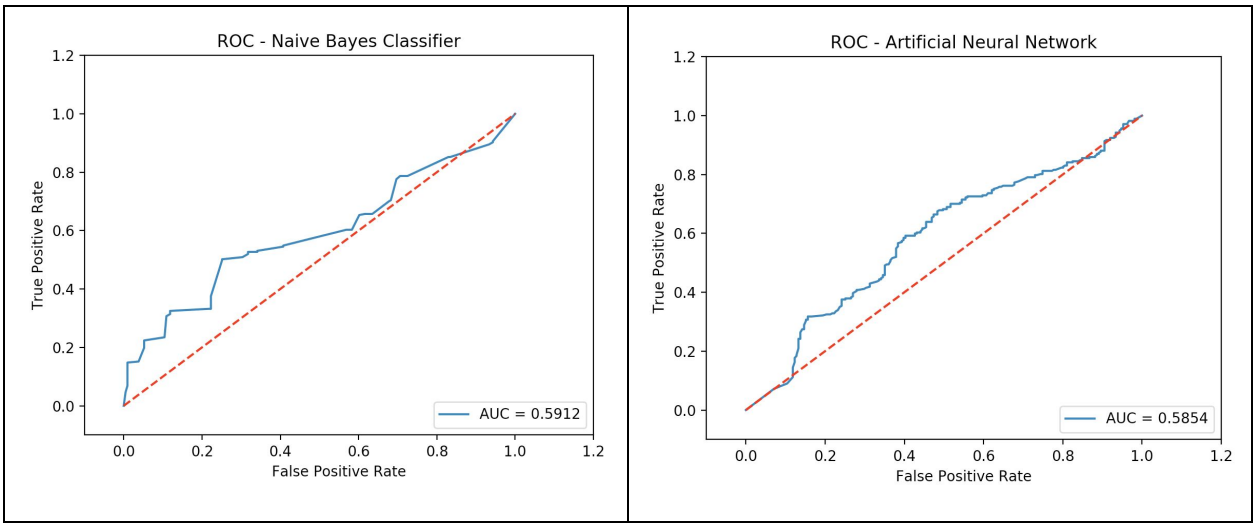

Figure 4.2: ROC Curve for Naive Bayes and MultiLayer Perceptron

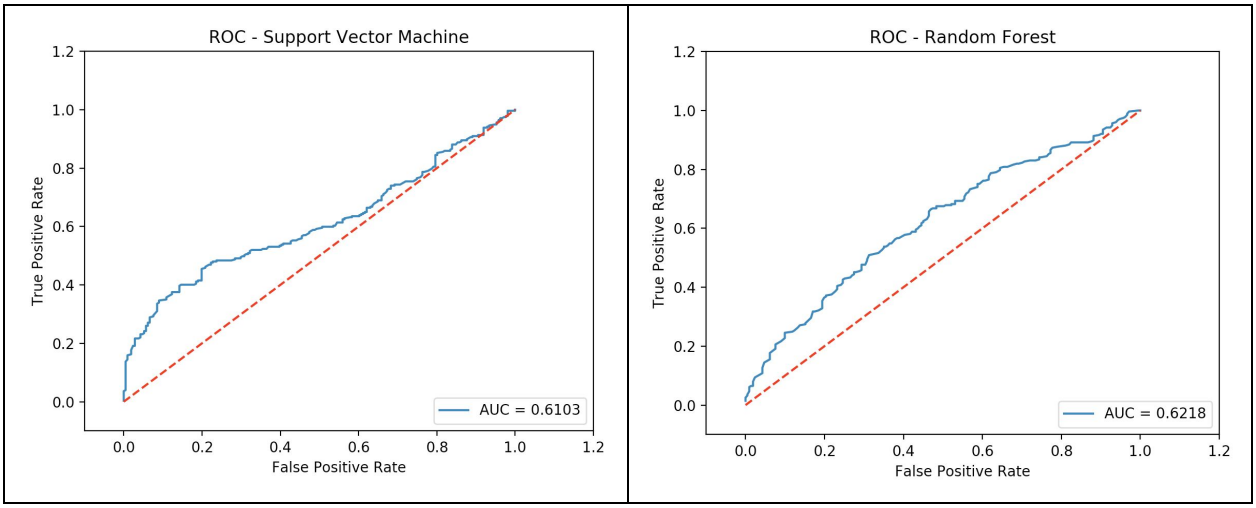

Figure 4.3: ROC Curve for Support Vector Machine and Random Forest

Observing above statistics and figures, performance of Artificial Neural Network and Naive Bayes classifier is comparable because all of the following values - Accuracy, F-measure and Area Under Curve (AUC) have values in the same range. Same is the case with Decision Tree and Rule Based Classifier. Support Vector Machine is giving the worst output in terms of Accuracy and F-measure but a slightly better AUC value. Random Forest has the maximum value of Accuracy and F-measure. But, AUC is maximum for Rule Based classifier. AUC has a large value for decision tree also, as compared to Random Forest. But considering overall performance, Random Forest is giving better results because its accuracy and F-measure values are significantly high as compared to others. Also, AUC value is not very low with respect to Rule Based or Decision Tree. Hence, Random Forest is the best method to classify the given dataset. However, it should be noted that time required to train and test the model using Random Forest is remarkably high as compared to other models since it goes through multiple stages while building classification model. Therefore, though Random Forest gives the best output, there is trade-off between speed and accuracy.