Neelkumar S Bhuva

Data Mining - Wine Report

1. INTRODUCTION :

In this report we explore the wine dataset. This report involves preliminary data analysis which includes observed trends with the data, summary statistics, some interesting graphs and correlation of attributes. Then we discuss transformations to the dataset before feeding it to the model for classification. Different models are used for classification and each of these models are evaluated.

2. PRELIMINARY DATA ANALYSIS :

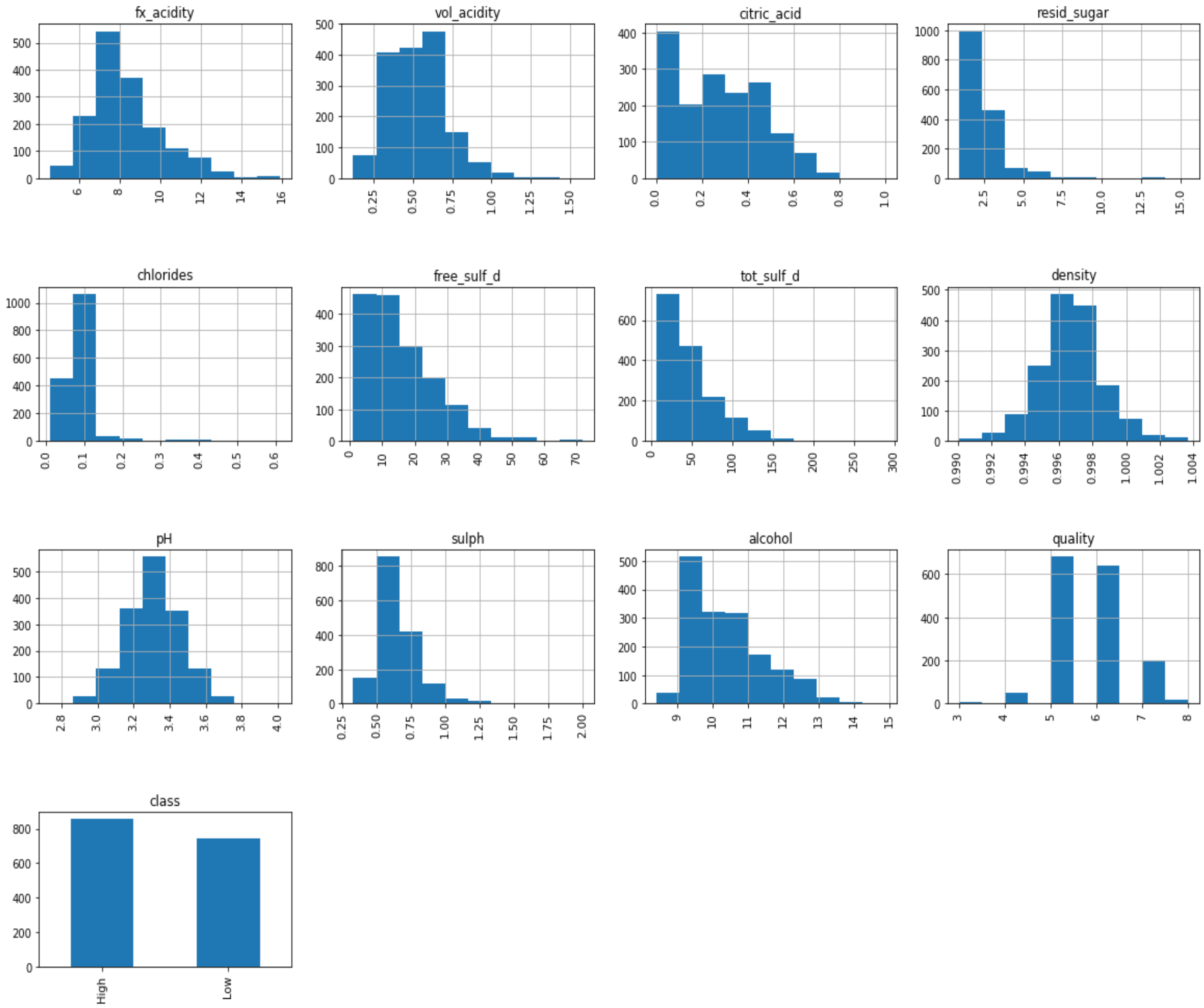

Fig 2.1 : Histograms showing frequency of all attributes

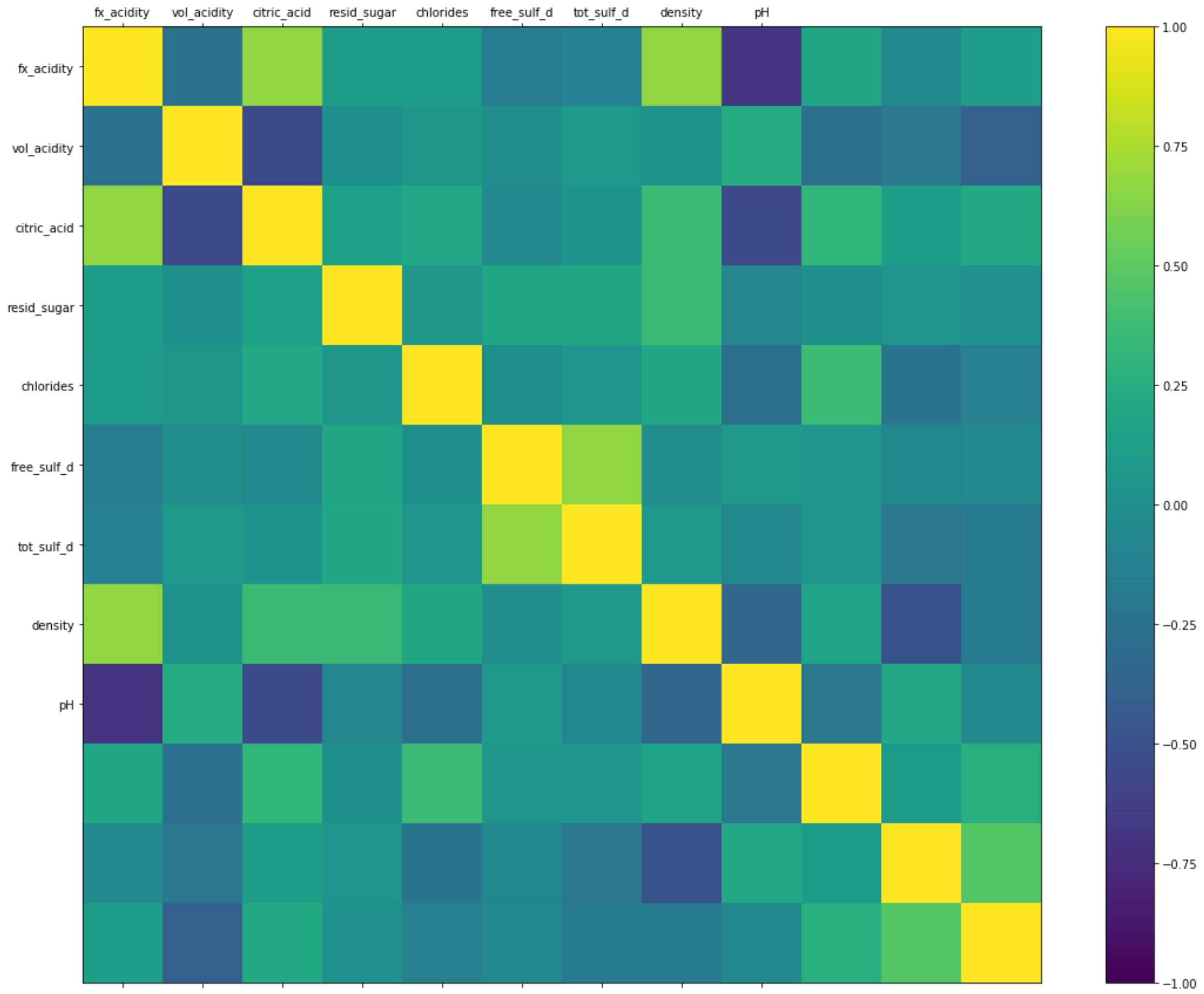

Fig 2.2 Correlation Matrix

Attribute ID is omitted as it has no meaning and does not influence the class to which the instance belongs to. From Fig 2.2 both positive and negative correlations exists between attributes. (Citric_acid,fx_acidity) and (density,fx_acidity) have higher correlation.

Following observations about the attributes are made from Fig 2.1 :

a) Fixed Acidity (fx_acidity) : Most of the instances have Fixed Acidity value between 6.5 and 8. The frequency of instances increases in the interval 0 - 6.5 and decreases from 8 to 16. A bell shaped curve (normal distribution) is observed. The mean value for fx_acidity is 8.319

b) Volatile Acidity (vol_acidity) : Most of the instances have Volatile Acidity value between 0.25 and 0.75. Mean is 0.527

c) Citric Acid (citric_acid): 132 instances have Citric Acid value as 0.0 Mean is 0.27 and median is 0.26

d) Residual Sugar (resid_sugar) : Most of the instances have values between 1.8 and 2.5. Mean is 2.53 and median is 2.2. Number of instances decreases (not strictly) with increasing resid_sugar value. Lower values have higher frequency than higher values.

e) Chlorides : Majority of the instances have values in the interval 0.0 - 0.1. Mean is 0.087 and median is 0.079. There are very few instances with values between 0.1 and 0.6

f) Free Sulphur Dioxide (free_sul_d) : Majority of the instances have values in the interval 3 - 20. Mean is 15.87 and median is 14. Number of instances decreases (not strictly) with increasing free_sul_d value. Lower values have higher frequency than higher values. Minimum is 1, maximum is 72 and variance is 109 which is high.

g) Total Sulphur Dioxide (totl_sulf_d): Mean is 46.46, median is 38. There are very few instances with value greater than 120. Number of instances decreases (not strictly) with increasing free_sul_d value.

h) Density : Mean and median are 0.996, has low variance of 3.55980179263e-06. This attribute seems to follow normal distribution.

i) pH : Mean and median are 3.311, has low variance of 0.0238202742411. This attribute seems to follow normal distribution like Density.

j) Sulphates (sulph) : Mean is 0.65, median is 0.62, variance is 0.028714. This attribute tends to follow normal distribution.

k) Alcohol : Mean is 10.42, median is 10.19, variance is 1.13. Interval 9-10 has high frequency.

From Fig 3.1, none of the attributes have highly skewed distribution.

3. DATA TRANSFORMATIONS :

A. Missing Values : There are no missing values in the dataset. So, there is no need to handle missing values.

B. Outliers : From Fig 3.1 there are quite a few outliers, but we do not know if that data is invalid and might contain critical information. Outliers will be included in training.

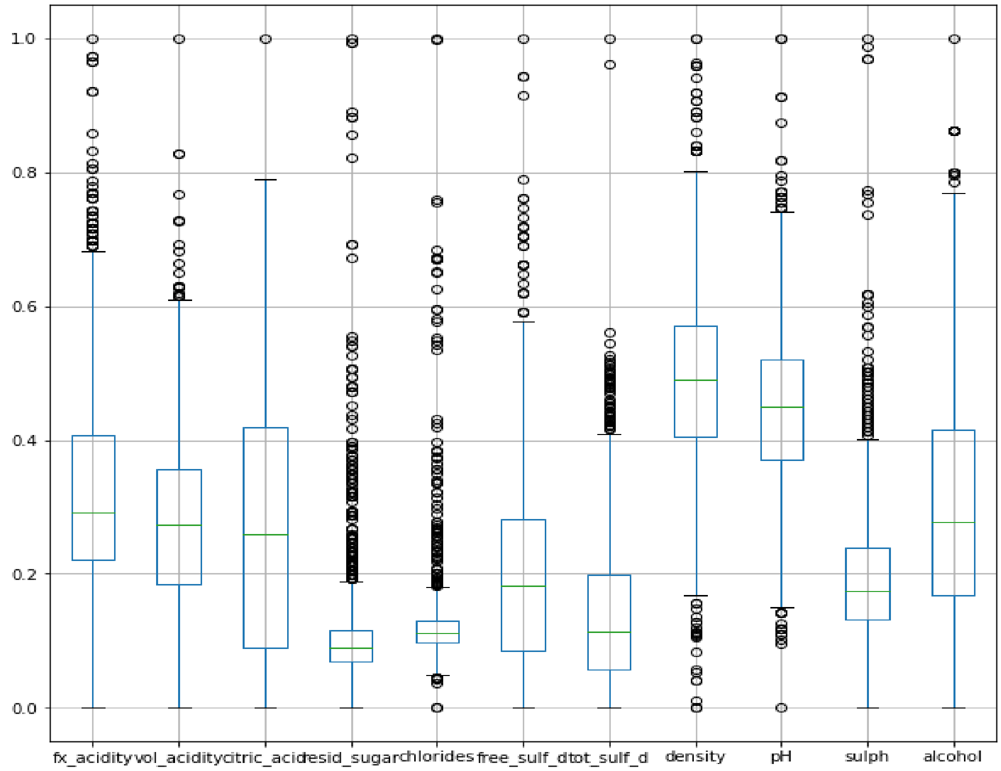

Fig 3.1 : Box plot for all attributes after normalization

C. Normalization :

All attributes are treated as real attributes can be normalized using min-max approach. That is x' = x - min(x) / max(x) - min(x) which gives values between 0 and 1.

D. Feature Subset Selection : At this point it is unclear if any feature is redundant (Except Quality which is obvious) or irrelevant (Except ID) or less important. So, we will use all the features for model development. Also, number of features are relatively less, using all features will not slow down the process.

4. MODEL DEVELOPMENT :

A. DECISION TREE :

Used J48 algorithm in WEKA Java API (weka.classifiers.trees.J48)

Parameter Experimentation :

Different values of confidence threshold were used (0.1 - 0.5). Classifier was used with and without reduced error pruning, with and without subtree raising. The parameter 'confidence factor' is used to test the effectiveness of post-pruning. Subtree raising selects a subtree and replaces it with the child one (ie, a "sub-subtree" replaces its parent)

Confidence threshold of 0.1, without reduced error pruning and with subtree raising gave the best accuracy of 74.359%.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.759	0.270	0.710	0.759	0.734	0.780	Low
0.730	0.241	0.777	0.730	0.753	0.780	High

```
=== Confusion Matrix ===
  a      b     <-- classified as
 565    179    |   a = Low
 231    624    |   b = High
```

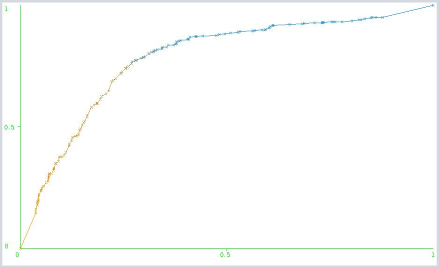

Fig 4.1 ROC curve with area = 0.78

B. A RULES-BASED CLASSIFIER :

Used PART rule based classifier with WEKA java API (weka.classifiers.rules.PART).

Parameter Experimentation :

Different values of confidence threshold were used (0.1-0.5). Classifier was used with and without reduced error pruning. Confidence factor of 0.2 without reduced error pruning gave the best accuracy of 72.79%.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.653	0.207	0.733	0.653	0.691	0.799	Low
0.793	0.347	0.724	0.793	0.757	0.799	High

```
=== Confusion Matrix ===
  a   b   <-- classified as
 486 258 |   a = Low
 177 678 |   b = High
```

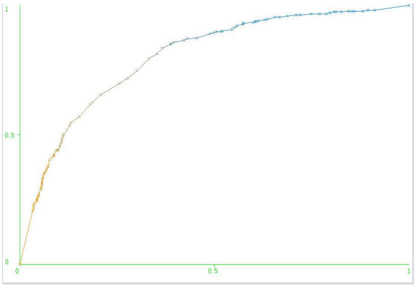

Fig 4.2 ROC curve with area = 0.799

C. NAIVE BAYES :

Used WEKA Java API for classification. (weka.classifiers.Bayes.NaiveBayes)
Accuracy using normal distribution for numerical attributes : 73.045%.
Accuracy using kernel density estimator rather than normal distribution : 74.29%
Kernel density estimator is selected as it gives higher accuracy.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.754	0.267	0.711	0.754	0.732	0.816	Low
0.733	0.246	0.774	0.733	0.753	0.816	High

=== Confusion Matrix ===

```
  a   b   <-- classified as
 561 183 |   a = Low
 228 627 |   b = High
```

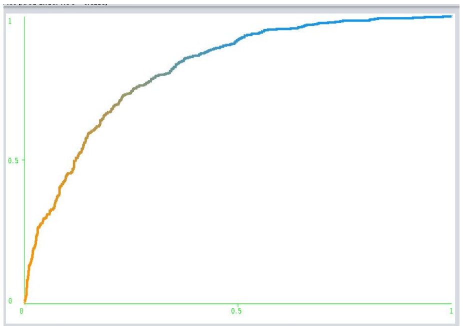

Fig 4.3 ROC curve with area = 0.816

D. ARTIFICIAL NEURAL NETWORK :

Used WEKA Java API (weka.classifiers.functions.MultilayerPerceptron).

Parameter Experimentation :

Learn rate : 0.01,0.1,0.2,0.3,0.4,0.5 Momentum : 0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9 and Epochs : 1,2,3,4,5,6,7,8,9,10,50,100,500 (Tried every combination)
Hidden Layers : 11,10,2 (Experimented with different number of layers, but accuracy did not improve)

Accuracy	Learn Rate	Momentum	# of epochs
75.859 %	0.1	0.1	500*

*with 100 epochs the accuracy is 75.4% which is very close to 75.859% and one might use 100 epochs instead, as it is significantly faster compared to using 500 epochs

```
TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
  0.737    0.222     0.743     0.737    0.740      0.826     Low
  0.778    0.263     0.772     0.778    0.775      0.826     High
```

```
=== Confusion Matrix ===
  a     b     <-- classified as
548   196   |  a = Low
190   665   |  b = High
```

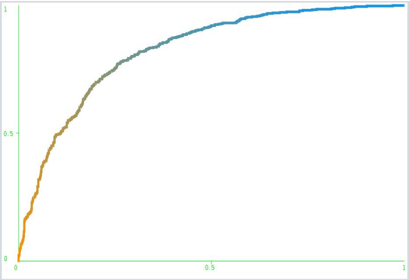

Fig 4.4 ROC curve with area 0.826

E. SUPPORT VECTOR MACHINE :

Used WEKA java API for SVM (weka.classifiers.functions.LibSVM)

Parameter Experimentation :

Grid search for best gamma and C parameters (best - high accuracy). C is the cost of classification and gamma is the parameter of a Gaussian radial basis function used in RBF kernel. Different kernels were chosen for the model. Rest of the parameters used were default in WEKA.

Polynomial Kernel	Linear Kernel	Sigmoid	RBF Kernel
66.03%.	65.79%.	56.03%	73.85%

Best C = 1000.0 and best Gamma = 0.001

RBF kernel is selected as it has highest accuracy.

```
TP Rate  FP Rate  Precision  Recall   F-Measure  MCC   ROC Area  PRC Area  Class
 0.708     0.235     0.724     0.708     0.716     0.474      0.737      0.648     Low
 0.765      0.292    0.751     0.765     0.758     0.474      0.737       0.700    High
```

```
=== Confusion Matrix ===
  a       b       <-- classified as
 527     217     |   a = Low
 201    654      |   b = High
```

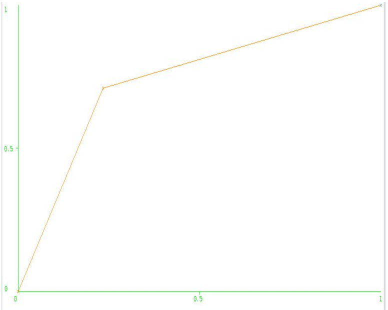

Fig 4.5 ROC Curve with area 0.737

F. ENSEMBLE LEARNER :

Used WEKA Java API (weka.classifiers.trees.RandomForest).

Parameter Experimentation :

Bag size is fixed : Same as training set size

Number of iterations : 10,20,50,70,100,110,150,250

Best accuracy = 83.3% with number of iterations = 100

```
TP Rate  FP Rate  Precision  Recall   F-Measure   ROC Area  Class
   0.821    0.157     0.820       0.821     0.821          0.903      Low
    0.843    0.179     0.844       0.843     0.844          0.903      High
=== Confusion Matrix ===
  a   b   <-- classified as
 611 133 |   a = Low
 134 721 |   b = High
```

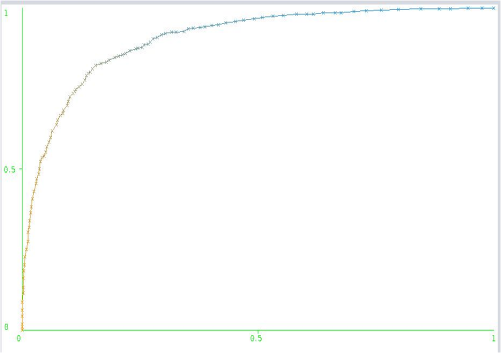

Fig 4.6 ROC with area 0.93

5. MODEL EVALUATION :

Used 10 fold cross validation for measuring performance. Here the data set is partitioned into 10 disjoint sets. In each iteration one set is used as test data and 9 other sets are used as training set. In every iteration the set used is different from those used in previous iterations. Performance is averaged across 10 sets.

6. CONCLUSION :

Ensemble classifier, random forest gives the best accuracy, F-measure and ROC area out of all the classifiers discussed so far. So, ensemble classifier is preferred as modelling approach for this data set.

Pros :

- Best performance results for the given data set.
- Training is fast compared to other approaches (Although it is not the fastest). For Example : Finding an optimal decision tree is NP-hard. Multilayer perceptron uses 100 epochs and has a lot of unknown parameters making it slow in training.
- Reduces the variance of the base classifier.
- Random Forest is simple and straightforward.
- Ensemble classifiers usually give good performance for a given problem and dataset as they combine results of multiple models.

Cons:

- Slow in testing as multiple classifiers (100 trees) are used to classify each test instance(voting). Can be solved by running in parallel (requires multiple cores).
- Consumes a lot of memory as 100 trees and 100 training sets are created and used for classification.