CSE 5243 – Introduction to Data Mining

Homework 4

Sayam Ganguly

ganguly.32@osu.edu

The Ohio State University

Table of Contents

1. Data Analysis

Data analysis has been done using Jupyter Python notebook

1.1 Dataset

The dataset for red wine is taken from UCI repository. The dataset has 1599 records in total with each record having 14 features. The dataset has two classes of wine: High & Low. Fig 1.1 summarizes the dataset.

	count	mean	std	min	25%	50%	75%	max
ID	1599.0	800.000000	461.735855	1.00000	400.5000	800.00000	1199.500000	1599.00000
fx_acidity	1599.0	8.319637	1.741096	4.60000	7.1000	7.90000	9.200000	15.90000
vol_acidity	1599.0	0.527821	0.179060	0.12000	0.3900	0.52000	0.640000	1.58000
citric_acid	1599.0	0.270976	0.194801	0.00000	0.0900	0.26000	0.420000	1.00000
resid_sugar	1599.0	2.538806	1.409928	0.90000	1.9000	2.20000	2.600000	15.50000
chlorides	1599.0	0.087467	0.047065	0.01200	0.0700	0.07900	0.090000	0.61100
free_sulf_d	1599.0	15.874922	10.460157	1.00000	7.0000	14.00000	21.000000	72.00000
tot_sulf_d	1599.0	46.467792	32.895324	6.00000	22.0000	38.00000	62.000000	289.00000
density	1599.0	0.996747	0.001887	0.99007	0.9956	0.99675	0.997835	1.00369
pH	1599.0	3.311113	0.154386	2.74000	3.2100	3.31000	3.400000	4.01000
sulph	1599.0	0.658149	0.169507	0.33000	0.5500	0.62000	0.730000	2.00000
alcohol	1599.0	10.422983	1.065668	8.40000	9.5000	10.20000	11.100000	14.90000
quality	1599.0	5.636023	0.807569	3.00000	5.0000	6.00000	6.000000	8.00000

Fig 1.1

The ‘ID’ attribute is just an identifier ant it will not have any impact on classification, hence it will be removed from the dataset. Also, the quality attribute has been transformed to the ‘class’ labels hence it will also be removed from the dataset before building the model.

1.2 Exploratory Data Analysis

1.2.1 Fixed Acidity

Fig 1.2.1 shows indicates that there is not much difference between the median of the ‘High’ and ‘Low’ class and there is a clear overlap. Also as outliers are present for both the classes hence I’m not removing them as they might have a necessary impact on classification.

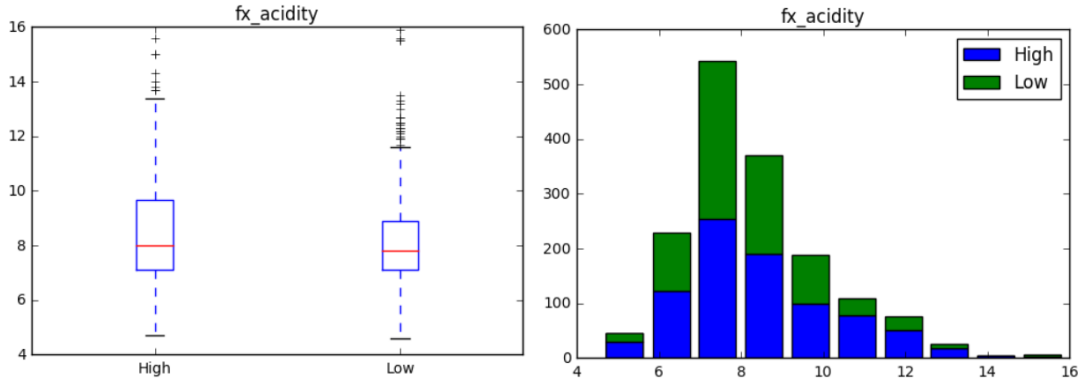


Fig 1.2.1

1.2.2 Volatile Acidity

Fig 1.2.2 indicates that the median for 'High' class is lower than the 'Low' class for 'vol_acidity'. Hence, it can be inferred that the quality of wine decreases with an increase in volatile acidity. There are more outliers for 'Low' than there is for 'High'. Here also outliers are not removed because of the way quality changes with increasing "vol_acidity".

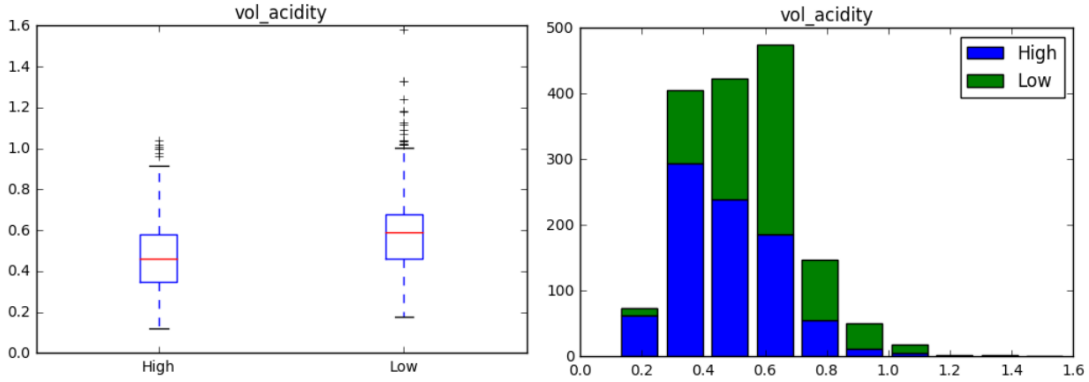


Fig 1.2.2

1.2.3 Citric Acid

Fig 1.2.3 indicates that there is a clear distinction between the median of both the class for 'citric_acid'. It seems that higher values of citric acid contribute to the quality of the wine. There is probably a single outlier for the class low which might be because of noise and it will be eliminated before classification.

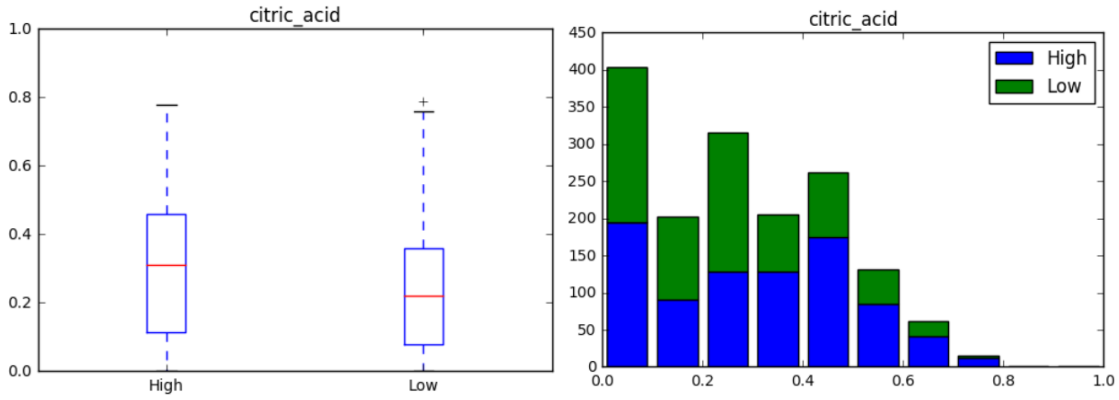


Fig 1.2.3

1.2.4 Residual Sugar

Fig 1.2.4 indicates that both the classes have nearly same median value and similar distribution. There are a large number of outliers in both the classes and it seems from the data that 'resid_sugar' does not have much impact on classifying the quality of wine

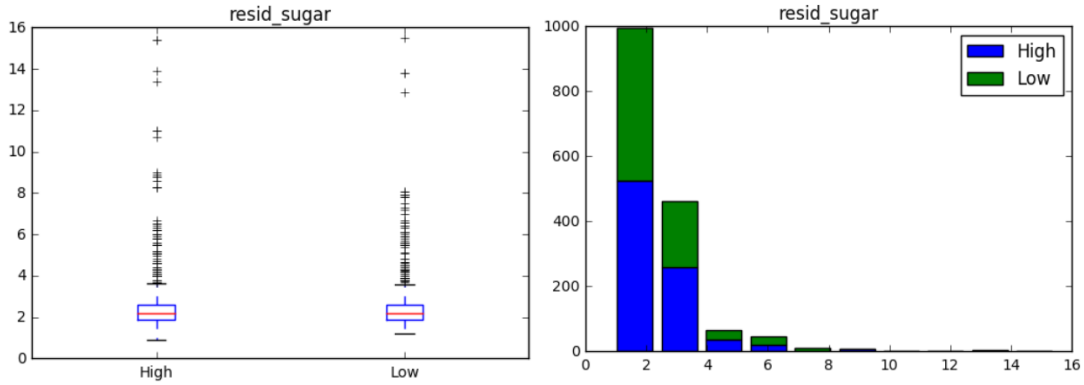


Fig 1.2.4

1.2.5 Chlorides

Fig 1.2.5 indicates a trend very similar to Residual Sugar in case of Chlorides. Though there is a small difference in the median of the two classes but there are many outliers which might impact classification. In particular those outliers will be removed which has a value greater than 0.5.

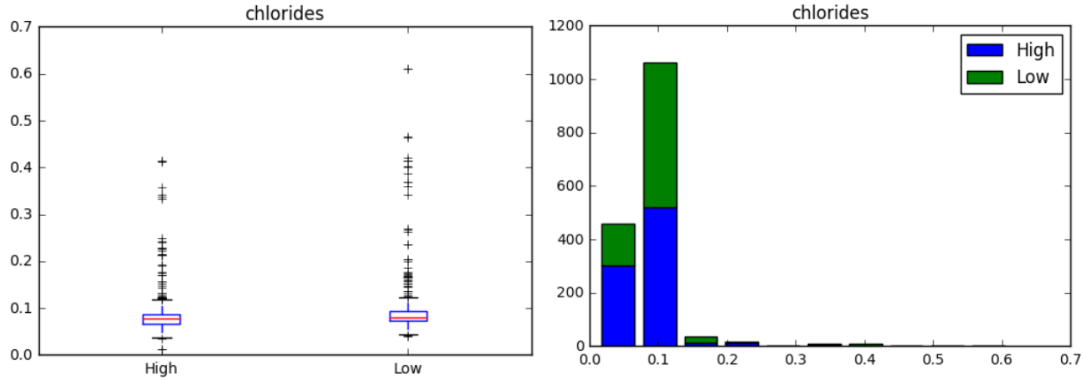


Fig 1.2.5

1.2.6 Free Sulfur Dioxide

Fig 1.2.6 indicates that the median value for the two classes are almost same. There are many outliers for both the classes in the same range, hence it seems that they can be removed.

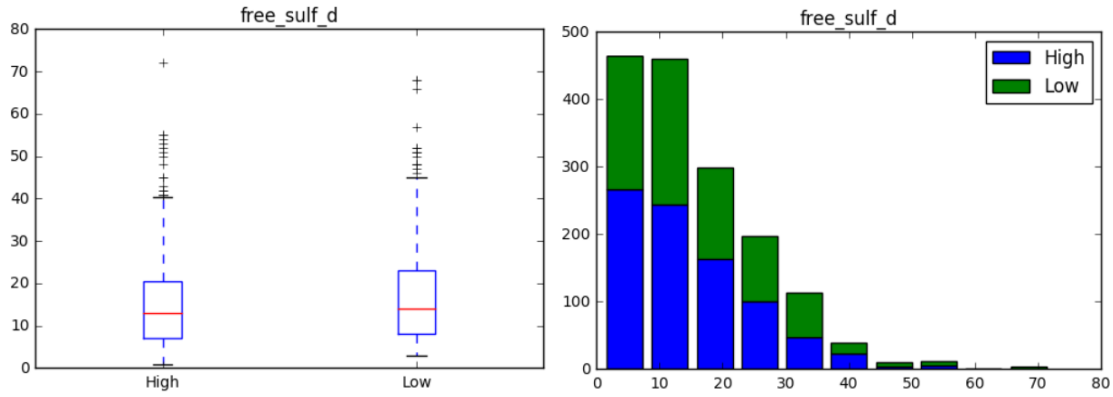


Fig 1.2.6

1.2.7 Total Sulfur Dioxide

Fig 1.2.7 indicates that 'Low' has a higher median than 'High' which means that a lower value contributes to higher quality of wine. There are some outliers for the 'High' class and couple of points have a very high value which seems to be due to noise. Hence, I choose remove outliers >200 value.

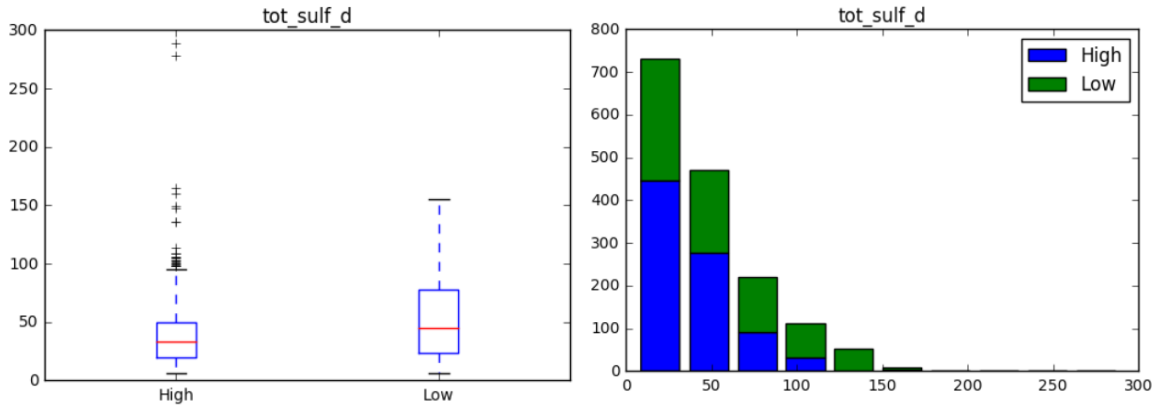


Fig 1.2.7

1.2.8 Density

Fig 1.2.8 indicates that the median density for 'Low' is slightly higher than 'High'. There are outliers that are similarly distributed for both the classes hence they should not have any negative impact.

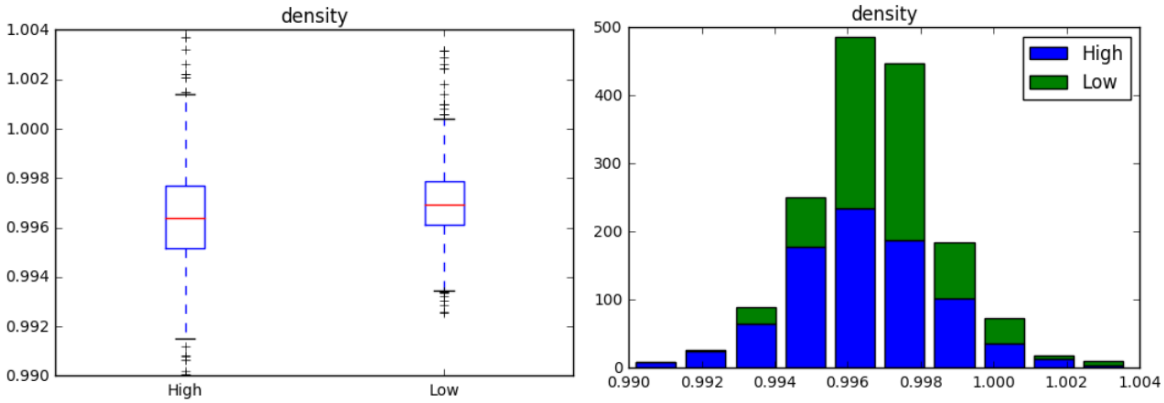


Fig 1.2.8

Fig 1.2.9 pH

Fig 1.2.9 shows that pH has almost identical distribution for both the classes indicating that it might not have much impact on classification

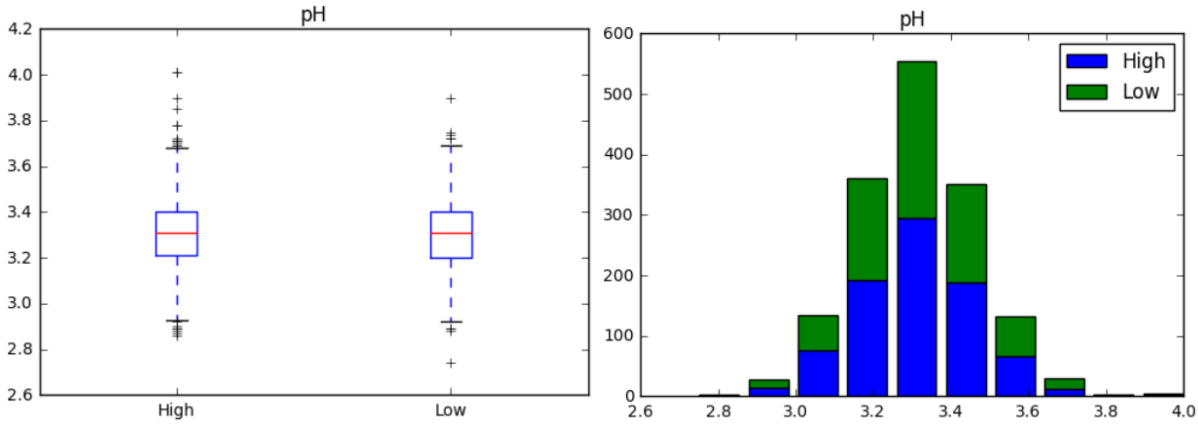


Fig 1.2.9

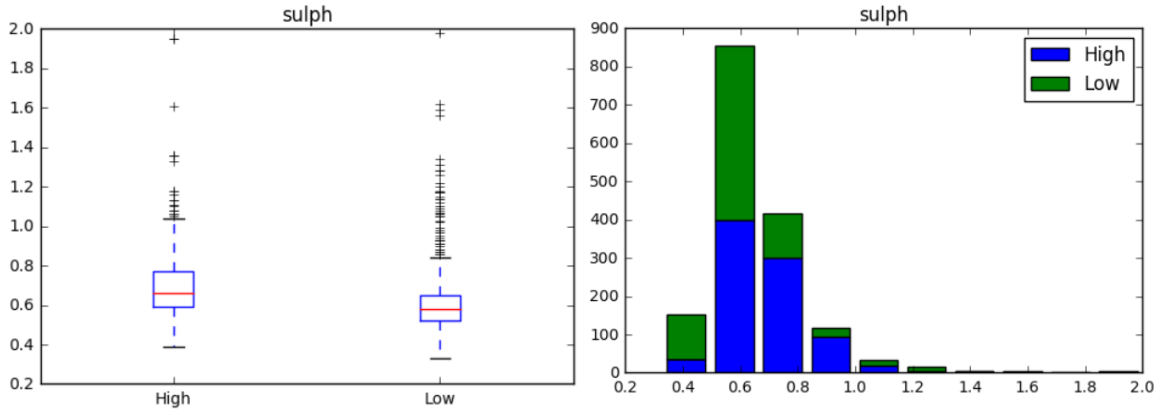


Fig 1.2.10

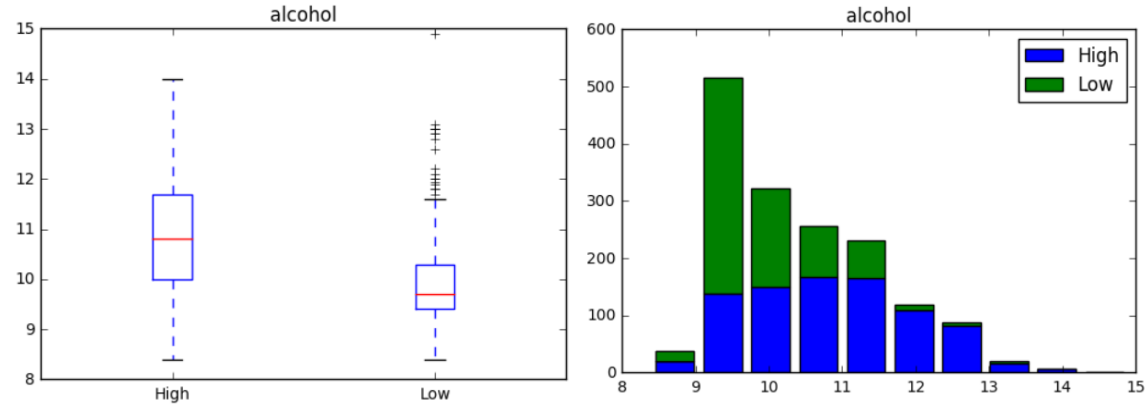


Fig 1.2.11

1.2.11 Alcohol

Fig 1.2.11 indicates that there is again a clear distinction between two classes and a larger amount of alcohol contributing to the overall quality. There are some outliers for 'Low' and particularly there is one which is way higher than the rest of the class. I choose to remove this record.

2. Feature Selection and Transformation

2.1 Feature elimination by Correlation

Table 2.1 shows the correlation values among attributes and Fig 2.1 depicts the heatmap of the same. The last column or the last row is of specific interest as it indicates the correlation of the attributes with the actual class.

	fx_acidity	vol_acidity	citric_acid	resid_sugar	chlorides	free_sulf_d	tot_sulf_d	density	pH	sulph	alcohol	quality
fx_acidity	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-0.682978	0.183006	-0.061668	0.124052
vol_acidity	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	0.234937	-0.260987	-0.202288	-0.390558
citric_acid	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-0.541904	0.312770	0.109903	0.226373
resid_sugar	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.355283	-0.085652	0.005527	0.042075	0.013732
chlorides	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.200632	-0.265026	0.371260	-0.221141	-0.128907
free_sulf_d	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.021946	0.070377	0.051658	-0.069408	-0.050656
tot_sulf_d	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.071269	-0.066495	0.042947	-0.205654	-0.185100
density	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.000000	-0.341699	0.148506	-0.496180	-0.174919
pH	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.341699	1.000000	-0.196648	0.205633	-0.057731
sulph	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.148506	-0.196648	1.000000	0.093595	0.251397
alcohol	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.496180	0.205633	0.093595	1.000000	0.476166
quality	0.124052	-0.390558	0.226373	0.013732	-0.128907	-0.050656	-0.185100	-0.174919	-0.057731	0.251397	0.476166	1.000000

Table 2.1

From both the table as well as the heatmap the attributes ‘resid_sugar’, ‘free_sulf_d’ and ‘pH’ have almost 0 correlation which confirms my previous intuition that these features do not have any impact on classification. Hence these features will not be considered for classification. Also in accordance with analysis before, ‘alcohol’ seems to be the feature that has the most correlation with wine quality.

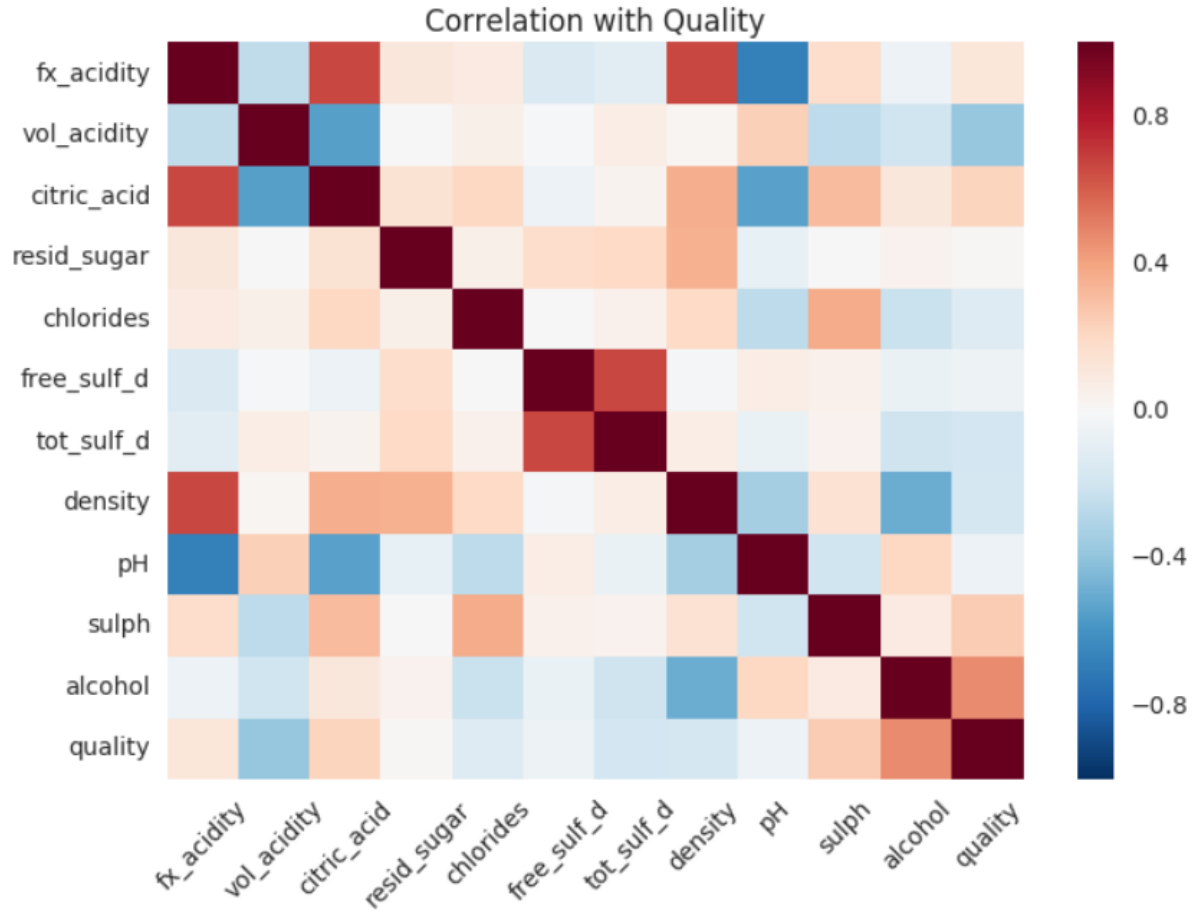


Fig 2.1

Hence the features that are retained for classification are ‘fx_acidity’, ‘vol_acidity, ‘citric_acid’, ‘chlorides’, ‘tot_sulf_d’, ‘density’, ‘sulph’ and ‘alcohol’.

2.2 Missing Values

Total 21 records are removed due to outliers according to the analysis done in the EDA section of which 9 belonged to class 'High' and the remaining 12 belonged to 'Low'. After removal, the dataset now contains 1578 records. The class distribution is shown if fig 2.2

High	846
Low	732

Fig 2.2

2.3 Feature Transformation

Feature normalization has been done for Artificial Neural Network and SVM. For other classifiers, such as Ensemble models, Decision Tree, Rule Based Classifier and Naïve Bayes normalization is not required.

3. Model Development

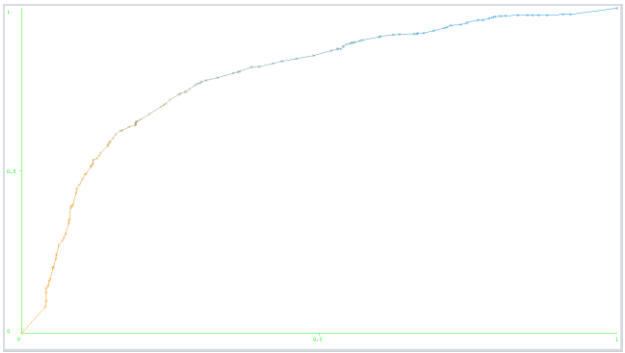
Weka has been used for model development and model evaluation.

3.1 Cross Validation

I'm using 10-fold stratified cross validation provided by Weka to evaluate the models. In this approach in each iteration the dataset gets divided into 10 partitions, then the model gets trained in 9 of the partitions and it's tested on the remaining 1 partition, and this gets repeated 10 times. Also during partitioning Weka makes sure that each partition has almost equal number of samples from both the classes.

3.2 Decision Tree

To build a decision tree model I used the J48 classifier from Weka which is an implementation of the C4.5 Decision Tree algorithm. Initial parameters chosen were 'Minimum number of samples per node = 2' and 'Reduced Error Pruning=false'. This model led to an accuracy of 72.62% with tree height = 15. Next the minimum samples per node was changed to 5 and this led to almost the same accuracy but with a less complex model with only 13 tree levels. Finally reduced error pruning was turned on which led to an accuracy of 73.51% with a tree height of 11. Hence this produced the best result with the simplest model and this is the final model I chose for Decision Tree. Performance metrics and the ROC curve is shown in Fig 3.2



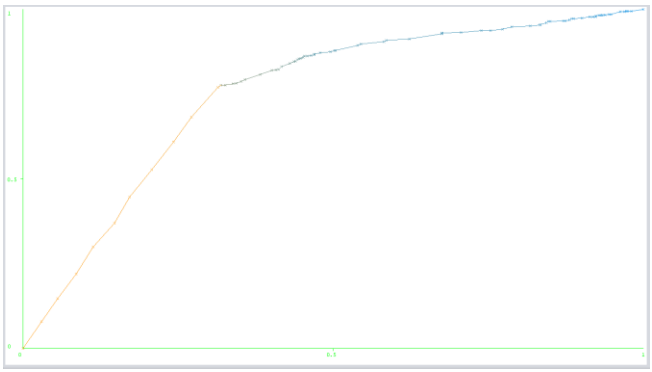
Predicted Class		
Actual Class	Low	High
Low	539	193
High	225	621

Classification Ratio: 73.51%
F-Measure: 0.748
Precision: 0.763
Recall: 0.734

Fig 3.2

3.3 Rule Based Classifier

To build a Rule based Classifier I selected the JRip classifier from Weka. The first run of the classifier was done without Reduced Error Pruning and it yielded an accuracy of 74.77%. The ruleset for this model consisted of 24 rules. Next the model was built again with Reduced Error Pruning. Accuracy went down by 1 percentage point but the model consisted of only 9 rules. Hence, I selected the later model as my final Rule base model as it is a much simpler model than the former with reasonable accuracy. The performance metrics of this model are shown n Fig 3.3



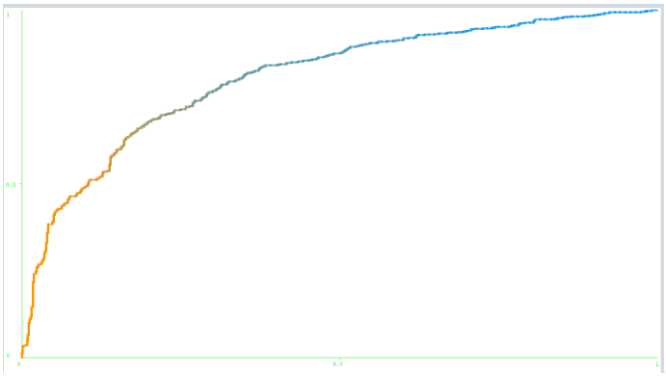
Predicted Class		
Actual Class	Low	High
Low	502	230
High	195	651

Classification Ratio: 73.06%
F-Measure: 0.754
Precision: 0.739
Recall: 0.770

Fig 3.3

3.4 Naïve Bayes Classifier

The standard Naïve Bayes Classifier from Weka was used to build Naïve Bayes classifier for this dataset. The model performed with an accuracy of 73.06%. Different performance metrics of this model and ROC curve is shown in figure 3.4.



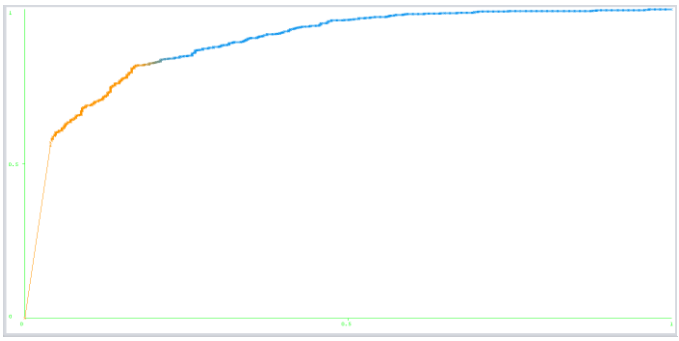
Predicted Class		
Actual Class	Low	High
Low	550	182
High	243	651

Classification Ratio: 73.06%
F-Measure: 0.739
Precision: 0.768
Recall: 0.713

Fig 3.4

3.5 AdaBoost

To build AdaBoost classifier I used the AdaBoost M1 classifier provided by Weka. For first run I used a Decision Stump as the base class with 10 iterations and got an accuracy of 73.57%. Then I changed the number of iterations to 30 and got an accuracy of 75.72%. After this any increase in number of iterations did not improve the accuracy anymore. Then I changed the base class to a J48 Decision Tree and this resulted in an accuracy of 81.36%. Although this is a complex model but the sharp improvement in accuracy led to the selection of this as the final AdaBoost model. Model performance metric and ROC curve are shown in fig 3.5



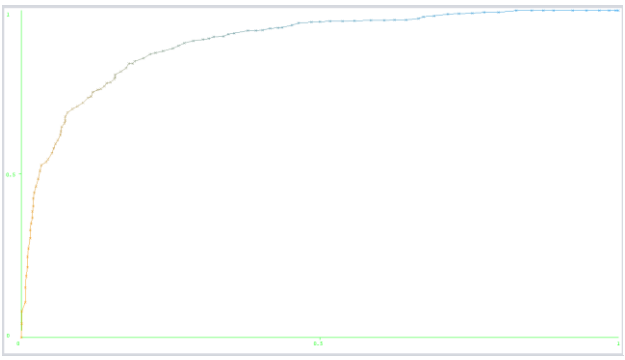
Predicted Class		
Actual Class	Low	High
Low	585	147
High	147	699

Classification Ratio: 81.36%
F-Measure: 0.826
Precision: 0.826
Recall: 0.826

Fig 3.5

3.6 Random Forrest

Standard Random Forrest classifier from Weka is used for this. Initial run was performed on the default parameters with 100% training dataset in each iteration which resulted in an accuracy of 82.06%. Next the model was run with 70% of the training dataset in each iteration and the accuracy got improved to 82.88%. Model metrics are shown in fig 3.6.



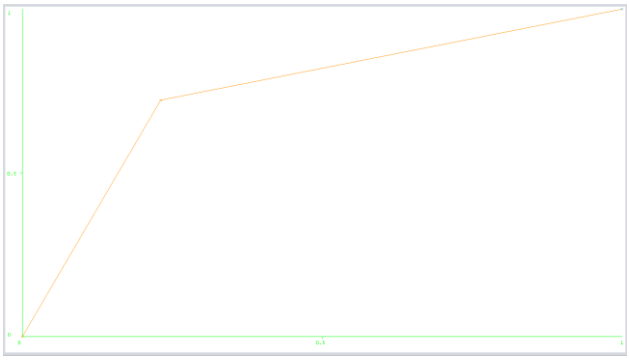
Predicted Class		
Actual Class	Low	High
Low	600	132
High	138	798

Classification Ratio: 82.88%
F-Measure: 0.840
Precision: 0.843
Recall: 0.837

Fig 3.6

3.7 Support Vector Machine

To implement SVM I'm using Weka's SMO classifier with Linear Kernel. This model achieved an accuracy of 74.33%. Changing to other kernels such as RBF or polynomial did not improve accuracy. Fig 3.7 shows the performance metric for this model.



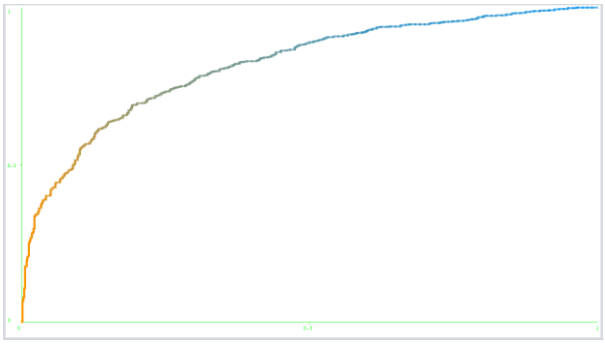
Predicted Class		
Actual Class	Low	High
Low	563	169
High	236	610

Classification Ratio: 74.33%
F-Measure: 0.751
Precision: 0.783
Recall: 0.721

Fig 3.7

3.8 Artificial Neural Network

For Artificial Neural Network classifier, the Multi Layer Perceptron model from Weka was used. This model uses the back propagation algorithm. After several variation of model parameters such as learning rate and number of layers the maximum accuracy of 74.01% was achieved using a learning rate of 0.3 and the number of hidden layers 5. Fig 3.8 shows the model performance metrics.



Predicted Class		
Actual Class	Low	High
Low	560	172
High	238	608

Classification Ratio: 74.33%
F-Measure: 0.748
Precision: 0.779
Recall: 0.719

Fig 3.8

4 Model Evaluation

4.1 Evaluation based on Receiver Operating Characteristics(ROC) curves

The ROC curves for all the models are shown in fig 4.1.1. It has been plotted using Weka Knowledge Flow module that allows evaluating multiple classifiers together. Fig 4.1.2 depicts the architecture of the knowledge flow network.

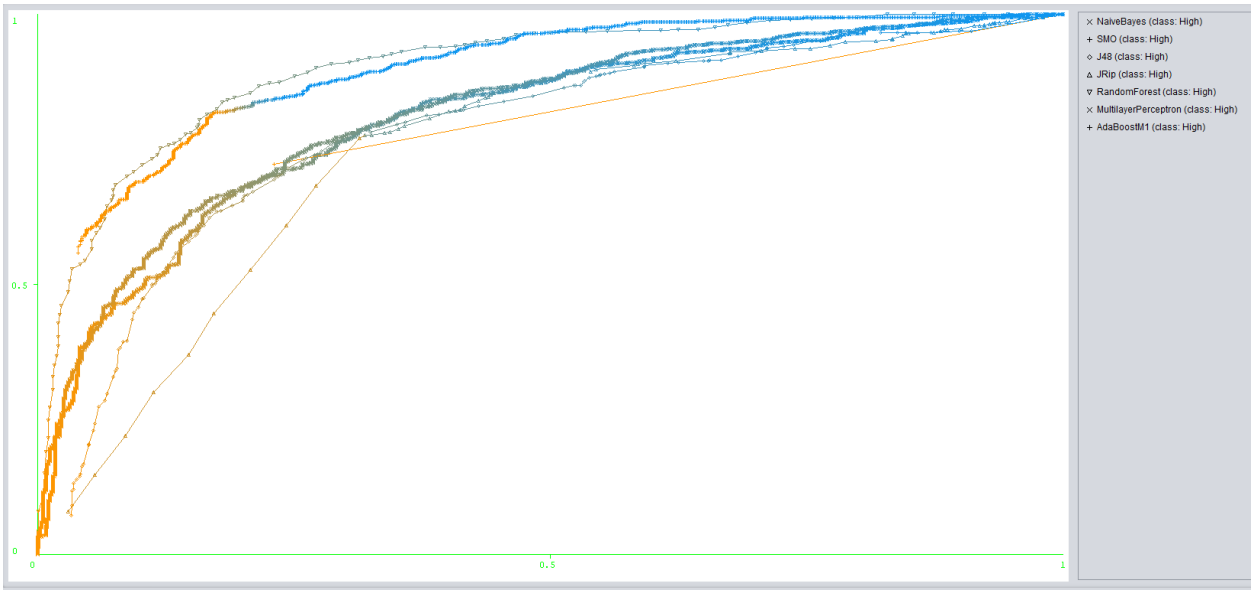


Fig 4.1.1

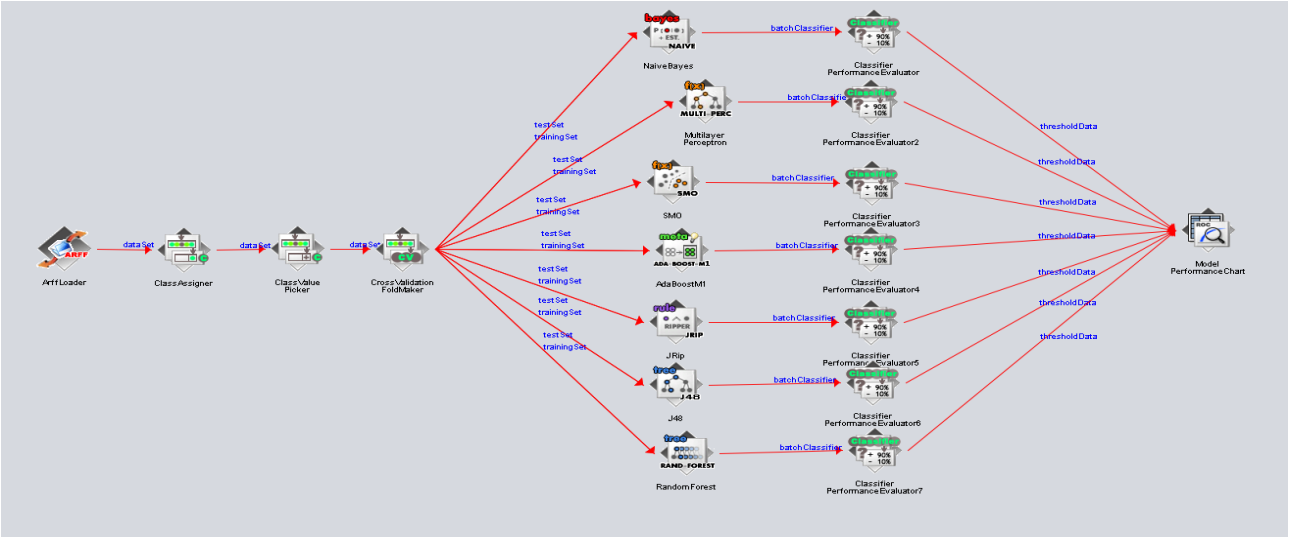


Fig 4.1.2

Fig 4.1.1 suggests that Random Forrest has the highest lift and AUC. Hence Random Forrest performed best for this training data. AdaBoost is also very close to Random Forrest and it even surpasses Random Forrest's performance a few times in the ROC curve.

4.2 Evaluation based on Accuracy and F-measure

Table 4.2 summarizes the accuracy and F-measure for different models on the Wine Dataset. Precision and Recall for each classifier has also been listed before in section 3 for each of the classifiers.

Classifier	Accuracy	F-measure
Decision Tree	73.51	0.748
Rule Based	73.06	0.754
Naïve Bayes	73.06	0739
AdaBoost	81.36	0.826
Random Forrest	82.88	0.840
Support Vector Machine	74.33	0.751
Artificial Neural Network	74.01	0.719

From table 4.2 also it is evident that Random Forrest is the best performing model closely followed by AdaBoost.

4.3 Conclusion

The two best performing classifiers Random Forrest and AdaBoost improved accuracy by combining weaker models together. But that comes at a computational cost. These models are quite complex and they take around a minute to train even in this reasonably small dataset of only 1578 records, whereas SVMs or Decision Trees take only seconds to train. Hence there will be a compromise between accuracy and speed when it comes to larger datasets. If speed is of priority then a simpler model like SVM, Decision Tree or Rule based might be preferred. On the other hand Ensemble models will be preferred where accuracy is important.