CSI 775

GRAPHICAL MODEL FOR INFERENCE AND DECISION MAKING

BAYESIAN CLASSIFICATION OF WINE QUALITY DATASET

MANVI JAIN.....G00989167

TABLE OF CONTENTS

ABSTRACT

The main objective of the project was to identify the features which majorly affect the quality of the red wine and then by using those features classify the red wines. Red wine data consist of 12 variables and 1599 data points. Among those 12 variables 'quality' is the dependent variable. Various chemicals can affect the quality of wine in different ways and to produce better quality wines and understand which chemical adds more zest to wine and which doesn't, this problem is quite important. This report applies Bayesian Networks to freely available wine quality data and attempts to classify into low and high quality wine. A Naives Bayes network andl TAN and hill climbing BN are. Predictive accuracy results show that even when compared to more complicated TANs and causal networks, the naives bayes classifier performs strongly.

INTRODUCTION

Red Wine data has been studied in this project to select the best features of the red wine. After selecting the best features, we have used the selected features to classify the quality of the red wines. The red wine data has been taken from the UCI Machine Learning Repository. This data is related to the Portuguese "Vinho Verde" wine. The data consist of 12 variables and 1599 data points. The 12 variables which are present in data are given below along with their significance,

- Fixed acidity – Acids give the sourness or tartness "a fundamental feature in wine taste". Reduced acidity makes for a "flat" taste. Examples: tartaric, malic, citric and succinic acids. They are all found in grapes except for succinic.
- Volatile acidity – Too much volatile acidity is undesirable. The main acid of the issue is acetic acid. This acid is to be distilled from the wine, leaving only fixed acids.
- Citric acid – Most citric acid in the grapes is consumed during fermentation. It is sometimes added to wine to acidify it and give it "freshness". However, this can lead to microbial growth. Tartaric acid is sometimes used instead.
- Residual sugar – This is the sugar left in the wine from the grapes. These are the sweet wines. The Brix scale is used to track sugar development and determine when to harvest.
- Chlorides – The saltiness of wine. Chlorides are a result of the grapes used.
- Free sulfur dioxide – Not bound to any compound in the wine and known as sulfites. It acts as an anti-microbial agent which limit the growth of harmful yeast/bacteria.
- Total sulfur dioxide – All sulfur dioxide in the wine, bound and free.
- Density – A comparison of the weight of a certain volume of wine to an equivalent volume of water.
- pH – It describes the acidity. If pH is lower, higher the acidity and it relates to the fixed acidity of the wine. There are "buffer acids" that do not contribute to acidity but help keep the pH level.
- Sulfates – Sulfates are mineral salts containing sulfur. They can be a byproduct of animal or plant decay as well as industrial processes. Sulfates may relate to fermenting nutrition, which affects wine aroma.
- Alcohol – Sugars are converted to alcohol in the yeast fermentation. Too much alcohol compared to other components leads "hot" wine.
- Quality – Quality is the parameter which describes the quality of the wine in-between 1 to 10. The best wine has a quality rating as 8 and worst wine has a quality rating as 3.

From the above 12 explained variables in the data, quality is the dependent variable which is dependent on other variables.

To analyze the wine data, we started with the Exploratory data analysis and data preprocessing which is explained in Section 2. After preprocessing we used that data for feature selection and detail explanation of feature selection is provided in section 3. Based on the selected features we built the models and test those models for predicting the quality of the red wines. Details about model building are explained in section 4. Further, we used those models to classify the red wines and results of classification are explained in section 5. The findings of this analysis are concluded in section 6 of the report.

This is how the dataset looks like:-

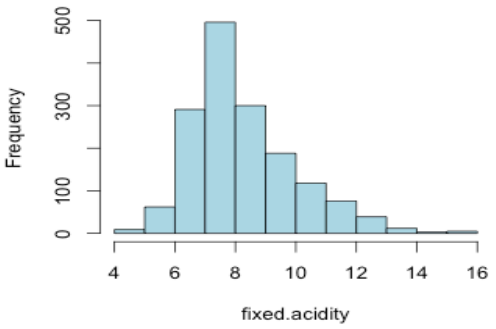
fixed acidity	volatile acidi	citric acid	residual suga	chlorides	free sulfur di	total sulfur d	density	pH	sulphates	alcohol	quality
7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5
7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	5
11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	6
7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7.4	0.66	0	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	5
7.9	0.6	0.06	1.6	0.069	15	59	0.9964	3.3	0.46	9.4	5
7.3	0.65	0	1.2	0.065	15	21	0.9946	3.39	0.47	10	7
7.8	0.58	0.02	2	0.073	9	18	0.9968	3.36	0.57	9.5	7
7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	5
6.7	0.58	0.08	1.8	0.097	15	65	0.9959	3.28	0.54	9.2	5
7.5	0.5	0.36	6.1	0.071	17	102	0.9978	3.35	0.8	10.5	5
5.6	0.615	0	1.6	0.089	16	59	0.9943	3.58	0.52	9.9	5
7.8	0.61	0.29	1.6	0.114	9	29	0.9974	3.26	1.56	9.1	5
8.9	0.62	0.18	3.8	0.176	52	145	0.9986	3.16	0.88	9.2	5
8.9	0.62	0.19	3.9	0.17	51	148	0.9986	3.17	0.93	9.2	5
8.5	0.28	0.56	1.8	0.092	35	103	0.9969	3.3	0.75	10.5	7
8.1	0.56	0.28	1.7	0.368	16	56	0.9968	3.11	1.28	9.3	5
7.4	0.59	0.08	4.4	0.086	6	29	0.9974	3.38	0.5	9	4
7.9	0.32	0.51	1.8	0.341	17	56	0.9969	3.04	1.08	9.2	6

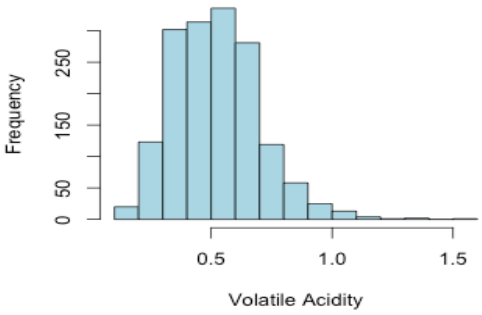
Looking at the dataset it is hard to understand and interpret the continuous values of wine data. This raises the scope for uncertainty in terms of what quantity of which component would be in the right range to produce a low, medium or good quality wine.

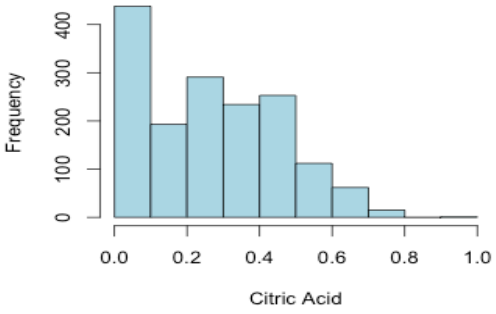
Data preprocessing mainly deals with the checking the quality of the data and make the data ready for the further analysis. In data preprocessing an analysis was done on the data for missing values and outliers. Different plots were presented to search the incomplete values in the data and to determine the quality of the data. The data which I obtained was in a .csv file and I have used MS-Excel initially to import the data and then used R to analyze the data. All the variables in the data are plotted using histograms and boxplots. Further, correlations were calculated using Pearson's correlation method to enhance the understanding of variables.

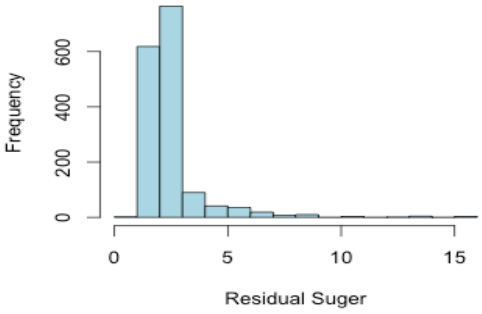
HISTOGRAMS

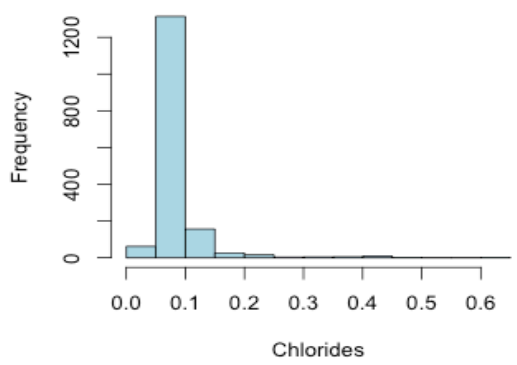
The Histogram is a graphical representation of the distribution of numerical data. The purpose of the histogram is to graphically summarize the distribution of a data. We have used the histograms for studying the spread and skewness of the data. The histograms for all the variables are as follows:

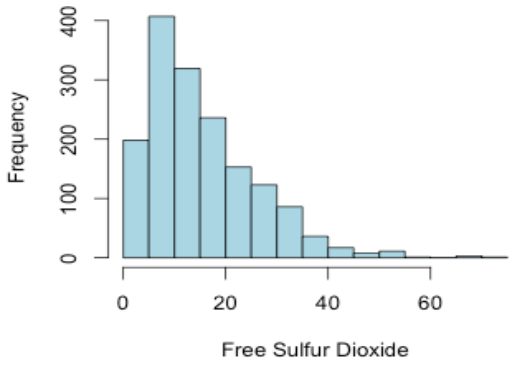


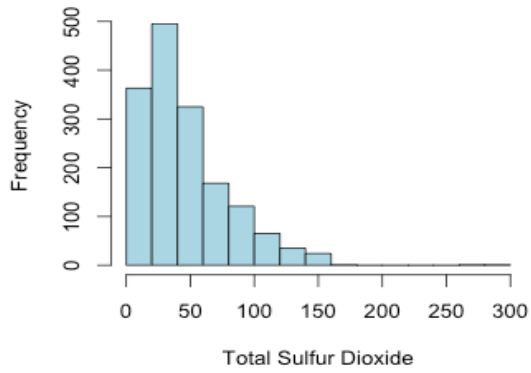


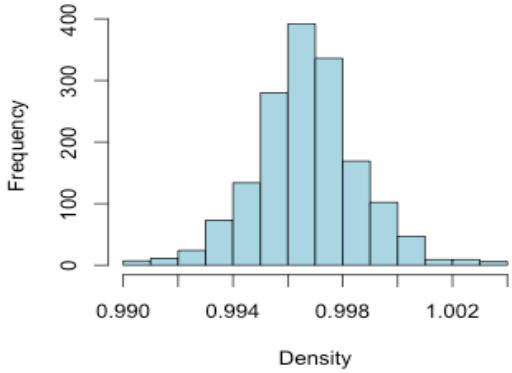








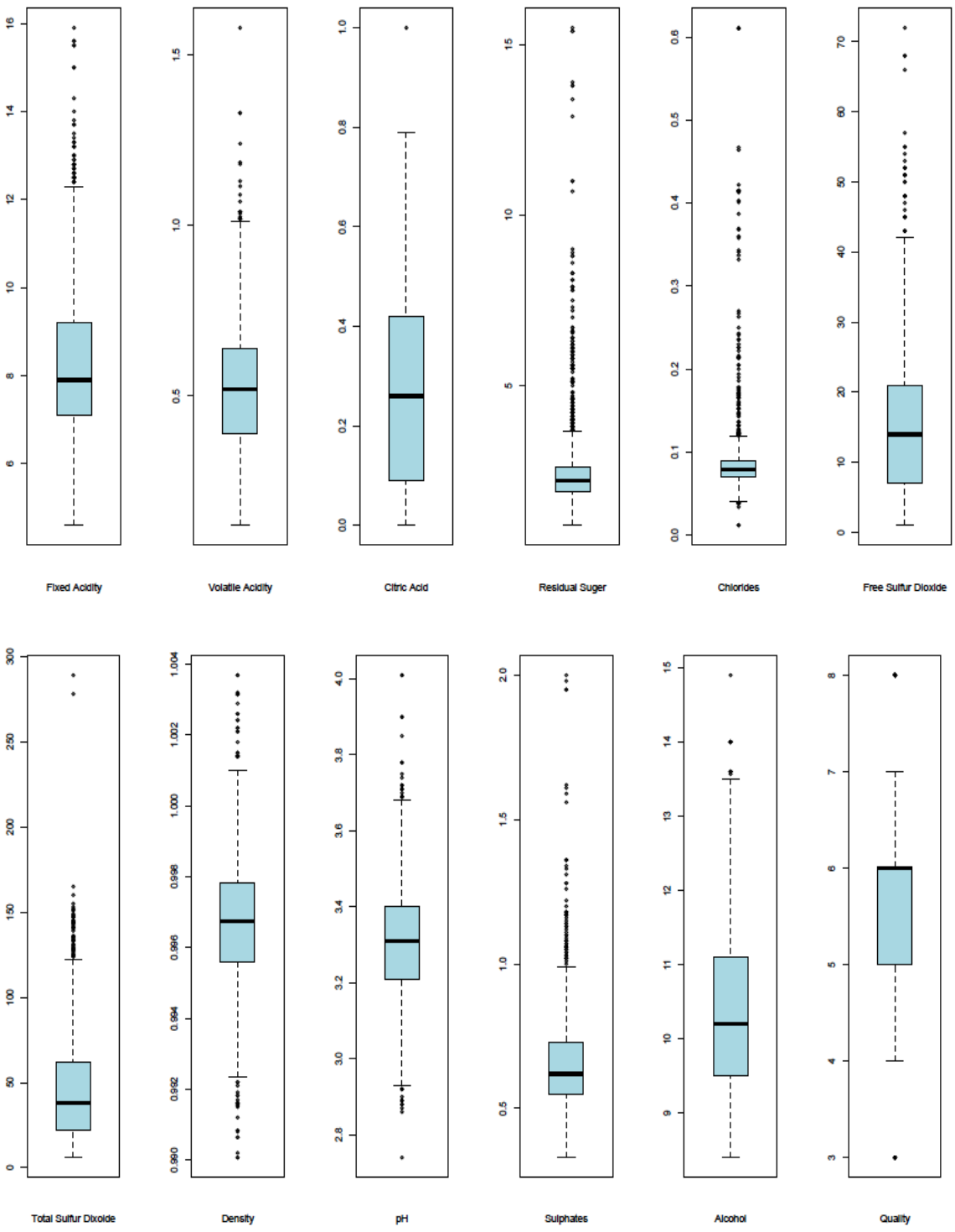




It can be observed from above plots that majority of the histograms are skewed right. The variables such as Fixed acidity, Volatile acidity, Citric acid, Residual sugar, Chlorides, Free sulfur dioxide, Total sulfur dioxide, Sulphates, and Alcohol are right skewed. It is also observed from the histograms of variable pH and Density that, pH is slightly right skewed and Density is symmetric.

BOX PLOTS

Box plots are excellent tools for understanding the range of the data and detecting the outliers. The box plots also explain about the range, mean and median about the data. The main purpose of the box plots used here is to detect the outliers and to know about the data features like mean, median etc. Box plots for all the variables are given on the next page:



From the above plot, it is observed that more outliers are present in the variables Fixed acidity, Volatile acidity, Residual sugar, chlorides, Free sulfur dioxide, Total sulfur dioxide, pH, and Sulphates. On the other hand, variables like Citric acid and Alcohol have fewer outliers. It is also observed that for most of the variables outliers are present on the higher end of the plot. The summary of the data from the box plots can be given as follows:

Parameters	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
Fixed Acidity	4.60	7.10	7.90	8.32	9.20	15.90
Volatile acidity	0.1200	0.3900	0.5200	0.5278	0.6400	1.5800
Citric Acid	0.0000	0.090	0.260	0.271	0.420	1.000
Residual Sugar	0.900	1.900	2.200	2.539	2.600	15.500
Chlorides	0.01200	0.07000	0.07900	0.08747	0.09000	0.61100
Free Sulfur Dioxide	1.00	7.00	14.00	15.87	21.00	72.00
Total Sulfur Dioxide	6.00	22.00	38.00	46.47	62.00	289.00
Density	0.9901	0.9956	0.9968	0.9967	0.9978	1.0037
pH	2.740	3.210	3.310	3.311	3.400	4.010
Sulphates	0.3300	0.5500	0.6200	0.6581	0.7300	2.000
Alcohol	8.40	9.50	10.20	10.42	11.10	14.90
Quality	3.000	5.000	6.00	5.636	6.000	8.000

Table 1: Summary of data with outliers

As it is observed from the box plots and summary of the data that it very important to remove the outliers from the data. It is also observed that majority of the outliers are present to the higher end of the plot. So, I have only removed the outliers from the higher end of the values. The formula which we used for the removal of the outliers is as follows:

$$Max = Q_3 + 1.5 * IQR$$

The above method is called as Tukey's method and this method depends on the interquartile range of the data and 3rd quartile. So the values which are greater than Max are considered as outliers and are removed from the data. We also observed that there are no missing values present in the data and in this way prepared the data for the further analysis.

Pearson's correlation

The main purpose of the Pearson's correlation is getting the brief idea about the relationship of one variable with another variable. By using Pearson's correlation, we understand that which variables are more important for quality. The table for the correlations of all the variables is given below:

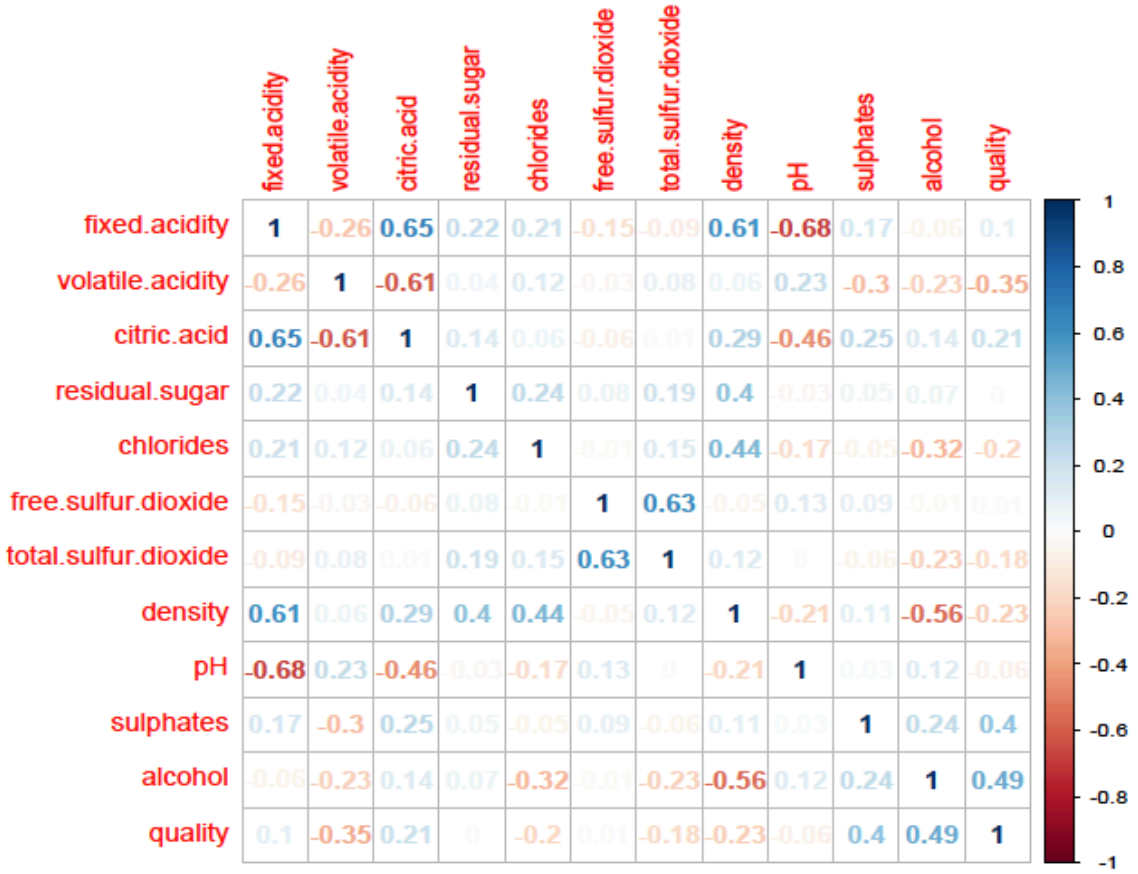


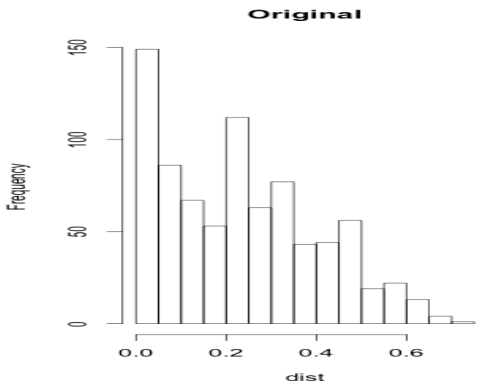
Table 2: Pearson's Correlation of all the variables

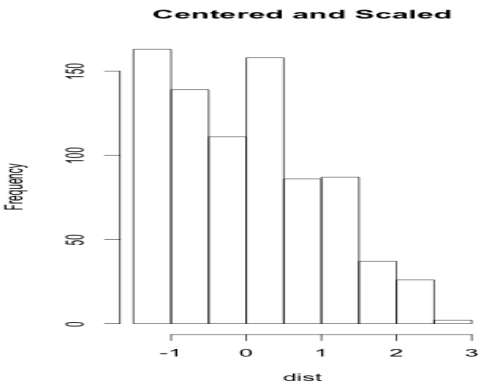
From the table 2, it can be observed that Volatile acidity (-0.35), Chlorides (-0.2), Total sulfur dioxide (-0.18) and density (-0.23) have a negative correlation with the quality of the wine. On the other hand, Sulphates (0.4), Alcohol (0.49), Fixed acidity (0.1) and Citric acid (0.21) have a positive correlation with the quality of the wine. There are some variables such as Residual sugar and Free sulfur dioxide which are not showing any relation with the quality of the wine. It is also observed that pH (-0.06) have a very less negative correlation with the quality.

PREPROCESSING

Centering transformation is basically reducing the Mean value of samples from all observations. So, the observations will have a mean value of Zero after this transformation. Scaling transformation is dividing value of predictor for each observation by standard deviation of all samples. This will cause the transformed values to have a standard deviation of One.

After outlier removal, checked the data for normal distribution, and performed centering and scaling to attain almost normalized data and to produce a better classifier.





DISCRETIZING THE DATA

In order to preprocess data to better learn Bayesian Networks.
Screen and transform the data to make them more suitable for structure and parameter learning.
"discretize" function from bnlearn is used for the same to bin all the continuous variables.
The quality variable is factorized to give 3 categories
5: Low
6 : Medium
7 : High

```
    fixed.acidity      volatile.acidity           citric.acid      residual.sugar
[6.09,8.17]:476   [0.269,0.515]:359     [-0.00073,0.243]:448   [1.6,2.27] :451
(8.17,10.2]:268   (0.515,0.76] :394     (0.243,0.487]   :273   (2.27,2.93]:302
(10.2,12.3]: 65   (0.76,1.01]  : 56     (0.487,0.731]   : 88   (2.93,3.6] : 56
         chlorides   free.sulfur.dioxide  total.sulfur.dioxide
[0.0609,0.0803]:409   [2.96,16]:491       [5.88,44.7]:473
(0.0803,0.0997]:319   (16,29]  :252       (44.7,83.3]:236
(0.0997,0.119] : 81   (29,42]  : 66       (83.3,122] :100
        density              pH            sulphates         alcohol
[0.995,0.997]:371   [3.12,3.31]:346   [0.469,0.64]:506   [8.7,10.1] :439
(0.997,0.999]:364   (3.31,3.49]:383   (0.64,0.81] :243   (10.1,11.4]:297
(0.999,1]    : 74   (3.49,3.68]: 80   (0.81,0.981]: 60   (11.4,12.8]: 73
quality
5:388
6:348
7: 73
```

FEATURE SELECTION

Feature selection is the process of selecting different relevant parameters or variables which are most important to the data. Feature selection is also called as a variable selection or attribute selection. Feature selection techniques are used for the three reasons:

- Simplification of models to make them easier to interpret.
- Shorten training times.
- Enhanced generalization by reducing overfitting.

The main theme behind the feature selection is that there are many features present in the data. Some of them are irrelevant or redundant and if we remove those features it will not result in much loss of information. But if we select the most important features of the data then it can be used for building the good predictive model.

For the red wine, 11 features are present in the data and we need to select the features which majorly affect the quality of the red wine. The red wine data initially consist of 1599 data points but after removal of outliers, it reached to 1200. For feature selection, we have used two methods Multiple Linear Regression and Random Forest. The details about the feature selection are as follows:

MULTIPLE LINEAR REGRESSION

Multiple Linear Regression is implemented using an automated method known as stepwise forward selection. The method stepwise forward selection starts with no variables in the model and then testing the addition of each variable using a chosen model fit criterion. After adding the variable automatically, the model fit criterion is calculated and if the variable is giving the improvement in the fit then that variable is selected. So this process is repeated for all the variables and combinations of all the variables for selecting the best variables for the model

Akaike Information Criterion (AIC) is used for the model comparison in stepwise forward regression. The AIC is a measure of the relative quality of statistical models for a given set of data. AIC estimates the quality of each model relative to each of the other model. So, AIC provides a means for model selection. The AIC value of the model can be computed as

$$AIC = 2K - 2 * \ln(L)$$

Where, L is the maximum likelihood function of the model and k is the number of estimated parameters in the model. The results obtained from the forward stepwise regression are given on the next page.

```
Start:  AIC=-1032.41
quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
    chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
    density + pH + sulphates + alcohol + quality


Step:  AIC=-1032.41
quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
    chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
    density + pH + sulphates + alcohol

                      Df Sum of Sq    RSS      AIC
- free.sulfur.dioxide  1    0.0018 219.20 -1034.40
- residual.sugar       1    0.0574 219.26 -1034.20
- pH                   1    0.1219 219.32 -1033.96
- density              1    0.3417 219.54 -1033.15
<none>                             219.20 -1032.41
- fixed.acidity        1    0.5897 219.79 -1032.24
- citric.acid          1    1.0917 220.29 -1030.39
- chlorides            1    1.5954 220.79 -1028.54
- total.sulfur.dioxide 1    1.6215 220.82 -1028.45
- volatile.acidity     1    3.3379 222.54 -1022.18
- sulphates            1   10.9771 230.18  -994.88
- alcohol              1   14.2217 233.42  -983.55

Step:  AIC=-1034.4
quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
    chlorides + total.sulfur.dioxide + density + pH + sulphates +
    alcohol

                      Df Sum of Sq    RSS      AIC
- residual.sugar       1    0.0585 219.26 -1036.19
- pH                   1    0.1252 219.32 -1035.94

Step:  AIC=-1034.4
quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
    chlorides + total.sulfur.dioxide + density + pH + sulphates +
    alcohol

                      Df Sum of Sq    RSS      AIC
- residual.sugar       1    0.0585 219.26 -1036.19
- pH                   1    0.1252 219.32 -1035.94
- density              1    0.3425 219.54 -1035.14
<none>                             219.20 -1034.40
- fixed.acidity        1    0.5886 219.79 -1034.23
- citric.acid          1    1.1143 220.31 -1032.30
- chlorides            1    1.5947 220.79 -1030.54
- total.sulfur.dioxide 1    2.9344 222.13 -1025.65
- volatile.acidity     1    3.4350 222.63 -1023.82
- sulphates            1   11.1715 230.37  -996.19
- alcohol              1   14.4233 233.62  -984.85

Step:  AIC=-1036.19
quality ~ fixed.acidity + volatile.acidity + citric.acid + chlorides +
    total.sulfur.dioxide + density + pH + sulphates + alcohol

                      Df Sum of Sq    RSS      AIC
- pH                   1    0.1691 219.43 -1037.56
- density              1    0.2899 219.55 -1037.12
- fixed.acidity        1    0.5359 219.79 -1036.21
<none>                             219.26 -1036.19
- citric.acid          1    1.1694 220.43 -1033.88
- chlorides            1    1.5674 220.83 -1032.43
- total.sulfur.dioxide 1    2.8974 222.16 -1027.57
- volatile.acidity     1    3.5047 222.76 -1025.36
- sulphates            1   11.2359 230.49  -997.76
- alcohol              1   21.1933 240.45  -963.54
```

```
Step:  AIC=-1037.56
quality ~ fixed.acidity + volatile.acidity + citric.acid + chlorides +
    total.sulfur.dioxide + density + sulphates + alcohol

                      Df Sum of Sq    RSS      AIC
<none>                             219.43 -1037.56
- density              1    0.9451 220.37 -1036.09
- citric.acid          1    1.1358 220.56 -1035.39
- chlorides            1    1.4296 220.86 -1034.31
- fixed.acidity        1    2.2807 221.71 -1031.20
- total.sulfur.dioxide 1    2.7985 222.23 -1029.31
- volatile.acidity     1    3.4429 222.87 -1026.97
- sulphates            1   11.1597 230.59  -999.43
- alcohol              1   23.8933 243.32  -955.95
```

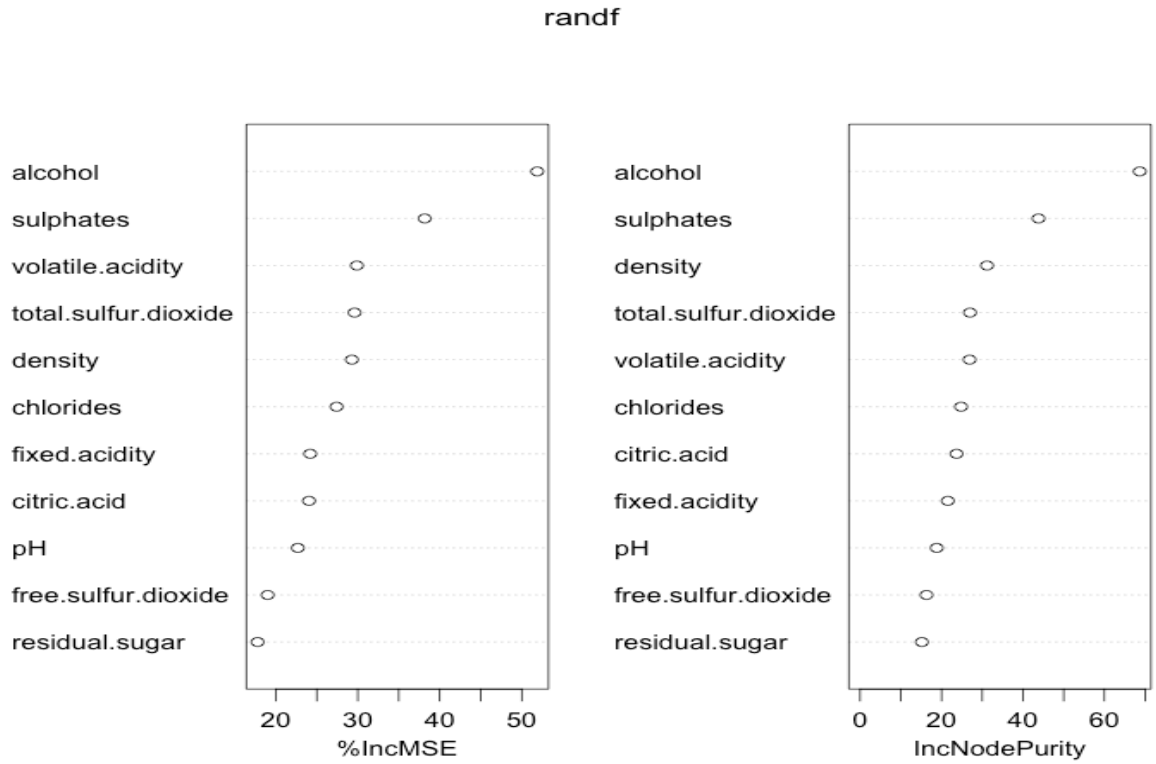
Results from the Forward stepwise regression

So after performing the analysis we got Alcohol, Sulphates, Volatile Acidity, pH, Citric acid and Total Sulfur Acid as the important features which can affect the quality of the red wine.

RANDOM FOREST

Random forests or Random Decision Forests are an ensemble learning method for classification, regression, and other tasks. Random Forests can be used for regression analysis and in fact called Regression Forests. They are an ensemble of different regression trees and are used for nonlinear multiple regression. Each leaf contains a distribution for the continues output variables. Random Forest package in R optionally produces two additional pieces of information: a measure of the importance of the predictor variables and a measure of the internal structure of the data. In this analysis, Variable importance of the Random Forest has been used for selecting important features of the data.

The Random Forest algorithm estimates the importance of a variable by looking at how much prediction error increases when data for that variable is permuted while all others are left unchanged. The necessary calculations are carried out tree by tree as the Random Forest is constructed. In the wine data, we used the variable importance function of the Random Forest and plot for the variable importance is given below



The importance of every variable can easily observe from the above figure. It is observed that alcohol is the most important feature for the red wine and residual sugar is the least important feature for the red wine. From the above plot, we decided the criterion that we should select the variables whose importance is more than 40%. Thus, four variables with higher importance (Alcohol, Sulphates, Volatile Acidity and Density) are selected.

In this way by using Multiple Linear Regression and Random Forest we have selected the features. Multiple Linear regression method helped us to extract six important features and by using Random Forest we extracted four important features. In next section, we will be building two models based on these features and will select one of them based on the accuracy.

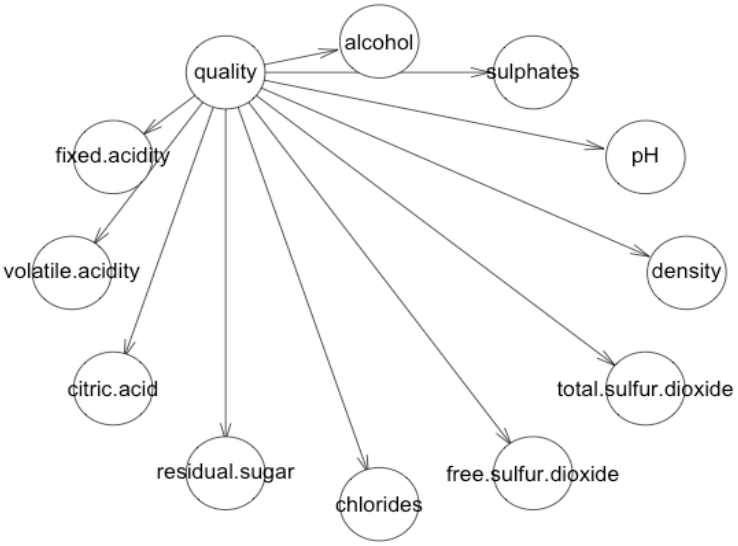
NAIVES BAYES

The Naive Bayes model is a special form of Bayesian network. This model is mainly used for classification problems. The important feature of Naive Bayes model is that, it has very strong independence assumptions
The naive.bayes functions creates the star-shaped Bayesian network form of a naive Bayes classifier; the training variable (the one holding the group each observation belongs to) is at the center of the star, and it has an outgoing arc for each explanatory variable.

MODEL1:

This was created by considering all the 11 predictor variables and the quality response variable. The accuracy obtained was 0.6108374

The data was divided into a 75:25 ratio of training and test data using random sampling

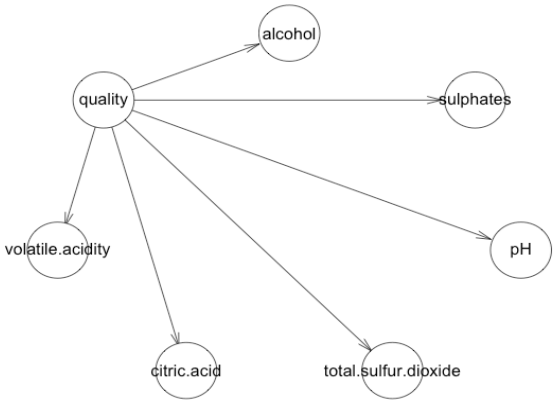


```
Confusion Matrix and Statistics

          Reference
Prediction  5  6  7
         5 81 25  2
         6 29 39  6
         7  1 10 10

Overall Statistics

               Accuracy : 0.6404
                 95% CI : (0.5702, 0.7064)
    No Information Rate : 0.5468
    P-Value [Acc > NIR] : 0.004296

                  Kappa : 0.3658
 Mcnemar's Test P-Value : 0.652690

Statistics by Class:

                     Class: 5 Class: 6 Class: 7
Sensitivity            0.7297   0.5270  0.55556
Specificity            0.7065   0.7287  0.94054
Pos Pred Value         0.7500   0.5270  0.47619
Neg Pred Value         0.6842   0.7287  0.95604
Prevalence             0.5468   0.3645  0.08867
Detection Rate         0.3990   0.1921  0.04926
Detection Prevalence   0.5320   0.3645  0.10345
Balanced Accuracy      0.7181   0.6279  0.74805
> 
```

MODEL 2:

This model was created by considering the feature selected 6 predictor variables and quality response variable. The accuracy obtained was 0.6403941.

For Naive Bayes , feature selected model gave a better accuracy than the model obtained using all predictor variables.




```
Confusion Matrix and Statistics

          Reference
Prediction  5  6  7
         5 77 26  1
         6 33 38  8
         7  1 10  9

Overall Statistics

               Accuracy : 0.6108
                 95% CI : (0.5401, 0.6783)
    No Information Rate : 0.5468
    P-Value [Acc > NIR] : 0.03848

                  Kappa : 0.3164
 Mcnemar's Test P-Value : 0.78850

Statistics by Class:

                     Class: 5 Class: 6 Class: 7
Sensitivity            0.6937   0.5135  0.50000
Specificity            0.7065   0.6822  0.94054
Pos Pred Value         0.7404   0.4810  0.45000
Neg Pred Value         0.6566   0.7097  0.95082
Prevalence             0.5468   0.3645  0.08867
Detection Rate         0.3793   0.1872  0.04433
Detection Prevalence   0.5123   0.3892  0.09852
Balanced Accuracy      0.7001   0.5978  0.72027
>
```

Tree augmented naive Bayes is a semi-naive Bayesian Learning method. It relaxes the naive Bayes attribute independence assumption by employing a tree structure, in which each attribute only depends on the class and one other attribute. A maximum weighted spanning tree that maximizes the likelihood of the training data is used to perform classification.

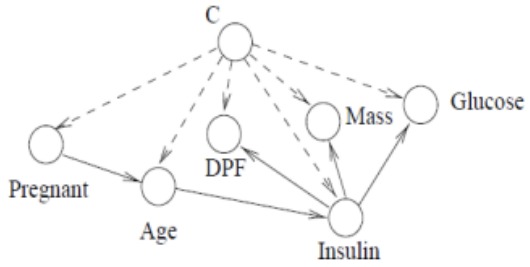
The Naive Bayes model as discussed previously, encodes incorrect independence assumptions that, given the class label, the attributes are independent of each other. But in the real world, the attributes of any system are mostly correlated and the case as in Naive Bayes rarely happens. In spite of such incorrect independent assumptions, the Naive Bayes model seems to perform fairly well. So, if the model also takes into account the correlations between the attributes, then the classification accuracy can be improved [5]. Bayesian networks capture all the correlations in the random variables in the form of a graph. Using such a Bayesian network, we can first calculate the conditional probabilities of all the random variables given its parents. Then using the independence statements encoded in the network, we can calculate the joint distribution using the local conditional probabilities[5]. But learning such a Bayesian network is very complex because there may be many random variables in a network, and each random variable may take many values. Also a single random variable can have many parents and finding the conditional distribution conditioned on all those parents increases the complexity. Above all, the main parameter that determines the classification in a Naive Bayes model, P(C | X1,...,Xn) takes into account all the variables. But a general Bayesian network may not have edges from the class node to all the variables. This might sometimes lead to lower accuracy in classification. Hence, we can conclude that, for better classification performance, we need a Bayesian network that encodes the structure of the Naive Bayes model and in addition to that, also captures the correlations between the variables in the system.

The solution to this issue can be an augmented Naive Bayesian network. An augmented Naive Bayesian network, maintains the structure of the Naive Bayesian network and augments it by adding edges between the variables in order to capture the correlations between the attributes. But this processes increases the computational complexity. "While the induction of the Naive Bayesian classifier requires only simple book keeping (storing conditional probabilities given the label), the induction of Bayesian networks requires searching the space of all possible networks, i.e., the space of all possible combination of edges.

In order to reduce the computational complexity and also take into account the correlations between the variables, restrictions need to be imposed on the level of interaction between the variables. One such model, is the Tree Augmented Naive Bayesian (TAN) model. This model imposes a restriction on the level of interaction between the variables to one. In a TAN model, all the variables are connected to the class variables by means of direct edges. Hence, it takes into account all the variable while determining P(C | X1,...,Xn). In addition to that, each variable can be connected to another variable in the network [5]. That is, each variable in the graph can have two parents viz., the class node and another variable node, except for one variable which is called root. The computational complexity of this model, is

greatly reduced, as each variable has a maximum of two parents. "Thus TAN maintains the robustness and computational complexity of the Naive Bayes model and at the same time displays better accuracy"

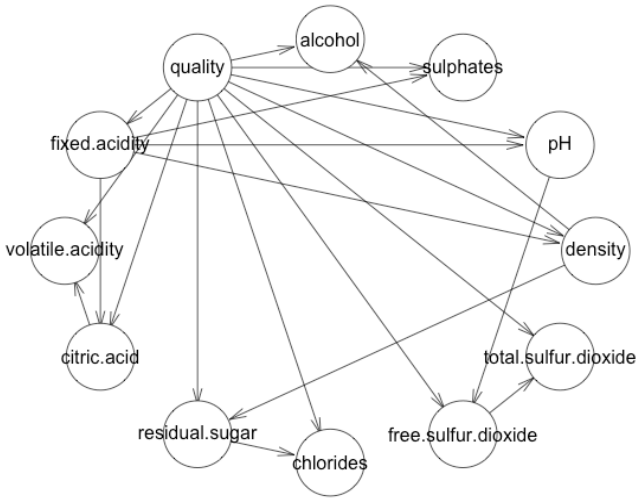
A Tree Augmented Naive Bayes model (TAN) imposes a tree structure on the Naive Bayes model, by restricting the interaction among the variables to a single level. A TAN model for a simple medical diagnostic system is shown in Figure.



MODEL 1:

This was created by considering all the 11 predictor variables and the quality response variable. The accuracy obtained was 0.635468

The data was divided into a 75:25 ratio of training and test data using random sampling

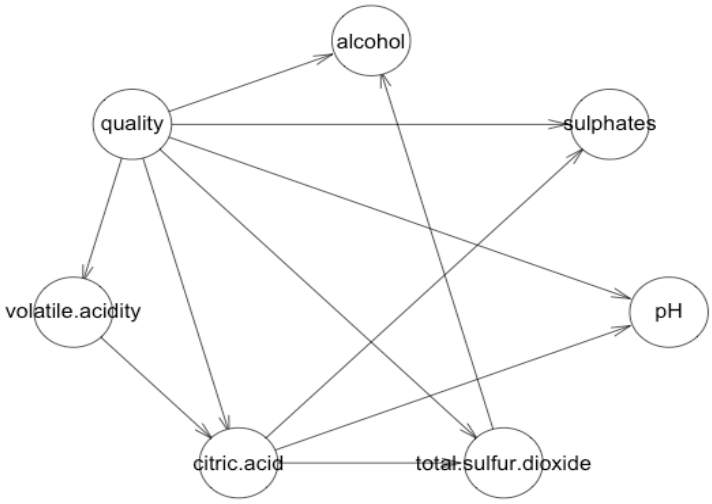


```
Confusion Matrix and Statistics

          Reference
Prediction  5  6  7
         5 78 21  3
         6 32 43  7
         7  1 10  8

Overall Statistics

               Accuracy : 0.6355
                 95% CI : (0.5652, 0.7017)
    No Information Rate : 0.5468
    P-Value [Acc > NIR] : 0.006494

                  Kappa : 0.3601
 Mcnemar's Test P-Value : 0.282444

Statistics by Class:

                     Class: 5 Class: 6 Class: 7
Sensitivity            0.7027   0.5811  0.44444
Specificity            0.7391   0.6977  0.94054
Pos Pred Value         0.7647   0.5244  0.42105
Neg Pred Value         0.6733   0.7438  0.94565
Prevalence             0.5468   0.3645  0.08867
Detection Rate         0.3842   0.2118  0.03941
Detection Prevalence   0.5025   0.4039  0.09360
Balanced Accuracy      0.7209   0.6394  0.69249
>
```

MODEL 2

This model was created by considering the feature selected 6 predictor variables and quality response variable. The accuracy obtained was 0.5369458

For Tree Augmented Naive Bayes, feature selected model gave worse accuracy than the model obtained using all predictor variables. The reason could be the overfitting on test data in case of feature selected variables.




```
Confusion Matrix and Statistics

          Reference
Prediction  5  6  7
         5 63 25  1
         6 36 35  6
         7 12 14 11

Overall Statistics

               Accuracy : 0.5369
                 95% CI : (0.4658, 0.607)
    No Information Rate : 0.5468
    P-Value [Acc > NIR] : 0.638436

                  Kappa : 0.2357
 Mcnemar's Test P-Value : 0.002307

Statistics by Class:

                     Class: 5 Class: 6 Class: 7
Sensitivity            0.5676   0.4730  0.61111
Specificity            0.7174   0.6744  0.85946
Pos Pred Value         0.7079   0.4545  0.29730
Neg Pred Value         0.5789   0.6905  0.95783
Prevalence             0.5468   0.3645  0.08867
Detection Rate         0.3103   0.1724  0.05419
Detection Prevalence   0.4384   0.3793  0.18227
Balanced Accuracy      0.6425   0.5737  0.73529
>
```

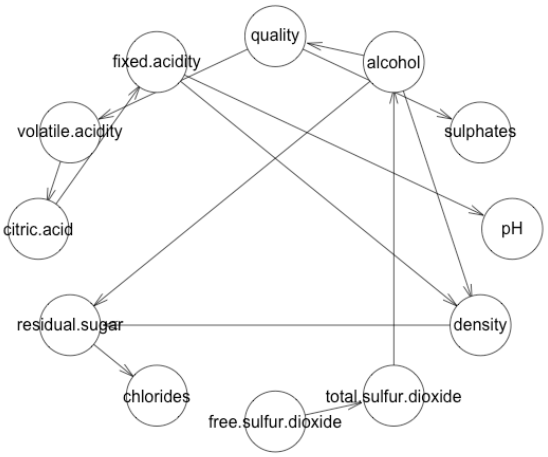
HILL CLIMBING

A greedy search on the space of the directed graphs. The optimized implementation uses score caching, score decomposability and score equivalence to reduce the number of duplicated tests.

MODEL 1

This was created by considering all the 11 predictor variables and the quality response variable. The accuracy obtained was 0.6453202

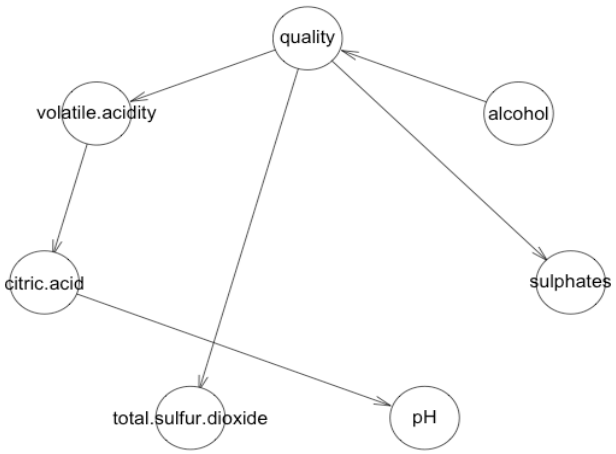
The data was divided into a 75:25 ratio of training and test data using random sampling




```
Confusion Matrix and Statistics

          Reference
Prediction  5  6  7
         5 82 25  2
         6 29 49 16
         7  0  0  0

Overall Statistics

               Accuracy : 0.6453
                 95% CI : (0.5753, 0.711)
    No Information Rate : 0.5468
    P-Value [Acc > NIR] : 0.0027873

                  Kappa : 0.3403
 Mcnemar's Test P-Value : 0.0003821

Statistics by Class:

                     Class: 5 Class: 6 Class: 7
Sensitivity            0.7387   0.6622  0.00000
Specificity            0.7065   0.6512  1.00000
Pos Pred Value         0.7523   0.5213      NaN
Neg Pred Value         0.6915   0.7706  0.91133
Prevalence             0.5468   0.3645  0.08867
Detection Rate         0.4039   0.2414  0.00000
Detection Prevalence   0.5369   0.4631  0.00000
Balanced Accuracy      0.7226   0.6567  0.50000
```

MODEL 2:

This model was created by considering the feature selected 6 predictor variables and quality response variable. The accuracy obtained was 0.6453202




```
Confusion Matrix and Statistics

          Reference
Prediction  5  6  7
         5 82 25  2
         6 29 49 16
         7  0  0  0

Overall Statistics

               Accuracy : 0.6453
                 95% CI : (0.5753, 0.711)
    No Information Rate : 0.5468
    P-Value [Acc > NIR] : 0.0027873

                  Kappa : 0.3403
 Mcnemar's Test P-Value : 0.0003821

Statistics by Class:

                     Class: 5 Class: 6 Class: 7
Sensitivity            0.7387   0.6622  0.00000
Specificity            0.7065   0.6512  1.00000
Pos Pred Value         0.7523   0.5213      NaN
Neg Pred Value         0.6915   0.7706  0.91133
Prevalence             0.5468   0.3645  0.08867
Detection Rate         0.4039   0.2414  0.00000
Detection Prevalence   0.5369   0.4631  0.00000
Balanced Accuracy      0.7226   0.6567  0.50000
```

COMPARISON OF ACCURACY OF THE 3 TYPES OF MODELS

	Model1 (11 variables)	Model2 (6 variables)	
Naïve Bayes	0.6108374	0.6403941	
Tree Augmented Naïve Bayes	0.635468	0.5369458	
Hill Climbing	0.6453202	0.6453202	

CONCLUSION

Naives bayes seems to be the best model classifying the data with a 64% accuracy.

REFERENCES

1. https://archive.ics.uci.edu/ml/datasets/Wine+Quality
2. http://www.bnlearn.com/bnrepository/
3. http://stackoverflow.com/questions/37896323/multinomial-naive-bayes-in-bnlearn-prediction-clarification
4. http://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1350&context=etd_projects
5. Bayesian Networks in R
6. Bayesian Data Analysis, Gelman
7. Allin Downey Github : https://github.com/AllenDowney
8. Bayesian network classifiers for the German credit data ,Scott A. Zonneveldt, Kevin B. Korb and Ann E. Nicholson

APPENDIX

Code for Reading and understanding Data and plotting the plots

```
library(corrplot)
library(lattice)
library(ggplot2)
library(caret)

mydata = read.csv("/Users/manvijain/Desktop/Bayesian Classification/red_1.csv",header =
TRUE, fill = TRUE)

summary(mydata)
sapply(mydata,class)
par(mfrow=c(2,2))
colnames(mydata) <-
c("fixed.acidity","volatile.acidity","citric.acid","residual.sugar","chlorides","free.sulfur.dioxid
e","total.sulfur.dioxide","density","pH","sulphates","alcohol","quality")


hist(mydata$fixed.acidity,col="light blue",xlab="fixed.acidity",main=NA)
hist(mydata$volatile.acidity,col="light blue",xlab="Volatile Acidity",main=NA)
hist(mydata$citric.acid,col="light blue",xlab="Citric Acid",main=NA)
hist(mydata$residual.sugar ,col="light blue",xlab="Residual Suger",main=NA)

par(mfrow=c(2,2))

hist(mydata$chlorides,col="light blue",xlab="Chlorides",main=NA)
hist(mydata$free.sulfur.dioxide,col="light blue",xlab="Free Sulfur Dioxide",main=NA)
hist(mydata$total.sulfur.dioxide,col="light blue",xlab="Total Sulfur Dioxide",main=NA)
hist(mydata$density,col="light blue",xlab="Density",main=NA)

par(mfrow=c(2,2))

hist(mydata$pH,col="light blue",xlab="pH",main=NA)
hist(mydata$sulphates ,col="light blue",xlab="Sulphates",main=NA)
hist(mydata$alcohol,col="light blue",xlab="Alcohol",main=NA)
hist(mydata$quality,col="light blue",xlab="Quality",main=NA)

par(mfrow=c(1,6))

boxplot(mydata$fixed.acidity,col="light blue",xlab="Fixed Acidity",main=NA)
boxplot(mydata$volatile.acidity,col="light blue",xlab="Volatile Acidity",main=NA)
boxplot(mydata$citric.acid,col="light blue",xlab="Citric Acid",main=NA)
boxplot(mydata$residual.sugar ,col="light blue",xlab="Residual Suger",main=NA)
```

```
boxplot(mydata$chlorides,col="light blue",xlab="Chlorides",main=NA)
boxplot(mydata$free.sulfur.dioxide,col="light blue",xlab="Free Sulfur Dioxide",main=NA)

par(mfrow=c(1,6))

boxplot(mydata$total.sulfur.dioxide,col="light blue",xlab="Total Sulfur Dixoide",main=NA)
boxplot(mydata$density,col="light blue",xlab="Density",main=NA)
boxplot(mydata$pH,col="light blue",xlab="pH",main=NA)
boxplot(mydata$sulphates ,col="light blue",xlab="Sulphates",main=NA)
boxplot(mydata$alcohol,col="light blue",xlab="Alcohol",main=NA)
boxplot(mydata$quality,col="light blue",xlab="Quality",main=NA)

pairs(~fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+
total.sulfur.dioxide+density+pH+sulphates+alcohol+quality,data=mydata,main="Scatter
Plot")

k1 = cor(mydata,method="pearson")
corrplot(k1,method="number")

k2 = cor(mydata,method="spearman")
corrplot(k2,method="number")

pca = princomp(mydata,cor=T)
summary(pca)
pca$loadings
load <- with(pca,unclass(loadings))
load
screeplot(pca,type="line")
```

Outlier detection and Preprocessing

```
library(corrplot)

mydata = read.csv("/Users/manvijain/Desktop/Bayesian Classification/red_1.csv",1)

data_iqr <- mydata
colnames(data_iqr) <-
c("fixed.acidity","volatile.acidity","citric.acid","residual.sugar","chlorides","free.sulfur.dioxid
e","total.sulfur.dioxide","density","pH","sulphates","alcohol","quality")

vars <-
c("fixed.acidity","volatile.acidity","citric.acid","residual.sugar","chlorides","free.sulfur.dioxid
e","total.sulfur.dioxide","density","pH","sulphates","alcohol","quality")

outliers <- c()

for(i in vars)
```

```
{
        max <- quantile(data_iqr[,i],0.75,na.rm=TRUE) + (IQR(data_iqr[,i],na.rm=TRUE)*1.5)
        min <- quantile(data_iqr[,i],0.75,na.rm=TRUE) - (IQR(data_iqr[,i],na.rm=TRUE)*1.5)

        idx <- which(data_iqr[,i] < min | data_iqr[,i] > max)

        print(paste(i,length(idx),sep=''))

        outliers <- c(outliers,idx)
}

outliers <- sort(outliers)

dsbase <- data_iqr[-outliers,]


#write.csv(dsbase,"/Users/manvijain/Desktop/Bayesian
Classification/outlier_free.csv",row.names = FALSE)

#testing outlier data for normal distribution

data_no_outlier = read.csv("/Users/manvijain/Desktop/wine data/outlier_free.csv")
colnames(data_no_outlier) <-
c("fixed.acidity","volatile.acidity","citric.acid","residual.sugar","chlorides","free.sulfur.dioxid
e","total.sulfur.dioxide","density","pH","sulphates","alcohol","quality")


hist(data_no_outlier$fixed.acidity,col="light blue",xlab="fixed.acidity",main=NA)
hist(data_no_outlier$volatile.acidity,col="light blue",xlab="Volatile Acidity",main=NA)
hist(data_no_outlier$citric.acid,col="light blue",xlab="Citric Acid",main=NA)
hist(data_no_outlier$residual.sugar ,col="light blue",xlab="Residual Suger",main=NA)

par(mfrow=c(2,2))

hist(data_no_outlier$chlorides,col="light blue",xlab="Chlorides",main=NA)
hist(data_no_outlier$free.sulfur.dioxide,col="light blue",xlab="Free Sulfur
Dioxide",main=NA)
hist(data_no_outlier$total.sulfur.dioxide,col="light blue",xlab="Total Sulfur
Dioxide",main=NA)
hist(data_no_outlier$density,col="light blue",xlab="Density",main=NA)

par(mfrow=c(2,2))

hist(data_no_outlier$pH,col="light blue",xlab="pH",main=NA)
hist(data_no_outlier$sulphates ,col="light blue",xlab="Sulphates",main=NA)
hist(data_no_outlier$alcohol,col="light blue",xlab="Alcohol",main=NA)
hist(data_no_outlier$quality,col="light blue",xlab="Quality",main=NA)
```

```
view(data_no_outlier)
trans <- preProcess(data_no_outlier, method = c("center","scale"))
transformed <- predict(trans, data_no_outlier)
par(mfrow=c(1,2))
hist(data_no_outlier$citric.acid, main="Original",xlab="dist")
hist(transformed$citric.acid    , main="Centered and Scaled",xlab="dist")
```

FEATURE SELECTION

```
library(corrplot)
library(randomForest)
library(ggplot2)
library(lattice)


wine_data <- read.csv("/Users/manvijain/Desktop/wine data/outlier_free.csv")
colnames(wine_data) <-
c("fixed.acidity","volatile.acidity","citric.acid","residual.sugar","chlorides","free.sulfur.dioxid
e","total.sulfur.dioxide","density","pH","sulphates","alcohol","quality")


#stepwise regression

step_reserveModel <- step(lm(quality ~1,
wine_data),scope=list(lower=~1,upper=~fixed.acidity + volatile.acidity + citric.acid +
residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates +
alcohol),direction="forward")
summary(step_reserveModel)
step_reserveModel_bck <-
step(lm(quality~fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur
.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol+quality,wine_data), direction =
"backward")
summary(step_reserveModel_bck)


#randomForest
randf <- randomForest(quality~. ,wine_data,importance =TRUE)
importance(randf)

par(mfrow = c(1,1))
varImpPlot(randf,sort =TRUE)
```

BN LEARN

```
install.packages(bnlearn)
```

```
install.packages(forecast)

library(bnlearn)
library(forecast)
library(xlsx)
library(caret)
#library(CORElearn)

wine_data <- read.csv("/Users/manvijain/Desktop/wine data/outlier_free.csv", header =
TRUE)
colnames(wine_data) <-
c("fixed.acidity","volatile.acidity","citric.acid","residual.sugar","chlorides","free.sulfur.dioxid
e","total.sulfur.dioxide","density","pH","sulphates","alcohol","quality")
head(wine_data)


#changing category of data
wine_data[c(6,7,12)] <- lapply(wine_data[c(6,7,12)],as.double)
summary(wine_data)

#Preprocess data : discretize data to make it more suitable for Structure and parameter
learning
bucket_data <- discretize(wine_data[,-12],method = "interval",breaks =3,ordered = FALSE
,debug = FALSE)
bucket_data$quality <-as.factor(wine_data[,12])
summary(bucket_data)

#Dividing data in training and test set
smp_size <- floor(0.75 * nrow(wine_data))
set.seed(123)
train_ind <- sample(seq_len(nrow(wine_data)),size = smp_size)
wine_train <- bucket_data[train_ind,]
wine_test<- bucket_data[-train_ind,]

head(wine_train)
summary(wine_train)
lapply(wine_train, class)

##----MODEL1---##

#HILL CLIMBING

wine_res = hc(wine_train)
plot(wine_res)
fitted = bn.fit(wine_res, wine_train)     # learning of parameters
pred = predict(fitted, "quality" ,wine_test)
cbind(pred, wine_test[, "quality"])          # compare the actual and predicted
```

```
mean(pred == wine_test$quality)
confusionMatrix(pred,wine_test$quality,positive = NULL,prevalence = NULL,)

#Naives Bayes

NB.net<- naive.bayes(wine_train, "quality")
plot(NB.net)
NB.fit <- bn.fit(NB.net,wine_train)
NB.pred = predict(NB.fit, wine_test)
summary(NB.pred)
#writeLines("\n Multinomial Naive Bayes \n")
mean(NB.pred == wine_test$quality)
confusionMatrix(NB.pred,wine_test$quality,positive = NULL,prevalence = NULL,)

#Tree Augmented Bayesian Network

TB.net <- tree.bayes(wine_train,"quality",whitelist = NULL, blacklist = NULL,
                     mi = NULL, root = NULL, debug = FALSE)
plot(TB.net)
TB.fit <- bn.fit(TB.net,wine_train)
TB.pred = predict(TB.fit,wine_test)
summary(TB.pred)
mean(TB.pred == wine_test$quality)
confusionMatrix(TB.pred,wine_test$quality,positive = NULL,prevalence = NULL,)



###----MODEL 2----##(trying with the feature selected variables)
wine_train2 <- wine_train[,c(2,3,7,9,10,11,12)]        # Model2
wine_test2 <- wine_test[,c(2,3,7,9,10,11,12)]

#HILL CLIMBING

wine_res2 = hc(wine_train2)
plot(wine_res2)
fitted2 = bn.fit(wine_res2, wine_train2)     # learning of parameters
pred2 = predict(fitted2, "quality" ,wine_test2)
cbind(pred2, wine_test2[, "quality"])          # compare the actual and predicted
mean(pred2 == wine_test2$quality)
confusionMatrix(pred2,wine_test2$quality,positive = NULL,prevalence = NULL,)


#Naives Bayes

NB.net2<- naive.bayes(wine_train2, "quality")
plot(NB.net2)
NB.fit2 <- bn.fit(NB.net2,wine_train2)
```

```
NB.pred2 = predict(NB.fit2, wine_test2)
summary(NB.pred2)
#writeLines("\n Multinomial Naive Bayes \n")
mean(NB.pred == wine_test2$quality)
confusionMatrix(NB.pred2,wine_test2$quality,positive = NULL,prevalence = NULL,)

#Tree Augmented Bayesian Network

TB.net2 <- tree.bayes(wine_train2,"quality",whitelist = NULL, blacklist = NULL,
                      mi = NULL, root = NULL, debug = FALSE)
plot(TB.net2)
TB.fit2 <- bn.fit(TB.net2,wine_train2)
TB.pred2 = predict(TB.fit2,wine_test2)
summary(TB.pred2)
mean(TB.pred2 == wine_test2$quality)
confusionMatrix(TB.pred2,wine_test2$quality,positive = NULL,prevalence = NULL,)
```