**Data Warehousing and Data Mining Project Report**

RED WINE QUALITY

**Abstract:**

Wine classification is a difficult task since taste is the least understood of the human senses. A good wine quality prediction can be very useful in the certification phase, since currently the sensory analysis is performed by human tasters, being clearly a subjective approach .An automatic predictive system can be integrated into a decision support system, helping the speed and quality of the oenologist performance. Furthermore, a feature selection process can help to analyze the impact of the analytical tests. If it is concluded that several input variables are highly relevant to predict the wine quality, since in the production process some variables can be controlled, this information can be used to improve the wine quality.It is not the only intriguing topic in wine industry. If we can build its predictive model, we can research and develop new products efficiently. In other words, it allows to reduce cost and develop time and even improve performance.

**Data Set:**

This dataset is related to red variants of the Portuguese "Vinho Verde" wine. There are 11 attributes which make good quality wine and the 12th attribute being the quality of the wine.

1. Fixed acidity: most acids involved with wine or fixed or non-volatile (do not evaporate readily)
2. Volatile acidity: the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste
3. Citric acid: found in small quantities, citric acid can add 'freshness' and flavor to wines
4. Residual sugar: the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet
5. Chlorides: the amount of salt in the wine
6. Free sulfur dioxide: the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of the wine
7. Total sulfur dioxide: the amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine
8. Density: the density of water is close to that of water depending on the percent alcohol and sugar content
9. pH: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale
10. Sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels, which acts as an antimicrobial and antioxidant
11. Alcohol: the percent alcohol content of the wine
12. Quality: output variable (based on sensory data, a score between 0 and 10)

**Implementation:**

The wine classification is done using 4 primary methods

1. Linear Regression
2. Decision Tree
3. Random Forest Classifier
4. Support Vector Classifier
5. **Linear Regression**

Linear regression is the problem of fitting a linear function to a set of input-output pairs given a set of training examples, in which the input and output features are numeric.

Suppose the input features are *X1,...,Xn*. A linear function of these features is a function of the form

*fw(X1,...,Xn) = w0+w1 ×X1 + ...+ wn ×Xn ,*where *w=⟨w0,w1,...,wn⟩* is a tuple of weights. To make *w0* not be a special case, we invent a new feature, *X0*, whose value is always 1.

We will learn a function for each target feature independently, so we consider only one target, Y. Suppose a set E of examples exists, where each example e∈E has values val(e,Xi) for feature Xi and has an observed value val(e,Y). The predicted value is thus

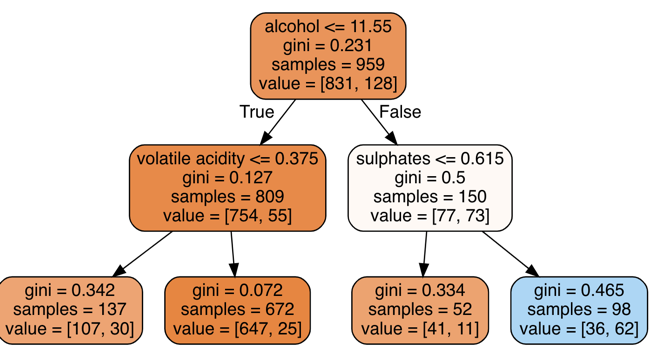
pvalw(e,Y) = w0+w1×val(e,X1) + ...+ wn×val(e,Xn)

= ∑i=0n wi×val(e,Xi) ,

where we have made it explicit that the prediction depends on the weights, and where *val(e,X0)* is defined to be 1.

1. **Decision Tree**

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.



1. **Random Forest Classifier**

A random forest is an ensemble of decision trees. In a random forest, each decision is trained on a random subset of the training set, usually sampled with replacement, with sample size equal to training set size. In the case of classifications, the class with the highest mean probability across all trees is selected. In the case of regressions, the predicted y value is the mean of predicted y values across all trees. In Scikit-learn, Random Forest Classifier is used for random forest classification.

We then use scikit-learn's Random Forest Classifier to train random forest classifiers. Random Forest Classifier has all the parameters from Decision Tree Classifier. Random Forest Classifier also has its unique parameters such as n\_estimators and n\_jobs. The n\_estimators is the number of trees in the forest, and n\_jobs is the number of jobs to run in parallel.

1. **Support Vector Classification**

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

In two dimensional space, this hyperplane is a line dividing a plane into two parts wherein each class lay in either side. The goal is to design a hyperplane that classifies all the training dataset in two classes.

The best choice will be the hyperplane that leaves maximum margin from both the classes.

**Result:**

1. **Linear Regression:**

MAE: 0.5035304415524379

MSE: 0.3900251439639539

RMSE: 0.6245199307980122

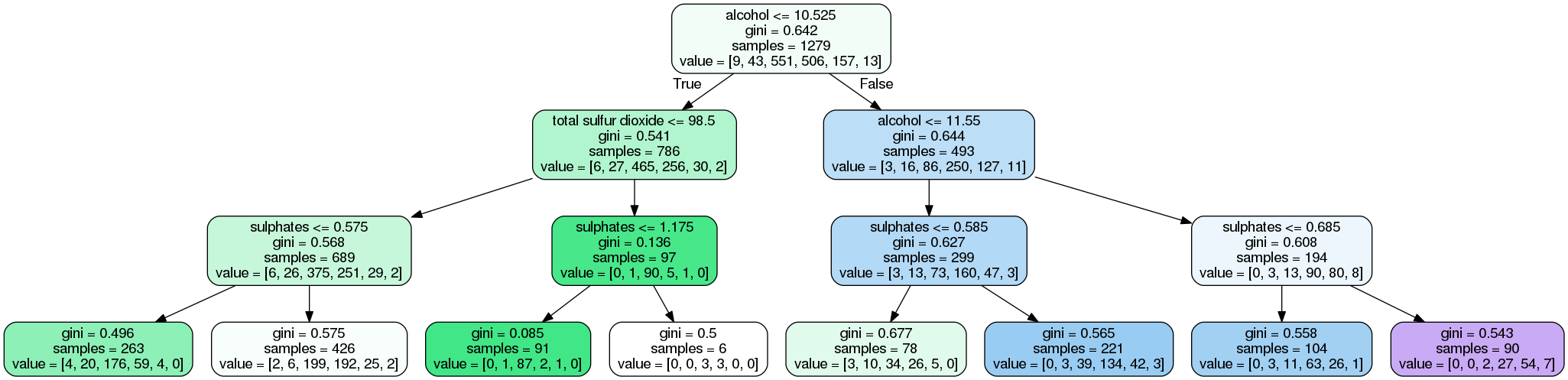
1. **Decision Tree:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | recall | f1-score | support |
| 3 | 0.00 | 0.00 | 0.00 | 1 |
| 4 | 0.00 | 0.00 | 0.00 | 10 |
| 5 | 0.54 | 0.91 | 0.68 | 130 |
| 6 | 0.53 | 0.31 | 0.39 | 132 |
| 7 | 0.42 | 0.24 | 0.30 | 42 |
| 8 | 0.00 | 0.00 | 0.00 | 5 |

micro avg 0.53 0.53 0.53 320

macro avg 0.25 0.24 0.23 320

weighted avg 0.49 0.53 0.48 320



1. **Random Forest Classifier:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | recall | f1-score | support |
| 3 | 0.00 | 0.00 | 0.00 | 1 |
| 4 | 0.00 | 0.00 | 0.00 | 10 |
| 5 | 0.71 | 0.77 | 0.74 | 130 |
| 6 | 0.64 | 0.70 | 0.67 | 132 |
| 7 | 0.62 | 0.48 | 0.54 | 42 |
| 8 | 0.00 | 0.00 | 0.00 | 5 |

micro avg 0.67 0.67 0.67 320

macro avg 0.33 0.32 0.33 320

weighted avg 0.63 0.67 0.65 320

1. **Support Vector Classification:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | recall | f1-score | support |
| 3 | 0.00 | 0.00 | 0.00 | 1 |
| 4 | 0.00 | 0.00 | 0.00 | 10 |
| 5 | 0.57 | 0.70 | 0.63 | 130 |
| 6 | 0.50 | 0.55 | 0.53 | 132 |
| 7 | 0.50 | 0.17 | 0.25 | 42 |
| 8 | 0.00 | 0.00 | 0.00 | 5 |

micro avg 0.53 0.53 0.53 320

macro avg 0.26 0.24 0.23 320

weighted avg 0.50 0.53 0.50 320

**Conclusion:**

**After implementing all, it is safe to say that Random Forest Classifier is best to use for this dataset among all the above methods used.**