THE IMPACTS OF PHYSICOCHEMICALS ON THE QUALITY OF RED WINE

Sevil Coskun, s250910

POLITECNICO DI TORINO
DEPARTMENT OF CONTROL AND COMPUTER ENGINEERING
MASTER OF SCIENCE IN COMPUTER ENGINEERING
Academic Year 2018-2019

DATA SPACES 01RLPOV
Prof. Francesco Vaccarino
Assoc. Prof. Roberto Fontana
Torino, January 2019

TABLE OF CONTENTS

1. Introduction

Wine is one of the most popular drink in the world, mostly people want to select best wine but they do not have very deep knowledge about wines. Therefore, deciding criteria mostly tends with price and brands. It is time to change this point of view and go deep down about which chemicals have more effects to increase wine quality.

In this analysis of the work, it will be determined which physiochemical properties make red wine 'good!' by using some machine learning techniques. When a person wants to take a qualify wine, he can easily find that just looking the chemicals inside of the wine thanks to this analysis.
In this dataset, there are specifically red wine variants of Poteguese "Vinho Verde" wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.). For more information, read [Cortez et al., 2009].

For this thesis, it is aimed that the analyzing which approach is good for predicting wine quality better with using the same classification algorithms by using binary and ordinal dataset. Content of the paper is starting with introduction of the data set, and some data visualization. Then some classification and one regression technique will be applied to the data and with cross validation accuracy will be evaluated.

2. Tool Information

Anaconda is a free and open source distribution of Python and R for scientific computing (data science, machine learning applications, large - scale data processing, predictive analytics, etc.) aimed at simplifying package management and deployment[1]. The combination of data analyst point of view and computer engineering perspective Python is the best language for working on it. Since, Python has many functions and libraries for Machine Learning technique. Additionally, diversity of the libraries is not only bounded by mathematical view, but also it has many visualizing libraries too. During the wine quality analysis, mainly used libraries are listed in below:
***Pandas**:* providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive[2].
N*umpy:* is an extension of support for the manipulation of arrays and multidimensional matrices that has provided me with high-level mathematical functions to work with.[3]
***Matplotlib**:* Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms[4].
Seaborn: Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics[5].
Sklearn: is a library that provides efficient versions of a large number of the most common algorithms for Machine Learning both supervised and unsupervised[6]. For each used algorithm sklearn has sublibrary such as LogisticRegression, or SVC(supper Vector Machine functions)

3. Explanation of Data

3.1 Information about Row Data

```
['winequality-red.csv']
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
fixed_acidity           1599 non-null float64
volatile_acidity        1599 non-null float64
citric_acid             1599 non-null float64
residual_sugar          1599 non-null float64
chlorides               1599 non-null float64
free_sulfur_dioxide     1599 non-null float64
total_sulfur_dioxide    1599 non-null float64
density                 1599 non-null float64
pH                      1599 non-null float64
sulphates               1599 non-null float64
alcohol                 1599 non-null float64
quality                 1599 non-null int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

In the data set, there are 1599 different wine as row data and 12 features as columns.
Each feature values are numeric it means the data is convenient for regression and classification.
Furthermore, there is no nun value to deal with it and all values are numeric means that input values are float and only output value is integer.

1 - fixed acidity: most acids involved with wine or fixed or nonvolatile
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: the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste
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; it prevents microbial growth and the oxidation of wine
7 - total sulfur dioxide: amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, SO2 becomes evident in the nose and taste of wine
8 - density: the density of water is close to 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 levels, acts as an antimicrobial and antioxidant
11 - alcohol: the percent alcohol content of the wine
12 - quality: the output of the dataset in range between 0 – 10

For distribution of the dataset was plotted, to analyze in more detail, in appendix Figure 6-1 and Figure 6-2. As it seems from the plots, data is not linearly separable, or there is no good distribution therefore, it is not select a model before analyzing the data. Therefore, different models will be used to analyze dataset better.

3.2 Quality of the Dataset

This data set has many different features and it is important to understand relationship between these in order to analyze dataset better. For that reason, correlation map helps to understand these relations in a single representation. Correlation map is made by calculating the covariance of each feature with respect to others, then each covariance value is divided by standard deviation of each variables and get results between -1, 0, 1.

$$R_{(x,y)} = \frac{COV(x,y)}{S_x S_y} \qquad COV(x,y) = \sum_{i=1}^{n} \frac{(x_i - \mu_x)(y_i - \mu_y)}{n-1}$$

-1 means: There is a negative relationship between dependent and independent variables.
0 means: There is no relationship between dependent and independent variables.
1 means: There is a positive relationship between dependent and independent variables.

According to these information, it can be made a good analyze about dataset and columns.

Just some example;
* Quality has
(+)relationship between alcohol
(-)relationship between volitile_acidicity.
()no relationship between residual_sugar, free_sulfur_dioxide, and pH.(corr =~ 0)

* Alcohol has
(+)relationship between quality and pH
(-)relationship between density
()no relationship between fixed_acidicity, residual_sugar, free_sulfur_dioxide, sulphates

* Volitile_acidicity has
(+)relationship between pH.
(-)relationship between citric_acid, fixed_acidicity and sulphates
()no relationship between residual_sugar, chlorides, free_sulfur_dioxide, total_sulfur_dioxide, density

It seems very hard to analyze dataset like that; therefore, it is better to go deep down analyzing with some other visualizations.

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000

When the wine dataset is analyzed quality of the data is the most important step before applying machine learning algorithms. Hence data distribution, balanced or unbalanced data shapes differently the algorithm and also could cause evaluate with some errors. For that reason, data should be analyzed for statistical view. For each feature of the data, summary of statistical measures is calculated and figured in the below.

Count: number of records for each attribute that corresponds to the number of wines

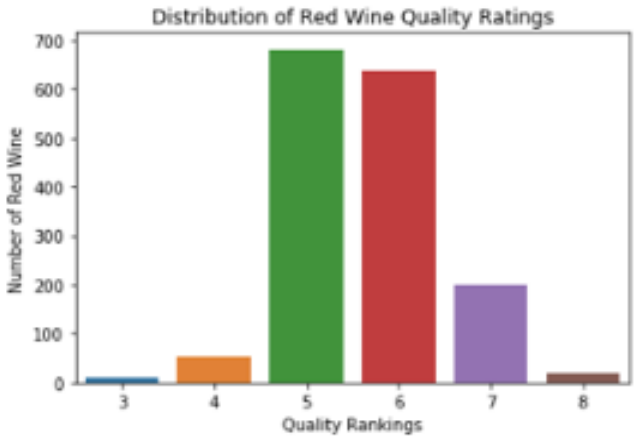
Mean: the average value for each attribute with related to total number of wines

Std: Standard deviation, is a measure that is used to quantify the amount of variation or dispersion of a set of data values

Min and Max: Lowest and Highest value in the attribute

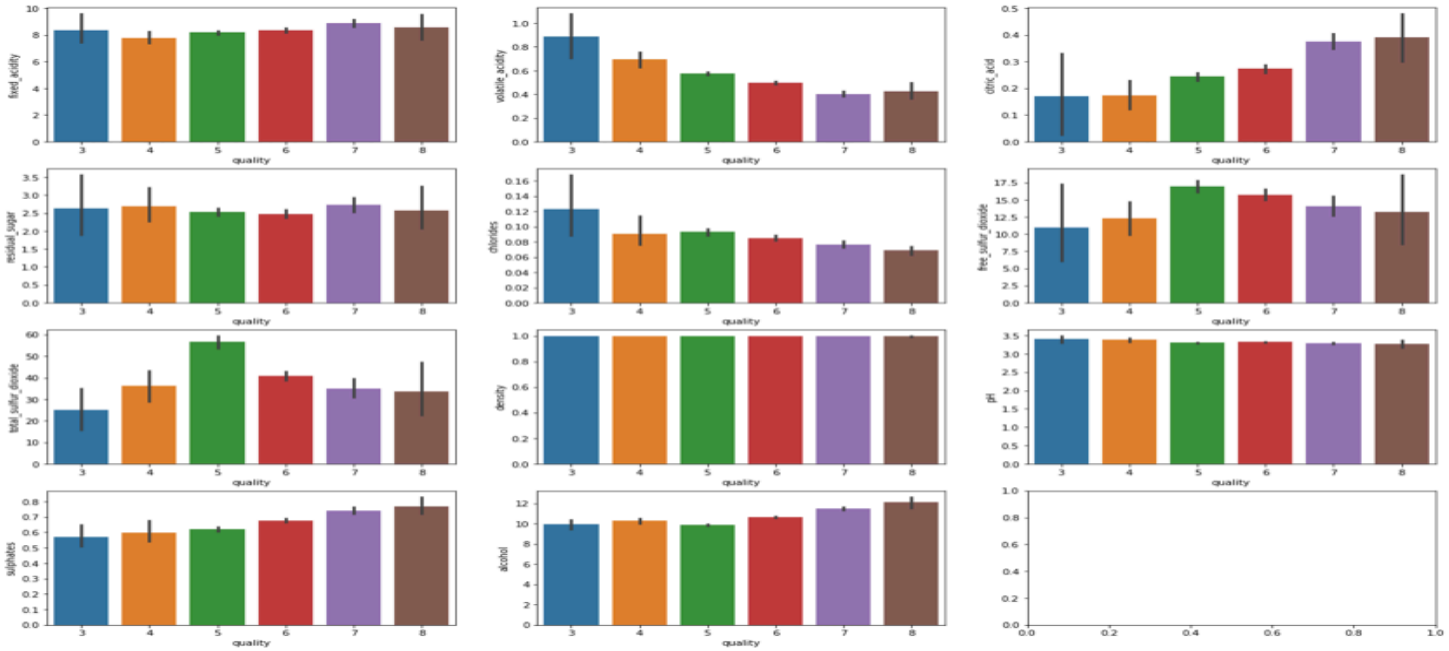
Percentiles: Based on the percentage output of the attributes

Additionally, as it seems from the output values the maximum value is 8 and minimum value is 3. Therefore, it is better to see visualized distribution of the data on quality(output) attribute.

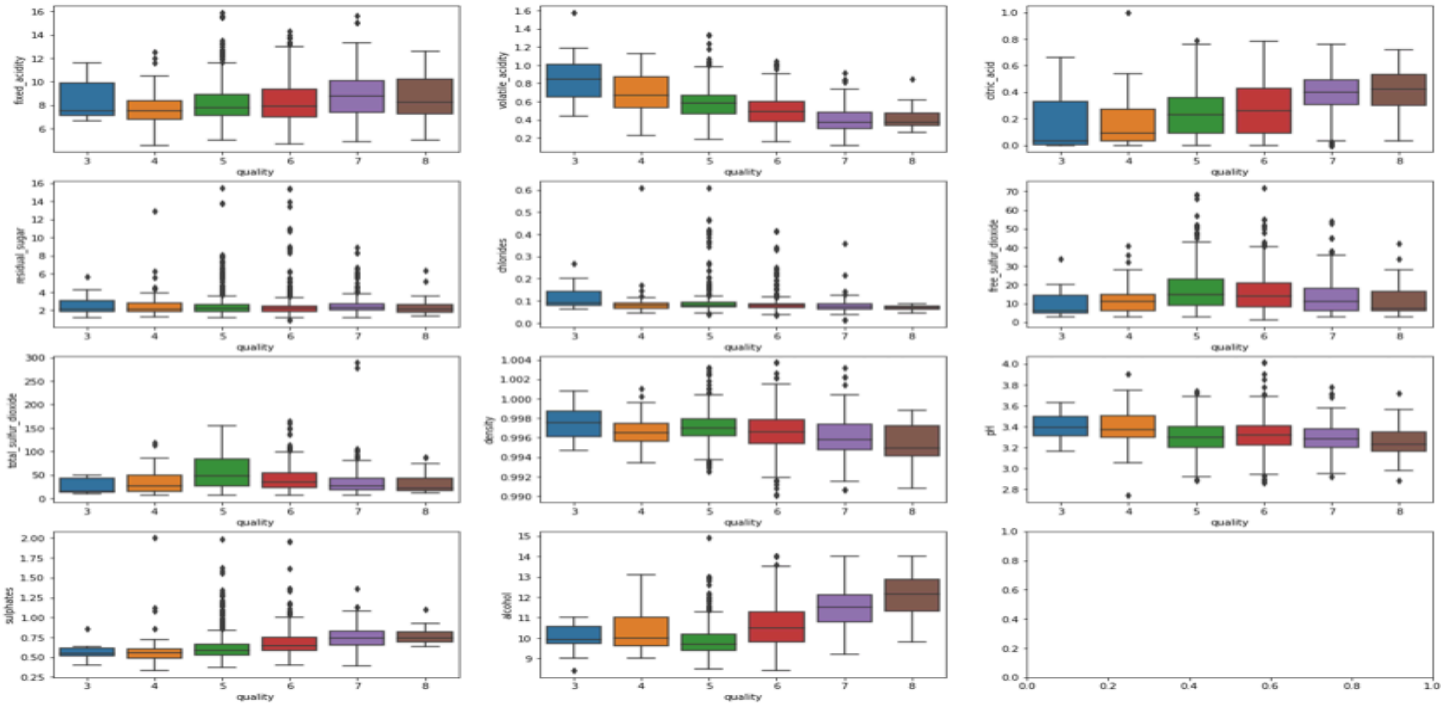


As it seems from the figure in left part, the data is unbalanced and row data is range between 3 and 8. Working with the unbalanced data could cause wrong estimations because the classification algorithms work with training and test data. If the proportion of the data is not almost equal, the loss of the estimation goes high and model couldn't predict the model well.

As it seems from the figure in left part, the data is unbalanced and row data is range between 3 and 8. Working with the unbalanced data could cause wrong estimations because classification algorithms work with training and test data. If the proportion of the data is not almost equal, the loss of the estimation goes high.



A boxplot is a graph that gives a good indication of how the values in the data are spread out. Although box lots may seem primitive in comparison to a histogram or density plot, they have the advantage of taking up less space, which is useful when comparing distributions between many groups or datasets. Box plotting gives a nice summary of one or several numeric values with respect to statistical views. X axis represents the output value and y axis is each feature. The line divides the box into two parts represents median of the data. The end of the box shows the upper and low quartiles (25% - 75%), the extreme lines shows highest and lowest value excluding outlier on data.



3.3 Data Preparation

This dataset can be proper for classification and regression techniques because the data is multivariate, contains numeric values as an input and output value is ordered and not balanced means that there are much more normal wines than excellent or poor ones. As it seems from the figure in above, data is not balanced and the ranges are between 3-8 score. For this reason, it is more convenient to start with making data balanced. There are some techniques to make balanced to data such as under or over sampling.

When it is tried to apply under sampling there is a problem about the 3 and 8 quality class have many few data it is not good for training due to few data. Also for oversampling, the difference between quality classes is very high and also it is not good for training the data with always repeating numbers.

It is also possible solution to break unbalancing, turns dataset into binary (categorical) dataset as good and bad. Categorical data set is more better to apply some machine learning algorithms, therefore data set quality output is modified by ‘good’ (means quality score is higher than 5) and ‘bad’ (means quality score is lower than 5) wines. After this modification of the data, it seems with the value dimension as balanced because the count of those two classes are very equal with each other. Good wine dataset equals to 855 items, and bad wine dataset equals to 744 items.

Additionally, PCA (Principle Component Analysis) may be used for decreasing the dimensionality but the features are not related(correlated) with each therefore PCA technique will not be successful. For that reason, PCA will not be applied for preparing the dataset. It will be seen in the end of analysis, after applying decision tree and random forest algorithms by selecting max depth parameters, feature election will be done itself.

4. Analysis of Classifications

Before analyzing the modified data, dataset should be normalized because when the data distributions are seemed there are some number variations are between 0-280000 or just 3-8. Therefore, before applying mathematical algorithms dataset should be normalized that given each value, puts to the same value intervals means between 0-1:

$$x_{normalized} = \frac{x - mean}{std}$$

For validation of the model, **K-fold Cross-Validation** technique will be used with 10 split folds, k-folds use for selecting the best model and to give an idea of the test error of final chosen model. The idea of the behind k-folds; randomly divide data into k equal size parts and leave out part k, fit the model to the other k-1 parts and obtain predictions for the left-out k-th part and lastly combines each results of k folds.

$$CV_{(k)} = \sum_{k=1}^{K} \frac{n_k}{n} MSE_k \qquad MSE_k = \sum_{i \in C} \frac{(y - y_i)^2}{n_k}$$

Also, traditional **train-test split** technique will be used and then calculating the accuracy. According to the purpose of the research, classification algorithms will be trained with ordinal dataset then calculating accuracies with the test data set then the same algorithms will be trained with binary(modified) dataset then calculating accuracies with the test dataset. To conclude, comparisons and result analysis will be done. To decide how the selected model predicts about the test data accuracy of the model will be calculated as summation of the correct prediction results then divide that sum into total number of test data;

$$\sum \frac{(y_{test} == y_{prediction})}{len(y_{test})}$$

The analysis of this paper is divided into 2 main part; First some classification algorithms will be trained and tested in **ordered dataset**. Secondly, the same algorithms will be used for training and test in **modified binary dataset**, Selected classification algorithms are in the below;

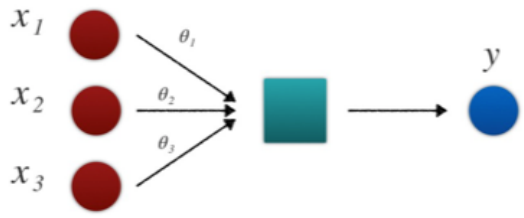
- Logistic Regression
- Support Vector Machines
- Decision Tree
- Random Forest

4.1 Classification Algorithms

4.1.1 Logistic Regression

Logistic regression is a predictive analysis of statistical method for supervised learning. It classifies data to look how related features with each other like in the linear regression but different from the linear regression, it does not predict the real value of the function as an output. Logistic regression has some different interpretation according to dataset, but for wine quality data set it is better to use categorical logistic regression because it deals with multiple categories is one by one meaning that each category has its own column. Each feature will be evaluated as binary feature for safe in the algorithm.

Logistic regression takes real values (as input) and makes predictions as to be probability of the input belonging to default class. After calculating each input values, labeled as binary class which is below or upper from the mean. It performs as numerical classification with calculating weights, and so the label outputs are belonging to the classes means that outcome in logistic regression is a probability between 0 and 1. Let P(y=1|x) be the probability that the one class output y is 1 given the input feature vector x. The coefficients b0, b1 are the weights that the algorithm is trying to learn. After each calculation learning rate is used to understand how fast/slow coefficients are changed. Functions have parameters/weights and we want to find the best values for them.



$$P(y = 0 \mid x) = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

$$P(y = 1 \mid x) = \frac{e^{-(b_0 + b_1 x)}}{1 + e^{-(b_0 + b_1 x)}}$$

Then calculates the error of the function for each prediction, but the purpose is the minimizing error. While analyzing the data, trying to assign data points with features on the classifier then each time it is necessary to decide function error. For reparametrize function like divides into 1 + t then takes derivative according to t. To find the best coefficients of the function, minimize(LSE) the error by updating coefficients and bias with gradient descent until the model becomes accurate.

$$Likelehood(\beta_0, \beta) = \prod_{i=0} P(x_i) \prod_{i=0} (1 - P(x_i)) \ , \ LSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i))^2$$

The outcome is measured with a variable (in which there are only two possible outcomes). It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. For this data set, firstly using ordinal logistic regression then it is better to using binary logistic regression with modified dataset.

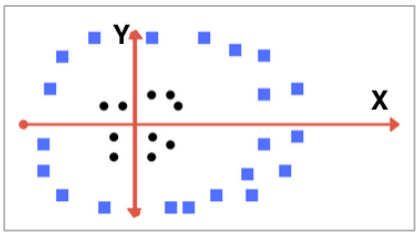
4.1.2 SVM (Support Vector Machine)

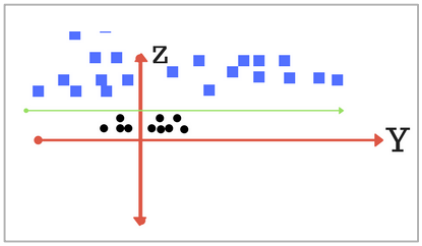
SVM is a discriminative classifier means linearly separable dataset with using a separating hyperplane. In the SVM, the main purpose is the classify data then maximize the margin between the data points and the hyperplane $(f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots \beta_n x_n)$ which is in p dimensions is a flat affine subspace of dimension p – 1. If f(X)>0 for points on one side of the hyperplane, and f(X) < 0 for points on the other. Data is trained with output class labels and then it is predicted with test data set then calculate accuracy how the algorithm predicts test data correctly. Among all separating hyperplanes, find the one that makes the biggest gap or margin (distance of closest examples from the decision line/ hyperplane) between the two classes. It means the purpose is the maximize the margin (1/||w||) with subject to $y_i\ (\beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots \beta_n x_n) \geq 1$ for all Bs.

Sometimes the data are separable, but noisy. This can lead to a poor solution for the maximal-margin classifier, therefore it maximizes a soft margin adding slack variable. Briefly, soft margin can misclassify number of slack variable. In order to maximize the margin, cost function (loss function) and gradient descent is used then according to the loss function result, model is trained in range the loss function result. The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. Briefly, cost function implies that how many point of each training dataset, you can misclassify.

$$c(x, y, f(x)) = \begin{cases} 0, & if\ y * f(x) \geq 1 \\ 1 - y * f(x), & else \end{cases}$$

Additionally, SVM is useful for non-linear classification by increasing dimension then find a new line classification in increased dimension. As it seems from figures in below, first image has two dimensions and changed dimension and with new z dimension, data is separable with a linear line; hyperplane.





In order to pick the best parameters for this algorithm **grid-search** technique will be used thanks to ***Sklearn*** library functionality. In the Sklearn library there is already predefined SVC function has some parameters like Cost parameter (soft margin) represented as C. This parameter tells the SVM optimization how many misclassifying point of each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. For very tiny values of C, there are misclassified examples, often even if training data is linearly separable.

4.1.3 Decision Tree

A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome (categorical or continues value). Tree models where the target variable can take a discrete set of values are called **classification trees**; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. A decision tree is drawn upside down with its root at the top involves partitioning the data into subsets that contain instances with similar values (homogenous), then on the middle there are condition/internal node based on the tree split into branches/edges. The end of the branch that doesn't split anymore it is the decision/leaf tree, means that they are the last classification nodes(qualities).

The base algorithm of the decision tree; recursive binary splitting according to ***Gini Index***. Gini Index is used as cost function that is used to evaluate splits in the dataset. A *Gini* score gives an idea how good a split is by how mixed the classes are in the two groups created by the split. A perfect separation results in Gini score of 0, whereas the worst case split that result in 50/50 classes. It is calculated for every row of data and split the data according to binary tree logic and repeat that work recursively. Decision Tree algorithm is like;

1- compute the gini index for the dataset
2- for every feature:
    a. calculate gini index for all categorical values
    b. take average for the current feature
    c. calculate gini again
3- pick the best gini attribute from the list
4- repeat until get tree will be constructed (until last leaf)

In this procedure, all the features are considered and different split points are tried and tested using a cost function (gini index). The split with the best cost (or lowest cost) is selected. The cost function is used to understand how model split and predict the split dataset classifications. A gini index can be computed by summing the probability pi of an item with label i being chosen probability of a mistake in categorizing that item. It reaches its minimum(zero) when all cases in the node fall into a single target category.

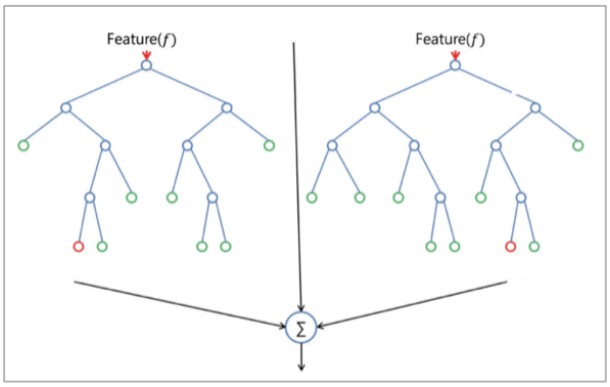
$$\sum_{k=1}^{K} P_{mk}(1 - P_{mk}) \qquad \text{Gini} = 1 - \sum_{i=1}^{J} P_i^2$$ j: #of unique labels

Additionally, decision tree has a feature that find the most related feature selection for splitting the tree by using gini score, the name is feature importance. **Feature importance** is calculated as the decrease in node weighted by the probability of reaching that node. The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples. The higher the value the more important the feature. In the decision tree algorithm, accuracy is also effected by depth of the tree. Depth of the tree is related with the feature importance means that getting the best related feature then split it according the order of features and go deep according to given max_depth parameter.

4.1.4 Random Forest

Random forests construct many individual decision trees at training and it uses the simplicity of decision trees with flexibility resulting in improvement the accuracy. Predictions from all trees are pooled to make the final prediction; the mode of the classes for classification or the mean prediction for regression. As they use a collection of results to make a final decision.

Random forest algorithm contains many variables, and many categorical variables with a large number of class labels. It gives results using data sets that show a loss or unbalanced distribution. When new trees are added into the random forest, algorithm updates itself with decreasing the loss by eliminating noises. While creating the big trees, the purpose is decreasing the standard deviation due to reaching least correlation between trees.



Random Forest is a model that allows us to make better estimates by using Decision Tree Algorithm N times on the data set as **Training and Test Set**. The number of times the Decision Tree Algorithm is run is determined by the n_estimators parameter. N is the average of the estimates obtained as a result of the algorithms run once, and produces a more accurate estimate.

In the random Forest algorithm, **feature importance** is calculated as the decrease in node weighted by the probability of reaching that node. The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples. The higher the value the more important the feature.

Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

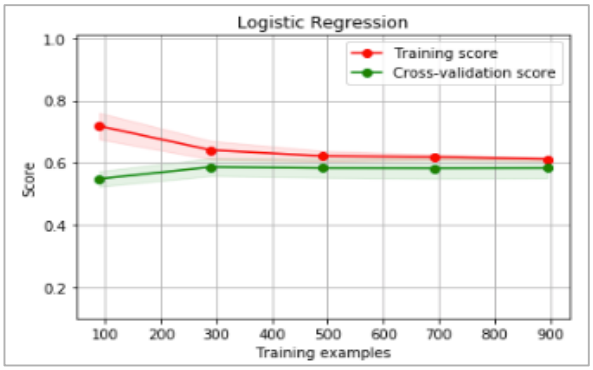
4.2 Classification Analysis in Ordered Dataset

4.2.1 Logistic Regression

Logistic regression model, trains the dataset with search the best parameters with grid search then fits the model on the training data.

```
Best Parameters for Logistic Regression:  LogisticRegression(C=1
0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='
warn',
          tol=0.0001, verbose=0, warm_start=False)
Best Score for Logistic Regression:  0.6041108132260947
```

Later divides the training set to get a better idea on the test accuracy thanks to cross-validation technique. According to the result of the classification algorithm, out of 10 trying, average accuracy is around 60%, means that the logistic regression classifier predicts 60% labels correctly. Learning curve is seen in the below figure, while training score is decreasing cross-validation score is increasing then at the end of each training accuracy is stop around 0.6 accuracy.



Lastly, predict on the split test set (not seen before), gets the test accuracy and confusion matrix. Thanks to sklearn library, logistic regression function is imported and used for predicting data which is randomly split. The result is in the below:

```
Mean Accuracy of Cross Validation: % 60.49
Std of Accuracy of Cross Validation: % 5.0
Confusion matrix of Logistic_Regression :
 [[  0   0   1   0   0   0]
 [  0   0  11   6   0   0]
 [  0   0 150  45   0   0]
 [  0   0  80 114   6   0]
 [  0   0   4  49   8   0]
 [  0   0   0   2   4   0]]
Accuracy of Logistic_Regression : 56.66666
================================
```

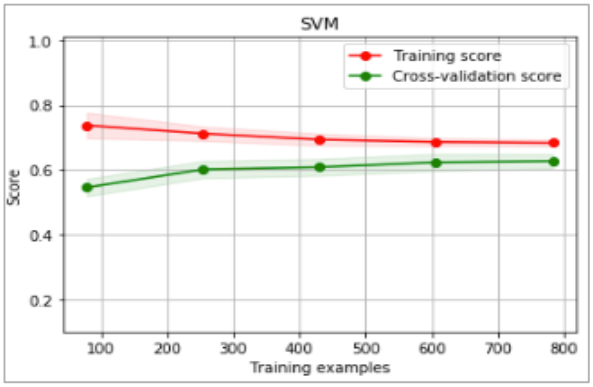
Then when analyzing to confusion matrix it is obvious that for average quality rates are predicted then the worst and best quality. The reason is that because of the unbalanced data, coefficients which are on the formula b0 and b1 are calculated according to these weights because average (5-6) quality has more data than others. Also, when looking to the accuracy it is around 56%, it seems good prediction rate but I don't think so because, the model predicted better for 5-6 quality rates and they have a big portion of all dataset, therefore model seems to have good accuracy rate.

4.2.2 SVM (Support Vector Machine)

SVM algorithm contains some parameters which are optimize the predictions better, but for finding the best parameters there is used grid search technique thanks to sklearn library of Python.

```
Best Parameters for SVM:  SVC(C=1.0, cache_size=200, class_weight=None
, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='rbf', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
Best Score for SVM:  0.613047363717605
```

Cross Validation technique is used to get a better idea on the test accuracy with training dataset. According to the result of the classification algorithm, out of 10 trying, average accuracy is around 61%, means that the SVM classifier predicts 61% labels correctly. Learning curve is seen in the below figure, while training score is decreasing cross-validation score is increasing then at the end of each training accuracy is stop around 0.61 accuracy. Additionally, there is a small gap between the training and cross-validation, means that it has high variance than logistic regression. Because SVM lie on a high variance with respect to logistic regression with the high margin technique.



After finding the best parameters for the SVM classifier, SVM kernel is rbf because data is not linearly separable, it can be checked data distribution from appendix figure 6-1. Rbf kernel gives the more accurate classification for this dataset and cost of the support vector machine should be 10 which is mentioned in above algorithm explanation part 4.1. in the text. According to those parameters SVM gives the best accuracy, the result is in the below:

```
Mean Accuracy of Cross Validation): % 61.82
Std of Accuracy of Cross Validation: % 5.0
Confusion matrix of SVM :
 [[  0   0   1   0   0   0]
 [  0   0  13   4   0   0]
 [  0   0 153  41   1   0]
 [  0   0  71 121   8   0]
 [  0   0   0  44  17   0]
 [  0   0   0   3   3   0]]
Accuracy of SVM : 60.62499999999999
================================
```

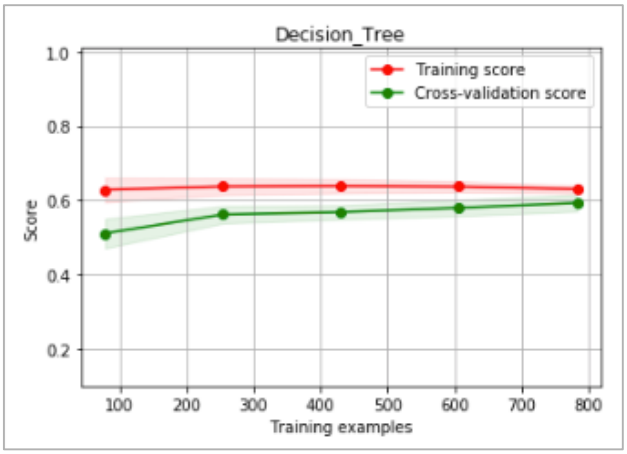
According to that accuracy result, it classifies better than logistic regression because it is supervised learning and thanks to cost function, it predicts more correct values than logistic, but still it is not find any low or high quality of the red wine.

4.2.3 Decision Tree

Decision tree algorithm contains some parameters which are optimize the predictions better, but for finding the best parameters there is used grid search technique thanks to sklearn library of Python.

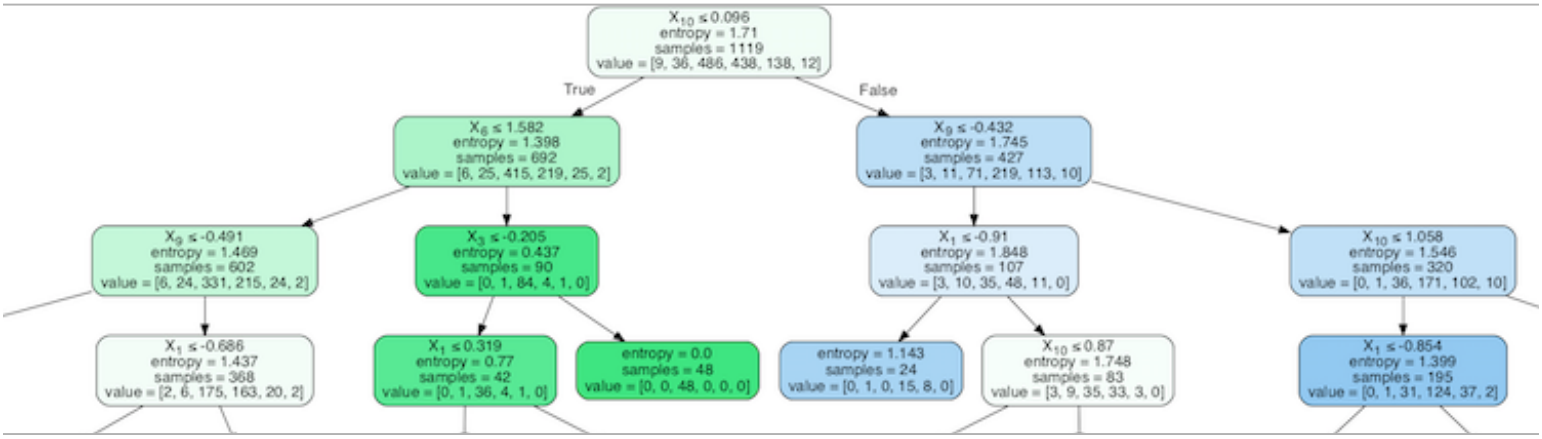
```
Best Parameters for Decision Tree: DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=4,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=17, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
Best Score for Decision Tree: 0.5969615728328865
```

For decision tree algorithm, max_depth parameter has a good impact for scoring the model. For that reason, grid_search is again used and found what should be the max_depth which is 4. After Grid search algorithm with the best parameters model is fitted with training data and calculate score with cross validation technique. Learning Curve of the decision tree seems very different than previous. Hence, samples take the same paths through the trees when training and predicting, therefore, training score is perfectly fit.

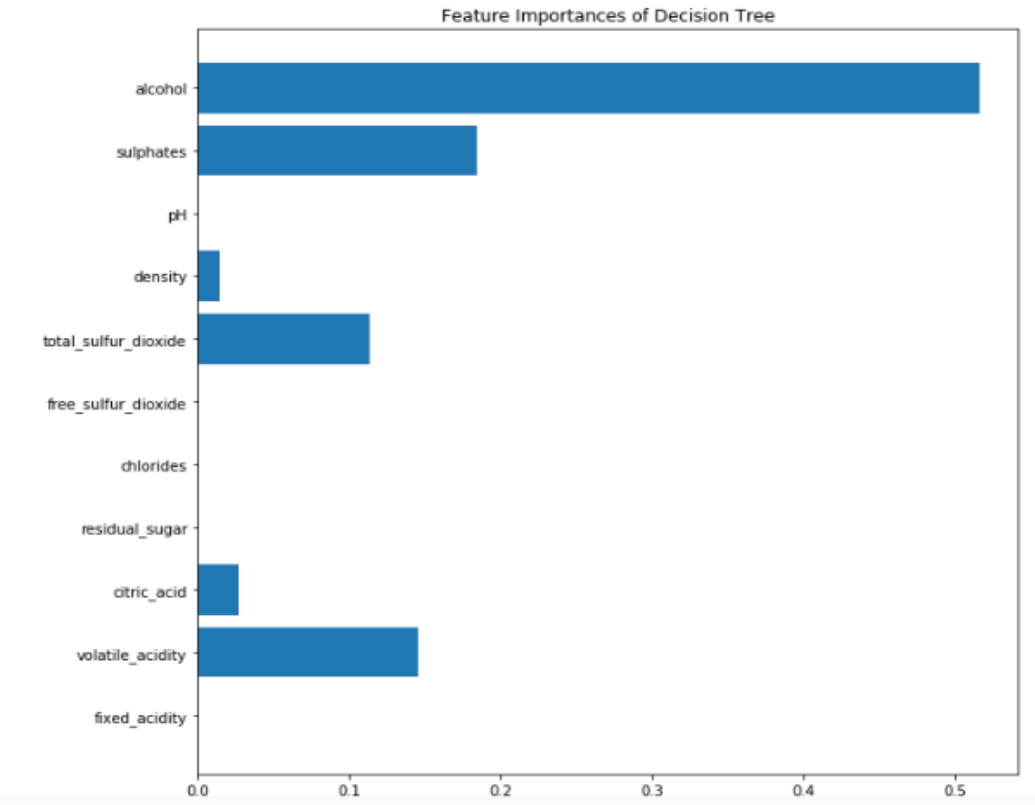


The cross-validated score increases a little because more samples both lowers bias (deeper trees + denser sampling from data structure) and lowers variance (decreased tree correlation + less sample error). By giving max depth = 4 then trained data then test, the accuracy seems more better than previous because decision tree algorithm uses feature importance by looking gini scores then split as binary separation. The some cut part decision tree output (for full version please check the appendix) is in the below with the accuracy score;

```
Mean Accuracy of Cross Validation: % 59.03
Std of Accuracy of Cross Validation: % 3.0
Confusion matrix of Decision_Tree :
 [[  0   0   1   0   0   0]
 [  0   0  11   6   0   0]
 [  0   0 154  39   2   0]
 [  0   0  93  88  19   0]
 [  0   0   8  39  14   0]
 [  0   0   0   4   2   0]]
Accuracy of Decision_Tree : 53.333333333333
================================
```



According to the output of the tree for full tree please check appendix figure 6-3, it is obvious that algorithm predicted only 3 category of the quality which are 5-6-7 not the rest. This is because of the maximum depth is restricted with 4. If the max depth remains default algorithm tries to predict all quality categories but it causes overfitting problems therefore max_depth should be selected as much as possible low. Additionally, this result shows that dataset is unbalanced and should be done something like decreasing the quality classification. Because of the wrong conditions, split was not good as much as possible because of the accuracy is not good enough to classify wine quality dataset. Also, Feature importance are shown in the below; as it seems from the plot alchol, sulphates, total_sulfur_dioxide, and volatile acidity has more weights than the other features therefore, tree split firstly checking those features.

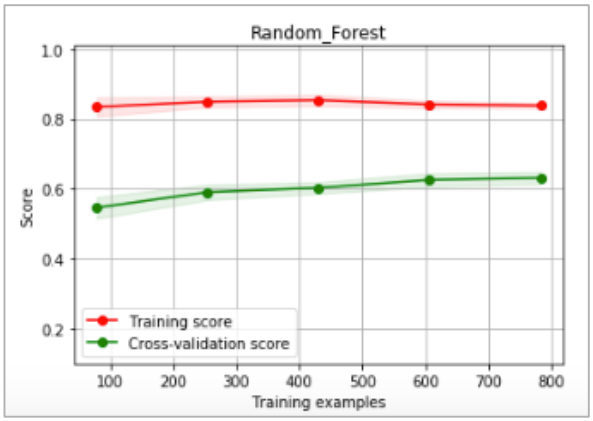


4.2.4 Random Forest

Random Forest algorithm runs number of times specified by the parameter of n_estimators = 100 as creating bootstrap decision trees. Therefore, it is obvious that the accuracy of this algorithm will be the best one because it runs 100 times with different random split dataset and then calculates the average of them by finding the best classification also. The result of this classification models is in the below;

```
Best Parameters for Random Forest:  RandomForestClassifier(bootstrap=False, class_weight=None, criterion='gini',
            max_depth=8, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=7, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
Best Score for Random Forest:  0.6353887399463807
```

As it seems from the best parameters, it is better with no bootstrapping, and criterion is better with gini, max_depth should be 8, bigger than then decision tree max_depth. With those best parameters, checking that the training and cross validation score by plotting with learning curve.

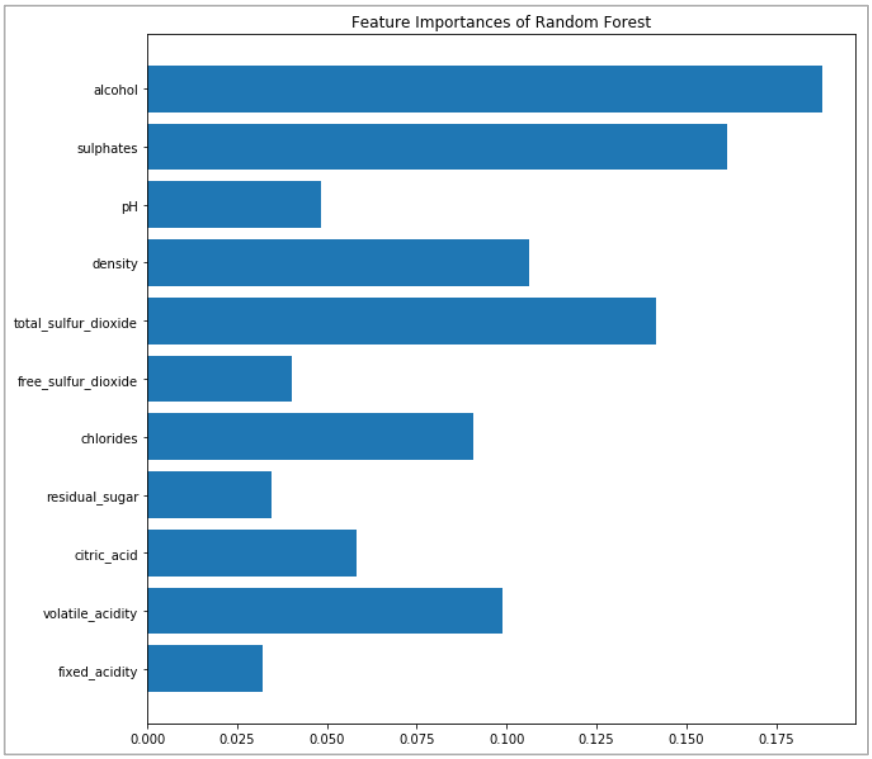


From the plot, it is obvious that it has more variance difference because of the gap between two lines. High variance is more better for training set because it trains well therefore it is expected that test accuracy should be higher than others.

```
Mean Accuracy of Cross Validation: % 64.6
Std of Accuracy of Cross Validation: % 5.0
Confusion matrix of Random_Forest :
 [[  0   0   1   0   0   0]
 [  0   0  11   6   0   0]
 [  0   0 151  40   4   0]
 [  0   0  61 124  15   0]
 [  0   0   5  35  21   0]
 [  0   0   0   3   3   0]]
Accuracy of Random_Forest : 61.666666666666
================================
```

It is obvious that, the best accuracy among those four classifications is random forest with the 61% accuracy. Hence, random forest algorithm is more powerful than others it tries 100 times with decision tree for classification to the data.

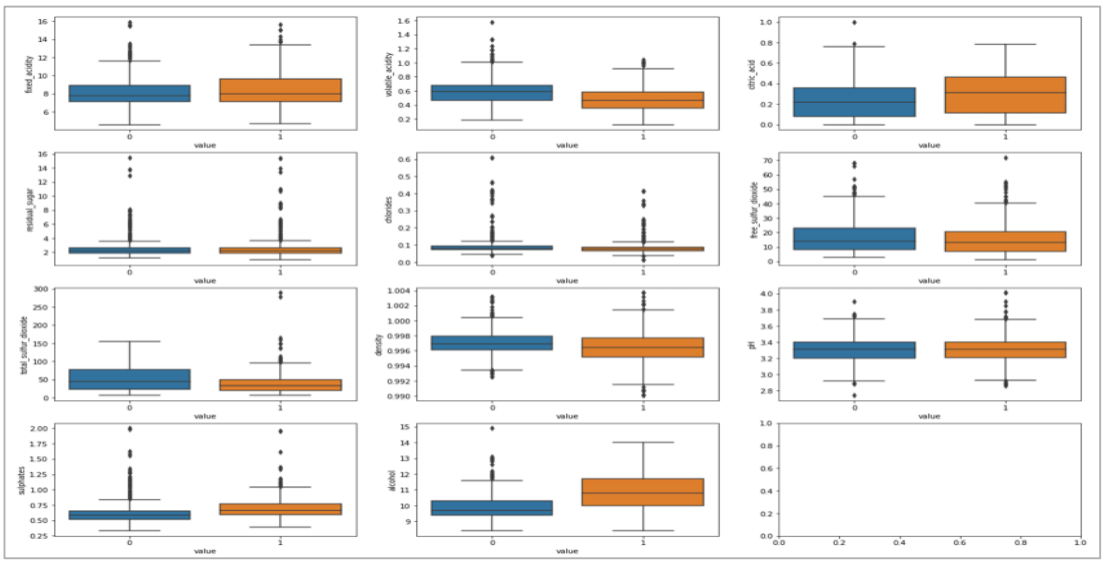
After calculating all ginis about the features, just look at the feature importance graph of random forest algorithm. It is quite different and detailed when it is compared with decision tree algorithm. Hence, the max_depth of the random forest is bigger than decision tree and also for each feature have some impact on the correct classification therefore it is more enhanced.



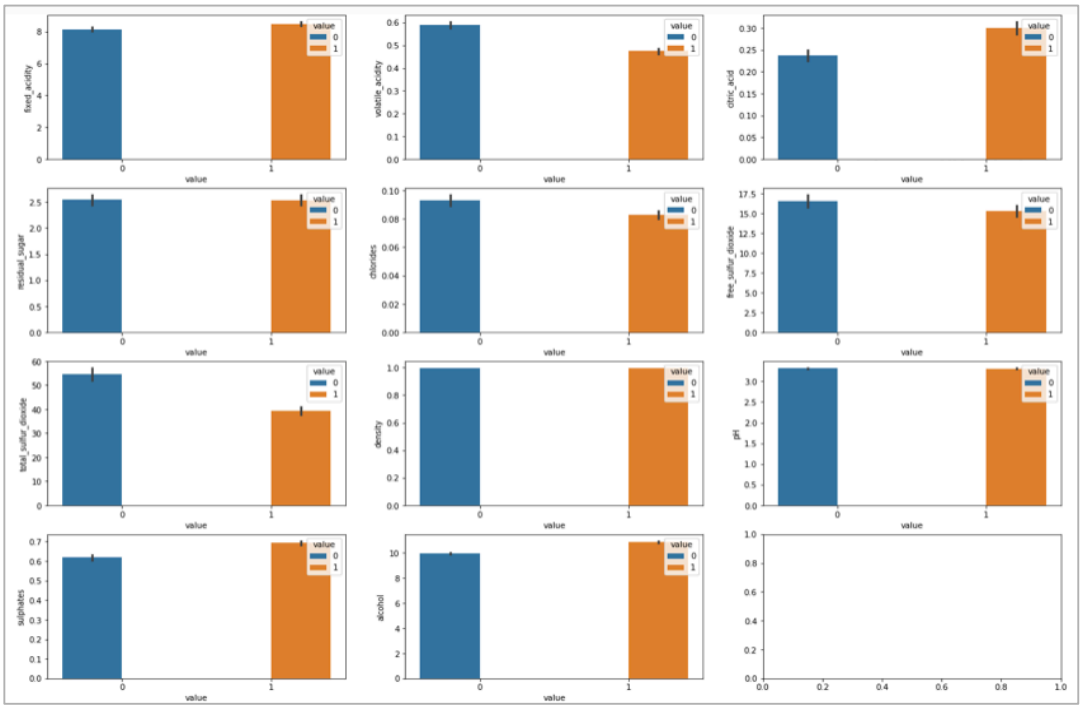
4.3 Classification Analysis in Binary Dataset

This dataset can be proper for classification and regression techniques because the data is multivariate, contains numeric values as an input and output value is ordered and not balanced means that there are much more normal wines than excellent or poor ones. For this reason, it is more convenient to start with making data balanced as turning the dataset as binary like “bad” and “good” wines. The condition of this altering is creating a new feature while checking quality scores are below than 5; wine is labeled as “bad”, otherwise “good”. After this update, with some features data can be separable (more detail in appendix Figure 6-5) with line and data seems with the value dimension as balanced because the count of those two classes seems very equal with each other Good wine dataset equals to 855, and bad wine dataset equals to 744. Then it can be modified quality column values with by value column values.

The distribution of the data with respect to value feature is shown in the figures (box plots and bar plots) in the below;



As it seems from the figure in the above, box plotting is a method for graphically represent groups of numerical data through their quartiles. The box extends from Q1 to Q3 quartile values of the data with a median line Q2. After updating data as binary, box plot diagram means more than the previous ordered dataset. Hence, now data seems more balanced therefore it gives relation more meaningful.



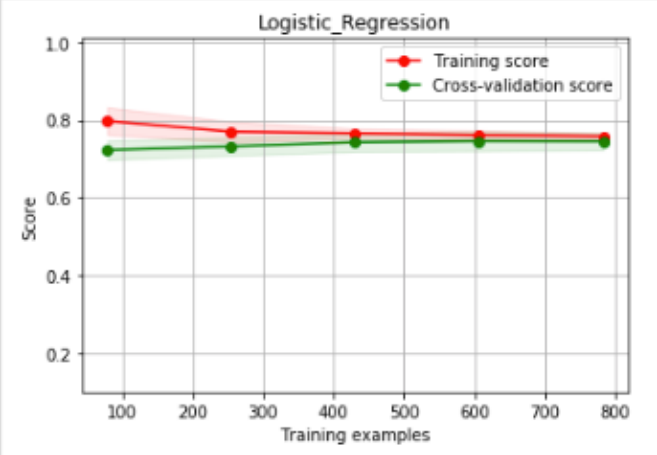
Also with bar plot diagram, making analysis turns easier than the previous one about which features have more relations with the value output label. As it seems from the figure, alcohol, citric_acid, sulphates features has obvious difference with output labels. Therefore, it is assumed that the predictions with same classification algorithm will be more better.

4.3.1 Logistic Regression

Logistic regression model, trains the dataset with search the best parameters with grid search then fits the model on the training data.

```
Best Parameters for Logistic Regression:  LogisticRegression(C=1, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l1', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
Best Score for Logistic Regression:  0.7479892761394102
```

After split data into test and training same as before, to get a better idea on the test accuracy thanks to cross-validation technique. According to the result of the classification algorithm, out of 10 trying, average accuracy is around 75%, means that the logistic regression classifier predicts 75% labels correctly. Learning curve is seen in the below figure, while training score is decreasing cross-validation score is increasing then at the end of each training accuracy is stop around 0.7 accuracy. Also, the difference between the lines are very few, it could be because binary classification is very easy to train and test.



After training, it is time to test with split test set with new logistic regression model to binary classification the output is in the below;

```
Mean Accuracy of Cross Validation: % 75.62
Std of Accuracy of Cross Validation: % 6.0
-----------------------------------------
Accuracy of Logistic_Regression : 73.75
Confusion matrix of Logistic_Regression :
 [[157  56]
 [ 70 197]]
=========================================
```

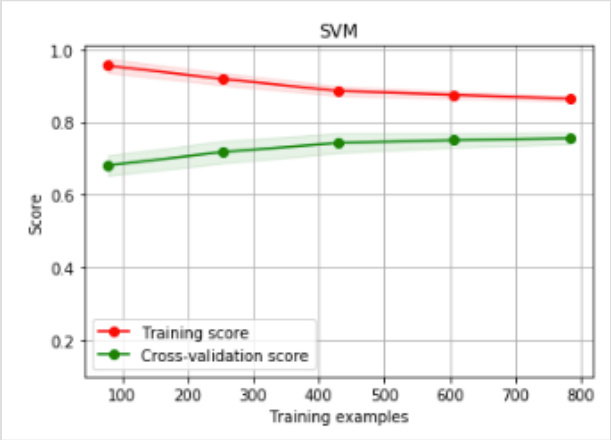
As it seems from the figure, accuracy is better than previous ordered dataset, because this dataset is easier to classify with just 2 class labels. Also, confusion matrix is easier to read, like model predicts bad wines correctly 157, and good wines 197. Model predicts the classifications wrongly with just 56 and 70 times. Therefore, accuracy increased thanks to seems balanced data.

4.3.2 SVM (Support Vector Machine)

SVM algorithm contains some parameters which should be optimized with grid search to increase the accuracy such as cost, kernel, gamma parameters.

```
Best Parameters for SVM: SVC(C=10.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='rbf', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
Best Score for SVM: 0.7569258266309205
```

As it seems from the figure in the above, SVM algorithm parameters are almost the same as before which working with the ordered dataset. Best score of training dataset is 75%, higher than the previous dataset because now classification with binary is easier to predict values. While predicting the average accuracy cross validation technique is used with 10 k folds that divides the training set into the same size 10 times and test with each time with split validation data. Then learning curve of this work is shown in the below;



SVM training score starting with very high accuracy around 90% then it decreased around 75% very sharply, on the other hand, testing score increased from 70% to 75% with slowly. Also, the gap between two lines are bigger than logistic regression because SVM algorithm is more better to finding high variance between values. After finding the best parameters for the SVM classifier, rbf kernel gives the more accurate classification for this dataset and cost of the support vector machine should be 10 same as before. According to those parameters SVM gives the best accuracy, the result is in the below:

```
Mean Accuracy of Cross Validation: % 76.51
Std of Accuracy of Cross Validation: % 5.0
----------------------------------------
Accuracy of SVM : 75.83333333333333
Confusion matrix of SVM :
 [[161  52]
 [ 64 203]]
========================================
```

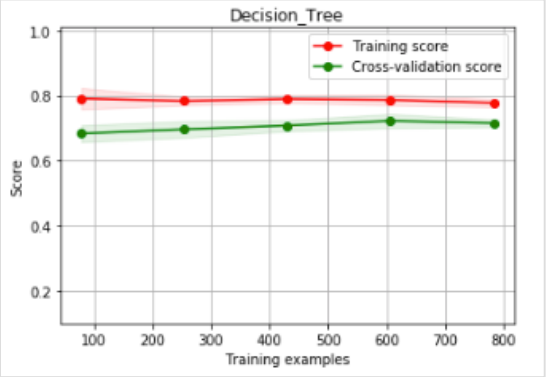
According to that accuracy result, it classifies better than logistic regression and also all classification with ordered data. Additionally, when confusion matrix was analyzed, correlation between the correct labeling is higher than logistic regression classification.

4.3.3 Decision Tree

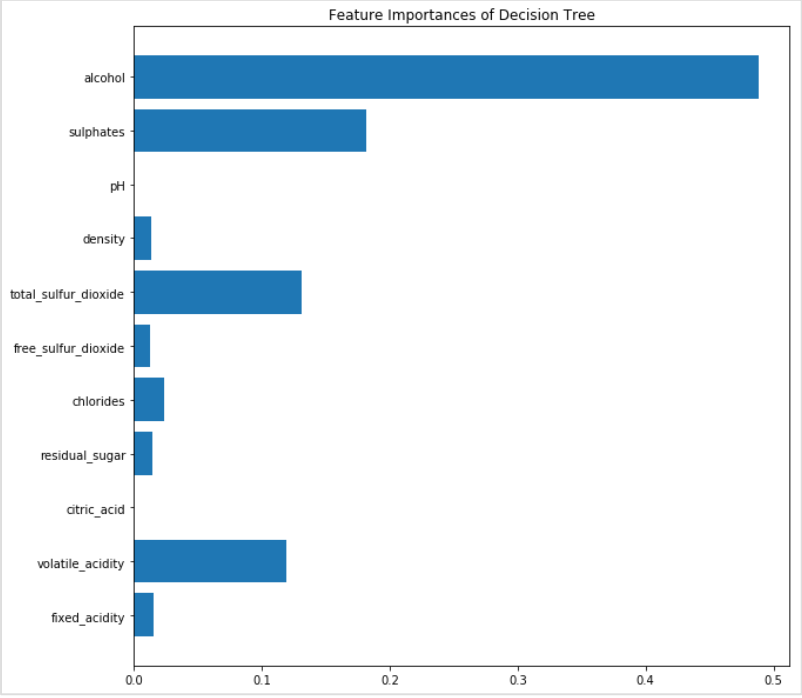
Decision tree is one of the best classification for binary dataset because the logic behind it is related with binary classification.

```
Best Parameters for Decision Tree:  DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=5,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=16, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
Best Score for Decision Tree:  0.739946380697051
```

Thanks to grid search algorithm, best parameters about decision tree was selected. Surprisingly, criterion is changed from gini to **entropy** which is information gain. Entropy uses logarithmic probability calculation different than gini. Generally, performance is not changed only entropy works slower than gini because of the logarithmic calculations. Learning curve is not so different than previous one the only difference is the accuracy because binary classification works easier.



According to entropy ordering, it is expected that features importance may be changed because of the criterion was changed, but surprisingly, there is not many differences in the feature importance values.

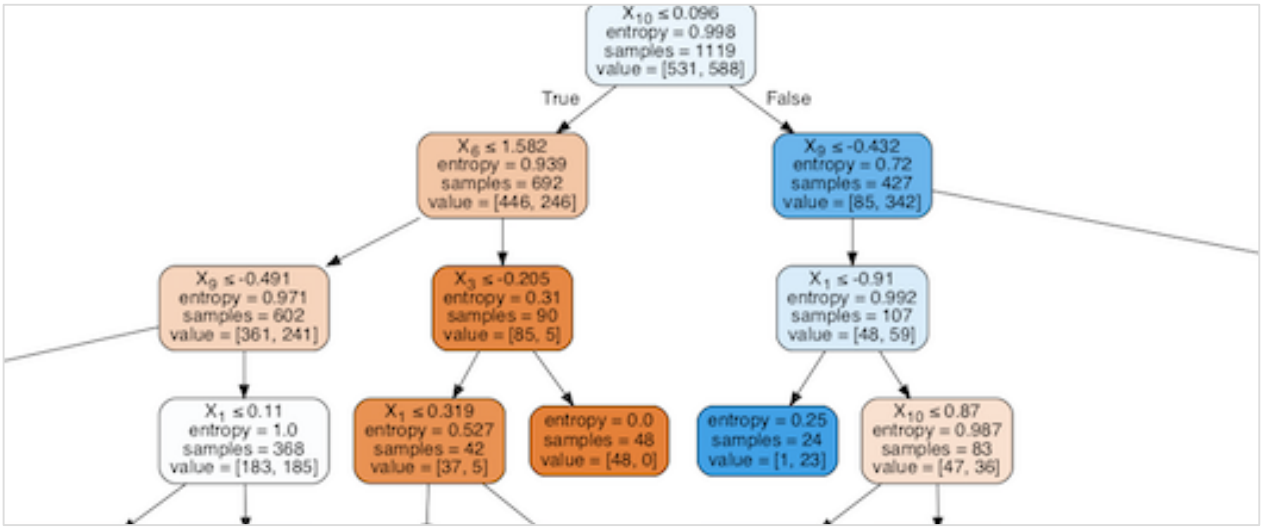


Only some features involved into creation of the tree, therefore, accuracy should be increased rapidly because every feature at least has some effect on the classification also dataset is changed to binary.

```
Mean Accuracy of Cross Validation: % 72.22
Std of Accuracy of Cross Validation: % 5.0
----------------------------------------
Accuracy of Decision_Tree : 70.20833333333333
Confusion matrix of Decision_Tree :
 [[168  45]
 [ 98 169]]
========================================
```

As it seems from the figure in the below, decision tree algorithm works better with binary dataset, but it works worse than SVM algorithm especially with good wine classification because almost all features used in SVM but in decision tree there was restrictions like max_depth.

As a result, there is a piece of the decision tree output is in the below; for more detail, please check the appendix Figure 6-4.

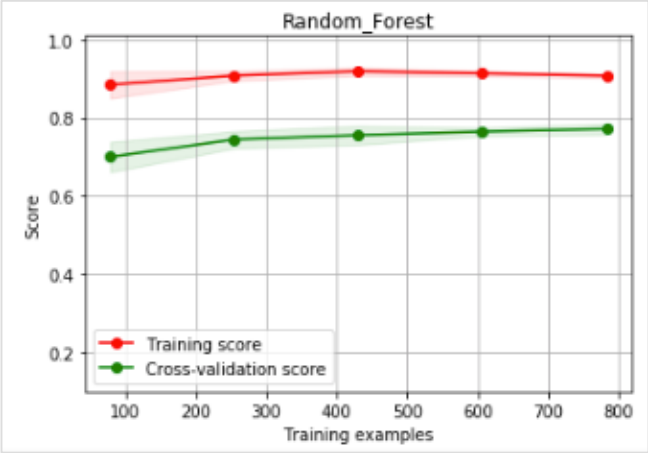


4.3.4 Random Forest

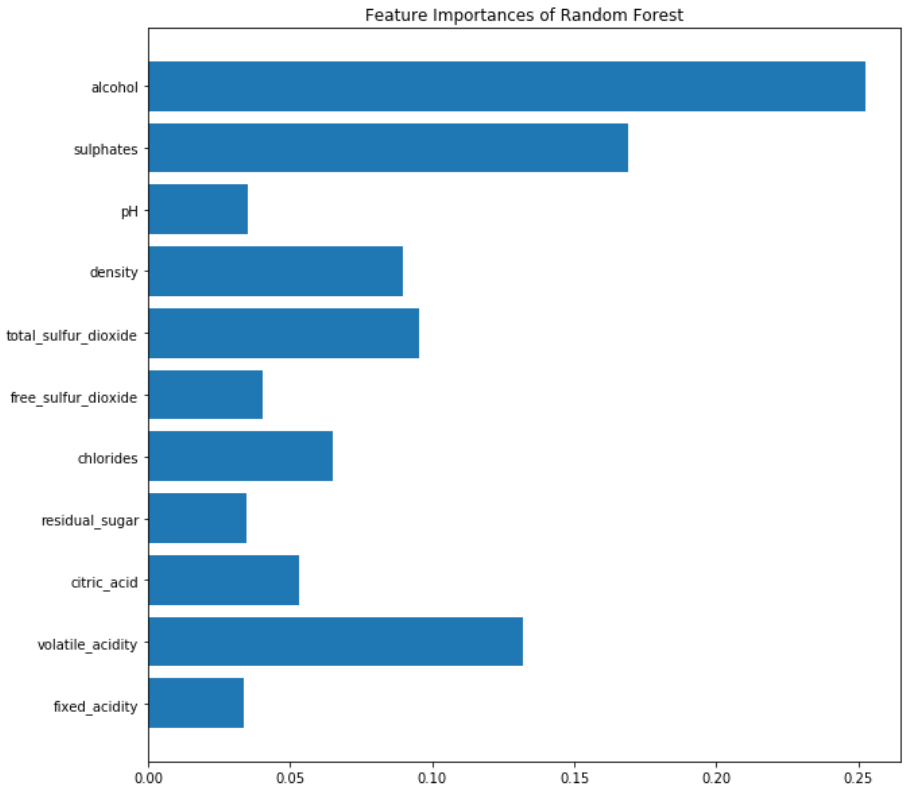
Random Forest algorithm runs number of times specified by the parameter of n_estimators = 100 as creating bootstrap decision trees for this dataset, bootstrap should be false according to best parameter estimators. Therefore, it is obvious that the accuracy of this algorithm will be the best one because it runs 100 times with different random split dataset and then calculates the average of them by finding the best classification also. The result of this classification models is in the below;

```
Best Parameters for Random Forest:  RandomForestClassifier(bootstrap=False, class_weight=None, criterion='gini',
            max_depth=9, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=7, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
Best Score for Random Forest:  0.773011617515639
```

With selected best parameters, training and validation scores were plotted in learning curve, the gap is higher than other classifications, it means that random forest algorithm is the best one for finding high variance and low error.



Differently from the decision tree, every feature has importance for classification because this algorithm runs many times n_estimator = 100 with randomly selected values. Therefore, every iteration there could be an effect of some features different than the decision tree, and each decision tree accuracy is taken and then calculating the accuracy of them then picking the best one for each iteration of the random forest algorithm.



When analyzing the scores, mean accuracy of the training is around 75% and standard deviation; variance of data reached the highest value around 6, it means this classification algorithm can predict labels very well like SVM.

```
-----------------------------------------
Mean Accuracy of Cross Validation: % 75.61
Std of Accuracy of Cross Validation: % 6.0
-----------------------------------------
Accuracy of Random_Forest : 74.375
Confusion matrix of Random_Forest :
 [[154  59]
 [ 64 203]]
=========================================
```

According to the output of the test result, accuracy is almost 75% similar with SVM, thanks to finding classes with high variance and low score, it means that 75% of time, this algorithm will classify data correctly.

5. Conclusion

For this thesis, it was aimed that the analyzing which psychochemical are more related with wine quality and which approach is good for prediction of wine quality better. After all work, it is obvious that working with binary classification is more better the predict good or bad wines.
During this research, four important machine learning techniques was used;

- Logistic regression
- Support Vector Machine
- Decision Tree
- Random Forest

From all algorithms, it was obvious that for this dataset, SVM and then Random Forest algorithm gave the best model and accuracy means that those algorithms predict correctly test data. If someone wants to analyze similar data like that it is better to work SVM or Random Forest. Hence, those algorithms variances are found better with high margin terminology, therefore with multiclass analysis, those algorithms will give the best accuracy.

After the analysis of this dataset, some features have more effect to deciding quality of the wine, there are some insights about the criteria about wine quality, you can compare just looking the some psychochemical on the label of wines;
Should be higher;

- Alcohol is the most important feature to decide quality of the wine. If the alcohol percentage is high enough, it means that quality of the wine should be better
- Sulphates is another selecting criteria for good wines, with high percentage sulphates wine quality is increasing
- Citric Acid is another selecting criteria, it should be higher to decide more better wine

Should be lower;

- Volatile Acidity should be less in the good wine
- Sulfur dioxide is another effect to decreasing wine quality and also it causes head ache therefore if there is less sulfur dioxide in wine, it should be selected
- Chlorides value has very less effect to quality of the wine but again it is obvious more value of it causes bad quality of the wine

Additionally, for marketing point of view, if a customer wants to buy a wine just looking with some psychochemical values can decide what s/he needs to buy. Of course, brand and price feature was evaluated on this research, therefore, it is not a good analysis for saying “it is good wine”. However, it can give some idea for the people who do not have more knowledge about wine for selecting the good wine maybe for just dinner or gift for friends!

6. Appendix

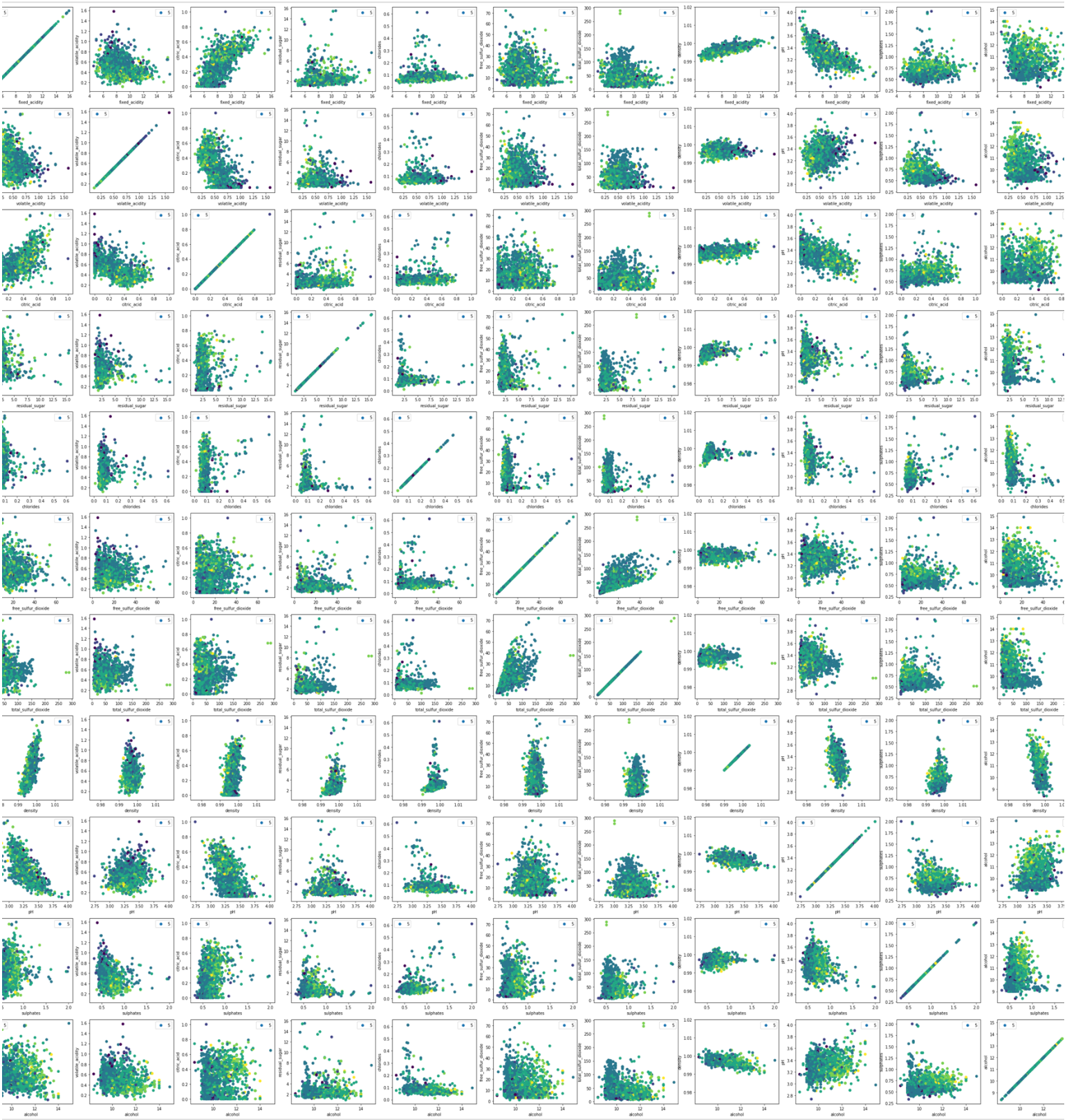


FIGURE 6-1: SCATTER PLOT OF DATA DISTRIBUTION

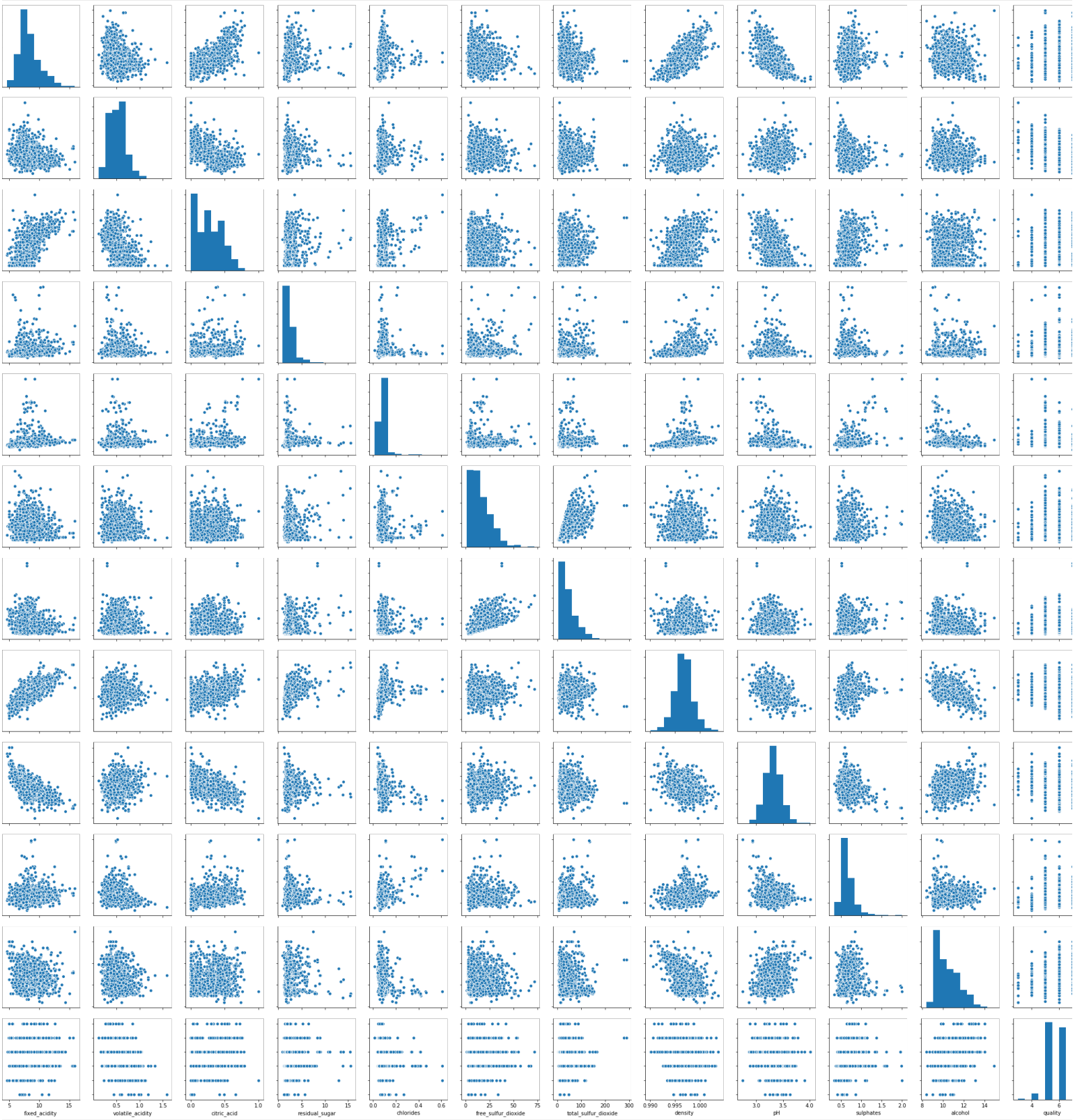


FIGURE 6-2: PAIR PLOT OF DATA DISTRIBUTION WITH RESPECT TO QUALITY OF WINE

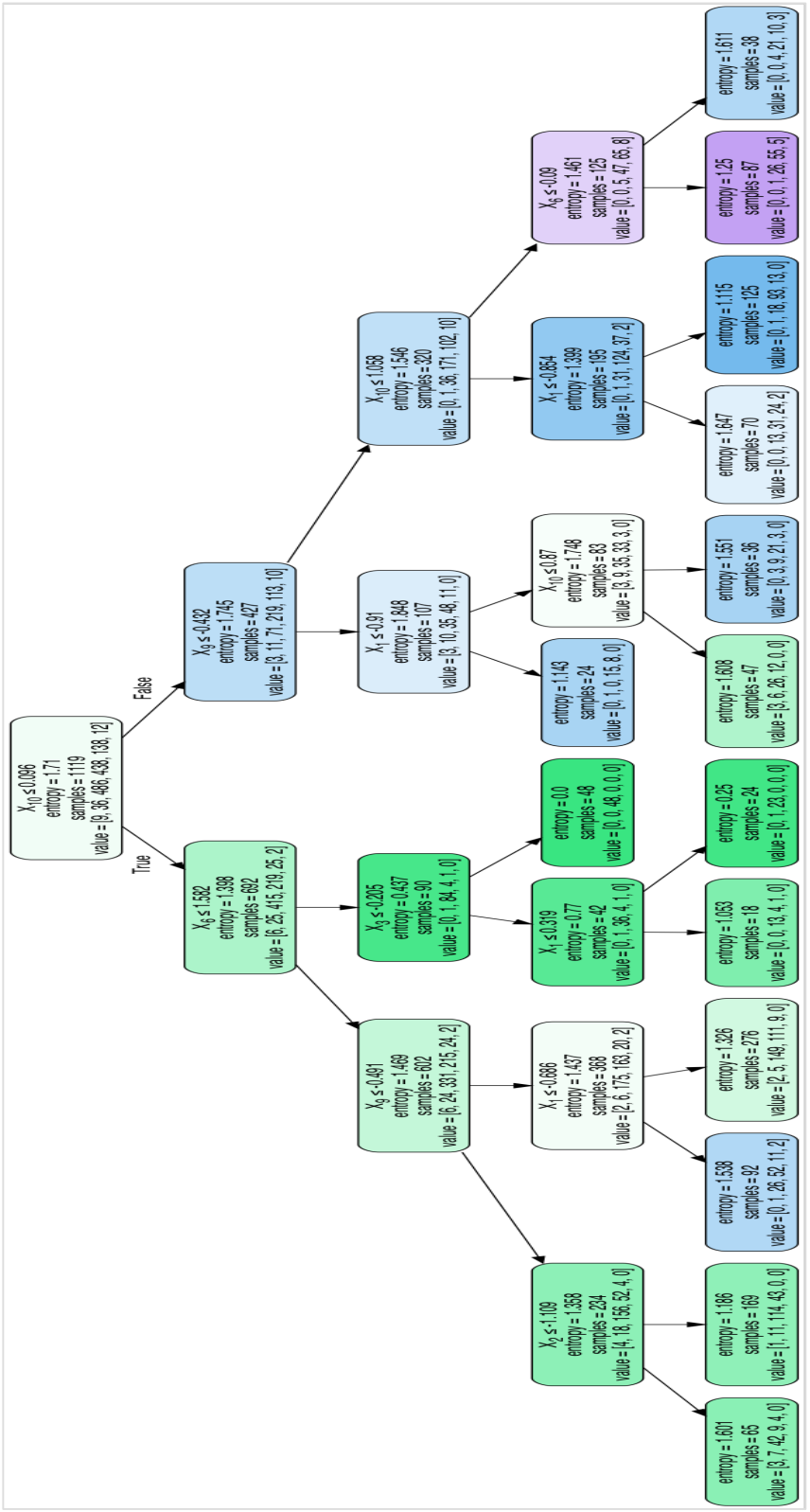


FIGURE 6-3: DECISION TREE WITH ORDERED DATASET

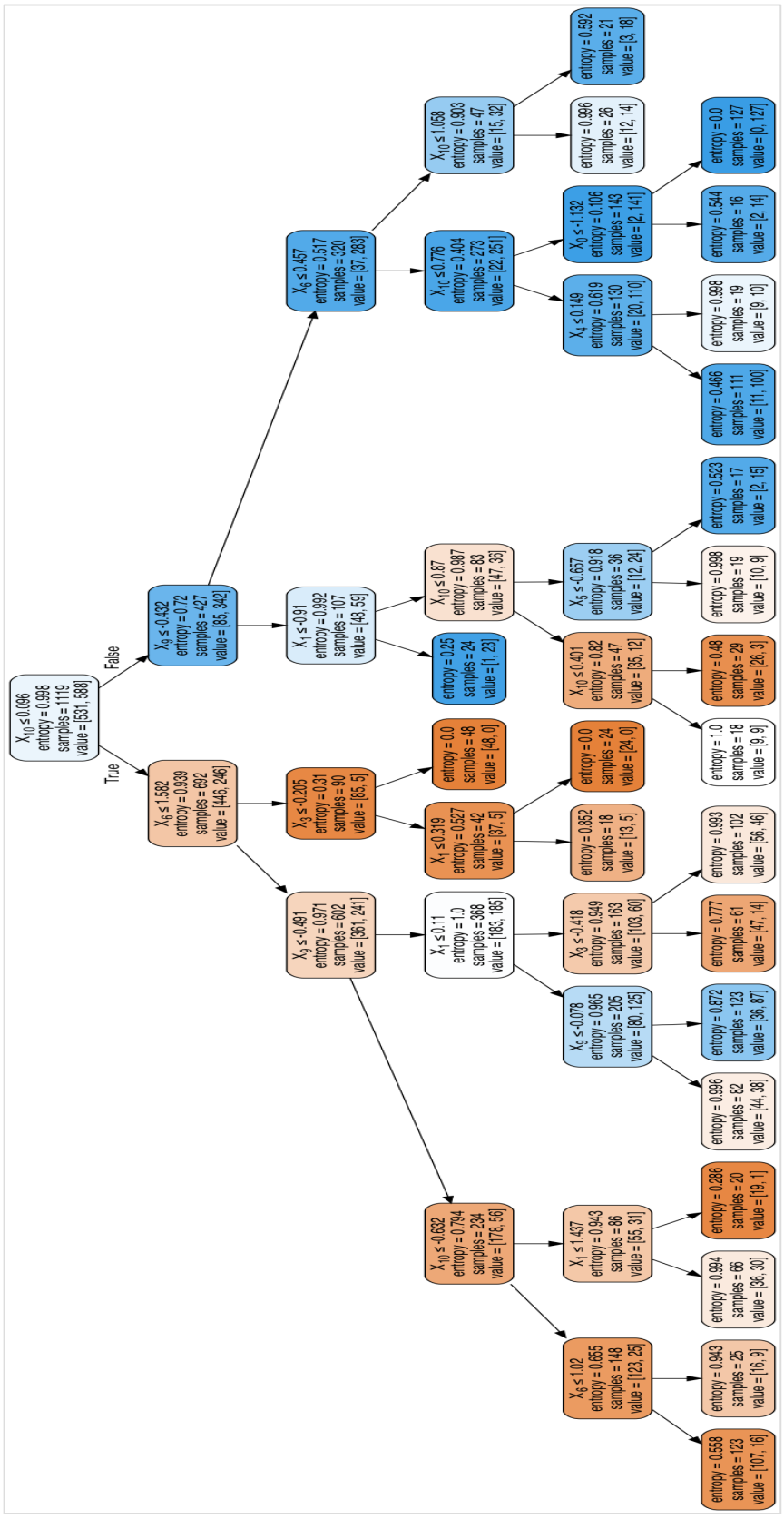


FIGURE 6-4: DECISION TREE WITH BINARY DATASET

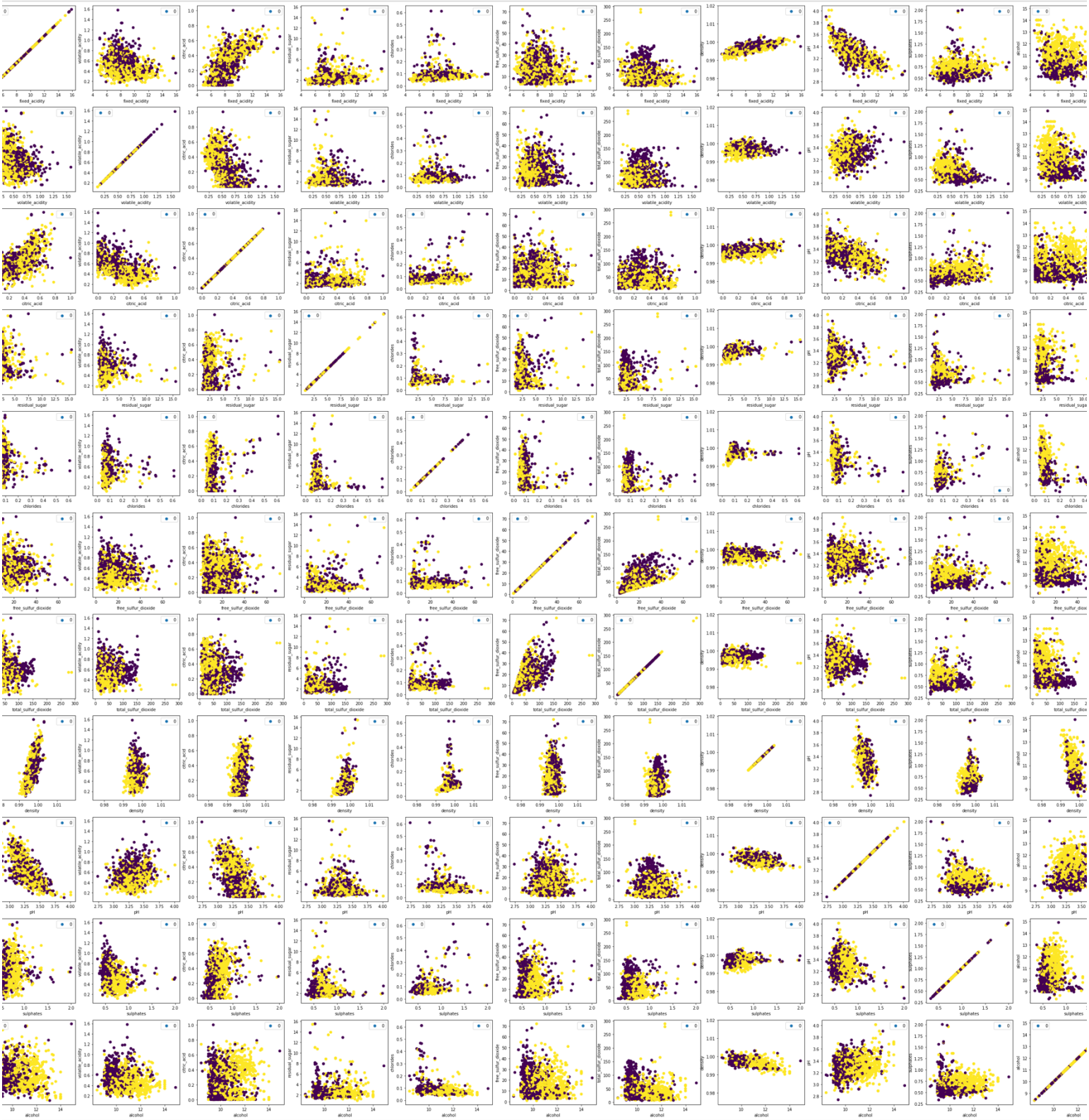


FIGURE 6-5: SCATTER PLOT OF BINARY DATA DISTRIBUTION

7. References

[1] https://www.anaconda.com/what-is-anaconda/
[2] https://pandas.pydata.org
[3] http://www.numpy.org
[4] https://matplotlib.org
[5] https://seaborn.pydata.org
[6] https://scikit-learn.org/stable/