**Mini Project on**

Title

**WINE QUALITY ANALYSIS**

By

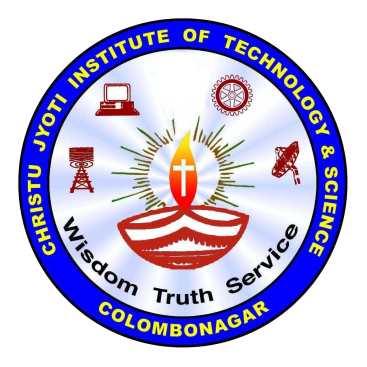
Name Roll No

M ABHIGNA REDDY 16681A0572

M BALA CHETHAN CHOWDARY 16681A0571

K SATHWIK REDDY 16681A0554

S CHAKRADHAR REDDY 16681A05A4



**Department of Computer Science & Engineering**

**CHRISTU JYOTHI INSTITUTE of technologY &Sciences**

**YESHWANTHAPUR,JANGAON-506167**

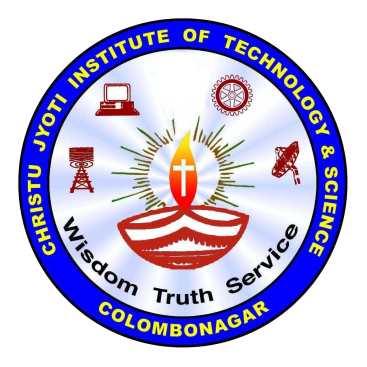
**(Affiliated to JNTUH, Accredited by NAAC)**

**CHRISTU JYOTHI INSTITUTE of technologY &Sciences**

**YESHWANTHAPUR, JANGAON-506167**

**(Affiliated to JNTUH, Accredited by NAAC)**

**CERTIFICATE**



This is to certify that the Mini project Report entitled “ WINE QUALITY ANALYSIS” is a Bonafide work of the students M Abhigna Reddy(16681A0572), M Bala Chethan Chowdary(16681A0571) , K Sathwik Reddy(16681A0554), S Chakradhar Reddy(16681A05A4)

**A SOWMYA S PURNA CHANDAR REDDY**

**Guide Head of the Department**

**S Chandrashekhar Reddy**

**Principal**

**ACKNOWLEDGEMENT**

Seminar & Presentation are important aspects for an engineering student’s carrier. It is basically to strengthen the practical concepts. During this seminar student gets acquainted with the latest technology and recent development and keep their selves updated with the latest technologies.

Firstly, I convey my sincere thanks to Mr. A Poorna Chandar Reddy **(HOD, CSE).** His love and guidance are omnipotent and incompatible throughout the preparation of this presentation. I convey special thanks to

A Sowmya **(lecturer, ECE)** for providing me the opportunity to undergo this training and I also express thanks to all faculties for their help and cooperation.

**Abstract**

Nowadays, industries are using product quality certifications to promote their products. This is a time taking process and requires the assessment given by human experts which makes this process very expensive. This paper explores the usage of machine learning techniques such as linear regression, neural network and support vector machine for product quality in two ways. Firstly, determine the dependency of target variable on independent variables and secondly, predicting the value of target variable. In this paper, linear regression is used to determine the dependency of target variable on independent variables. On the basis of computed dependency, important variables are selected those make significant impact on dependent variable. Further, neural network and support vector machine are used to predict the values of dependent variable. All the experiments are performed on Red Wine and White Wine datasets. This paper proves that the better prediction can be made if selected features (variables) are being considered rather than considering all the features.

## Introduction

Today, all type of industries is improving by adopting new technologies and applying these in all areas. These technologies are also helpful to enhance the production and making the whole process smooth. But, still there are different areas, which demands human expertise such as product quality assurance. Nowadays, it becomes an expensive process as the demand of product is growing over the time. Therefore, this paper explores different machine learning techniques such as linear regression, neural networks (*NN*) and support vector machines (*SVM*) for product quality assurance. These techniques performs quality assurance process with the help of available characteristics of product and automate the process by minimizing human interfere. The work also identifies the important features to predict the values of dependent variables.

In this work, all above mentioned machine learning techniques are used to support wine industry. Wine quality assessment is one of the key elements in this context and this assessment can be used for certification. Such type of quality certification helps to assure wine quality in market. Wine has various characteristics like density, pH value, alcohol and other acids. Wine quality can be assessed by two types of tests; first is physicochemical test and second is sensory test [1]. Physicochemical test can be determined by lab tests and no human expert is required but for sensory test, a human expert is required. Moreover, Wine quality assessment is very difficult as the relationships between the physicochemical and sensory analysis are complex and still not fully understood [2].

In literature, some researchers have used machine learning techniques to assess wine quality, but still a huge scope is available for improvement. Sun et al. [3] predicted *six* geographic wine origins based on neural networks fed with *15* input variables. They used 170 samples of data from Germany for their experiments. They got *100*% predictive rate. Vlassides et al. [4] also used neural network for classification of Californian wine. Grape maturity level and chemical analysis are used for wine classification. A sample of *36* examples was used for experiments and achieved only *6%* error. Moreno et al. [5] classified 54 wine samples into two red wine classes using probabilistic neural network. Yu et al. [6] classified 147 bottles of rice wine to predict *three* categories of wine using spectral measurements. Beltran et al. [7] used SVM, neural network and linear discriminate analysis to classify Chilean wine. The experiments and analyses were performed on three different varieties of Chilean wine. Cortez et al. [8] compared several classification of wine dataset. Jambhulkar et al. [9] used various techniques to predict heart disease using wireless sensor network.

They collected data from Cleveland dataset and extracted important attributes to predict heart disease. Zaveri et al. [10] predicted different diseases like TB, cancer, diabetes etc. using data mining techniques.

In this paper, linear regression, NN and SVM are implemented to determine dependency of wine quality on different *11* physicochemical characteristics. Moreover, the predictions are also made for wine quality on the basis of important variables/characteristics, selected according to their dependencies.

The paper is organized as follows: Section 2 provides the description and statistics of dataset used in this work. Section 3 discusses the proposed methodology in detail. Experimental results and analysis are explained in section 4. Conclusion is drawn in section 5.

## Dataset

In this work, Wine dataset is used for all the experiments. Wine dataset is a collection of white and red wines [11]. White wine consists of *4898* samples and red wine contains *1599* samples. Each sample of both types of wine consists of *12* physiochemical variables: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality rating. The quality rating is based on a sensory test carried out by at least three sommeliers and scaled in 11 quality classes from *0* - very bad to *10* - very excellent.

It is not possible to use both type of wine collections without preprocessing due to some deficiencies. One of the major deficiencies is the large amplitude of variable values e.g. sulfates (*0.3–2*) vs. sulfur dioxide (*1–72*). Moreover, some variables have values between 0 and 1. Such type of inconsistency may affect predictions due to more influence making by some variables than others. One of the ways to deal with such problem is linear transformation. Linear transformation can be achieved by dividing all the input values by maximum variable value.

## Proposed methodology

In this work, machine learning techniques are used to determine dependency of wine quality on other variables and in wine quality predictions. This section gives insights of proposed methodology. First Wine dataset is pre- processed as explained in previous section. Further, linear regression is applied to determine dependency of Wine quality on other *11* independent variables (predictors). Then after, important predictors are selected according to dependency of wine quality on independent variables. At last, Wine quality is predicted with the help of support vector machine and neural network considering all predictors and selected predictors. The working of used machine learning techniques is discussed in following subsections.

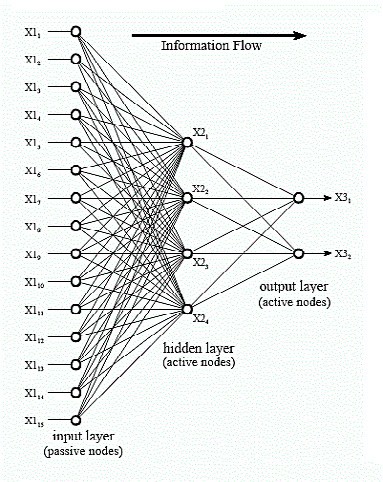
* 1. Linear regression

Linear regression models the relationship between a dependent variable and two or more than two independent variables. It is used to predict the value of a variable based on the value of two or more other variables. It also allows determining the overall fit of the model and the relative contribution of each of the predictors to the dependent variable.

A linear regression model with *k* number of predictors (independent variables) and one response (dependent) variable can be expressed as (1).

*Y = β0 + β1X1 + β2X2 + ∙ ∙ ∙ βk Xk + ϵ* (1)

where, *Y* is response variable and *Xi* are predictors (independent variables). *ϵ* is the residual term of the model, which is used for inference on the remaining model parameters. *β0, β1, β2, ..., βk* are regression coefficients.



**Fig. 1. Artificial neural network**.

1.1.1Neural network

Neural network is used by all human beings and animals to process information. It uses trillions of neurons for exchanging information through electrical pulses. But in the context of computer, NN is known as Artificial Neural Network (*ANN*). In this work, implemented neural network consists of *three* layers: input, hidden and summation, output as shown in Fig. 1. The function at input layer which passes input *Xi* to the hidden layer can be expressed as (2).

*n* *D2*

 *Yi exp(  i )*

(2)

*Y( X )*  *i**1*

*n*

*2 2*

*D2*

 *exp( i )*

*i* *1*

*2 2*

where *σ* is the width of the kernel. The output node computes *A/B*, which is *Y*.

The hidden layer consists of all training patterns *Xi*. When an unknown pattern *X* appears to this neural network then the squared distance between *X* and each *Xi* is computed. The squared distance can be expressed as (3).

*Di2 = (X − Xi )T (X − Xi )* (3)

This squared distance is passed to kernel function. Then after, these distances are passed through summation function.

* 1. Support vector machine

*SVMs* are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. *SVM* constructs a hyper plane or set of hyper planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. First, the model is trained by data, which is a set of points of the form as expressed in (4).

*D*  *{(x ,c ) | x*  *p ,c*  *{* *1,1 }}n*

*i i i i i* *1*

(4)

where *D* is training dataset *ci* represents the class which is either *1 or −1*. *xi* indicates a data point which is a *p*- dimensional real vector and each *xi* belongs to corresponding *ci*.

*SVM* creates a *p−1* dimensional hyperplane during training that separates data points into *two* classes. *SVM*

selects *one* hyperplane which has maximal distance to the support vectors (nearest data points).

## Experimental results and analysis

There are total *12* variables in both types of wine collections as discussed in section 2. The variable quality rating is considered as dependent variable and other *11* variables are assumed as predictors or independent variables in this work. Two types of analysis are done in this paper: firstly, the importance of each predictor for wine quality is identified and secondly, the value of wine quality is predicted using predictors.

Table 1. Linear regression summary of dependent variable (Quality) for Red Wine.

R= .60045958 R²= .36055170 Adjusted R²= .35611948 p < 0.0000

Standard regression

Standard error of b\*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | coefficient (b\*) |  | coefficient (b) |  | | |
| fixed acidity | 0.053879 | 0.055944 | 0.0250 | 0.02595 | 0.96308 | 0.335653 |
| volatile acidity | **0.240261** | **0.026851** | **1.0836** | **0.12110** | **8.94780** | **0.000000** |
| citric acid | -0.044038 | 0.035502 | -0.1826 | 0.14718 | -1.24044 | 0.214994 |
| residual sugar | 0.028513 | 0.026192 | 0.0163 | 0.01500 | 1.08860 | 0.276496 |
| chlorides | **0.109230** | **0.024436** | **1.8742** | **0.41928** | **4.47007** | **0.000008** |
| free sulfur dioxide | **0.056491** | **0.028124** | **0.0044** | **0.00217** | **2.00864** | **0.044745** |
| total sulfur dioxide | **0.132979** | **0.029684** | **0.0033** | **0.00073** | **4.47983** | **0.000008** |
| density | -0.041789 | 0.050558 | -17.8812 | 21.63310 | -0.82657 | 0.408608 |
| pH | **0.079080** | **0.036628** | **0.4137** | **0.19160** | **2.15897** | **0.031002** |
| sulphates | **0.192336** | **0.023999** | **0.9163** | **0.11434** | **8.01430** | **0.000000** |
| alcohol | **0.364470** | **0.034948** | **0.2762** | **0.02648** | **10.42901** | **0.000000** |

Raw regression

Standard t value p-value error of b

* 1. Determining important features for prediction

Table 1 represents the regression summary for dependent variable i.e. quality for Red Wine. This table shows the dependency of quality on all predictors individually. Here, *R* is the co-relation coefficient which indicates how much dependent variable (quality) is co-related with all predictors as whole. *R2* and *adjusted R2* are also computed from *R*. The value of *Adjusted R2* is 0.*3561* which shows *35*.*61*% dependency of quality on all predictors as whole. This value is less than *50%* which indicates that there are one or more predictors; those are not good for predicting the value of quality. At the same time, *P-value* is much less than 0.05 as shown in Table 1, which indicates that adjusted R2 is significantly different from *Zero* and deny null hypothesis.

Table 1 also shows the standardized regression coefficients (*b\**), standard error of *b\*,* raw regression coefficients (*b*), standard error of *b,* t-test (*t*) and *p-value*. The values of these coefficients are used to compare the relative contribution of each predictor in prediction of quality. The predictors such as volatile acidity, chlorides, free sulfur dioxide, total sulfur dioxide, pH, sulphates and alcohol are important predictors of quality as *p-value* for these predictors is less than 0.05 (95% confidence interval). The raw or un-standardized regression coefficient (*b*) for volatile acidity is 1.0836, which indicates that if all other predictors are controlled (constant) then increment of one unit in volatile acidity increases the quality by 1.0836. The same statement can be made for other predictors.

Similarly, Table 2 represents the regression summary for quality of White Wine. Table 2 clearly indicates that quality of White Wine is *28.02%* dependent on all predictors as whole. This value is less than 50%, it means there are one or more predictors; those are not good for predicting the white wine quality. Table 2 also shows that *p-value* for individual predictors such as fixed acidity, volatile acidity, residual sugar, free sulfur dioxide, density, pH, sulphates and alcohol is much less than 0.05, which means these predictors predict quality more significantly in comparison to others. The un-standardized regression coefficient (*b*) is 0.06 for fixed acidity, which indicates that if all other predictors are controlled (constant) then increment of one unit in fixed acidity increases the quality by 0.06. The same statement can be made for other predictors.

Table 2. Linear regression summary of dependent variable (Quality) for White Wine.

R= .53091465 R²= .30675369 Adjusted R²= .28025362 p < 0.0000

Standard

Standard

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| regression error of b\* regression error of b t value | | | | | | p-value |
|  | coefficient (b\*) |  | coefficient (b) |  |  |  |
| fixed acidity | **0.062430** | **0.019889** | **0.066** | **0.02087** | **3.1389** | **0.001706** |
| volatile acidity | **0.212048** | **0.012951** | **1.863** | **0.11379** | **16.3733** | **0.000000** |
| citric acid | 0.003019 | 0.013087 | 0.022 | 0.09577 | 0.2307 | 0.817589 |
| residual sugar | **0.466653** | **0.043109** | **0.081** | **0.00753** | **10.8249** | **0.000000** |
| chlorides | -0.006100 | 0.013483 | -0.247 | 0.54654 | -0.4524 | 0.650973 |
| free sulfur dioxide | **0.071681** | **0.016210** | **0.004** | **0.00084** | **4.4219** | **0.000010** |
| total sulfur dioxide | -0.013712 | 0.018142 | 0.000 | 0.00038 | -0.7558 | 0.449791 |
| density | **0.507528** | **0.064417** | **150.284** | **19.07451** | **7.8788** | **0.000000** |
| pH | **0.117021** | **0.017967** | **0.686** | **0.10538** | **6.5131** | **0.000000** |
| sulphates | **0.081374** | **0.012936** | **0.631** | **0.10039** | **6.2905** | **0.000000** |
| alcohol | **0.268840** | **0.033656** | **0.193** | **0.02422** | **7.9878** | **0.000000** |

Raw

Standard

2.2 Predicting value of dependent variable (Quality)

Two different machine learning techniques neural network and *SVM* have been used to predict the wine quality in this work. The wine quality is predicted using all features of dataset and for selected features (determined from previous subsection) of dataset.

Table 3 represents neural network regression analysis for Red Wine data. In this work, 11-5-1 neural network architecture is used. Table 3 shows the predicted values against original values of quality for Red Wine in both the cases: for all features and for selected features. It is not feasible to show results for all documents, therefore the results are shown for few documents randomly. This table clearly indicates that wine quality is predicted more accurately for selected features in comparison to using all features. The overall summary of neural network regression for Red Wine dataset is also shown in Table 4. This table clearly depicts that the values of training error, testing error and validation error are less when selected features used for predictions rather than all features used for predictions.

Table 3. Predicting quality rating value using neural network regression analysis for Red Wine.

|  |  |  |  |
| --- | --- | --- | --- |
| Document No. Original Quality | | Quality Output (11- | Quality Output (7-5-1) for |
|  | value | 5-1) for all features | selected features |
| 1 | 5.000000 | 4.825105 | 5.023361 |
| 82 | 5.000000 | 5.118246 | 4.989526 |
| 228 | 5.000000 | 5.260592 | 5.116120 |
| 394 | 5.000000 | 4.780642 | 4.982879 |
| 709 | 6.000000 | 6.024094 | 6.007278 |
| 864 | 5.000000 | 4.966148 | 5.007411 |
| 1061 | 6.000000 | 6.267385 | 6.028679 |
| 1289 | 5.000000 | 5.210451 | 5.044965 |
| 1354 | 5.000000 | 5.209148 | 5.036600 |
| 1553 | 6.000000 | 5.935853 | 5.998063 |

Table 4. Overall summary of neural network regression analysis for Red Wine.

|  |  |  |  |
| --- | --- | --- | --- |
| Network name | Training error | Test error | Validation error |
| (All features) MLP 11-5-1 | 0.187312 | 0.195660 | 0.169588 |
| (Selected features) MLP 8-5-1 | 0.145736 | 0.146024 | 0.140383 |

Table 5 represents the predicting values of quality using SVM analysis for red wine dataset. This table proves that SVM gives more accurate predictions for selected features in comparison to using all features.

**Table 5. Predicting quality rating value using SVM analysis for Red Wine.**

|  |  |  |  |
| --- | --- | --- | --- |
| Document No. Original Quality | | Quality Output for | Quality Output for selected |
|  | value | all features | features |
| 1 | 5.000000 | 4.571732 | 4.725478 |
| 82 | 5.000000 | 4.955524 | 4.995552 |
| 228 | 5.000000 | 5.246439 | 5.089935 |
| 709 | 6.000000 | 6.134489 | 6.097158 |
| 864 | 5.000000 | 4.707895 | 4.914136 |
| 1061 | 6.000000 | 5.279055 | 5.780928 |
| 1289 | 5.000000 | 4.716610 | 4.953137 |
| 1354 | 5.000000 | 5.393295 | 5.070797 |
| 1553 | 6.000000 | 5.839632 | 5.949534 |

Similar analysis is done for white wine as shown in Table 6-8. Table 6 represents neural network regression analysis for white wine data. In this work, 11-5-1 neural network architecture is used to predict the values. Table 6 shows the predicted values against original values of quality for White Wine dataset in both the cases: for all features and for selected features. This table clearly indicates that more accurate predicted values are obtained for selected features in comparison to all features. The overall summary of neural network regression for white wine data is also shown in Table 7. This table shows that the values of test error, training error and validation error obtained for selected features are less than the model which includes all features. Table 8 represents the predicting values of quality using *SVM* analysis for white wine dataset. This table proves that *SVM* gives more accurate predictions for selected features in comparison to using all features.

Table 6. Predicting quality rating value using neural network regression analysis for White Wine.

|  |  |  |  |
| --- | --- | --- | --- |
| Document No. Original Quality | | Quality Output (11- | Quality Output (8-5-1) for |
|  | value | 5-1) for all features | selected features |
| 1 | 6.000000 | 5.533817 | 5.796299 |
| 26 | 6.000000 | 5.774385 | 5.854882 |
| 82 | 6.000000 | 5.944356 | 5.998266 |
| 179 | 4.000000 | 4.592797 | 4.209509 |
| 657 | 6.000000 | 6.035119 | 6.014234 |
| 1179 | 5.000000 | 5.432416 | 4.997073 |
| 1828 | 6.000000 | 5.665227 | 5.852938 |
| 2665 | 7.000000 | 6.715398 | 6.803775 |
| 2736 | 6.000000 | 5.271117 | 5.617078 |
| 4732 | 6.000000 | 6.414085 | 6.205033 |

Table 7. Overall summary of neural network regression analysis for White Wine.

|  |  |  |  |
| --- | --- | --- | --- |
| Network name | Training error | Test error | Validation error |
| (All features) MLP 11-5-1 | 0.234133 | 0.241568 | 0.243491 |
| (Selected features) MLP 8-5-1 | 0.190578 | 0.207456 | 0.199758 |

Table 8. Predicting quality rating value using SVM analysis for White Wine.

|  |  |  |  |
| --- | --- | --- | --- |
| Document No. Original Quality | | Quality Output for | Quality Output for selected |
|  | value | all features | features |
| 1 | 6.000000 | 5.694713 | 5.868677 |
| 26 | 6.000000 | 5.818790 | 5.951792 |
| 82 | 6.000000 | 5.076027 | 5.138848 |
| 179 | 4.000000 | 4.174945 | 4.094459 |
| 657 | 6.000000 | 5.848917 | 5.948470 |
| 1179 | 5.000000 | 4.585354 | 4.888971 |
| 1828 | 6.000000 | 5.650180 | 5.801910 |
| 2665 | 7.000000 | 7.429853 | 7.043530 |
| 2736 | 6.000000 | 5.797010 | 5.965150 |
| 4732 | 6.000000 | 6.087934 | 6.043868 |

Finally, it can be summarized that SVM is a better machine learning technique for wine quality predictions on the basis of results. At the same time, more precise predictions can be made by SVM and neural network using selected predictors rather than all predictors.

**CODE AND TEST CASE RESULTS**

In [1]:

!pip install scipy==1.2 –upgrade

Collecting scipy==1.2

Downloading https://files.pythonhosted.org/packages/67/e6/6d4edaceee6a110ecf6f318482f5229792f143e468b34a631f5a0899f56d/scipy-1.2.0-cp36-cp36m-manylinux1\_x86\_64.whl (26.6MB)

|████████████████████████████████| 26.6MB 2.8MB/s

Requirement already satisfied, skipping upgrade: numpy>=1.8.2 in /opt/conda/lib/python3.6/site-packages (from scipy==1.2) (1.16.4)

ERROR: allennlp 0.8.4 requires awscli>=1.11.91, which is not installed.

ERROR: allennlp 0.8.4 requires flaky, which is not installed.

ERROR: allennlp 0.8.4 requires responses>=0.7, which is not installed.

ERROR: kmeans-smote 0.1.2 has requirement numpy<1.16,>=1.13, but you'll have numpy 1.16.4 which is incompatible.

ERROR: kmeans-smote 0.1.2 has requirement scikit-learn<0.21,>=0.19.0, but you'll have scikit-learn 0.21.2 which is incompatible.

Installing collected packages: scipy

Found existing installation: scipy 1.3.0

Uninstalling scipy-1.3.0:

Successfully uninstalled scipy-1.3.0

Successfully installed scipy-1.2.0

In [2]:

import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.preprocessing import Normalizer

import statsmodels.api as sm

**Data Exploration**

In [3]:

sns.set(color\_codes=True)

wines = pd.read\_csv("../input/winequality.csv")

wines.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1599 entries, 0 to 1598

Data columns (total 12 columns):

fixedacidity 1599 non-null float64

volatileacidity 1599 non-null float64

citricacid 1599 non-null float64

residualsugar 1599 non-null float64

chlorides 1599 non-null float64

freesulfurdioxide 1599 non-null float64

totalsulfurdioxide 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 [4]:

wines.head()

Out[4]:

|  | fixedacidity | volatileacidity | citricacid | residualsugar | chlorides | freesulfurdioxide | totalsulfurdioxide | density | pH | sulphates | alcohol | quality |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 | 5 |
| 1 | 7.8 | 0.88 | 0.00 | 2.6 | 0.098 | 25.0 | 67.0 | 0.9968 | 3.20 | 0.68 | 9.8 | 5 |
| 2 | 7.8 | 0.76 | 0.04 | 2.3 | 0.092 | 15.0 | 54.0 | 0.9970 | 3.26 | 0.65 | 9.8 | 5 |
| 3 | 11.2 | 0.28 | 0.56 | 1.9 | 0.075 | 17.0 | 60.0 | 0.9980 | 3.16 | 0.58 | 9.8 | 6 |
| 4 | 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11.0 | 34.0 | 0.9978 | 3.51 | 0.56 | 9.4 | 5 |

In [5]:

wines.describe()

Out[5]:

|  | fixedacidity | volatileacidity | citricacid | residualsugar | chlorides | freesulfurdioxide | totalsulfurdioxide | density | pH | sulphates | alcohol | quality |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| count | 1599.000000 | 1599.000000 | 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 | 10.422983 | 5.636023 |
| std | 1.741096 | 0.179060 | 0.194801 | 1.409928 | 0.047065 | 10.460157 | 32.895324 | 0.001887 | 0.154386 | 0.169507 | 1.065668 | 0.807569 |
| min | 4.600000 | 0.120000 | 0.000000 | 0.900000 | 0.012000 | 1.000000 | 6.000000 | 0.990070 | 2.740000 | 0.330000 | 8.400000 | 3.000000 |
| 25% | 7.100000 | 0.390000 | 0.090000 | 1.900000 | 0.070000 | 7.000000 | 22.000000 | 0.995600 | 3.210000 | 0.550000 | 9.500000 | 5.000000 |
| 50% | 7.900000 | 0.520000 | 0.260000 | 2.200000 | 0.079000 | 14.000000 | 38.000000 | 0.996750 | 3.310000 | 0.620000 | 10.200000 | 6.000000 |
| 75% | 9.200000 | 0.640000 | 0.420000 | 2.600000 | 0.090000 | 21.000000 | 62.000000 | 0.997835 | 3.400000 | 0.730000 | 11.100000 | 6.000000 |
| max | 15.900000 | 1.580000 | 1.000000 | 15.500000 | 0.611000 | 72.000000 | 289.000000 | 1.003690 | 4.010000 | 2.000000 | 14.900000 | 8.000000 |

Exploring the high rated wines.i.e, the wines rated as 8 quality

In [6]:

high\_rated\_wines = wines.loc[wines['quality'] == 8]

high\_rated\_wines.describe()

Out[6]:

|  | fixedacidity | volatileacidity | citricacid | residualsugar | chlorides | freesulfurdioxide | totalsulfurdioxide | density | pH | sulphates | alcohol | quality |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| count | 18.000000 | 18.000000 | 18.000000 | 18.000000 | 18.000000 | 18.000000 | 18.000000 | 18.000000 | 18.000000 | 18.000000 | 18.000000 | 18.0 |
| mean | 8.566667 | 0.423333 | 0.391111 | 2.577778 | 0.068444 | 13.277778 | 33.444444 | 0.995212 | 3.267222 | 0.767778 | 12.094444 | 8.0 |
| std | 2.119656 | 0.144914 | 0.199526 | 1.295038 | 0.011678 | 11.155613 | 25.433240 | 0.002378 | 0.200640 | 0.115379 | 1.224011 | 0.0 |
| min | 5.000000 | 0.260000 | 0.030000 | 1.400000 | 0.044000 | 3.000000 | 12.000000 | 0.990800 | 2.880000 | 0.630000 | 9.800000 | 8.0 |
| 25% | 7.250000 | 0.335000 | 0.302500 | 1.800000 | 0.062000 | 6.000000 | 16.000000 | 0.994175 | 3.162500 | 0.690000 | 11.325000 | 8.0 |
| 50% | 8.250000 | 0.370000 | 0.420000 | 2.100000 | 0.070500 | 7.500000 | 21.500000 | 0.994940 | 3.230000 | 0.740000 | 12.150000 | 8.0 |
| 75% | 10.225000 | 0.472500 | 0.530000 | 2.600000 | 0.075500 | 16.500000 | 43.000000 | 0.997200 | 3.350000 | 0.820000 | 12.875000 | 8.0 |
| max | 12.600000 | 0.850000 | 0.720000 | 6.400000 | 0.086000 | 42.000000 | 88.000000 | 0.998800 | 3.720000 | 1.100000 | 14.000000 | 8.0 |

In [7]:

sns.pairplot(high\_rated\_wines)

Out[7]:

<seaborn.axisgrid.PairGrid at 0x7f6f6fe61940>

In [8]:

for feat **in** high\_rated\_wines.columns.drop('quality') :

print(feat)

print(high\_rated\_wines[feat].min())

print(high\_rated\_wines[feat].max())

fixedacidity

5.0

12.6

volatileacidity

0.26

0.85

citricacid

0.03

0.72

residualsugar

1.4

6.4

chlorides

0.044000000000000004

0.086

freesulfurdioxide

3.0

42.0

totalsulfurdioxide

12.0

88.0

density

0.9908

0.9988

pH

2.88

3.72

sulphates

0.63

1.1

alcohol

9.8

14.0

The two main features it looks like we need to pay attention to seems like: 'freesulfurdioxide' and 'totalsulpurdioxide'. Let's take a close look at that. I would really appreciate it if someone could help me identify outliers and how to analyse and help get rid of them.

In [9]:

wines.isnull().any().any()

Out[9]:

False

There are no null values or missing values in any of the columns, and the quality is an integer. So I am going to proceed in my regression analysis. From the above description of the columns, the features have different units are measured differently. So to be able to compare them during the analysis, I will Normalize my data.

In [10]:

transformed = Normalizer().transform(wines).reshape(1599,12)

transformed\_df = pd.DataFrame(transformed, columns = wines.columns)

transformed\_df.describe()

Out[10]:

|  | fixedacidity | volatileacidity | citricacid | residualsugar | chlorides | freesulfurdioxide | totalsulfurdioxide | density | pH | sulphates | alcohol | quality |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| count | 1599.000000 | 1599.000000 | 1599.000000 | 1599.000000 | 1599.000000 | 1599.000000 | 1599.000000 | 1599.000000 | 1599.000000 | 1599.000000 | 1599.000000 | 1599.000000 |
| mean | 0.215313 | 0.013444 | 0.006922 | 0.062610 | 0.002210 | 0.305282 | 0.823186 | 0.025545 | 0.084877 | 0.016670 | 0.269655 | 0.145597 |
| std | 0.123825 | 0.009042 | 0.006969 | 0.043870 | 0.001614 | 0.114296 | 0.127156 | 0.013206 | 0.044087 | 0.009074 | 0.146528 | 0.079768 |
| min | 0.027054 | 0.001027 | 0.000000 | 0.007761 | 0.000112 | 0.021607 | 0.325111 | 0.003401 | 0.010308 | 0.001747 | 0.042122 | 0.023972 |
| 25% | 0.114058 | 0.007056 | 0.001896 | 0.033486 | 0.001156 | 0.218865 | 0.759075 | 0.014536 | 0.048592 | 0.009351 | 0.149010 | 0.078805 |
| 50% | 0.183688 | 0.011177 | 0.004321 | 0.052861 | 0.001865 | 0.291105 | 0.849563 | 0.023041 | 0.076643 | 0.014783 | 0.241074 | 0.130899 |
| 75% | 0.296968 | 0.017290 | 0.010001 | 0.077837 | 0.002762 | 0.382126 | 0.919613 | 0.035081 | 0.116043 | 0.022927 | 0.372155 | 0.200347 |
| max | 0.661059 | 0.089465 | 0.035993 | 0.480944 | 0.018949 | 0.618892 | 0.989920 | 0.062126 | 0.209330 | 0.057165 | 0.710668 | 0.395291 |

Now I am going to see if the data is suitable for a Linear Regression Model. Im going to plot scatter plots of response variable (quality) against the features. This will allow me to visually determine whether or not there is a linear correlation between the features and dependent variable.

In [11]:

q2 ='quality'

Y = transformed\_df.loc[:,'quality']

for q1 **in** transformed\_df.columns :

plt.figure()

sns.regplot(x = q1, y = q2, data = transformed\_df, color='green')

plt.xlabel(q1)

plt.ylabel(q2)

So, from the above scatter plots we can see that 'fixedacidity', 'totalsulfurdioxide', 'sulphates', 'alcohol', and 'pH seem to have a linear correlation. To justify this a reggression line should be fit to each of these features and the correlation parameters like R squared, P-value, Coefficients, etc should be evaluated.

In [12]:

for f **in** ['fixedacidity', 'totalsulfurdioxide', 'sulphates', 'alcohol', 'pH'] :

X = transformed\_df.loc[:, f]

X = sm.add\_constant(X)

model = sm.OLS(Y, X).fit()

print(f, "vs Quality")

print(model.summary())

print("**\n\n\n**")

fixedacidity vs Quality

OLS Regression Results

==============================================================================

Dep. Variable: quality R-squared: 0.786

Model: OLS Adj. R-squared: 0.786

Method: Least Squares F-statistic: 5878.

Date: Wed, 03 Jul 2019 Prob (F-statistic): 0.00

Time: 15:50:19 Log-Likelihood: 3008.8

No. Observations: 1599 AIC: -6014.

Df Residuals: 1597 BIC: -6003.

Df Model: 1

Covariance Type: nonrobust

================================================================================

coef std err t P>|t| [0.025 0.975]

--------------------------------------------------------------------------------

const 0.0226 0.002 12.212 0.000 0.019 0.026

fixedacidity 0.5713 0.007 76.667 0.000 0.557 0.586

==============================================================================

Omnibus: 115.291 Durbin-Watson: 1.493

Prob(Omnibus): 0.000 Jarque-Bera (JB): 371.571

Skew: 0.320 Prob(JB): 2.06e-81

Kurtosis: 5.273 Cond. No. 8.46

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

totalsulfurdioxide vs Quality

OLS Regression Results

==============================================================================

Dep. Variable: quality R-squared: 0.795

Model: OLS Adj. R-squared: 0.795

Method: Least Squares F-statistic: 6182.

Date: Wed, 03 Jul 2019 Prob (F-statistic): 0.00

Time: 15:50:19 Log-Likelihood: 3040.8

No. Observations: 1599 AIC: -6078.

Df Residuals: 1597 BIC: -6067.

Df Model: 1

Covariance Type: nonrobust

======================================================================================

coef std err t P>|t| [0.025 0.975]

--------------------------------------------------------------------------------------

const 0.6060 0.006 102.282 0.000 0.594 0.618

totalsulfurdioxide -0.5592 0.007 -78.627 0.000 -0.573 -0.545

==============================================================================

Omnibus: 58.541 Durbin-Watson: 1.704

Prob(Omnibus): 0.000 Jarque-Bera (JB): 80.492

Skew: -0.366 Prob(JB): 3.32e-18

Kurtosis: 3.819 Cond. No. 13.2

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

sulphates vs Quality

OLS Regression Results

==============================================================================

Dep. Variable: quality R-squared: 0.799

Model: OLS Adj. R-squared: 0.799

Method: Least Squares F-statistic: 6367.

Date: Wed, 03 Jul 2019 Prob (F-statistic): 0.00

Time: 15:50:19 Log-Likelihood: 3059.5

No. Observations: 1599 AIC: -6115.

Df Residuals: 1597 BIC: -6104.

Df Model: 1

Covariance Type: nonrobust

==============================================================================

coef std err t P>|t| [0.025 0.975]

------------------------------------------------------------------------------

const 0.0146 0.002 7.797 0.000 0.011 0.018

sulphates 7.8599 0.099 79.793 0.000 7.667 8.053

==============================================================================

Omnibus: 173.977 Durbin-Watson: 1.582

Prob(Omnibus): 0.000 Jarque-Bera (JB): 1043.648

Skew: -0.307 Prob(JB): 2.37e-227

Kurtosis: 6.910 Cond. No. 110.

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

alcohol vs Quality

OLS Regression Results

==============================================================================

Dep. Variable: quality R-squared: 0.910

Model: OLS Adj. R-squared: 0.910

Method: Least Squares F-statistic: 1.613e+04

Date: Wed, 03 Jul 2019 Prob (F-statistic): 0.00

Time: 15:50:19 Log-Likelihood: 3699.1

No. Observations: 1599 AIC: -7394.

Df Residuals: 1597 BIC: -7384.

Df Model: 1

Covariance Type: nonrobust

==============================================================================

coef std err t P>|t| [0.025 0.975]

------------------------------------------------------------------------------

const 0.0056 0.001 4.438 0.000 0.003 0.008

alcohol 0.5193 0.004 126.993 0.000 0.511 0.527

==============================================================================

Omnibus: 308.057 Durbin-Watson: 1.777

Prob(Omnibus): 0.000 Jarque-Bera (JB): 2676.924

Skew: -0.641 Prob(JB): 0.00

Kurtosis: 9.208 Cond. No. 7.33

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

pH vs Quality

OLS Regression Results

==============================================================================

Dep. Variable: quality R-squared: 0.874

Model: OLS Adj. R-squared: 0.874

Method: Least Squares F-statistic: 1.110e+04

Date: Wed, 03 Jul 2019 Prob (F-statistic): 0.00

Time: 15:50:19 Log-Likelihood: 3432.7

No. Observations: 1599 AIC: -6861.

Df Residuals: 1597 BIC: -6851.

Df Model: 1

Covariance Type: nonrobust

==============================================================================

coef std err t P>|t| [0.025 0.975]

------------------------------------------------------------------------------

const 0.0020 0.002 1.307 0.192 -0.001 0.005

pH 1.6918 0.016 105.377 0.000 1.660 1.723

==============================================================================

Omnibus: 200.122 Durbin-Watson: 1.709

Prob(Omnibus): 0.000 Jarque-Bera (JB): 1637.556

Skew: -0.272 Prob(JB): 0.00

Kurtosis: 7.928 Cond. No. 22.9

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The parameters above tell us that we can conclude that these variables are linearly related to quality, and maybe used as the main predictors of quality.

**LinearRegression Model**

Let us seperate training set and test set, and fit the model. We will do one model that includes all the features and one model that includes on the features that we have shown to be highly correlated to the dependent variable.

In [13]:

from sklearn.model\_selection import KFold

from sklearn.linear\_model import LinearRegression

from sklearn import metrics

import statistics

In [14]:

Y = transformed\_df.quality

X = transformed\_df.drop('quality', axis = 1)

In [15]:

scores = []

r2 = []

kf = KFold(n\_splits=10)

for train\_index, test\_index **in** kf.split(X,Y) :

x\_train, x\_test = X.iloc[train\_index], X.iloc[test\_index]

y\_train, y\_test = Y.iloc[train\_index], Y.iloc[test\_index]

RegModel = LinearRegression()

RegModel.fit(x\_train, y\_train)

y\_pred = RegModel.predict(x\_test)

scores.append(metrics.mean\_squared\_error(y\_test,y\_pred))

r2.append(metrics.r2\_score(y\_test,y\_pred))

print(statistics.mean(scores))

print(statistics.mean(r2))

0.00042087411786056046

0.9279217540518573

Now let's create a Regression model with only those features that we pointed out were major predictors in the quality of the wine.

In [16]:

features = ['fixedacidity', 'totalsulfurdioxide', 'sulphates', 'alcohol', 'pH']

Y = transformed\_df.quality

X = transformed\_df[features]

In [17]:

feat\_scores = []

feat\_r2 = []

kf = KFold(n\_splits=10)

for train\_index, test\_index **in** kf.split(X,Y) :

x\_train, x\_test = X.iloc[train\_index], X.iloc[test\_index]

y\_train, y\_test = Y.iloc[train\_index], Y.iloc[test\_index]

RegModel = LinearRegression()

RegModel.fit(x\_train, y\_train)

y\_pred = RegModel.predict(x\_test)

feat\_scores.append(metrics.mean\_squared\_error(y\_test,y\_pred))

feat\_r2.append(metrics.r2\_score(y\_test, y\_pred))

print(statistics.mean(feat\_scores))

print(statistics.mean(feat\_r2))

0.0004761593172931765

0.9183769539772801

So from the above two models, we can see that both the models return different values of regression accuracy. The value of R2 for the feature regression is very close to the Regression with all inputs. From this I believe it's safe to assume that the selected features contribute to a lot of the prediction of quality. Let's draw the residual plots to see if the data is normally distributed. The Q-Q plot that I do below will be for one instance.

In [18]:

from sklearn.model\_selection import train\_test\_split

import pylab

In [19]:

for i **in** range(0,10) :

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.33)

reg = LinearRegression()

reg.fit(X\_train,y\_train)

pred\_vals = reg.predict(X\_test)

true\_vals = y\_test

residuals = true\_vals - pred\_vals

sm.qqplot(residuals, line='s')

pylab.show()

print("**\n**")

As you can see the QQ plot in the 10 samples is not completely linear. This could mean that there might be some outliers in the data (I don't know how to identify and analyze these). This definately means that the residuals are not showing complete homoscedasticity. This could affect the performance of the model, since homoscedasticity is a major assumption of simple Linear Regression. Because of this discrepency let us try to see if we can get a better model using other Regression Methods and not Simple Linear Regression

**Decision Tree Regressor**

We will take all the features as input and use DecisionTreeRegressor model.

In [20]:

from sklearn.tree import DecisionTreeRegressor

In [21]:

Y = transformed\_df.quality

X = transformed\_df.drop('quality', axis = 1)

In [22]:

tree\_scores = []

tree\_r2 = []

kf = KFold(n\_splits=10)

for train\_index, test\_index **in** kf.split(X,Y) :

x\_train, x\_test = X.iloc[train\_index], X.iloc[test\_index]

y\_train, y\_test = Y.iloc[train\_index], Y.iloc[test\_index]

RegTree = DecisionTreeRegressor()

RegTree.fit(x\_train, y\_train)

y\_pred = RegTree.predict(x\_test)

tree\_scores.append(metrics.mean\_squared\_error(y\_test,y\_pred))

tree\_r2.append(RegTree.score(x\_test, y\_test))

print(statistics.mean(feat\_scores))

print(statistics.mean(feat\_r2))

0.0004761593172931765

0.9183769539772801

**RandomForestRegressor Model**

In [23]:

from sklearn.ensemble import RandomForestRegressor

In [24]:

Y = transformed\_df.quality

X = transformed\_df.drop('quality', axis = 1)

In [25]:

ranfores\_scores = []

ranfores\_r2 = []

kf = KFold(n\_splits=10)

for train\_index, test\_index **in** kf.split(X,Y) :

x\_train, x\_test = X.iloc[train\_index], X.iloc[test\_index]

y\_train, y\_test = Y.iloc[train\_index], Y.iloc[test\_index]

RanForest = RandomForestRegressor(n\_estimators=100)

RanForest.fit(x\_train, y\_train)

y\_pred = RanForest.predict(x\_test)

ranfores\_scores.append(metrics.mean\_squared\_error(y\_test,y\_pred))

ranfores\_r2.append(RegTree.score(x\_test, y\_test))

print(statistics.mean(ranfores\_scores))

print(statistics.mean(ranfores\_r2))

0.00039560567145713413

0.9825571241484184

Selection of Final Model and Summary

From the analysis below, we can see that the RandomForestRegressor has a better R2 score, so that would be the model that is the best. As for the features that will most likely predict the quality, the LinearRegression model and our data analysis has confirmed that ['fixedacidity', 'totalsulfurdioxide', 'sulphates', 'alcohol', 'pH'] are the minimum number of features required to approximate the quality.

Predictor RandomForestRegressor Model

In [26]:

features = ['fixedacidity', 'totalsulfurdioxide', 'sulphates', 'alcohol', 'pH']

Y1 = transformed\_df.quality

X1 = transformed\_df[features]

In [27]:

pranfores\_scores = []

pranfores\_r2 = []

kf = KFold(n\_splits=10)

for train\_index, test\_index **in** kf.split(X1,Y1) :

x\_train, x\_test = X.iloc[train\_index], X.iloc[test\_index]

y\_train, y\_test = Y.iloc[train\_index], Y.iloc[test\_index]

pRanForest = RandomForestRegressor(n\_estimators=100)

pRanForest.fit(x\_train, y\_train)

y\_pred = RanForest.predict(x\_test)

pranfores\_scores.append(metrics.mean\_squared\_error(y\_test,y\_pred))

pranfores\_r2.append(RegTree.score(x\_test, y\_test))

print(statistics.mean(pranfores\_scores))

print(statistics.mean(pranfores\_r2))

8.310063587996954e-05

0.9825571241484184

## Conclusion and future directions

The interest has been increased in wine industry in recent years which demands growth in this industry. Therefore, companies are investing in new technologies to improve wine production and selling. In this direction, wine quality certification plays a very important role for both processes and it requires wine testing by human experts. This paper explores the usage of machine learning techniques in two ways. Firstly, how linear regression determines important features for prediction. Secondly, the usage of neural network and support vector machine in predicting the values. The benchmark Wine dataset is used for all experiments. This dataset has two parts: Red Wine and White Wine data. Red wine contains 1599 samples and white wine contains 4898 samples. Both red and white wine dataset consists of 12 physicochemical characteristics. One (quality) is dependent variable and other 11 are predictors. The experiments shows that the value of dependent variable can be predicted more accurately if only important features are considered in prediction rather than considering all features. In future, large dataset can be taken for experiments and other machine learning techniques may be explored for wine quality prediction.