VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JnanaSangama, Belgaum-590014

A Machine Learning Internship Report
On

"REDWINE QUALITY ANALYSIS"

Submitted in Partial fulfillment of the Requirements for the VII Semester of the Degree of

Bachelor of Engineering
In
Computer Science & Engineering
By

MEGHANA G(1CE17CS059)

Under the Guidance of

Mrs. Ambika P R
Asst. Professor, Dept. of CSE

CITY ENGINEERING COLLEGE

Doddakallasandra, Kanakapura Road,
Bengaluru-560061

CITY ENGINEERING COLLEGE

Doddakallasandra, Kanakapura Road, Bengaluru-560061

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Machine Learning Project work entitled **"REDWINE QUALITY ANALYISIS"** has been carried out By **MEGHANA G (1CE17CS059),** bonafide student of City Engineering College in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visveshvaraya Technological University, Belgaum during the year **2019-2020**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Machine Learning Mini Project Report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Mrs.Ambika P R	**Mr. B Vivekavardhana Reddy**	**Dr. V. S Rama Murthy**
Asst.Prof, Dept.of CSE	Head, Dept. of CSE	Principal

External Viva

Name of the examiners — Signature with date

1.

2.

Abstract

Nowadays people try to lead a luxurious life. They tend to use the things either for show off or for their daily basis. These days the consumption of red wine is very common to all. So it became important to analyze the quality of red wine before its consumption to preserve human health. Hence this research is a step towards the quality prediction of the red wine using its various attributes. Red wine quality and style are highly influenced by the qualitative and quantitative composition of aromatic compounds having various chemical structures and properties and their interaction within different red wine matrices. The understanding of interactions between the wine matrix and volatile compounds and the impact on the overall flavor as well as on typical or specific aromas is getting more and more important for the creation of certain wine styles. Based on the data visualisation of python processing, classical visualization tools such as boxplot, correlation matrix, jointplot and various algorithms for the result.

ACKNOWLEDGEMENT

While presenting this Machine Learning Project on **"RedWine Quality Prediction"**, I feel that it is my duty to acknowledge the help rendered to me by various persons.

Firstly I thank God for showering his blessings on us. I am greatful to my institution City Engineering College for providing me an congenial atmosphere to carry out the project successfully.

I would like to express my heartful gratitude to **Dr. V S Ramamurthy,Principal**, CEC, Bangalore, for extending his support.

I would like to express my heartful gratitude to **Prof.Vivekavardhana Reddy**, HOD, Computer Science and Engineering whose guidance and support was truly invaluable.

I am very grateful to our guide, **Mrs. Ambika P R** Asst. Prof, Department of Computer Science, for her able guidance and valuable advice at every stage of our project which helped me in the successful completion of our project.

I would also have indebted to our Parent and Friends for their continued mora and material support throughout the course of project and helping me in finalize the presentation.

My heartly thanks to all those who have contributed bits,, bytes and words to accomplish this project.

MEGHANA(1CE17CS059)

TABLE OF CONTENTS

Chapter	Title	Page No
Chapter-1	**Introduction**	**1**
Chapter-2	**Company Profile**	**3**
	2.1 Company Profile	
	2.2 Mission	
	2.3 Vission	
Chapter-3	**Literature Survey**	**5**
	3.1 Logistic Regression	
	3.2 Decission Tree Classifier	
	3.3 Random Forest Classifier	
	3.4 K Nearest Neighbour	
	3.5 Support Vector Machine	
Chapter-4	**Problem Statement**	**11**
	4.1 Problem Statement	
Chapter-5	**Data Summary**	**12**
	5.1 Attribute Information	
	5.2 Libraries Imported	
Chapter-6	**Implementation**	**14**
	6.1 Source Code	
Chapter-7	**Conclusion**	**25**

Chapter 1

INTRODUCTION

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

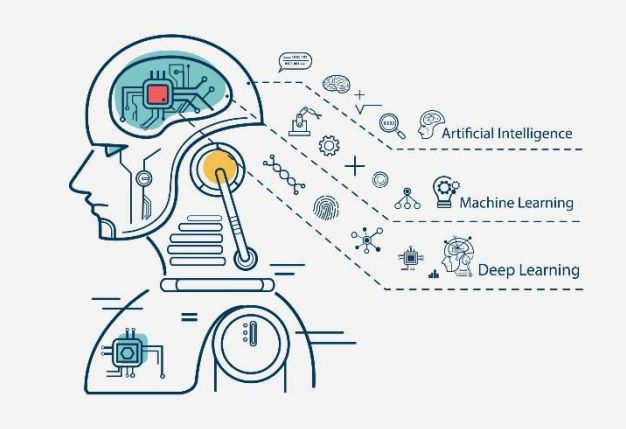


Fig 1.1: Introduction to ML

How does Machine Learning work:

Machine learning is a form of artificial intelligence (AI) that teaches computers to think in a similar way to how humans do learning and improving upon past experiences. It works by exploring data, identifying patterns, and involves minimal human intervention. Almost any task that can be completed with a data-defined pattern or set of rules can be automated with machine learning. This allows companies to transform processes that were previously only possible for humans to perform think responding to customer service calls, bookkeeping, and reviewing resumes.

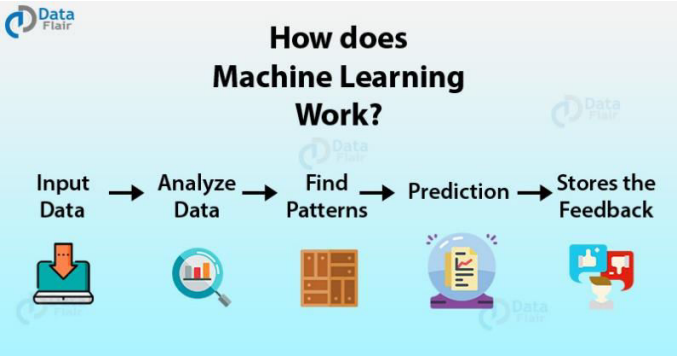


Fig 1.2: How does Machine Learning Work

Machine Learning Techniques:

Machine learning uses two types of techniques:

●**Supervised learning**: which trains a model on known input and output data so that it can predict future outputs. It allows you to collect data or produce a data output from a previous ML deployment. Supervised learning is exciting because it works in much the same way humans actually learn.

●**Unsupervised learning**: which finds hidden patterns or intrinsic structures in input data. helps you find all kinds of unknown patterns in data. In unsupervised learning, the algorithm tries to learn some inherent structure to the data with only unlabeled examples. Two common unsupervised learning tasks are clustering and dimensionality reduction.

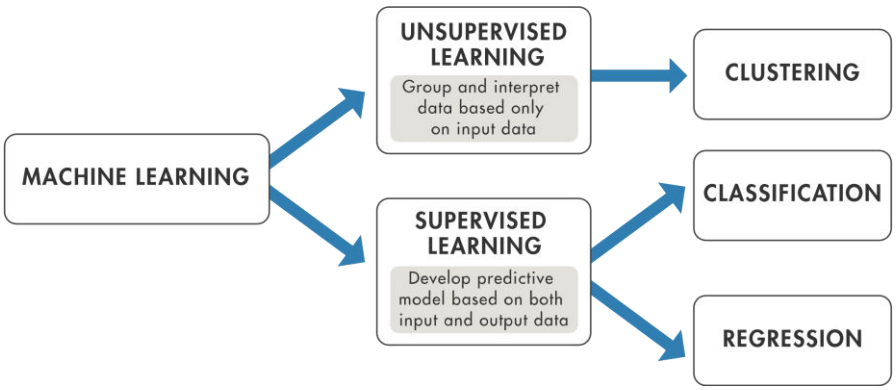


Fig 1.3: Techniques of Machine Learning

This report contains the "**REDWINE QUALITY PREDICTION**" based on the datset which contains the attributes such as fixed acidity, volatile acidity, citric acid, residual sugar,chlorides, free sulphuric acis, total sulphuric acid, density, pH, sulphates, alcohol, quantity.The Prediction is done through various machine learning algorithms such as

- Logistic Regression
- Support Vector Machine
- Decission Tree Classifier
- Random Forest Classifier

By using these algorithms we get a detailed view of the tested results and the trained results of the machine.We can visualize the results using Boxplot, Correlation matrix, jointplot and so on.

Objectives

The objectives of this project are as follows:

1.To experiment with different classification methods to see which yields the highest accuracy

2.To determine which features are the most indicative of a good quality wine

Chapter 2

COMPANY PROFILE

2.1 Company Profile:

The Tech Fortune group was originally incepted in 2013 to focus only on EPC assignment of Infra and Real Estate Projects. Later the Company shifted its focus towards software development, Training ,BPO, Sourcing, Food Business, Health Care and Strategic Advisory Services. The Tech Fortune group Tech Fortune was born in 2013 with an objective to create a landmark initiative by a group of highly qualified technology oriented professionals in the software domain. A Software development Firm head quartered in Vijapur and operating for 5 years in with 3 offices across Karnataka. Since its inception in 2013, Tech fortune group has grown rapidly with the help of its valued Customers, professionals & Business Associates who have been continuously contributing and monitoring the Company's business activities in the operations of Project Management, Education Consultancy, QMS and Six Sigma implementation and many other domain of expertise.

Tech Fortune Technogies, is an emerging technology organization in the fields of business process outsourcing, software development, end-to-end ERP solutions, Artificial Intelligence, Blockchain technology with a focus on providing customized solutions to the various business needs of a diverse global clientele.

2.2 Mission:

Being slow and steady, our mission is to gain the confidence of our clients and by dint of our integrity, innovation and dynamism, deliver their requirements on time with full quality thus bridging the gap between demand and delivery.

2.2 Vission:

With an unyielding focus on integrity and backed by strong founders and management team , Sourcing wants to make a mark in the field of IT services by applying innovation to simplify complex business processes and add value to clients' business.

Chapter 3

LITERATURE SURVEY

3.1 Logistic Regression

Introduction to Logistic Regression: Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts P(Y=1) as a function of X.

It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc. Types of Logistic Regression: Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types –

Binary or Binomial: In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

Multinomial: In such a kind of classification, dependent variable can have 3 or more possible unordered types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

Ordinal: In such a kind of classification, dependent variable can have 3 or more possible ordered types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3.

Advantages:

- It is a widely used technique because it is very efficient, does not require too many computational resources, it’s highly interpretable, it doesn’t require input features to be scaled, it doesn’t require any tuning, it’s easy to regularize, and it outputs well- calibrated predicted probabilities.

- Like linear regression, logistic regression does work better when you remove attributes that are unrelated to the output variable as well as attributes that are very similar (correlated) to each other. Therefore Feature Engineering plays an important role in regards to the performance of Logistic and also Linear Regression.

Another advantage of Logistic Regression is that it is incredibly easy to implement and very efficient to train. I typically start with a Logistic Regression model as a benchmark and try using more complex algorithms .

• Because of its simplicity and the fact that it can be implemented relatively easy and quick, Logistic Regression is also a good baseline that you can use to measure the performance of other more complex Algorithms

Disadvantages:

A disadvantage of it is that we can't solve non-linear problems with logistic regression since it's decision surface is linear.

3.2 Decision Tree Classifier

Introduction to Decision Tree: In general, Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decisions trees are the most powerful algorithms that falls under the category of supervised algorithms. They can be used for both classification and regression tasks. The two main entities of a tree are decision nodes, where the data is split and leaves, where we got outcome. The example of a binary tree for predicting whether a person is fit or unfit providing various information like age, eating habits and exercise habits. We have the following two types of decision trees.

• Classification decision trees − In this kind of decision trees, the decision variable is categorical. The above decision tree is an example of classification decision tree.

• Regression decision trees − In this kind of decision trees, the decision variable is continuous.

Advantages:

1. Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.

2. A decision tree does not require normalization of data.

3. A decision tree does not require scaling of data as well.

4. Missing values in the data also does NOT affect the process of building decision tree to any considerable extent.

5. A Decision trees model is very intuitive and easy to explain to technical teams as well as stakeholders.

Disadvantages:

1. A small change in the data can cause a large change in the structure of the decision tree causing instability.

2. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.

3.3 Random Forest Classifier

Introduction:

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Working of Random Forest Algorithm:

We can understand the working of Random Forest algorithm with the help of following steps −

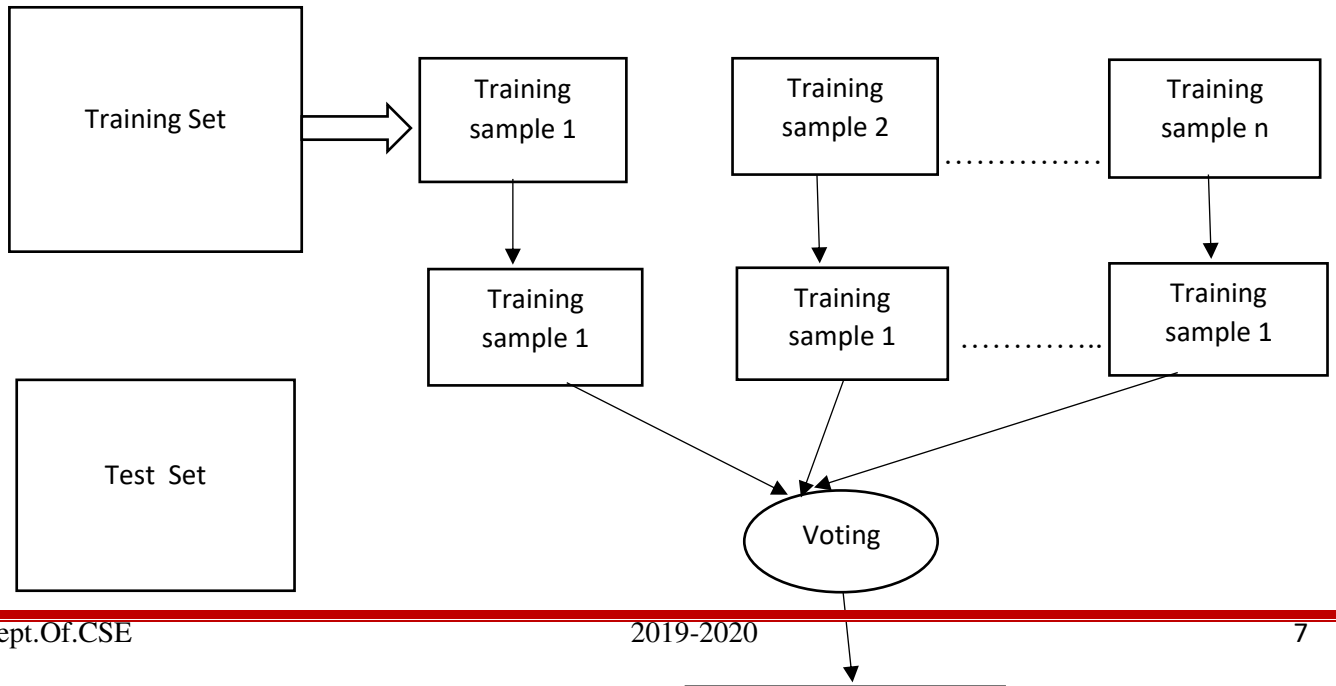
Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

The following diagram will illustrate its working −



Advantages:

The following are the advantages of Random Forest algorithm –

• It overcomes the problem of overfitting by averaging or combining the results of different decision trees.

• Random forests work well for a large range of data items than a single decision tree does.

• Random forest has less variance then single decision tree.

• Random forests are very flexible and possess very high accuracy.

• Scaling of data does not require in random forest algorithm. It maintains good accuracy even after providing data without scaling.

• Random Forest algorithms maintains good accuracy even a large proportion of the data is missing.

Disadvantages:

The following are the disadvantages of Random Forest algorithm –

• Complexity is the main disadvantage of Random forest algorithms.

• Construction of Random forests are much harder and time-consuming than decision trees.

• More computational resources are required to implement Random Forest algorithm.

• It is less intuitive in case when we have a large collection of decision trees.

• The prediction process using random forests is very time-consuming in comparison with other algorithms.

3.4. K-NearestNeighbours Classifier

Introduction:

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well –

• Lazy learning algorithm − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.

• Non-parametric learning algorithm − KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data. Working of KNN Algorithm: K-nearest neighbors (KNN) algorithm uses 'feature similarity' to predict the values of new datapoints which further means that the new

data point will be assigned a value based on how closely it matches the points in the training set.

We can understand its working with the help of following steps –

Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following –

• 3.1 − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.

• 3.2 − Now, based on the distance value, sort them in ascending order.

• 3.3 − Next, it will choose the top K rows from the sorted array.

• 3.4 − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 − End

Advantages:

• It is very simple algorithm to understand and interpret.

• It is very useful for nonlinear data because there is no assumption about data in this algorithm.

• It is a versatile algorithm as we can use it for classification as well as regression.

• It has relatively high accuracy but there are much better supervised learning models than KNN

Disadvantages:

• It is computationally a bit expensive algorithm because it stores all the training data.

• High memory storage required as compared to other supervised learning algorithms.

• Prediction is slow in case of big N.

• It is very sensitive to the scale of data as well as irrelevant features.

5. Support Vector Machine(SVM)

Introduction to SVM:

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

Working of SVM:

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

The followings are important concepts in SVM –

• Support Vectors − Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.

• Hyperplane − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

• Margin − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

The main goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH) and it can be done in the following two steps −

• First, SVM will generate hyperplanes iteratively that segregates the classes in best way.

• Then, it will choose the hyperplane that separates the classes correctly.

Advantages of SVM classifiers:

• SVM classifiers offers great accuracy and work well with high dimensional space. SVM classifiers basically use a subset of training points hence in result uses very less memory.

Disadvantages of SVM classifiers:

• They have high training time hence in practice not suitable for large datasets. Another disadvantage is that SVM classifiers do not work well with overlapping classes.

Chapter 4

PROBLEM STATEMENT

4.1 Problem Definition:

The red wine dataset contains different chemical information about red wine. It has 1599 instances with 12 variables each. The dataset is good for classification and regression tasks. The model can be used to predict red wine quality. Perform various different algorithms like regression, decision tree, random forests, etc and differentiate between the models and analyse their performances.

Here I am Performing various different Classification algorithms like Logistics Regression, Decision Tree Classifier, Random Forest Classifier, Stochastic Gradient Descent Classifier, Naive Bayes Classifier, K-NearestNeighbours Classifier and Support Vector Machine(SVM) and trying to differentiate between the models and analyse their performances.

Chapter 5

DATA SUMMARY

5.1 Attribute Information:

The quality of a wine is determined by 11 input variables:

1. Fixed acidity
2. Volatile acidity
3. Citric acid
4. Residual sugar
5. Chlorides
6. Free sulfur dioxide
7. Total sulfur dioxide
8. Density
9. pH
10. Sulfates
11. Alcohol

Output variable: quality (score between 0 and 10)

Missing Attribute Values: None

5.2 Libraries Imported:

import pandas as pd - pandas is a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a NumPy matrix array. This makes pandas a trusted ally in data science and machine learning. Similar to NumPy, pandas deals primarily with data in 1-D and 2-D arrays; however, pandas handles the two differently.

import matplotlib.pyplot as plt - matplotlib.pyplot is stateful, in that it keeps track of the current figure and plotting area, and the plotting functions are directed to the current axes and can be imported using import matplotlib.pyplot as plt.

import seaborn as sns - Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

import numpy as np - Numpy provides a large set of numeric datatypes that you can use to construct arrays. Numpy tries to guess a datatype when you create an array, but functions that construct arrays usually also include an optional argument to explicitly specify the datatype.

%matplotlib inline - %matplotlib inline sets the backend of matplotlib to the 'inline' backend: With this backend, the output of plotting commands is displayed inline within frontends like the Jupyter notebook, directly below the code cell that produced it.

from sklearn.linear_model import LogisticRegression- This class implements regularized logistic regression using the 'liblinear' library, from sklearn.datasets

from sklearn.model_selection import train_test_split- train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train_test_split will make random partitions for the two subsets.

Chapter 6

IMPLEMENTATION

6.1 Source Code:

[1]:
```
# import libraries
import pandas as pd
import numpy as np
import sklearn
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
%matplotlib inline
```

[2]:
```
data = pd.read_csv("redwine.csv")
```

[3]:
```
data.head()
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	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 [4]:
```
data.columns
```

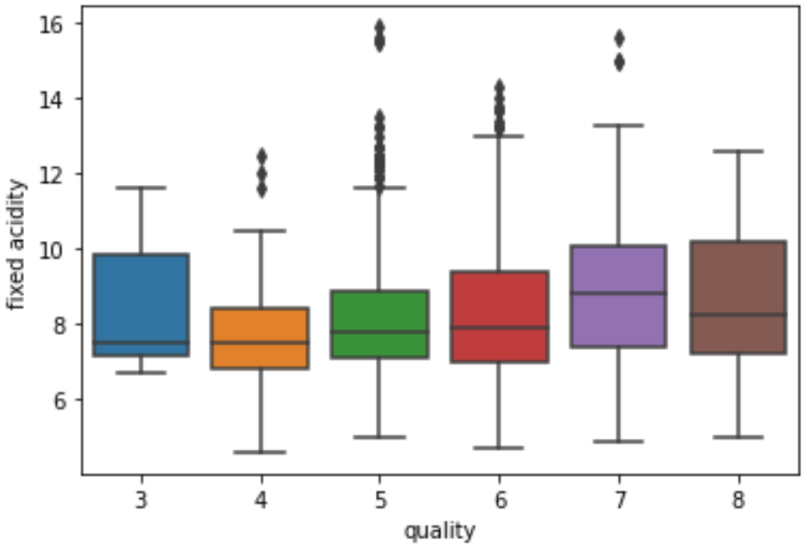
Out[4]:
```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

```
In [5]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

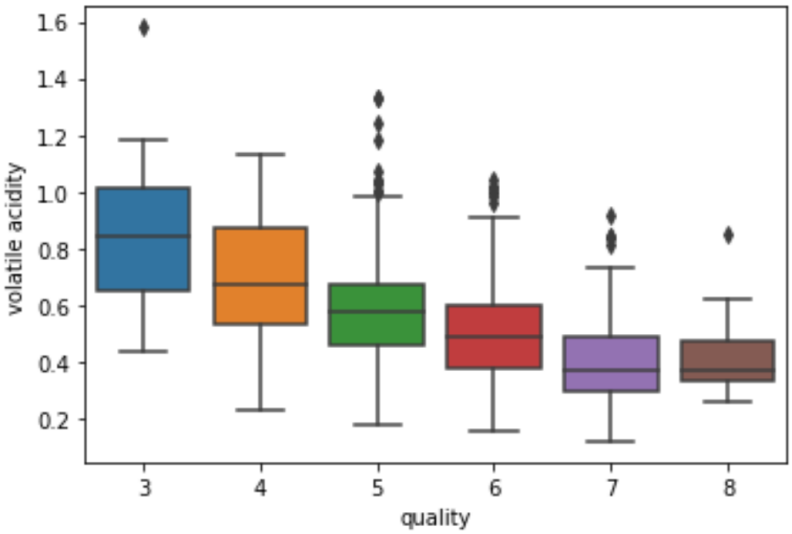
```
In [6]: sns.boxplot('quality', 'fixed acidity', data = data)
```

```
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x231d08eb550>
```



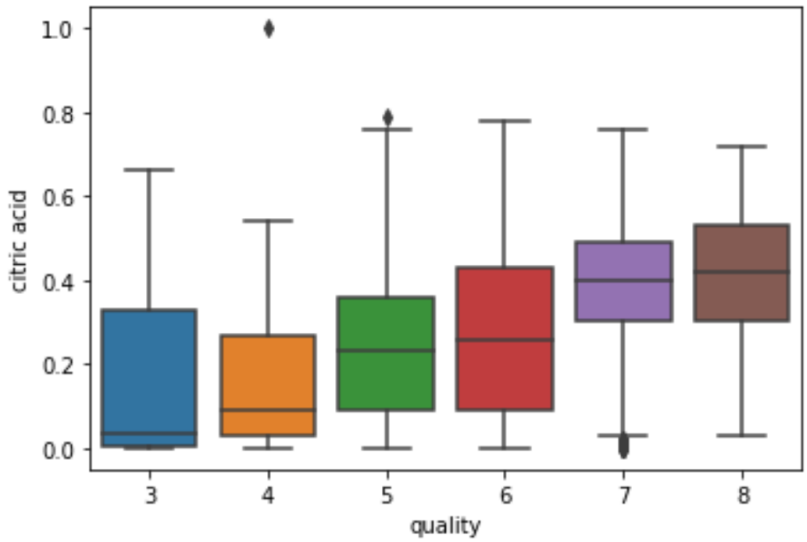
Visualization: Here we see that fixed acidity does not give any specification to classify the quality.

```
In [7]: sns.boxplot('quality', 'volatile acidity', data = data)
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x231d1024c40>
```



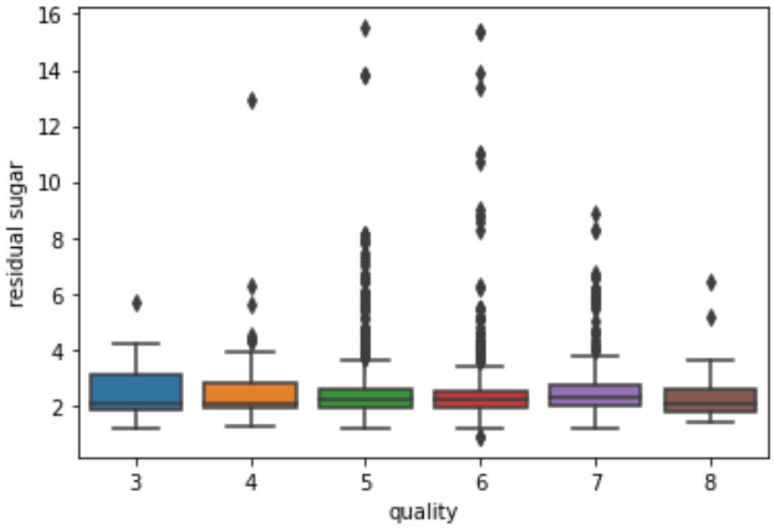
Visualization: Here we see that its quite a downing trend in the volatile acidity as we go higher the quality.

```
In [8]: sns.boxplot('quality', 'citric acid', data = data)
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x231d11792e0>
```



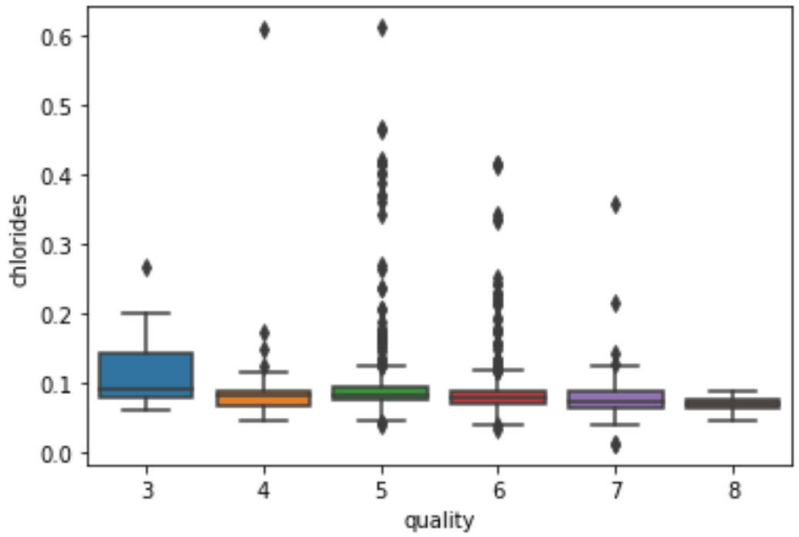
Visualization: Composition of citric acid goes higher as we go higher in the quality of the wine.

```
In [9]: sns.boxplot('quality', 'residual sugar', data = data)
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x231d11791c0>
```



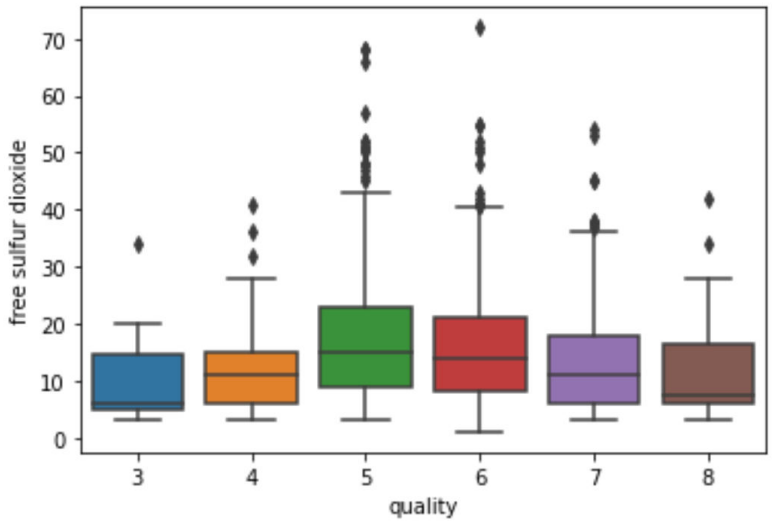
Visualization: Composition of residual sugar is uniformly distributed over the different quality level.

```
In [10]: sns.boxplot('quality', 'chlorides', data = data)
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x231d13025b0>
```



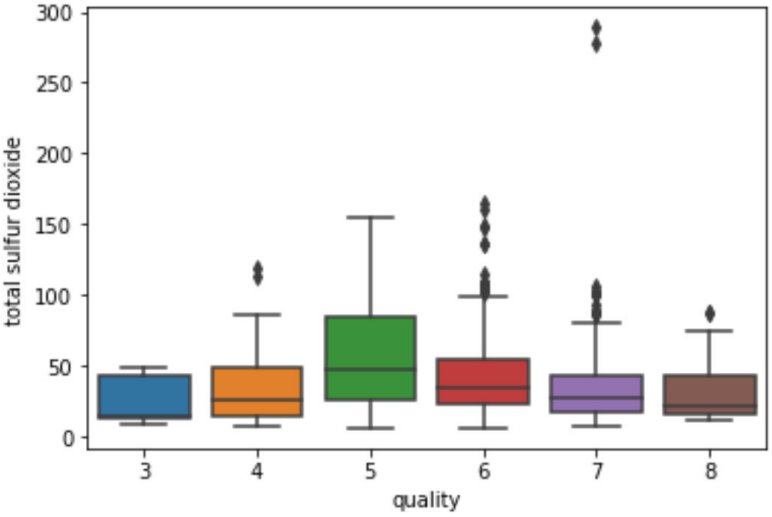
Visualization: Composition of chlorides also goes down as we go higher in the quality of the wine.

```
In [11]: sns.boxplot('quality', 'free sulfur dioxide', data = data)
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x231d13bf400>
```



Visualization: Composition of free sulphur dioxide somewhat increases a bit in the quality levels of 5 and 6 and then again decreases.

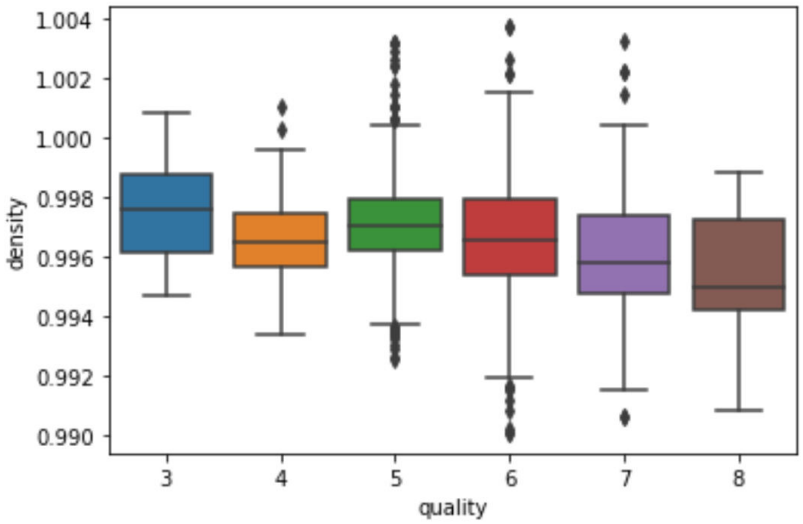
```
In [12]: sns.boxplot('quality', 'total sulfur dioxide', data = data)
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x231d1474820>
```



Visualization: Composition of total sulphur dioxide lowers towards the high quality levels with an sudden increase in the quality level of 5.

```
In [13]: sns.boxplot('quality', 'density', data = data)
```

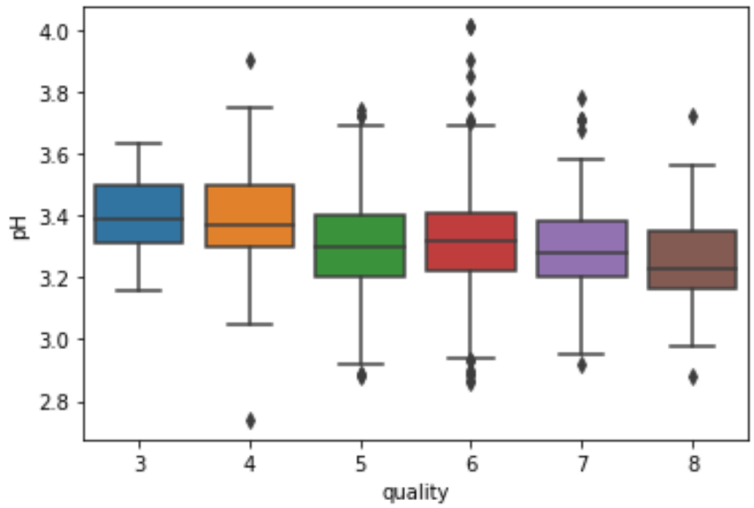
```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x231d1523280>
```



Visualization: Density of the red wine seems to be larger at the low quality levels and tends to decrease towards the high quality levels.

```
In [14]: sns.boxplot('quality', 'pH', data = data)
```

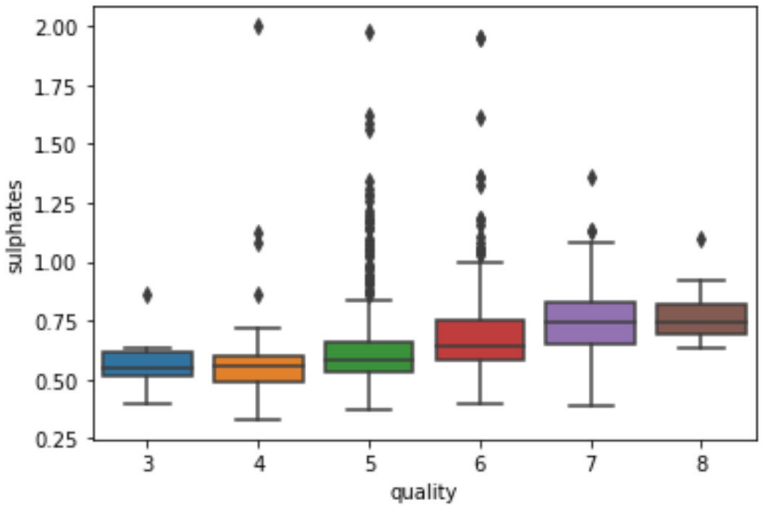
```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x231d15fbfd0>
```



Visualization: The pH levels of the red wine seems to be uniformly distributed with the quality levels with the little increase in the pH at the quality level 3 and 4.

```
In [15]: sns.boxplot('quality', 'sulphates', data = data)
```

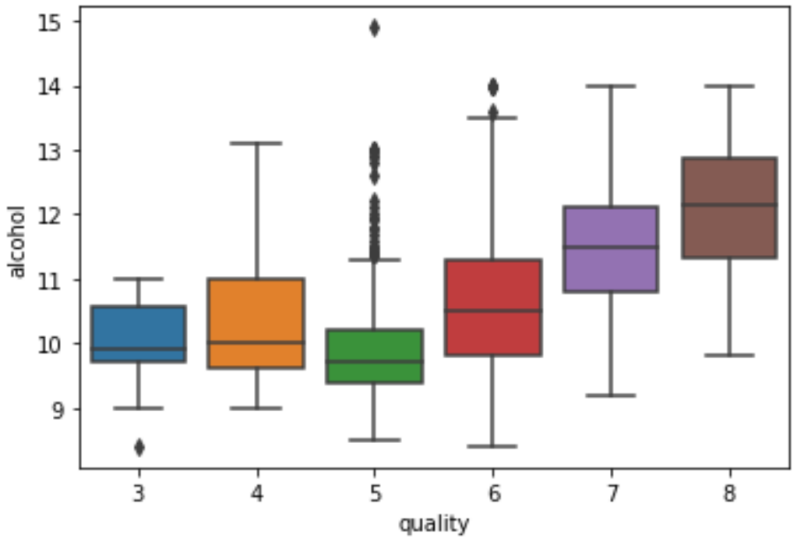
```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x231d16ab460>
```



Visualization: Sulphates level goes higher with the quality of wine.

```
In [16]: sns.boxplot('quality', 'alcohol', data = data)
```

```
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x231d140f6d0>
```



Visualization: Alcohol level also goes higher as the quality of wine increases.

```
In [17]: data.describe()
```

Out[17]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
count	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
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000

```
In [24]: bins = (2, 6.5, 8)
         group_names = ['bad', 'good']
         data['quality'] = pd.cut(data['quality'], bins = bins, labels = group_names)
```

```
In [28]: label_quality = LabelEncoder()
```

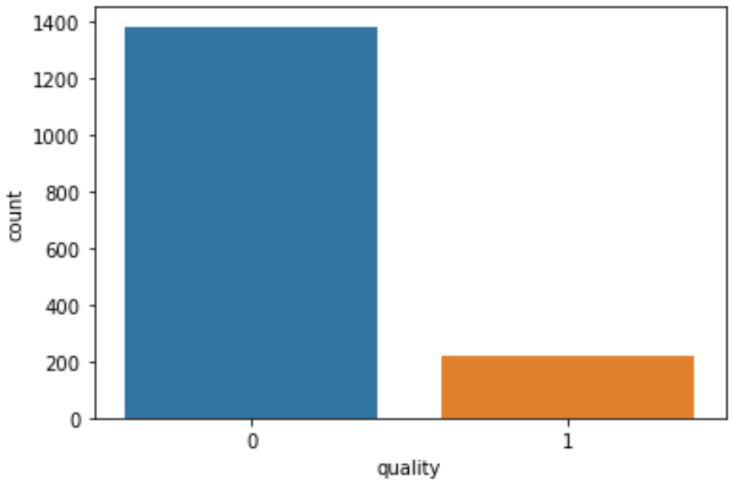
```
In [31]: data['quality'] = label_quality.fit_transform(data['quality'])
```

```
In [33]: data['quality'].value_counts()
```

```
Out[33]: 0    1382
         1     217
         Name: quality, dtype: int64
```

```
In [35]: sns.countplot(data['quality'])
```

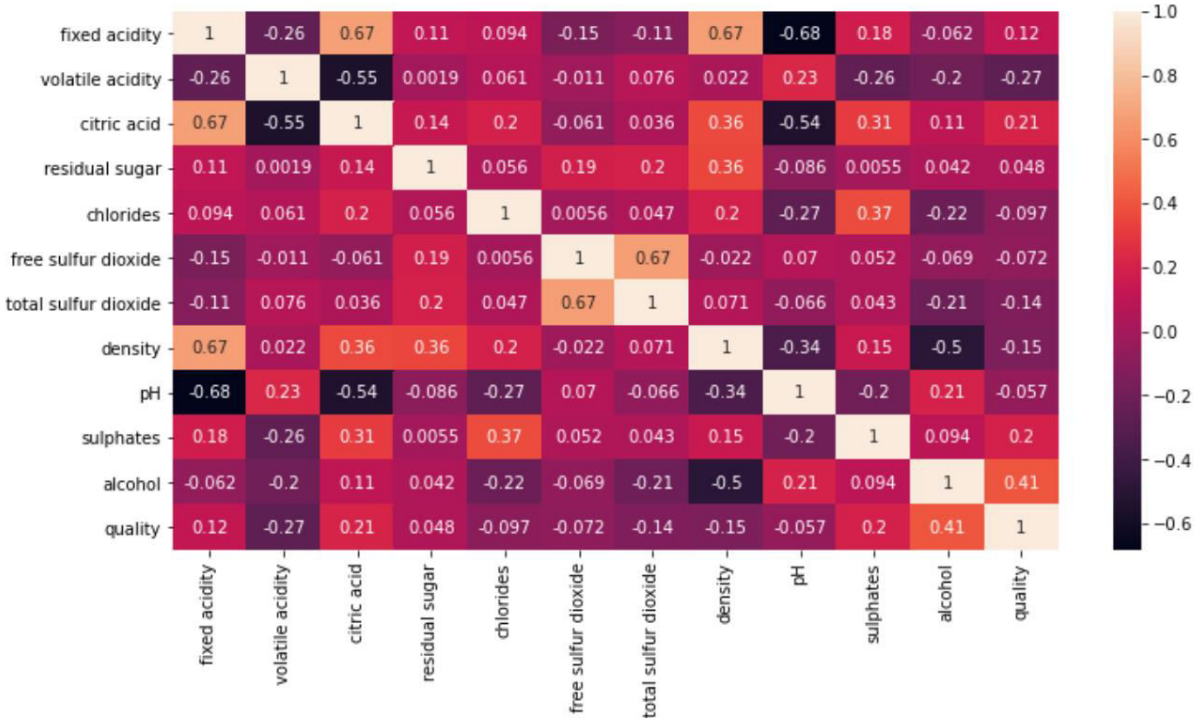
```
Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x231d1ee6820>
```



Visualization: From the above countplot we can see that there are large no. of wines with good quality=0 means whose quality levels are less than 7.

In [38]:
```
plt.figure(figsize=(12,6))
sns.heatmap(data.corr(),annot=True)
```

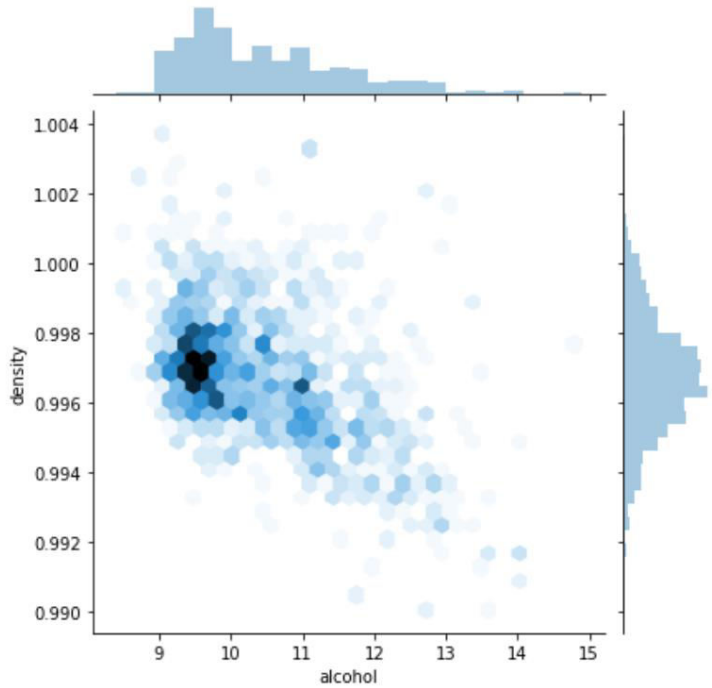
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x231d1b2de20>



In [40]:
```
plt.figure(figsize=(12,6))
sns.jointplot(y=data["density"],x=data["alcohol"],kind="hex")
```

Out[40]: <seaborn.axisgrid.JointGrid at 0x231d2325460>

<Figure size 864x432 with 0 Axes>



In [44]:
```
X = data.drop('quality', axis = 1)
y = data['quality']
```

In [45]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

In [46]:
```
sc = StandardScaler()
```

In [47]:
```
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

In [58]:
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
rf_predict=rf.predict(X_test)
```

In [59]:
```
rf_conf_matrix = confusion_matrix(y_test, rf_predict)
rf_acc_score = accuracy_score(y_test, rf_predict)
print(rf_conf_matrix)
print(rf_acc_score*100)
```

```
[[264   9]
 [ 28  19]]
88.4375
```

In [53]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score
lr = LogisticRegression()
lr.fit(X_train, y_train)
lr_predict = lr.predict(X_test)
```

In [54]:
```
lr_conf_matrix = confusion_matrix(y_test, lr_predict)
lr_acc_score = accuracy_score(y_test, lr_predict)
print(lr_conf_matrix)
print(lr_acc_score*100)
```

```
[[268   5]
 [ 35  12]]
87.5
```

In [55]:
```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(X_train,y_train)
dt_predict = dt.predict(X_test)
```

In [56]:
```
dt_conf_matrix = confusion_matrix(y_test, dt_predict)
dt_acc_score = accuracy_score(y_test, dt_predict)
print(dt_conf_matrix)
print(dt_acc_score*100)
```

```
[[247  26]
 [ 24  23]]
84.375
```

In [65]:
```
from sklearn.svm import SVC
svc = SVC()
svc.fit(X_train,y_train)
pred_svc =svc.predict(X_test)
```

In [66]:
```
from sklearn.metrics import classification_report,accuracy_score
print(classification_report(y_test,pred_svc))
```

```
              precision    recall  f1-score   support

           0       0.88      0.98      0.93       273
           1       0.71      0.26      0.37        47

    accuracy                           0.88       320
   macro avg       0.80      0.62      0.65       320
weighted avg       0.86      0.88      0.85       320
```

In [63]:
```
lin_svc_conf_matrix = confusion_matrix(y_test, rf_predict)
lin_svc_acc_score = accuracy_score(y_test, rf_predict)
print(lin_svc_conf_matrix)
print(lin_svc_acc_score*100)
```

```
[[264   9]
 [ 28  19]]
88.4375
```

In [68]:
```
conclusion = pd.DataFrame({'models': ["Random Forest","Logistic Regression","Decission Tree","Supprot vector machine"
'accuracies': [accuracy_score(y_test, rf_predict),accuracy_score(y_test, lr_predict),accuracy_score(y_test, dt_predict),accuracy_
conclusion
```

Out[68]:

	models	accuracies
0	Random Forest	0.884375
1	Logistic Regression	0.875000
2	Decission Tree	0.843750
3	Supprot vector machine	0.875000

Chapter 7

CONCLUSION

I have understood the various equipment's working principle, company details, specifications and distributors details. The whole internship period has motivated me to design a system, component, or process to meet desired needs with realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. It made me understand the function on multidisciplinary teams and inspired me to create a novel system to solve engineering problems. It has made me Understand professional and ethical responsibility and to Communicate effectively. I have obtained the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context.

From All the Classification Algorithms we can see that the **Random Forest Classifier** algorithm which yields heighest Accuracy of **88.5 %.** Along with Random Forest Classifier, SVM, Decision tree and Logistic Regression also gives a good Accuracy of 87%.

DECLARATION

We the students of 7th semester BE, Computer Science and Engineering hereby declare that project entitled "RedWine Quality Prediction" has been carried out by us at City Engineering College, Bengaluru and submitted in partial fulfilment of the cource requirement for the award of the degree of **Bachelor of Engineerirng in computer Science and Engineering of Visvesvaraya Technological University,Belgaum,** during the academic year 2019-2020.

I also declare that, to the best of the knowledge and belief ,the work reported here does not form the part of dissertation on the basis of which a degree or award was conferred on a earlier occasion on this by any other student.

Date :

Place: Bangalore

MEGHANA G

(1CE17CS059)