A Mini-Project Report On

"Wine Quality Testing"

Submitted By

Sujeet Mandal

PRN No: 1132220108

Sejal Sawant

PRN No: 1132220689

Bhushan Kadhane

PRN No: 113221010

F.Y. M.Sc. (Data Science and Big Data Analytics)

School of Computer Science and Engineering Department of computer science and application

MIT – World Peace University

Pune – 411038

Academic Year 2022-2024

APRIL – 2023

Dr. Vishwanath Karad MIT WORLD PEACE UNIVERSITY, PUNE

SCHOOL OF COMPUTER SCIENCE

Certificate

This is to certify that
Sujeet Mandal Prn No: 1132220108,

Of **M.Sc. (Data Science and Big Data Analytics)** successfully completed his Mini-Project in

Machine Learning

"Wine Quality Testing"

to our satisfaction and submitted the same during the academic year 2022- 2024

towards the partial fulfillment of degree of Master **of Science in Data Science and Big**

Data Analytics of Dr. Vishwanath Karad MIT WORLD PEACE UNIVERSITY, PUNE

SCHOOL OF COMPUTER SCIENCE.

Dr. Shubhalaxmi Joshi	Prof. Surabhi Thatte	Prof. Sachin Bhoite
Associate Dean	Program Head	Professor
Faculty of Science	School of Computer Science	School of Computer Science
MITWPU	MITWPU	MITWPU

Acknowledgement

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project guide Prof. Sachin Bhoite for the guidance, inspiration and constructive suggestions that help us in the preparation of this project. We also thank our colleagues who have helped in successful completion of the project.

Sujeet Mandal

Prn No: 1132220108

Table of Contents

1. INTRODUCTION

1.1 Domain of the problem statement

Testing of wine is one of the major tasks today , as the taste of the wine depends on many various factors which are still unknow to people . Each and every factor affects the taste and quality of the wine. Our project focuses on predicting quality of wine based on these various features.

In this mini project we will be predicting the quality of wine .

1.2 Motivation

Winemakers need to ensure that their wines meet the desired quality standards, comply with regulatory requirements, and are safe for consumption. However, there are various factors that can affect the quality and safety of the wine, such as grape variety, fermentation conditions, storage conditions, and processing techniques. Therefore, winemakers need to conduct thorough testing and analysis of their wines to identify any potential issues and optimize their production processes. The challenge is to select the appropriate tests and analytical methods that can provide accurate and reliable results, as well as to interpret and apply the data effectively to improve the quality, safety, and marketability of the wine.

1.3 Problem statement

To explore wine testing prediction and classification models for the Winemakers so that their wine quality is up to the mark and so that it helps the industry to optimize meet the supply and demand.

2 LITERATURE SURVEY

Sr.No.	Paper Title	Publication Year	Author's Name	Outcome/ Accuracy	Advantages	Limitations
1.	Prediction of red wine quality using 1-D Convolutional Neural Network	18 Jan 2023	Yang Yang, Shengnan Di	KNN-0.768 SVM-0.765 LR-0.755 RF-0.810 1D CNN-0.832 DNN-D-0.812 DNN-0.825	we design 1D-CNN networks for the task of wine quality prediction. The 1D-CNN block can process adjacent features in one convolutional step. In addition, CNN requires fewer parameters and is less prone to overfitting problems even when the data is relatively small. We further design a Dropout block that includes 3 dropout layers and batch normalization layers to improve the robustness of the model and eliminate overfitting.	it may need to explore more interaction between the features from a physicochemical perspective. For example, certain physical properties may be evolving along with the changes in multiple chemical compounds. In addition, there is also interconversion within chemical components. It remains an open problem to analyze the effect of different external variables, such as temperature, and light, on the quality of the wine.

2.	**A Study and Analysis of Machine Learning Techniques in Predicting Wine Quality**	May 30, 2021.	Mohit Gupta, Vanmathi C	SVM-0.6234 KNN-0.6139 Random Forestt-0.7325 J48-0.56 CART-0.7075	The dataset of both red and white wine is composed of 11 physicochemical properties. This work deduces that the classification method should provide space for corrective steps to be taken during production to enhance the quality of the wine.	In the future, broad data set may be used for experiments and other machine learning techniques may be explored for prediction of wine quality, and we will expand this analysis to include feature development methods to test whether or not the model's predictive power may be increased.

3. SOLUTION DESIGN

3.1 Solution Approach

We have used Wine Quality data from Kaggle. This dataset contains more than 1500 records. First, we have performed EDA on all dataset to understand the data if it contains any null value, missing value, any outliers, etc. After performing EDA we got the basic understanding of our dataset. Based on our observations we have considered some columns and those which were highly correlated were removed by performing label encoding. After performing all these task we have implemented certain machine learning algorithm to know the accuracy and precision of our data.

3.2 Technology Stack

We have used Jupyter notebook platform
Language:- python
Libraries:-numpy
Panda
matplotlib.pyplot
seaborn
Sklearn-EDA, pre-processing as well as for model building.

3.3 Designing model

On our Wine Quality dataset, we have applied Logistic regression, Random forest ,Gaussian Naive Bayes and Ensemble model. Out of these algorithms Ensemble classifier & Logistic Regression gives highest accuracy.

Our goal is to classify the types of wine and their quality depending on the other factors such as alcohol quantity, fixed acidity, volatile acidity, determination of density, pH etc.

4. SOLUTION IMPLEMENTATION AND RESULT

4.1 Obtaining Data

The information gathered is from Kaggle. It produces several types of wines. It's a known fact that the older the wine, the better the taste. However, there are several factors other than age that go into wine quality certification which include physiochemical tests like alcohol quantity, fixed acidity, volatile acidity, determination of density, pH, and more.

We have collected data from online source i.e

https://www.kaggle.com/

Total Observation: 1599-Rows,13-Columns

The above data is collected from a smart small-scale wine producing industry.

4.2 EDA

Using python and Jupyter notebook we have done EDA on Wine Quality Testing dataset. We described the dataset and based on that performed further processing. Also plotted heat map to see correlation between the data points so that we could proceed for feature selection. Also plotted the distribution

Data reading using pandas frame

In [4]:
```
rwine_data=pd.read_csv('C:/Users/USER/Pictures/data/winequality_red.csv')
rwine_data.shape #no.of rows and col
print(rwine_data['wine_type'])
```

```
0       12
1       12
2       12
3       12
4       12
        ..
1594    12
1595    12
1596    12
1597    12
1598    12
Name: wine_type, Length: 1599, dtype: int64
```

In [5]:
```
wwine_data=pd.read_csv('C:/Users/USER/Pictures/data/winequality_white.csv')
wwine_data=wwine_data[0:1599] #no.of rows and col
print(wwine_data.shape)
```

```
(1599, 13)
```

APPENDING BOTH DATASETS IN ONE FRAME

In [6]:
```
frames = [rwine_data,wwine_data]
result = pd.concat(frames)
```

In [7]:
```
result.shape
```

Out[7]:
```
(3198, 13)
```

In [8]:
```
print(result)
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0               7.4              0.70         0.00             1.9      0.076
1               7.8              0.88         0.00             2.6      0.098
2               7.8              0.76         0.04             2.3      0.092
3              11.2              0.28         0.56             1.9      0.075
4               7.4              0.70         0.00             1.9      0.076
...             ...               ...          ...             ...        ...
1594            7.0              0.29         0.49             3.8      0.047
1595            6.4              0.27         0.49             7.3      0.046
1596            6.6              0.55         0.01             2.7      0.034
1597            6.4              0.27         0.49             7.3      0.046
1598            6.3              0.24         0.74             1.4      0.172

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                    11.0                  34.0   0.9978  3.51       0.56
1                    25.0                  67.0   0.9968  3.20       0.68
2                    15.0                  54.0   0.9970  3.26       0.65
3                    17.0                  60.0   0.9980  3.16       0.58
4                    11.0                  34.0   0.9978  3.51       0.56
...                   ...                   ...      ...   ...        ...
1594                 37.0                 136.0   0.9938  2.95       0.40
1595                 53.0                 206.0   0.9956  3.24       0.43
1596                 56.0                 122.0   0.9906  3.15       0.30
1597                 53.0                 206.0   0.9956  3.24       0.43
1598                 24.0                 108.0   0.9932  3.27       0.39

      alcohol  wine_type  quality
0         9.4         12        5
1         9.8         12        5
2         9.8         12        5
3         9.8         12        6
4         9.4         12        5
...       ...        ...      ...
1594      9.4         11        6
1595      9.2         11        6
1596     11.9         11        5
1597      9.2         11        6
1598      9.9         11        6

[3198 rows x 13 columns]
```

Data Understanding

CHECKING FOR NULL VALUES,DATA TYPES AND UNIQUES VALUES

In [116]:
```
print(result.shape)
result.isnull().sum() #checking missing values
```

```
(3198, 13)
```

Out[116]:
```
volatile acidity         0
citric acid              0
chlorides                0
density                  0
pH                       0
sulphates                0
alcohol                  0
wine_type                0
quality                  0
fixed_acidity            0
residual_sugar           0
free_sulphur_di_oxide    0
Sulphur_di_oxide         0
dtype: int64
```

In [10]:
```
result.nunique()
```

Out[10]:
```
fixed acidity            98
volatile acidity        171
citric acid              84
residual sugar          228
chlorides               188
free sulfur dioxide      99
total sulfur dioxide    258
density                 501
pH                       98
sulphates               108
alcohol                  69
wine_type                 2
quality                   7
dtype: int64
```

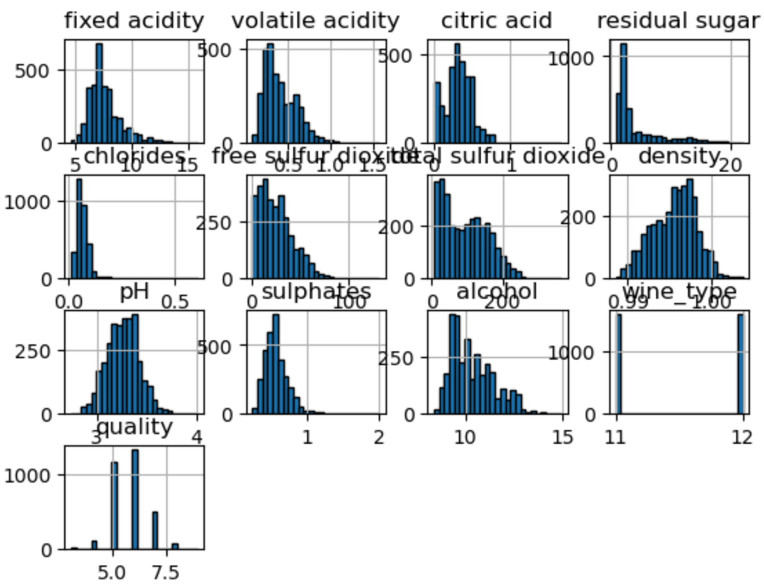
In [11]:
```
result.dtypes
```

Out[11]:
```
fixed acidity           float64
volatile acidity        float64
citric acid             float64
residual sugar          float64
chlorides               float64
free sulfur dioxide     float64
total sulfur dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
wine_type                 int64
quality                   int64
dtype: object
```

plotting histogram to see the distribution of each column

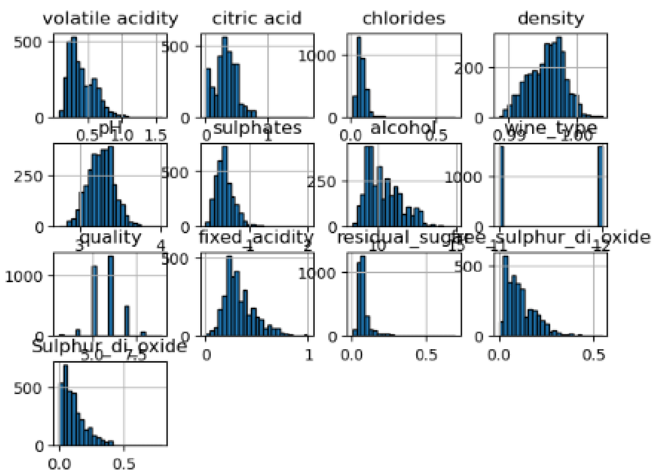
In [12]:
```
result.hist(edgecolor='black',bins=25)
#Pandas.DataFrame.hist() function is useful in understanding the distribution of numeric variables
```

Out[12]:
```
array([[<Axes: title={'center': 'fixed acidity'}>,
        <Axes: title={'center': 'volatile acidity'}>,
        <Axes: title={'center': 'citric acid'}>,
        <Axes: title={'center': 'residual sugar'}>],
       [<Axes: title={'center': 'chlorides'}>,
        <Axes: title={'center': 'free sulfur dioxide'}>,
        <Axes: title={'center': 'total sulfur dioxide'}>,
        <Axes: title={'center': 'density'}>],
       [<Axes: title={'center': 'pH'}>,
        <Axes: title={'center': 'sulphates'}>,
        <Axes: title={'center': 'alcohol'}>,
        <Axes: title={'center': 'wine_type'}>],
       [<Axes: title={'center': 'quality'}>, <Axes: >, <Axes: >,
        <Axes: >]], dtype=object)
```



```
In [21]: result.hist(edgecolor='black',bins=25) #updated histogram
```

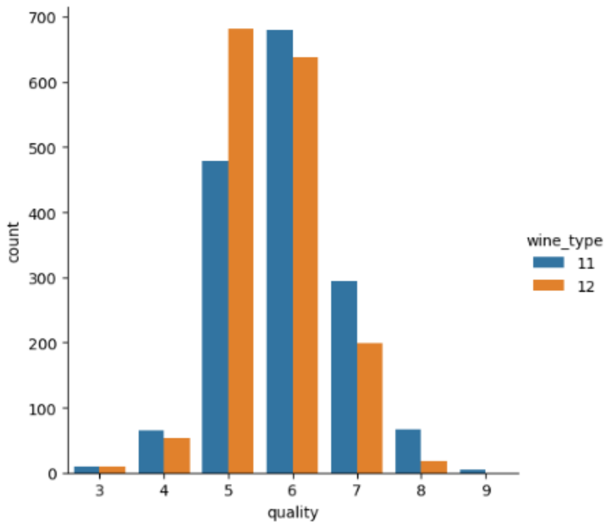
```
Out[21]: array([[<Axes: title={'center': 'volatile acidity'}>,
          <Axes: title={'center': 'citric acid'}>,
          <Axes: title={'center': 'chlorides'}>,
          <Axes: title={'center': 'density'}>],
         [<Axes: title={'center': 'pH'}>,
          <Axes: title={'center': 'sulphates'}>,
          <Axes: title={'center': 'alcohol'}>,
          <Axes: title={'center': 'wine_type'}>],
         [<Axes: title={'center': 'quality'}>,
          <Axes: title={'center': 'fixed_acidity'}>,
          <Axes: title={'center': 'residual_sugar'}>,
          <Axes: title={'center': 'free_sulphur_di_oxide'}>],
         [<Axes: title={'center': 'Sulphur_di_oxide'}>, <Axes: >, <Axes: >,
          <Axes: >]], dtype=object)
```



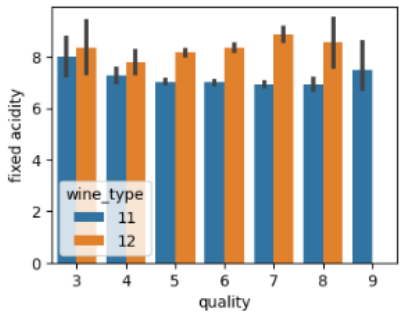
DATA ANALYSIS & Visualize

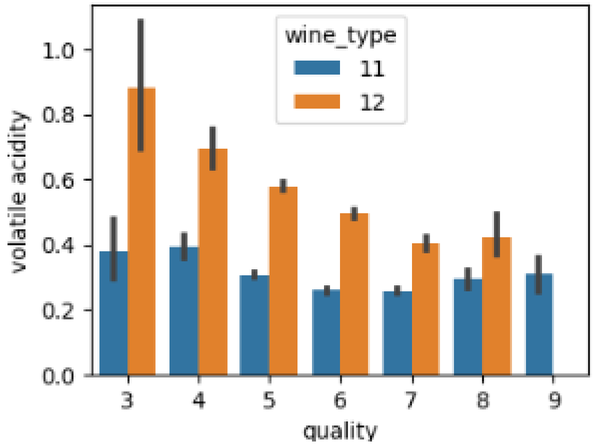
```
In [26]: #Quality check
         ss.catplot(x='quality',data=result,kind='count',hue='wine_type')
         #catplot shows frequencies (or optionally fractions or percents) of the categories of one, two or three categorical variables)
```

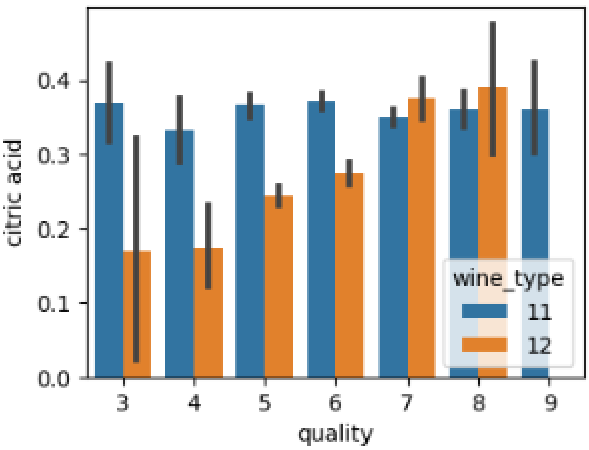
```
Out[26]: <seaborn.axisgrid.FacetGrid at 0x1f329a17130>
```

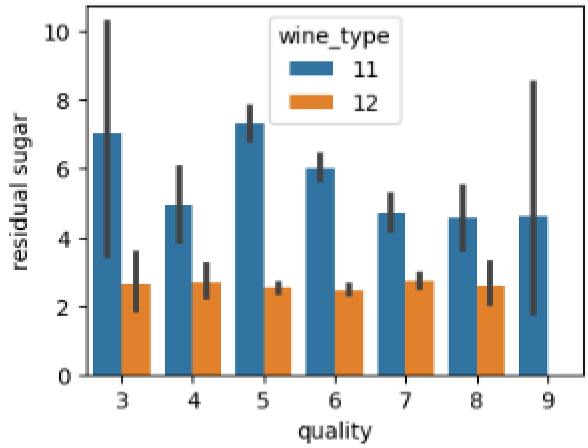


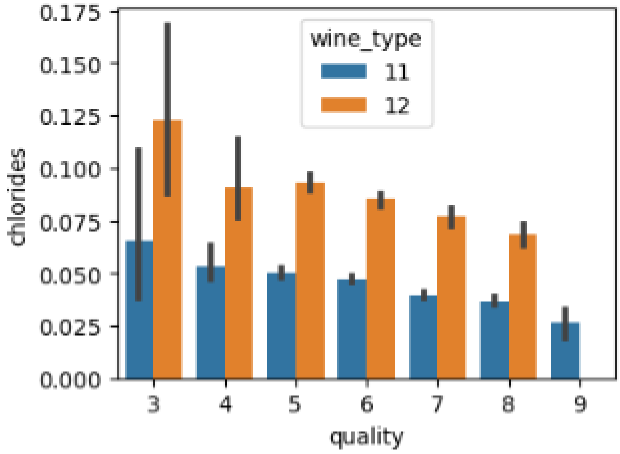
In [27]:
```python
#comparison of columns with quality
arr = np.array(["fixed acidity","volatile acidity","citric acid","residual sugar",
               "chlorides","free sulfur dioxide","total sulfur dioxide","density","pH","sulphates","alcohol"])
for i in arr:
    plot=plt.figure(figsize=(4,3))
    ss.barplot(x="quality",y=i,data=result,hue='wine_type')
```

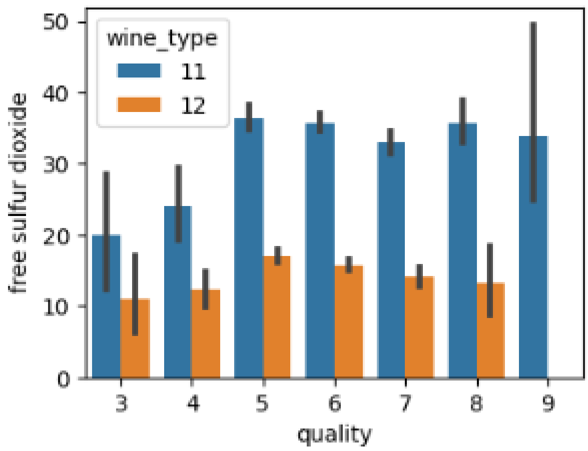


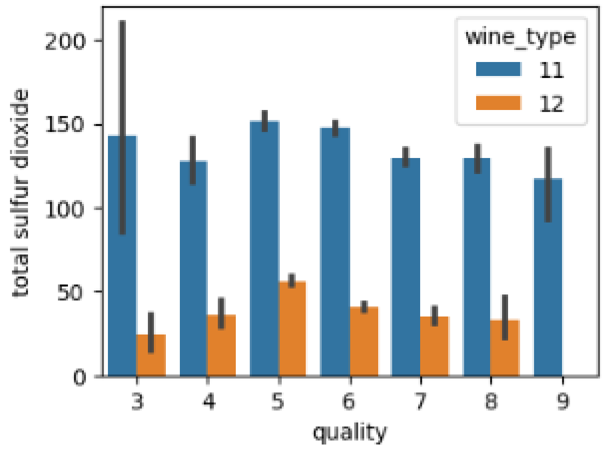


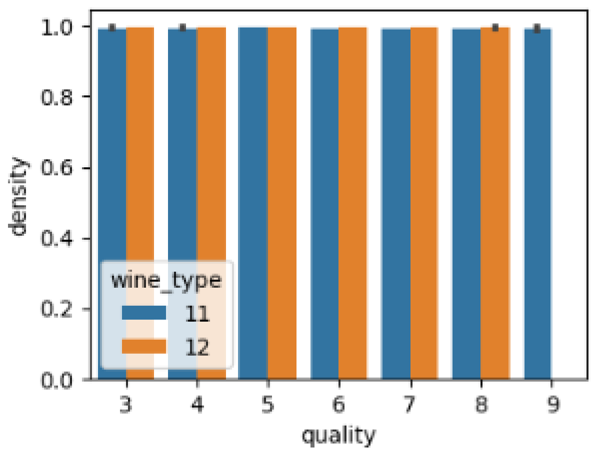


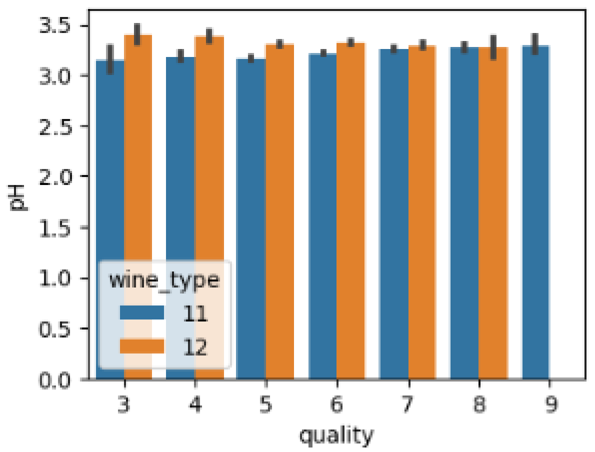

wine_type
11
12
residual sugar
0
2
4
6
8
10
3
4
5
6
7
8
9
quality

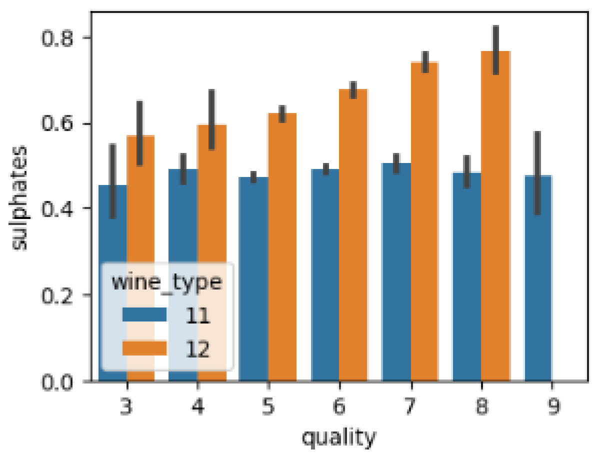

wine_type
11
12
chlorides
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
3
4
5
6
7
8
9
quality

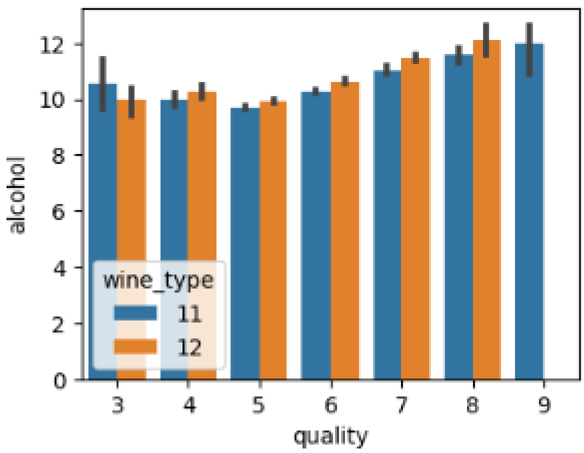

wine_type
11
12
free sulfur dioxide
0
10
20
30
40
50
3
4
5
6
7
8
9
quality


wine_type
11
12
total sulfur dioxide
0
50
100
150
200
3
4
5
6
7
8
9
quality


wine_type
11
12
density
0.0
0.2
0.4
0.6
0.8
1.0
3
4
5
6
7
8
9
quality


wine_type
11
12
pH
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
3
4
5
6
7
8
9
quality

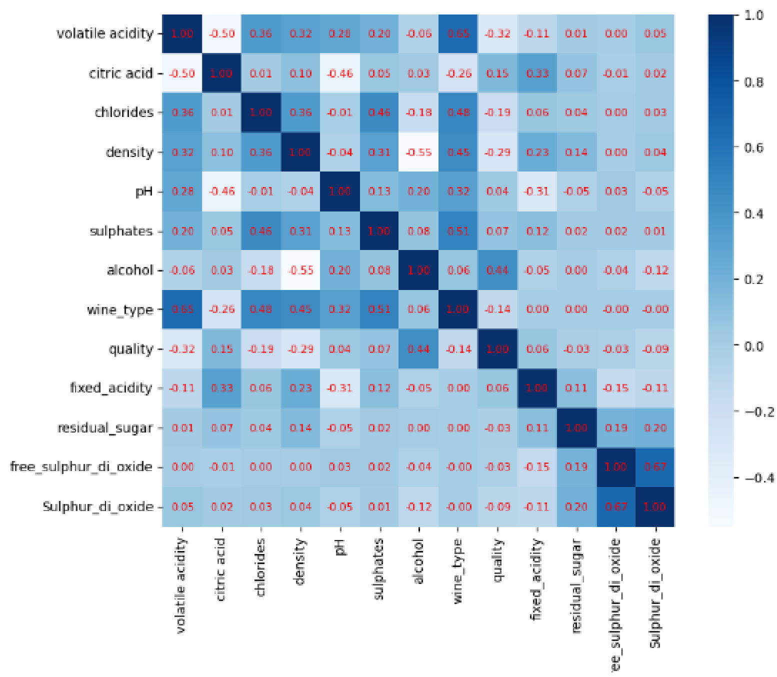




In [28]:

```
###CORRELATION with quality it is usually presented in heatmap
correlation=result.corr()
plot2=plt.figure(figsize=(9,7))
ss.heatmap(correlation,cbar=True,square=True,fmt='.2f',annot=True,annot_kws={'size':8,'color':'red'},cmap='Blues')
```

Out[28]: <Axes: >



4.3 Pre-processing

Feature Engineering

#converting the columns which are not normally distributed to numpy array

```
In [13]: acid=np.asarray(result['fixed acidity'])
         resid=np.asarray(result['residual sugar'])
         free=np.asarray(result['free sulfur dioxide'])
         sulfur=np.asarray(result['total sulfur dioxide'])
```

defining a function to convert values between 0-1 using min_max techniques

```
In [14]: def min_max(array):
             mini=min(array)
             maxi=max(array)
             for i in range(len(array)):
                 array[i]=(array[i]-mini)/(maxi-mini)

         min_max(acid)
         min_max(resid)
         min_max(free)
         min_max(sulfur)
```

#Dropping the previous columns and adding new columns with normalized values

```
In [18]: result=result.drop(['fixed acidity','residual sugar','free sulfur dioxide','total sulfur dioxide'],axis=1)
```

```
In [19]: result['fixed_acidity']=pd.DataFrame(acid)
         result['residual_sugar']=pd.DataFrame(resid)
         result['free_sulphur_di_oxide']=pd.DataFrame(free)
         result['Sulphur_di_oxide']=pd.DataFrame(sulfur)
```

4.4 Machine Learning Algorithm used

After the Dataset is pre-processed, it is then ready to feed to the Machine Learning Model. We have used Logistic regression, Random forest Gaussian Naive Bayes and Ensemble model. The model which performs the best will be used for deployment. We have used classifier models as the target variable is a categorical value. The features selected for prediction are Usage alcohol quantity, fixed acidity, volatile acidity, determination of density, pH . These features were selected based on correlation with target variable and within the features itself.

4.5 Results

Logistic Regression:

We are using logistic regression to classify load types. The model uses a logi stic function, which maps any input value to an output value between 0 and 1. This output value represents the probability of the binary outcome bein g 1. The logistic regression model estimates the coefficients of the independ ent variables that maximize the likelihood of the observed data given the m odel. After applying logistic regression our model is giving 0.946875 as accuracy score.
The data was split into 90-10% for training data and testing data respectively, with the random state as 0.
Provides us with 0.92 recall for Red Wine & 0.93 for White wine

Random Forest Classifier:

Random Forest Classifier is a popular machine learning algorithm that is widely used for classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make more accurate predictions.

Random Forest Classifier works by building multiple decision trees on random subsets of the training data and random subsets of the features. The decision trees are constructed using a random selection of features at each node to split the data. This helps to reduce the correlation between the trees and improve the overall performance.

The data was split into 90-10% for training data and testing data respectively, with the random state as 0

The model was fitted by using the best hyparameters.

(max_depth=25,min_samples_leaf=15,n_jobs=-1,oob_score=True,

random_state=42)

We used 100 Decision Trees

The accuracy for the model is 0.61875 , we have focused more on recall for class 2(High Load) as our goal is to help the food & wine industry predict t he quality of wine based on its feature values

Gaussian Naive Bayes :

Gaussian Naive Bayes is commonly used in classification problems with continuous input features, and it can work well with small datasets. However, its assumption of independent features can limit its performance on complex datasets where the features are correlated.

The data was split into 90-10% for training data and testing data respectively, with the random state as 0
Accuracy = 0.478125

Ensemble model :

An ensemble model is a machine learning model that combines the predictions of multiple individual models to improve overall prediction accuracy and robustness. The basic idea behind ensemble modelling is that multiple models can provide better performance than a single model, especially if the individual models have different strengths and weaknesses. Ensemble models are commonly used in machine learning to improve the accuracy and stability of predictions, especially in complex and high-dimensional datasets.

Ensemble models can provide several benefits, including improved prediction accuracy, reduced overfitting, and increased model stability. However, they can also be more computationally intensive and require more data to train than individual models.

Models used:-Gaussian Naive Byes,Random Forest

The data was split into 90-10% for training data and testing data respectively, with the random state as 0

Accuracy of mean_squared_error =0.4890625

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

After observing the accuracy and performance metrics of the all the models, it is concluded that Logistic Model is the best suited model for the task of predicting wine type & Boosting using Random forest & Gaussian Naive Byes is best for task of predicting quality of wine.

5.2 Future Work

In the future, to improve the accuracy of the individual models using more varied datasets , it is clear that the algorithm or the data must be adjusted. We recommend feature engineering, using potential relationships between wine quality by applying more chemical features , or applying the boosting algorithm on the more accurate method. In addition, by applying the other performance measurement and other machine learning algorithms for the better comparison on results. This study will help the food industries to predict the quality of the different types of wines based on certain features, as it will be helpful for them to make a good product.

6 References:

- https://www.kaggle.com/
- https://realpython.com/logistic-regression
- https://seaborn.pydata.org/
- https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/