Wine Quality Prediction Using Machine Learning

INTRODUCTION

The quality of the wine is a very important part for the consumers as well as the manufacturing industries. Industries are increasing their sales using product quality certification. Nowadays, all over the world wine is a regularly used beverage and the industries are using the certification of product quality to increase their value in the market. Previously, testing of product quality will be done at the end of the production, this is a time taking process and it requires a lot of resources such as the need for various human experts for the assessment of product quality which makes this process very expensive. Every human has their own opinion about the test, so identifying the quality of the wine based on human experts is a challenging task.

There are several features to predict the wine quality but the entire features will not be relevant for better prediction.

This project aims to search for the elements which affect wine quality by using multiclass decision classification methods such as Logistic Regression, Confusion Matrix, Accuracy, Error Rate to get the best accuracy.

The given dataset consists of two types of wines, namely red and white variants of the Portuguese "Vinho Verde" wine.

GENERAL INFORMATION REGARDING DATA

Type: Two types of wines such as white wine and red wine.

Fixed acidity: Fixed acids include tartaric, malic, citric, and succinic acids which are found in grapes(except succinic). Acids are one of the fundamental properties of wine and contribute greatly to the taste of the wine, Acidity in food and drink tastes tart and zesty. Tasting acidity is also sometimes confused with alcohol. Wines with higher acidity feel lighter-bodied because they come across as “spritzy”. Reducing acids significantly might lead to wines tasting flat.

Volatile acidity: These acids are to be distilled out from the wine before completing the production process. It is primarily constituted of acetic acid though other acids like lactic, formic and butyric acids might also be present. Excess of volatile acids are undesirable and lead to unpleasant flavour.

Citric acid: This is one of the fixed acids which gives a wine its freshness. Usually most of it is consumed during the fermentation process and sometimes it is added separately to give the wine more freshness.

Residual sugar: This typically refers to the natural sugar from grapes which remains after the fermentation process stops, or is stopped.

Chlorides: Chloride concentration in the wine is influenced by land and its highest levels are found in wines coming from countries where irrigation is carried out using salty water or in areas with brackish terrains.

Free sulfur dioxide: This is the part of the sulphur dioxide that when added to a wine is said to be free after the remaining part binds. Winemakers will always try to get the highest proportion of free sulphur to bind. They are also known as sulfites and too much of it is undesirable and gives a pungent odour.

Total sulfur dioxide: This is the sum total of the bound and the free sulfur dioxide. This is mainly added to kill harmful bacteria and preserve quality and freshness.

There are usually legal limits for sulfur levels in wines and excess of it can even kill good yeast and give out undesirable odour.

Density: This can be represented as a comparison of the weight of a specific volume of wine to an equivalent volume of water. It is generally used as a measure of the conversion of sugar to alcohol.

pH: Also known as the potential of hydrogen, this is a numeric scale to specify the acidity or basicity of the wine. Fixed acidity contributes the most towards the pH of wines. Solutions with a pH less than 7 are acidic, while solutions with a pH greater than 7 are basic. With a pH of 7, pure water is neutral. Most wines have a pH between 2.9 and 3.9 and are therefore acidic.

Sulphates: These are mineral salts containing sulfur. They are a regular part of winemaking around the world and are considered essential. They are connected to the fermentation process and affect the wine aroma and flavour.

Alcohol: It is usually measured in % vol or alcohol by volume(ABV).

Quality: Wine experts grade the wine quality between 0(very bad) and 10(excellent). The eventual quality score is the median of at least three evaluations made by the same wine experts.

TOOLS AND LIBRARIES

- **Language:** Python
- **Platform:** Jupyter Notebook
- **Libraries:** 1) pandas
  2) NumPy
  3) scikit-learn
  4) Seaborn
  5) Matplotlib

PROJECT PROCESS

1. Data collection
2. Data preprocessing
3. Data analysis and visualisation
4. Model selection, training and testing
5. Model evaluation

CODE & OUTPUT

Importing Packages

In [129…]
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
from scipy.stats import norm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix as cm
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.model_selection import cross_validate, cross_val_score
```

In [130…]
```
wine_df = pd.read_csv('/Users/homefolder/wine_ml/winequalityN.csv')
```

About Data

In [131…]
```
wine_df.head()
```

Out[131]:

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	white	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	white	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	white	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	white	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	white	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

In [132…]
```
wine_df.tail()
```

Out[132]:

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
6492	red	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
6493	red	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	NaN	11.2	6
6494	red	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
6495	red	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
6496	red	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

In [133…]
```
wine_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   type                  6497 non-null   object 
 1   fixed acidity         6487 non-null   float64
 2   volatile acidity      6489 non-null   float64
 3   citric acid           6494 non-null   float64
 4   residual sugar        6495 non-null   float64
 5   chlorides             6495 non-null   float64
 6   free sulfur dioxide   6497 non-null   float64
 7   total sulfur dioxide  6497 non-null   float64
 8   density               6497 non-null   float64
 9   pH                    6488 non-null   float64
 10  sulphates             6493 non-null   float64
 11  alcohol               6497 non-null   float64
 12  quality               6497 non-null   int64  
dtypes: float64(11), int64(1), object(1)
memory usage: 660.0+ KB
```

In [134…]
```
wine_df.describe()
```

Out[134]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	6487.000000	6489.000000	6494.000000	6495.000000	6495.000000	6497.000000	6497.000000	6497.000000	6488.000000	6493.000000	6497.000000	6497.000000
mean	7.216579	0.339691	0.318722	5.444326	0.056042	30.525319	115.744574	0.994697	3.218395	0.531215	10.491801	5.818378
std	1.296750	0.164649	0.145265	4.758125	0.035036	17.749400	56.521855	0.002999	0.160748	0.148814	1.192712	0.873255
min	3.800000	0.080000	0.000000	0.600000	0.009000	1.000000	6.000000	0.987110	2.720000	0.220000	8.000000	3.000000
25%	6.400000	0.230000	0.250000	1.800000	0.038000	17.000000	77.000000	0.992340	3.110000	0.430000	9.500000	5.000000
50%	7.000000	0.290000	0.310000	3.000000	0.047000	29.000000	118.000000	0.994890	3.210000	0.510000	10.300000	6.000000
75%	7.700000	0.400000	0.390000	8.100000	0.065000	41.000000	156.000000	0.996990	3.320000	0.600000	11.300000	6.000000
max	15.900000	1.580000	1.660000	65.800000	0.611000	289.000000	440.000000	1.038980	4.010000	2.000000	14.900000	9.000000

Switching Column Names Into Suitable Format

In [135…
```
print(*wine_df.columns, sep='\n')
```

```
type
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
```

In [136…
```
wine_df.columns = ('type', 'fixed_acidity', 'volatile_acidity', 'citric_acid',
       'residual_sugar', 'chlorides', 'free_sulfur_dioxide',
       'total_sulfur_dioxide', 'density', 'pH', 'sulphates', 'alcohol',
       'quality')
```

Checking For NULL Values

In [137…
```
Sum = wine_df.isnull().sum()
Percentage = (wine_df.isnull().sum()/wine_df.isnull().count())
pd.concat([Sum,Percentage], axis=1, keys=['Sum', 'Percentage'])
```

Out[137]:

	Sum	Percentage
type	0	0.000000
fixed_acidity	10	0.001539
volatile_acidity	8	0.001231
citric_acid	3	0.000462
residual_sugar	2	0.000308
chlorides	2	0.000308
free_sulfur_dioxide	0	0.000000
total_sulfur_dioxide	0	0.000000
density	0	0.000000
pH	9	0.001385
sulphates	4	0.000616
alcohol	0	0.000000
quality	0	0.000000

Filling of Row Data

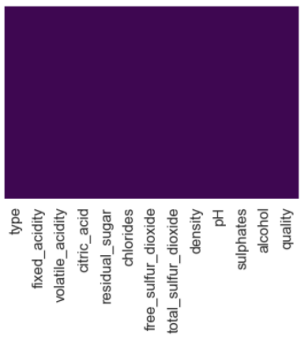
In [138…
```
def null_cell(wine_df):
    total_missing_values = wine_df.isnull().sum()
    missing_values_per = wine_df.isnull().sum()/wine_df.isnull().count()
    null_values = pd.concat([total_missing_values, missing_values_per], axis=1, keys=['total_null', 'total_null_perc'])
    null_values = null_values.sort_values('total_null', ascending=False)
    return null_values[null_values['total_null'] > 0]
```

In [139…
```
fill_list = (null_cell(wine_df)).index
```

In [140…
```
wine_df_mean = wine_df.copy()
for col in fill_list:
    wine_df_mean.loc[:, col].fillna(wine_df_mean.loc[:, col].mean(), inplace=True)
```

In [141…
```
sns.heatmap(wine_df_mean.isnull(), yticklabels=False, cbar=False, cmap='viridis')
```

Out[141]:
```
<AxesSubplot:>
```

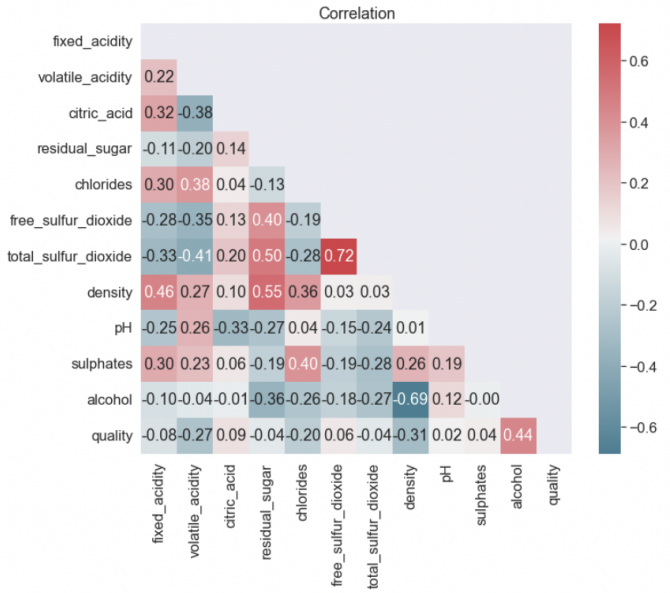


In [142…
```
corr_matrix = wine_df_mean.corr()
corr_list = corr_matrix.quality.abs().sort_values(ascending=False).index[0:]
```

In [143…
```
corr_list
```

Out[143]:
```
Index(['quality', 'alcohol', 'density', 'volatile_acidity', 'chlorides',
       'citric_acid', 'fixed_acidity', 'free_sulfur_dioxide',
       'total_sulfur_dioxide', 'sulphates', 'residual_sugar', 'pH'],
      dtype='object')
```

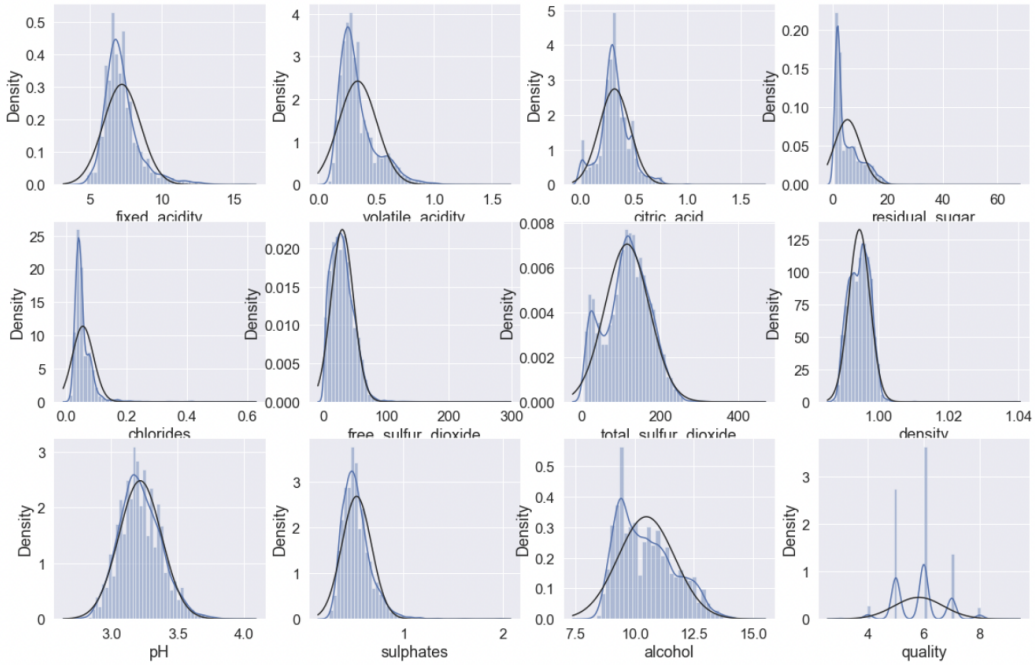
In [144…
```
plt.figure(figsize=(11,9))
plt.title('Correlation')
dropSelf = np.zeros_like(corr_matrix)
dropSelf[np.triu_indices_from(dropSelf)] = True
sns.heatmap(corr_matrix, cmap=sns.diverging_palette(220, 10, as_cmap=True), annot=True, fmt=".2f", mask=dropSelf)
sns.set(font_scale=1.5)
```



Wine quality has the highest correlation with alcohol. Other relation degrees are very low with each other, such as citric_acid, free_sulfur_dioxide, sulphates and pH. Quality also has a low negative correlation with density, volatile_acidity, chlorides, total_sulfur_dioxide and residual_sugar.

Distribution Of Variables

```
In [145…
plt.figure(figsize = (20,22))
for i in range(1,13):
    plt.subplot(5,4,i)
    sns.distplot(wine_df_mean[wine_df_mean.columns[i]], fit=norm)
```



Two Types Of Wine Quality Classes

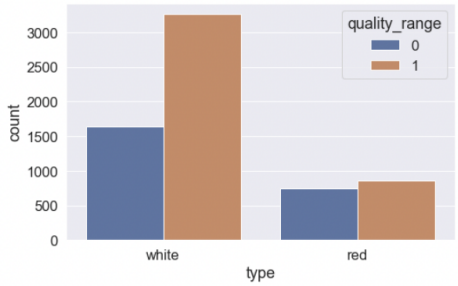
```
In [146…
wine_df_bins= wine_df_mean.copy()
```

```
In [147…
bins = [0,5,10]
labels = [0, 1]  # 'low'=0, 'high'=1
wine_df_bins['quality_range']= pd.cut(x=wine_df_bins['quality'], bins=bins, labels=labels)
print(wine_df_bins[['quality_range','quality']].head(5))
wine_df_bins = wine_df_bins.drop('quality', axis=1)
```

```
  quality_range  quality
0             1        6
1             1        6
2             1        6
3             1        6
4             1        6
```

Quality In Different Wine Types

```
In [148…
plt.figure(figsize=(8,5))
sns.countplot(x='type', hue='quality_range', data=wine_df_bins)
plt.show()
```

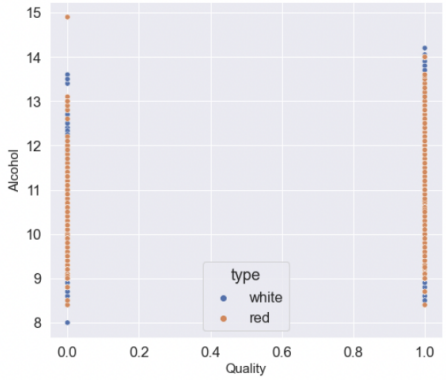


As we see on the chart, low quality red wine has the highest numerical value in data set as well as low quality white wine. High quality white and red wines have little place in data.

Relation Between Quality And Alcohol

In [149…

```
plt.figure(figsize=(8,7))
sns.scatterplot(x='quality_range',
                y='alcohol',
                hue='type',
                data=wine_df_bins);
plt.xlabel('Quality', size=15)
plt.ylabel('Alcohol', size =15)
plt.show()
```

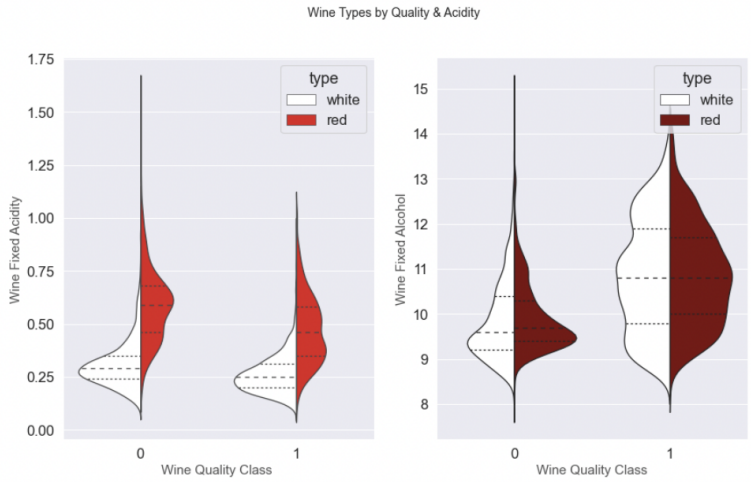


Red and White wines has similar results on the chart. High quality wines are mostly red wines and have more alcohol levels.

Quality And Volatile Acidity In Different Wine Types

In [150…

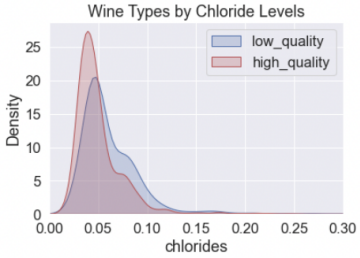
```
f, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 8))
f.suptitle('Wine Types by Quality & Acidity', fontsize=14)
sns.violinplot(x='quality_range', y='volatile_acidity', hue='type', data=wine_df_bins, split=True, inner='quart', linewidth=1.3,
               palette={'red': 'red', 'white': 'white'}, ax=ax1)
ax1.set_xlabel("Wine Quality Class ", size = 15, alpha=0.8)
ax1.set_ylabel("Wine Fixed Acidity", size = 15, alpha=0.8)
sns.violinplot(x='quality_range', y='alcohol', hue='type', data=wine_df_bins, split=True, inner='quart', linewidth=1.3,
               palette={'red': 'darkred', 'white': 'white'}, ax=ax2)
ax2.set_xlabel("Wine Quality Class", size = 15, alpha=0.8)
ax2.set_ylabel("Wine Fixed Alcohol", size = 15, alpha=0.8)
plt.show()
```



Fixed acidity level is low in both wine types, especially in white wine while red wine has more in low quality type up to 1.70. Fixed alcohol level is again high in red wine class comparing white wine in low quality. High quality type has the highest fixed alcohol level in booth wine types.

Chloride Levels In Different Wine Types

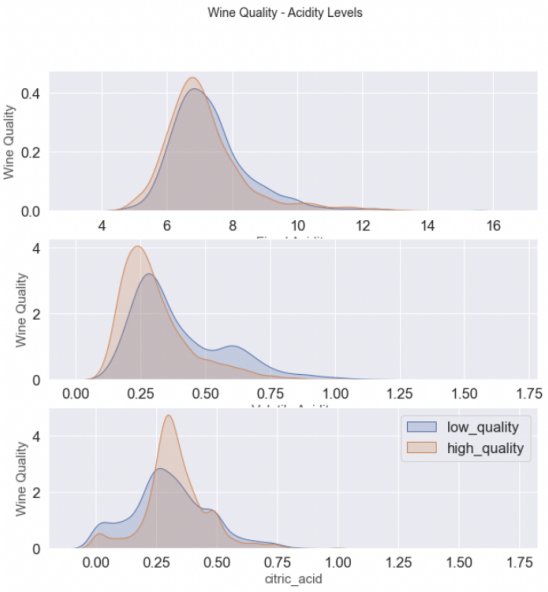
In [151…
```
plt.figure(figsize= (6,4))
low_quality = wine_df_bins [wine_df_bins['quality_range'] == 0]['chlorides']
high_quality = wine_df_bins [wine_df_bins['quality_range'] == 1][ 'chlorides']
ax = sns.kdeplot(data= low_quality, label= 'low_quality', shade=True, color=None)
ax = sns.kdeplot(data= high_quality, label= 'high_quality', shade=True, color= "r")
plt.title("Wine Types by Chloride Levels")
plt.xlim(0.0,0.3)
plt.legend()
plt.show()
```



Chloride levels are a bit higher in red wine in contrast with white wine.

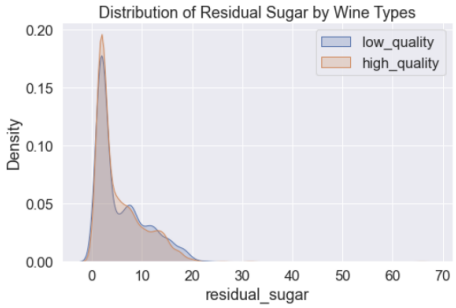
Fixed Acidity & Volatile Acidity & Citric Acid Density in Different Wine Types

In [152…
```
f, (ax1, ax2, ax3) = plt.subplots(3, figsize = (10,10))
f.suptitle('Wine Quality - Acidity Levels', fontsize=14)
fixed_acidity_low_quality    = wine_df_bins [wine_df_bins['quality_range'] == 0]['fixed_acidity']
fixed_acidity_high_quality   = wine_df_bins [wine_df_bins['quality_range'] == 1]['fixed_acidity']
volatile_acidity_low_quality = wine_df_bins [wine_df_bins['quality_range'] == 0]['volatile_acidity']
volatile_acidity_high_quality = wine_df_bins [wine_df_bins['quality_range'] == 1]['volatile_acidity']
citric_acid_low_quality      = wine_df_bins [wine_df_bins['quality_range'] == 0]['citric_acid']
citric_acid_high_quality     = wine_df_bins [wine_df_bins['quality_range'] == 1]['citric_acid']
sns.kdeplot(data=fixed_acidity_low_quality, label="low_quality", shade=True, ax=ax1)
sns.kdeplot(data=fixed_acidity_high_quality, label="high_quality", shade=True, ax=ax1)
ax1.set_xlabel("Fixed Acidity", size = 15, alpha=0.8)
ax1.set_ylabel("Wine Quality", size = 15, alpha=0.8)
sns.kdeplot(data=volatile_acidity_low_quality, label="low_quality", shade=True, ax=ax2)
sns.kdeplot(data=volatile_acidity_high_quality, label="high_quality", shade=True, ax=ax2)
ax2.set_xlabel("Volatile Acidity", size = 15, alpha=0.8)
ax2.set_ylabel("Wine Quality", size = 15, alpha=0.8)
sns.kdeplot(data=citric_acid_low_quality, label="low_quality", shade=True,ax=ax3)
sns.kdeplot(data=citric_acid_high_quality, label="high_quality", shade=True, ax=ax3)
ax3.set_xlabel("citric_acid", size = 15, alpha=0.8)
ax3.set_ylabel("Wine Quality", size = 15, alpha=0.8)
plt.legend()
plt.show()
```



Residual Sugar Levels In Different Wine Types

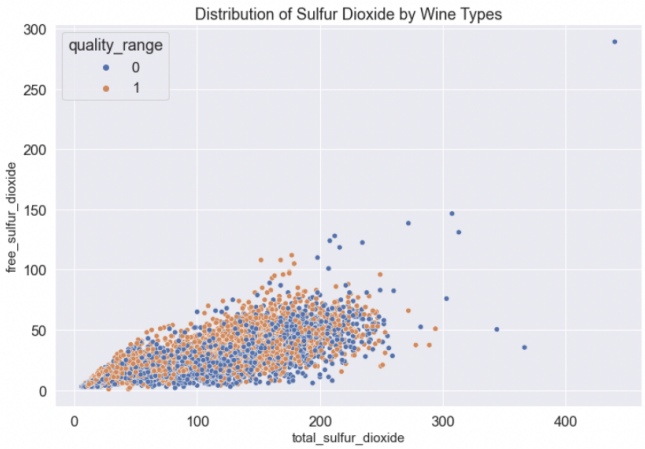
In [153…
```
plt.figure(figsize=(8,5))
residual_sugar_low = wine_df_bins [wine_df_bins['quality_range'] == 0]['residual_sugar']
residual_sugar_high = wine_df_bins [wine_df_bins['quality_range'] == 1]['residual_sugar']
ax = sns.kdeplot(data = residual_sugar_low, label = 'low_quality', shade=True)
ax = sns.kdeplot(data = residual_sugar_high, label = 'high_quality', shade=True)
plt.title("Distribution of Residual Sugar by Wine Types")
plt.legend()
plt.show()
```



Sulfur Dioxide Levels In Different Wine Types

In [154…
```
plt.figure(figsize=(12,8))
sns.scatterplot(x = 'total_sulfur_dioxide', y = 'free_sulfur_dioxide', hue = 'quality_range',data = wine_df_bins);
plt.title("Distribution of Sulfur Dioxide by Wine Types")
plt.xlabel('total_sulfur_dioxide',size=15)
plt.ylabel('free_sulfur_dioxide', size =15)
```

Out[154]:
```
Text(0, 0.5, 'free_sulfur_dioxide')
```

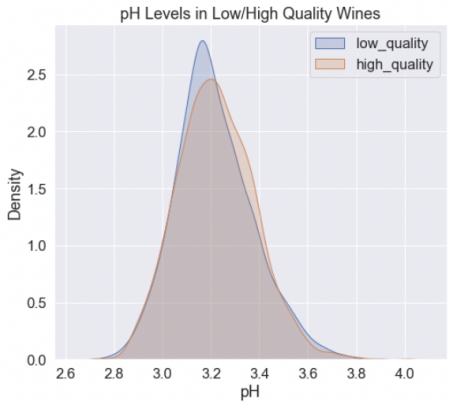


There are some extreme values in low quality wine type. Total sulfur dioxide level is getting higher in some low quality wine type while general disturubution is standing up to 100 levels of free sulfur dioxide.

pH Levels In Different Wine Types

In [155…

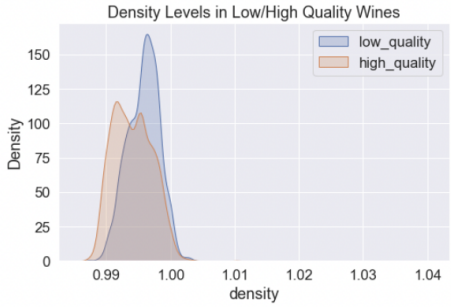
```
plt.figure(figsize=(8,7))
pH_low_quality  = wine_df_bins [wine_df_bins['quality_range']  == 0]['pH']
pH_high_quality = wine_df_bins [wine_df_bins['quality_range']== 1][ 'pH']
ax = sns.kdeplot(data = pH_low_quality, label = 'low_quality', shade=True)
ax = sns.kdeplot(data = pH_high_quality, label = 'high_quality',   shade=True)
plt.title("pH Levels in Low/High Quality Wines")
plt.xlabel('pH')
plt.legend()
plt.show()
```



Density In Different Wine Types

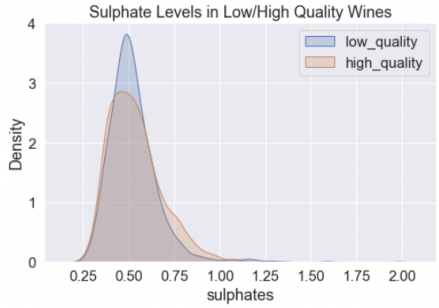
In [156…

```
plt.figure(figsize=(8,5))
density_low_quality  = wine_df_bins [wine_df_bins['quality_range'] == 0]['density']
density_high_quality = wine_df_bins [wine_df_bins['quality_range'] == 1]['density']
ax = sns.kdeplot(data = density_low_quality, label = 'low_quality', shade=True)
ax = sns.kdeplot(data = density_high_quality, label = 'high_quality', shade=True)
plt.title("Density Levels in Low/High Quality Wines")
plt.xlabel('density')
plt.legend()
plt.show()
```



Sulphate Levels In Different Wine Types

In [157…
```
plt.figure(figsize=(8,5))
sulphates_low_quality = wine_df_mean [wine_df_bins['quality_range'] == 0]['sulphates']
sulphates_high_quality = wine_df_mean [wine_df_bins['quality_range'] == 1][ 'sulphates']
ax = sns.kdeplot(data = sulphates_low_quality, label = 'low_quality',  shade=True)
ax = sns.kdeplot(data = sulphates_high_quality, label = 'high_quality', shade=True)
plt.title("Sulphate Levels in Low/High Quality Wines")
plt.xlabel('sulphates')
plt.legend()
plt.show()
```



There is more low quality wine in between 0.4 and 0.6 levels of sulphate levels. Both quality types have similar values.

Logistic Regression Model

In [158…
```
wine_df_bins.type = wine_df_bins.type.map({'white':0, 'red':1})
```

In [159…
```
X = wine_df_bins[['type', 'alcohol', 'density', 'volatile_acidity', 'chlorides',
       'citric_acid', 'fixed_acidity', 'free_sulfur_dioxide',
       'total_sulfur_dioxide', 'sulphates', 'residual_sugar', 'pH']]
y = wine_df_bins.quality_range
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=40)
```

In [160…
```
lr = LogisticRegression(random_state=40)
lr.fit(X_train, y_train)
```

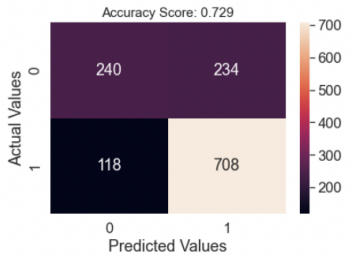
Out[160]:
```
LogisticRegression(random_state=40)
```

In [161…
```
train_accuracy = lr.score(X_train, y_train)
test_accuracy = lr.score(X_test, y_test)
print('One-vs-rest', '-'*35,
      'Accuracy in Train Group : {:.2f}'.format(train_accuracy),
      'Accuracy in Test  Group : {:.2f}'.format(test_accuracy), sep='\n')
```

```
One-vs-rest
-----------------------------------
Accuracy in Train Group : 0.74
Accuracy in Test  Group : 0.73
```

Confusion Matrix

In [162…
```
predictions = lr.predict(X_test)
score = round(accuracy_score(y_test, predictions), 3)
cm1 = cm(y_test, predictions)
sns.heatmap(cm1, annot=True, fmt=".0f")
plt.xlabel('Predicted Values')
plt.ylabel('Actual Values')
plt.title('Accuracy Score: {0}'.format(score), size = 15)
plt.show()
```



In [163…
```
pred_test  = lr.predict(X_test)
pred_train = lr.predict(X_train)
```

In [164…
```
quality_pred = LogisticRegression(random_state=40)
quality_pred.fit(X_train,y_train)
```

Out[164]:
```
LogisticRegression(random_state=40)
```

Accuracy

In [165…
```
print("Accuracy Score : ",  quality_pred.score(X_test, y_test))
```

```
Accuracy Score :  0.7292307692307692
```

Error Rate

In [166…
```
Error_Rate = 1- (accuracy_score(y_test, pred_test))
print("Error Rate :", Error_Rate)
```

```
Error Rate : 0.27076923076923076
```

Results

In [167…
```
C_values = [0.001,0.01,0.1,1,10,100, 1000]
accuracy_wine_df = pd.DataFrame(columns = ['C_values','Accuracy'])

accuracy_values = pd.DataFrame(columns=['C Value', 'Accuracy Train', 'Accuracy Test'])

for c in C_values:

    # Apply logistic regression model to training data
    lr = LogisticRegression(penalty = 'l2', C = c, random_state = 0)
    lr.fit(X_train,y_train)
    accuracy_values = accuracy_values.append({'C Value': c,
                                              'Accuracy Train' : lr.score(X_train, y_train),
                                              'Accuracy Test': lr.score(X_test, y_test)
                                              }, ignore_index=True)
display(accuracy_values)
```

	C Value	Accuracy Train	Accuracy Test
0	0.001	0.663845	0.651538
1	0.01	0.694439	0.676154
2	0.1	0.727535	0.723077
3	1.0	0.736579	0.729231
4	10.0	0.736771	0.726923
5	100.0	0.732346	0.723077
6	1000.0	0.731768	0.717692

In [168…
```
lrm = LogisticRegression()
cv = cross_validate(estimator=lrm,
                    X=X,
                    y=y,
                    cv=10,return_train_score=True
                   )
print('Test Scores : ', cv['test_score'], sep = '\n')
print("-"*50)
print('Train Scores : ', cv['train_score'], sep = '\n')
```

```
Test Scores :
[0.63230769 0.71076923 0.72307692 0.70153846 0.73538462 0.74
 0.72923077 0.80893683 0.66101695 0.76733436]
--------------------------------------------------
Train Scores :
[0.73473576 0.74157688 0.73319651 0.72994698 0.7367881  0.73541987
 0.72413203 0.71545828 0.73837209 0.73067715]
```

In [169…
```
print('Mean of Test Set : ', cv['test_score'].mean())
print('Mean of Train Set : ', cv['train_score'].mean())
```

```
Mean of Test Set :  0.7209595827900912
Mean of Train Set :  0.7320303658709777
```

The average accuracy score is calculated from 10 different accuracy scores from the model.

CONCLUSION

This project uses the two types of wine dataset, red and white, of Portuguese "Vinho Verde" wine to predict the quality of the wine based on the physicochemical properties.

In this project, we checked the general characteristics of the dataset. Data has some NULL values. Even though dropping missing values is still an option due to the low percentage of missing values in data, we preferred to fill them by the mean of data. Data set shows that red wine is very rich in wine quality with a high correlation with alcohol. We also looked at quality levels in each variable by using suitable charts for a general understanding.

The accuracy of the Logistic Regression algorithm came out to be 72.92%.

We focused on classification methods in this project. In the classification algorithms, by selecting the appropriate features and balancing the data, we can improve the performance of the model.

FUTURE SCOPE

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