

WinequalityEDA

October 28, 2025

0.1 EDA With Red Wine Data

Data Set Information:

The two datasets are related to red and white variants of the Portuguese “Vinho Verde” wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods.

Attribute Information:

Input variables (based on physicochemical tests): - 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 - sulphates - 11 - alcohol

Output variable (based on sensory data): - 12 - quality (score between 0 and 10)

```
[2]: import pandas as pd
df=pd.read_csv('./resources/winequality-red.csv', sep=';')
df.head()
```

```
[2]:    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

    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

    alcohol  quality
0        9.4        5
```

1	9.8	5
2	9.8	5
3	9.8	6
4	9.4	5

```
[48]: ## Summary of the data set
df.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
```

```
[49]: df.describe()
```

```
[49]:
```

	fixed acidity	volatile acidity	citric acid	residual sugar	\
count	1599.000000	1599.000000	1599.000000	1599.000000	
mean	8.319637	0.527821	0.270976	2.538806	
std	1.741096	0.179060	0.194801	1.409928	
min	4.600000	0.120000	0.000000	0.900000	
25%	7.100000	0.390000	0.090000	1.900000	
50%	7.900000	0.520000	0.260000	2.200000	
75%	9.200000	0.640000	0.420000	2.600000	
max	15.900000	1.580000	1.000000	15.500000	

	chlorides	free sulfur dioxide	total sulfur dioxide	density	\
count	1599.000000	1599.000000	1599.000000	1599.000000	
mean	0.087467	15.874922	46.467792	0.996747	
std	0.047065	10.460157	32.895324	0.001887	
min	0.012000	1.000000	6.000000	0.990070	
25%	0.070000	7.000000	22.000000	0.995600	
50%	0.079000	14.000000	38.000000	0.996750	
75%	0.090000	21.000000	62.000000	0.997835	

max	0.611000	72.000000	289.000000	1.003690
-----	----------	-----------	------------	----------

	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000
mean	3.311113	0.658149	10.422983	5.636023
std	0.154386	0.169507	1.065668	0.807569
min	2.740000	0.330000	8.400000	3.000000
25%	3.210000	0.550000	9.500000	5.000000
50%	3.310000	0.620000	10.200000	6.000000
75%	3.400000	0.730000	11.100000	6.000000
max	4.010000	2.000000	14.900000	8.000000

```
[50]: df.shape
      #(row, column)
```

```
[50]: (1599, 12)
```

```
[51]: ## List down all column names
      df.columns
```

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

```
[52]: df['quality'].unique()
```

```
[52]: array([5, 6, 7, 4, 8, 3])
```

```
[53]: ## Missing values in the dataset
      df.isnull().sum() #For here no missing values
```

```
[53]: fixed acidity      0
      volatile acidity  0
      citric acid       0
      residual sugar    0
      chlorides         0
      free sulfur dioxide 0
      total sulfur dioxide 0
      density          0
      pH              0
      sulphates        0
      alcohol          0
      quality          0
      dtype: int64
```

```
[54]: ## Find out Duplicate records
      df[df.duplicated()]
```

```
[54]:
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides \
4	7.4	0.700	0.00	1.90	0.076
11	7.5	0.500	0.36	6.10	0.071
27	7.9	0.430	0.21	1.60	0.106
40	7.3	0.450	0.36	5.90	0.074
65	7.2	0.725	0.05	4.65	0.086
...
1563	7.2	0.695	0.13	2.00	0.076
1564	7.2	0.695	0.13	2.00	0.076
1567	7.2	0.695	0.13	2.00	0.076
1581	6.2	0.560	0.09	1.70	0.053
1596	6.3	0.510	0.13	2.30	0.076

	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates \
4	11.0	34.0	0.99780	3.51	0.56
11	17.0	102.0	0.99780	3.35	0.80
27	10.0	37.0	0.99660	3.17	0.91
40	12.0	87.0	0.99780	3.33	0.83
65	4.0	11.0	0.99620	3.41	0.39
...
1563	12.0	20.0	0.99546	3.29	0.54
1564	12.0	20.0	0.99546	3.29	0.54
1567	12.0	20.0	0.99546	3.29	0.54
1581	24.0	32.0	0.99402	3.54	0.60
1596	29.0	40.0	0.99574	3.42	0.75

	alcohol	quality
4	9.4	5
11	10.5	5
27	9.5	5
40	10.5	5
65	10.9	5
...
1563	10.1	5
1564	10.1	5
1567	10.1	5
1581	11.3	5
1596	11.0	6

[240 rows x 12 columns]

```
[55]: ## Removing Duplicates
df.drop_duplicates(inplace=True)
```

```
[56]: df[df.duplicated()]
```

```
[56]: Empty DataFrame
      Columns: [fixed acidity, volatile acidity, citric acid, residual sugar,
      chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates,
      alcohol, quality]
      Index: []
```

```
[57]: df.shape
```

```
[57]: (1359, 12)
```

```
[58]: ## Correlation
```

```
df.corr()
```

```
[58]:
```

	fixed acidity	volatile acidity	citric acid	\
fixed acidity	1.000000	-0.255124	0.667437	
volatile acidity	-0.255124	1.000000	-0.551248	
citric acid	0.667437	-0.551248	1.000000	
residual sugar	0.111025	-0.002449	0.143892	
chlorides	0.085886	0.055154	0.210195	
free sulfur dioxide	-0.140580	-0.020945	-0.048004	
total sulfur dioxide	-0.103777	0.071701	0.047358	
density	0.670195	0.023943	0.357962	
pH	-0.686685	0.247111	-0.550310	
sulphates	0.190269	-0.256948	0.326062	
alcohol	-0.061596	-0.197812	0.105108	
quality	0.119024	-0.395214	0.228057	

	residual sugar	chlorides	free sulfur dioxide	\
fixed acidity	0.111025	0.085886	-0.140580	
volatile acidity	-0.002449	0.055154	-0.020945	
citric acid	0.143892	0.210195	-0.048004	
residual sugar	1.000000	0.026656	0.160527	
chlorides	0.026656	1.000000	0.000749	
free sulfur dioxide	0.160527	0.000749	1.000000	
total sulfur dioxide	0.201038	0.045773	0.667246	
density	0.324522	0.193592	-0.018071	
pH	-0.083143	-0.270893	0.056631	
sulphates	-0.011837	0.394557	0.054126	
alcohol	0.063281	-0.223824	-0.080125	
quality	0.013640	-0.130988	-0.050463	

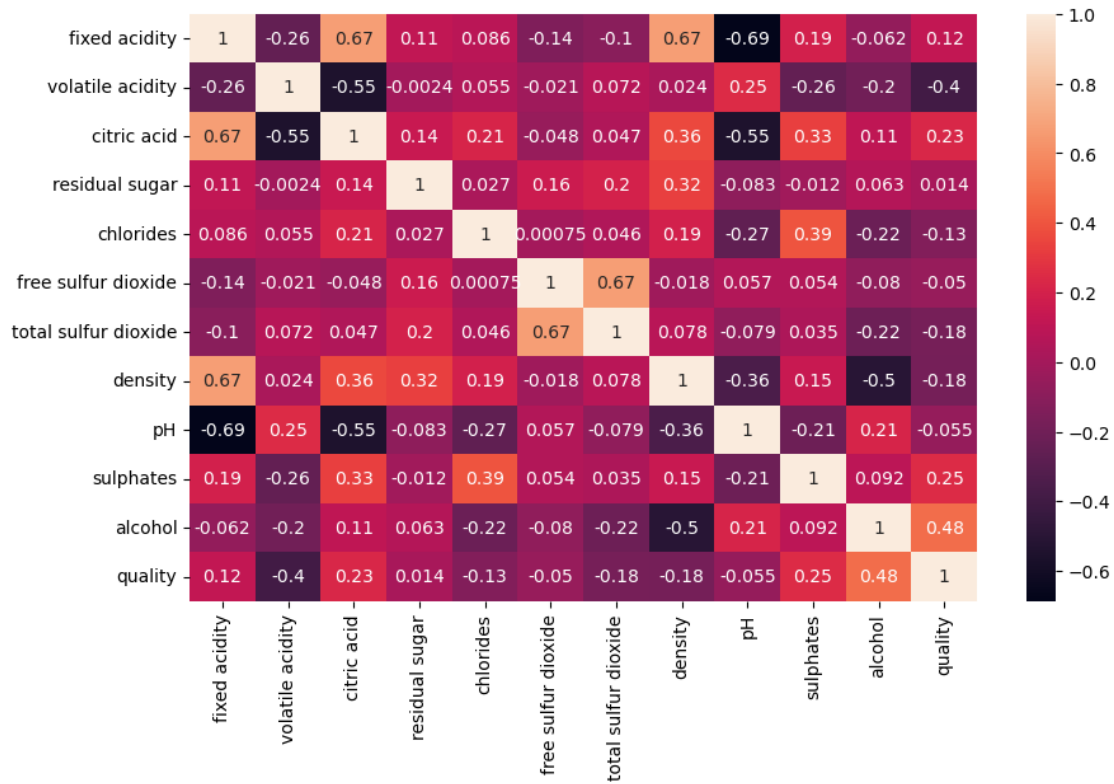
	total sulfur dioxide	density	pH	sulphates	\
fixed acidity	-0.103777	0.670195	-0.686685	0.190269	
volatile acidity	0.071701	0.023943	0.247111	-0.256948	
citric acid	0.047358	0.357962	-0.550310	0.326062	
residual sugar	0.201038	0.324522	-0.083143	-0.011837	

chlorides	0.045773	0.193592	-0.270893	0.394557
free sulfur dioxide	0.667246	-0.018071	0.056631	0.054126
total sulfur dioxide	1.000000	0.078141	-0.079257	0.035291
density	0.078141	1.000000	-0.355617	0.146036
pH	-0.079257	-0.355617	1.000000	-0.214134
sulphates	0.035291	0.146036	-0.214134	1.000000
alcohol	-0.217829	-0.504995	0.213418	0.091621
quality	-0.177855	-0.184252	-0.055245	0.248835

	alcohol	quality
fixed acidity	-0.061596	0.119024
volatile acidity	-0.197812	-0.395214
citric acid	0.105108	0.228057
residual sugar	0.063281	0.013640
chlorides	-0.223824	-0.130988
free sulfur dioxide	-0.080125	-0.050463
total sulfur dioxide	-0.217829	-0.177855
density	-0.504995	-0.184252
pH	0.213418	-0.055245
sulphates	0.091621	0.248835
alcohol	1.000000	0.480343
quality	0.480343	1.000000

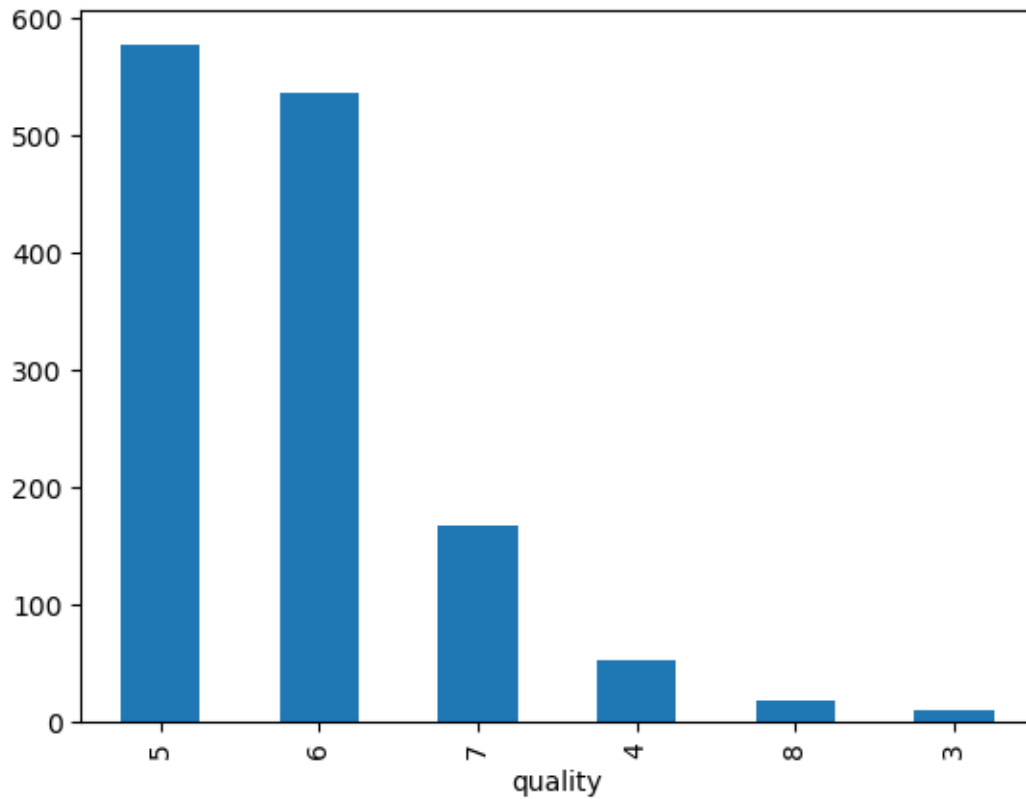
```
[59]: import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True)
```

```
[59]: <Axes: >
```



```
[60]: ## Visuaization
df.quality.value_counts().plot(kind='bar')
```

```
[60]: <Axes: xlabel='quality'>
```



```
[61]: ##Conclusion  
# It is an Imbalanced dataset  
## Lets balance this dataset
```

```
[67]: df.quality.value_counts()
```

```
[67]: quality  
5      577  
6      535  
7      167  
4       53  
8       17  
3       10  
Name: count, dtype: int64
```

```
[70]: X=df.drop('quality', axis=1)  
y=df['quality']
```

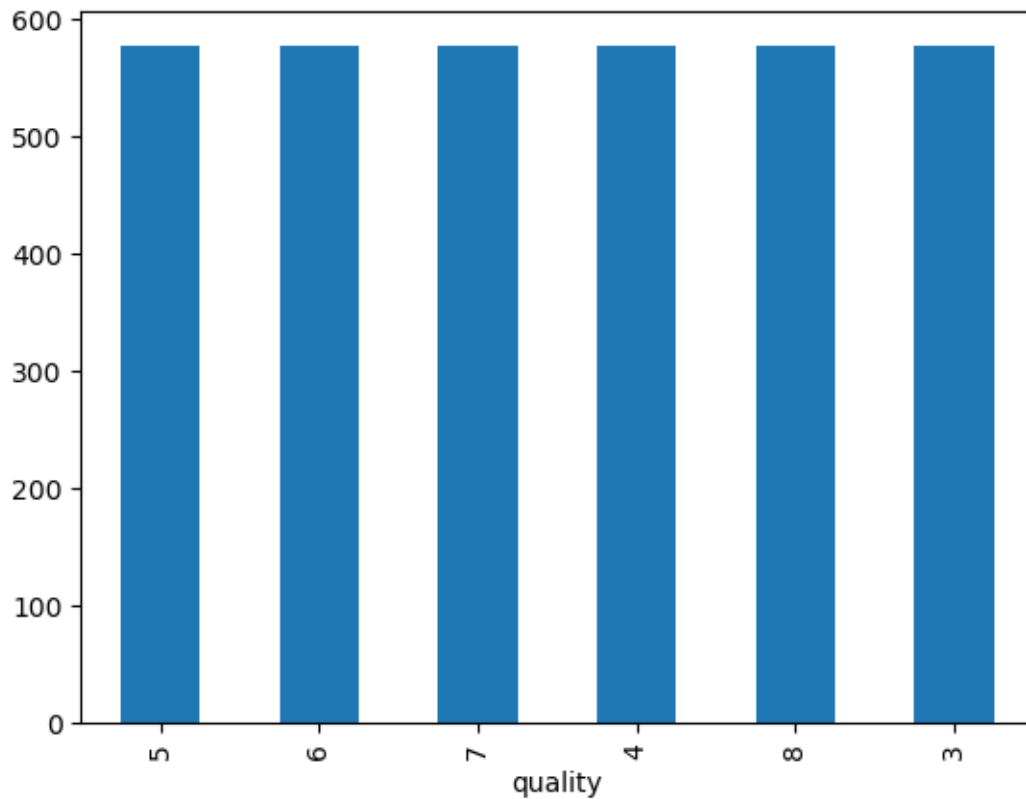
```
[73]: from imblearn.over_sampling import RandomOverSampler, SMOTE  
  
smote = SMOTE(random_state=42)
```



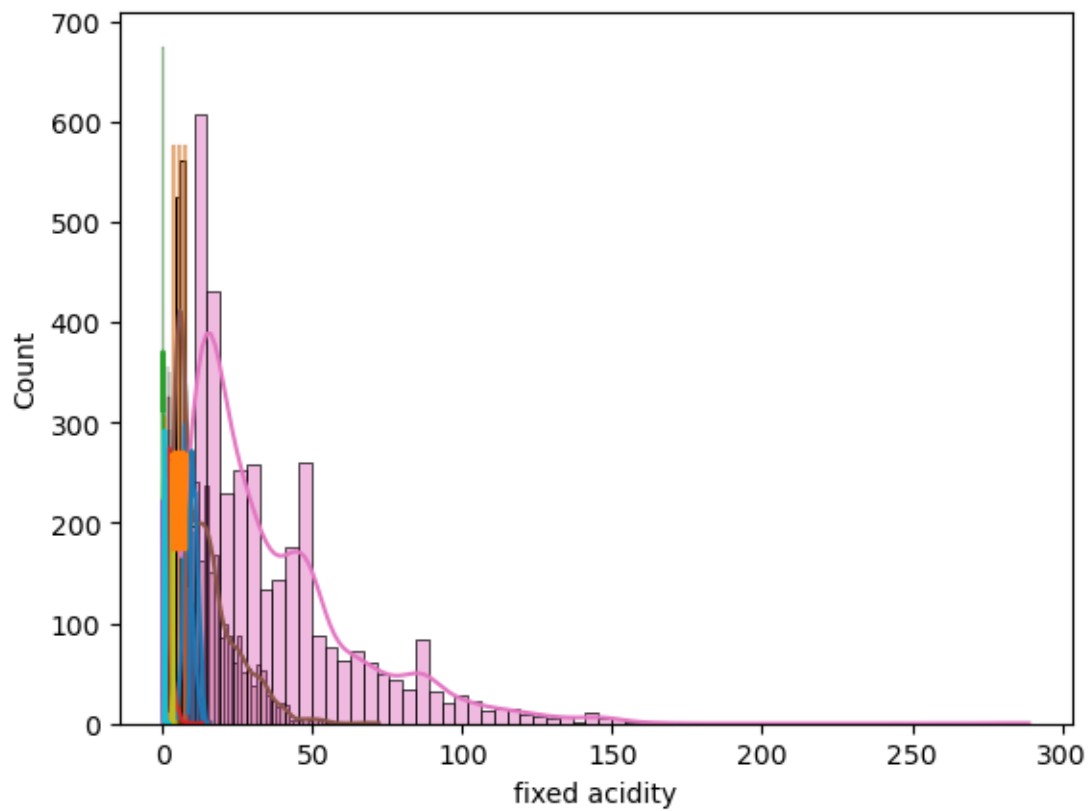
```
X_res, y_res = SMOTE().fit_resample(X, y)
df = pd.concat([pd.DataFrame(X_res, columns=X.columns),
                pd.DataFrame(y_res, columns=['quality'])], axis=1)
```

```
[74]: df.quality.value_counts().plot(kind='bar')
```

```
[74]: <Axes: xlabel='quality'>
```

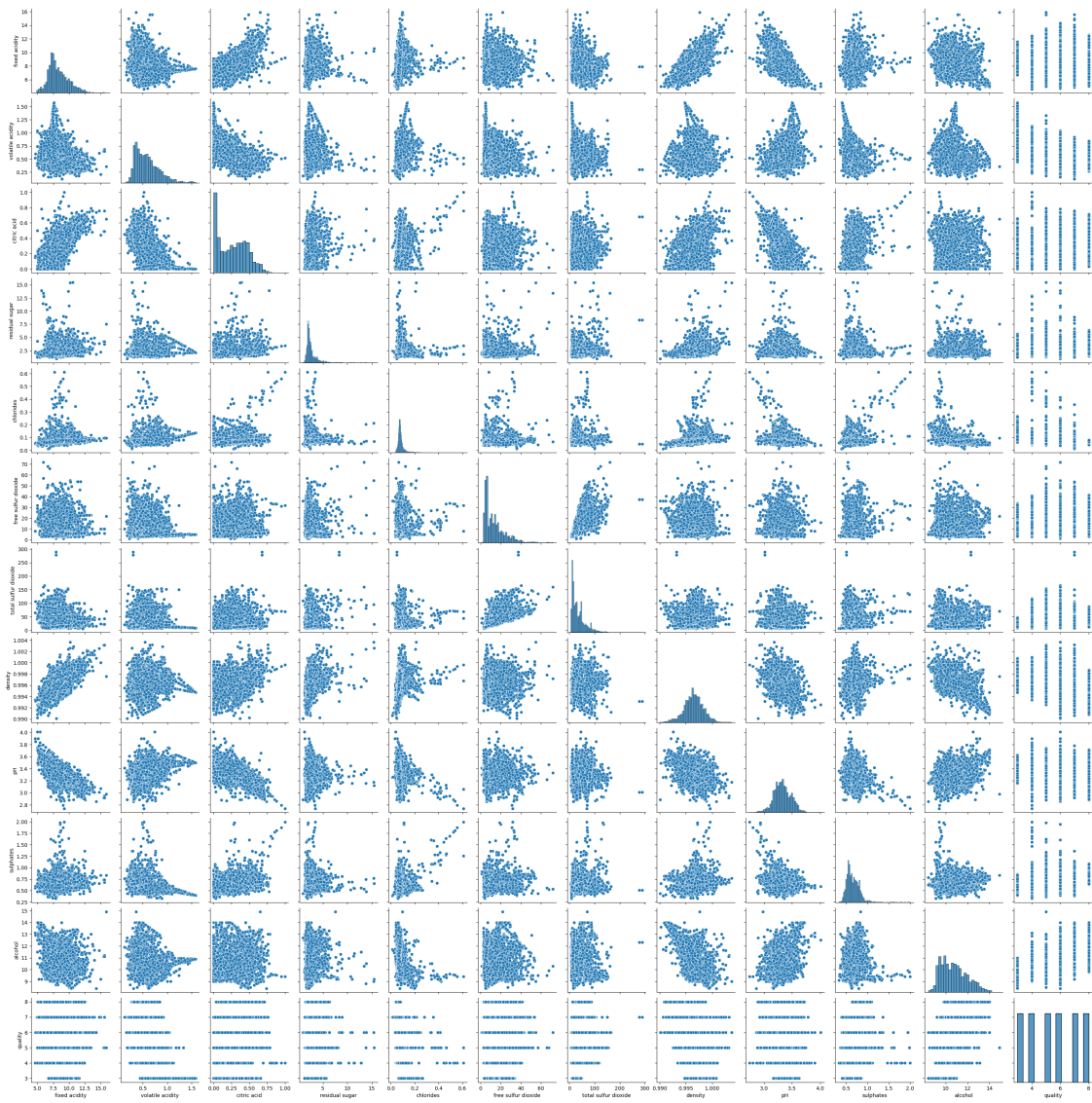


```
[ ]: for column in df.columns:
      sns.histplot(df[column], kde=True)
```



```
[76]: #univariate, Bivariate, Multivariate analysis  
sns.pairplot(df)
```

```
[76]: <seaborn.axisgrid.PairGrid at 0x18f2139d6a0>
```



```
[77]: ## Categorical plot
sns.catplot(x='quality', y='alcohol', data=df, kind='box')
```

```
[77]: <seaborn.axisgrid.FacetGrid at 0x18f2139d550>
```

