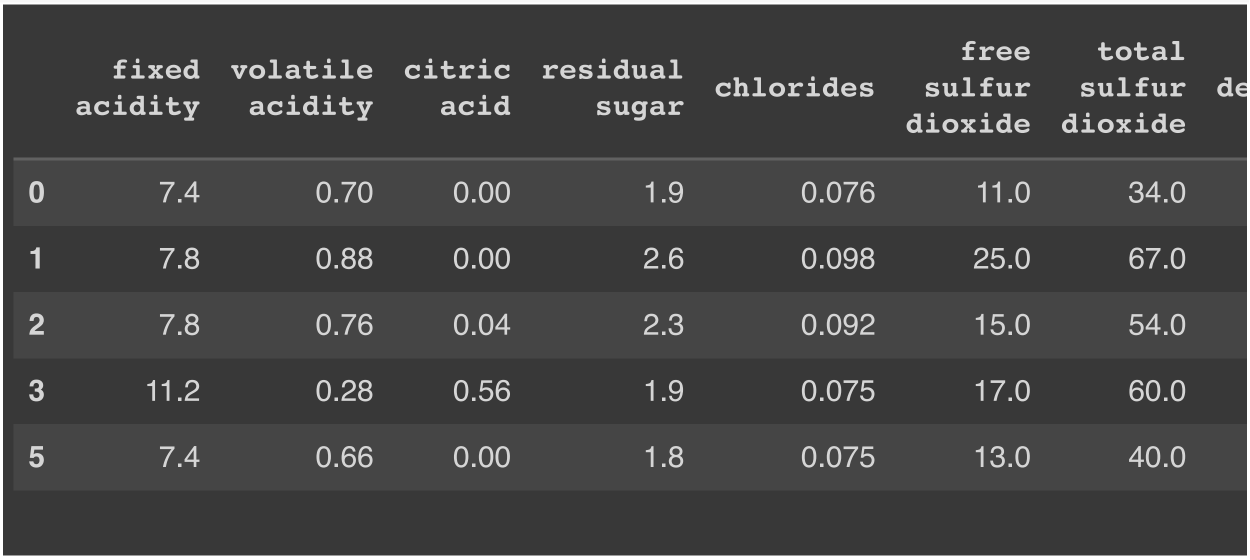
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import IsolationForest
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_sco
```

```
df=pd.read_csv('/content/wine classification.csv')
df
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0
...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0

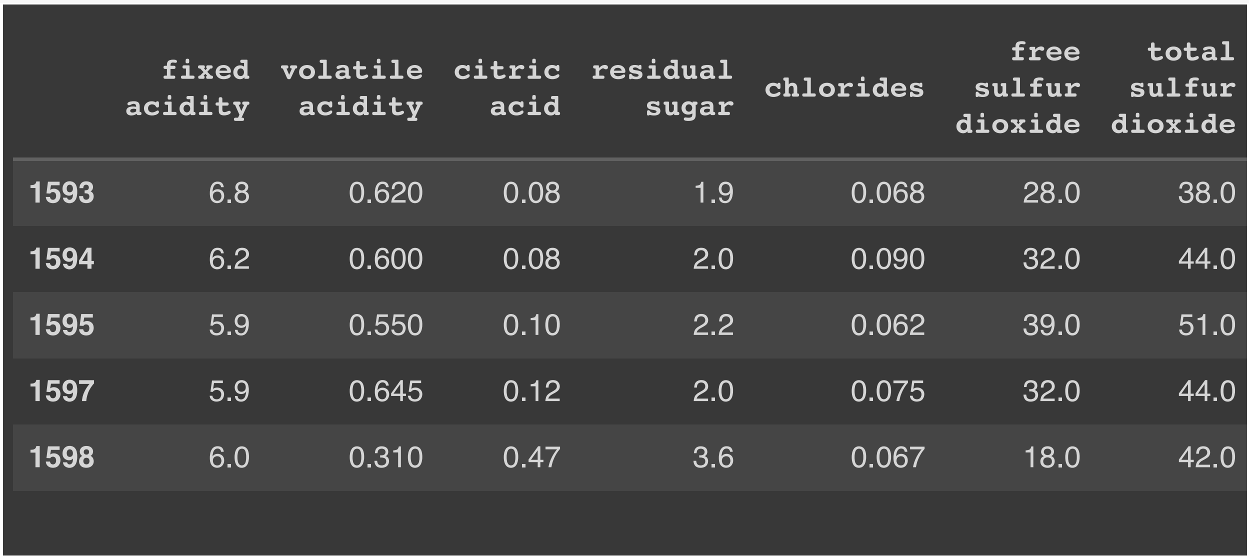
1599 rows × 12 columns

```
df.head()
```



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	de
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	

```
df.tail()
```



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide
1593	6.8	0.620	0.08	1.9	0.068	28.0	38.0
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1360 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1360 non-null   float64
 1   volatile acidity      1360 non-null   float64
 2   citric acid           1360 non-null   float64
 3   residual sugar        1360 non-null   float64
 4   chlorides             1360 non-null   float64
 5   free sulfur dioxide   1360 non-null   float64
 6   total sulfur dioxide  1360 non-null   float64
 7   density               1360 non-null   float64
 8   pH                    1360 non-null   float64
 9   sulphates             1360 non-null   float64
 10  alcohol               1360 non-null   float64
 11  quality               1360 non-null   float64
dtypes: float64(12)
memory usage: 138.1 KB
```

```
print("missing values:")
df.isnull().sum()
```

```
missing values:
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
```

```
print("duplicateed rows")
df.duplicated().sum()
```

```
duplicateed rows
239
```

```
print("removing duplicates")
df.drop_duplicates(inplace=True)
```

```
removing duplicates
```

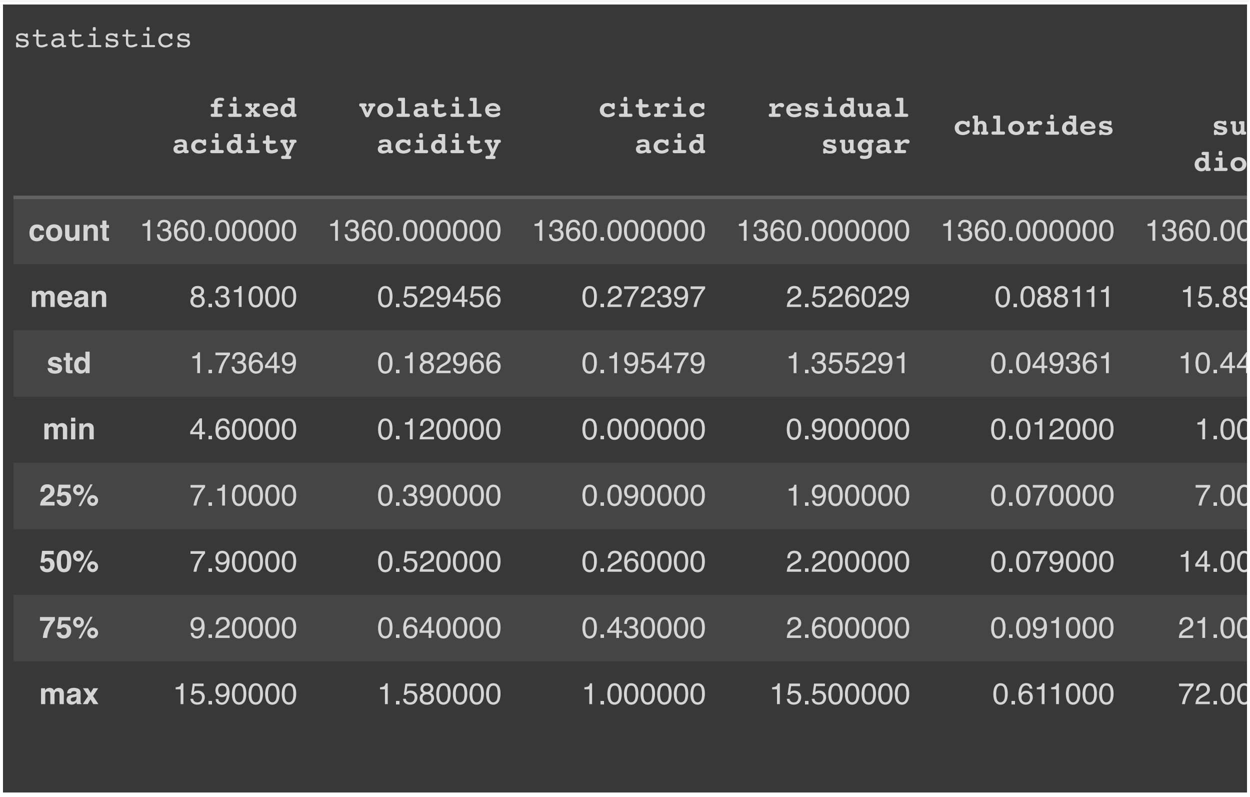
```
print("now we are replacing the missing values with mean")
df.fillna(df.mean(),inplace=True)
```

```
now we are replacing the missing values with mean
```

```
print("duplicateed rows are remove successfully")
df.duplicated().sum()
```

```
duplicateed rows are remove successfully
0
```

```
print("statistics")
df.describe()
```

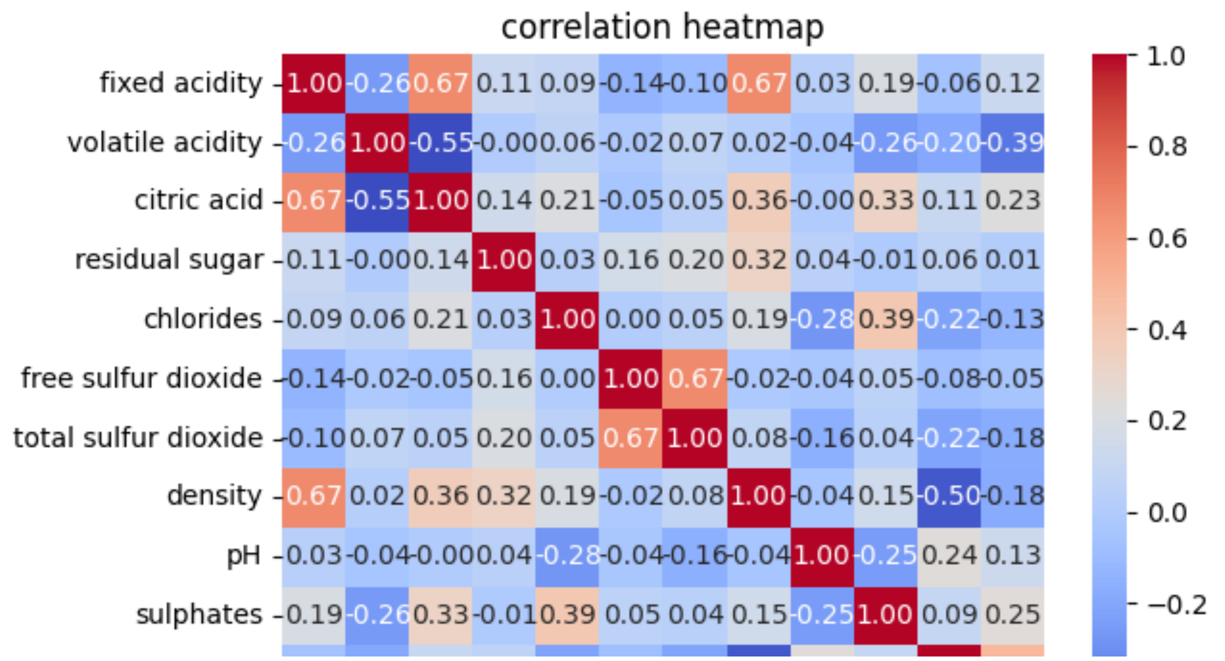


```
statistics
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	su dio
count	1360.00000	1360.000000	1360.000000	1360.000000	1360.000000	1360.00
mean	8.31000	0.529456	0.272397	2.526029	0.088111	15.89
std	1.73649	0.182966	0.195479	1.355291	0.049361	10.44
min	4.60000	0.120000	0.000000	0.900000	0.012000	1.00
25%	7.10000	0.390000	0.090000	1.900000	0.070000	7.00
50%	7.90000	0.520000	0.260000	2.200000	0.079000	14.00
75%	9.20000	0.640000	0.430000	2.600000	0.091000	21.00
max	15.90000	1.580000	1.000000	15.500000	0.611000	72.00

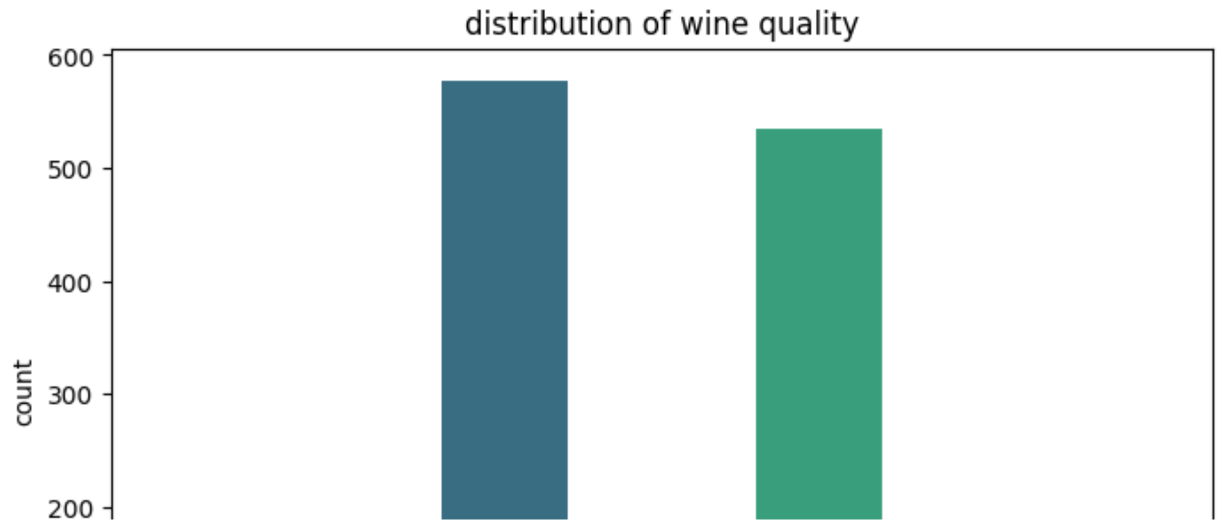
```
print("checking correlation")
sns.heatmap(df.corr(),annot=True,cmap='coolwarm',fmt='.2f')
plt.title('correlation heatmap')
plt.show()
```

```
checking correlation
```



```
print("wine quality")
plt.figure(figsize=(8,5))
sns.countplot(x='quality',data=df,palette='viridis')
plt.title("distribution of wine quality")
plt.xlabel("quality")
plt.ylabel("count")
plt.show()
```

```
wine quality
```



```
print("splitting data for analysis")
x=df.drop(['quality'],axis=1)
y=df['quality']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_s
```

```
splitting data for analysis
```

```
print("isolation forest model")
model=IsolationForest(contamination=0.05,random_state=42)
outlier_predictions=model.fit_predict(x_train)
```

```
isolation forest model
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarn
  warnings.warn(
```

```
eva_df = pd.DataFrame({'y_true': y_train, 'outlier_predictions': outlier_
```

```
print("setting inliers to 0 and outliers to 1")
eva_df['outlier_predictions'][eva_df['outlier_predictions']==1] = 0
eva_df['outlier_predictions'][eva_df['outlier_predictions']==-1] = 1
```

```
setting inliers to 0 and outliers to 1
<ipython-input-54-a6f6bb5131d1>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/panda
  eva_df['outlier_predictions'][eva_df['outlier_predictions']==1] = 0
<ipython-input-54-a6f6bb5131d1>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/panda
  eva_df['outlier_predictions'][eva_df['outlier_predictions']==-1] =
```

```
print("regression evaluation:")
print(f"Mean Absolute Error: {mean_absolute_error(eva_df['y_true'], eva_d
print(f"Mean Squared Error: {mean_squared_error(eva_df['y_true'], eva_df
print(f"R^2 Score: {r2_score(eva_df['y_true'], eva_df['outlier_predictio
```

```
regression evaluation:
Mean Absolute Error: 5.5805
Mean Squared Error: 31.8838
R^2 Score: -45.9369
```

```
wine_data = df.drop_duplicates()
```

```
columns_for_analysis = ['fixed acidity', 'volatile acidity', 'citric acid
                         'free sulfur dioxide', 'total sulfur dioxide',
```

```
data_for_analysis = wine_data[columns_for_analysis]
```

```
# Fit the Isolation Forest model
model = IsolationForest(contamination=0.05, random_state=42)
model.fit(data_for_analysis)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarn
  warnings.warn(
▾                    IsolationForest
IsolationForest(contamination=0.05, random_state=42)
```

```
outlier_predictions = model.predict(data_for_analysis)
```

```
wine_data['outlier'] = outlier_predictions
```

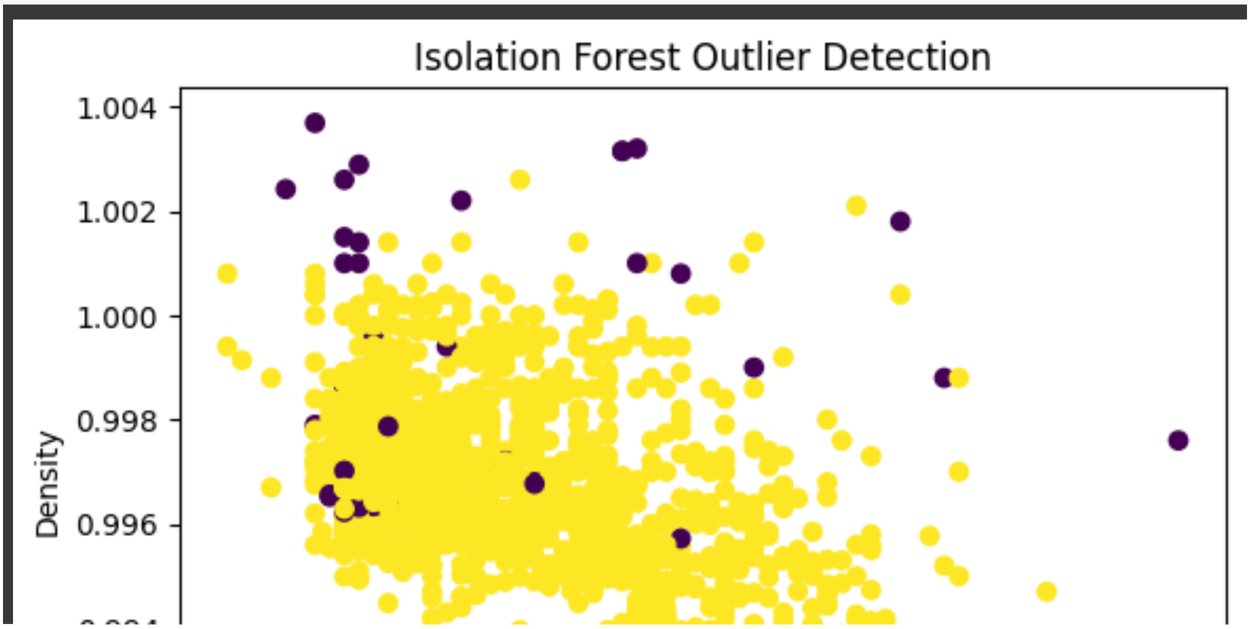
```
outliers = wine_data[wine_data['outlier'] == -1]
print("Outliers (Potentially Excellent or Poor Wines):")
display(outliers)
```

```
Outliers (Potentially Excellent or Poor Wines):
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide
14	8.9	0.620	0.18	3.8	0.176	52.0	145.0
15	8.9	0.620	0.19	3.9	0.170	51.0	148.0
17	8.1	0.560	0.28	1.7	0.368	16.0	56.0
19	7.9	0.320	0.51	1.8	0.341	17.0	56.0
33	6.9	0.605	0.12	10.7	0.073	40.0	83.0
...	...	...	...	...	...	...	...
1370	8.7	0.780	0.51	1.7	0.415	12.0	66.0
1434	10.2	0.540	0.37	15.4	0.214	55.0	95.0
1474	9.9	0.500	0.50	13.8	0.205	48.0	82.0
1558	6.9	0.630	0.33	6.7	0.235	66.0	115.0
1574	5.6	0.310	0.78	13.9	0.074	23.0	92.0

68 rows × 13 columns

```
plt.scatter(data_for_analysis['alcohol'], data_for_analysis['density'], c
plt.title('Isolation Forest Outlier Detection')
plt.xlabel('Alcohol')
plt.ylabel('Density')
plt.show()
```


Isolation Forest Outlier Detection
Density