HW2

Note：数据集采用winequality-red.csv。对于winequality-white.csv的分析，与前者相似。

Data Cleaning

Missing Value Handling: Identify and deal with missing values in the data.

Code:

```
null_values = df.isnull()
null_values.sum()
df.dropna(inplace=True)
```

使用isnull函数检查是否存在空值，并用sum函数观察每一列中空值的数量。如果有空值，将会用该值所在列的平均值替代。

结果：

```
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
```

发现数据集中并没有空值。

Duplicate Data Handling: Identify and deal with duplicate values in the data.

Code:

```
print(f'去除重复元素前df的形状: {df.shape}')
df.drop_duplicates(inplace=True)
print(f'去除重复元素后df的形状: {df.shape}')
```

数据集中存在重复值，去除重复行的前后结果如下：

```
去除重复元素前df的形状: (4898, 12)
去除重复元素后df的形状: (3961, 12)
```

Data Integration

Code:

```
df['total acidity'] = df['fixed acidity'] + df['volatile acidity']
df.head(3)
```

前三行的结果如下:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	total acidity
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6	7.27
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6	6.60
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6	8.38

Data Transformation

Normalization: Normalize the “quality” data to the [0,1] range

Code:

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler


# 初始化 MinMaxScaler
scaler = MinMaxScaler()
# Fit scaler 到数据并转换
df['quality'] = scaler.fit_transform(df[['quality']])
df.head(3)
```

创建了一个`MinMaxScaler`对象，并将其赋值给变量`scaler`。`MinMaxScaler`是一个用于将特征缩放到给定的最小值和最大值之间的工具，通常是0和1之间。归一化可以提高算法的性能，特别是当不同特征的数值范围相差很大时。

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	total acidity
0	7.0	0.27	0.36	20.70	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	0.500000	7.27
1	6.3	0.30	0.34	1.60	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	0.500000	6.60
2	8.1	0.28	0.40	6.90	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	0.500000	8.38
3	7.2	0.23	0.32	8.50	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	0.500000	7.43
6	6.2	0.32	0.16	7.00	0.045	30.0	136.0	0.9949	3.18	0.47	9.6	0.500000	6.52
9	8.1	0.22	0.43	1.50	0.044	28.0	129.0	0.9938	3.22	0.45	11.0	0.500000	8.32
10	8.1	0.27	0.41	1.45	0.033	11.0	63.0	0.9908	2.99	0.56	12.0	0.333333	8.37
11	8.6	0.23	0.40	4.20	0.035	17.0	109.0	0.9947	3.14	0.53	9.7	0.333333	8.83
12	7.9	0.18	0.37	1.20	0.040	16.0	75.0	0.9920	3.18	0.63	10.8	0.333333	8.08
13	6.6	0.16	0.40	1.50	0.044	48.0	143.0	0.9912	3.54	0.52	12.4	0.666667	6.76

Discretization: Discretize the continuous attribute “fixed acidity” into three levels: “low,” “medium,” and “high.”

Code:

```
import pandas as pd

# 假设 df 是你的 DataFrame 并且 'fixed acidity' 是其中的一个连续属性列
# 使用 qcut 自动定义阈值，并将 'fixed acidity' 离散化为三个等级
df['fixed acidity level'] = pd.qcut(df['fixed acidity'], 3, labels=['low',
'medium', 'high'])
df.head(3)
```

新增一列fixed acidity level，用于把连续变量fixed acidity离散化。

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	total acidity	fixed acidity level
7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	0.5	7.27	medium
6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	0.5	6.60	low
8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	0.5	8.38	high

Data Reduction

Code:

```
import pandas as pd
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif

X = df.drop(['quality','fixed acidity level','quality_normalize'], axis=1)  #
feature columns
y = df['quality']  # target column

# Feature selection using ANOVA F-test
selector = SelectKBest(score_func=f_classif, k=3)
X_new = selector.fit_transform(X, y)

# Get the indices of the features that were selected
selected_indices = selector.get_support(indices=True)

# Get the feature names based on the indices
selected_features = [X.columns[i] for i in selected_indices]

print('The top three features that have the most significant impact on the
quality rating of wine are:')
for feature in selected_features:
    print(feature)
```

通过单因素方差分析的F值比较，对酒类quality影响力最大的三个变量分别是：volatile acidity、density和alcohol。

```
The top three features that have the most significant impact on the quality rating of wine are:
volatile acidity
density
alcohol
```