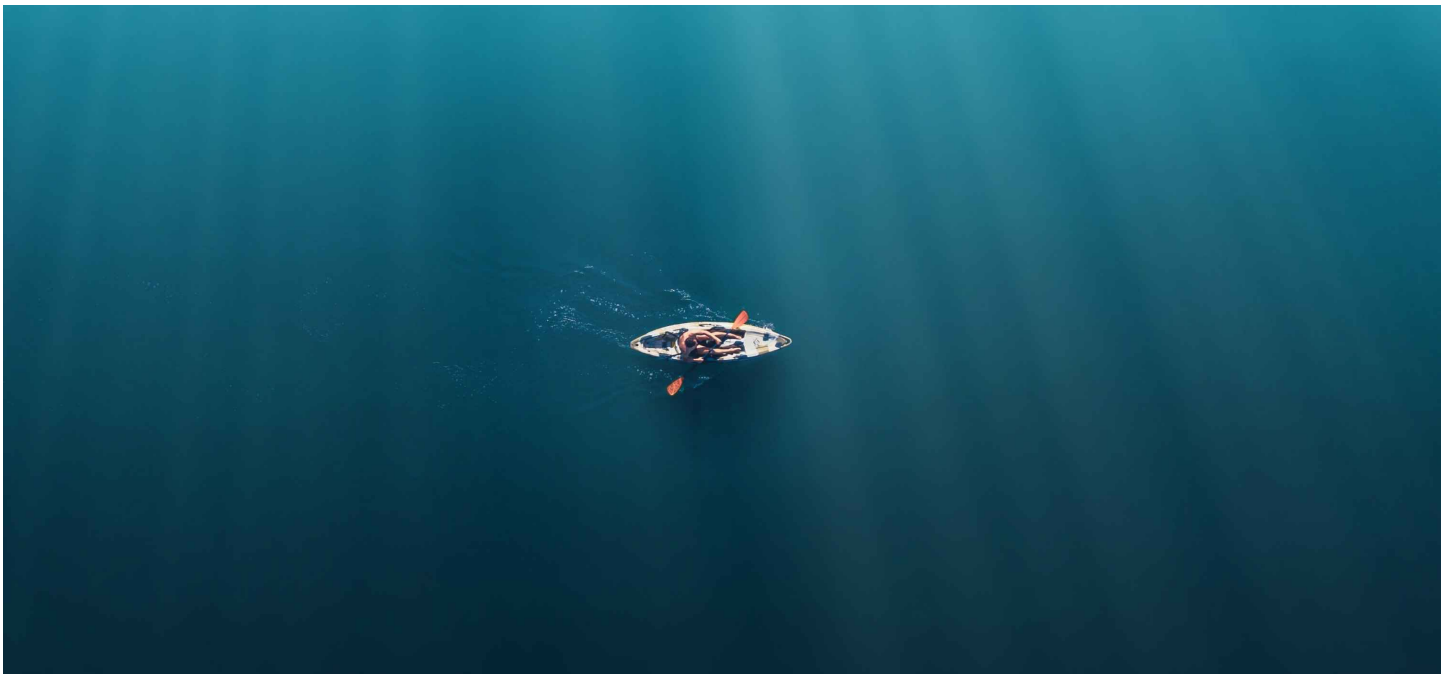

Assignment 2: Data Pre-processing

2051498 储岱泽

任务目标

本次作业的目标是熟悉数据预处理的基本步骤，包括数据清洗（Data Cleaning）、数据集成（Data Intergration）、数据转换（Data Transformation）和数据降维（Data Reduction）。通过实践，将加深对数据预处理重要性的理解，并掌握数据预处理的基本技能。

数据集来源

本次作业使用的数据集是从著名的UCI机器学习仓库中获取的“葡萄酒质量”数据集。该数据集包含葡萄酒的理化性质和质量评分，适用于数据预处理和后续分析。

数据集链接：

https://archive.ics.uci.edu/dataset/186/wine+quality

数据处理

数据导入

首先，我们使用ucimlrepo库将该数据集中的数据导入项目之中，并且将该数据集中所包含的特征和目标变量合并成一个dataframe，以方便后续对数据的处理。

```python
from ucimlrepo import fetch_ucirepo
wine_quality = fetch_ucirepo(id=186)
# data (as pandas dataframes)
X = wine_quality.data.features
y = wine_quality.data.targets

# 将特征和目标变量合并为一个DataFrame
wine_data = pd.concat([X, pd.DataFrame(y, columns=['quality'])], axis=1)
```

数据清洗（Data Cleaning）

接下来，我们对数据进行清洗，主要分为两个步骤：对缺失值的处理以及对重复值的处理。

1. 缺失值处理（Data Missing）

在我的程序中，我首先计算了每一列缺失值的数量，查看缺失值在dataframe中的一个基本的情况。

```python
# 计算每列缺失值的数量
missing_values = wine_data.isna().sum()
# 打印每列缺失值的数量
print("每列缺失值的数量：")
print(missing_values)
```

```
每列缺失值的数量：
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
```

然后，我们查看输出的缺失值的情况，发现在dataframe中没有发现缺失值，所以我们暂且不需要对其进行处理。

不过，假如说我们发现dataframe中存在缺失值，我们也可以用下面的语句将其去除。

```python
wine_data.dropna(inplace=True)
```

2. 重复值处理（Duplicate Data Handling）

对于重复值，我们简单的将其删去即可。

```
wine_data.drop_duplicates(inplace=True)
```

数据集成（Data Intergration）

按照作业中的要求，我们，我们将计算“总酸度（total_acidity）”，即“固定酸度（fixed_acidity）”和“挥发性酸度（volatile_acidity）”的总和，并将其作为新列添加到数据集中。

```
# 计算总酸度并添加为新列
wine_data['total_acidity'] = wine_data['fixed_acidity'] + wine_data['volatile_acidity']
print("数据的前几行：")
print(wine_data.head())
```

```
数据的前几行：
   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
5            7.4              0.66         0.00             1.8      0.075

   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
5                 13.0                  40.0   0.9978  3.51       0.56

   alcohol  quality  total_acidity
0      9.4        5           8.10
1      9.8        5           8.68
2      9.8        5           8.56
3      9.8        6          11.48
5      9.4        5           8.06
```

数据转换（Data Transformation）

1. 标准化（Normalization）

将“质量（quality）”数据标准化到[0,1]范围内。

1. **数据统一范围：**标准化可以使得不同特征的数据处于相同的量纲范围内，这有助于比较不同特征对目标的影响程度。在这个作业中，将质量数据标准化到[0,1]范围内，可以确保所有质量值都在统一的区间内，方便进行后续分析和比较。
2. **模型训练优化：**一些机器学习算法对数据的范围敏感，如果数据的范围不一致，可能会影响模型的性能和收敛速度。通过将质量数据标准化到[0,1]范围内，可以帮助机器学习模型更快地收敛，并提高模型的稳定性和性能。

```
scaler = MinMaxScaler()
wine_data['quality_normalized'] = scaler.fit_transform(wine_data[['quality']])
wine_data.head()
```

citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality	total_acidity	quality_normalized
0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	8.10	0.333333
0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	8.68	0.333333
0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	8.56	0.333333
0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	11.48	0.500000
0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5	8.06	0.333333

2. 离散化（Discretization）

然后，我们可以对得到的数据进行离散化，将连续属性“fixed acidity”离散化为三个级别：“低”、“中”和“高”。

在进行离散化之前，为了尽可能的把数据在“低”，“中”和“高”三个级别上分布的均匀，我们可以先查看一下fixed acidity的最小值和最低值，以及三分之一数量的分界点和三分之二分界点的位置。

```
# 使用describe()函数查看统计信息
fixed_acidity_stats = X['fixed_acidity'].describe()

# 输出结果
print("最小值:", fixed_acidity_stats['min'])
print("最大值:", fixed_acidity_stats['max'])
print("三分之一分位点:", X['fixed_acidity'].quantile(1/3))
print("三分之二分位点:", X['fixed_acidity'].quantile(2/3))
```

```
最小值: 3.8
最大值: 15.9
三分之一分位点: 6.6
三分之二分位点: 7.4
```

于是，我们可以按照以下方式进行三个等级的划分：

1. **low：**[0 , 6.6]
2. **medium：**[6.6 , 7.4]

3. **high**：[7.4 , 16]

```
# 2. 离散化
bins = [0, 6.6, 7.4, 16]  # 设置分箱边界
labels = ['low', 'medium', 'high']  # 设置分箱标签
wine_data['fixed_acidity_discretized'] = pd.cut(wine_data['fixed_acidity'], bins=bins, labels=labels)
wine_data.head()
```

gar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality	total_acidity	quality_normalized	fixed_acidity_discretized
1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	8.10	0.333333	medium
2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	8.68	0.333333	high
2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	8.56	0.333333	high
1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	11.48	0.500000	high
1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5	8.06	0.333333	medium

数据降维（Data Reduction）

数据降维的目的是通过减少数据特征的数量，从而减少数据集的复杂度，提高模型的效率和性能。

在我们这个wine_quelity的问题当中，我们主要研究的是酒的各种特征对酒的质量的影响，从而找到用来衡量酒的特征的最佳属性。所以我们可以使用方差分析（ANOVA）选择对葡萄酒品质评级影响最显著的前三个特征。

ANOVA（即方差分析）是一种统计方法，用于比较三个或三个以上组之间的平均值是否存在显著差异。它基于对数据的方差进行分解，将总体方差分解为组内方差和组间方差。通过比较组间方差与组内方差的比值来确定组之间的差异是否显著。

在特征选择中，ANOVA可以用于确定哪些特征对目标变量有显著影响。通过计算每个特征与目标变量之间的**F统计量和p值**，可以评估特征之间的显著性差异。

通常，具有较大F统计量和较小p值的特征被认为对目标变量具有显著影响，因此可以被选择为最重要的特征。

- F值越大，说明该特征对因变量的解释程度越高，即影响越显著
- P值越小，表示观察到的数据与零假设的偏差越大，即影响也越显著。

而在计算F与p的值的时候，sklearn库中有两种关于ANOVA分析计算F值来筛选特征的函数，一个是f_classif，一个是f_oneway，通过查阅资料，我们对这两个函数的使用场景进行了总结：

1. **f_oneway：**
    - **用途：**主要用于执行单因素方差分析，适用于比较两个或多个组之间的均值是否有显著差异。
    - **参数：**接受一系列数组，每个数组代表一个组的数据。
    - **返回值：**返回F统计量和p值，用于判断组之间的均值是否有显著差异。
2. **f_classif：**

- **用途：**主要用于特征选择，通过计算各个特征的ANOVA F统计量和p值来评估特征与目标变量之间的相关性。
- **参数：**接受特征矩阵和目标变量，用于计算特征与目标变量之间的F统计量和p值。
- **返回值：**返回各个特征的F统计量和p值，可以用于选择对目标变量影响显著的特征。

于是，通过分析，我们可以引入 `sklearn.feature_selection` 库中的 `f_classif` 函数来进行计算。

下面是我的计算过程：

```python
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif

# 创建特征选择器
selector = SelectKBest(f_classif, k=3)

# 使用特征选择器来拟合并转换 x
senew = selector.fit_transform(X, y)

# 打印被选择的特征
mask = selector.get_support()
new_features = X.columns [mask]
print(new_features)

# 打印所有特征的F 值
for feature, score in zip(X.columns, selector.scores_):
    print(f"Feature: {feature}, Score: {score}")
```

```
Index(['volatile_acidity', 'density', 'alcohol'], dtype='object')
Feature: fixed_acidity, Score: 8.004192723298823
Feature: volatile_acidity, Score: 96.67402209622112
Feature: citric_acid, Score: 9.31019870644853
Feature: residual_sugar, Score: 9.111366826994608
Feature: chlorides, Score: 50.84971882802702
Feature: free_sulfur_dioxide, Score: 14.939170402675012
Feature: total_sulfur_dioxide, Score: 7.716088194433556
Feature: density, Score: 136.95123571943913
Feature: pH, Score: 2.021462044859375
Feature: sulphates, Score: 4.325772856450053
Feature: alcohol, Score: 320.59344780302115
```

综上所述，我们筛选出的对酒的影响最显著的三个特征分别是：**volatile_acidity、density和alcohol**。