[](http://www.swufe.edu.cn/version3/xuexiaogaikuang/xiaohui.jpg)**西南财经大学**

Southwestern University of Finance and Economics

**课程论文**

|  |  |
| --- | --- |
| **学年学期：** | **2022-2023学年第1学期** |
| **课程名称：** | **数据挖掘** |
| **论文题目：** | **基于历史数据的白葡萄酒品质预测** |
| **学生学号：** | **42051089** |
| **学生姓名：** | **廖晨皓** |
| **学 院：** | **工商管理学院** |
| **年级专业：** | **2020级市场营销（金融服务与营销）** |

|  |
| --- |
| 评语：  **得 分：**  **评阅教师签字： 年 月** |

# 一、数据概况与实验目的

本次实验的数据来自加州大学尔湾分校所提供的机器学习数据仓库中所提供的白葡萄酒和“Vinho Verde”葡萄酒化学分析数据 (UCI Machine Learning Repository, 1987)。本数据集最初来自于 Cortez et al. (2009) 的论文中。本数据集一共包含数据4898条、字段12个，其中有以1-10分对酒品质量进行评价的量化分数以及11个记录酒体不同理化成分的字段。

在本实验中，我将利用这一数据集，通过不同模型对数据进行处理并对结果进行比较，以期通过相应的理化性质对酒体质量进行相对准确的预测之目的。

需要说明的是，本次实验的代码均已上传Github。为优化版式，正文中可能将仅提供展示思路的“伪代码”而不再提供全部代码。完整可运行的代码请在[Github仓库](https://github.com/LisPerfect/DataMiningExp)中进行下载。

# 二、数据预处理

在数据挖掘中，数据预处理流程十分重要。通过适当的预处理后的数据才能在应用中呈现更优的质量。

在数据预处理中，数据清洗是其中的很大一部分工作。数据清洗（data cleaning）主要通过填补数据集中的缺失值，平滑噪声数据，处理异常值并解决数据不一致性等问题来处理“脏”数据，以达到“清洗”数据的目的。在数据清洗中常见的操作包括删除无关或重复的数据、平滑噪声、处理缺失、处理异常值等操作。

首先，我们检查数据集是否存在重复或空缺数据。

通过引入Pandas库并读取数据集后使用Pandas所提供的函数对数据集进行去重、去缺失值的处理。

|  |
| --- |
| //以下函数来自Pandas库  data.drop\_duplicates(inplace=True)  data.dropna(inplace=True)#参数inplace=True代表直接在原DataFrame上操作 |

通过这两个函数的操作后，发现剩余数据有3954条，仍是一个较大的数据量，理应满足我们接下来分析的需要。因此，为保证模型质量，此处对缺失值直接做删除处理，不再保留或进行插值。

对于异常值分析，此处通过计算每一列的Z-score，然后对出现超过3的行进行整行抛弃的做法来进行。

Z-score是统计学中的一个无量纲量，其计算公式为 ，其中为需要被标准化的原始值、为总体的平均值、为总体的标准差。Z值的量代表着原始分数和总体平均值之间的距离，是以标准差为单位计算。在原始分数低于平均值时Z则为负数，反之则为正数。换句话说，Z值是从感兴趣的点到均值之间有多少个标准差。在此我们认为Z-score绝对值大于3的值为异常值，应当予以提出。“3个标准差” (又称为3-Sigma)这一标准，是在大量文献和实践中被广泛接受的一个标准，在3个标准差之内包括了99.7%的数据。

再次说明，同本实验的其他代码一样，这一部分的代码仍然已[上传至Github](https://github.com/LisPerfect/DataMiningExp)，此处仅展示部分核心思路。

|  |
| --- |
| //Preprocessing.py  # 通过Z-Score方法判断异常值  df\_zscore = data.copy()  # 复制一个用来存储Z-score得分的数据框  cols = data.columns  #  获得列表框的列名  for col in cols:      df\_col = data[col]  #  得到每一列的值      z\_score = (df\_col - df\_col.mean()) / df\_col.std()  #  计算每一列的Z-score得分      df\_zscore[col] = z\_score.abs() > 3  # 判断Z-score得分是否大于3，如果是则是True，否则为False  df\_drop\_outlier = data.copy()  for col in cols:      df\_drop\_outlier = data[df\_zscore[col] == False] # 丢弃这一列所有Z-score>3的值，通过for循环达到“清除每一列”的目的  print(df\_drop\_outlier) #观察是否达到清除异常值的目的 |

在经过异常值检测后，Z-score小于3的数据有3929条。本次实验接下来的部分将以这一经过预处理后的数据集为基础来进行。

自此，我们便完成了数据清理工作。

由于一些模型对输入的数据有特定的格式要求，因此数据预处理的一个重要步骤就是数据变换。数据变换是将数据从一种格式转换为另一种格式的过程。主要是对数据进行规范化的操作，将数据转换成“适当的”格式，以适用于挖掘任务及算法的需要。

目前，Python中对于数据变换已有相对成熟的第三方库供使用。

|  |
| --- |
| //Preprocessing.py  from sklearn import preprocessing  from sklearn.preprocessing import StandardScaler |

一般而言，涉及到“空间”、“距离”相关的算法，都需要对数据进行标准化和归一化。因此，我们在此处提前对数据进行这一处理，以备后用。

|  |
| --- |
| //Preprocessing.py  //此处复用上文中的检测异常值小节曾使用过的代码，只不过是将其用途转化为计算并保存Z-score到本地CSV文件  for col in cols:      df\_col = df\_drop\_outlier[col]      z\_score = (df\_col - df\_col.mean()) / df\_col.std()      df\_zscore[col] = z\_score  df\_zscore.to\_csv('./z-score.csv') |

对于完成数据清洗但未进行数据变换的数据集，此处将其存于本地，以备后用

|  |
| --- |
| //Preprocessing.py  df\_drop\_outlier.to\_csv('after\_preprocessing.csv') |

到此，我们对于数据的预处理步骤就基本完成。

# 三、模型提出与检验

# 四、数据可视化

# 总结

# 参考文献

*P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.*

*UCI Machine Learning Repository: Wine Data Set*. (1987). http://archive.ics.uci.edu/ml/datasets/Wine

# 附录