华中科技大学

计算机科学与技术学院

《机器学习》结课报告

专　　业：	计算机科学与技术
班　　级：	CS2205
学　　号：	U202215650
姓　　名：	严浩洋
成　　绩：	
指导教师：	何琨

完成日期：2024 年 5 月 10 日

目录

1. 实验要求

本次实验中我们需要自行编写代码，构建模型，使用机器学习算法对模型进行训练和预测，实现对肥胖风险的多分类预测。

本次实验的数据集是从“肥胖与心脑血管疾病风险”数据集中训练的深度学习模型生成的，包括 train.csv，test.csv 两部分。

训练集 train.csv 中，内容如表 1 所示：

表 1 训练集数据内容与格式

表项	内容	类型
id	唯一编号	整数
Gender	性别，Male 或 Female	字符串
Age	年龄，用年表示	浮点数
Height	身高，推测单位为“米”	浮点数
Weight	重量，推测单位为“千克”	浮点数
family_history_with_overweight	家族是否有超重史，yes 或 no	字符串
FAVC	高热量食物消费频率高或低，yes 或 no	字符串
FCVC	蔬菜消费频率，介于 1-3 之间	浮点数
NCP	主餐数，介于 1-4 之间	浮点数
CAEC	两餐之间进食的频率，取值为 Always 等	字符串
SMOKE	是否抽烟，yes 或 no	字符串
CH2O	每日摄入饮用水量，介于 1-3 之间	浮点数
SCC	是否进行热量消耗监控，yes 或 no	字符串
FAF	体育活动频率，介于 0-3 之间	浮点数
TUE	使用技术设备追踪健康情况的频率，0-2 之间	浮点数
CALC	饮酒量，取值为 Sometimes 等	字符串
MTRANS	使用的交通工具，取值为 Automobile 等	字符串
NObeyesdad	肥胖类型，包括 Normal_Weight 等	字符串

其中，NObeyesdad 是数据的标签，作为分类的目标。test.csv 中除标签列缺失外，其余内容与 train.csv 一致。

本次实验中我们将从以上特征出发，构建机器学习模型进行训练，以达到肥胖风险类别的预测。

2. 算法设计与实现

(1) 数据预处理与可视化

数据集中每一个对象包含的特征过多，高达十几种，若均用于作为训练的特征，可能会导致训练时间过长，且效果未必能达到最好。因此，我们需要分析各项特征之间的相关程度，选择最为合适的特征。考虑使用 Pearson 相关系数进行分析。

Pearson 相关系数由如下公式计算：

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

取值范围为[-1,1]，1 为标准的正比例，-1 与之相反。

由于数据集中有一部分项目为字符串，无法直接计算相关系数，因此需要先对数据进行预处理。由于表项中字符串内容均为对选择性问题的回答，取值有限，我们可以按照程度由低到高将字符串赋予对应的整型值。

以对标签 NObeyesdad 重新赋值为例，Insufficient_Weight 意味“体重不足”，程度最低，赋值为 0；Overweight_Level_II 为“超重等级 II”，肥胖程度最高。按照这种方式，其对应关系如表 2 所示：

表 2 NObeyesdad 表项的重新赋值对应关系

原值	对应值
Insufficient_Weight	0
Normal_Weight	1
Obesity_Type_I	2
Obesity_Type_II	3
Obesity_Type_III	4
Overweight_Level_I	5
Overweight_Level_II	6

基于以上对应关系即可将字符串类型的值转化为整型值，其余表项类似。

预处理完成后，即可算出相关系数，并画出热力图，如图 1 所示：

从图 1 中我们可以看出，最后一列中 NObeyesdad 与体重 Weight 相关度最高，相关系数高达 0.92。但根据现实情况分析，一个人的肥胖状况不能只用体重衡量，因为体重大小也与身高有关。从热力图中我们也可以看出：体重与身高的相关系数为 0.42。因此考虑使用身高和体重两个维度，对肥胖程度的大小进行分析和预测。

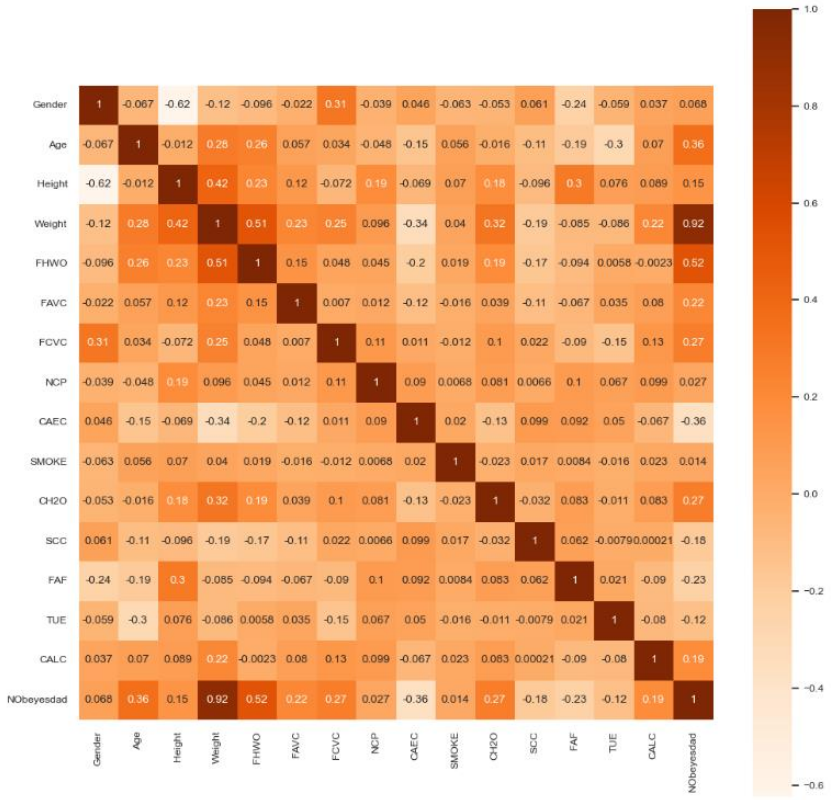

图 1 各项特征 Pearson 相关系数的热力图

利用身高，体重二维坐标和 NObeyesdad 标签，可以画出双变量关系图，如图 2 所示。

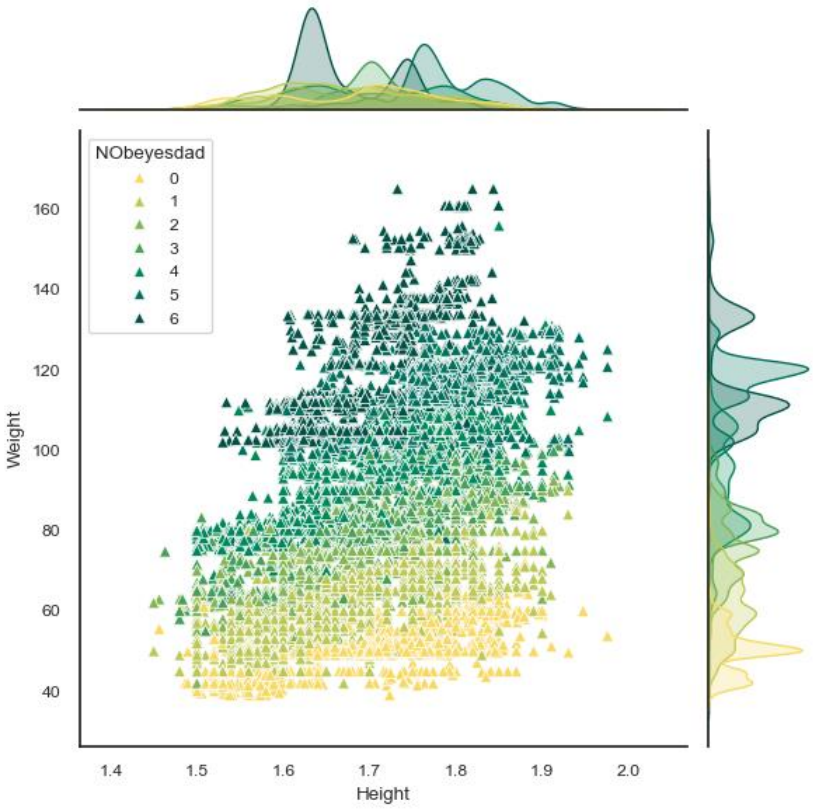

图 2 身高、体重关于肥胖程度的双变量关系图

如图 2 所示，利用身高体重二维坐标画出的双变量关系图，各类肥胖程度的样本聚集分布在邻近的位置，因此使用身高、体重两个维度进行分析具有合理性和可行性。

(2) 基于 Logistic 回归的多分类模型

首先考虑使用 Logistic 回归对数据进行处理。由于使用 sigmoid 函数只能进行二分类。我们需要对其改造以进行多分类预测。

数据预处理中，我们将肥胖类别转化为 0-6 的整型值，因此函数也应在相应范围内取值。因此可将 sigmoid 函数改造为如下的函数：

$$f(x)=\frac{6}{1+e^{-x}} \tag{2}$$

使用这一函数作为归一化函数，可将输入值转化为 0-6 之间的值，用于模拟对类别的预测。其中对应关系如表 3 所示：

表 3 预测值与目标类别的对应关系

目标值	预测值
Insufficient_Weight	<0.5
Normal_Weight	0.5≤x<1.5
Obesity_Type_I	1.5≤x<2.5
Obesity_Type_II	2.5≤x<3.5
Obesity_Type_III	3.5≤x<4.5
Overweight_Level_I	4.5≤x<5.5
Overweight_Level_II	≥5.5

对于模型的构建，我们可构建一个三元函数，用于表示各项特征与肥胖类别之间的关系：

$$f(x)=w_0x_0+w_1x_1+w_2x_2 \tag{3}$$

其中，x_0=1，另外两个自变量分别表示身高和体重。使用梯度下降法[1]对参数进行优化，有：

$$w_i=w_i-\eta(a-y)x_i \tag{4}$$

其中，η表示学习率，a 为肥胖类别的预测值，y 为肥胖类别的实际值。

根据以上分析，即可编写代码对肥胖程度进行预测，但得到的准确率较低，均在 0.15 左右。综合分析后发现准确率较低的现象可能与原始特征数据有关。身高特征的取值在 1.5 左右，而体重的取值可高达 150，这会导致参数的取值范围不同，对精度的要求也不同，因此考虑对特征变量数据进行再次进行预处理。

考虑到身高的均值约为 1.5，体重的均值约为 80，将原始数据均除以对应的均值，即可使特征数据均分布在(0,3)之间，对精度要求的差距变小。

利用处理后的数据再次进行训练，这次的准确率较未处理的结果提升明显，准确率在 0.35 左右。

图 3 Logistic 回归的最终预测准确率

但无论怎么调整参数或增加特征，预测率均无较大改善。这可能与归一化函数的构造有关，也可能与 Logistic 回归对多分类问题的支持不佳有关。因此我们考虑更换模型进行预测。

(3) 基于 KNN 的多分类预测模型

从图 2 双变量关系图中我们可以看出，使用身高、体重二维坐标时，相同类别的样本点均分布在邻近的位置，这恰好与 K 近邻算法的定义接近，我们只需计算每一个预测集中的身高、体重坐标与训练集中最近的 K 个样本的类别，并取数量最多的类别即可。

具体地，我们定义一个 knn 函数，每次输入一个测试集中的二维坐标，分别计算其与训练集各样本点的欧氏距离[2]：

$$d_i = \sqrt{(test[0] - train[i][0])^2 + (test[1] - train[i][1])^2} \tag{5}$$

然后将其进行排序，得到距离最小的 K 个点的索引，根据索引找到这 K 个点的类别，以数量最多的类别作为这一测试样本的预测值。

取 k=3，运行代码进行预测，得到的预测准确率为 0.77，相较于 Logistic 回归有极大的提升。因此考虑进行优化。

首先尝试按照上一小节 Logistic 回归中的方式，对特征数据进行近似归一处理，利用处理后的数据进行训练，得到的准确度大幅降低。这可能与身高、体重与肥胖程度的相关程度不同有关。

由图 1 热力图可知，肥胖程度与体重的相关程度较高，与身高关系较小。而数据集中体重均值大于身高，在 knn 计算距离时权重较大，与相关程度相吻合。从图 2 中我们也可以看出，各类样本点根据体重不同由上到下呈现“分层”的趋势，若进行归一化后，原本按层分散的样本点会变得更为紧凑，不利于分类预测。因此简单地照搬上一节中的方法不可行。

然后考虑调整算法参数。由于 KNN 算法中只有一个参数 k，因此我们可以考虑使用遍历的方式找到准确率最高的 k 值。由于计算距离和排序需要较长的时

间，因此我们截取一小部分训练集进行处理。

在[1,100]内遍历 k，发现在 k=10 左右准确率较高。令 k=10，重新训练并预测，最终得到的预测的准确率为 0.85 左右，相较 Logistic 回归预测的结果有大幅提升。提交结果如图 4 所示：

图 4 KNN 最终预测准确率

3. 实验环境与平台

(1) 实验环境

硬件环境：PC 机，AMD Ryzen 7 5800H 3.20GHz，内存 32GB；

操作系统：Microsoft Windows 11 23H2，64 位操作系统；

算法运行平台：Python 3.10.11 64-bit。

(2) 调用开源库

在 KNN 算法和 Logistic 回归模型中，使用了 numpy 用于数组的处理和排序算法的实现，使用了 pandas 用于读取 csv 文件中的数据和生成预测结果 csv 文件。源代码参考资料包中 knn.py 和 Logi.py

在数据可视化文件中，除上述库外，还调用了 seaborn 和 matplotlib 用于绘制热力图和双变量关系图。源代码参考 visible.ipynb。

(3) 提交平台

本实验中预测的结果均提交至 kaggle.com 进行检测和打分。其中，kaggle 是基于“准确度”对预测结果打分的。

4. 结果与分析

从第 2 节中可以看出，KNN 算法相较于 Logistic 回归在预测准确率方面有明显优势。因此在这一节中，我们先基于 KNN 算法预测的结果进行一系列分析，然后对比分析两个算法在本次实验中的优劣。

(1) 召回率分析

本次实验是对肥胖风险的预测，目的为根据每个样本的一系列已知的特征，预测其肥胖的程度。因此，对肥胖人群实现“应检尽检”，即对于原本是肥胖的“正样本”，应尽可能预测出其为肥胖人群。所以可用召回率来评价模型的好坏。

召回率定义为：

$$召回率=\frac{TP}{TP+FN} \tag{6}$$

其中，TP 为真正类样本数，FN 为假负类样本数。

对于本次实验的数据集，我们将 Insufficent_Weight 和 Normal_Weight 视为体重正常的负样本，Obesity_Type_I 及肥胖程度更高的类别视为正样本。将标签已知的训练集 train.csv 利用 train_test_split 分为训练集和测试集，利用 KNN 算法进行训练，得到预测数据，并与原始标签比较，若均为正，则为预测成功。

最终得到 KNN 算法的召回率为 0.979，效果良好。

(2) 稳健性分析

使用 F1 score[3]对模型的稳健性进行分析，其计算公式为：

$$F1=2\times\frac{recall\times precision}{recall+precision} \tag{7}$$

其中 recall 为召回率，precision 为精确率。F1 实际上是召回率和精确率的一个平均，召回率反映了对正样本的识别能力，精确率反映了对负样本的识别能力。因此 F1 值反映了模型的稳健性。

最终得到 KNN 算法的召回率为 0.980。可见模型具有良好的稳健性。

(3) 模型对比分析

在 Logistic 回归模型中，预测准确率不高的主要原因是归一化函数的性能较差。由于 sigmoid 函数不支持多分类，本模型中使用的函数是由 sigmoid 函数直接更改乘上类别数实现的，这一改造更改了函数的值域，但忽视了函数的映射关系。其次，在从预测值映射到对应类别时，每个类别也只是根据取值范围直接映射，且每个类别对应的范围大小不同。

因此，更改更为合适的归一化函数，细化预测值到最终类别的映射，可以提高模型预测的准确率。

在 KNN 算法中，预测率较高的主要原因是其本质与本次实验的数据集适合程度较好。选择身高、体重维度的二维坐标后，样本点按照类别实现了分层分布，相邻的样本点类别一致。但使用 KNN 进行预测的缺点是时间效率较低。由于训练集样本规模较大，需要计算距离并排序，对于每一个测试样本，时间复杂度为 O(nlogn)，在 Python 环境下运行时间较长，这给后期参数调优、模型分析等提高了难度，也不利于更大样本的预测。

5. 个人体会

本次实验中，我们综合使用了机器学习相关算法，对肥胖风险进行了多分类预测。最终使用 KNN 算法对模型进行预测，准确率达到 0.85。

在本次实验的过程中，我从对机器学习算法的理论有着初步认识，到自己编写代码实现模型的构建、训练和预测，收获颇丰。但实验中我也遇到了一些问题，

并在解决问题的过程中收获了经验和体会，记录如下：

首先，选择合适的机器学习算法是本次实验的最大难点。对于某一特定的数据集，选择最为合适的算法构建模型，可以起到事半功倍的效果。例如在本次实验中，我首先使用了 Logistic 回归模型进行分类，效果较差。后来我注意到实验的数据是根据聚类算法生成的，这意味着同类的数据很可能在某些维度上分布在邻近的位置，便选择了 KNN 算法进行分类。后来我尝试使用 sklearn 库中封装好的模型进行对比，选择了随机森林分类器进行训练，发现即便调整参数，预测率也很难达到 80%。因此，即便是集成学习这种基于复杂理论的强分类器，效果也不见得比理论简单但合适的算法好。

其次，选择合适的特征也是一个重要的议题。本实验中我主要是基于生活常识，选择了身高和体重作为训练的特征值，然后使用相关系数、双变量关系图等方式加以印证。但更多时候我们事先不知道某些特征对于其类别的“贡献”。在这种情况下我们可以像本实验中一样，先画出相关系数热力图，找到最相关的一系列特征。当然，我们也可以使用更为科学的方式，例如主成分分析等。

当然，对数据的处理也是细微但必不可少的过程。本实验中的原始数据相当一部分是字符串，难以直接进行训练，需要对其进行预处理。那么构建合适的映射就成为了难点。既要保证与原数据一一对应，数值也要体现原数据的程度变化，如本实验中肥胖程度的递增。

最后，感谢本次课程让我对算法和模型有了新的理解。课程开始前，我已经参加过两三次数学建模竞赛。但即便我们写出了完整的论文，甚至取得了一些奖项，我却一直对“‘模型’究竟是什么”这个问题完全不理解，不知道我究竟“建”了一个什么“模”。这次课程之后我对这个问题终于有了初步的认识，明白建立和训练模型像拟合函数一样，是一个对收集的数据进行描述并得到目标结果的过程。

机器学习的旅程就告一段落了，希望未来能够在数据与信息科学领域不断深入，学习更多知识，掌握更多技术，并应用到未来的科研和工作中。

6. 参考文献

[1] 周志华. 机器学习 [M]. 北京：清华大学出版社, 2016.

[2] 李航. 统计学习方法（第 2 版）[M]. 北京：清华大学出版社, 2019.

[3] F-score. https://zh.wikipedia.org/wiki/F-score