

《大数据挖掘》实验报告2

题 目 ：NBA当前赛季比赛结果胜负预测

专 业 数据科学与大数据技术

学 号

姓 名

日 期 2025年4月17日

**目 录**

[第1章 数据获取与特征工程 2](#_Toc195891744)

[1.1数据获取 2](#_Toc195891745)

[1.2特征工程 2](#_Toc195891746)

[1.2.1 schedule.csv时序划分 3](#_Toc195891747)

[1.2.2当前赛季各个球队的等级分elo 3](#_Toc195891748)

[1.2.3当前赛季的球员生物特征age、bmi 4](#_Toc195891749)

[1.2.4上个赛季的队伍相对排名rk 4](#_Toc195891750)

[1.2.5队伍连胜/连负特征streak 4](#_Toc195891751)

[1.2.6距离上次比赛的天数time 5](#_Toc195891752)

[第2章 模型训练与参数调优 6](#_Toc195891753)

[2.1模型训练 6](#_Toc195891754)

[2.1.1 PCA降维 6](#_Toc195891755)

[2.1.2逻辑回归 6](#_Toc195891756)

[2.2模型参数调优 6](#_Toc195891757)

[2.3 训练效果分析 6](#_Toc195891758)

[第3章 实战预测 7](#_Toc195891759)

# 第1章 数据获取与特征工程

## 1.1数据获取

本文使用3个数据集：

（1）上个赛季对各个赛队的整体统计信息Advanced Stats，包括赛队的排名、投篮命中率等。

（2）当前赛季每场比赛的具体信息Schedule and Results，每一行都是一场比赛的信息，包括比赛日期、主队名称和得分、客队名称和得分等。

（3）当前赛季各个球队球员的生物信息Players Bio，包括年龄、体重、所在球队等。

可以从官网上直接获得csv格式的Advanced Stats、Schedule and Results，将它们分别保存为total\_stats.csv和player\_bios.csv。对于Player Bios需要使用爬虫来获得：向NBA Stats网站发送HTTP GET请求，获取球员生物统计数据，然后将返回的JSON数据转换为一个Pandas DataFrame并保存为total\_stats.csv。

获取的数据集保存到对应赛季年份的raw目录下。本文需要对2024~2025赛季进行预测，因此将player\_bio.csv和schedule.csv保存在data/2025/raw目录下，total\_stats.csv保存在data/2024/raw目录下。

为了方便后续构造特征和模型训练，需要对raw目录下的文件进行数据预处理，包括删除无用属性，构造标签home\_win表示本场比赛主队是否胜利，将player\_bio.csv中队伍的缩写转换为全称。数据预处理后的结果保存在data/2024/clean和data/2025/clean目录下。

## 1.2特征工程

本文认为两个队伍的胜负情况主要取决于两个方面：队伍实力和场外因素（心理、近期比赛状态）

赛队实力可以由以下特征来表示。

（1）当前赛季各个球队的累计elo。

（2）当前赛季的球员生物特征。

（3）上个赛季的两个队伍相对排名。

场外因素可以由以下特征来表示。

（1）队伍连胜、连负情况。

（2）距离上次比赛的天数。

## 1.2.1 schedule.csv时序划分

本文基于data/2025/clean/schedule.csv构建特征，并采用时序划分策略完成训练、验证与预测任务。具体数据划分方法如下：

（1）**预测集定义**：设定预测起始日期predict\_date，在数据中定位该日期对应的行号predict\_index，将predict\_index至数据末尾的所有样本划为测试集。

（2）**训练集与验证集划分**：对predict\_index之前的样本，按8:2比例划分训练集与验证集。首先计算验证集分割点validate\_index = 0.8 × predict\_index，将0至validate\_index的样本作为训练集，validate\_index至predict\_index的样本作为验证集。

此方法严格遵循时序依赖性，避免未来信息泄露，同时通过动态调整predict\_date支持滚动预测场景。

### 1.2.2当前赛季各个球队的等级分elo

#### 1.2.2.1 Elo等级分系统

Elo等级分系统是指由Arpad Elo创建的一个衡量各类对弈活动水平的评价方法，被广泛应用于国际象棋、围棋、足球、篮球等运动。在NBA比赛中，每个队伍都有一个Elo分数，这个分数代表了它的“实力”，数值越高说明实力越强。比赛结果会更新Elo分数，如果一支队伍赢了，它的Elo分数就会上升，如果输了，它的分数就会下降。赢得强队比赢得弱队加的分更多，反之亦然。

相对于简单地计算胜率，Elo等级分能衡量队伍的真实实力，因为它更具体地考虑胜率中的两种情况：队伍战胜弱队和队伍战胜强队。Elo等级分系统中认为队伍战胜强队比队伍战胜弱队更能说明该队伍的实力强，因此在这两种情况下该队伍的等级分增加情况是不同的，而在计算胜率时，这两种情况被视为等价的，因此相对于计算胜率，Elo等级分系统能够更好地衡量队伍的实力。

假设棋手A和B当前等级分分别为和，则按Logistic distribution，A对B的胜率期望值为：

|  |  |
| --- | --- |
|  | （1） |

类似地，B对A的胜率为：

|  |  |
| --- | --- |
|  | （2） |

若棋手在比赛中的实际得分（胜=1分，和=0.5分，负=0分）与胜率期望值不同，则等级分调整公式为：

|  |  |
| --- | --- |
|  | （3） |

其中，和分别为调整前后的等级分，大师级比赛中K通常为16。

#### 1.2.2.2 添加各个队伍的等级分属性elo

为避免时序数据泄露，在构建队伍等级分属性（Elo）时需遵循以下核心原则。

（1）分阶段更新策略：队伍Elo分数仅在训练集阶段进行动态更新，验证集与预测集的特征构造必须直接沿用训练集最终生成的Elo分数结果，禁止使用验证集或预测集中的比赛结果信息更新分数，以确保模型评估与预测时不接触未来数据。

（2）特征构造与Elo分数更新的时序隔离：针对单场比赛的特征构造流程，需严格遵循“先提取历史分数，后更新当前结果”的操作顺序：在为主客队添加Elo属性时，必须基于两队在本场比赛前的历史等级分进行计算生成特征，待特征构造完成后，方可依据本场比赛结果更新两队的Elo分数，供后续比赛使用。这一流程避免了因误用当前比赛信息导致的数据泄露风险。

在具体实现中，可以使用字典team\_elo来记录每个队伍的累计等级分。每个队伍的初始等级分为1600，参数K=24用于控制分数波动幅度。

### 1.2.3当前赛季的球员生物特征age、bmi

从data/2025/clean/player\_bio.csv中获得各个球员的年龄age、身高height、体重weight、球员所在球队team。以team为分组字段做聚集操作，计算同一球队的所有球员的平均年龄和平均BMI，然后将这两个特征添加到数据集中，属性名为age和bmi。在具体实现时，需要将球员的身高和体重转换为厘米和千克，然后再计算BMI。

### 1.2.4上个赛季的队伍相对排名rk

从data/2024/clean/total\_stats.csv中获得各个队伍的排名，将每个队伍的名称和队伍的排名存储到字典team\_rank中，计算差值rk = 主队排名–客队排名来构造相对排名属性rk。

### 1.2.5队伍连胜/连负特征streak

本文通过动态追踪队伍连胜与连败状态，构建反映球队近期表现的趋势特征。具体流程如下。

（1）从比赛记录中提取主客队名称，查询两队当前的连胜或连败次数。

（2）根据预设规则计算综合趋势值：若主队连胜且客队连败，赋予streak值为2；若主队连败且客队连胜，赋予streak值为-2；单一队伍出现连胜或连败则分别赋予+1或-1；双方同趋势时归零。

构造连胜/连负特征时需要避免数据泄露：在训练集上，每场比赛计算特征后再更新连胜连败计数器，验证集和预测集则直接应用训练阶段积累的计数器状态，避免数据泄露。

### 1.2.6距离上次比赛的天数time

本文通过跟踪队伍比赛间隔时间构建时间差特征，用于量化主客队赛程密集度差异。针对每场比赛，记录主客队各自最近一次参赛日期，计算两者间隔天数作为特征值，然后同时更新两队最新参赛日期至当前比赛日。该特征可反映球队体能恢复周期差异。

# 第2章 模型训练与参数调优

## 2.1模型训练

### 2.1.1 PCA降维

由于上述构造的特征之间并不是完全独立的，为了获得能够影响队伍胜负的关键特征，本文在之前构造的特征基础上进行了PCA降维。经过调优分析，维度降低到5维是最优的。

### 2.1.2逻辑回归

本文使用逻辑回归模型作为分类器。由于数据是有时序关系的，因此需要按照上文计算好的划分点：validate\_idx、predict\_idx来划分训练集、验证集和预测集。在具体实现上，调用sklearn库的LogisticRegression.fit方法训练模型，再调用LogisticRegression.predict方法对验证集和预测集进行分类，将预测集的分类结果保存为新属性’home\_win\_predict’。

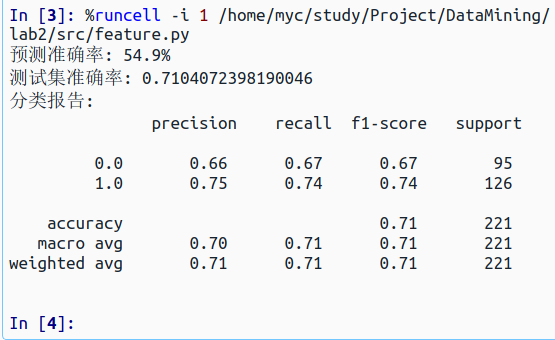
## 2.2模型参数调优

经过调优分析，本文确定的最优参数是：PCA降到为5维，逻辑回归使用L2正则项，正则化强度的倒数为1.0，最大迭代次数为100，使用lbfgs求解器。

## 2.3 训练效果分析

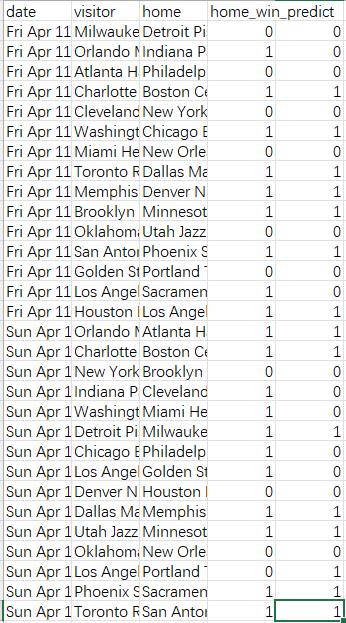
为了体现模型预测准确度的提升，本文先将所有预测标签都设置为1，即默认各场比赛的结果均为主队获胜，该基准准确率为54.9%。经验证，使用上述模型得到的准确率为71.04%，相较于基准准确率提高了17%。

从分类结果看，模型对类别1.0（胜场）的识别能力更强，其精确率、召回率和F1值均达到74%-75%，而对类别0.0（负场）的识别稍弱，三项指标约66%-67%。这可能是因为验证集中包含更多的主队获胜的样本（126个），较少的主队失败的样本（95个），导致模型更擅长捕捉主队胜的模式。



# 第3章 实战预测

本文对2025年4月11日和4月13日的比赛进行预测，预测结果如图所示。



4月11日比赛共有15场，正确预测13场。4月13日比赛共有15场，正确预测10场。准确率在76%左右，说明模型有较好的泛化能力。