《与星共舞》数据预处理全流程详解

一、问题本质：这不是普通数据集，而是“规则生成型数据”

核心洞察：

这个数据集是由**比赛规则**生成的，不是随机观察得到的。这意味着：

1. 缺失值有特定模式（规则性缺失）
2. 0 分有特殊含义（淘汰占位符）
3. 时间维度不对齐（不同赛季长度不同）

处理原则：

不能用传统的数据清洗方法（如均值填充、插值），而必须**重构数据以符合比赛逻辑**。

二、分步详解预处理过程

步骤 1：理解原始数据结构（宽表 → 矩阵）

原始数据形式：

```
选手 A | week1_judge1=7 | week1_judge2=6 | ... | week11_judge4=N/A 选手 B | week1_judge1=5 | week1_judge2=4 | ... | week11_judge4=0
```

问题：

- 这是一个 **44 列的评分矩阵**（11 周×4 法官）
- 但矩阵中有大量 `N/A` 和 `0`
- **关键：**需要区分"真正的 0 分"和"淘汰后占位 0"

Python 代码分析：

步骤 2：解析淘汰信息（核心逻辑）

任务：从 `results` 列提取淘汰周次

文本模式：

- `"Eliminated Week 3"` → 第 3 周淘汰

- `"1st Place"` → 未被淘汰（冠军）
- `"Withdrew"` → 中途退出（特殊处理）

数学表达：

设选手 i 在赛季 s 的淘汰周为：W_{\text{elim}}^{(i,s)} =

\begin{cases}

w & \text{如果淘汰周为 } w \\

\infty & \text{如果是冠军/决赛选手} \\

\text{特殊值} & \text{如果是中途退出}

\end{cases}**Python 实现：**

为什么重要：

> 没有准确的淘汰周信息，**无法确定选手在每周是否仍在比赛**，后续所有建模都会出错。

步骤 3：处理"淘汰后 0 分"（最关键步骤）

问题：

原始数据中，选手被淘汰后，后续周的评分都记为 0。例如：

- 选手第 3 周淘汰
- 第 4-11 周的所有评分都是 0

错误做法：

> 直接把这些 0 当作真实分数计算平均值

正确理解：

> 这些 0 是**占位符**，代表"该选手本周未表演"，应该视为缺失值

数学处理：

对于选手 i 在赛季 s：

- 如果 $w > W_{\text{elim}}^{(i,s)}$（即本周在淘汰之后）
- 则设置所有评分 $score_{i,w,j}$ = NaN

Python 实现：

数据验证：

步骤 4：宽表转长表（为时间序列分析准备）

为什么要转换：

- 宽表适合人类阅读，但不适合机器学习
- 长表格式：每行是一个"选手-周-法官"的观察值

转换示意图：

```
宽表（原始）：选手 | week1_judge1 | week1_judge2 | week2_judge1 | ...A
 | 7           | 6           | 8           | ...长表（处理后）：选手 | 周
 | 法官 | 分数 A    | 1  | 1    | 7A   | 1  | 2    | 6A   | 2  | 1
 | 8...
```

Python 实现：

数学意义：

设原始矩阵为 $M \in \mathbb{R}^{n\times m}$，其中 n 是选手数，m 是周评分列数。

转换后的矩阵为 $L \in \mathbb{R}^{k\times 4}$，其中 k 是有效评分数量，每行包含：选手赛季周法官分数(选手 ID,赛季,周,法官,分数)

步骤 5：创建周级数据集（每个选手每周一条记录）

目标：为每个选手在每周创建一个汇总记录

需要计算的指标：

1. **本周总分：**所有法官评分之和
2. **本周平均分：**总分除以有效评分法官数
3. **法官人数：**本周实际评分的法官数
4. **是否存活：**本周是否仍在比赛中
5. **本周是否被淘汰：**是否在本周被淘汰

Python 实现：

数学验证：

对于每个选手 i，周 w，定义存活函数：

$$\text{alive}(i,w) = \mathbb{I}(w \le W_{\text{elim}}^{(i)})$$

其中 $\mathbb{I}$ 是指示函数。

必须满足的一致性条件： $\text{alive}(i,w) = \boldsymbol{1} \Rightarrow \text{score}_{i,w} > \boldsymbol{0}\text{alive}(i,w) = \boldsymbol{0} \Rightarrow \text{score}_{i,w} = \text{NaN}$

步骤 6：特征工程（为建模创建有用特征）

6.1 人口统计特征

年龄分组：

是否美国出生：

6.2 时序特征

赛季进度（归一化的周数）：

$$\text{progress} = \frac{w}{W_{\max}^{(s)}}$$

其中 $W_{\max}^{(s)}$ 是赛季 s 的最大周数。

分数动量（与上周的变化）：

$$\text{momentum} = \text{score}_{w} - \text{score}_{w-1}$$

6.3 竞争特征

每周排名：

分数百分位：

排名

$$\text{percentile} = \frac{\text{排名} - 1}{N - 1}$$

其中 N 是本周参赛人数。

危险区标志（是否在淘汰边缘）：

步骤 7：创建赛季级数据集（用于跨赛季分析）

目标：每个选手在每个赛季有一条总结性记录

需要计算的指标：

1. **平均分数**：整个赛季的平均周分数
2. **分数稳定性**：周分数的标准差
3. **存活周数**：在比赛中持续了多少周
4. **最佳表现**：最高单周分数
5. **最差表现**：最低单周分数

Python 实现：

数学表达：

对于选手 i 在赛季 s：

$$\bar{x}^{(i,s)} = \frac{1}{W_{\text{alive}}^{(i,s)}} \sum_{w=1}^{W_{\text{alive}}^{(i,s)}} x_w^{(i,s)} \quad \sigma^{(i,s)} = \sqrt{\frac{1}{W_{\text{alive}}^{(i,s)} - 1} \sum_{w=1}^{W_{\text{alive}}^{(i,s)}} (x_w^{(i,s)} - \bar{x}^{(i,s)})^2}$$

其中 $W_{\text{alive}}^{(i,s)}$ 是选手 i 在赛季 s 中存活的周数。

步骤 8：验证预处理结果

8.1 逻辑一致性检查

检查 1：被淘汰选手后续周没有分数

检查 2：淘汰周与 results 列一致

8.2 数据完整性检查

检查 3：所有冠军都没有淘汰周

8.3 统计合理性检查

检查 4：分数分布合理

三、预处理后的数据结构（可直接用于建模）

1. 周级数据集 (`dwts_weekly_processed.csv`)

每行代表一个选手在某一周的表现：

列名	类型	描述
celebrity_name	字符串	选手姓名
season	整数	赛季编号
week	整数	周数
weekly_total	浮点数	本周总分
weekly_avg	浮点数	本周平均分
judges_count	整数	本周评分法官数
alive	布尔值	本周是否仍在比赛
eliminated_this_week	布尔值	本周是否被淘汰
weekly_rank	整数	本周排名
score_percentile	浮点数	本周分数百分位
age_group	类别	年龄分组
us_born	整数	是否美国出生
week_progress	浮点数	赛季进度

用途：Q1（粉丝投票估计）、Q2（投票规则比较）

2. 赛季级数据集 (dwts_seasonal_processed.csv)

每行代表一个选手在一个赛季的整体表现：

列名	类型	描述
celebrity_name	字符串	选手姓名
season	整数	赛季编号
avg_score	浮点数	赛季平均分
score_std	浮点数	分数标准差
weeks_survived	整数	存活周数
final_placement	整数	最终名次
top_3	布尔值	是否前三名
age	整数	年龄
industry	字符串	行业
us_born	整数	是否美国出生

用途：Q3（影响因素分析）

四、预处理对后续建模的影响

对 Q1（粉丝投票估计）的影响

如果没有预处理：

- 无法确定每周的实际参赛者集合
- 会把淘汰后 0 分当作真实分数
- 投票反演的约束条件会错误

正确预处理后：

- 可以准确构建每周的参赛者集合
- 可以正确设置投票反演的约束条件
- 可以计算合理的法官百分比和排名

对 Q2（投票规则比较）的影响

关键：需要在**相同的数据基础上**比较两种规则

- 需要知道每周哪些选手在比赛
- 需要正确的法官总分来计算百分比
- 需要正确的排名信息

对 Q3（影响因素分析）的影响

需要：

- 干净的特征变量（无错误编码）
- 合理的因变量（法官分数、粉丝投票估计值）
- 控制变量（周数、赛季等）

五、美赛论文中如何描述这一部分

建议结构：

1. **4.1 数据特征与挑战**
    - 描述原始数据结构

- 指出关键问题（结构性缺失、淘汰后 0 分）

2. **4.2 预处理方法**
   - 分步骤说明处理逻辑
   - 强调与比赛规则的一致性
3. **4.3 特征工程**
   - 说明创建的特征及其理论依据
   - 展示特征如何支持后续建模
4. **4.4 验证与质量控制**
   - 展示数据一致性的检查结果
   - 证明预处理的有效性

关键表述示例：

> "Unlike conventional datasets where missing values are random, the DWTS dataset exhibits **rule-generated structural missingness**. We therefore **reconstruct** the data according to competition rules rather than imputing missing values."
>
> "Post-elimination zeros are **placeholders**, not performance scores. We replace them with NaN to exclude them from analysis, ensuring that our models only consider actual performances."

六、完整代码执行流程

下面给你一条**“按步骤走、每一步解决什么问题、产出什么结果、用什么模型（基础+冲奖）”**的 O 奖路线图。你照着做，代码/论文都能自然长出来。

总览：一条主线贯穿 Q1→Q4

主线逻辑：

数据重构（季-周面板）→ 反演粉丝投票（隐变量）→ 赛制仿真对比（机制评估）→ 影响因素建模（解释）→ 新赛制设计（优化）→ 不确定性与敏感性（可信）

Step 1 数据预处理：把“宽表评分矩阵”变成“可建模的周级面板”

解决什么问题

- 原始数据是 weekX_judgeY 的宽表 + 结构性缺失 + 淘汰后 0 分占位
- 不转成周级面板，你后面所有“每周淘汰约束/仿真”都会错位

你要做的事（关键动作）

1. **宽转长**：(season, celebrity, week, judge) -> score
2. **识别存活集合**：构造 alive_{i,w}（淘汰后周次剔除，0 分不当真实成绩）
3. **周级汇总**：
    - judge_total_{i,w} 评委总分
    - judge_rank_{i,w} 评委排名
    - judge_share_{i,w} 评委占比（用于百分比制）

产出物（写论文&代码都要）

- 表 A：weekly_panel（一行=选手-赛季-周）
- 图 A：每季周数分布/评委人数变化
- 说明 A：N/A 与 0 的规则解释（证明你理解题意）

Step 2 赛制规则形式化：把“排名制/百分比制/评委救”写成统一函数

解决什么问题

- 不同赛季规则可能不同（题面说第 28 季左右改动）
- 你必须能在同一套框架里切换规则，才能做 Q2/Q4

你要做的事

定义一个**统一的“淘汰判定器”**：

输入：

- 当周存活选手集合 (A_w)

- 评委指标：`judge_rank` 或 `judge_share`
- 粉丝指标：`fan_rank` 或 `fan_share`（未知/估计）
- 规则类型：ranking / percentage / judge-save

输出：

- 当周淘汰者（或底二 + 被救者）

基础版规则（论文可直接写）

- **排名制**：(R_i = r^J_i + r^V_i)，最大差=最差，淘汰 (\arg\max R_i)
- **百分比制**：(S_i = p^J_i + p^V_i)，淘汰 (\arg\min S_i)
- **评委救**：先取底二，再由评委比较 (p^J) 或 (J) 决定淘汰

产出物

- 公式表 B：三种规则的数学表达
- 伪代码 B：`eliminate(rule, week_data)`

Step 3 Q1 核心：反演每周粉丝投票（隐变量）并复现淘汰

这是 C 题的“发动机”。反演做得好，后面一切都顺。

Step 3A（基础保分）约束反演：可行域 + 最小偏好原则

解决什么问题

- 粉丝票真实值未知，但必须能解释实际淘汰
- 解不唯一，需选“最合理”的一组

模型怎么写（清晰版）

对每一周 (w)，设粉丝投票份额向量 (v_w=(v_{1w},...,v_{nw}))：

- 约束：(v_{iw}\ge0, \sum_i v_{iw}=1)
- “淘汰一致”约束：
  - 用当前规则把 (v_w) 映射为综合得分
  - 强制实际淘汰者综合得分最低（或排名最差）

目标函数（选一个即可，论文好解释）：

- **平滑**：(\min \sum_i (v_{iw}-v_{i,w-1})^2)（人气不该剧烈跳）
- 或 **最大熵**：(\max -\sum_i v_{iw}\ln v_{iw})（最少假设）

输出物

- 表 C：每周每人 (\hat v_{iw})
- 指标 C：
  - 淘汰复现率（周级/季级）
  - 可行解区间（多解时给 min/max 或分位数）

Step 3B（冲奖）层次贝叶斯动态人气：规则似然 + 后验不确定性

解决什么问题

- 题目要求“投票总量是否随周/人变化”“确定性度量”
- 约束反演给点估计，贝叶斯直接给区间

结构（写成一句话就高级）

- 人气基底 (a_i)（受行业/年龄/舞伴影响）
- 周波动 (\delta_{iw})（随机游走/AR(1)）
- (v_{iw}=\text{softmax}(a_i+\delta_{iw}+\beta\cdot \text{judge_share}_{iw}))
- 用“淘汰发生”作为似然（被淘汰者更可能综合最低）

输出物

- 投票后验分布（均值+95%区间）
- 争议周解释：为何“高评委分仍淘汰”（人气后验低）

Step 4 Q2：赛制对比与争议分析（反事实仿真）

解决什么问题

- 排名制 vs 百分比制，哪种更偏向观众？
- 引入评委救会发生什么？

做法（按步骤）

1. 取 Step3 得到的投票（点估计或后验抽样）

2. 对每季每周分别套用：
    - 规则 1：ranking
    - 规则 2：percentage
    - 规则 3：judge-save（底二→评委选）
3. 得到三条“淘汰路径”和“最终名次”
4. 定义“偏向指标”（必须量化）

建议 3 个偏向指标（够用且好写）

- **观众主导度**：淘汰者是否为 fan_share 最低者的比例
- **技术保护度**：评委前半区选手被淘汰比例（争议率）
- **稳定性**：对投票 ± ε 扰动后淘汰改变比例

输出物

- 每季对比表：冠军/前 3 是否改变、淘汰序列差异
- 争议案例图：当周 judge_rank vs fan_rank 的冲突散点

Step 5 Q3：解释模型（舞伴/名人特征对评委分与粉丝票影响）

解决什么问题

- 哪些特征影响“技术分”？哪些影响“人气票”？是否相同？
- 舞伴效应是否真实，还是“强舞伴配强名人”的偏差？

基础保分：双回归/混合效应（可解释）

- 模型 J：`judge_total ~ 名人特征 + 周次 + (舞伴效应)`
- 模型 F：`logit(fan_share) ~ 名人特征 + 周次 + (舞伴效应)`
- 舞伴效应用固定效应/随机效应（混合模型）

冲奖：树模型 + SHAP（非线性解释）+ bootstrap 区间

- XGBoost/随机森林分别预测 `judge_total` 与 `fan_share`
- SHAP 输出“特征贡献排名”

- bootstrap 给置信区间（论文加分）

输出物

- 特征影响对比表：同一特征对评委 vs 观众影响是否反向
- 舞伴效应排名（技术提升 vs 人气提升）

Step 6 Q4：新赛制设计（机制设计/优化）

解决什么问题

- 提出“更公平/更好看”的系统，并说明为何应采纳

基础保分：加权百分比 + 技术阈值保护

- 综合分：(S=\lambda\cdot judge_share + (1-\lambda)\cdot fan_share)
- 保护规则：若 judge_share 前 k，则不可直接淘汰（进入底二由评委救）

冲奖：鲁棒多目标优化（权重随阶段自适应）

- 目标（多目标）：
  a. 公平（技术保护）
  b. 节目性（适度冷门）
  c. 稳健（对投票噪声不敏感）
- 决策变量：(\lambda_w)、k、是否救、救的触发阈值
- 方法：网格搜索/遗传算法 + 蒙特卡洛投票抽样（鲁棒）

输出物

- 新规则流程图（给制作方能看懂）
- 历史回测对比（指标雷达图/前沿图）

Step 7 不确定性与敏感性（O 奖常见分水岭）

解决什么问题

- 你的结论是否依赖某个假设？
- 投票估计多解时，制度结论是否稳健？

必做两类

1. **规则敏感性**：第 28 季起是否采用排名法/评委救？分别跑情景
2. **投票不确定性传播**：从投票后验/区间抽样 → 重跑 Q2/Q4 指标，给置信区间

Step 8 论文工程化呈现（按评委阅读路径）

1. Data & Preprocessing（强调结构性缺失与 0 分规则）
2. Vote Inference（点估计 + 不确定性）
3. System Simulation（统一仿真器）
4. Factor Analysis（评委 vs 观众双模型）
5. New System Design（可优化、可回测）
6. Validation & Sensitivity（稳健性）
7. Memo（1 页结论 + 2–3 条可执行建议）