数据预处理

5.1 数据重构：从“宽表”到“长表”

原始数据采用“宽格式”（Wide Format），即每个评委在每一周的打分都作为独立的列存在（如 `Week 1 Judge 1`, `Week 1 Judge 2`...）。这种格式不利于时序分析和数学建模。

- **操作思路**：利用 MATLAB 的 `stack` 函数或 Python 的 `melt` 方法，将数据降维。
- **目标格式**：建立以“（赛季 - 周次 - 选手）”为唯一主键的面板数据（Panel Data）。
- **新结构**：`[Season, Week, Celebrity_Name, Judge_Score_Raw, ...]`
- **目的**：统一处理不同周次评委人数不一致（3 人或 4 人）的问题，使模型具有通用性。

5.2 文本挖掘：生命周期提取

选手的生存状态隐含在非结构化的 `Results` 文本列中（例如 "Eliminated Week 5"）。我们需要将其转化为可计算的数值变量。

- **正则表达式提取 (Regex)**：
  - 针对淘汰者：匹配模式 $(Eliminated|Withdrew).*Week(\backslash d+)$，提取数字作为「淘汰周数 (Elimination Week)」。
  - 针对幸存者（Winner/Runner-up）：若匹配失败且非空，标记其「淘汰周数」为 99（或该赛季最大周数），代表其存活至最后。
- **退赛处理 (Withdrew)**：识别包含 "Withdrew" 的记录，建立布尔变量「是否退赛 (IsWithdrew)」。在后续模型中，这些样本将不参与“基于分数的淘汰预测”，因为其离开是非技术原因。

5.3 核心特征构造：归一化与排名

这是本题建模最关键的一步。为了消除不同周次评委总分差异（如满分 30 vs 40）的影响，必须从“绝对分数”转向“相对份额”。

- **变量 1：评委平均分 (Judge Average Score)**

计算公式：$JudgeAvg = \frac{\sum_{i=1}^{n} Score_i}{n}$（n 为当周有效打分评委数）

用途：仅用于 **数据可视化 (Visualization)**，直观展示选手水平波动。

- **变量 2：评委分占比 (Judge Score Share) —— 建模核心**

计算公式：$JudgeShare = \frac{\sum_{i=1}^{n} Score_i}{\sum_{all} \sum_{i=1}^{n} Score_i}$

其中$\sum_{i=1}^{n} Score_i$ 是选手 x 在第 w 周获得的评委总分；$\sum_{all}\sum_{i=1}^{n} Score_i$是当周所有选手的评委总分之和。

用途：**线性规划 (Linear Programming)** 的输入系数。在百分比制投票规则下，该变量直接决定了选手需要多少粉丝票才能存活。

- **变量 3：当周排名 (Weekly Rank)**

对当周所有参赛者的 JudgeShare 进行降序排列，计算「当周排名 (Rank)」。

用途：用于对比 “排名制” 与 “百分比制” 的公平性分析。

5.4 缺失值与样本筛选策略

针对不同的分析目的，采用分叉处理策略：

- **策略 A：用于可视化 (Dataset for Visualization)**

保留被淘汰后的行，并将得分填充为 `NaN`。

目的：在 MATLAB 绘图（`plot`）时，`NaN` 会自动导致线条断开，从而清晰地展示选手 “何时离场”，避免折线坠落至 0 的视觉误导。

- **策略 B：用于数学模型 (Dataset for Modeling)**

严格剔除所有 `NaN` 行、得分均值为 0 的行（未参赛）以及已淘汰选手的后续记录。

目的：保证线性规划求解器（`linprog`）和回归模型（`fitlm`）的输入矩阵是稠密且有效的。

5.5 辅助特征编码

为了完成 Task 4（人口统计学分析），需要将文本特征数值化。

- **行业分类 (Industry)**：将 `Celebrity Industry` 映射为大类（如：运动员 = 1, 演艺界 = 2, 真人秀 = 3, 其他 = 4）。
- **性别推断 (Gender)**：(如果数据未提供) 可选择忽略或利用 `Partner Name` 进行辅助推断（通常是异性搭档）。