

基于 Spotify 平台的音乐流行度预测及决策建议： 多模型比较、鲁棒性和可解释性研究

杨谊瑶 大数据 2301 班

1 引言

1.1 研究背景及意义

随着数字音乐平台的普及，Spotify、Apple Music 等平台已成为音乐传播的核心渠道，歌曲流行度直接关系到创作者收益、平台推荐策略优化及音乐产业的市场导向。音乐流行度预测的核心需求之一是让创作者、平台方理解“哪些因素能提升流行度”，从而更好地设置音乐推荐机制，而非仅追求预测精度。当前音乐流行度的影响机制仍缺乏系统性分析：是音乐本身的技术特性，如节奏、能量感等因素起着主导作用，还是创作者知名度、发行时代等外部因素影响更大？不同预测模型对流行度的拟合效果及可解释性如何？这些问题的解答对音乐产业具有重要实践价值。同时，数据驱动的预测模型易受数据投毒攻击，可能导致模型输出失真，进而影响产业决策。因此，在分析流行度影响因素的基础上，还要验证模型对数据投毒的鲁棒性。

1.2 研究问题及目标

本研究的核心目标是揭示音乐流行度的影响机制，为创作者优化作品、平台优化推荐策略提供可解释的参考；明确不同算法的适用场景并在数据投毒场景下为模型鲁棒性提升提供支撑。基于上述背景，本研究围绕以下核心问题展开：

1. 模型能否较为准确地预测一首歌的流行度？音乐流行度的关键影响因素是什么？不同因素对流行度的影响方向和强度如何？
2. 四类差异化机器学习算法在流行度预测任务中，从泛化能力、计算代价、鲁棒性、可解释性、隐私保护五个角度的表现差异如何？
3. 一首新歌最有可能因为什么而成功？基于模型的可解释性，在 Spotify 平台的内容创作和推荐机制方面，能提出什么决策建议？

1.3 数据来源及概述

本研究使用的数据集来自 Kaggle 平台公开的 Spotify 音乐轨迹数据集 [1]。该数据集原始样本总量 169,909 条，特征维度为 19 维。数据集由 Spotify 官方基于其平台海量音乐资源生成，音乐技术特征均通过 Spotify 标准化算法提取。同时覆盖 1920 年代至 2020 年代的跨世纪音乐样本，涵盖爵士、摇滚、电子、嘻哈等多种流派，具备较强的研究价值。

数据集的特征构成如下：

表 1: Spotify 音乐数据集特征说明

特征名称	数据类型	描述
id	字符串 (str)	歌曲的唯一标识符
name	字符串 (str)	歌曲名称
artists	字符串 (str)	歌曲的创作者
duration_ms	浮点型 (float)	歌曲时长 (单位: 毫秒)
release_date	日期型 (date)	歌曲的发行日期
year	整型 (int)	歌曲的发行年份
acousticness	浮点型 (float)	歌曲声学特性的量化指标
danceability	浮点型 (float)	歌曲舞蹈性的量化指标 (值越高越适合跳舞)
energy	浮点型 (float)	歌曲能量感的量化指标 (值越高节奏越强劲)
instrumentalness	浮点型 (float)	歌曲器乐元素占比的量化指标 (值越高人声占比越低)
liveness	浮点型 (float)	歌曲现场感的量化指标 (值越高越可能为现场录制)
loudness	浮点型 (float)	歌曲的响度
speechiness	浮点型 (float)	歌曲语音化程度的量化指标 (值越高说唱/独白元素越多)
tempo	浮点型 (float)	歌曲的节奏 (单位: 每分钟节拍数)
valence	浮点型 (float)	歌曲效价 (情绪积极性) 的量化指标 (值越高情绪越乐观)
mode	整型 (int)	歌曲的调式 (1 代表大调, 0 代表小调)
key	整型 (int)	歌曲的调号 (0-11 对应不同音高)
popularity	整型 (int)	歌曲的流行度得分
explicit	整型 (int)	歌曲是否包含露骨内容 (1 代表包含, 0 代表不包含)

2 数据清洗与特征工程

2.1 数据清洗

- 重复值和缺失值处理。数据集重复率为 0%，且所有特征的缺失值数量均为 0，无需额外去重和填充操作。
- 数据拆分与泄露防护。采用分层随机抽样策略，按 8:2 比例将清洗后的数据拆分为训练集（135,927 条）与测试集（33,982 条）。拆分后训练集与测试集的目标变量（popularity）分布一致，无显著偏差。
- 冗余字段剔除。剔除无分析意义的字段（id），初步精简特征结构。

2.2 特征工程

2.2.1 特征衍生

结合音乐数据的领域特性，构建具有分析价值和可解释性的特征。首先，在训练集上处理 artists 特征。原始数据的 artists 是以列表形式出现的，需要进行单独的统计和处理才具有可分析价值。统计每行歌曲的创作者人数，生成 singer_num 特征，表示每首歌曲参与创作的艺术家人数；并基于创作者在全训练集的出现频次，筛选出出现次数高于均值(7.23)的“核心艺术家”，形成核心艺术家列表，同时生成二分类特征 is_occur_mainSinger，表示歌曲创作者

中是否包含核心艺术家。只要该行的 artists 列表中有任意一位是核心艺术家, 该行数据即在 is_occur_mainSinger 列被标记为 1, 反之为 0。训练集中包含的艺术家共有 24,971 位, 其中核心艺术家共有 4,368 位。

接着处理 name 特征, 该特征是原始的歌曲名称, 不经过拆分和处理无法直接进行分析。根据 name 列的特点, 生成 3 个新特征: name_len (歌曲名称长度, 字符数)、is_featured (是否包含 “feat” 标识, 1 表示包含合作元素, 0 表示不包含)、is_remixed (是否包含 “remix” 标识, 1 表示混音版本, 0 表示原版)。其中 name_len 的均值为 23.6, 取值范围 1-255, 分布合理; is_featured 与 is_remixed 的占比分别为 2% 和 0.5%, 基本符合音乐产业中合作曲与混音曲的实际占比的大致情况。

然后处理 release_date 和 year 列。根据发行月份生成季节特征 season, 包含 winter、spring、summer、autumn、Unknown 五个类别; 同时将发行年份按音乐发展历程分箱, 生成时代特征 era, 包含 EarlyRecord、RockBud、RockGolden、ElectroHipHop、DigitalGlobal 五个类别, 分别对应 1921-1945、1946-1959、1960-1979、1980-1999、2000-2020 等五个关键阶段, 便于捕捉不同时代音乐的流行度规律。

2.2.2 对数转换

对某些偏态分布严重的特征进行对数转换, 使其向正态性方向适当改善。偏态严重的变量经过变换后的偏度和峰度对比如表2:

表 2: 数值变量偏度与峰度对数变换前后对比

变量名	偏度 (前)	峰度 (前)	偏度 (后)	峰度 (后)
duration_ms	6.8217	128.2249	-0.1073	3.8859
instrumentalness	1.6825	1.1221	-0.1636	-1.4117
speechiness	4.2370	19.3955	-3.2799	75.6624
singer_num	5.8103	62.8993	2.2524	5.0288
name_len	2.2209	7.3418	0.0205	0.0910

观察结果, 发现只有 speechiness 这一列在对数转换后情况变得比转换前更糟。通过其分布可以推测, 可能是由于 speechiness 原始分布有严重的近零堆积问题, 大量歌曲的 speechiness 非常接近 0, log 变换会把接近 0 的值拉到在负轴方向较远的地方, 导致反向恶化。

鉴于这一点, 在特征工程中放弃对 speechiness 的对数转换, 保留其原始分布; 仅对 duration_ms, singer_num, name_len, instrumentalness 这四个变量进行对数转换。

2.2.3 分类特征编码

针对多分类特征(season、era)与二分类特征(mode、explicit、is_occur_mainSinger、is_featured、is_remixed), 采用差异化编码策略:

对二分类特征, 其本身已为 0/1 编码格式, 直接保留原始取值, 无需额外处理;

对多分类特征进行独热编码, 为避免共线性问题, 分别以 season 中的 Unknown 类和 era 中的 EarlyRecord 类为基准组, 剔除对应编码列。编码后新增 8 个特征, 将分类语义转化为模型可识别的数值形式。

2.2.4 数值特征标准化

原始数值特征的量纲差异较大, 比如 duration_ms 单位为毫秒, danceability 单位为比例, 需通过标准化消除量纲影响。采用 Z-score 标准化方法, 在训练集中对 13 个数值特征进行处理, 使每个特征的均值为 0、标准差

为 1。测试集的标准化直接复用训练集的均值与标准差，以确保训练集与测试集的特征分布一致，避免数据泄露。

2.2.5 最终特征体系

经上述流程处理后，最终建模数据集的特征体系共包含 23 个特征。具体特征信息见表3。

表 3: 音乐数据建模特征体系总览

数值特征（13 个）	分类特征（13 个）	目标变量（1 个）
duration_ms (歌曲时长) acousticness (声音程度) danceability (舞蹈性) energy (能量感) instrumentalness (乐器性) liveness (现场感) loudness (响度) speechiness (语言性) tempo (速度) valence (积极度) key (音调) singer_num (歌手数量) name_len (歌曲名长度)	mode (调式: 1= 大调, 0= 小调) explicit (是否包含露骨内容) is_occur_mainSinger (是否出现主唱) is_featured (是否为合作歌曲) is_remixed (是否为混音版) season_autumn (秋季发布) season_spring (春季发布) season_summer (夏季发布) season_winter (冬季发布) era_RockBud (摇滚萌芽时代) era_RockGolden (摇滚黄金时代) era_ElectroHipHop (电子嘻哈时代) era_DigitalGlobal (数字全球化时代)	popularity (歌曲流行度)

3 数据初步探索与描述性统计

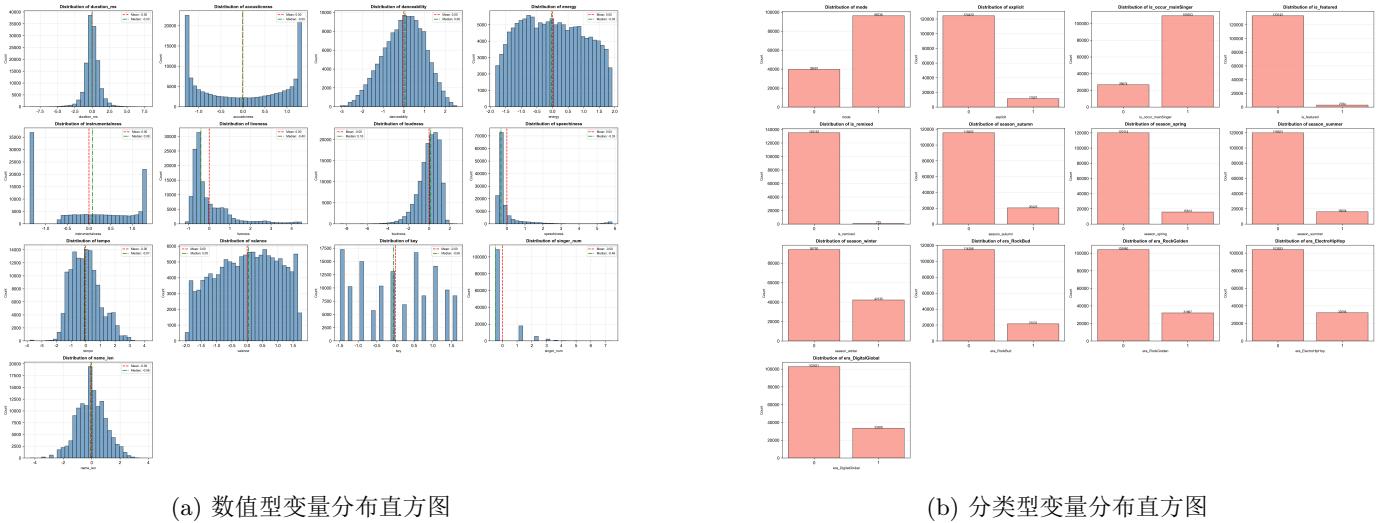


图 1: 特征变量分布情况

为深入理解 Spotify 音乐数据集的特征分布及特征与流行度 (popularity) 的潜在关联，本文基于预处理后的

数据集完成描述性统计分析，包括基础变量分布情况、相关性分析及广义可加模型（GAM）的两种核心方法，为后续建模提供先验支撑。数值型变量和分类型变量的分布见图1a、图1b。

数值变量的相关性关系见图2。从图中可以看出各个变量与流行度的直接关联特征。其中正相关特征有 danceability（舞蹈性，0.22）、energy（能量感，0.50）、loudness（响度，0.47），说明舞蹈性强、能量感高、响度大的歌曲可能更易流行；负相关特征有 acousticness（声学特性，-0.59）、instrumentalness（器乐元素占比，-0.30）、name_len（歌曲名称长度，-0.32），说明原声质感强、纯器乐占比高、名称长的歌曲可能会导致流行度更低。energy 与 loudness 的相关系数达 0.78，强正相关；其他特征间相关系数绝对值均 <0.6，无严重冗余。

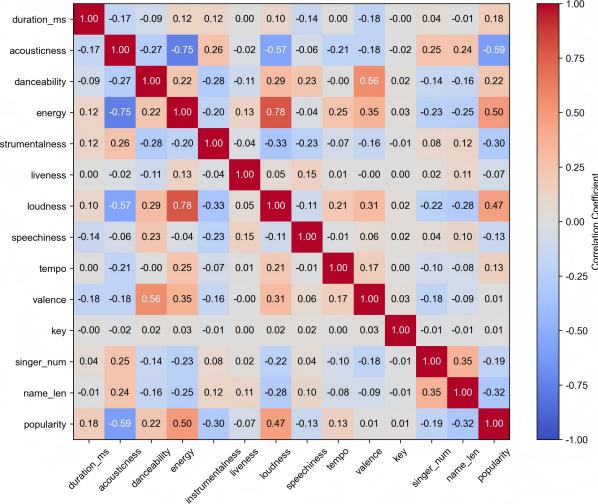


图 2: 数值型变量的皮尔森相关性热图

3.1 先验方法一：基于广义可加模型思想的非参数单特征描述统计

进一步挖掘特征与流行度的非线性边际关系，利用平滑基函数捕捉单特征的非线性趋势，以提升特征表达能力。训练集 $\bar{X} = \{(x_i, y_i)\}_{i=1}^m$, $x_i = (x_i^{(1)}, \dots, x_i^{(p)})^T$ 为 p 维特征向量, $y_i \in \mathbb{R}$ 为歌曲流行度，具体步骤如下：

- 1) 将原始数据集按特征维度拆分为 p 个单特征-目标变量数据集 $\bar{X}_k = \{(x_i^{(k)}, y_i)\}_{i=1}^m$ ($k = 1, 2, \dots, p$)，其中 $x_i^{(k)}$ 表示第 i 个样本的第 k 维特征。
- 2) 在每个数据集 \bar{X}_k 上构建 GAM 的单特征平滑函数 $g_k(\cdot)$ ，采用三次样条基实现非线性拟合：

$$g_k(t) = \sum_{j=1}^K \beta_{k,j} B_j(t) \quad (1)$$

其中 $B_j(t)$ 为三次样条基函数， K 为样条节点数 ($K = 10$)， $\beta_{k,j}$ 为样条系数，最小化损失函数求解：

$$\hat{\beta}_k = \arg \min_{\beta_k} \sum_{i=1}^m \left(y_i - g_k(x_i^{(k)}) \right)^2 + \lambda \int (g_k''(t))^2 dt \quad (2)$$

式中第二项为平滑惩罚项， $\lambda \geq 0$ 为惩罚系数，用于控制函数平滑度，避免过拟合。最终得到映射 $g_k : \mathbb{R} \mapsto \mathbb{R}$ ，即第 k 维特征的非线性变换函数。

- 3) 将原始特征 $x_i^{(k)}$ 替换为变换后的值 $z_i^{(k)} = g_k(x_i^{(k)})$ ，形成新的特征矩阵 $Z = (z_i^{(k)})_{m \times p}$ ，其中 $z_i = (z_i^{(1)}, \dots, z_i^{(p)})^T$ 。

3.2 先验方法二：强泛化模型驱动的单特征描述统计

由于单变量特征描述依赖于自变量无关性假设，本文还进行另一种特征描述方式，与方法一进行对比和补充描述。先通过强泛化能力的黑盒模型捕捉数据中的复杂模式，再通过边际效应分析提取单特征的变换函数。沿用数据集定义 $\bar{X} = \{(x_i, y_i)\}_{i=1}^m$ ，具体步骤如下：

- 1) 基于完整训练集 \bar{X} ，训练一个强泛化能力的非线性模型 $f : \mathbb{R}^p \mapsto \mathbb{R}$ （本文选用 XGBoost 梯度提升树），建模目标为最小化预测误差：

$$\hat{f} = \arg \min_f \sum_{i=1}^m \mathcal{L}(f(x_i), y_i) \quad (3)$$

其中 $\mathcal{L}(\cdot, \cdot)$ 为平方误差损失函数 $\mathcal{L}(a, b) = (a - b)^2$ 。

- 2) 定义第 k 维特征的边际效应函数 $g_k(\cdot)$ ，固定其他所有特征为其全局均值，仅让第 k 维特征自由变化，即：

$$g_k(t) = \hat{f}(\mu_1, \mu_2, \dots, \mu_{k-1}, t, \mu_{k+1}, \dots, \mu_p) \quad (4)$$

式中 $\mu_j = \frac{1}{m} \sum_{i=1}^m x_i^{(j)}$ 为第 j 维特征的全局均值， $t \in \mathbb{R}$ 为第 k 维特征的取值。该函数是强模型 \hat{f} 在第 k 维特征上的边际依赖曲线，反映了该特征对目标变量的净效应。

- 3) 对原始特征 $x_i^{(k)}$ 进行变换，得到 $z_i^{(k)} = g_k(x_i^{(k)})$ ，形成新特征矩阵 $Z = (z_i^{(k)})_{m \times p}$ 。

3.3 两种先验方法的边际依赖描述性统计结果

通过绘制两种先验方法的部分依赖图，可直观对比单特征与流行度的边际关系，结果如图3a（方法一）与图3b（方法二）所示。

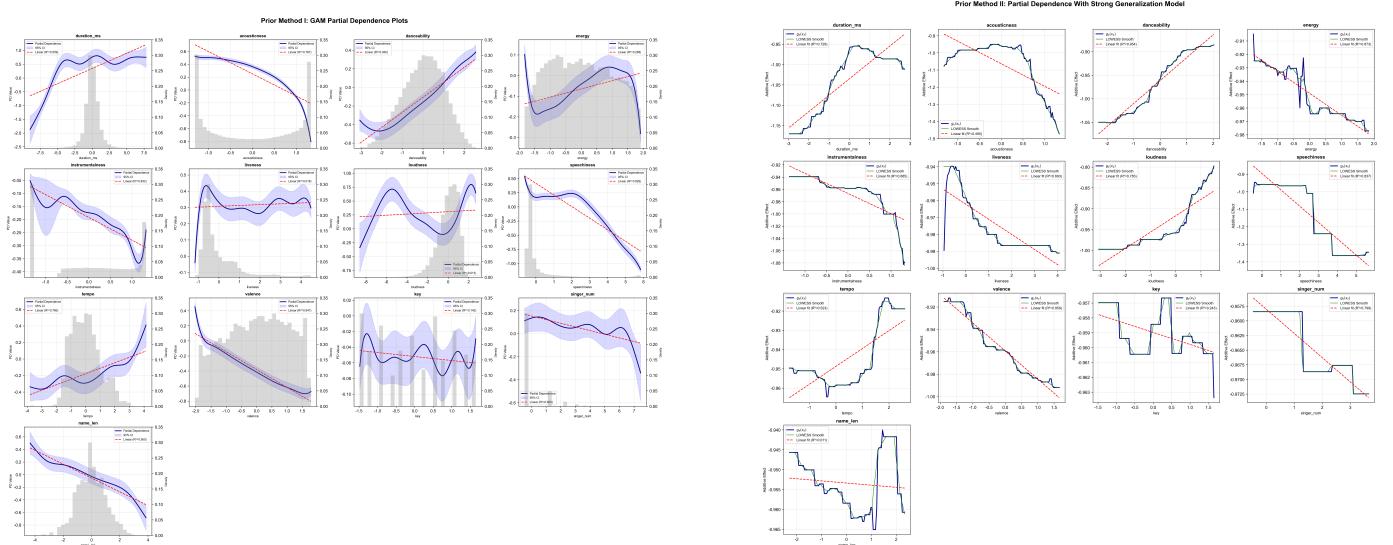


图 3: 单特征描述性统计

两种方法对大多数核心特征的边际依赖趋势呈现显著一致性，验证了这些特征主效应的稳定性。比如，两种方法均显示 `danceability`（舞蹈性）与流行度呈单调正相关，即舞蹈性越高，歌曲流行度的边际贡献越强，符合大众对高舞蹈性音乐的偏好；`speechiness`（独白程度）与流行度单调负相关，说明说唱、独白等语音元素占比越高，

歌曲流行度越低，符合主流音乐市场“人声旋律为主、语音元素为辅”的创作规律；name_len（歌曲名称长度）与流行度呈负相关，即名称越简短的歌曲更易获得高流行度，和“简洁标识更易传播”的传播学逻辑相符。

但也观察到少数变量在两种统计方法下呈现差异化趋势：energy（能量感）与 liveness（现场感）的边际依赖趋势有明显分歧。方法一中，能量感的边际依赖呈“单峰分布”，中等能量程度对应的流行度边际贡献最高，过高或过低的能量均会降低流行度；现场感边际依赖有明显波动，出现局部峰值。但方法二中，能量感的边际依赖呈单调下降趋势，能量感越高，流行度边际贡献越低，与方法一的趋势完全相反；现场感的边际依赖也持续下降，现场感越高，流行度边际贡献越低，与方法一的局部峰值趋势也是矛盾的。

上述趋势分歧的核心原因可能是两种方法对特征交互效应的处理差异。方法一的核心假设是“特征效应相互独立”，在建模时将数据集拆分为单特征子集、独立拟合平滑函数，完全忽略了特征间的交互效应。比如，通常情况下，高能量的歌曲往往伴随高积极度，能量感和积极度之间有 0.35 的正相关性关系；而积极度是流行度的负相关特征，也就是说更受欢迎的歌曲中往往掺杂了较多复杂、痛苦和纠结的情感，人们更倾向于喜欢表达负向情感的歌曲。方法一在对能量感这一变量独立建模时，无法剥离积极度的负向干扰，因此呈现和方法二相悖的结果。

方法二是全局拟合模型，其先通过梯度提升树捕捉所有特征的交互效应和各个特征之间的协同影响，再计算“固定其他特征为均值”的边际依赖。此时得到的是已排除其他特征的干扰的净效应。同理，现场感的趋势分歧也大概率源于交互效应：方法一忽略了“高现场感的现场版歌曲往往伴随高声音性”这一现象，更受欢迎的歌曲通常有更多的合成器和复杂的编曲，而不单单只有清唱人声，所以声音性和流行度呈负向关系；但方法二捕捉到了特征之间的交互趋势。

4 建模算法原理

4.1 局部化的广义线性可加模型（Localized GAM, L-GAM）

本文使用的一个算法是局部化方法和广义线性可加方法的结合：先用无监督方法做空间划分，再在每个区域内拟合一个 GAM。模型用 $\hat{y} = f(x)$ 来拟合目标变量 y ：

$$\hat{y} = f(x) = \sum_{c=1}^C \mathbf{1}\{x \in \mathcal{R}_c\} \left(\beta_0^{(c)} + \sum_{j=1}^p g_j^{(c)}(x_j) \right) \quad (5)$$

其中， \mathcal{R}_c 表示第 c 个局部区域，每个区域内是一个 GAM，区域之间参数不共享。算法实施分为两阶段：

1. KMeans 聚类，做空间划分。聚类依据为：

$$\min_{\{\mu_c\}_{c=1}^C} \sum_{i=1}^n \sum_{c=1}^C \mathbf{1}\{z_i = c\} \|x_i - \mu_c\|_2^2 \quad (6)$$

其中 μ_c 表示第 c 个聚类中心； $z_i = \arg \min_c \|x_i - \mu_c\|^2$ 。于是输入空间被划分成：

$$\mathbb{R}^p = \bigcup_{c=1}^C \mathcal{R}_c, \quad \mathcal{R}_c = \{x : z(x) = c\} \quad (7)$$

2. 在第 c 个区域，拟合独立的 GAM 模型。

- 假设空间。对于区域 R_c 内的数据，用 \hat{y} 拟合 y ，所有满足以下结构的函数 $f^{(c)}(x)$ 集合构成假设空间：

$$\hat{y} = f^{(c)}(x) = \beta_0^{(c)} + \sum_{j=1}^p g_j^{(c)}(x_j) \quad (8)$$

其中 $\beta_0^{(c)}$ 是该区域的截距， $g_j^{(c)}(x_j)$ 是第 j 个特征在该区域的平滑基底函数。

- 优化策略。采用带正则化的经验风险极小化，加入平滑惩罚项防止过拟合，得到优化问题：

$$\min_{\beta_0^{(c)}, g_1^{(c)}, \dots, g_p^{(c)}} \sum_{i \in R_c} \left(y_i - \beta_0^{(c)} - \sum_{j=1}^p g_j^{(c)}(x_{ij}) \right)^2 + \sum_{j=1}^p \lambda \int (g_j^{(c)''}(t))^2 dt \quad (9)$$

用三次样条基函数将每个 $g_j^{(c)}(x_j)$ 参数化展开：

$$g_j^{(c)}(x_j) = \sum_{k=1}^K \beta_{j,k}^{(c)} B_k(x_j) \quad (10)$$

其中 $B_k(x_j)$ 是三次样条基函数， $\beta_{j,k}^{(c)}$ 是需要拟合的系数，所有参数 $\beta_0^{(c)}$ 和 $\beta_{j,k}^{(c)}$ 由最小化目标函数得到。代入基函数展开式后，第 c 个区域的最终模型为：

$$\hat{y} = f^{(c)}(x) = \beta_0^{(c)} + \sum_{j=1}^p \sum_{k=1}^K \beta_{j,k}^{(c)} B_k(x_j) \quad (11)$$

- 学习算法。优化问题本质是最小二乘问题：

$$\min_{\beta_c} \|\mathbf{y}_c - \mathbf{X}_c \boldsymbol{\beta}_c\|_2^2 + \boldsymbol{\beta}_c^\top \boldsymbol{\Omega}_c \boldsymbol{\beta}_c \quad (12)$$

直接求 $\hat{\boldsymbol{\beta}}_c$ 的解析解为 $(\mathbf{X}_c^\top \mathbf{X}_c + \boldsymbol{\Omega}_c)^{-1} \mathbf{X}_c^\top \mathbf{y}_c$ ，其中 \mathbf{X}_c 为基函数矩阵， $\boldsymbol{\Omega}_c$ 为惩罚矩阵。

4.2 基于下采样方法的快速多项式核回归 (Fast Polynomial kernel Regression, FPR)

本文使用的第二个算法是下采样方法与核方法的结合。由于传统核方法在大规模数据下存储代价和计算代价都很大，因此基于 Kolmogorov 宽度理论，考虑运用下采样的策略，通过数据相关的特征映射逼近多项式核再生核希尔伯特空间，并结合合适的学习算法进行加速。

- 假设空间。对于输入空间 $\mathcal{X} \subset \mathbb{R}^d$ ，定义 s 次多项式核 $K_s(x, x') = (1 + x^\top x')^s$ 。该核对应的再生核希尔伯特空间 \mathcal{H}_s 是 d 变量、次数不超过 s 的多项式空间 P_s^d ，维数为 $\dim(\mathcal{H}_s) = \binom{s+d}{d}$ 。当 d 较大时，直接在高维空间中操作计算代价昂贵，因此算法采用随机锚点下采样近似。

从训练集中随机选取 m 个锚点 $\{c_j\}_{j=1}^m \subset X_{\text{train}}$ ，构造特征映射：

$$\phi(x) = [(1 + x^\top c_1)^s, (1 + x^\top c_2)^s, \dots, (1 + x^\top c_m)^s]^\top \in \mathbb{R}^m \quad (13)$$

其中 c_j 是从训练数据中随机选择的第 j 个锚点； s 是多项式次数； m 是锚点数量。学习模型为锚点特征的线性组合：

$$f(x) = \phi(x)^\top u = \sum_{j=1}^m u_j (1 + x^\top c_j)^s \quad (14)$$

其中 $u \in \mathbb{R}^m$ 为待学习的系数向量。当锚点随机均匀选择且 m 足够大时， $\text{span}\{\phi(\cdot)\}$ 以高概率逼近 P_s^d 。

- 优化策略。采用带正则化的经验风险极小化。给定训练样本 $\{(x_i, y_i)\}_{i=1}^n$ ，定义特征矩阵 $A \in \mathbb{R}^{n \times m}$ ，其中 $A_{ij} = (1 + x_i^\top c_j)^s$ 。本文采取 L1 正则化， $\lambda > 0$ 为正则化参数，则原始优化问题为：

$$\min_u \frac{1}{2} \|Au - y\|_2^2 + \lambda \|u\|_1 \quad (15)$$

- **学习算法。** 使用交替方向乘子法 (ADMM) 进行优化求解。为应用 ADMM，先引入辅助变量 v ，将问题重构为有约束优化问题：

$$\begin{aligned} \min_{u,v} \quad & \frac{1}{2} \|Au - y\|_2^2 + \lambda \|v\|_1 \\ \text{s.t.} \quad & u = v \end{aligned} \tag{16}$$

对应的增广拉格朗日函数为：

$$\mathcal{L}_\rho(u, v, w) = \frac{1}{2} \|Au - y\|_2^2 + \lambda \|v\|_1 + w^\top(u - v) + \frac{\rho}{2} \|u - v\|_2^2 \tag{17}$$

其中 $w \in \mathbb{R}^m$ 为拉格朗日乘子，即对偶变量； $\rho > 0$ 为惩罚参数。

ADMM 进行优化的算法流程如下：

1. 初始化 $u^{(0)} = v^{(0)} = w^{(0)} = 0 \in \mathbb{R}^m$ ，并预计计算 $A^\top A$ 和 $A^\top y$ 以减少迭代计算量。
2. u -子问题（最小二乘更新）：固定 $v^{(k)}$ 和 $w^{(k)}$ ，求解

$$u^{(k+1)} = \arg \min_u \left\{ \frac{1}{2} \|Au - y\|_2^2 + \frac{\rho}{2} \|u - (v^{(k)} - w^{(k)})\|_2^2 \right\} \tag{18}$$

解析解为线性方程组：

$$u^{(k+1)} = (A^\top A + \rho I)^{-1} [A^\top y + \rho(v^{(k)} - w^{(k)})] \tag{19}$$

3. v -子问题（proximal 算子更新）：固定 $u^{(k+1)}$ 和 $w^{(k)}$ ，求解

$$v^{(k+1)} = \arg \min_v \left\{ \lambda \|v\|_1 + \frac{\rho}{2} \|v - (u^{(k+1)} + w^{(k)})\|_2^2 \right\} \tag{20}$$

推导出闭式解为：

$$v^{(k+1)} = \mathcal{S}_{\lambda/\rho}(u^{(k+1)} + w^{(k)}) \tag{21}$$

其中软阈值算子 $\mathcal{S}_\kappa(z) = \text{sign}(z) \max(|z| - \kappa, 0)$ 。

4. 对偶变量更新：

$$w^{(k+1)} = w^{(k)} + (u^{(k+1)} - v^{(k+1)}) \tag{22}$$

5. 迭代直至满足 $\|u^{(k+1)} - u^{(k)}\|_2 < 10^{-4}$ 。

4.3 结合贪婪思想的递增式字典学习 (Incremental Dictionary Learning, IDL)

本文使用的第三个算法结合了递增式学习和字典学习进行前向特征选择：在递增式学习的框架下，尝试利用贪婪算法的思想，来逐步构建特征字典，而不是一次性选择固定字典。尽管最终模型仅依赖于一个特征子集，但该子集并非通过一次性特征筛选获得，而是由残差驱动的递增式学习过程逐步前向构建。

- **假设空间。** 对于输入空间 $\mathcal{X} \subset \mathbb{R}^d$ ，本算法通过递增式字典学习构建假设空间。字典 $D \in \mathbb{R}^{K \times d}$ 由 K 个基向量构成，每个基向量为单位基向量：

$$d_k = e_{j_k}, \quad k = 1, 2, \dots, K, \tag{23}$$

其中 $j_k \in \{1, 2, \dots, d\}$ 由贪婪策略从训练数据残差中选择。对于输入样本 $x \in \mathbb{R}^d$ ，定义特征映射：

$$\phi(x) = [\langle x, d_1 \rangle, \langle x, d_2 \rangle, \dots, \langle x, d_K \rangle]^\top \in \mathbb{R}^K. \tag{24}$$

由于 d_k 为单位基向量，有 $\langle x, d_k \rangle = x_{j_k}$ 。因此，假设空间是由这些原子张成的线性空间：

$$\mathcal{H} = \left\{ f(x) = \phi(x)^\top w = \sum_{k=1}^K w_k \langle x, d_k \rangle = \sum_{k=1}^K w_k x_{j_k} \mid w \in \mathbb{R}^K \right\}. \quad (25)$$

其中权重 w 由 Lasso 回归学习得到。这个假设空间依赖于选择的基向量集合 D ，而 D 又是通过训练数据以贪婪的方式学习得到的。通过逐步扩张假设空间并递增构造特征字典，属于递增式学习方法范畴。

- **优化策略。**目标仍然是最小化经验风险，但这里采用**两阶段策略**。在第一阶段，并不直接使用目标变量信息，而是通过重构输入数据来学习字典；第二阶段才使用标签进行监督学习：

1. **字典学习阶段（无监督）：**通过逐步贪婪选择原子，不使用目标变量信息，优化问题为：

$$\min_{d_k, z_k} \|R_{k-1} - z_k d_k^\top\|_F^2, \quad k = 1, \dots, K, \quad (26)$$

其中 $R_0 = X_{\text{train}}$ 为初始残差， R_k 为第 k 步更新后的残差。每步仅优化当前列 z_k 。

2. **Lasso 回归阶段（监督）：**在得到稀疏表示 $Z = [z_1, \dots, z_K]$ 后，建立 Lasso 形式的优化问题学习权重：

$$\hat{w} = \arg \min_w \frac{1}{2} \|Zw - y\|_2^2 + \lambda \|w\|_1. \quad (27)$$

- **学习算法。**无监督阶段利用贪婪思想求解，有监督阶段利用 Lasso 问题常用求解算法进行求解。给定训练数据 $X_{\text{train}} \in \mathbb{R}^{n \times d}$ 、标签 $y \in \mathbb{R}^n$ 、字典大小 K 、松弛参数 $\nu \in (0, 1]$ 、正则化参数 λ 。算法流程如下：

1. 初始化： $\mathcal{D} = \emptyset$, $Z = 0^{n \times K}$, $R_0 = X_{\text{train}}$.

2. 对 $k = 1, 2, \dots, K$:

- (a) 贪婪选择原子： $s_j = \sum_{i=1}^n R_{k-1,ij}^2$, $s_{\max} = \max_j s_j$. 构建候选集 \mathcal{C}_k ，并从中随机选择 j_k : $\mathcal{C}_k = \{j \mid s_j \geq \nu \cdot s_{\max}\}$ 。
- (b) 构造原子与计算系数： $d_k = e_{j_k}$, $z_k = R_{k-1} d_k$.
- (c) 更新残差： $R_k = R_{k-1} - z_k d_k^\top$.

3. 将字典整理为矩阵 $D = [d_1, \dots, d_K]^\top$ ，并采用经典方法求解权重 \hat{w} ，如梯度下降或 ADMM。

4.4 梯度提升树（XGBoost）

本文采用的最后一种算法是 XGBoost，这一经典的 Boosting 算法也属于**递增式学习**的范畴。但与上一种算法的区别在于，XGBoost 原子选择的是 **CART 树**，而非单位特征向量；同时 XGBoost 也并不是两阶段学习，而是残差直接驱动预测优化，但上一种算法残差仅仅驱动的是特征选择。

- **假设空间。**由多棵 CART 树函数的线性组合张成的空间：

$$\mathcal{H}_{\text{XGB}} = \left\{ F(x) = \sum_{m=1}^M f_m(x) \mid f_m \in \mathcal{F}_{\text{CART}} \right\} \quad (28)$$

其中 $\mathcal{F}_{\text{CART}}$ 是所有可能的回归树集合；每棵树 f_m 相当于原子，逐步扩张整个假设空间。

- **优化策略。**依旧采用带正则项的经验风险极小化，优化问题：

$$\min_F \sum_{i=1}^n L(y_i, F(x_i)) + \sum_{m=1}^M \Omega(f_m) \quad (29)$$

其中 $\Omega(f_m)$ 是树的正则化项，控制叶节点数、叶权重大小等。为了方便增量构建，利用一阶泰勒展开将损失函数在当前模型 F_{m-1} 附近线性近似，因此每轮迭代都可以通过拟合残差 $r_i^{(m)} = -g_i$ 来逐步减小训练误差。

- 学习算法。利用贪婪算法进行学习：

1. 初始化模型 $F_0(x) = 0$

2. 对每轮 $m = 1, \dots, M$ ：

- (a) 计算负梯度，表示当前模型在样本 i 上的残差或改进方向： $r_i^{(m)} = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \Big|_{F=F_{m-1}}$

- (b) 构建新 CART 树 f_m 拟合残差 $r_i^{(m)}$ 。每个节点选择增益最大的分裂点，是贪心策略下的局部最优。

- (c) 模型增量更新， η 是学习率： $F_m(x) = F_{m-1}(x) + \eta f_m(x)$

5 四种算法在数据集上的性能对比

为了评估本文提出的四种不同算法的预测性能，在同一数据集上对它们进行 5 折交叉验证及全量训练，并计算了训练、验证和测试的 RMSE 与 R^2 值，同时记录了训练耗时。本文从泛化能力、计算代价、鲁棒性、可解释性、隐私保护这五个方面分别进行系统分析。

5.1 泛化能力、计算代价

表 4: 各模型预测性能和时间代价综合对比

模型	训练集		5 折交叉验证		测试集		Time (s)
	RMSE	R^2	RMSE	R^2	RMSE	R^2	
L-GAM(10 簇)	0.4618	0.7868	0.4679	0.7810	0.4787	0.7718	249.0
FPR-ADMM	0.4634	0.7853	0.4672	0.7817	0.5296	0.7203	4.0
IDL-Greedy	0.4941	0.7558	0.4942	0.7557	0.5000	0.7508	5.3
XGBoost	0.4229	0.8211	0.4510	0.7966	0.4675	0.7821	12.0

首先是泛化能力。从表4中训练集和验证集结果来看，XGBoost 在 RMSE 和 R^2 指标上均显著优于其他模型，表明该模型具有最强的函数逼近能力。这一结果得益于 XGBoost 通过逐步构建回归树的方式，不断扩张假设空间，从而能够有效捕捉特征之间复杂的高阶非线性关系。

L-GAM (10 簇) 与 FPR-ADMM 在训练和验证阶段的预测性能较为接近，在当前数据集上具有相当的建模能力；但在测试集上，L-GAM 较 FPR-ADMM 体现出明显优势。L-GAM 通过聚类将输入空间划分为多个局部子空间，并在每个子空间内拟合独立的加性模型，因此面对测试集中分布略有变化的样本，仍能在相应局部区域内给出较为精确的预测。相比之下，FPR-ADMM 的泛化性能显著下降，甚至是四种算法中测试集表现最差的，可能由于其依赖下采样得到的有限多项式基函数来近似整体核空间。当下采样得到的基函数未能充分覆盖测试数据所对应的特征区域时，模型的表示能力会下降。

IDL-Greedy 的训练与验证误差相对较高，表明其在训练阶段的拟合能力弱于其他三种方法；但其测试性能与训练和验证阶段结果是四种算法中最为一致的。这一现象与其采用单位基向量构成字典、并通过贪婪策略逐步扩展模型复杂度的设计密切相关。由于字典原子的表达能力受到显式限制，IDL-Greedy 的假设空间规模相对受控，更偏向于低复杂度建模，从而在一定程度上牺牲了训练拟合能力，但换来了较为稳定的泛化表现。

在计算效率方面，各模型差异明显。IDL-Greedy 的训练时间明显快于其他方法，是所有模型中计算效率最高的。这一优势主要来源于其递增式字典构造与线性 Lasso 回归相结合的两阶段学习策略。字典原子通过贪婪方式逐步加入模型，每一步仅需计算当前残差与候选原子的相关性，适合大规模数据或快速原型建模场景。FPR-ADMM

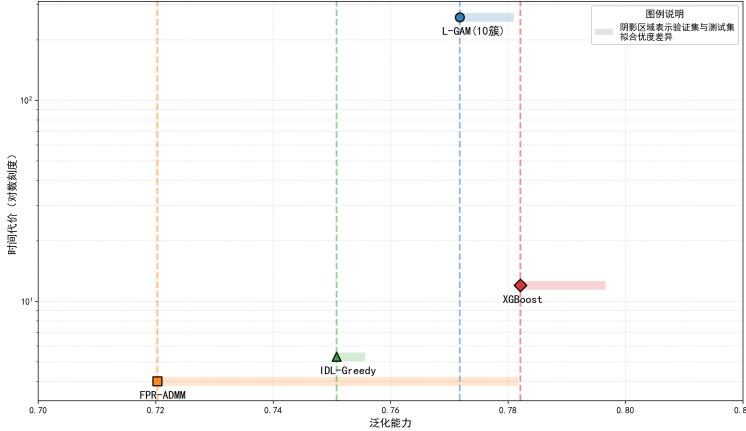


图 4: 四种算法时间代价与泛化能力综合比较图

的训练时间为 4.0 秒，显著低于 L-GAM，且是四种算法中效率最高的，表明基于下采样的多项式核近似有效降低了原始核方法的计算复杂度。同时，ADMM 将原问题分解为多个易于求解的子问题，也在一定程度上提升了优化效率，这也对应图中其时间代价处于较低水平的特点。相比之下，L-GAM 由于需要对多个局部簇分别拟合 GAM 模型，导致整体训练时间较长。XGBoost 在保持较高预测性能的同时，训练时间仅为 12.0 秒，在精度与效率之间取得了较好的平衡。

5.2 数据投毒场景下的鲁棒性

为系统评估不同模型在数据投毒场景下的鲁棒性，本文在训练集上人为引入不同比例的投毒样本，并固定测试集保持干净不变，以刻画模型在训练数据分布发生扰动时的泛化稳定性。在给定投毒比例 $\rho \in [0, 0.5]$ 的条件下，从训练集中随机选取 ρn 个样本作为投毒样本集合。

在投毒过程中，同时对输入特征与目标变量施加小幅扰动。对于目标变量，向选定样本的标签添加与其标准差成比例的高斯噪声；对于输入特征，则在随机选取的部分特征维度上叠加一定尺度的随机噪声。在该设置下，模型仅在被投毒的训练数据上进行学习，而验证与测试阶段均在未受污染的数据上完成，从而能够较为客观地反映模型对训练分布扰动的敏感程度及其鲁棒性表现。

实验结果见图5。IDL-Greedy、L-GAM 和 XGBoost 在投毒比例从 0 增加至 0.5 的过程中，测试集 RMSE 和 R^2 几乎保持不变，整体波动极小。这说明三种方法在模型较为稳定：IDL-Greedy 通过递增式特征选择与稀疏正则化限制模型自由度；L-GAM 通过局部聚类与加性结构，有效降低了异常样本对全局拟合的影响；XGBoost 基于多棵弱学习器的集成机制，有效降低了模型方差，从而缓解了部分异常样本带来的不利影响。因此，这几类方法对局部噪声和异常点具有较强的抑制能力，表现出良好的鲁棒性。

相比之下，FPR-ADMM 在投毒比例增加时表现出显著的不稳定性，其测试集 RMSE 与 R^2 出现明显波动，且在部分投毒比例下性能明显劣化。这一现象可能与其模型结构相关。FPR-ADMM 首先通过随机锚点构造多项式核映射，其特征映射形式为：

$$\phi(x) = (1 + x^\top c_j)^s, \quad (30)$$

该非线性映射在高阶情形下会对输入扰动产生放大效应。对于任意特征扰动 δx ，泰勒展开一阶近似为：

$$(1 + (x + \delta x)^\top c_j)^s - (1 + x^\top c_j)^s \approx s(1 + x^\top c_j)^{s-1} (c_j^\top \delta x) \quad (31)$$

当 $s > 1$ 时，扰动 δx 会被放大 $(1 + x^\top c_j)^{s-1}$ 倍。高阶多项式对极端值敏感，小噪声可能在映射后产生很大偏差，尤其当原始输入 x 或锚点 c_j 较大时。根据以上的泰勒展开可知，输入特征或训练样本中的微小扰动在映射后可

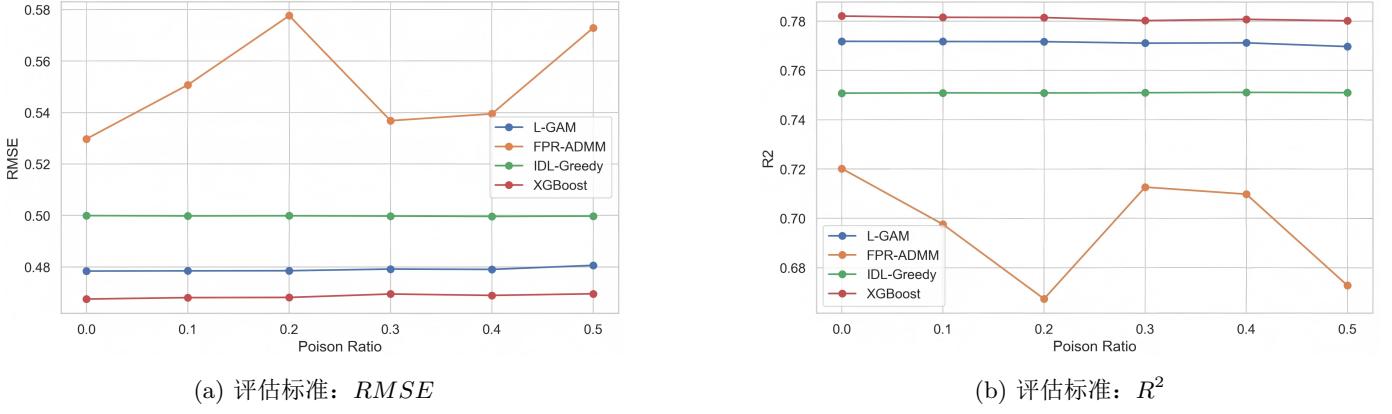


图 5: 数据投毒场景下四种算法的鲁棒性

能被成倍放大，从而改变映射空间中的几何结构。在数据投毒场景下，这种放大效应使得投毒样本对模型参数估计产生非线性累积影响，导致模型对训练数据分布变化高度敏感。另外，其学习算法 ADMM 对于初始化的敏感程度也可能影响到整个模型的稳定性。

综合来看，基于树的集成模型、递增式特征选择与加性分解模型在面对训练数据扰动时表现出更强的稳定性；而依赖高阶核映射的模型则更容易受到投毒样本的放大影响。

5.3 可解释性和隐私保护

在可解释性方面，各模型存在本质差异。L-GAM 由于其加性结构和光滑函数形式，能够直接刻画单个特征对预测结果的边际影响，是可解释性最强的模型。本文选择划分 10 簇，在每个簇内，计算每个特征对预测的影响强度。数值越大，这个特征在该簇的预测中作用越明显；反之说明该特征几乎没什么影响。表 5 展示了 L-GAM 算法 10 个簇中分别对预测最重要的特征。

表 5: L-GAM 特征解释表

簇	特征	影响强度	簇	特征	影响强度
0	era_DigitalGlobal	1.921215	5	era_DigitalGlobal	2.445344
1	duration_ms	2.224014	6	era_DigitalGlobal	2.232133
2	danceability	2.653478	7	liveness	2.777016
3	era_DigitalGlobal	2.145877	8	liveness	3.556252
4	liveness	2.713819	9	speechiness	7.416945

从表 5 中可以看出，era_DigitalGlobal（数字化全球时代）是出现频率最高的核心特征，在多个簇（簇 0、簇 3、簇 5、簇 6）中均具有较高的影响强度，说明该时代特征在多类样本簇的预测中均发挥关键作用。danceability 在簇 2 中影响最大，说明在该簇的歌曲中舞蹈性是预测流行度的重要因素。liveness 在簇 4、簇 7 和簇 8 的影响强度较高，提示现场感特征对这些簇的预测较为关键。而簇 9 中 speechiness 的影响强度显著高于其他特征，表明语音元素在这一类歌曲的预测中起到主导作用。

IDL-Greedy 通过递增式贪婪策略选择字典原子，其最终模型可以表示为少量原始特征或特征组合的线性加权形式，并由 Lasso 进一步施加稀疏约束，因此在高维场景下仍具有较好的结构可解释性。每一步选择一个最能解释残差的特征方向作为原子，所有选中的原子形成字典矩阵用来预测，表 6 展示了在 IDL-Greedy 生成的稀疏字

典中，哪些特征被选中，以及它们对目标变量的贡献强度和方向。

表 6: IDL_Greedy 特征解释表

特征	贡献	特征	贡献
era_DigitalGlobal	1.794103	instrumentalness	-0.058850
era_ElectroHipHop	1.178939	danceability	0.048977
era_RockGolden	0.821565	valence	-0.042071
acousticness	-0.126174	name_len	-0.034693
speechiness	-0.067027	season_spring	0.029431

分析表6中数据可知，特征对目标变量的贡献存在明显的强弱分化与正负差异：时代类特征占据主导正向贡献地位，其中 era_DigitalGlobal（数字化全球时代）是所有选中特征中贡献最大的，其次是 era_ElectroHipHop（电子嘻哈时代）和 era_RockGolden（摇滚黄金时代），这表明特定音乐时代背景对预测目标具有强烈的正向驱动作用。在负向贡献特征中，acousticness（声学特性）的负向贡献最强，其次是 speechiness（独白性）和 instrumentalness（器乐性）。其余特征如 danceability（舞蹈性）、valence（积极度）等的贡献绝对值相对较小，但仍优于其他没被选作字典的变量的贡献。IDL-Greedy 模型最终仅保留了少数核心特征及弱贡献特征，有效剔除了冗余信息，进一步印证了其在高维场景下的结构可解释性优势。

相比之下，FPR-ADMM 和 XGBoost 均依赖隐式特征映射或复杂的树集成结构，其预测结果难以直接归因于单个特征或简单函数形式，属于黑箱模型，解释性较弱。但 FPR-ADMM 使用的是多项式核，这一核函数的选择在一定程度上缓解了核方法在可解释性方面的弱势。

在隐私保护方面，不同模型对训练数据的依赖方式及其参数与原始样本之间的可逆性存在明显差异。IDL-Greedy 与 FPR-ADMM 在建模过程中并不直接学习样本的映射关系，而是通过逐步选择特征方向或固定形式的基函数来逼近目标函数，其最终模型参数主要反映的是全局或子空间层面的统计结构，而非单个样本的具体特征取值。这在一定程度上降低了模型参数被用于重构原始样本或识别特定样本的风险，从而表现出较为友好的隐私特性。

而 L-GAM 在局部簇内直接基于样本进行非参数函数拟合，其平滑函数参数在一定程度上刻画了局部数据分布特征。当簇内样本规模较小或分布较为集中时，模型参数更容易反映局部统计特性，因此其隐私保护能力有限。XGBoost 作为基于树的集成模型，需要多次对原始样本进行划分并生成复杂的决策路径，其分裂阈值和叶节点结构对训练样本较为敏感，因此在隐私保护方面也不具备天然优势。

6 基于模型可解释结果对 Spotify 提出的决策建议

6.1 哪些特征真正驱动流行度？

基于可解释性分析，本文有如下发现：

1. 时代/风格类特征（DigitalGlobal, ElectroHipHop, RockGolden 等）对流行度具有**最强的全局影响**；
2. 音频内容特征（speechiness, liveness, danceability, acousticness 等）影响也不可忽略，且有显著的分群差异；
3. 传统上被认为重要的情绪或技术性指标（valence、instrumentalness 等）对整体流行度有贡献，但贡献有限。

这说明音乐是否“流行”更多是内容定位与市场匹配问题，而非单纯的音频质量或制作技术问题。平台的内容策略也需要从“是否好听”转向“是否匹配当下主流文化结构”。

6.2 决策建议

6.2.1 优先切合当代主流语境

模型的特征选择解释了“一首新歌最可能因为什么而成功”这个问题。IDL-Greedy 中, era_DigitalGlobal 与 era_ElectroHipHop 具有最高的正向权重, 说明当前 Spotify 用户对具有数字化背景下诞生的音乐和嘻哈、电子融合风格的内容接受度更高, 而这正是当下流行音乐的主流范式。

根据这一点, 在 Spotify 的原创内容或重点合作企划中, 更应当优先扶持符合当代全球流行语境的曲风, 例如电子、嘻哈及其跨界融合形态; 在签约或推广新艺术家时, 需要将其音乐风格与当前主流时代标签进行匹配评估; 在推荐系统中, 应当显式强化对时代风格的建模, 将其作为影响流行度的重要因子, 而非仅作为附属标签。对于不处于主流时代语境的风格, 不应简单通过提高曝光强度来强推, 而应结合受众画像进行更精准的分发。

6.2.2 避免单一音频指标导向, 避免同质化

L-GAM 的分簇解释结果显示, speechiness、liveness、danceability 等音频特征在不同簇中的影响方向和强度存在显著差异: 某些特征在部分簇中是强正向驱动因素, 而在其他簇中影响极弱, 甚至方向相反。这说明不存在一个对所有歌曲都最优的音频特征配置。如果平台或创作者工具过度强调某些全局最优的音频指标, 比如统一追求更高的舞蹈性或更低的器乐性, 可能会削弱平台整体的内容多样性。

Spotify 应避免对创作者施加统一的音频指标优化目标, 而是要鼓励创作者在明确目标受众的前提下进行风格探索, 提供差异化的音频调整建议。在平台目前自带的创作者工具“Spotify for Artists”中, 可引入基于局部模型的分簇解释结果, 让创作者理解, 在他们所处的细分市场中, 哪些特征才是他们真正需要的。

6.2.3 利用局部解释进行差异化推荐

不同聚类中的用户对同一音频特征的反应差异显著, 这意味着推荐系统应当更加关注用户所处的内容空间和定位, 而非简单的全局打分。Spotify 可以在推荐系统中引入用户-内容的局部匹配机制, 对不同音乐子市场设置差异化的推荐逻辑。从技术角度看, 可以将 L-GAM 的局部解释结果嵌入到推荐策略中, 比如对于簇 7 和簇 8 的高 liveness 歌曲, 推荐系统应在现场音乐类歌单中提高曝光权重, 而非进入通用流行榜单, 从而提升用户体验的细腻度和长期满意度。

7 结论

尽管 XGBoost 在预测性能上略优, 但其黑箱特性难以直接支持内容策略制定。FPR-ADMM 在计算效率上较高, 但面对可能被投毒的数据, 表现并不稳定。相比之下, IDL-Greedy 与 L-GAM 在保持合理预测精度的同时, 提供了可操作的决策方案。对于 Spotify 平台而言, 真正有价值的并非“哪首歌会火”, 而是“为什么会火”。基于解释性强和更稳定的模型的分析, 能够帮助平台从内容生产、推荐机制与战略规划多个层面做出更加理性和可持续的决策。

参考文献

- [1] Alexander Kapturov. Spotify data from pyspark course. <https://www.kaggle.com/datasets/kapturovalexander/spotify-data-from-pyspark-course>, 2024. Accessed: 2024-01-20.