广东外语外贸大学信息科学与技术学院
《文本信息处理工程实践》课程大作业

大作业题目： Kaggle - TMDB 电影票房预测

组名： 我爱 NLP 小组

组员：
陈镇鸿 学号：20181002907
高子雄 学号：20181002989
游畅 学号：20181003005
管泽陶 学号：20181002911

指导老师： 李霞
作业提交时间： 2021.06.18

中文摘要

中国电影市场不断发展成熟，电影成为我们日常文化生活重要的一部分。电影业界通常以票房收入作为评价电影优秀与否的指标。票房预测像是一个精明的数学家，通过计算电影当中每一项因素对利润的贡献，运用工程技术客观地评估电影的质量，来预测一部电影的票房收入。票房预测重要性包括：

（1）在电影项目前期投资决策环节，投资者对项目投资有一个正确预期；

（2）在电影后期的营销和发行环节，发行方正确把握影片营销策略和发行定档。

因此，本文将通过机器学习方法，对电影数据进行挖掘，项目流程为：①探索性数据分析（EDA）与可视化；②数据清洗和预处理；③特征选择与特征构造；④LightGBM 和 XGBoost 建模；⑤微调参数；⑥目标预测结果。

同时，分析回归模型 LightGBM 和 XGBoost 的实验效果，力图获取最准确的预测值，为电影发行方和投资者挖掘出精确的票房收入预测，提供更优的选择方案。

本项目来自 Kaggle 平台上 Playground 赛题：TMDB（The Movie DataBase）票房预测。比赛提供的 TMDB 数据库，包含来自电影数据库的过去 7000 多部电影的原始数据，每条数据的属性列包括演员、工作人员、情节关键字、预算、海报、发行日期、语言、制作公司和国家等待。

Abstract

As the Chinese film market developing and maturing, films have become an important part of our dairy life.The film industry usually takes box office revenue as the index to evaluate whether the film is excellent or not. Box office prediction is lake a smart mathematician. By calculating the contribution of each factor to profit, and using engineering technology to objectively evaluate the quality of the film to predict the box office revenue of a film.

Box office forecast importance includes:

1. In the early stage of the film project investment decision-making, to make investors a correct expectation of the project investment;

2. In the marketing and distribution of the later stage of the film, the publisher can correctly grasp the marketing strategy and distribution file.

Therefore, this paper will use machine learning method to mine movie data. The project process is as follows: ① exploratory data analysis (EDA) and visualization; ② Data cleaning and preprocessing; ③ Feature selection and feature construction; ④ LightGBM and XGBoost modeling; ⑤ Fine tuning parameters; ⑥ Target prediction results.

Also, we analyzes the experimental effect of each models, trying to obtain the most accurate prediction value, and provide a better choice for the film distributors and investors to mine the accurate box office revenue forecast.

This project comes from the box office forecast of the playground competition topic: TMDB (The Movie DataBase) on the kaggle platform.

The TMDB database provided by the competition contains the original data of more than 7000 movies in the past from the movie database. The attribute columns of each data include actors, staff, plot keywords, budget, posters, release date, language, production company and country.

1 选题背景及意义

1.1 引言

（1）平台简介：

Kaggle 是一个大型、开放、在线的，举办机器学习与数据科学的相关竞赛的平台。该平台成立于 2010 年，至今已吸引了数以万计的研究者、开发人员、数据科学爱好者以及在校学生参加比赛。

尽管参赛门槛很低，然而 Kaggle 对学、工业界的推动是有目共睹的，这点由最优秀的队伍可得到公司高达 5000~10000 美刀的奖金便可见一斑。“任何公司和组织都可以受益于机器学习的进步。”此外，Kaggle 对个人也是极为有用的。抛开亮眼的简历不谈，不难想象在如此竞争激烈的环境中个人的能力将得到多大提升。

（2）选题背景：

随着中国电影市场发展成熟，电影成为我们日常文化生活重要的一部分，丰富了人们的精神生活。电影业界通常以票房收入作为评价电影优秀与否的指标。近年来，我国的电影《流浪地球》、《战狼系列》等在票房方面取得了空前绝后的巨大成功，人们在欣赏剧情之外，对票房收入等话题的讨论同样是津津乐道。

比起导演艺术单纯关注电影拍摄技巧，票房预测更像是一个精明的数学家。票房预测需要计算电影当中每一项因素对利润的贡献——例如一个得奖的出色编剧价值多少；是否该选择某位男演员以争取潜在的女性观众；这部剧本的续作是否值得翻拍……票房不仅仅为电影产业带来资金支持，也带来了理性的工程技术。值得一提的是，早在上世纪 80 年代，美国人巴里·利特曼就以特征构建为基础，融合影片的各种属性类型建立了票房分析模型。

1.3 研究意义

尽管目前国内的电影市场日渐火热，2010 年至今的电影票房呈指数型增长。同样地，历年观影人数以及电影银幕数量同样呈现着指数增长的趋势，然而在我国庞大的人口基数下，每 10 万人拥有的银幕数（2 块）远低于同期美国人均的近 15 块。

要想继续促进电影行业的健康发展，继续充实扩大人们的精神生活，就需要想办法提升电影的票房收入了。一部电影的票房收入，不仅仅是大家讨论的话题，同样也是电影投资方确保投资回报的保障。

在这样的大趋势下，对电影票房进行预测分析便显得尤为重要。票房预测分析有助于考察电影自身的特质对票房的主要影响因素，经过了这个步骤，投资方才能在拍摄初期做出合适的决策，资方和片方才能调整资源分配，更好地适应宣传需求，摄制出满足人们精神需求的好电影。这样，在电影项目前期投资决策环节，投资者对项目投资有一个正确预期；在电影后期的营销和发行环节，发行方正确把握影片营销策略和发行定档。

因此，电影票房预测对于电影行业有着重要意义。本次作业，我们将利用数据库建立模型，并利用竞赛平台给出的评分评判模型的可用性与准确性。

此外，值得一提的是，提供竞赛的平台 Kaggle 的开放性给了大家一个参与项目的机会，在这里可以接触真正的业界案例，收获实际的项目经验。面对这些前沿、真实的问题情景，解决过程能检验我们平日所学知识，提升解决实际案例的能力，也能极大锻炼我们的 coding 能力。除此之外，Kaggle 还是一个交流度高、讨论氛围良好的社区，不乏榜顶高手在赛后分享经验与解法，有助于扩宽我们的视野，更是一次不可多得的学习机会。

2 研究现状分析

在王伟[1]的文章中，探索了电影票房的预测研究的三个阶段：

第一阶段始于上世纪 40 年代，此时更多通过调研的方法集中面向观众反馈来预估票房，此

时对于后续的研究的贡献在于获取了不少影响票房的因素，如演员阵容，宣发手段，剧情内容，口碑等，这些因素不少都在后续的研究中被不断引用。

第二阶段则以预测模型为标志，巴瑞·李特曼首次以多个影响票房的因素作为自变量，票房（在当时应为电影租金收入）作为因变量，通过线性回归建立一个电影票房预测的模型，此后，学者们大体分为了两个方向：其一是以李特曼的思路为方向，保持多元数据，多元因子的线性回归思路，改进票房预测模型；其二则是逐一研究上述提到的影响票房的因子，其中，明星，影评，口碑等因子被重点研究。

第三阶段，学者们则更新了实验方法，通过网民在线生成的海量数据作为预测的主要数据来源，建立了更加高效，准确的票房预测模型。

另外，J　Li.[3] 等人在研究房子月租的预测任务中，使用到基于梯度提升的决策树模型 LightGBM 算法对任务进行深入探索，结果得到了很大的收获。同样的，对于 LightGBM 模型的研究中，Guolin Ke[4] 等人在实验中发现，该算法是一种高效的梯度提升框架，在合适的应用场景中，模型轻便、训练效率高。我们也在 Chen T[1] 研究中发现，另外一个模型算法 XGBoost 在回归任务中表现优异。基于此调研结果，以及对我们项目任务驱动方向，我们最后选择 LightGBM 和 XGBoost 作为我们实验项目的模型，开始对电影票房收入的任务深入研究，实验证明我们选择的算法框架是正确的。

3.本文算法

3.1 算法概述

3.1.1 XGBoost 概述

XGBoost 是一种集成学习算法，高效地实现了 GBDT（Gradient Boosting Decision Tree，梯度提升决策树）算法，并进行了算法和工程上的许多改进，被广泛应用在机器学习竞赛中并取得了不错的成绩。相比传统的 GBDT 在优化时只用到一阶导数信息，XGBoost 则对损失函数进行了二阶泰勒展开，同时用到了一阶和二阶导数。XGBoost 还增加了自动处理缺失值特征的策略。通过把带缺失值样本分别划分到左子树或者右子树，比较两种方案下目标函数的优劣，从而自动对有缺失值的样本进行划分，无需对缺失特征进行填充预处理。XGBoost 具有高效、灵活、轻便等优点。

3.1.2 LightGBM 概述

LightGBM 与 XGBoost 一样是对 GBDT 的高效实现，原理上它和 GBDT 及 XGBoost 类似，都采用损失函数的负梯度作为当前决策树的残差近似值，去拟合新的决策树。但 XGBoost 的计算量巨大，内存占用多，时间上也有较大开销。为此，lightGBM 在传统的 GBDT 算法上进行了一定优化，避免上述 XGBoost 的缺陷。

LightGBM 算法具有有以下特点：基于 Histogram 的决策树算法、带深度限制的 Leaf-wise 的叶子生长策略、直方图做差加速、直接支持类别特征(Categorical Feature)、Cache 命中率优化、基于直方图的稀疏特征优化、多线程优化。诸多改进，使得 LightGBM 训练速度获得提升，具有准确率更高、内存占用少、轻便等优点。

3.2 算法各模块流程图

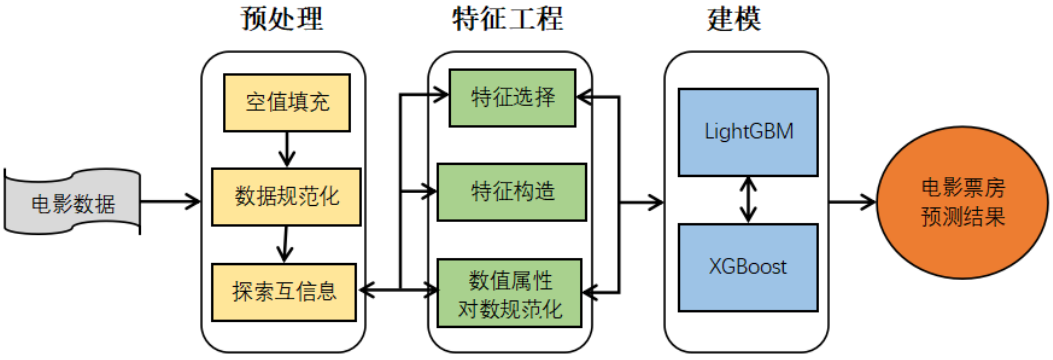

图 3.1 算法各模块流程图

我们项目的算法流程如上图所示，首先，我们对原数据集进行深入探索和分析，初步完成数据预处理的工作，包括对数据空值合理填充、数据规范化和每条数据的属性列的探索分析。接着，经过清洗和预处理后的数据，开始进入特征工程阶段，该阶段包含特征选择、特征构造和数值属性进行对数转化三个子任务。然后，使用 LightGBM 和 XGBoost 两种算法框架建立模型，使用十折交叉验证进行训练。最后，训练完成的模型可以对测试集中每一条电影数据的票房情况进行预测，得到最终结果。

3.3 算法细节

3.3.1 XGBoost 的研究

XGBoost 是一个开源的机器学习项目，高效地实现了 GBDT（Gradient Boosting Dicision Tree，梯度提升决策树）算法，并进行了算法和工程上的许多改进。

3.3.1.1 XGBoost 案例描述：

在原作者陈天奇的论文中，有这么一个例子。我们要预测一家人对电子游戏的喜好程度。由经验主义出发可以假设，年轻人和老年人相比会更喜好电子游戏；男性与女性相比会更喜好电子游戏，故我们先区分家人们的年龄大小，再区分家人们的性别，如图所示：

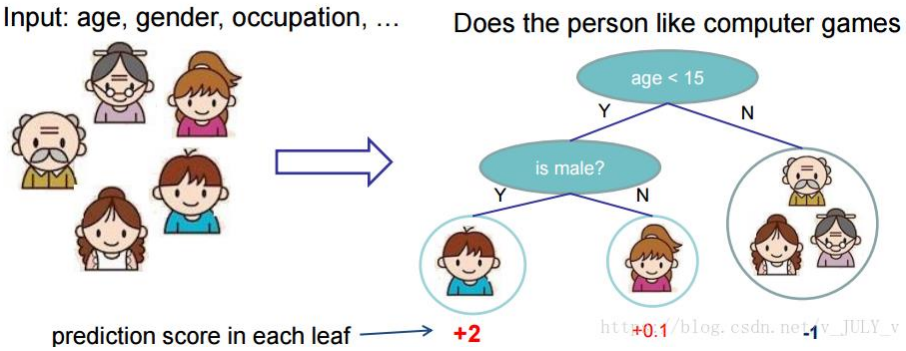

图 3.2 一家人的组成

如此我们便得到了第一棵树 t_1，即函数 f_1。接下来我们再假设：经常使用电脑的人相比少使用的人会更喜好电子游戏，然后重复上面的步骤，我们就可以得到第二棵树 t_2，即函数 f_2。两棵树的结论累加（函数输出值）起来便是最终的结论，所以小男孩的预测分数就是两棵树中小男孩所落到的结点的分数相加：2 + 0.9 = 2.9。爷爷的预测分数同理：-1 + （-0.9）= -1.9。具体如下图所示：

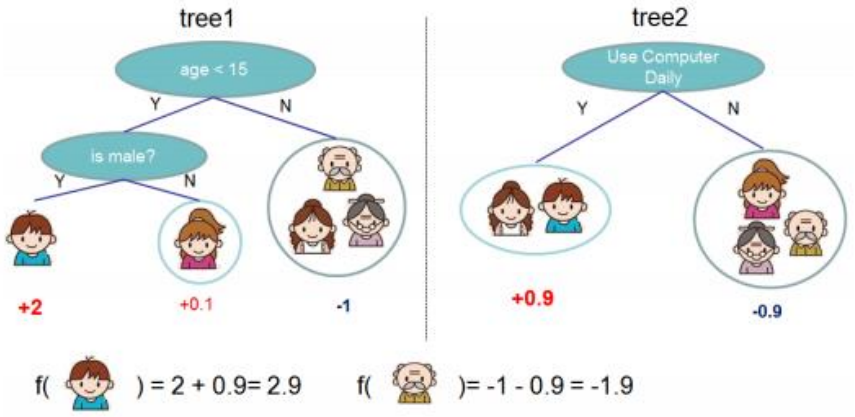

图 3.3 两棵树的得分

类似这样的 n 个弱分类器的集成，就是一颗 XGBoost 树。它的算法核心思想可以用下列的步骤来表示：

（1）不断地添加树（即利用特征分裂生长树）。每次添加树，就相当于学习新函数，用于拟合上一次预测的残差。

$$\hat{y} = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i)$$

（2）当我们训练完成得到 k 棵树，我们要预测一个样本的分数，其实就是根据这个样本的特征，在每棵树中会落到对应的一个叶子节点，每个叶子节点就对应一个分数。

（3）将对象在每棵树上对应的节点分数加起来即是该样本的预测值。

算法：XGBoost 模型算法实现流程

xgboost 的目标函数定义如下所示：

$$Obj(t) = \sum_{i=1}^{n} l\left(y_i, \hat{y}^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + C \quad \#(1)$$

然后使用二阶的泰勒展开近似原来的目标：

$$Obj(t) \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + C \quad \#(2)$$

其中，函数 l(x)即是损失函数，例如平方损失函数，或 logistic 损失函数； Ω(x)即是正则项，可以是 L1 范式或者 L2 范式；C 是常数项 Constant。

我们可以近似将 #(2)简化看做：

$$Obj(\theta) = L(\theta) + \Omega(\theta) \quad \#(3)$$

容易理解，一切目标函数追求的都是较小的损失函数（即尽可能地拟合训练数据）。因此，

当我们的目光回到 #(2)就很直接地得知，我们需要选取函数 f_t，以便让 $l(y_i, \hat{y}^{(t-1)})$这一项（即#(3)中的$L(\theta)$）变得最小（追求较小的损失函数）。有 #(2)可以知道，最终的目标函数只依赖于每个数据在误差函数上的一二阶导数，这便是 XGBoost 相较于 GBDT 的优化之处。

3.3.2 LightGBM 的研究

与 XGBoost 相同，LightGBM 也属于基于机器学习模型 GBDT 的改进版本。与 XGBoost 相比，LightGBM 在许多方面表现得更为出色，例如更快的速度，更高的准确率，以及更低的内存占用。

可以看出，LightGBM 比 XGBoost 快将近 10 倍，内存占用率大约为 XGBoost 的 1/6，并且准确率也有提升。

简单地讲，LightGBM 之所以能在上述方面得到如此大的优化，主要是 Histogram optimization（直方图算法）、GOSS（基于梯度单边采样）与 EFB（互斥特征捆绑）三种方法的特性所带来的。

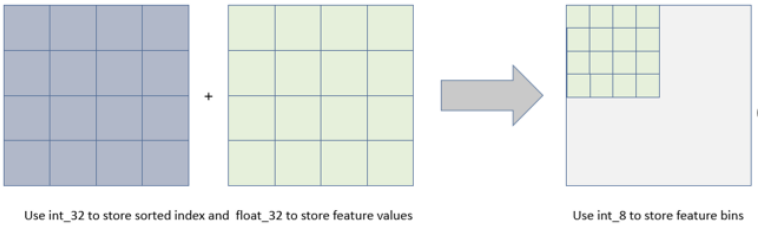

图 3.4 直方图算法减少内存消耗

算法：LightGBM 模型算法实现流程

直方图算法（HO）的思想类似于数据挖掘中的分箱法，简单却行之有效：首先将连续的浮点数据转换为 bin 数据，然后确定每个特征需要多少桶 bin，最后均分之，将属于某个桶的浮点数据更新为 bin 的值。尽管这样的做法相当于做了一次初始的正则化，并减少了大量的内存消耗，但同时也意味着放弃了一些数据的细节特征。因此，bin 是一个需要我们仔细考虑的超参数，它的数量越少，惩罚越严重，欠拟合风险也越高。

基于梯度的单边采样（GOSS）的中心思想是随机采样小梯度的样本（意味着样本间差异小）的同时，全部保留预先设定阈值或最高百分位间上的样本（意味着样本间差异大）。本质上是一种尝试在减少数据量与保证模型精度之间取得平衡的算法。

在 GBDT 的决策树中，确定分裂点的定义为：

$$V_{j|o}(d) = \frac{1}{n_0}\left(\left(\sum_D g_i\right)^2 / n^i_{j|o}(d) + \left(\sum_D g_i\right)^2 / n^j_{r|o}(d)\right)$$

其中 D:{$x_i \in o$, $x_{ij} \leq d$}，即数据集的全集；利用该公式计算得出的分裂点生成左右子叶。

在 GOSS 中，分裂点的定义修改为：

$$\tilde{V}_j(d)=\frac{1}{n}\left(\left(\sum_A g_i+\frac{1-a}{b}\sum_B g_i\right)^2/n^j l(d)+\left(\sum_A g_i+\frac{1-a}{b}\sum_B g_i\right)^2/n_r^j(d)\right)$$

其中集合 A 是 top a 个数据实例（样本差异大者），集合 B 是剩下数据中随机采样的子集（样本差异小者）。

此处分裂点的计算方式差异便是 LightGBM 运行速度得到较大提升的直接原因。

尽管上述两种做法都在一定程度上减少了精度。然而实际实验表明，LightGBM 的精度并不差，甚至相较于 XGBoost 好一些。这是因为选择树本身属于弱学习器，HO 算法起到了正则化的效果，避免了模型的过拟合。

互斥特征捆绑（EFB）也是一种降维方式，通过设立成为冲突比率的新指标衡量一对特征的不互斥程度，当这个值较小时，我们便可选择捆绑（合并）不完全互斥的两个特征，而不影响最后的精度。EFB 能够将高维稀疏特征转为低维稠密特征，将时间复杂度从 O(#data)降到 O(#non_zero_data)。

综上，结合了 GOSS 以及 EFB 的 GBDT 算法就是 LightGBM。

4. 实验结果或系统展示

4.1 实验数据探索与分析

本次任务要求利用平台给出的数据集进行预测，得出一份未给出票房的电影名单的票房预测值。不难判断，这种由一系列已知属性，预测某个具体数值的问题属于回归问题。

所给数据集的格式为 CSV 格式，其中训练集由 cast，post，budget 等 22 个加上票房属性 revenue 共 23 个属性列构成，共有 3000 条数据；验证集的属性值除 revenue 一列待预测置空外，其余属性列与训练集相同，共有 4398 条数据。

如此多的属性带来的模型开销是十分巨大的，与此同时也有些属性值从逻辑角度而言不会对 revenue 产生任何影响，如“imdb_id（在影片库中的 id）”，“poster_path（海报在网络上的链接）”等。因此，我们需要做的第一项工作便是 EDA（探索性数据分析），以期从中筛选出对预测票房结果影响最大的属性，并在预测效果与模型开销的代价之间找到可接受的平衡。

表 4.1：训练集的 23 个属性

编号	属性名	解释
1	id	电影的唯一 id
2	belongs_to_collection	这个电影系列的系列名称（不是系列电影就是空值）json 格式
3	budget	电影的预算，0 代表未知
4	genres	电影的类型以及类型对应的 TMDB 上的 ID，使用 json 格式封装信息
5	homepage	电影官方主页
6	imdb_id	电影在 TMDB 的 ID

7	original_language	电影的原始语言
8	original_title	电影的原始名称
9	overview	简短的描述
10	popularity	电影的流行程度，使用浮点数代表
11	poster_path	电影海报链接
12	production_companies	电影的出品公司，使用 json 格式
13	production_countries	电影出品公司所在国家，使用 json 格式
14	release_date	发行时间
15	runtime	电影时长
16	spoken_languages	电影语言，json 格式
17	status	电影的状态，是否已经发布
18	tagline	电影的宣传标语
19	title	电影英文名
20	keywords	电影的关键词以及相应关键词在 TMDB 上的 ID
21	cast	演员的姓名/id/性别，使用 json 格式
22	crew	职员（导演/编辑/摄影...）的姓名/id/性别，使用 json 格式
23	revenue	电影总收入，即票房，本次任务的预测值

4.2 数据探索性分析（EDA）结果

我们项目流程中很重要的一个工作就是数据探索。在数据预处理阶段：被选择的特征（属性列）中，我们发现其存在着①数值型数值缺失；②格式不兼容（如存在不同种的日期表示法）；③数据类型需要转化（如存在 JSON 形式的数据列）预处理。针对以上问题，处理方式如下：①使用众数填充；②提取年、月、日、季度与周几，新增时间属性列数据；③将 JSON 型数据转化为 dict，便于下一步的处理。

此外，还有一些地方值得我们的注意。例如 homepage 属性列中，显然 homepage 的“取值”为何对票房是不产生影响的，关键在于该部影片是否拥有 homepage。因为若一部影片拥有 homepage，则可以从侧面说明一些制作方的特征（例如经费是否充足，拍摄是否精心，宣传力度大小等），有助于预测影片票房。因此对于这种类型的数据我们可将其简化为“有”或“无”。同理，production_companies、production_countries 的处理方式也大同小异。

接下来就是探索性分析及可视化，我们发现许多隐藏的数据特点，对于后续特征工程阶段作了铺垫，更明确地选择特征信息，使得模型学习更加有效。

4.2.1 类别型属性

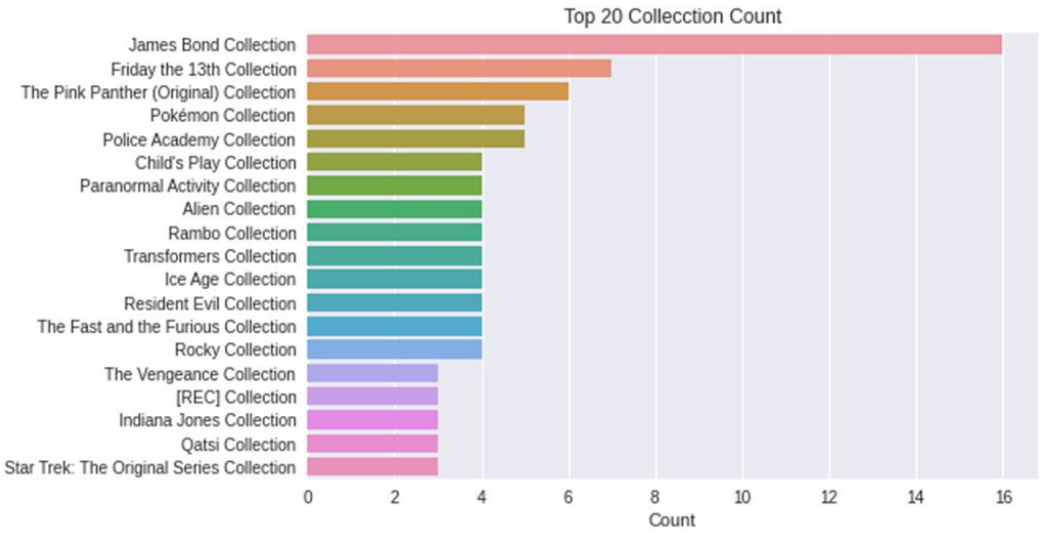

图 4.1 系列 Top20 频率条形图

不少电影都具有系列的规划，形成“IP”，对于一些票房可观的系列，其续作的票房可以以此为预测参考（如图中的第一“007”系列，票房独占鳌头，其续作票房可观的概率则较高。）

其中，只有约 20%的行有关于集合的信息，其余皆为空。其中，海报与背景都是图像信息，需要使用集合名称进行建模。

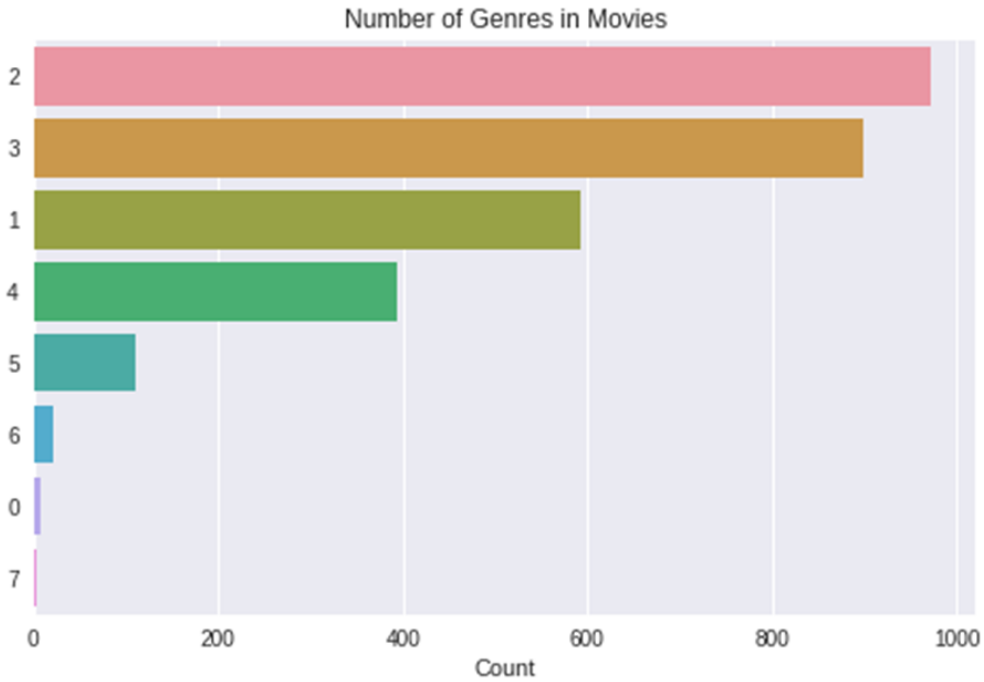

图 4.2 ‘genres’（各体裁）频率条形图

可以看出，大部分电影类型的主题选取都在 3 个以内，少数具有更多的主题定义或没有，可以判断电影的主题数选取在 1-3 的范围内对票房收入有一定的正面效益。

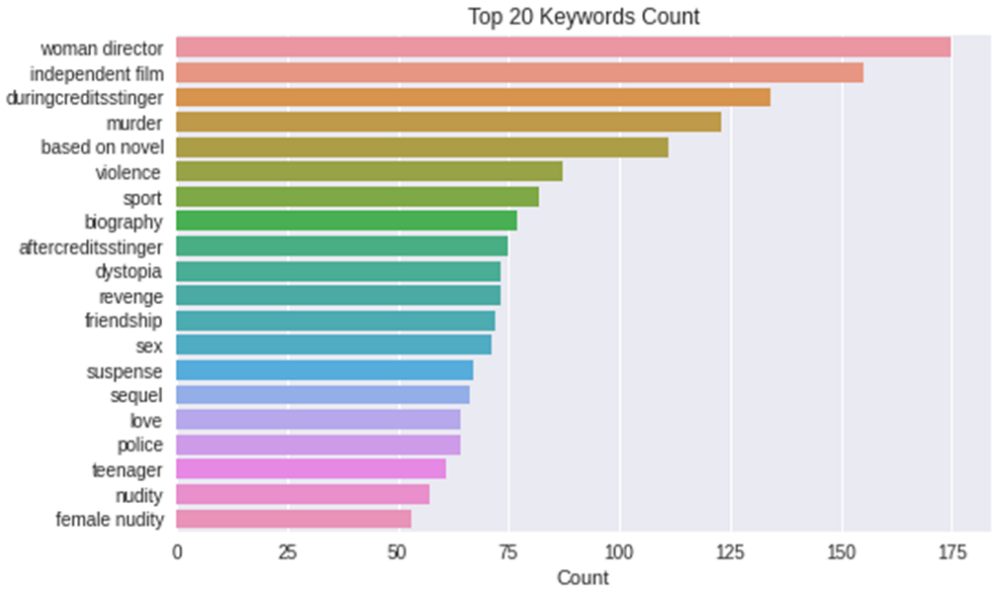

图 4.3 ‘keyword’（关键词）频率条形图

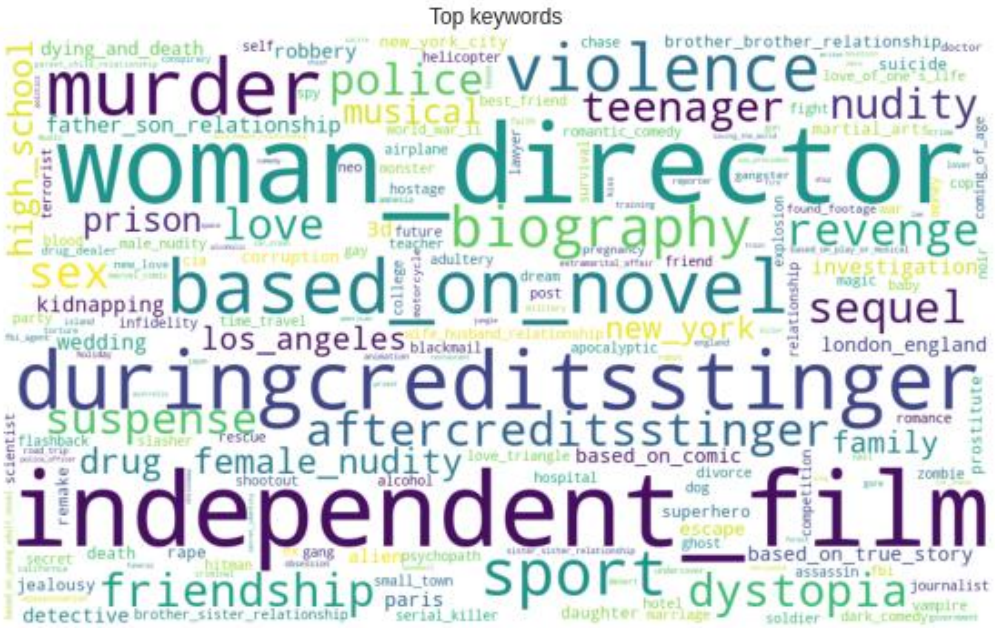

图 4.4 关键词的词云图

从电影的关键词的频率图和词云图，可以发现不少关键词具有相当的次数出现，推论出以动作与感情为导向的关键词占比较大，可作为目标预测的属性之一。

4.2.2 数值型属性

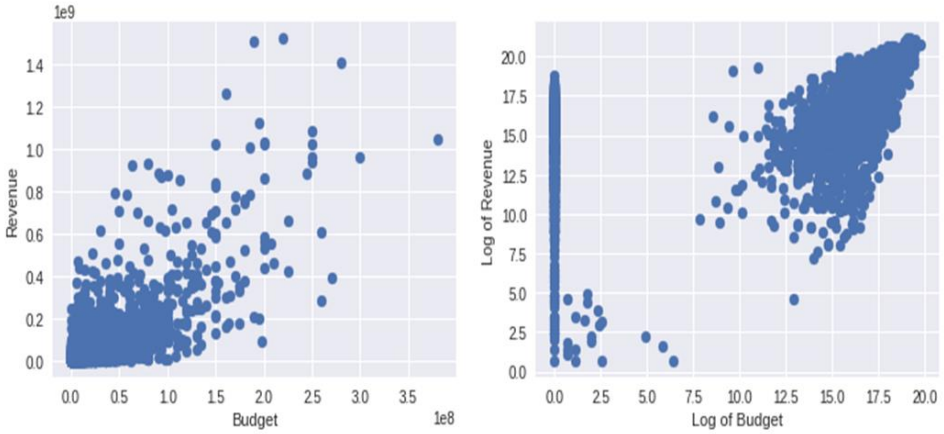

图 4.5 收入与预算的关系

在数值型属性的探索中，收入与预算直接的相关性显得尤为重要。从可视化途中可以看出两者具有明显的线性关系，少部分属于异常数据，包括低预算高收入，高预算低收入两种数据，即票房大赚/大卖。

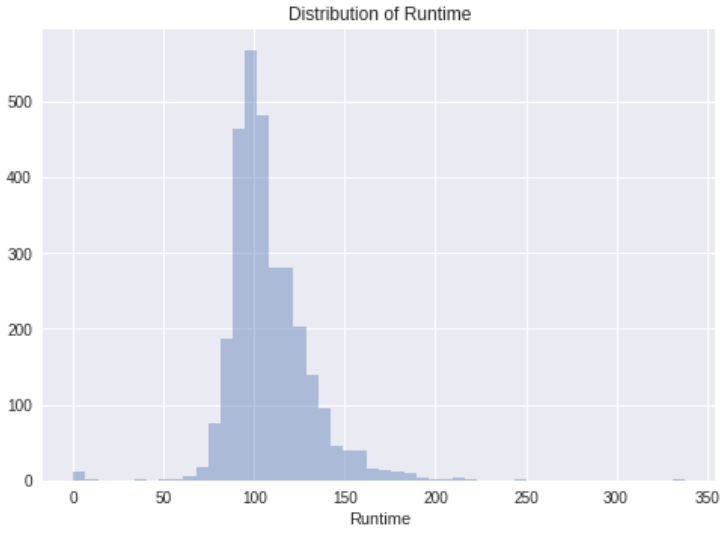

图 4.6 电影时长分布

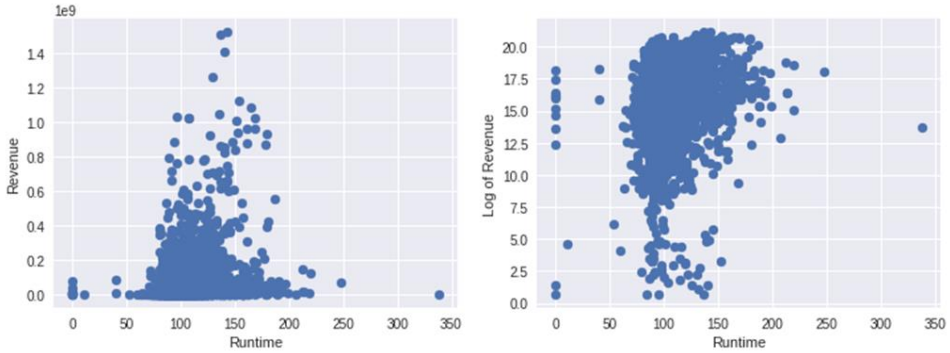

图 4.7　收入与时长的关系

图 4.6 和图 4.7 中，我们发现绝大部分电影的时长处于大约 80-140 分钟的范围内，小部分的电影时长小于 80/大于 140 分钟。时长在 80-140 分钟内的电影，收入与其并没有很明显的关系，而时长过短如低于 50 分钟，与过长如长于 200 分钟的电影，收入都处于较低的水平，在此推断应该是创作本意即为实验性质的原因。

下图是重要独立特征的依赖图：

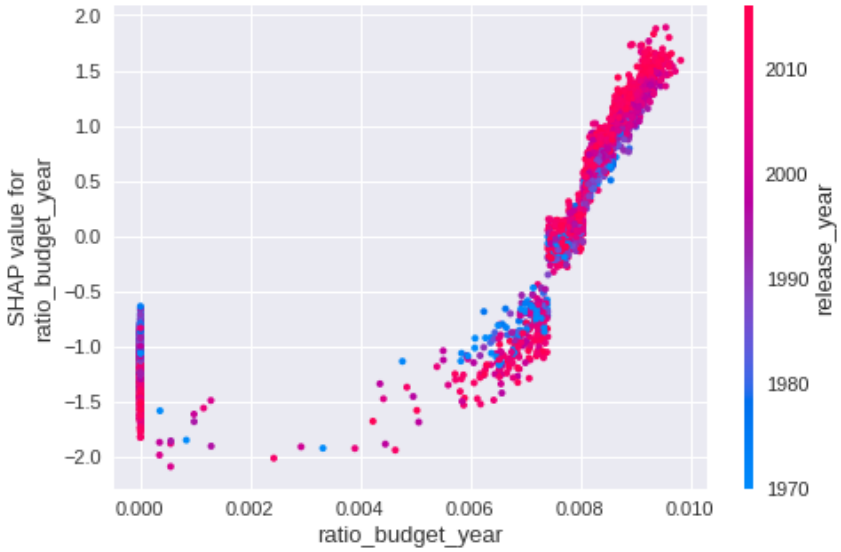

图 4.8　年份与票房比率

图 4.9 年份与流行度比率

另外一些数值型特征，我们探索了与目标票房收入的相关性系数，为特征构造提供理论支持和方向。探索的属性包括：
’ revenue’ ,’ budget’ ,’ popularity’ ,’runtime’ ,release_year’ ,’ release_month’ ,’ release_day’ ,’ release_dayofweek’

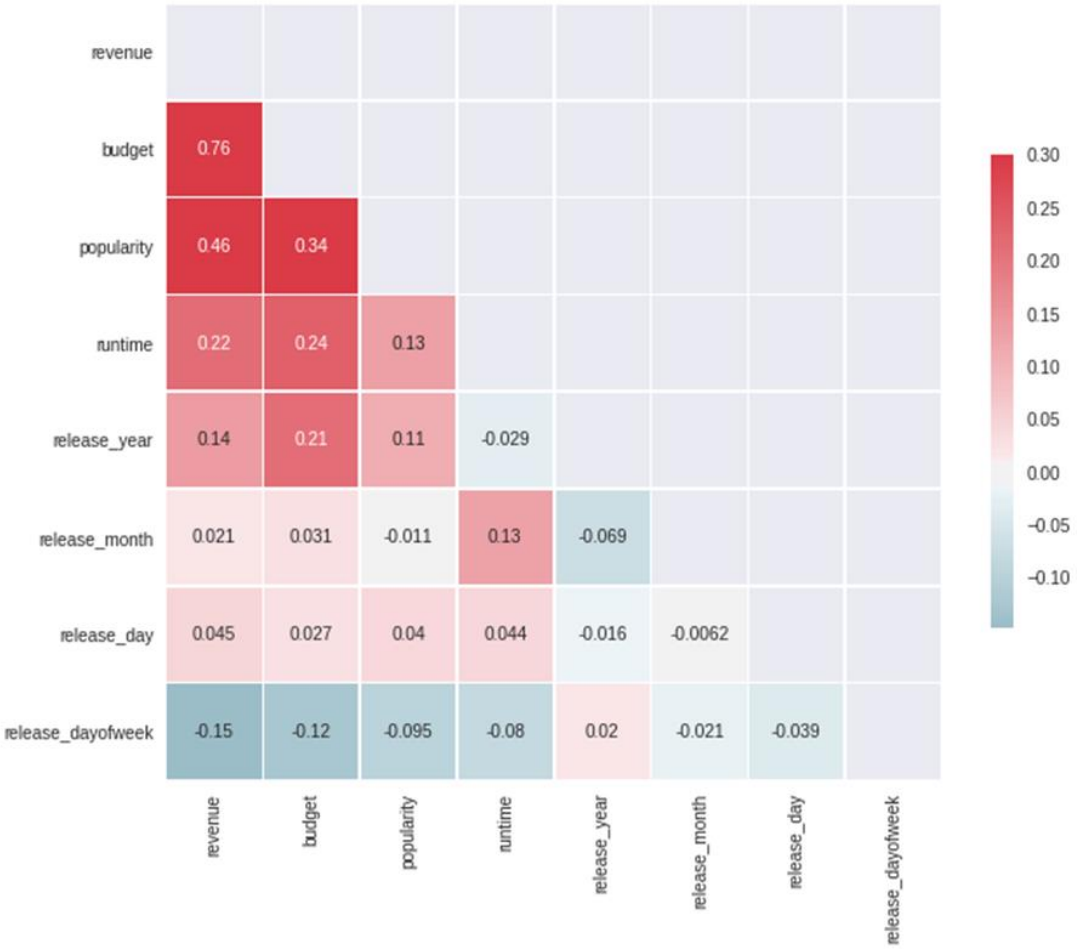

图 4.10 收数值变量之间的关系

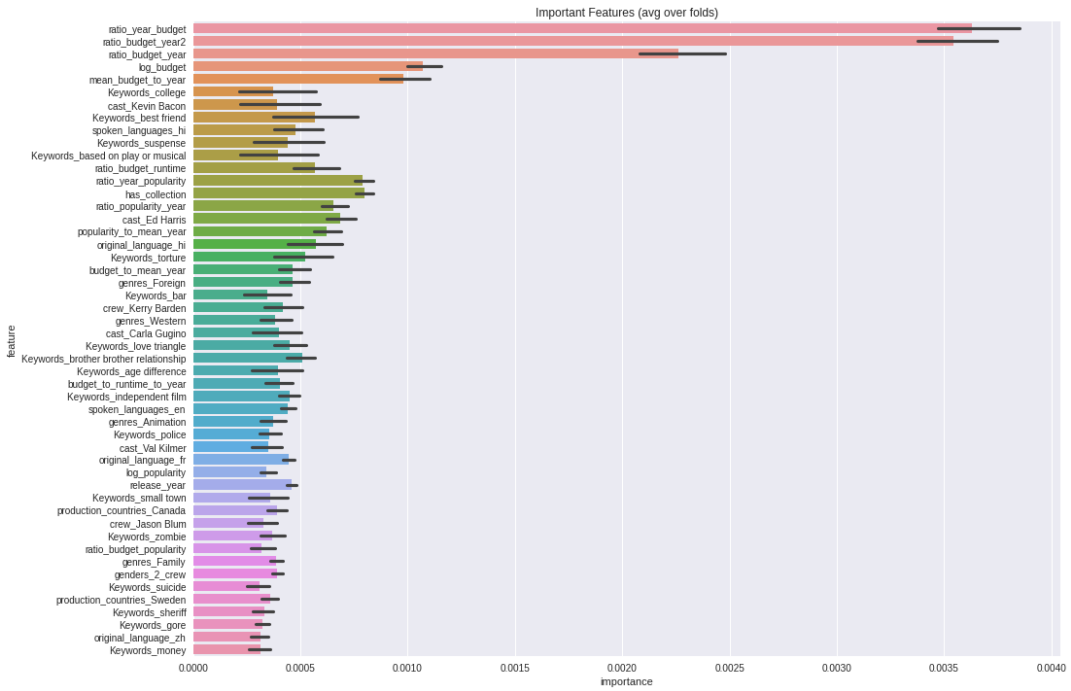

图 4.11 最终特征选择

4.3 评价指标

在本次任务中，我们选择的评价指标是 RMSLE（Root Mean Squared Logarithmic Error，均方根对数误差）。

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2}$$

其中，n 是数据集的总数，p_i 是预测值，a_i 是真实值。

RMSE 易被较大的离群值主导（对应于票房格外火爆的影片），一般应用与评估固定的平均分布的预测值。因此将预测值取对数之后再做 RMSE 能更好地衡量模型精度。

4.4 实验参数设置

我们对实验模型进行了参数调整，以便找到最优参数。表 3 是 LightGBM 模型调参过程展示，由表 3 可知，在 n_estimators 由 2000 提升到 10000 和 early_stopping_rounds 由 200 变为 500 的情况下，模型继续收敛，效果皆得到一定的提升。经过多次实验，特征选择比例 feature_fraction 为 0.7 时，实验效果最佳。值得注意的是，训练集和测试集中存在部分错误的数据记录，我们将其修正后，模型效果取得较大的提升，最佳实验结果为 1.89393，这表明错误数据记录对模型影响较大。我们的 LightGBM 模型的参数最终设置如表 3 所示。

表 4.1. LightGBM 模型调参

Rmse	n_estimators	early_stopping_rounds	feature_fraction	修正数据
2.05479	2000	500	0.9	flase
2.05422	100000	200	0.9	flase
2.05401	10000	500	0.9	flase
2.04603	10000	200	0.7	flase
2.04550	10000	500	0.7	flase
1.89393	10000	500	0.7	true

表 4.2. LightGBM 模型参数设置

参数	值
objective	regression
num_leaves	40
min_data_in_leaf	10
max_depth	6
learning_rate	0.01
boosting	gbdt
feature_fraction	0.7
bagging_freq	1

bagging_fraction	0.7
bagging_seed	2021
lambda_l1	0.2
n_estimators	10000
early_stopping_rounds	500

对于 XGB 模型，我们尝试对参数 n_estimators 和 early_stopping_rounds 进行调整,调试过程如表 4 所示。实验发现，在 n_estimators 为 10000 和 early_stopping_rounds 为 500 的情况下，模型效果较好。此后，继续提高两个参数，模型效果提升并不高。与 LGB 模型类似，数据集中部分错误的数据记录对模型影响较大，修正数据后，效果获得较大提升，最佳实验结果为 1.88941。我们的 XGBoost 模型的参数最终设置如表 5 所示。

表 4.3. XGBoost 模型调参

Rmse	n_estimators	early_stopping_rounds	是否修正数据
2.04956	10000	200	flase
2.04938	10000	500	flase
1.88941	10000	500	true

表 4.4. XGBoost 模型参数设置

参数	值
objective	reg:squarederror
eta	0.01
max_depth	6
min_child_weight	10
subsample	0.8
colsample_bytree	0.7
colsample_bylevel	0.5
seed	2021
n_estimators	10000
early_stopping_rounds	500

在模型融合部分，我们发现 n_estimators 为 10000 和 early_stopping_rounds 为 500 的时候，模型效果较好。我们对 lgb_w、xgb_w 和 cat_w 三个权重参数进行调整实验，结果如表 6 所示。实验结果表明，三个权重值比例约为 3：1：6 的时候，模型效果最佳。而相比单个模型，其效果获得了一定的提升，这表明了模型融合的有效性。在模型融合部分，我们的各个模型参数最终设置如表 7 所示

表 4.5. 模型融合调参

RMLSE	n_estimators	early_stopping_rounds	lgb_w	xgb_w	cat_w
1.78517	5000	200	0.33	0.33	0.33
1.78463	10000	200	0.33	0.33	0.33
1.78794	10000	500	0.4	0.4	0.2
1.78101	10000	500	0.33	0.33	0.33

1.77763	10000	500	0.2	0.4	0.4
1.77632	10000	500	0.3	0.2	0.5
1.77560	10000	500	0.3	0.1	0.6

表 4.6. 模型融合时各模型参数

lgb	xgb	cat
objective=regression	objective=reg=squarederror	bagging_temperature= 0.2
num_leaves= 60	max_depth= 5	l2_leaf_reg=1
min_data_in_leaf=10	min_child_weight=3	
max_depth=10		depth=10
learning_rate=0.005	eta=0.01	learning_rate=0.005
'boosting=gbdt	colsample_bylevel=0.5	colsample_bylevel= 0.7
feature_fraction=0.7	colsample_bytree=0.7	
bagging_freq= 1		
bagging_fraction=0.7	subsample=0.8	
bagging_seed=22	seed=22	random_seed= 22
'lambda_l1=0.2		
early_stopping_rounds=500	early_stopping_rounds=500	early_stopping_rounds=500
n_estimators=10000	n_estimators=10000	iterations=30000
lgb_w=0.3	xgb_w=0.1	cat_w=0.6

4.5 实验结果

表 4.7. 实验结果

模型	RMSE	RMSLE	排名
LightGBM	1.89393	1.79340	304
XGBoost	1.88941	1.79098	303
LightGBM+XGBoost	1.89149	1.79026	303
LGB+XGB+CAT	1.85265	1.77526	297

我们在 Kaggle 平台上获取的数据进行实验，结果如表 8 所示。可以看到，经过参数调整后，LightGBM 模型的 RMSE 为 1.89393，线上提交结果为 1.79340，排名达到 304。而 XGBoost 模型效果与 LightGBM 模型相差不大，RMSE 为 1.88941，RMSLE 为 1.79098。相比 LightGBM 模型,排名上升了 1 位。

然后我们进行了模型融合实验，对 LightGBM 与 XGBoost 进行模型融合，效果却并不明显，线上排名保持不变。我们分析其原因是这两种模型学习到的信息特征相似，模型融合效果不佳。之后我们在模型融合中加入 CatBoost 模型，最终 RMSE 达到 1.85265，线上提交结果 RMSLE 为 1.77526，相比单独训练的 XGBoost，模型融合获得了一定的改进，最终排名第 297 位，提升了 6 位，这表明了模型融合的有效性。

5.讨论和分析

我们对模型的预测效果进行分析。将 id 为 3002 的样本作为例子，模型对其预测结果为 3007298.93663408。在众多电影票房中，这属于中等层次。而该部电影的 budget 属性为 88000，也是中等层次的预算水平，符合前面我们对这两者的分析，即电影票房与预算存在一定的线性关系。另外，这部电影属于高收益电影，收入高于预算，这是因为该电影的部分属性对票房收入产生了正面影响。在前面的数据分析中，可以知道，电影的主题数选取在 1-3 的范围内对票房收入有一定的正面效益，而该电影的 genres 属性为[{Horror', 'Science Fiction'}]，具有两个主题，同时这两个主题在能产生收益的电影中较为常见。其它属性诸如具有一个标题和标语等都能使模型对该部电影的票房预测造成正面影响。

总体而言，模型的预测效果是比较好的。当然它也存在一定的局限性，对部分数据的预测与真实结果之间存在一定差距，这可能是样本数据本身的不确定性造成的，也可能是模型对数据的特征表示学习不到位。这需要继续加强特征工程这方面的工作，以致提高模型对样本特征信息的学习效率。

6.结论

对于电影票房预测这个项目，模型要想获得好的效果，需要重视特征工程这块工作。在这次项目中，样本的属性比较多、属性类型也比较丰富，即存在数值型属性，也存在文本属性，如何更有效的利用这些特征，需要我们认真去思考。另外，可以通过可视化的方法，去分析各个特征之间的关系，这有利于更合理地利用数据，也有利于构造出新的特征。最后，在模型算法的选择中，LightGBM 和 XGBoost 表现出各自的优势，模型融合效果有明显提升。目前，在实验过程中，我们得出以下结论：

（1）实验数据探索预处理很重要

（2）不同模型的融合效果明显提升

（3）模型最佳参数搜索很关键

（4）目前实验模型效果到达了瓶颈，之后可以继续探索，思考新的解决方法来提升预测准确度。

7.学习体会和建议

这次项目完成之后，感触很深的就是我们研究课题的选择，开始我找了几个 Kaggle 题目备选，我们小组对课题的选取就十分重视，我们集思广益，不断交流想法，最后对每个课题任务进行初步分析，选择了票房预测这一个回归类型的任务。好的开始是成功的一半。

另外我们对本项目选题与老师交流，得到了老师的肯定，并给予了许多建议和指导。通过比赛促进学习，也是这次项目的一个重要目标。从数据分析、预处理，特征工程的构建，不断挖掘数据文本的关键特征信息，建立模型进行结过预测，经过几十次甚至上百次实验，不断调优模型，获得了模型最佳参数设置和模型预测效果。

我们小组 4 人协力分工合作，最终成功地完成了本次文本信息工程与实践的项目研究，并且在该比赛中，以 1.77526 的成绩在 1935 支队伍中获得第 297 名，之后还可以继续努力，这离不开老师的指导和我们小组成员的不断探索。此次大作业，让我真正掌握了理论到实践的知识。

——陈镇鸿

在本次大作业中，我们小组选择了 Kaggle 平台上的电影票房预测这个项目。合适的选题会使我们更加有动力，所以课题的选取十分重要。另外，在比赛平台上选取课题是有很多好处的。首先，平台提供了我们需要的数据集，不必另外花费时间去寻找数据，这有利于我们专心地完成这个任务。其次，我们还可以用比赛提交的测试结果来衡量我们模型的效果。

在这次大作业中，可以将我们平时学到的自然语言处理的理论知识运用起来，这让我们对

其有了更深的体会，我们对自然语言处理这方面的知识也更加感兴趣了。在这个项目中，特征工程是比较重要的，我们模型效果的提升很多时候都是依赖于对数据进行一定的处理，构造出新的特征。这需要我们对实验数据进行一定的探索，其中可利用数据可视化来辅助我们分析各属性特征之间的关系。并且在实验阶段，如何找到最佳的模型参数，如何提高不同模型的融合效果，这些也是关键。就目前而言，我们的模型依然是能有所提升的，但这需要我们继续深入学习深入研究，我们会更加努力，继续加油的。

——高子雄

在这次的实践任务中，我承担了部分文档撰写和 PPT 制作的工作。

我们小组的分工明确，交流合作频繁且高效。有关文档方面，我们的文档更迭了几个版本，并且每次的需求都得到了较好的改进与反馈。

我的感想是：按照模板框架撰写文档的挑战之处在于，我们必须在保持逻辑清晰的情况下适配模板的框架。在这方面我的队友从“旁观者清”的角度为我提供了很好的建议以及优化。

此外我的不足之处表现在对文档格式的认知不够清晰，例如缩进、标题级别等没有一个可以严格执行的标准。

——游畅

本次课程的大作业中，我主要负责的是文档部分的内容。

虽然对于程序实现参与较少，但是在小组的分析与讨论中，由于不少内容我们互相之间需要通过与其他组员沟通来确定，修改内容，这个过程对其他组员的学习，同样让我获益匪浅。

另外，在文档的编写中，对于文案的书写与格式的要求，有不少的成长，老师在答疑过程中也有很多宝贵的意见帮助到文档的工作当中。

在本次任务中，我认为小组整体完成度相对较高，同时仍有进步空间，如在部分文档修改过程中，对于细节分工的沟通相对不够明晰，但整体而言还是在要求内完成了任务。

——管泽陶

8.小组成员贡献

组员姓名	负责内容	贡献比值
陈镇鸿	选题，数据预处理，模型实验，撰写模型算法部分，整合修改文档	25%
高子雄	数据探索分析，模型实验，撰写实验结果	25%
管泽陶	EDA 分析，撰写研究现状，可视化分析	25%
游畅	提供原理资料，撰写摘要，算法细节部分，展示 PPT	25%

参考文献

[1] 王伟. 基于微博数据的电影票房预测研究[D]. 重庆大学.
[2] Chen T , Tong H , Benesty M . xgboost: Extreme Gradient Boosting[J]. 2016.
[3] J Li. Monthly Housing Rent Forecast based on LightGBM (Light Gradient Boosting) Model.
[4] Guolin Ke , Qi Meng, LightGBM: A Highly Efficient Gradient Boosting Decision Tree