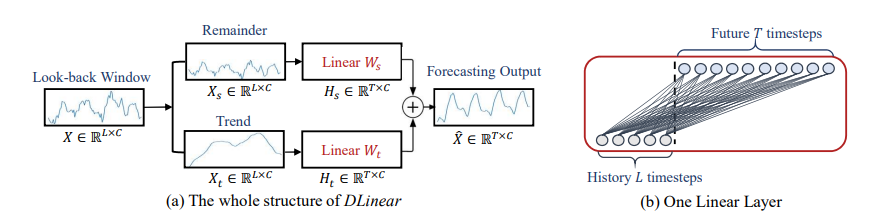
# 基于深度学习…预测票房（依旧没想好高级名字）

## 模型选择与理论基础

在电影票房预测领域，时间序列分析是捕捉票房随时间动态变化的关键方法。每日票房数据通常呈现出明显的趋势性（例如，上映初期的增长和后续的衰减）和季节性（例如，周末效应、节假日效应），同时也受到多种复杂因素的影响，包括电影本身的属性、观众口碑的实时变化以及市场环境等。为了全面捕捉这些不同类型的模式和影响，本研究采用了一种混合建模策略，结合了 D-Linear 模型和 XGBoost 模型，并通过模型堆叠 (Model Stacking) 技术进行融合。

**1.** D-Linear 模型：捕捉时序动态

D-Linear 模型作为一种新兴的时间序列预测线性模型，因其简洁性、高效性以及在多种长时程预测基准任务上超越复杂 Transformer 模型的优异表现而受到关注 [5, 6]。其核心优势在于将时间序列显式地分解为趋势（Trend）和季节性/剩余（Seasonal/Remainder）两个部分，并分别使用独立的单层线性网络进行预测，最后将两部分预测结果相加得到最终预测值 [6, 8]。这种分解-预测的策略非常适合捕捉票房数据中固有的趋势和周期性波动。相较于 ARIMA 等传统统计模型可能存在的调参复杂性 [12, 44] 或 LSTM、Transformer 等深度学习模型的高计算成本和潜在的过拟合风险 [1, 6]，D-Linear 在保证预测精度的同时，提供了更高的计算效率和更强的可解释性 [9]。在本研究中，D-Linear 主要负责处理票房数据的内在时序依赖性，并整合动态协变量（如情感指数）和静态协变量（如电影属性）。



遵循 D-Linear 的架构 [引用]，对于输入的历史每日票房时间序列 *X*（长度为 L，即回顾窗口 *seq\_len*），模型首先执行趋势-季节性分解：

**圈圈1** 趋势成分提取: 通过应用一个移动平均滤波器（核大小为 *k*）来平滑原始序列，提取其长期趋势 *Trend*。

* 数学上近似表示为：

**圈圈2** 季节性/剩余成分提取: 从原始序列中减去提取的趋势成分，得到季节性/剩余成分 *Seasonal*。

* 数学上表示为： 。

**圈圈3** 独立线性预测: 将历史的趋势成分 Trend和季节性成分 *Seasonal* 分别输入到两个独立的单层线性网络（权重分别为 和 ）中，进行未来 *H* 步（预测范围 ，在本研究中主要关注 的单步预测）的预测。

**圈圈4**最终预测合成: 将预测的趋势和季节性成分相加，得到最终的票房预测值 。

而对于复杂多元的变量数据，我们采用了以下的整合策略：

**圈圈1** 目标序列: 核心预测目标，即电影的每日票房收入时间序列。

**圈圈2** 历史协变量:

* 每日情感指数: 由前序章节计算得出的每日情感得分序列。该序列与目标序列在时间上对齐，反映了对应日期的网络舆论情感倾向。我们假设当日或过去几日的情感指数会对未来的票房产生影响。
* 其他可能的动态变量: 例如，每日排片占比、大盘指数等（如果可用）。

**圈圈3** 未来协变量:

* 已知未来信息: 例如，是否为周末、是否为法定节假日等。这些信息通常可以预先知道，并可能显著影响票房。Darts 中的 DLinear 模型可以利用已知的未来信息。

**圈圈4** 静态协变量:

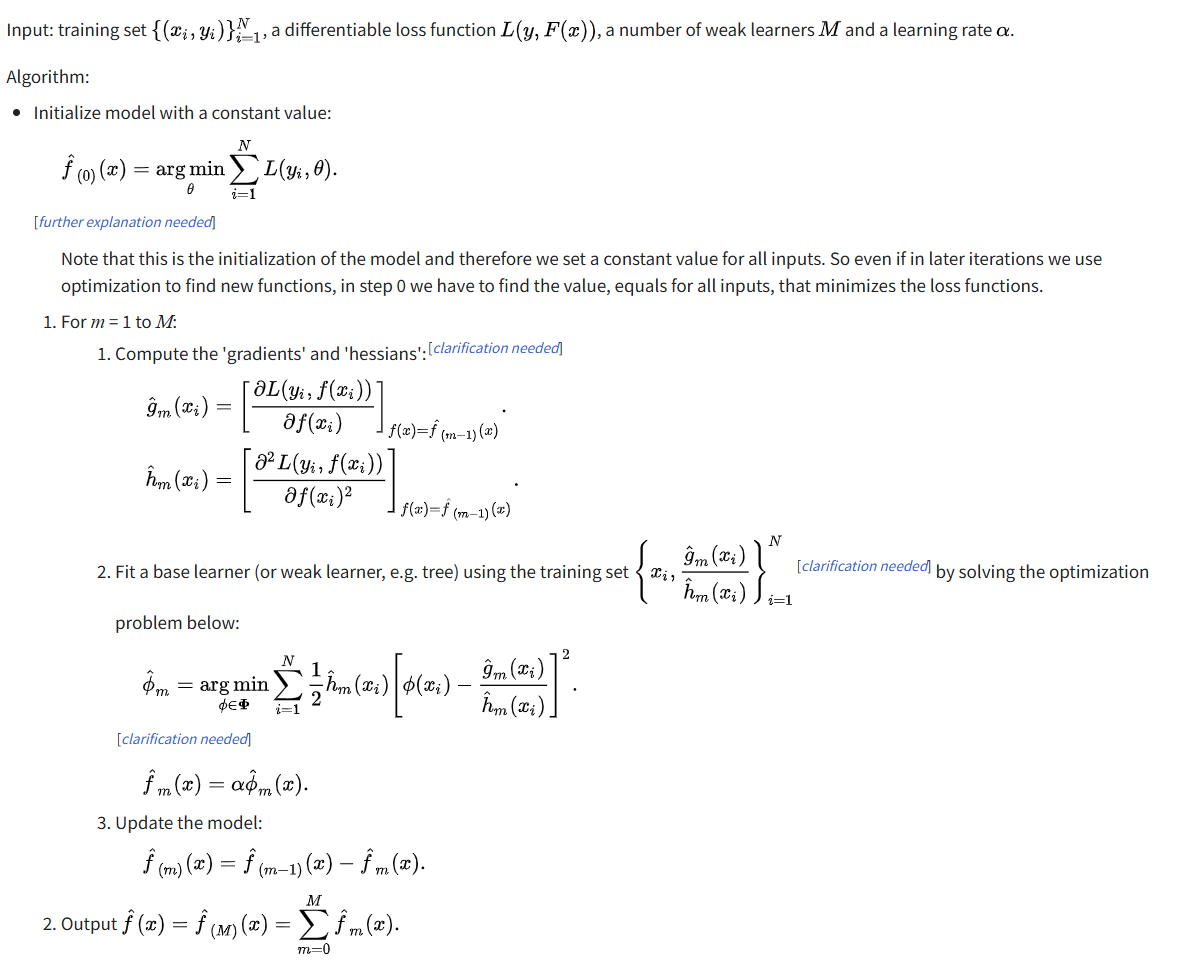
* 电影属性: 包括导演、主要演员（可能需要量化，如基于历史票房或某种评分）、电影类型（可进行独热编码）、制作预算（可进行分箱或归一化处理）、MPAA 分级等 [1]。这些变量在整个预测期间保持不变。Darts 库支持在模型训练时为每个时间序列关联静态协变量，模型可以在内部利用这些信息（例如，通过特定的嵌入层或影响线性层的权重，具体取决于 Darts 的实现细节或模型变种）。

# 通过 *past\_covariates*, *future\_covariates*, 和 *static\_covariates* 参数，Darts 库能够将这些不同类型的变量有效地融入 D-Linear 模型的训练和预测过程，使得模型不仅学习历史票房的自身模式，还能考虑外部动态信息和电影的固有特性。

**2.** XGBoost 模型：捕捉非线性关系与复杂交互

XGBoost (Extreme Gradient Boosting) 是一种基于梯度提升决策树 (GBDT) 的高效、灵活且强大的机器学习算法 [引用 XGBoost 论文]。它通过集成多个弱学习器（决策树）来构建一个强预测模型，在处理表格数据、捕捉特征间的复杂非线性关系和交互效应方面表现出色 [引用]。在票房预测场景中，除了时序模式外，票房还可能受到各种特征（如特定导演与演员组合的效应、情感指数变化的非线性影响、预算与类型间的交互作用等）的复杂影响。XGBoost 擅长从这些特征中学习复杂的映射关系，可以作为 D-Linear 的有力补充。我们将使用历史票房信息（如滞后项）、时间特征、情感指数以及电影静态属性作为 XGBoost 的输入特征。

XGBoost (eXtreme Gradient Boosting) 的核心思想是通过串行地构建多个决策树，并将这些弱学习器的预测结果进行加权求和，最终得到一个强学习器。与传统的梯度提升方法不同，XGBoost 在每一轮迭代中，新的决策树都旨在拟合先前所有决策树预测结果的残差（更准确地说是负梯度）。具体而言，在每一轮训练中，XGBoost 会计算当前模型的损失函数关于预测值的梯度。随后，一个新的决策树会被训练出来，其目标是预测这些梯度。通过这种方式，每一棵新树都在努力纠正之前模型的不足，从而逐步提升整体模型的预测性能。XGBoost 的关键优势在于其对损失函数的优化过程，它不仅考虑了一阶梯度信息，还引入了二阶梯度信息（Hessian 矩阵），这使得其在优化过程中能够更精确地确定下降方向和步长，从而更快地收敛并获得更高的预测精度。此外，XGBoost 还内置了对模型复杂度的正则化项，有效防止过拟合，并具备处理缺失值、并行计算等多种优化特性。



--- 写一个伪代码­

为了使 XGBoost 能有效处理时序预测问题和利用多源信息，我们通常需要进行特征工程，因此我们构建了以下特征：

**圈圈1：**滞后特征: 过去 N 天的每日票房收入 (e.g., lag 1, lag 2, ..., lag 7)。

**圈圈2：**时间特征:

- 星期几

- 一年中的第几周

- 是否周末

- 是否节假日

- 上映天数

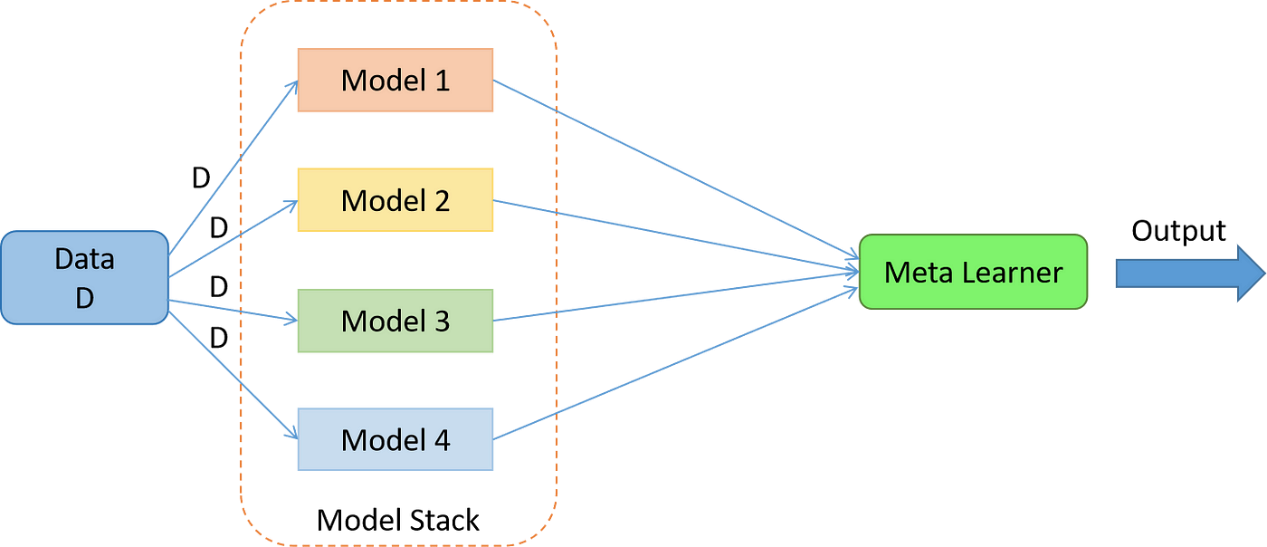
**圈圈3：**移动平均/统计特征: 过去 N 天票房的移动平均值、移动标准差等。

**圈圈4：**情感指数特征: 当天及过去 N 天的每日情感指数。

**圈圈5：**静态特征: 与 D-Linear 类似，将电影属性进行编码（如独热编码、目标编码或 Embedding）。数值特征（如预算）可能需要归一化/标准化。

**3.** 模型堆叠 (Model Stacking)：融合优势

单一模型往往有其局限性。D-Linear 擅长时序分解和线性关系建模，而 XGBoost 更擅长捕捉复杂的非线性特征交互。为了结合两者的优势，我们采用模型堆叠 [引用 Wolpert, 1992 或 Breiman, 1996] 的方法。Stacking 是一种集成学习技术，它使用多个基础模型（Level 0 models，此处为 D-Linear 和 XGBoost）的预测结果作为新的特征，来训练一个元模型（Level 1 model），由元模型产生最终的预测输出。这种方法旨在通过学习如何最优地组合基础模型的预测，来获得比任何单一基础模型更好的性能。



模型堆叠通常包含两个层次：

* Level 0 (基础模型层): 包含我们选择的 D-Linear 和 XGBoost 模型。
* Level 1 (元模型层): 一个用于组合 Level 0 模型预测结果的简单模型。

在训练时，为了防止数据泄露，Level 0 模型的预测结果（用于训练 Level 1 元模型）必须通过交叉验证 (Cross-Validation) 的方式产生：

--- 这部分写个伪代码

1. 将原始训练集划分为 K 折 (e.g., K=5)。

2. 进行 K 次迭代：

- 在第 i 次迭代中，使用 K-1 折数据训练 D-Linear 和 XGBoost 模型。

- 用训练好的模型对剩余的第 i 折（验证集）进行预测。记录下这些 "out-of-fold" (OOF) 预测值。

3. K 次迭代后，所有训练集样本都有了对应的 OOF 预测值。这些 OOF 预测值（来自 D-Linear 的预测 和 来自 XGBoost 的预测）将作为 Level 1 元模型的训练特征。对应的真实票房值作为训练目标。

4. 使用 完整的 原始训练集分别训练最终的 D-Linear 和 XGBoost 模型。

---

**4.** 变量整合策略概述

本研究的创新之处在于将前序章节构建的每日网络情感指数作为动态协变量/特征，并将电影的固有属性（如导演、演员、类型、制作预算等）作为静态协变量/特征，同时融入 D-Linear 和 XGBoost 模型的训练中。我们假设，实时的网络情感反馈能够捕捉到观众对电影口碑的动态变化，而电影的静态属性则决定了其基础的票房潜力。通过两个异构模型的分别建模和最终的 Stacking 融合，我们期望更全面地利用这些信息，实现更精准的票房预测。

## 数据准备与模型训练

**1.** 数据收集与预处理

数据来源: 历史每日票房数据来源于猫眼专业版；网络评论数据来源于 小红书、知乎、豆瓣、微博等；电影静态属性数据来源于豆瓣电影，kaggle 数据集。

数据清洗:

* 缺失值处理: 对每日票房数据和情感指数序列中的缺失值进行检查。少量缺失可采用前向填充 (forward fill) 或线性插值 [14]；若单部电影缺失数据过多，考虑剔除该样本 [25]。
* 异常值处理: 检测票房数据中可能存在的极端异常值（例如，数据录入错误），可采用如 3σ 法则进行识别，并根据情况进行修正或平滑处理 [14]。

特征工程:

* 情感指数: 使用第二章所述方法生成每日情感指数序列。
* 静态特征编码: 对类别型静态特征（如类型、导演、演员）进行处理，例如使用独热编码 (One-Hot Encoding) 或目标编码 (Target Encoding)。对数值型静态特征（如预算）进行归一化或标准化。
* 时间特征: 生成是否周末、是否节假日等二元特征，作为未来协变量。
* XGBoost 特定: 滞后特征、移动平均特征等。

数据标准化: 对目标序列（票房收入）和动态数值协变量（情感指数）应用标准化 (Standardization) 或最小-最大归一化 (Min-Max Scaling) [5, 32]，以消除量纲影响，加速模型收敛。选择哪种方法取决于数据分布特性。

**2.** 实验环境与模型配置

**实现工具:** 使用 Python 语言及 Darts [10] 时间序列预测库进行模型实现。Darts 提供了 DLinear 模型的便捷接口。Python, Darts, xgboost, scikit-learn (用于 Stacking 的交叉验证和元模型)。

**数据集划分:** 对每部电影的票房时间序列数据，按照时间顺序划分为训练集、验证集和测试集。例如，可以采用前 70% 的数据作为训练集，接下来的 15% 作为验证集（用于调整超参数），最后 15% 作为测试集（用于最终评估模型性能）。需确保划分过程不破坏时间序列的连续性（即“未来”数据不泄露给训练阶段）。

# 这部分怎么写？D-Linear 超参数:

* `input\_chunk\_length` (回顾窗口 `L`): 模型进行预测时回看的历史时间步数。需要通过验证集进行调优，例如尝试 [7, 14, 21, 30] 天等。
* `output\_chunk\_length` (预测范围 `H`): 模型一次性预测的未来时间步数。本研究主要关注 `H=1` 的每日预测。
* `kernel\_size` (移动平均核大小 `k`): 用于趋势提取的移动平均窗口大小。通常选择奇数，如 [5, 7, 9, ...]。需要调优。
* 其他训练参数: 如 `n\_epochs` (训练轮数), `batch\_size` (批处理大小), `optimizer` (优化器，如 Adam), `lr` (学习率)。这些参数也需要在验证集上进行选择。

# XGBoost 超参数: (n\_estimators, max\_depth, learning\_rate, gamma, subsample, colsample\_bytree, etc.) 通过验证集或交叉验证调优。

# Stacking 配置: K-Fold 数量 (K)，元模型选择及其超参数（如有）。

# 协变量使用: 在 Darts 的 `DLinearModel` 初始化和 `fit()` 方法中，明确传入处理好的 `past\_covariates`, `future\_covariates`, 和 `static\_covariates`。

**3.** 模型评估

为了定量评估 D-Linear 模型在每日票房预测任务上的性能，我们选用以下常用的回归评估指标 [19, 33]：

1. 均方根误差 (Root Mean Squared Error, RMSE):

RMSE 对较大的预测误差更为敏感 [19]。其量纲与目标变量（票房收入）相同，易于解释，但数值大小受票房绝对值影响。

2. 平均绝对误差 (Mean Absolute Error, MAE):

MAE 直接衡量预测误差的平均绝对大小，对异常值不如 RMSE 敏感 [19]。同样，其量纲与目标变量相同。

我们将计算模型在测试集上关于这些指标的值，并与基线模型（例如，朴素预测 - 使用上一天的票房作为预测值；简单的移动平均模型；或不包含情感指数和静态特征的 D-Linear 模型）进行比较，以验证我们提出的整合方法的有效性。

## 预测结果与分析

## 讨论与小结