



# 電影票房預測系統

報告人：方佳文



# 用兩周的票房 預測未來一周的營收！



輔助商業決策



即時預警系統



破除網路口碑假象

# 票房預測 X 異常警示 X 行動建議

### 電影票房預測儀表板

AI 駕駛的票房預測與分析平台

本週上映電影: 12  
預測準確率: 92%  
R<sup>2</sup>: 0.96

預警電影: 3  
總銷售數: 156  
本月累計: 156

快速開始:

- 查看範例電影預測
- 瀏覽所有電影
- 預測分析工具
- 下載報表

最新預測結果:

電影名稱	當前週次	下週預測	衰退率	狀態	操作
科技風暴	第8週	NT\$850萬	-28%	注意	查看詳情
愛在深秋	第10週	NT\$320萬	-15%	正常	查看詳情

### 科技風暴

電影基本資訊

片長: 148 分鐘  
導演: 張導演  
演員: 李演員, 王演員, 蔣演員  
發行國家: 台灣  
上映日期: 2025年10月01日

總票房: NT\$1.6億  
總觀影人數: 523,525  
平均票價: NT\$300

最高總票房: NT\$5000.0萬  
平均衰退率: -30.0%  
衰退較快

### 票房趨勢與未來預測

票房趨勢與預測

總票房: NT\$5000.0萬  
總觀影人數: 523,525  
平均票價: NT\$300

票房預測詳細數據

週次	類型	票房	觀影人數	票數	衰退率	信心區間
第7週	高規	NT\$588.2萬	19,608	30	-	-
第8週	高規	NT\$411.8萬	13,725	20	-	-
第9週	中規	NT\$288.2萬	9,608	-	-30.0%	NT\$230.6萬 ~ NT\$345.9萬
第11週	低規	NT\$201.8萬	6,725	-	-30.0%	NT\$161.4萬 ~ NT\$242.1萬
第13週	低規	NT\$141.2萬	4,707	-	-30.0%	NT\$113.0萬 ~ NT\$169.5萬

票票房預測的關鍵指標是.....

衰退

# 三大模型的較量

我們測試了三種不同的模型，交叉驗證模型的準確度與可解釋性！

## 線性迴歸

$R^2 = 0.956$

易於解釋票房的衰退特性  
但無法捕捉極端值的非線性關係  
(EX:強檔大片)

## 決策樹迴歸

$R^2 = 0.877$

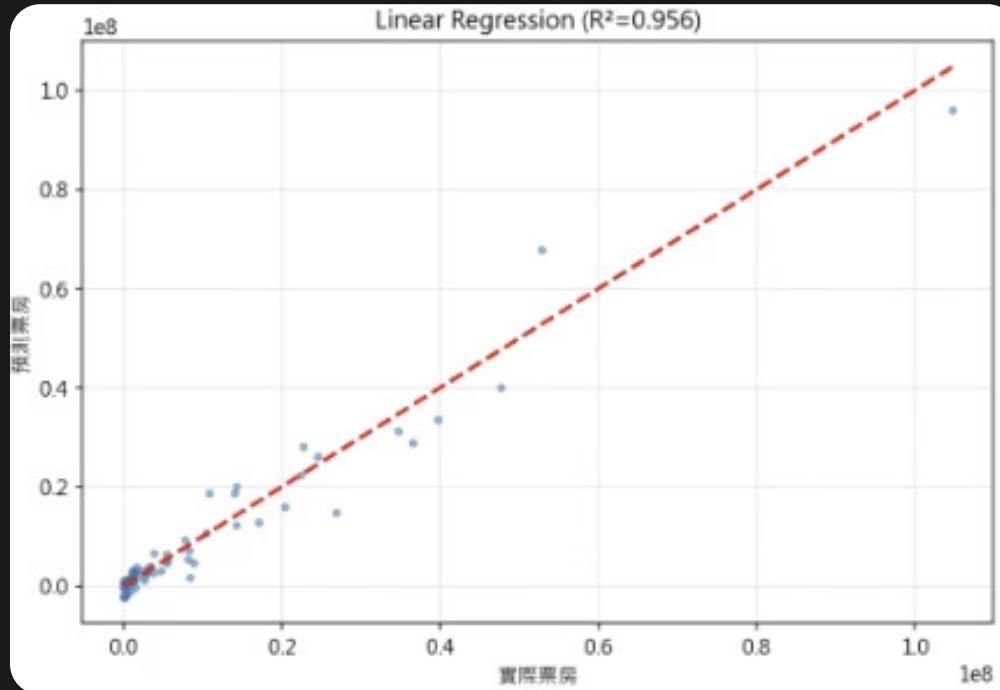
能處理非線性關係，把資料切成  
不同區塊，易於解釋每個分支決  
策，但容易過擬合

## LightGBM

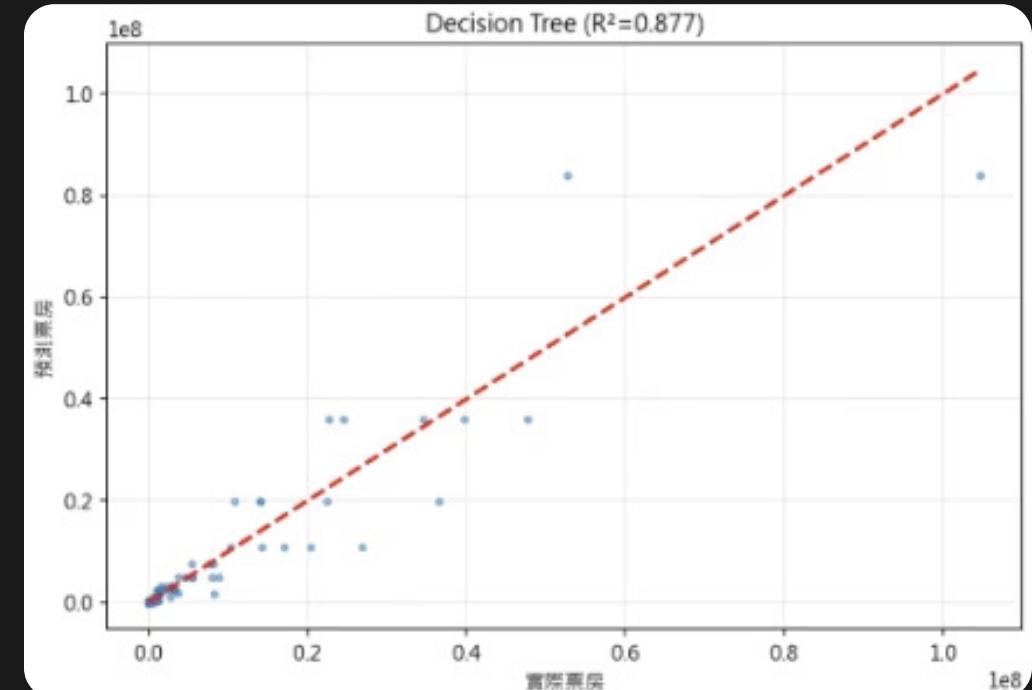
$R^2 = 0.829$

一樣能處理非線性關係，相較於  
單一樹，更好的處理過度擬合的  
問題，且能處理更大的資料量

# 線性模型 VS 迴歸決策樹



$R^2 = 0.956$

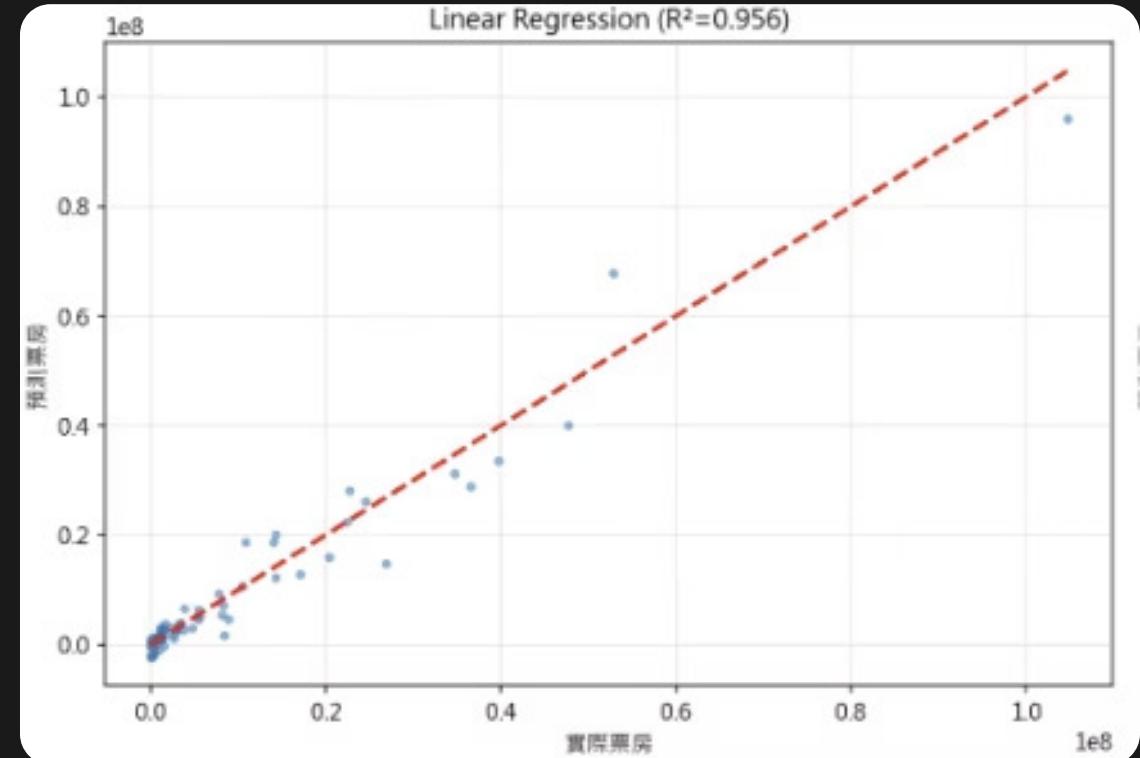


$R^2 = 0.877$

# 為什麼線性模型表現更好？

線性衰退的特性是  
線性迴歸的強項

決策樹 試圖捕捉非線性關係  
反而學到了資料中的 雜訊  
導致預測能力下降



# 資料來源與處理



## 全國票房統計

資料範圍涵蓋2025/9/29 至 11/2  
共五週在院線上映的約130部電影



## 每週更新

映後七天公布上週票房  
實際數據每天都會進行微幅調整



## 離群值處理

移除80筆異常資料  
5部電影

860

## 總訓練筆數

124部電影的完整票房資料

19

## 特徵欄位

有35個特徵欄位可選擇

8:2

## 訓練集比例

647筆訓練 / 213筆測試

# 正在爬《全國電影票房統計》的資料....

The screenshot shows a terminal window titled 'Cinpos\_model' running in a dark-themed code editor. The left sidebar displays a file tree for a project named 'CINPOS\_MODEL'. The current directory is 'scripts'. Inside 'scripts', there is a file named 'run\_boxoffice\_batch.py'. The terminal output shows the command being run:

```
(.venv) C:\Users\Wen\Desktop\Projects\Cinpos_model>uv run scripts/run_boxoffice_batch.py --start 2025-06-01 --end 2025-08-29 --interval 7
```

The script itself contains the following code and documentation:

```
#!/usr/bin/env python
# ...
# 票房資料批次處理腳本
# -----
# 功能說明:
# - 指定時間範圍和頻率，批次執行票房資料的爬取與清洗流程
# - 按照固定順序執行 5 個步驟：
#   1. 爬取單周電影票房
#   2. 清洗單周票房資料
#   3. 爬取單部電影累計票房
#   4. 清洗單部電影票房資料
#   5. 合併電影資訊
```

The terminal also shows the user's environment variables and session information at the bottom.

票房資料要怎麼處理？

聚合 vs 展開

# 建模成功的關鍵：特徵欄位設計

01

## 時間序列特徵

輪次、真實週次、活躍週次,精確追蹤電影生命週期

02

## 近期趨勢特徵

前兩週的票房、觀影人次、院線數,捕捉衰退動能

03

## 開片實力特徵

首週票房、首輪日均票房,判斷電影量級與基準表現

04

## 累積資訊特徵

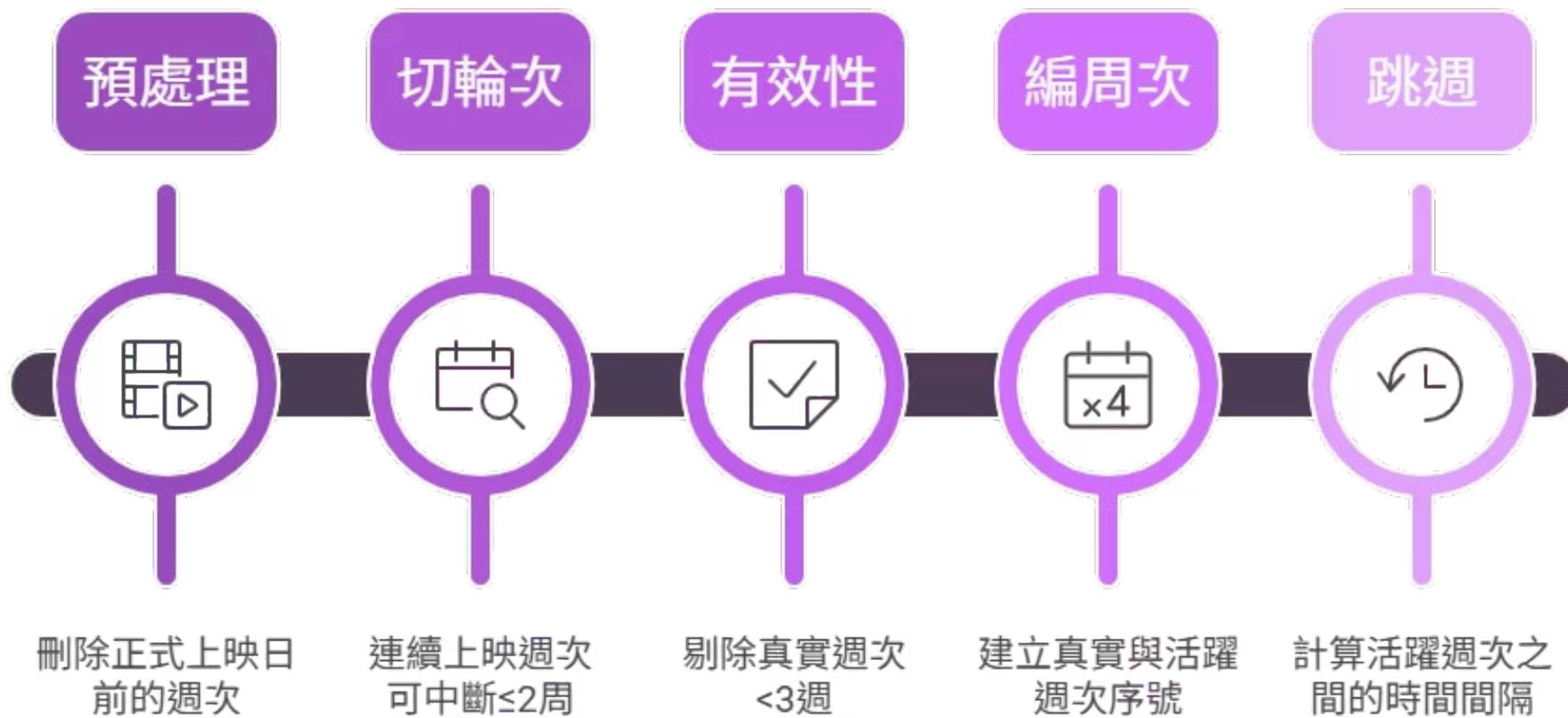
歷史累計票房與觀影人次,評估長期表現

05

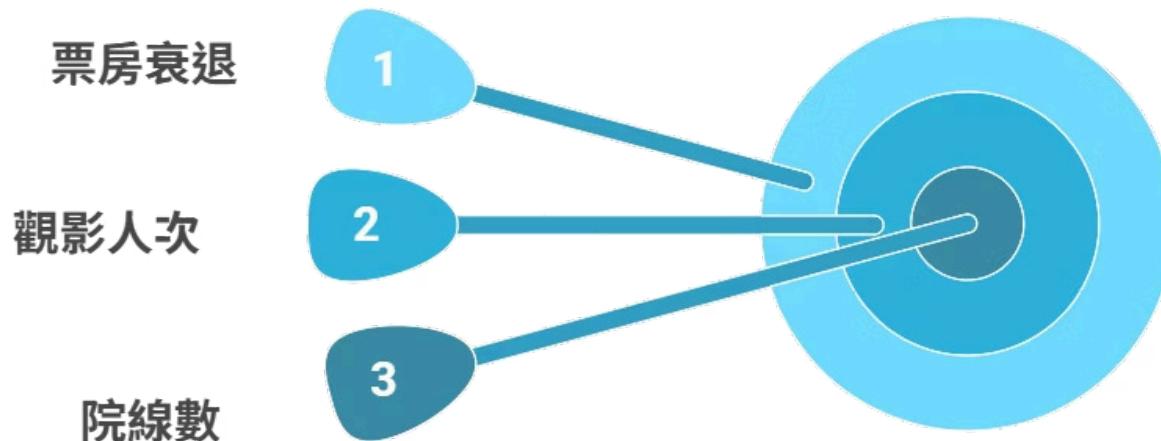
## 市場資訊特徵

上映檔期、片長、分級,反映市場環境影響

# 時間序列特徵



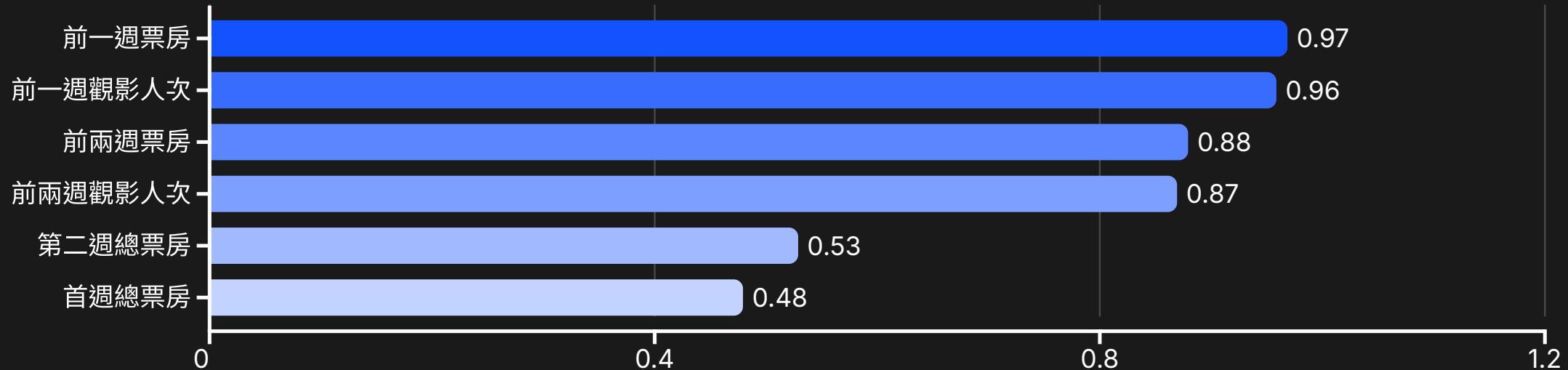
# 趨勢特徵：讓模型學習 衰退



# 其他特徵



# Top 10 相關特徵重要性分析



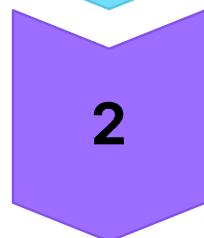
# 資料流架構設計



1

## 階段一：爬取原始資料

從《全國電影票房統計》抓取上週上映電影清單，並逐部查詢詳細資訊



2

## 階段二：時間序列攤平

將逐週票房進行時間排序,建立輪次、真實週次與活躍週次的對應關係



3

## 階段三：特徵工程

計算近期趨勢、累積資訊、開片實力與市場資訊等35個特徵欄位，後續可因應模型需求刪減

整個資料流設計具有高度彈性與可擴充性，可依據模型需求靈活調整特徵組合。

# The End