ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH

BÀI TẬP LỚN MÔN KHAI PHÁ DỮ LIỆU (CO3029)

Phân tích và dự báo doanh thu ngành điện ảnh

GVHD: Dỗ Thanh Thái

SVTH1: Nguyễn Quốc Thịnh - 2213296
SVTH2: Trần Đình Tưởng - 2213892
SVTH3: Trần Quốc Khánh - 2311538

Nhóm 8 – Lớp L02

Mã nguồn dự án:
github.com/TranDinhTuong/Movie-Analytics-CO3029

Video báo cáo:
https://youtu.be/hzQeWGdUEEo

TP. Hồ Chí Minh năm học HK251

Mục lục

1 Giới thiệu

1.1 Bối cảnh nghiên cứu

Ngành công nghiệp điện ảnh là một thị trường giải trí toàn cầu với giá trị kinh tế khổng lồ và mức độ cạnh tranh gay gắt. Sự thành bại của một tác phẩm điện ảnh là một ẩn số phức tạp, chịu sự chi phối của vô số yếu tố định lượng và định tính, bao gồm ngân sách sản xuất, sự lựa chọn diễn viên, thể loại, thời điểm phát hành, và phản ứng ban đầu của khán giả.

Trong bối cảnh Dữ liệu lớn (Big Data) và sự bùng nổ của các nền tảng phân tích, việc khai thác thông tin từ các tập dữ liệu lịch sử không chỉ là cơ hội mà còn là yêu cầu chiến lược. Khả năng **dự đoán chính xác doanh thu** của một bộ phim trước khi ra rạp cung cấp một lợi thế cạnh tranh then chốt, hỗ trợ các nhà sản xuất và nhà đầu tư đưa ra các quyết định sáng suốt về marketing và phân bổ nguồn lực.

Bài tập lớn môn **Khai phá Dữ liệu (CO3029)** này được thực hiện với mục tiêu áp dụng các kỹ thuật học máy tiên tiến để định lượng hóa sự thành công thương mại trong lĩnh vực điện ảnh, từ đó góp phần giải quyết bài toán dự báo đầy thách thức này.

1.2 Mục tiêu Báo cáo

Báo cáo này là kết quả của quá trình áp dụng chu trình Khai phá Dữ liệu (Data Mining Pipeline) và được xây dựng để đạt được các mục tiêu chính sau:

1. **Phân tích Khám phá Dữ liệu (EDA):** Khám phá và xác định các đặc trưng quan trọng nhất trong bộ dữ liệu điện ảnh có ảnh hưởng mạnh mẽ đến biến mục tiêu là doanh thu.

2. **Xây dựng Mô hình Dự đoán:** Triển khai một mô hình học máy dựa trên các đặc trưng đã được tiền xử lý để dự báo doanh thu của phim.

3. **Đánh giá Hiệu suất và Thảo luận Thách thức:** Đánh giá hiệu quả của mô hình thông qua các chỉ số đo lường thống kê (R^2, RMSE), đồng thời làm rõ các thách thức kỹ thuật gặp phải. Đặc biệt, báo cáo sẽ phân tích **hiện tượng overfitting** và nguyên nhân sâu xa của nó (do đặc tính phân phối lệch của biến doanh thu và sự xuất hiện của các outlier) như đã được quan sát trong quá trình nghiên cứu thực nghiệm.

1.3 Bộ dữ liệu sử dụng

	id	genre_ids	popularity	vote_average	vote_count	budget	original_language	revenue
0	19995	[28, 12, 14, 878]	150.437577	7.2	11800	237000000	en	2.787965e+09
1	285	[12, 14, 28]	139.082615	6.9	4500	300000000	en	9.610000e+08
2	206647	[28, 12, 80]	107.376788	6.3	4466	245000000	en	8.806746e+08
3	49026	[28, 80, 18, 53]	112.312950	7.6	9106	250000000	en	1.084939e+09
4	49529	[28, 12, 878]	43.926995	6.1	2124	260000000	en	2.841391e+08
5	559	[14, 28, 12]	115.699814	5.9	3576	258000000	en	8.908716e+08

Hình 1: Bộ dữ liệu sử dụng

Bộ dữ liệu được sử dụng trong nghiên cứu này là tập dữ liệu metadata của các bộ phim, được thu thập từ nền tảng **The Movie Database (TMDB)** và được truy cập thông qua các nguồn dữ liệu công khai (ví dụ: Kaggle). Bộ dữ liệu này cung cấp một cái nhìn toàn diện về các yếu tố ảnh hưởng đến thành công thương mại của phim.

Tập dữ liệu chứa thông tin phong phú về các thuộc tính quan trọng, bao gồm: Ngân sách (*budget*), doanh thu (*revenue*), thể loại (*genres*), độ nổi tiếng (*popularity*), ngôn ngữ gốc (*original language*), và các chỉ số đánh giá cộng đồng. (*vote average và vote count*)

2 Cơ sở Lý thuyết

2.1 Thuật toán Random Forest (Rừng ngẫu nhiên)

Thuật toán **Random Forest (Rừng ngẫu nhiên)** là một trong những mô hình học máy thuộc nhóm **Học tập Tổ hợp (Ensemble Learning)** được sử dụng để giải quyết bài toán hồi quy trong báo cáo này. Nó được phát triển dựa trên sự kết hợp của nhiều cây quyết định độc lập nhằm tăng cường tính ổn định và khả năng dự đoán so với việc chỉ sử dụng một cây đơn lẻ.

2.1.1 Cơ chế hoạt động và Dự đoán

Random Forest xây dựng một tập hợp các cây quyết định thông qua kỹ thuật **Bagging (Bootstrap Aggregating)**, kết hợp với việc lựa chọn ngẫu nhiên các đặc trưng tại mỗi nút phân tách. Sự ngẫu nhiên hóa này giúp giảm thiểu phương sai và ngăn chặn hiện tượng *overfitting* thường thấy ở các cây quyết định.

- Mỗi cây được huấn luyện trên một tập hợp con dữ liệu được lấy mẫu ngẫu nhiên và có thay thế (bootstrap sample). Tuy nhiên, trong các thư viện như scikit-learn, Bootstrap có thể được bật/tắt thông qua tham số bootstrap=True/False.
- Tại mỗi điểm phân tách, chỉ một tập hợp con ngẫu nhiên của các đặc trưng được xem xét.

Trong bài toán hồi quy (dự đoán doanh thu), kết quả dự đoán cuối cùng của mô hình là **giá trị trung bình** của các dự đoán được tạo ra bởi tất cả các cây quyết định trong rừng.

2.1.2 Vai trò và Hạn chế trong Dự án

Random Forest được sử dụng như một mô hình baseline mạnh mẽ ban đầu. Mô hình cho kết quả tốt đối với nhóm phim có doanh thu thấp và trung bình, đồng thời giúp xác định **tầm quan trọng của các đặc trưng** trong việc dự đoán doanh thu. Tuy nhiên, qua quá trình đánh giá, Random Forest đã bộc lộ hạn chế nghiêm trọng: mô hình gặp phải hiện tượng **overfitting** mạnh (thể hiện qua khoảng cách lớn giữa training score và cross-validation score). Nguyên nhân chính là do **đặc tính phân phối lệch phải** của biến doanh thu (*revenue*) và sự tồn tại của nhiều *outlier* (giá trị ngoại lai), khiến mô hình khó học được các quy luật tổng quát.

3 Tiền xử lý dữ liệu

```python
def load_tmdb_movies_df(input_csv_path: str) -> pd.DataFrame:
    """
    Đọc file TMDB, chuẩn hóa các cột số, giữ nguyên cột 'genres' và 'original_language'
    để xử lý sau bằng MultiLabelBinarizer và OneHotEncoder.
    """
    df = pd.read_csv(input_csv_path, encoding="utf-8")
    df=encode_genres(df)
    df["original_language"]=df["original_language"].apply(
        lambda x:x if x == "en" else "other"
    )
    #print(df["original_language"].value_counts())

    # Chuẩn hóa kiểu dữ liệu số
    numeric_cols = ['popularity', 'vote_average', 'vote_count', 'budget', 'revenue']
    for c in numeric_cols:
        if c in df.columns:
            df[c] = pd.to_numeric(df[c], errors='coerce')

    # Loại bỏ revenue bị 0 hoặc NaN
    if 'revenue' in df.columns:
        df['revenue'] = df['revenue'].replace(0, np.nan)
        df = df.dropna(subset=['revenue'])

    # Giữ lại các cột cần thiết (không đụng tới genres và original_language)
    cols = ['id', 'genre_ids', 'popularity', 'vote_average', 'vote_count',
            'budget', 'original_language', 'revenue']
    df = df[[c for c in cols if c in df.columns]].copy()

    return df
```

```python
def prepare_dataset_for_sklearn(df: pd.DataFrame, target: str = "revenue"):
    data = df.copy()
    #data=data.explode("genre_ids")
    # Chỉ impute 4 cột này (0/NaN -> mean)
    cols_impute = ["popularity", "vote_average", "vote_count", "budget"]
    impute_zero_nan_with_mean(data, cols_impute)

    numeric_features = ["popularity", "vote_average", "vote_count", "budget"]
    lang_feature = ["original_language"]
    gen_feature=["genre_ids"]
    mlb=MultiLabelBinarizer()
    genre_ohe=mlb.fit_transform(data["genre_ids"])
    genre_cols = [f"genre_{g}" for g in mlb.classes_]
    preprocessor=ColumnTransformer(
        [
            ("num","passthrough",numeric_features),
            ("lang",OneHotEncoder(sparse_output=False),lang_feature),
        ],
        remainder="drop"
    )


    X_transform = preprocessor.fit_transform(data)
    X=np.hstack([X_transform,genre_ohe])
    num_cols = numeric_features
    lang_cols = preprocessor.named_transformers_["lang"].get_feature_names_out(lang_feature)
    feature_cols = list(num_cols) + list(lang_cols) + genre_cols
    y = data[target].astype(float).values
    X=pd.DataFrame(X,columns=feature_cols)
    y=pd.DataFrame(y,columns=[target])
    return X, y, feature_cols
```

3.1 Làm sạch dữ liệu

Các bước chính:

- Chuyển tất cả cột số về numeric.
- Giá trị `revenue = 0` hoặc `NaN` được loại bỏ.
- Các cột số (budget, popularity, vote_count,...) được thay 0 bằng mean.

3.2 Xử lý thể loại phim

- Trường `genres` là chuỗi JSON.
- Parse thành danh sách ID thể loại.
- Mã hóa bằng MultiLabelBinarizer (tạo nhiều cột genre_x).

3.3 Xử lý ngôn ngữ phim

- Vì đa số là ngôn ngữ en, nên ở cột original language, các hàng có giá trị là en sẽ giữ nguyên, không sẽ là "other"
- Mã hóa bằng OneHotEncoder.

3.4 Tạo ma trận đặc trưng

Tổng số đặc trưng sau xử lý:

- 4 đặc trưng số
- 2 đặc trưng ngôn ngữ
- >20 đặc trưng thể loại

```
print(X.head(5))
```

```
   popularity  vote_average  vote_count       budget  original_language_en
0  150.437577           7.2     11800.0  237000000.0                   1.0
1  139.082615           6.9      4500.0  300000000.0                   1.0
2  107.376788           6.3      4466.0  245000000.0                   1.0
3  112.312950           7.6      9106.0  250000000.0                   1.0
4   43.926995           6.1      2124.0  260000000.0                   1.0

   original_language_other  genre_12  genre_14  genre_16  genre_18  ...  \
0                      0.0       1.0       1.0       0.0       0.0  ...
1                      0.0       1.0       1.0       0.0       0.0  ...
2                      0.0       1.0       0.0       0.0       0.0  ...
3                      0.0       0.0       0.0       0.0       1.0  ...
4                      0.0       1.0       0.0       0.0       0.0  ...

   genre_53  genre_80  genre_99  genre_878  genre_9648  genre_10402  \
0       0.0       0.0       0.0        1.0         0.0          0.0
1       0.0       0.0       0.0        0.0         0.0          0.0
2       0.0       1.0       0.0        0.0         0.0          0.0
3       1.0       1.0       0.0        0.0         0.0          0.0
4       0.0       0.0       0.0        1.0         0.0          0.0

   genre_10749  genre_10751  genre_10752  genre_10769
0          0.0          0.0          0.0          0.0
1          0.0          0.0          0.0          0.0
2          0.0          0.0          0.0          0.0
3          0.0          0.0          0.0          0.0
4          0.0          0.0          0.0          0.0

[5 rows x 25 columns]
```

Hình 2: Dữ liệu sau bước tiền xử lý

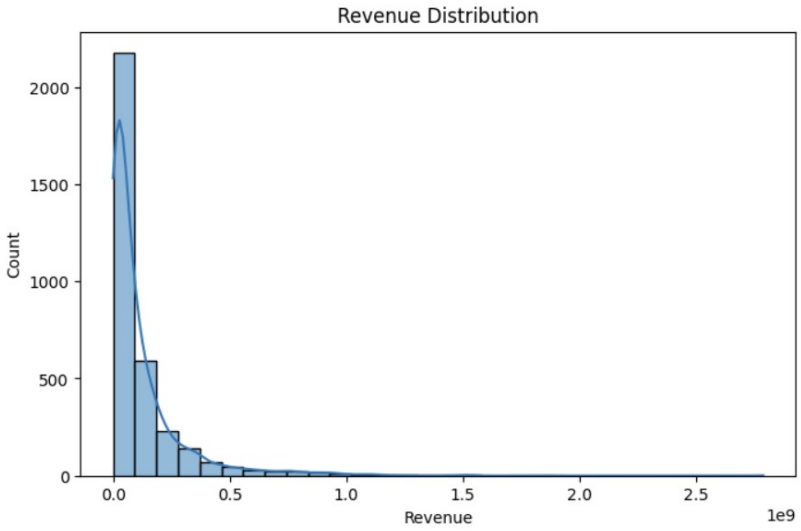
4 Xây dựng mô hình

4.1 Lý do chọn Random Forest

- Mạnh trên dữ liệu dạng tabular
- Hiệu quả với quan hệ phi tuyến
- Chống overfitting nhờ cơ chế bagging
- Không cần chuẩn hóa dữ liệu đầu vào

4.2 Log-transform biến mục tiêu

Do doanh thu phân phối lệch phải:



Hình 3: Phân phối doanh thu phim trong dataset

$$y' = \log(1 + y)$$

Mô hình sử dụng TransformedTargetRegressor sẽ train với log(1+revenue) nhưng sẽ tự động đảo ngược log khi dự đoán.

4.3 Tuning tham số

Sử dụng **RandomizedSearchCV** với:

- n_estimators: 100, 300, 500, 800
- max_depth: None, 10, 20, 30, 40, 50
- min_samples_split: 2, 5, 10, 20
- min_samples_leaf: 1, 2, 5, 10
- max_features: `sqrt`, `log2`, 0.3, 0.5, `None`

4.4 Model cuối cùng

```
log_transformer = FunctionTransformer(np.log1p, inverse_func=np.expm1, validate=True)

model=TransformedTargetRegressor(
    regressor=RandomForestRegressor(random_state=42),
    #random_state= seed ngẫu nhiên,seed=42=> khi mà ông gọi random=>7,10,30,1,4
    transformer=log_transformer
    )
from sklearn.model_selection import RandomizedSearchCV
#gridsearch
param_distrib={
    "regressor__n_estimators":[100,300,500,800],
    "regressor__max_depth":[None,10,20,30,40,50],
    "regressor__min_samples_split":[2,5,10,20],
    "regressor__min_samples_leaf":[1,2,5,10],
    "regressor__max_features":["sqrt","log2",0.3,0.5,None],
    "regressor__bootstrap":[True,False]
}
# 1 mô hình machine learning, sẽ ko biết chọn các tham số trên như nào cho hợp lý
#dòng randomied_search
rnd=RandomizedSearchCV(
    model,# cái mô hình
    param_distributions=param_distrib,#
    n_iter=30,# chạy 30 lần, chọn mô hình tốt nhất trong 30 lần
    cv=5,
    scoring="neg_root_mean_squared_error",
    #sqrt((y_train_du_doan-y_train_that)^2)/ (samples)
    n_jobs=-1,#tận dụng gpu chạy song song
    random_state=42

)
```

- Random Forest đã tuning

5 Phân tích và Đánh giá Mô hình

5.1 Đánh giá Hiệu suất Tổng quan

Phần này trình bày các chỉ số hiệu suất chính của mô hình Random Forest trên tập kiểm tra (Test Set) sau khi hoàn thành quá trình huấn luyện.

```
MSE train: 1103265448334052.50, RMSE train: 33,215,439.91, R2 train: 0.962
MSE test : 15100617485585192.00, RMSE test : 122,884,569.76, R2 test : 0.707
```

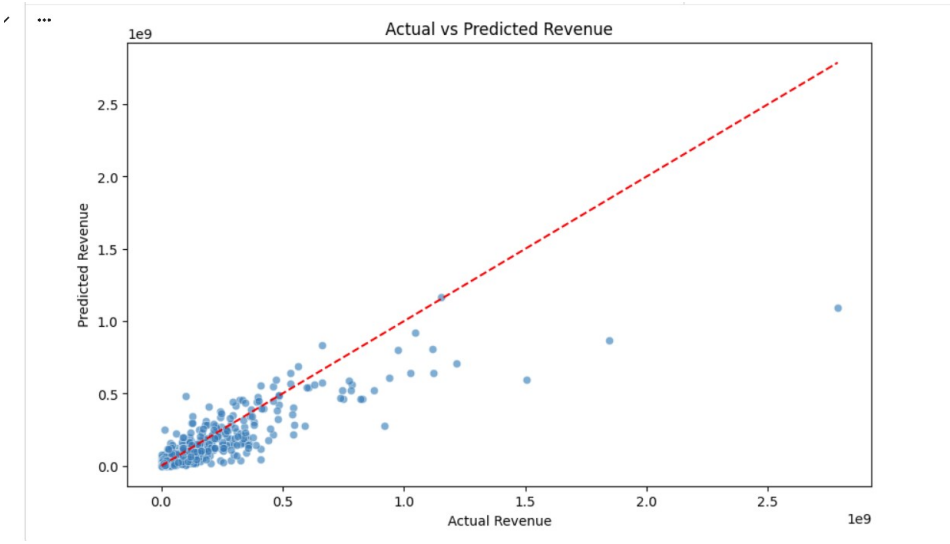
Hình 4: Các chỉ số hiệu suất của mô hình Random Forest

- **Sai số Căn quân phương (RMSE) trên tập huấn luyện (Train)**: Đạt khoảng 33 triệu USD.
- **Sai số Căn quân phương (RMSE) trên tập kiểm tra (Test)**: Cao hơn nhiều, đạt khoảng 122 triệu USD. Khoảng cách lớn giữa hai giá trị này là dấu hiệu rõ ràng của hiện tượng **overfitting**.

- **Hệ số Xác định (R^2) trên tập kiểm tra**: Đạt khoảng 0.707, cho thấy mô hình giải thích được khoảng 70.7% phương sai của biến doanh thu.

5.2 Phân tích Dự đoán (Actual vs Predicted)

Biểu đồ so sánh giữa doanh thu thực tế và doanh thu dự đoán giúp trực quan hóa khả năng dự đoán của mô hình trên từng điểm dữ liệu.

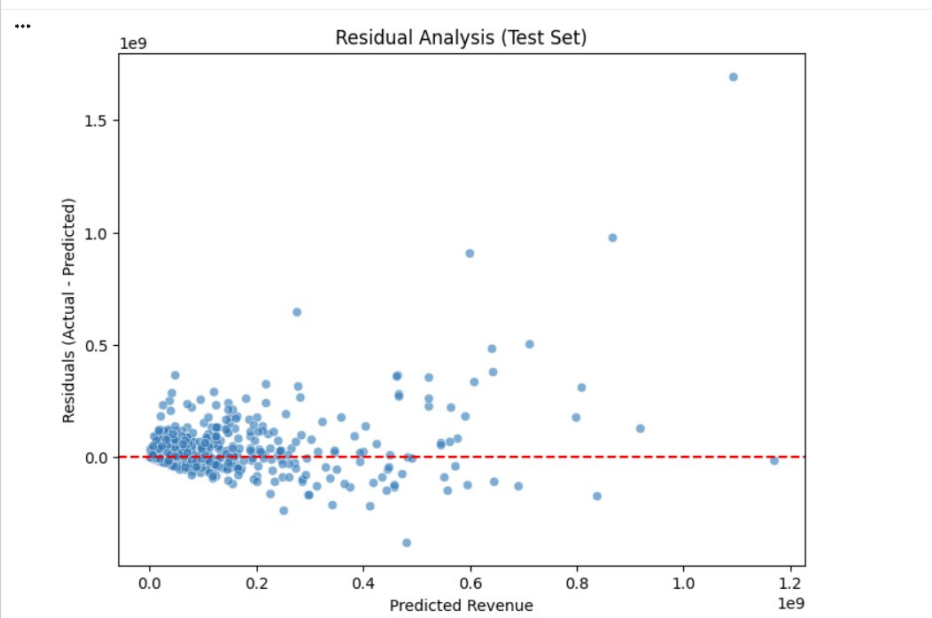


Hình 5: Biểu đồ so sánh giữa doanh thu dự đoán và doanh thu thật dựa trên test set

Mô hình dự đoán tốt các bộ phim có doanh thu thấp và trung bình. Tuy nhiên, mô hình trở nên **kém chính xác** và có xu hướng đánh giá thấp đáng kể đối với các bộ phim bom tấn, tức là các giá trị ngoại lai (*outliers*) có doanh thu rất cao.

5.3 Phân tích Phần dư (Residual Analysis)

Phân tích phần dư cho thấy sự phân bố của sai số dự đoán (Residuals = Actual - Predicted).

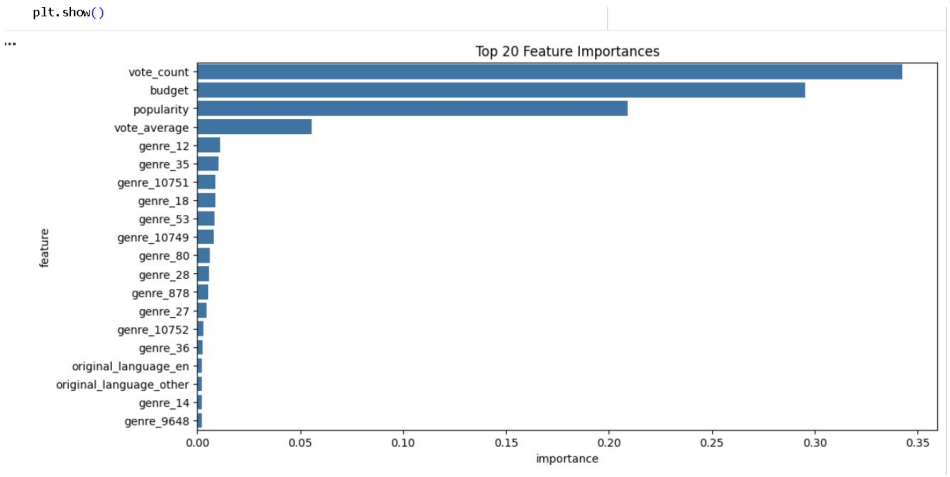


Hình 6: Biểu đồ phân tích phần dư (residual analysis)

Sai số dự đoán **tăng mạnh** và **phân tán rộng** khi giá trị doanh thu thực tế tăng. Hiện tượng này khẳng định rằng mô hình gặp khó khăn trong việc học và dự đoán chính xác doanh thu của các phim có giá trị cao, hay còn gọi là các phim bom tấn.

5.4 Phân tích Mức độ Quan trọng của Đặc trưng (Feature Importance)

Mô hình Random Forest cung cấp khả năng xác định mức độ quan trọng của từng đặc trưng đầu vào.



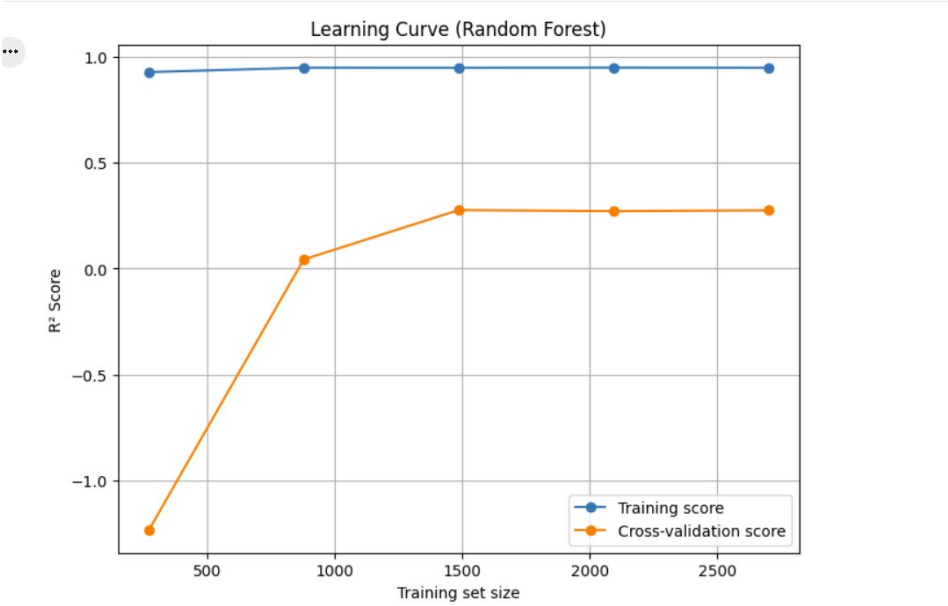
Hình 7: Mức độ quan trọng của các đặc trưng trong mô hình

Các đặc trưng có ảnh hưởng lớn nhất đến kết quả dự đoán doanh thu bao gồm:

- **Ngân sách (*budget*)**
- **Mức độ phổ biến (*popularity*)**
- **Số lượng phiếu bầu (*vote_ count*)**
- **Điểm trung bình (*vote_ average*)**

5.5 Phân tích Đường cong Học tập (Learning Curve)

Đường cong học tập cung cấp cái nhìn sâu sắc về khả năng tổng quát hóa của mô hình theo kích thước tập huấn luyện.



Hình 8: Đường cong Học tập (Learning Curve) của mô hình Random Forest

- **Overfitting Nghiêm trọng: Training score** (đường trên) luôn duy trì ở mức rất cao (xấp xỉ $0.95 - 1.0$), cho thấy mô hình đã học thuộc lòng tập huấn luyện. Ngược lại, **Cross-validation score** (đường dưới) ổn định quanh mức $R^2 \approx 0.15 - 0.20$. **Khoảng cách lớn** giữa hai đường này là bằng chứng rõ ràng nhất của **overfit mạnh**.
- **Giới hạn của Dữ liệu:** Đường cross-validation gần như **không tăng thêm** khi số lượng mẫu vượt quá 1500, cho thấy việc bổ sung thêm dữ liệu huấn luyện không giúp cải thiện đáng kể hiệu suất tổng quát hóa của mô hình.
- **Nguyên nhân:** Vấn đề này bắt nguồn từ đặc tính **phân phối lệch phải mạnh** của biến doanh thu (*revenue*) và sự xuất hiện của nhiều **outlier**, khiến mô hình Random Forest khó học được các quy luật tổng quát.

6 Kết luận và Định hướng Phát triển

6.1 Các Kết quả Đạt được

Dự án đã triển khai thành công mô hình **Random Forest** để dự báo doanh thu ngành điện ảnh và đạt được những thành tựu quan trọng sau:

- Mô hình đã thể hiện khả năng dự đoán rất tốt đối với các bộ phim thuộc phân khúc **doanh thu thấp và trung bình**, khẳng định Random Forest đã nắm bắt hiệu quả các quy luật cơ bản chi phối hoạt động thương mại thông thường của ngành.
- Phân tích mức độ quan trọng của đặc trưng đã chứng minh các yếu tố như **Ngân sách (Budget)**, **Mức độ phổ biến (Popularity)**, và **Số lượng phiếu bầu (Vote Count)** là những yếu tố cốt lõi và trực tiếp nhất tác động đến doanh thu của một bộ phim.

6.2 Thách thức của Bài toán Thực tế và Định hướng Cải tiến

Mặc dù mô hình đạt được hiệu suất tốt trên phân khúc dữ liệu lớn nhất, các phân tích chuyên sâu đã chỉ ra rằng mô hình cần được phát triển thêm để xử lý tốt hơn tính chất phức tạp cố hữu của bài toán thực tế:

- **Tính Ngoại lai (Outlier) của Thị trường Bom tấn:** Thị trường điện ảnh đặc trưng bởi sự xuất hiện của các sự kiện hiếm nhưng có tác động lớn (**Blockbuster**). Các phân tích cho thấy mô hình hiện tại chưa thể mô hình hóa chính xác các trường hợp có doanh thu cực cao này, đòi hỏi một phương pháp có khả năng học hỏi tốt hơn từ các giá trị hiếm.
- **Cân bằng giữa Độ chính xác và Khả năng Tổng quát hóa:** Hiện tại, mô hình cần được cải thiện khả năng tổng quát hóa trên tập kiểm tra. Vấn đề này xuất phát từ tính chất **phân phối lệch mạnh** và sự biến động lớn của biến doanh thu, đặt ra nhu cầu về một phương pháp học tập tổ hợp tốt hơn.

Để giải quyết những thách thức này và mở rộng khả năng dự đoán trên toàn bộ dải doanh thu, nhóm đề xuất định hướng phát triển tập trung vào các mô hình tiên tiến:

- **Chuyển đổi sang Mô hình Tổ hợp Nâng cao:** Bước tiếp theo là triển khai và đánh giá các thuật toán thuộc nhóm **Boosting** mạnh mẽ như **Gradient Boosting, XGBoost, hoặc LightGBM**. Các mô hình này có cơ chế hoạt động giúp tập trung xử lý các lỗi dự đoán lớn, từ đó có khả năng mô hình hóa tốt hơn tính phân phối lệch của biến doanh thu và cải thiện đáng kể khả năng tổng quát hóa của dự án.

7 Phụ lục

7.1 Mã nguồn và Bộ dữ liệu

- **Mã nguồn (GitHub):** Mã nguồn hoàn chỉnh của dự án (bao gồm các bước tiền xử lý, huấn luyện mô hình và phân tích) được lưu trữ công khai tại: https://github.com
- **Bộ dữ liệu:** Dữ liệu về metadata phim được sử dụng trong dự án (TMDB Movie Metadata) được lấy từ Kaggle: https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata

7.2 Tài liệu Tham khảo Kỹ thuật

- **Thuật toán Random Forest:** Tài liệu chính thức về lớp RandomForestRegressor trong thư viện Scikit-learn (Mô hình hồi quy Random Forest). https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- **Kỹ thuật Học tập Tổ hợp (Ensemble Learning):** Nguồn tham khảo về lý thuyết Bagging và Boosting (được đề xuất cải tiến). https://scikit-learn.org/stable/modules/ensemble.html