Predicting the Next Big Blockbuster: An Exercise on Machine Learning

Fuentes, Pecundo, Villegas

INTRODUCTION

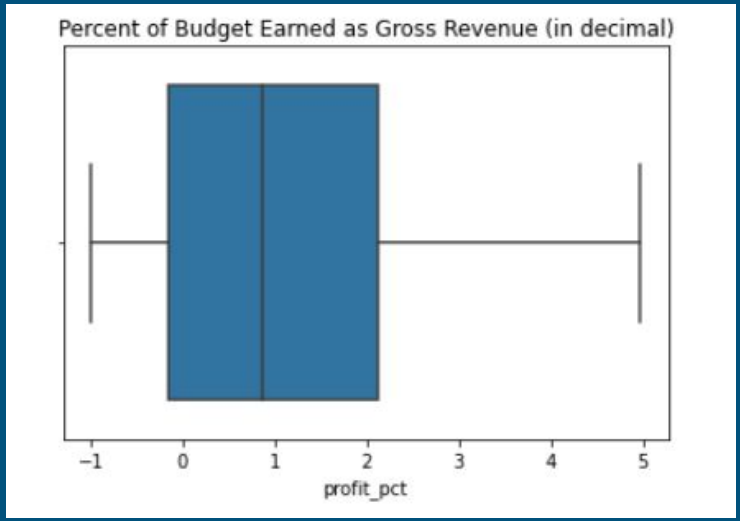
BACKGROUND

- The film industry generates a multitude of jobs; a major contributor to the economy
- Earning the title of “blockbuster” can be attributed to:
  - Cast lineup
  - Genre
  - Release Date
  - Budget
- Previous research: budget is positively correlated with box office revenue
  - But, is this always the case?

SIGNIFICANCE OF STUDY

- Figure shows that movies earn as much as about 200% of their original budget or as low as 0% of their original budget

- Therefore, the film industry can be unpredictable



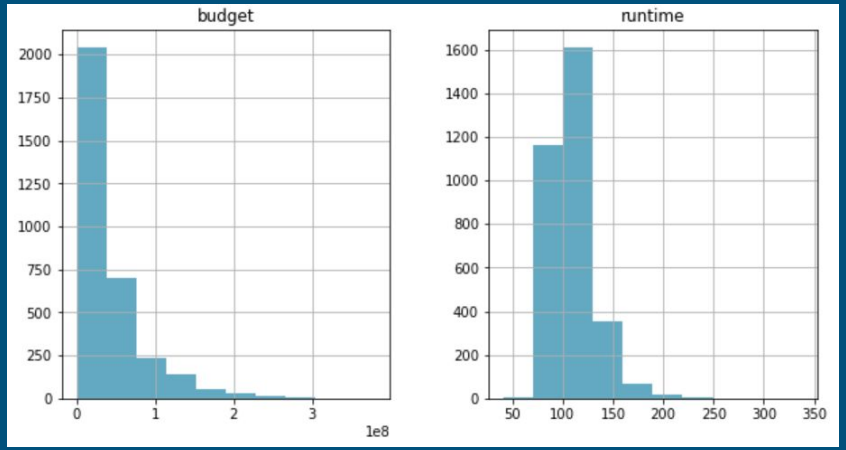
*profit_pct = (revenue - budget) / (budget)

Note: Some outliers removed from graph for visualization purposes

SIGNIFICANCE OF STUDY

- Careful planning and making sound decisions is a must for movie companies

- Creating a model that can predict success for movies can aid in making smarter decisions



OBJECTIVES

- Can we predict whether a movie is “blockbuster” (successful) before showing?
- What are the factors that contribute to the success of a movie?

TMDB 5000 DATABASE

- Sourced from Kaggle
- Two csv files:
  - Tmdb_5000_credits.csv
    - 4 columns
  - Tmdb_5000_movies.csv
    - 20 columns

FEATURES

Popularity

Popularity Score, Vote Count, Vote Average

Monetary

Budget, Revenue

Film Specific

Title, Genre, Keywords, Language, Runtime

Cast & Crew

Cast, Crew, Production Company

METHODOLOGY: PREPROCESSING

RECORDS WITH ATYPICAL VALUES WERE CLEANED

	budget	id	popularity	revenue	runtime	vote_average
count	4.803000e+03	4803.000000	4803.000000	4.803000e+03	4801.000000	4803.000000
mean	2.904504e+07	57165.484281	21.492301	8.226064e+07	106.875859	6.092172
std	4.072239e+07	88694.614033	31.816650	1.628571e+08	22.611935	1.194612
min	0.000000e+00	5.000000	0.000000	0.000000e+00	0.000000	0.000000
25%	7.900000e+05	9014.500000	4.668070	0.000000e+00	94.000000	5.600000
50%	1.500000e+07	14629.000000	12.921594	1.917000e+07	103.000000	6.200000
75%	4.000000e+07	58610.500000	28.313505	9.291719e+07	118.000000	6.800000
max	3.800000e+08	459488.000000	875.581305	2.787965e+09	338.000000	10.000000

After cleaning

	budget	id	popularity	revenue	runtime	vote_average
count	3.228000e+03	3228.000000	3228.000000	3.228000e+03	3228.000000	3228.000000
mean	4.066642e+07	44778.817844	29.042156	1.212803e+08	110.724907	6.309665
std	4.439840e+07	74620.916870	36.168131	1.863197e+08	20.968920	0.873846
min	1.000000e+00	5.000000	0.019984	5.000000e+00	41.000000	0.000000
25%	1.050000e+07	4956.250000	10.468206	1.700000e+07	96.000000	5.800000
50%	2.500000e+07	11446.500000	20.412963	5.519150e+07	107.000000	6.300000
75%	5.500000e+07	45269.750000	37.340747	1.463434e+08	121.000000	6.900000
max	3.800000e+08	417859.000000	875.581305	2.787965e+09	338.000000	8.500000

- “budget” field which contained value of 0 were removed
- “revenue” field which contained value of 0 were removed
- Removed movies on the list were also not “released”
- 4803 observations -> 3228 observations (33% reduction of original dataset)

GENRES WERE ENCODED AS FEATURES

	genres
0	[{"id": 28, "name": "Action"}, {"id": 12, "nam...
1	[{"id": 12, "name": "Adventure"}, {"id": 14, "...
2	[{"id": 28, "name": "Action"}, {"id": 12, "nam...
3	[{"id": 28, "name": "Action"}, {"id": 80, "nam...
4	[{"id": 28, "name": "Action"}, {"id": 12, "nam...

movie_id	genre_Action	genre_Adventure	genre_Comedy	genre_Crime
5	0	0	1	1
11	1	1	0	0
12	0	0	0	0
13	0	0	1	0
14	0	0	0	0

- Each movie can be classified into multiple genres.
- “Genre” field is a dictionary column that was unpacked so that the values can be used as categorical features for the model.

SOME FEATURES WERE ENGINEERED

- **Year** feature was extracted from release date
- **Genres** column was unpacked as categorical features
  - Each unique genre was considered a categorical column; movies can have multiple genres
  - Genres with smaller count of observations were grouped into “others” (from 22 genres to 11 genres)

- **“success”** column was created for the purpose of being the target variable
  - 1, if the revenue of the movie was at least 100% more than the budget
  - 0, if otherwise

OTHER FEATURES WERE DISREGARDED

- Features were limited to those that are either:
  - Within the control of the producer or studio
  - Factors that are known before a movie is shot
  - Possible to pre-process within the time constraints of the exercise

- Example of features that do not meet the criteria are:
  - Popularity - You only know a movie's popularity once it is out
  - Cast - Contains multiple columns (in the thousands) which will need careful preparation

FINAL FEATURES USED

Input Features:

- Budget
- Year
- Runtime
- Genre encoded as (Action, Adventure, Comedy, Crime, Drama, Family, Horror, Romance, Sci-fi, Thriller, and Others)

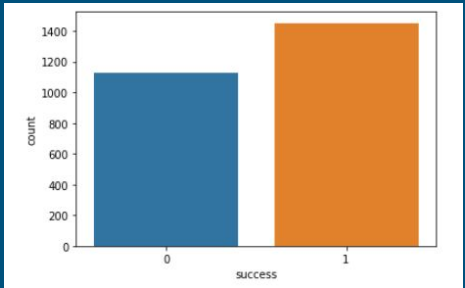
Output (Target) Features:

- “success”

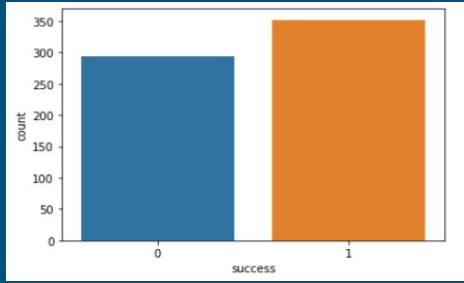
METHODOLOGY: MODEL BUILDING

MODEL TRAINING SPECIFICATIONS

- For training, data was split into training and test sets
  - Model was trained on the training set (80% of the observations)
  - Model accuracy was measured on the test set (20% of the observations)



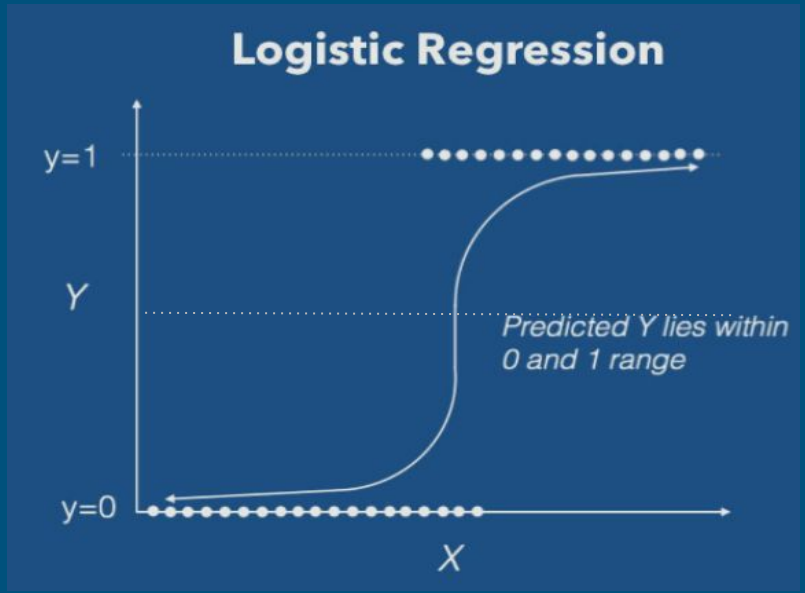
Training Set Class Distribution



Test Set Class Distribution

- 3 Models were built for the experiment
  - Model 1: Logistic Regression
  - Model 2: Naive Bayes
  - Model 3: Decision Tree (max depth = 5)

MODEL 1: LOGISTIC REGRESSION



$$y = \frac{1}{1 + e^{-(mX + b)}}$$

- Algorithm appropriate for binary classification
  - Predicts y as probability between (0,1)
  - If y > 0.5, assign as 1, 0 otherwise.
  - Final outcome is 1 or 0

Sources: [1] Practical Statistics for Data Scientists, [2] Class Lecture Materials

MODEL 2: NAIVE BAYES

Objective:

What is P (Class 1 | Feature 1, Feature 2, …)?

What is P (Class 0 | Feature 1, Feature 2, ...)?

Observation belongs to class that has the greatest probability, given its features

What we normally know:

P (Feature 1), P (Feature 2), … from the data (both observed or incoming)

Bayes Theorem:

P (Class 1 | Feature 1, Feature 2,…) = (P (Feature 1, Feature 2,.. | Class 1) * P (Class 1))/ P (Feature 1, Feature 2, ..)

Naive Bayes Assumption:

P (Feature 1, Feature 2, .. | Class 1) is unknown for all cases unless we assume features are independent.

P (Feature 1, Feature 2, .. | Class 1) = P (Feature 1 | Class 1) * P (Feature 2 | Class 1) * … **(Naive Assumption)**

Source: Class Lecture Materials

MODEL 2: NAIVE BAYES

Under the Naive assumption:

P (Class 1 | Feature 1, Feature 2,…) =

(P (Feature 1 | Class 1) * P (Feature 2 | Class 1) * …) * P (Class 1)) / P (Feature 1, Feature 2, ..)

and

P (Class 2 | Feature 1, Feature 2,…) =

(P (Feature 1 | Class 0) * P (Feature 2 | Class 0) * …) * P (Class 0)) / P (Feature 1, Feature 2, ..)

Since P (Feature 1, Feature 2, ..) will be consistent for both expressions, you only need to compute:

P (Feature 1 | Class 0) * P (Feature 2 | Class 0) * …) * P (Class 0)

And

P (Feature 1 | Class 1) * P (Feature 2 | Class 1) * …) * P (Class 1)

Sources: [1] Practical Statistics for Data Scientists, [2] Class Lecture Materials

Algorithm:

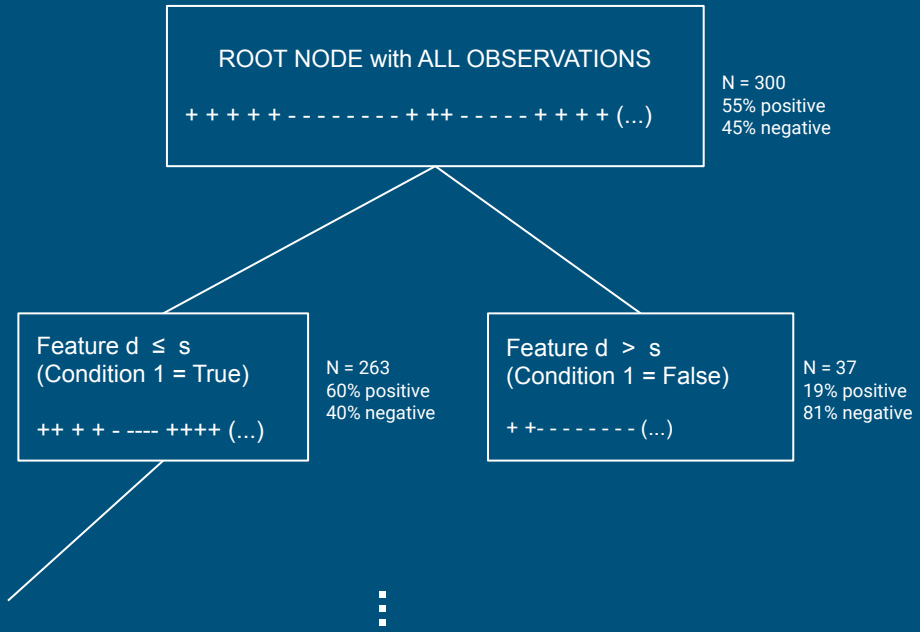
Step 1: For every class, identify the probability of a class A having certain feature X. Repeat calculation of probabilities for all features in the data.

Step 2: Multiply probabilities the identified probabilities per class group (e.g., P (X_1 | A) * P (X_2 | A) …) with the probability of the observation belonging to that same class (e.g., P(A)).

Step 3: Repeat step 1 and 2 for each target class

Step 4: Future observations get assigned to the class that yields that largest probability computed from Step 3

MODEL 3: DECISION TREE



Algorithm:

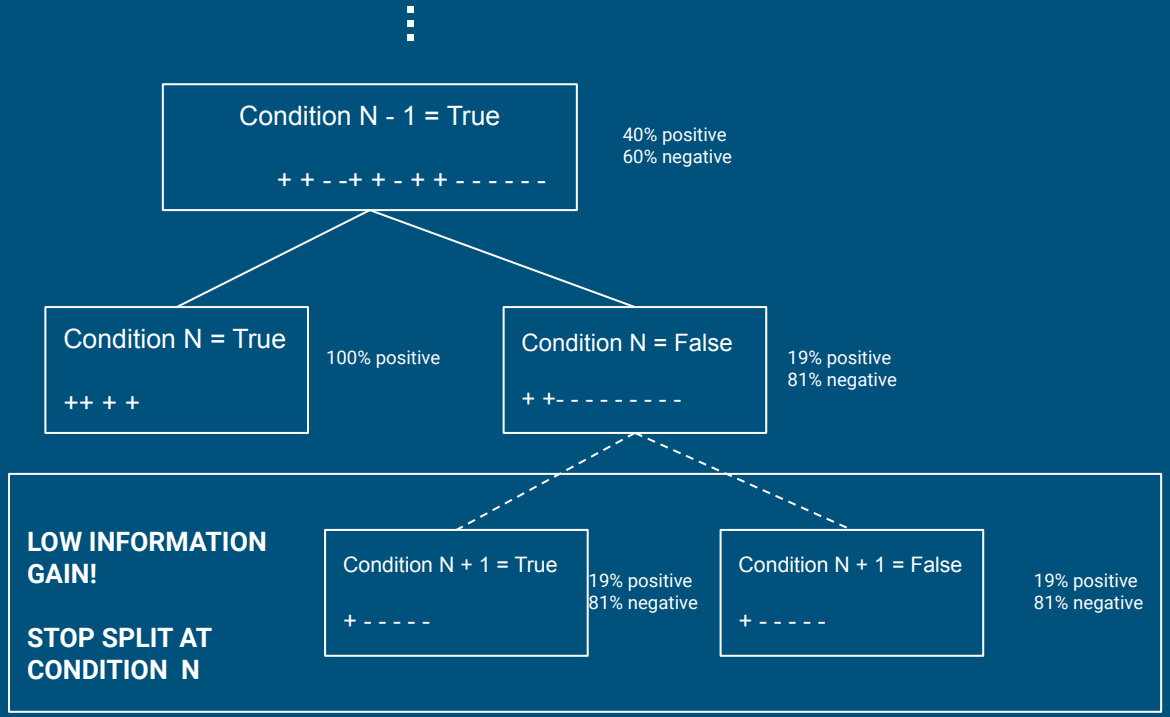
Step 1: Start at root node (all observations) and identify a single feature *d* among all features D to partition data

Step 2: Identify split value *s* at feature *d* in which to partition data; optimal split based on selected criteria

- Sample of criteria is gini index (i.e., an indicator of how will the resulting split distinguish between well between classes)

Source: [1] Introduction to Statistical Learning with R 7th ed., [2] Practical Statistics for Data Scientists

MODEL 3: DECISION TREE



Sources: Introduction to Statistical Learning 7th ed., Practical Statistics for Data Scientists

Inspiration of illustration: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

Algorithm:

Step 3: Repeat Step 1 (Feature identification) and 2 (Split value identification at feature) until stopping criteria is met;

- Some common stopping criteria are:
  - (1) max tree depth,
  - (2) gini index
  - (3) minimum observations in a node

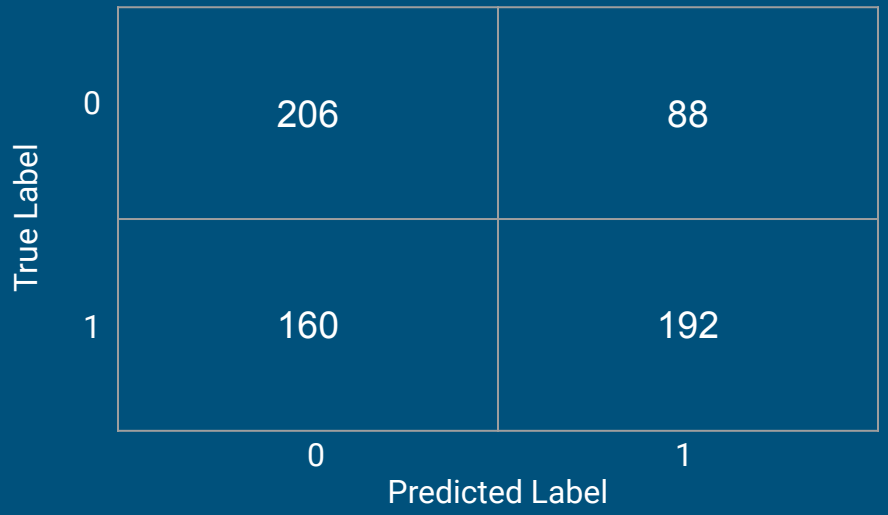
Step 4: Predict future observations based on conditions of the final tree

MODEL RESULTS

Model Type	Precision	Recall	Accuracy
Logistic Regression	0.54	1.00	0.54
Naive Bayes	0.55	0.84	0.54
Decision Tree	0.69	0.55	0.62

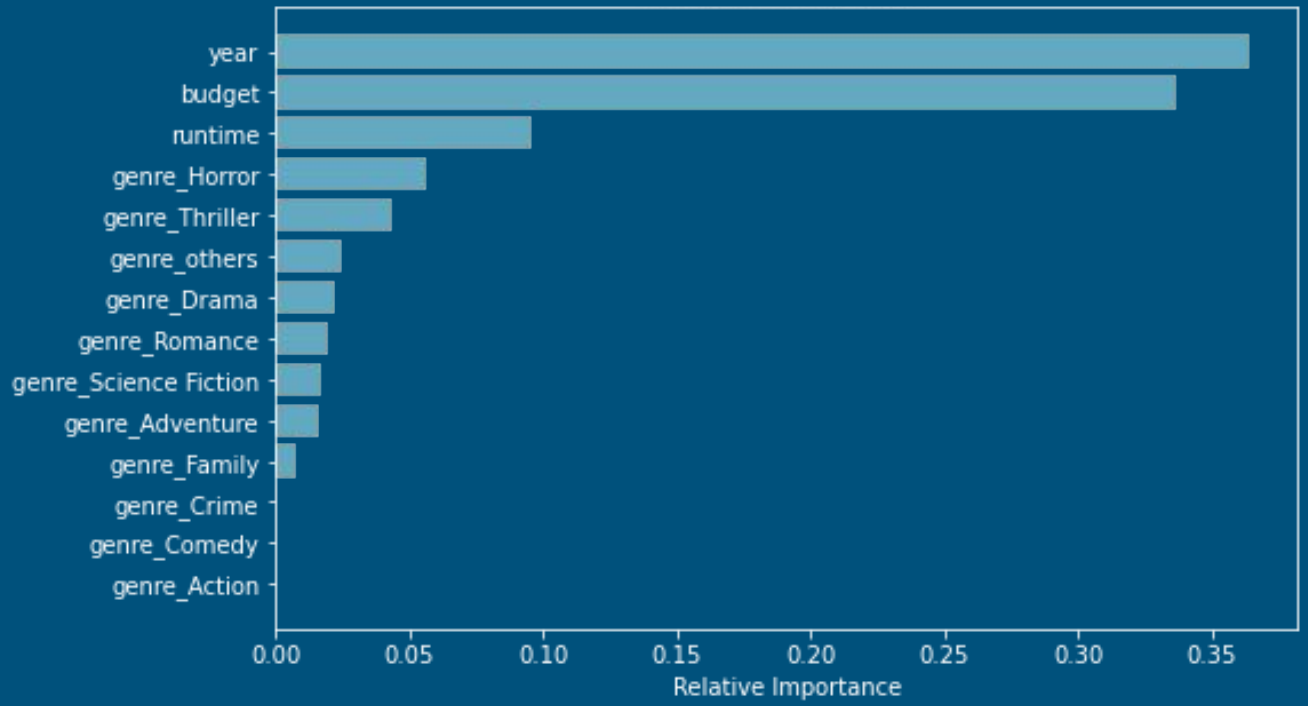
The Decision Tree model was used as the final model given the relatively higher accuracy and good balance between precision and recall.

FINAL MODEL - CONFUSION MATRIX

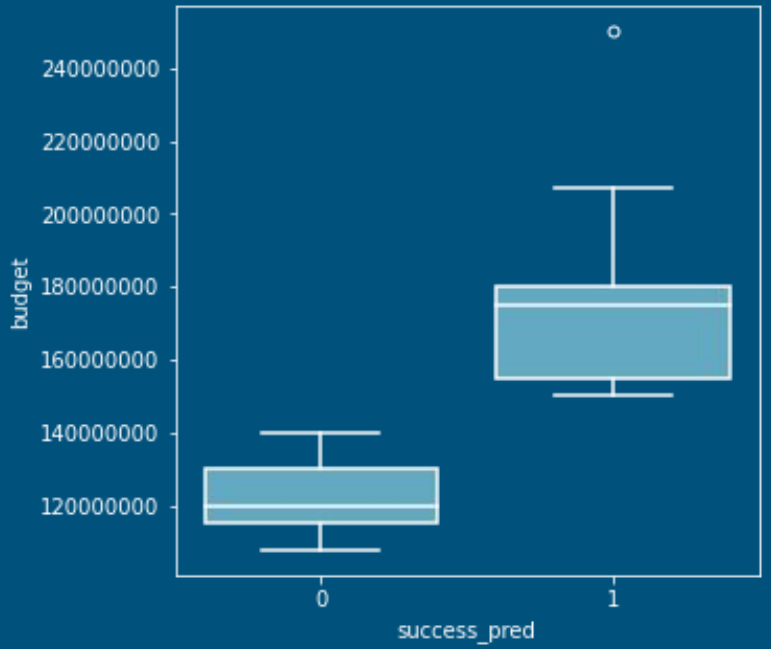


INSIGHTS

Important Features

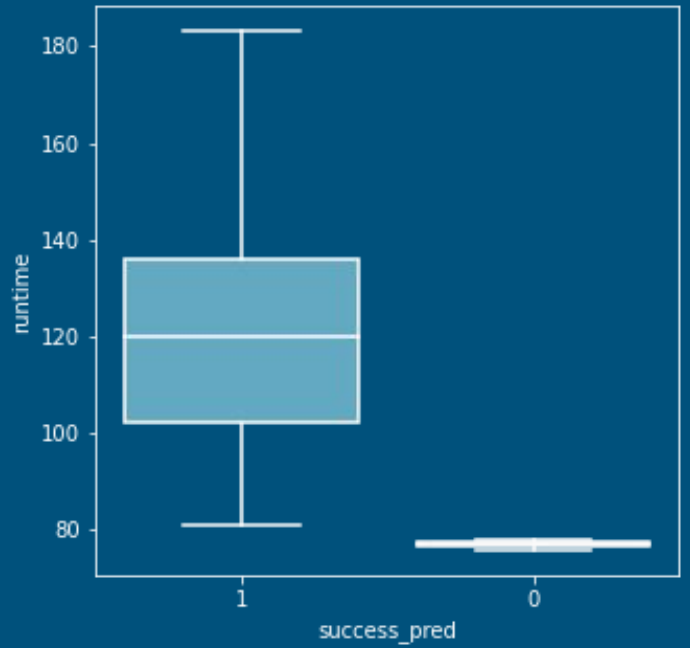


Invest big in Drama Films



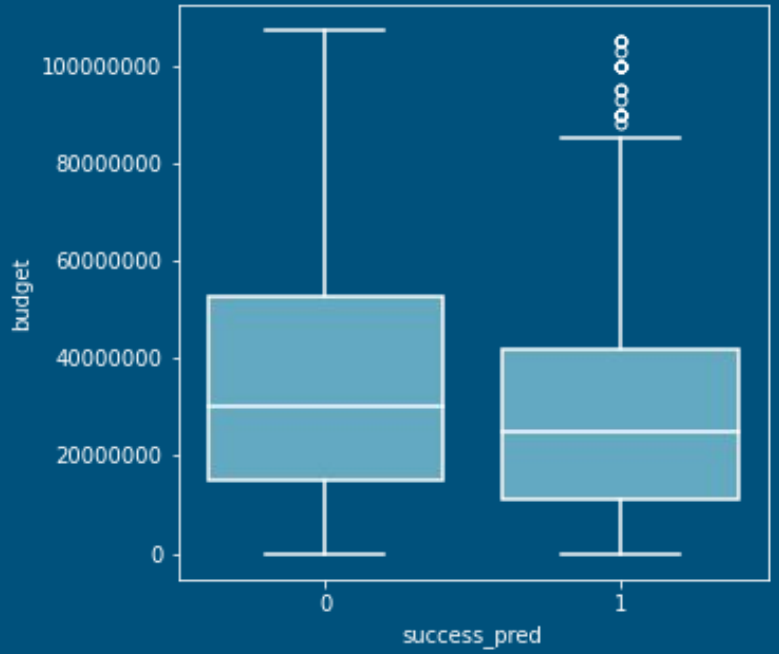
End node where budget is greater than $145M and Genre is drama

Non-Drama films, keep it at least 79 minutes long



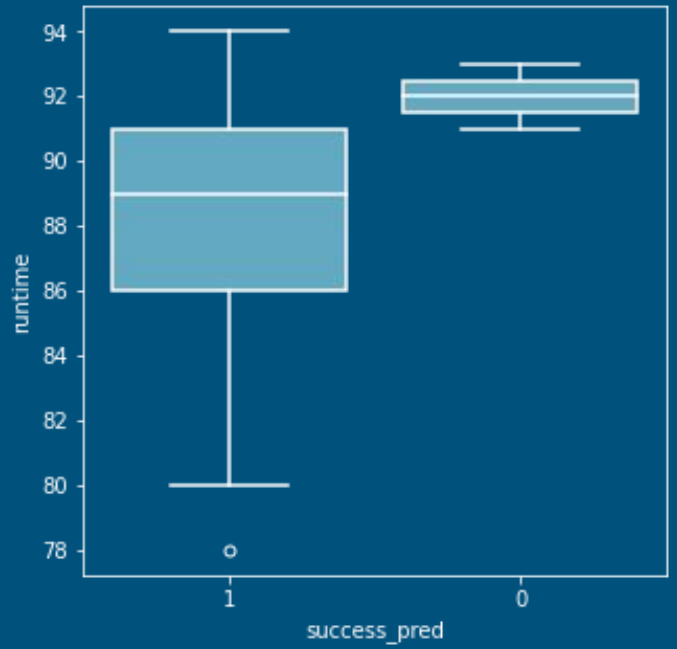
End node where budget is greater than $107.5M and genre is non-drama

Non-horror films with less budget can still be successful



Decision node where budget is greater than $107.5M and genre is non-horror.

Keep Non-Romance horrors short



End node where genre is horror and non-romance and budget is greater than $107.5M

CONCLUSION

SUMMARY

- We can predict if a movie will be successful or not using limited variables that are known before the release.
- Some of the important features from the Trained Decision Tree model are year, budget, and runtime.
- From the results of the splitting done by the Decision Tree Model, we were able to identify the characteristics of the movie that are most likely to be predicted as successful.

Q&A