Domestic lifetime gross prediction for PG-13 movies

Abstract

The goal for this project is to use regression models to predict the domestic lifetime gross in order to help movie investors make the investment decision. I started with a base linear model, then did some feature engineering, adding polynomial terms, did regularization by LassoCV, and tried the Random Forest model(RF) to improve my R2 by 10.65% from base linear model (0.7328) to (RF 0.8393).

Design/Background Story

Our client is a movie investing firm who is going to invest in several PG13 movies in the coming year. They want us to help them predict the domestic lifetime gross for the movies that they are interested in, they want to know which movies can bring them the highest return.

Data

Web Scraping from ‘Box office Mojo’ website - 1000 PG13 movies.

Target: 'lifetime_gross'

Features: 'link_stub', 'title', 'rank',, 'rank_overall', 'year', 'budget', 'domestic_distributor', 'running_time', 'earliest_release_date', 'genres', 'MPAA'

Algorithms

Base Model

- Getting simple base model by relevance numerical features

Feature Engineering

- Converting categorical features to binary dummy variables
- Adding complexity on base model by adding categorical dummies
- Select the best model among feature engineering models

Models - 5 fold cross validation for 4 candidate models

- Model 1: Simple linear model
- Model 2: Polynomial linear model(degree 2)
- Model 3: LassoCV + Polynomial(degree 2) linear model
- Model 4: Random Forest

Model Evaluation and Selection (by comparing validation score & RMSE & MAE)

- Model 1: val score - 0.748, RMSE - 41,538,596, MAE - 25,389,385
- Model 2: val score - 0.764, RMSE - 39,440,843, MAE - 24,326,173
- Model 3: val score - 0.764, RMSE - 39,049,303, MAE - 24,228,014
- Model 4: val score - **0.770**, RMSE - **39,064,999**, MAE - **23,901,101**

Final model: refit RF model on entire dataset(we do want to keep the important information in outliers rows- RF model will deal with missing values and outliers)

- *val score - 0.8393;* ***test score - 0.8393***

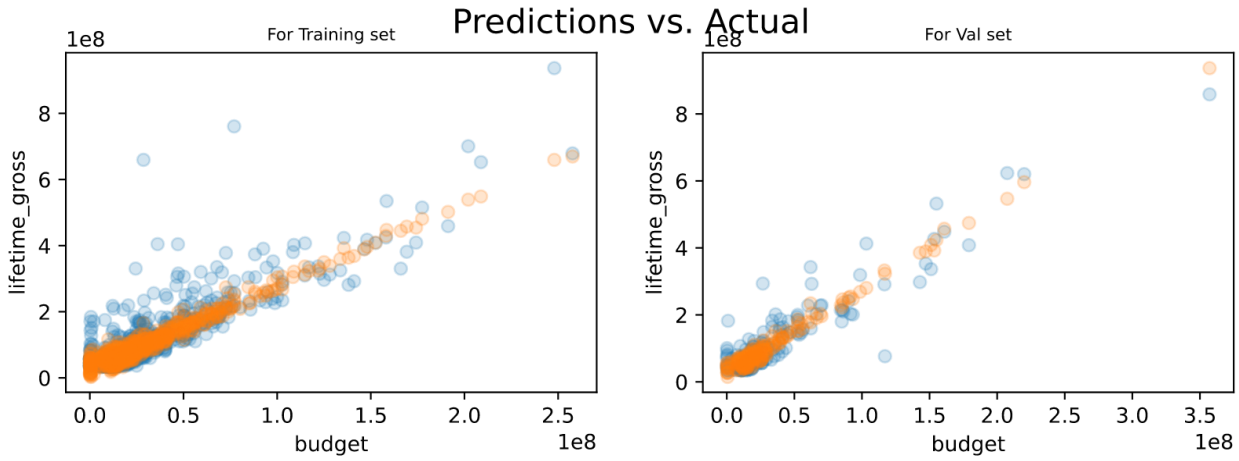
Tools

- Beautiful soup Python package for web scraping
- Numpy and Pandas for data manipulation
- Scikit-learn for modeling
- Matplotlib and Seaborn for plotting

Communication

Below orange dots are y_predicted, blue dots are y_true for my final model.
I used my model to predict random movies from the test set. Here's a couple results.

- 'The River Wild': Prediction: 48,979,015; Actual: 46,816,343
- 'Norbit': Prediction: 109,733,112; Actual: 95,673,607



Future Work

1.Getting more data - directors, writers, etc... from the website to see if it can improve my model.
2. For linear model, as our base model is with higher variance along x axis, I may try Weighted Least Square(WLS) for my linear model to see if it can fit my data better.