Box office revenue predictor

Eilert Skram, Torbjørn Moen, 18.11.2022

DESCRIBE THE PROBLEM

SCOPE

- *“Business objective”: Describe the project’s goal and “business impact.”*
- *How will the solution produced in the project be used? What similar solutions already exist / how is the problem solved today? How would you do the task manually without using machine learning?*
- *How will the performance be measured via “business metrics”?*
- *If your machine learning model will be part of a more extensive “pipeline” or system, describe the system’s components. Consider how changes in one part of the system may impact other parts.*
- *Describe the stakeholders of the project*
- *Describe a tentative timeline for the project. Include milestones.*
- *Define what resources, for example, computational resources and personnel, will be required to complete the project.*

The “main " goal of the project is to create a solution that will be able to predict the world wide revenue of a potential up and coming movie. This solution can be used by both production companies and investors to help gain insight if a project has a good trajectory or i.e how much should be put in compared to expected output. It can also help with guidance. I.e what production studios to go for if you want to create say an action movie, as some will have a track record of higher earnings in certain genres.

Solving it manually would require collecting data of similar movies(similar in the instance of genre, cast, budget etc) and making a prediction how the potential movie would perform, based on your estimation from the comparison of previous performances. The solution will streamline this process and allow for bigger datasets to be used, hopefully yielding in a better prediction of revenue performance

The solution would be part of a website that uses user input to make predictions on the parameters of the potential movie. This will require data validation and processing, before the prediction made by the model is being returned as output. The model will also have to make recurring pulls from the database to train the model on recent releases, since trends etc keep developing.

The stakeholders of the project can be split into the company developing model and the consumer(production/movie investors)

Timeline: basic model -> processing pipeline -> model improvement -> first deploy -> **LOOP**: feedback -> (optional) new data pulls -> exploration of different angles and improvements -> (optional) changes -> deploy -> monitoring -> feedback

The project obviously will need computational power to process the data and to train the model, this can either be done locally or via cloud(I.e Kaggle). The model will be deployed on a website, so a server and framework is required. We will use Flask for the framework. If the project was bigger scale, personnel like feature engineers etc could be included, to get a better model, but at a higher cost.

METRICS

The project would be considered a success if the model manages to separate between the parameters that decides what yields a “good” movie or not. The number in itself might just give an indication, but the model can contribute to showing if the choice of cast, production company, genre etc will yield a successful movie.

The model should be easy to interact with, the format of data asked of the user needs to be easily understood so it's simple to use. It should also not have too high latency, which will require the model to allow for a larger error margin in trade for speed. We are using root-mean-square deviation(RMSE) to calculate the performance of the model.

DATA

The dataset consists of 3000 movies collected from TMDB(The Movie Database). As new movies get added, we can make further pull requests through the TMDB API to keep the data consistent and for the model to continue learning. It will also allow us to capture new trends.

The data contains columns ranging from “useful” columns, i.e revenue and budget, to more “site functional” columns, i.e posterpath etc. Most of the data is stored in J-son format, so most of the work will be manipulating this data and exploring what makes certain movies earn more.

First iteration of the model we will attempt to encode the categorical values to numeric values. Collection, website, keywords and tagline simplified down to if in collection or not, a boolean represented by 1 or 0. We are doing similar actions with genre, spoken and original language, production company and country. Here we are either using one-hot encoding or numeric representation of a bool, to indicate if the movie belongs in a genre or i.e created by one of the biggest movie studios.

The data is fed through a pipeline that does the aforementioned encoding, before normalizing and using mean values to fill null data.

A lot of the information is lost because of the broad assumptions and big cuts. At later iterations, it will be vital to explore more of the data stored in the j-son lists. Find what groupings etc that will lead to higher revenue. A lot of the assumptions are easy to word, but hard to extract and get down to code. I.e certain famous actors, will lead to a higher revenue, because it's a “bigger” movie.

The data might include western bias, which means it could perform worse with eastern market due to lack of representation in the dataset.

MODELING

The model needs to predict revenue for a potential movie, this implies using some form of regressor. The models explored in the first iteration were regressors via XGBoost, GradientBoosting, RandomForest, DecisionTree and LinearRegression. RandomForest seemed to yield lowest mean and standard deviation.

We tried tweaking the hyperparameters via randomized search, first with 3 folds, 50 candidates. But even at 3 folds and 100 candidates, the best estimator performed significantly worse.

All the models will naturally perform poorly during the first iteration, as a lot of vital information is lost. This can mean that when further data exploration has been done, most of the models will be explored again at a later stage.

DEPLOYMENT

The model will be exported and deployed via JobLib and SKlearn. Using Flask, the model will be deployed on a website. The website will ask its user for the params for the potential movie, the model will then give the expected revenue of the potential movie.

As previously mentioned, the first iteration of the model will perform poorly. This means the model will quickly need improvements and overhauls, which will lead to various degrees of changes made to every section of the system to adapt to the requirements of the new model.

REFERENCES

https://en.wikipedia.org/wiki/Major_film_studios
https://towardsdatascience.com/adjusting-prices-for-inflation-in-pandas-daaaa782cd89
https://en.wikipedia.org/wiki/Hindi_cinema