Machine Learning Engineer Nanodegree

Capstone Proposal

Movie Box Office Revenue Prediction with Gradient Boosting Models

Daniel Tejeda
September 17, 2019

Proposal

Domain Background

In a world where movies made an estimated 41.7 billion in 2018 and expected to go over 50 billion by 2020, the film industry is more popular than ever. But which movies make the most money at the box office? How much does a director or the budget matter? In this capstone project, I will build a model to answer that question, working with metadata on over 7,000 past films from The Movie Database published as part of TMDB Box Office Prediction Kaggle competition.

Forecasting the financial performance of a movie before its release has been the motivation of a number of research papers. One of the most cited ones is from Ramesh Sharda (Predicting box-office success of motion pictures with neural networks), in which he used neural networks to create a multinomial classification model. I take a different regression approach, stacking boosting models mainly because they have been proven to be effective in Kaggle competitions.

Problem Statement

Given its economic potential, the movie box office revenue prediction is a problem that is being actively researched by data scientist and production houses.

The main purpose of this project is to build a model that predicts the revenue of a movie with only data that would have been available before the movie is released and to determine with exploratory data analysis, which features of the movie are more relevant for its monetary success. Such a model could be of high value for the decision making process of a movie production at different stages or even before the movie gets funding, saving production companies millions of dollars.

Datasets and Inputs

This dataset has been collected from TMDB and published on their Kaggle competition (https://www.kaggle.com/c/tmdb-box-office-prediction/data). The movie details, credits and keywords have been collected from the TMDB Open API.

The dataset contains a Train set with 3000 records and Test set with 4398 records. The Test files do not have the target variable on them because of the nature of Kaggle competitions, so we will discard this and will split the Train set 80/10/10 for training, validation and test.

Attributes

- ID - Integer unique id of each movie
- Belongs_to_collection - Contains the TMDB Id, Name, Movie Poster and Backdrop URL of a movie in JSON format. You can see the Poster and Backdrop Image like this: https://image.tmdb.org/t/p/original/.
  Example: https://image.tmdb.org/t/p/original//iEhb00TGPucF0b4joM1ieyY026U.jpg
- Budget:Budget of a movie in dollars. 0 values mean unknown.
- Genres : Contains all the Genres Name & TMDB Id in JSON Format
- Homepage - Contains the official homepage URL of a movie.
  Example: http://sonyclassics.com/whiplash/ , this is the homepage of Whiplash movie.
- Imdb_id - IMDB id of a movie (string). You can visit the IMDB Page like this: https://www.imdb.com/title/
- Original_language - Two digit code of the original language, in which the movie was made. Like: en = English, fr = french.
- Original_title - The original title of a movie. Title & Original title may differ, if the original title is not in English.
- Overview - Brief description of the movie.
- Popularity - Popularity of the movie in float.
- Poster_path - Poster path of a movie. You can see the full image like this: https://image.tmdb.org/t/p/original/
- Production_companies - All production company name and TMDB id in JSON format of a movie.
- Production_countries - Two digit code and full name of the production company in JSON format.

- Release_date - Release date of a movie in mm/dd/yy format.
- Runtime - Total runtime of a movie in minutes (Integer).
- Spoken_languages - Two digit code and full name of the spoken language.
- Status - Is the movie released or rumored?
- Tagline - Tagline of a movie
- Title - English title of a movie
- Keywords - TMDB Id and name of all the keywords in JSON format.
- Cast - All cast TMDB id, name, character name, gender (1 = Female, 2 = Male) in JSON format
- Crew - Name, TMDB id, profile path of various kind of crew members job like Director, Writer, Art, Sound etc.
- Revenue - Total revenue earned by a movie in dollars.

Solution Statement

The solution is a regression model that predicts the movie revenue given its metadata. I will first use exploratory data analysis to get a better understanding of the data and its relationships with the target variable revenue. The dataset contains a range of features such as cast, crew, genre, production company, etc, that will require feature engineering prior modelling.

I am going to use three popular gradient boosting models: XGBoost, CATBoost and LightGBM. I decided on boosting algorithms because they've been proven to be useful in Kaggle competitions with limited training data, little training time and little expertise for parameter tuning. I will present the results for each of the three models and build a stacked model that will combine them to improve the overall score.

Benchmark Model

A K-nearest neighbors (KNN) regressor based on the three more relevant features (budget, popularity and runtime) will be used as Stage-1 benchmark. I expect all three boosting algorithms to beat the KNN.

Stage-2 benchmark will be the best of the three boosting algorithms, which I will then try to beat using the final stacked model.

Evaluation Metrics

Root-mean-squared-logarithmic-error (RMSLE) will be the main metric for evaluation of all models. This metric is similar to Root-mean-squared-error (RMSE), only calculated in logarithmic scale. To calculate it we use:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(y_i+1)-\log(\hat{y}_i+1))^2}$$

The main reason to use this metric instead of RMSE is because of the magnitude of the target variable revenue. Since this value can range up to the millions of dollars and we want to get the relative error without considering magnitude, RMSLE is more suited to this scenario.

Project Design

This project will be implemented in Python 3.7. Libraries involved will be numpy, pandas, matplotlib, seaborn, xgboost, lightgbm, catboost, scikit-learn.

The workflow for this project will be in the following order:

1. Exploratory data analysis
2. Data cleansing and Feature engineering
3. Train the KNN benchmark model based on budget, popularity and runtime
4. Stage-1: Boosting Models
    - Code and train XGBoost model
    - Code and train CATBoost model
    - Code and train LightGBM model
    - Hyperparameter tuning for the three models
    - Evaluate results against KNN and select new benchmark from the boosting models to be the new benchmark.
5. Stage 2: Stacked final model
    - Select stacking approach and regression algorithm for the final model
    - Train regression algorithm with the outputs of the base boosting models combined with the original features, according to the stacking approach
    - Hyperparameter tuning for the final model
    - Evaluate and report final results against the benchmark