Machine Learning Engineer Nanodegree – Capstone Proposal

TMDB Box Office Prediction

Srinivas C Reddy
7/11/2019

Domain Background

We live in movie obsessed world, where movies made an estimated $41.7 billion in 2018, if we include home entertainment revenue global film industry is worth $136 billion. (IBISWorld, 2018). The film industry is more popular than ever. But what movies make the most money at the box office? How much does a director matter? Or the budget? For some movies, it's "You had me at 'Hello.'" For others, the trailer falls short of expectations and you think "What we have here is a failure to communicate." (Kaggle Competiton, 2019)

Problem Statement

In this public competition hosted by Kaggle, I am presented with metadata on over 7,000 past films from The Movie Database to try and predict their overall worldwide box office revenue. Data points provided include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries. I can also collect other publicly available data to use in your model predictions, but in the spirit of this competition, use only data that would have been available before a movie's release.

The goal of this capstone project is to predict the overall worldwide box office revenue given data about movies.

Datasets and Inputs

In this dataset, I am provided with 7398 movies and a variety of metadata obtained from The Movie Database (TMDB). Movies are labeled with `id`. Data points include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries.
I am expected to predict the worldwide revenue for 4398 movies in the `test` file.
Note - many movies are remade over the years, therefore it may seem like multiple instance of a movie may appear in the data, however they are different and should be considered separate movies. In addition, some movies may share a title, but be entirely unrelated.
E.g. *The Karate Kid* (`id: 5266`) was released in 1986, while a clearly (or maybe just subjectively) inferior remake (`id: 1987`) was released in 2010. Also, while the *Frozen* (`id: 5295`) released by Disney in 2013 may be the household name, don't forget about the less-popular *Frozen* (`id: 139`) released three years earlier about skiers who are stranded on a chairlift.

This dataset has been collected from TMDB. The movie details, credits and keywords have been collected from the TMDB Open API. This competition uses the TMDB API but is not endorsed or certified by TMDB. Their API also provides access to data on many additional movies, actors and actresses, crew members, and TV shows.

Solution Statement

The solution will be worldwide box office revenue prediction, which is continuous. Because of nature of solution, I will be able to quantify in math or logical terms. I will evaluate predictions with the validation and test datasets to calculate error measurement Root-Mean-Squared-Logarithmic-Error (RMSLE), logs are taken to not overweight blockbuster movies. Goal of this ML exercise is to minimize the error term on the prediction. Here are list of techniques and algorithms I intend to try:

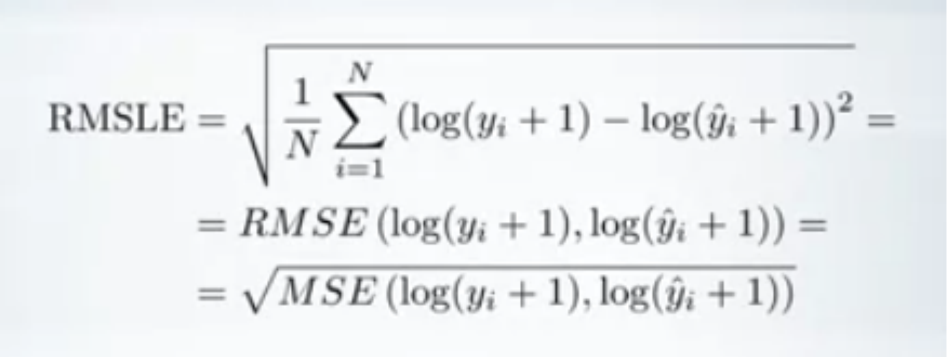
1. Data Exploration, feature engineering, dimensionality reductions
   a. Visualize distributions for each feature
   b. Transform and/or normalize, scale features as required
   c. Find outliers and determine if they need to be removed
   d. Encode the ordinal or nominal feature
   e. Use PCA to discover dimensions about features that can maximize variance.
2. Supervised model selection using grid-search/k-folds:
   a. DecisionTreeRegressor
   b. RandomForestRegressor
   c. AdaBoostRegressor
   d. XGBoostRegressor
   e. CatBoostRegressor
   f. MLPRegresssor

Benchmark Model

I will use DecisionTreeRegressor as my benchmark. Other models' performances will be compared to that of DecisionTreeRegressor, as all models are fitted with the same train and validation datasets.

Evaluation Metrics

The main evaluation metric used will be Root-Mean-Squared-Logarithmic-Error (RMSLE), which is given by:



$$\mathrm{RMSLE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\log(y_i+1) - \log(\hat{y}_i+1)\right)^2} =$$

$$= RMSE\left(\log(y_i+1), \log(\hat{y}_i+1)\right) =$$

$$= \sqrt{MSE\left(\log(y_i+1), \log(\hat{y}_i+1)\right)}$$

RMSLE is Root Mean Square Error calculated in logarithmic scale. The targets are usually non-negative, but if target is equal to 0, logarithm of 0 is not defined, that is the reason for adding constant 1 before taking logarithm. I will perform hyperparameter tuning based on how model performs in minimizing RMSLE.

Project Design

Here is the proposed project design I will use for the project:

1. Explore data – I will perform feature exploration, data profiling, looking at max, min, median and histograms, using pandas_profiling package
2. Clean data – I will look at each feature, its relevance to the target. Will look at any outliers and missing data and come up with an approach to manage them. For outliers, I plan to use transformations, and in extreme cases eliminate them. For missing data, simplest approach is to remove the observation, however, that could significantly reduce the dataset, will plan to use Imputer to fill the missing values based on analysis
3. Prepare data – I will use modules to split train and validate data, including k-fold validation techniques due to limited size of training data.
4. Feature Engineering, selection – based on analysis from exploration I will apply dimensionality reduction techniques like PCA, and feature importance packages, and based on results will plan on reducing dimensionality.
5. Model Selection – As this is a regression problem, I will experiment with different algorithms. For each algorithm, I will optimize hyperparameters with gird search, during each attempt I will look at how error terms are minimizing and how well if the models are fitting, and check for overfitting. I will use visualization to illustrate how I am making data driven decisions as I progress along with project. I will select an optimal final model by comparing performance of models Here is some models I am planning to consider:
    a. DecisionTreeRegressor
    b. RandomForestRegressor
    c. AdaBoostRegressor
    d. XGBoostRegressor
    e. CatBoostRegressor
    f. MLPRegresssor