# Machine Learning Engineer Nanodegree

# Capstone Proposal

## TMDB Box Office Prediction

### Can you predict a movie's worldwide box office revenue?

**Sougata Ghosh**

**March 5, 2019.**

## Domain Background

In a world where movies made an estimated $41.7 billion in 2018, the film industry is more popular than ever. It is at the forefront of creating artistic and cultural trends that impact billions of viewers worldwide. Films can bring to us stories that need to be told and help shape public opinion through their unique blend of entertainment and widespread reach. For the industry to continue to thrive and make meaningful contributions to society it is important for it also to be profitable and for films to make money. 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."

In this Udacity Machine Learning Engineer Nanodegree capstone project, which is based on an ongoing Kaggle competition, we are presented with metadata on over 7,000 past films from The Movie Database to try and predict their overall worldwide box office revenue.

## Problem Statement

Using the data from The Movie Database train and test datasets, our aim is to predict the worldwide revenues for films in the test data set. Data points in both datasets provide a variety of information on each movie. The models trained will use the information given above as well as new features created from them. Since the target variable (revenue) is continuous, this is essentially a regression task that can be addressed with one of several different machine learning algorithms we are going to use. The project will involve data cleaning, data exploration using visualizations and testing various regression algorithms to solve the problem.

## Datasets and Inputs

In this dataset, we are provided with two csv files containing names of 7398 movies and a variety of metadata obtained from The Movie Database (TMDB). Movies are labelled with id. Data points include cast, crew, plot keywords, overview, budget, posters, release dates, languages, production companies and countries, the genres each movie belongs to, its popularity and runtime, etc in both datasets. The train dataset has further information on the actual box office revenues for 3000 movies to train our machine learning algorithms on. The train set has 3000 records and 23 columns.

The task is to predict the worldwide revenues for 4398 movies in the test file. The test data has 22 columns, one less than the train dataset as the revenue information for the test set movies is not given.

This dataset has been collected from TMDB. The movie details, credits and keywords have been collected from the TMDB Open API as part of the Kaggle competition. The train and test data along with a sample submission file can be obtained here. They are also included in the capstone proposal github repository.

## Solution Statement

The solution will proceed by training several gradient boosting decision tree (GBDT) algorithms, namely XGBoost, LightGBM and CatBoost. As this article here discusses, such algorithms are currently the best techniques for building predictive models from structured data. The best decision tree packages can train on large datasets with sublinear time complexity, parallel over many cpus distributed amongst a cluster of computers and can extract most of the information from a dataset under a minute.

Ever since its introduction in 2014, XGBoost has been lauded as the holy grail of machine learning hackathons and competitions. From predicting ad click-through rates to classifying high energy physics events, XGBoost has proved its mettle in terms of performance – and speed. It is usually the first algorithm of choice in any Machine Learning competition and we will use its powerful features here. XGBoost introduces two techniques to improve performance. First, the idea of Weighted Quantile Sketch, which is an approximation algorithm for determining how to make splits in a decision tree (candidate splits). The second is Sparsity-aware split finding which works on sparse data, data with many missing cells. Apart from this it has other useful features such as L1 and L2 regularization and faster computing through using multiple cores on CPU.

LightGBM from Microsoft is often considered an improvement on XGBoost because of its faster training speed and higher efficiency, lower memory usage, higher accuracy through using a leaf-wise split approach and compatibility with large datasets.

CatBoost is a recently open-sourced machine learning algorithm from Yandex that provides state of the art results and is competitive with any leading machine learning algorithm on the performance front. Its advantage over LightGBM is that it handles categorical features better. We can use CatBoost without any explicit pre-processing to convert categories into numbers. CatBoost converts categorical values into numbers using various statistics on combinations of

categorical features and combinations of categorical and numerical features.  It reduces the need for extensive hyper-parameter tuning and lower the chances of overfitting also which leads to more generalized models.

The final step in obtaining the optimal solution will be to stack these models to create an ensemble of strong yet diverse models. This is especially popular in data science competitions as this article here discusses. We intend to use a simple stacking approach that averages the base models.

## Benchmark Model

As this is a Kaggle competition a benchmark model would be the best score on the competition's public leaderboard which currently stand at 1.7696 (root-mean-squared-logarithmic error). The holder of the position has published a kernel on Kaggle detailing the model and methodology which our model can be evaluated against.

## Evaluation Metrics

Since this is a Kaggle competition we have already been provided with a suitable evaluation metric on which the Kaggle submissions are evaluated. Submissions are evaluated on Root-Mean-Squared-Logarithmic-Error (RMSLE) between the predicted value and the actual revenue. The root mean square error is a common evaluation metric for regression tasks. In the present case logs are taken to not overweight blockbuster revenue movies.

## Project Design

The project will comprise the following stages:

- *Data Exploration* – Visualizing the dataset, detecting outliers, cleaning the data, checking relevance of each column to the target column, feature extraction and engineering.
- *Training* – Training multiple GBDT models as discussed above using cross-validation and random search for parameter optimization. Stacking the base models in order to arrive at a powerful ensemble model.
- *Predicting and submitting* – Using individual base models and stacked model to perform prediction on the test set and submitting predictions to Kaggle to see how individual and final models are performing both in terms of evaluation metric and position on competition leaderboard.