MLE Capstone Proposal: Predicting Box Office Returns of Films with Classification and Regression Techniques

Thomas Butler

January 2021

1 Domain Background

The film industry is in crisis. Global shutdowns due to COVID-19 have had harsh effects on studios, and arguably worse effects on the future of independent cinema. The 2010s had already brought an era of franchises - successful (Disney's MCU), and not (Universal's Dark Universe) - spawning more sequels than ever and causing major studios to buy up IP in the hope of striking gold for their own cinematic universe. The need for studios to know what films are safe investments for bringing profit from the box office has become increasingly important for the industry to survive, let alone thrive. This project will explore how data and machine learning techniques may help.

2 Problem Statement

The goal of this project is to train and deploy two machine learning models with AWS SageMaker to predict

1. if a film will make a profit, and
2. what the expected revenue from a film shall be,

where the latter may then be fed into a profit calculation. A binary classification model and regression model shall be built to answer the first and second requirements respectively.

3 Datasets and Inputs

Data shall be taken from the three sources discussed below.

1. **MovieLens Group**

   A list of films last updated September 2018, with scope from 1995 onwards.

2. **TMDB API**

   An open for public use on signing up for free account and requesting an API key. It provides additional information that is crowd source (similar to Wikipedia, in their words) to fill in information about films such as their genre and credits.

3. **Federal Reserve Bank of St. Louis**.

   A source of US macroeconomic data that shall be used to match the month of release for each of the films. This addition is motivated by the recognition of economic conditions affecting public spending, e.g., penny-pinching in a recession.

The combination of the MovieLens Group and TMDB API data sources will closely follow the dataset provided on Kaggle by the user *Rounak Banik* [1]. Choosing to go to the Kaggle dataset's source rather than taking it ready-made allows for a greater representation of an end-to-end pipeline for new data being collected for this project and customisation of the data input.

3.1 Notes on the data

Some key features of the data may be noted. There are 5,555 instances in the data. For the classification part of the project, only two classes will be considered: 1 for if the film makes a profit (3,878 instances), and 0 if it did not (1,677), giving a 70:30 split respectively.

Before feature engineering, there are 18 numeric features, one date feature and seven categorical features. Some of these features will have to be dropped. For example, there is a 'popularity' feature that is only possible to know after a the release of a film. However, features on director, writer and actor popularity will be retained. These are measures as at the building of the dataset during this project. It is assumed that popularity is stationary in time in these cases, a simplification to account for the lack of time-sensitive data.

3.2 Train-test splits and validation

The data shall be split 80:20 into train and test sets respectively. Stratified sampling based on the profit/loss class will be used to best represent the population. A validation set will be taken as a subset of the training set, also in an 80:20 split, to be used for validating the trained model before looking at the test set.

4 Solution

The process flow for building the solution of this project will come in two sections as detailed below. A high-level overview is given here with more detail on data pre-processing, training and validation given in section 7.

4.1 Local Development

1. Randomly split the data into train, test and validation sets.
2. Explore the training data set. This will involve (but is not limited to) studying the noisiness and distribution of the attributes, the correlations between attributes, and visualising the data.
3. Prepare the data for a machine learning model with tasks of data cleaning and feature engineering where appropriate.
4. Shortlist promising models. This will start with quickly training several models from different categories (e.g., linear, SVM, Random Forest, etc.) with their standard parameters. Then I will compare their performance by the evaluation metrics covered in section 6 and take two or three forward to the next stage.
5. Fine-tune hyperparameters to optimise the shortlisted models.

4.2 SageMaker and Deployment

1. Take the findings from the local development and implement them in a SageMaker notebook instance to create the model objects.
2. Deploy the models to an AWS endpoint.
3. Create a very simple web app that can interact with the models, where input will be processed by a custom Lambda function and API in the corresponding AWS services.

5 Benchmark Model

The benchmark model is set in reference to a May 2019 paper [2] in which the reported results were as follows:

- *Accuracy* = 0.83 on the test data with a Random Forest classifier for the classification of films into profit-/loss-makers, and

- $RMSE = 0.5$ on the test data with an *rpart* regression model (an R package for decision trees) for predicting film revenue.

6 Evaluation Metrics

6.1 Classification Metrics

The first part of the problem is a classification on a binary variable for which four metrics will be considered. These are *accuracy*, *precision*, *recall* and the *F1 score* (the harmonic mean of precision and recall). These metrics are defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{and} \tag{3}$$

$$F1 = \frac{TP}{TP + \frac{1}{2}\big(FP + FN\big)}, \tag{4}$$

where TP if the number of True Positives, FP is the number of False Positives, TN is the number of True Negatives and FN is the number of False Negatives. Each of the metrics have their merits. Accuracy is an overall score looking at how well the model correctly predicted the classes. This could become inflated if there is a strong imbalance in classes. For example, let's say 99% of the films make a profit. Then, the model would achieve a 99% guaranteed accuracy by predicting that *all* films will make a profit. However, without exploring the data, it is assumed that such a strong class imbalance does not exist.

Precision penalises a high number of False Positives; recall penalises a high number of False Negatives. The former would be a problem for the hypothetical studio executives if they made a decision on the model, pumping money into films that turned out to bomb. On the contrary, the latter would cause executives to miss out on profit (although the absence of money is not easily measurable). The F1 score, as mentioned above, is the harmonic mean of precision and recall, useful for when one seeks a balance between the two.

In this business context, the number of False Positives (incorrectly stating that a film will make a profit) is the figure to be minimised on assumption that this is the worst case scenario. Therefore precision shall be used as the primary metric for the classification part of the problem. The other metrics shall also be monitored throughout to ensure precision is not maximised to the detriment of other performance measures.

6.2 Regression Metrics

The regression model invites consideration of several metrics. The four metrics in consideration are the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R^2, and Adjusted-R^2. These metrics are defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \quad \text{and} \tag{6}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}^2)}, \tag{7}$$

$$\text{Adjusted-}R^2 = 1 - \frac{(1 - R^2)(n-1)}{n - k - 1} \tag{8}$$

where n is the number of observations, y is the true value, $\hat{y}$ is the predicted value, $\bar{y}$ is the mean of the target variable and k is the number if independent variables included in the model.

MAE gives a linear response on the error, whereas RMSE penalises larger deviations from the mean. Both are simple metrics good for monitoring the model performance with respect to errors.

R^2 can be interpreted as the amount of variability in the response explained by the model. Consequently, adding more features will add to the variance in the data captured. Adjusted-R^2 resolves this behaviour of rewarding models with a high number of features by adding a penalty to every additional feature that does not add to the captured variance.

RMSE shall be used for the sake of comparison to the benchmark model, supplemented by the R^2 and Adjusted-R^2 measures to monitor to variance of the data captured by the model.

7 Project Workflow

This section gives further details on the planned implementation of the project.

7.1 Data Engineering and Pre-Processing

The base dataset has been engineered to have one row per film with basic numeric and categorical features. The engineering took care to remove any null values through dropping a row if it had any null value. In section 3.1 it is stated that there are 5,555 rows. The strict exclusion of rows if any null value occurred did not have a significant impact on the row count. The step prior consisted of 5,602 rows, equating to a 0.85% discount.

The categorical features must be converted into numerical ones by encoding them. For example, the genres will be converted into dummy features where binary values (1 or 0) will be assigned based on, say, a film was a comedy or not.

In the case of a feature like the director, extra steps are taken. Most machine learning models are applied under the assumption that the number of rows is much greater than the number of features. Encoding the names of all unique directors would greatly increase the number of features in the dataset. Therefore, by using the director popularity metric (included in the dataset from the TMDB data), a cut-off for the top 30 directors for popularity may be implemented. Only these 30 would be converted into dummy variables. A similar process would be done for actors and writers.

All features will have been converted to a numeric data type. These shall be scaled, transforming each feature into the $[0, 1]$ range.

7.2 Training and validation

Linear models will be preferred in developing the machine learning solution. The planned selection to explore includes linear/logistic regression, Supper Vector Machines, decision trees and random forests. Deep learning techniques will be avoided due to the lack of explainability.

The exact implementation will depend on the model training costs. SciKit-Learn's *K-fold cross-validation* feature could, for example, train a model ten times with the validation set taken as a unique 10% of the training data each time. This would give an array of ten evaluation scores, which would also allow a variance measure to be calculated on the model performance. If $K = 10$ folds is too expensive, K will be reduced accordingly.

SciKit-Learn's grid search will be used to tune hyperparameters for improving the model.

References

[1] Rounak Banik. *The Movies Dataset*. 2018. URL: `https://www.kaggle.com/rounakbanik/the-movies-dataset`.

[2] Sarah E. Joseph. "What Makes a Movie Successful : Using Analytics to Study Box Office Hits". In: (2019). URL: `https://trace.tennessee.edu/utk_chanhonoproj/2252`.