Movie Box Office Prediction

Aditi Dandekar
aadandek@ncsu.edu

Priyanka Krishna Hebsur
phebsur@ncsu.edu

Shreya Parikh
scparik2@ncsu.edu

Suraj Patel
sjpate22@ncsu.edu

1 Background and Introduction

1.1 Problem Statement

In the entertainment world, movies made an estimated $42.5 billion in 2019. So the capacity to foresee a film's revenue before its actual release can diminish its financial risk factor. Based on the meta-data related to movies such as casts, director, release dates, estimated budget, etc, we can define a useful correlation between movie-related data and movie box-office revenues. In this project, we aim to explore the data set, identify the relevant attributes, design and evaluate different prediction models. These models will predict the Box Office Revenue for the movies which in turn will help us to gain meaningful insights into vast movie data set. We also intend to compare the prediction models against a set accuracy and loss metric to analyze and choose the best approach to predict a movie's revenue.

1.2 Literature Survey

Movie industry being a multi-billion dollar business, the ability to predict box office revenues can help investors assess the financial risk involved. However, accurate predictions are not easily available due to the complex relationship between movie related data and corresponding revenues. Moreover, the increasing amount of information available on online movie databases poses a challenge for their effective analysis.

In the paper titled 'Movie Revenue Prediction Using Regression and Clustering'[1], the authors study relation between movie factors and its revenue and build prediction models. It was also shown in their paper that linear regression without clustering gives a high RMSE while linear regression with EM clustering yields lowest RMSE.

Researchers have also used deep learning for this particular problem. There have been some attempts to generate a meaningful analysis using deep neural networks for prediction and convolution neural networks for feature extraction from movie posters[2]. In this particular paper titled 'Predicting movie-box office revenues using deep neural networks', authors use a convolutional neural network (CNN) to extract features from movie posters. They also conduct a comparative study of the proposed multi-modal deep neural network with other prediction techniques. Other attempts include utilization of support vector machine and natural language processing techniques to efficiently understand the feature space of pre and post release data of movies[3]. Based on their study, the authors conclude that budget, IMDb votes and number of screens play vital role when predicting a movie's box office success.

We follow similar line of thought using Random Forest, Linear regression and Deep neural networks. We've compared the results obtained from each of these techniques and come up with an understanding about which one of these would be most suitable for our use-case.

2 Method

2.1 Approach

Regression analysis a set statistical processes for estimating relationships between a dependent variable and one or more independent variables. When output variable is real or continuous the problem is considered to be Regression problem. For the current project, as we are trying to predict movie revenue, it is considered to be a Regression problem. Based on this understanding, the techniques that we considered for the problem statement are given below:

- **Random Forest**: Random forests or random decision forests are an ensemble learning method for regression that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees. Random Forests often gives high-accuracy through cross-validation. It will handle the missing values and maintain the accuracy of a large proportion of data. The non-linear nature of Random Forest makes it a better choice over linear algorithms.

- **Linear Regression**: Linear regression is an approach to model the relationship between the target attribute and the one or more independent attributes. In our case the target attribute was revenue and the independent attributes were the other features in the dataset.

- **Neural Networks**: Neural networks are a series of algorithms that mimic the operations of a human brain to recognize relationships between vast amounts of data. Basically, it tries to estimate the coefficients of some non-linear function defined by our network architecture.

We also considered using K-Fold Cross-Validation technique to assess each technique more accurately. The loss function used was Root Mean Squared Log Error.

2.2 Rationale

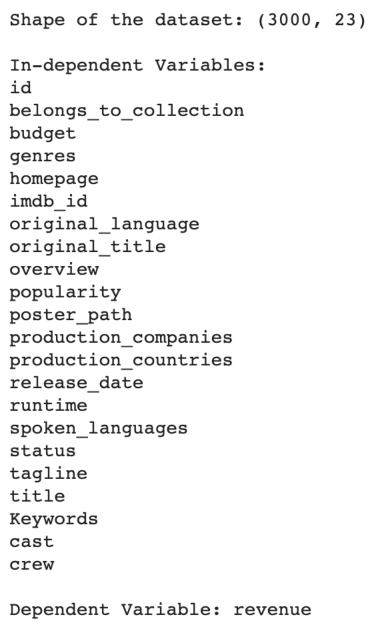
We chose techniques that would be suited for a regression problem like this. The above techniques were implemented after performing an extensive data exploration task. We identified the different attributes and their relation to the target variable. We also saw that the revenue attribute had a skewed distribution, hence we decided to perform a log transformation and use Root Mean Squared Log Error as the loss function. We also performed simplification tasks like one-hot encoding, replacing missing values, simplifying complex dictionaries into simple attributes etc which are explained in the next section.

3 Plan and Experiment

3.1 Data set

The given dataset consists of metadata on over 7,000 past films from The Movie Database. The number of training samples is equal to 3000, and the number of testing samples, or the number of predictions needed to be made is equal to 4398. Although this was a huge dataset, only 3000 records were used in our implementation as the test data set consisting of 4398 records did not the ground truth values for revenue. Data points provided include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries. Each movie is labeled with an id.

This data is acquired from Kaggle competition which includes movie details, credits and keywords.



```
Shape of the dataset: (3000, 23)

In-dependent Variables:
id
belongs_to_collection
budget
genres
homepage
imdb_id
original_language
original_title
overview
popularity
poster_path
production_companies
production_countries
release_date
runtime
spoken_languages
status
tagline
title
Keywords
cast
crew

Dependent Variable: revenue
```

Figure 1. Dataset shape and columns

3.2 Hypotheses

Our experiment aims to predict the revenue of a movie given its meta data. We also aim to understand how the various attributes like release date, cast, production company and so on impact the box office revenue of a movie.

Some attributes like language and genre could have multiple values for a movie. We wanted to understand the most relevant values of these attributes which may impact the revenue. For eg. let's consider attribute - 'genre, it consisted of 20 value types. Based on the provided data set, all these 20 values didn't seem to be relevant for estimation purposes. Moreover, considering all these values would only increase the vector dimensions when one-hot encoding is performed on these attributes. So there was a need to reduce this values such that we only focus on the attribute values have most impact or summarizing the overall trend of our data set. We conducted such reduction for multiple attributes like genre, production companies, countries and languages.

By looking at the data we found out that the target variable revenue had a skewed distribution. The problem at hand was to identify how the movie revenue was spread. From the graph shown in Figure 7, we saw that revenue variable was skewed. To resolve this we conducted a log transformation of the target variable revenue. This also helped us in calculation of the loss function RMLSE. As the log transformation was already done, in order to calculate loss we just had to measure the RMSE of predicted revenue.

3.3 Experimental Design

3.3.1 Data Exploration and Feature Engineering

The first part of our project was to explore the data and then pre-process the data. It also included feature engineering where we had to explore how much each feature has an impact on the revenue and extract only those which are relevant. Pre-processing included handling missing values, noise, structuring some of the features represented as complex structures. Based on our hypotheses we conducted experiments to determine the significance of each attribute on the revenue and using those observations, we implemented the following techniques to clean and process the data to get better predictions:

- Check for null values and replace them with default values
- Discretize multi-valued features like "production companies", "cast", "genre", "production country", "spoken languages". The data transformation included processing a complex

dictionary into new features of binary type. For example, genre is represented by a list of dictionary with id and name as the key. Only the name is of importance for our analysis. Hence the dictionary was converted into a list of genre names that a movie belongs to. Upon data exploration, we found that most genres have very few movies mapped to them. The results of that experiment can be seen in the figure below.Hence we set a threshold value for the number of movies in a genre and converted genre into categorical features, one for each genre. The values allowed for each feature is either 0 or 1. Similar transformation was done to language and country. The results of exploration can also be seen in the bar plot below.

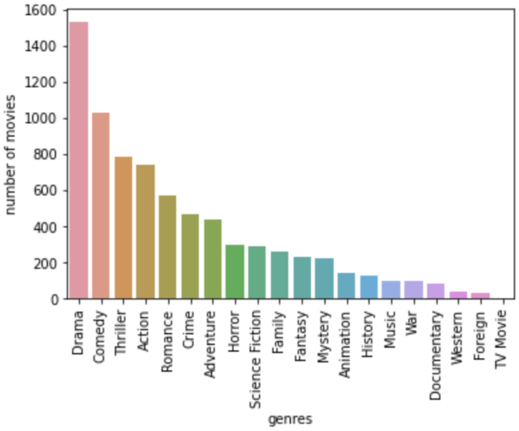


Figure 2. No. of movies belonging to each genre

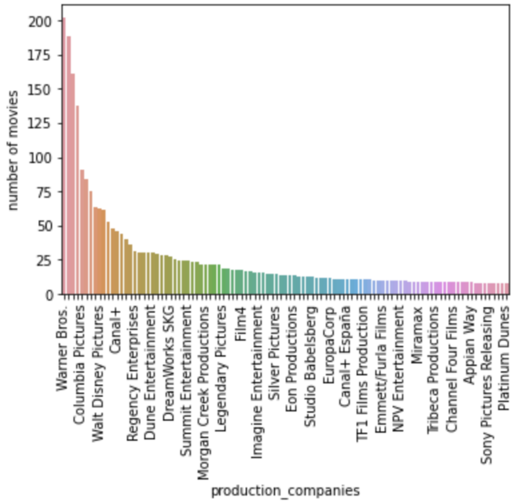


Figure 3. No. of movies produced by each prod company

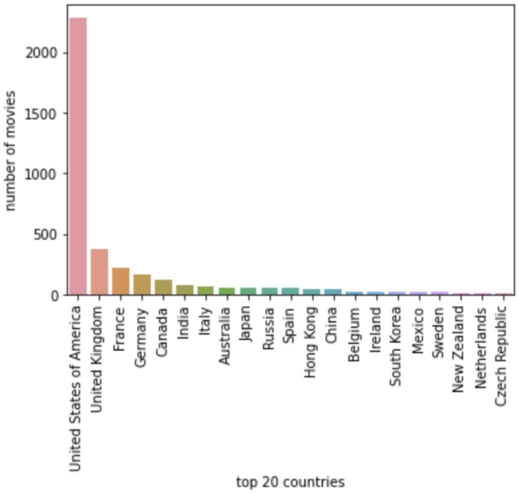


Figure 4. No. of movies from top 20 countries

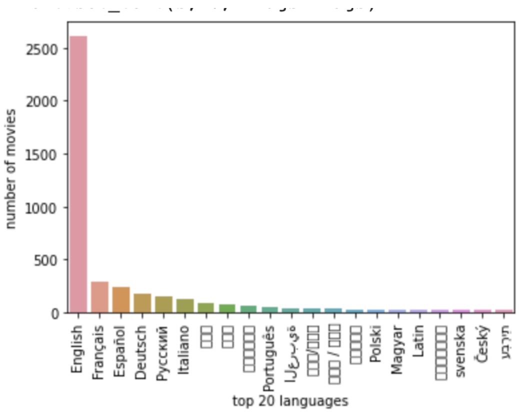


Figure 5. No. of movies in top 20 languages

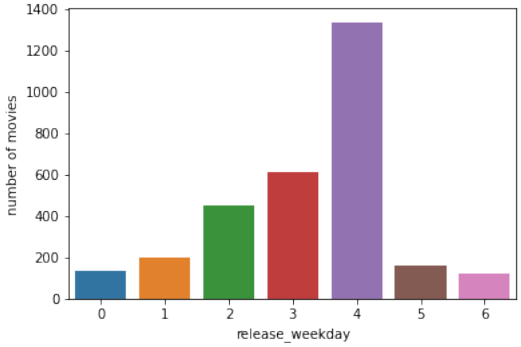


Figure 6. No. of movies released on each weekday

- The next observation was that we had a large number of records for which the movie release date was on a specific weekday or quarter. This can be seen in Figure 6 where most movies are released on a Friday. Hence we converted the "release date" feature into "release month", "release year" "release day", "release quarter"
- Some nominal attributes like "homepage", "tagline" were converted to binary attributes based on whether the value is available or not.
- Some nominal attributes like "id", "imdb id", "title" etc were dropped as they did not have any influence on the revenue of the movie.
- The target variable "revenue" has a skewed distribution. Performing a log transformation makes the plot closer to a normal distribution. Hence, log transformation was done on the "revenue" attribute

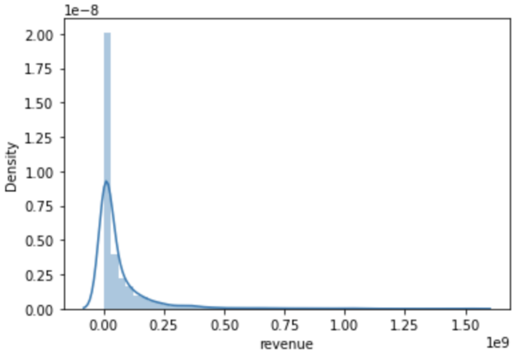


Figure 7. Original Skewed Distribution for Revenue

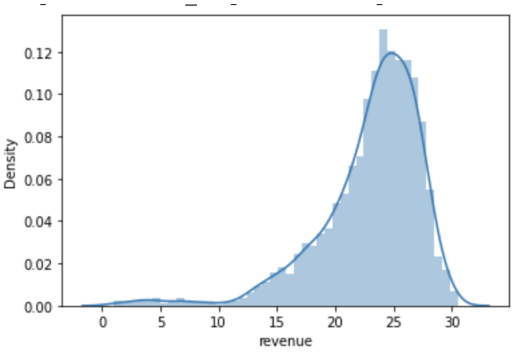


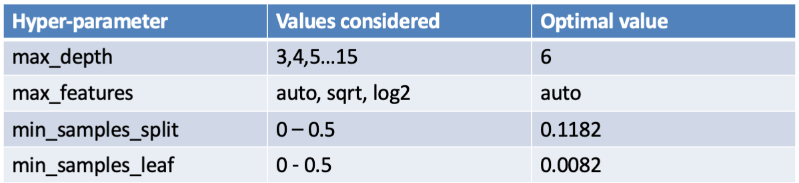
Figure 8. Revenue Distribution after log transformation

3.3.2 Prediction Models

Once the data pre-processing was done, we then moved onto building the prediction models. As stated in the approach, we built 3 prediction models to predict the revenue of the movie. The details on those are as follows:

- Random Forest: It is an ensemble learning method for regression. The hyper-parameters taken into consideration were:
  - max_depth: It refers to maximum depth of the tree
  - max_features: The number of features to consider when looking for the best split
  - min_samples_split: the minimum number of samples required to split an internal node
  - min_sample_leaf: minimum number of sample required to be at a leaf node

  We used sklearn.ensemble library in python to build a Random Forest Regressor. We used hyperopt to optimize the model. hyperopt is a python library for carrying out hyperparameter optimization for machine learning algorithms. We used this library to obtain the most optimum hyperparameters for each of our regression in random forest.



Hyper-parameter	Values considered	Optimal value
max_depth	3,4,5…15	6
max_features	auto, sqrt, log2	auto
min_samples_split	0 – 0.5	0.1182
min_samples_leaf	0 - 0.5	0.0082

Table 1. Hyperparameters for Random Forest

- Linear Regression: We had a large number of features. Hence, we got a complex linear equation with more than 100 coefficients and intercept value of about 3.102. We used sklearn.linear_model library in python to build a Linear Regression Model
- Deep Neural Networks: We used a Fully connected layered Neural Network architecture for the model. After some experiments we finalized the architecture of the Neural Network model to be 4 Dense Layers with 64, 32, 16 and 1 neurons in each. The hyper-parameters and their values for the model are given below:
  - Adam Optimizer with learning rate 0.005
  - 50 epochs
  - Sigmoid Activation Function

  We used keras library in python to build our Neural Network Model.

3.3.3 Validation

The validation method used was K-Fold CV. K-Fold CV is a cross validation method to estimate the skill of the machine learning models. In this technique, the data set is split into k groups and for each unique group:1 partition is taken as a test set and the remaining as the training data set. The model is fit on the training set and evaluated on the test set. This is repeated to calculate an evaluation score of the model. K-Fold CV was used for Random Forest Regressor and Neural Networks with the value of k being 4 and 5 respectively. The python library used to implement K-Fold CV is sklearn.model_selection. It is worth mentioning that due to limitation of testing data as discusses initially (under dataset section), we have had to split the available training data into 70-30% to get the testing data as well.

4 Results

After performing the aforementioned data pre-processing, modelling, training, analyzing and visualization techniques, we have obtained the following results. Figure below shows us the correlation between each of the numeric attributes.

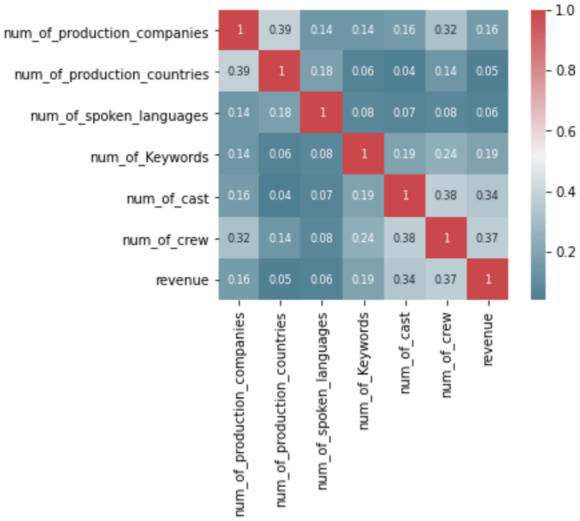


Figure 9. Correlation Matrix for numeric attributes

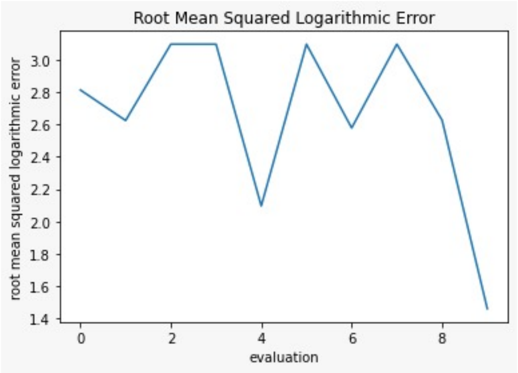


Figure 10. RMSLE for Random Forest

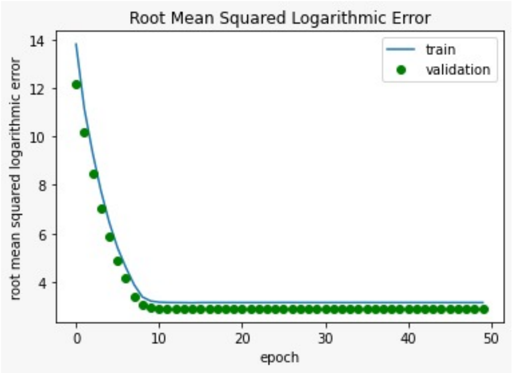
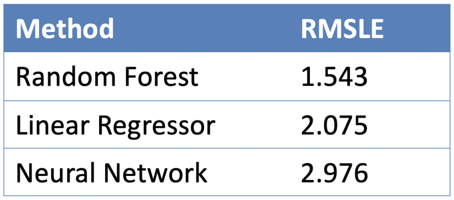


Figure 11. RMSLE for Neural Network Model

Figure 10 shows that how RMSLE decreases as we move forward with number of evaluations being perform by fmin to select the best hyper-parameters with respect to lowest RMSLE. Figure 11 shows the trend in training and validation errors as we increase the epochs at each iteration for our Neural network.



Method	RMSLE
Random Forest	1.543
Linear Regressor	2.075
Neural Network	2.976

Table 2. RMSLE value for each prediction model

We used Root Mean Squared Log Error to evaluate the models. It is the ratio of (the log) between the actual values in our data and predicted values in the model. Initially revenue underwent a log transformation. So, on prediction, root mean square error was calculated to obtain the RMSLE value. All the 3 models gave almost similar results with Random Forest Regressor being the best. Random Forest regressor gave the best results, followed by Linear Regression and lastly Neural Networks. Neural networks had the highest error mainly because of a small dataset. It would have performed better with more records. We obtained a value of 1.543 for Random Forest Regressor, 2.075 for Linear Regression and 2.976 for Neural Network model. The results were compared with the top ranked submissions on the Kaggle leaderboard and found out to be positively comparable to those results.

5 Conclusions

The aim of this project is to predict movie revenue based on various movie attributes. We have implemented several data pre-processing techniques as mentioned earlier in this report and we have seen success in terms of transforming our raw, complex data into clean data. We realized that data exploration is one of the main phases in building prediction models. Many attributes were not necessarily impacting the revenue were discarded. On comparing our RMSLE score, We stand on 12th position in the Kaggle leaderboard with >1000 entries.

We also understood the revenue prediction is a complex problem since there are a lot of attributes that impact the revenue of a particular movie. We implemented three prediction models to predict the revenue. All three models gave similar results although Random Forest out-performed others because it is an ensemble method and is better suited for a regression problem like this. Linear Regression works better only in a few cases for example when the underlying function is truly linear. This was not the case in our data. Neural Network would have performed better with a larger data set. One of the future scopes of this project is to collect more data to train on Neural Network model. We can collect this through the The Movie Database API.

6 Code Reference

The Jupyter notebook containing the implementation and the data set can be found in the following Github repository: Repo Link

References

[1] Pakom Walanaraya, Weerapat Puengpipattrakul, and Daricha Sutivong. Movie Revenue Prediction Using Regression and Clustering. In *2018 2nd International Conference on Engineering Innovation (ICEI).*

[2] Yao Zhou, Lei Zhang and Zhang Yi. Predicting movie box-office revenues using deep neural networks

[3] Nahid Quader, Md. Osman Gani, Dipankar Chaki and Md. Haider Ali A machine learning approach to predict movie box-office success. *2017 20th International Conference of Computer and Information Technology (ICCIT).*