DS 4400: Machine Learning and Data Mining I

Spring 2024

Project Report

Project Title: Reel Success: Machine Learning Analysis for Movie Financial and Rating Performances

Team Members: Valerie Jap, Tarun Badarvada, Julian Getsey

Link to the code: DS4400_FinalProject.ipynb
Link to the presentation video recording:DS4400-Final Presentation.mp4

Problem Description

According to Forbes, 80% of films lose money. Similarly to the philosophy of gamblers, film companies rely on the success of one or few movies to make up for the losses of many others. While this has proven to be a successful approach for top movie companies, independent film producers don't possess this same safety net and have a much lower likelihood of success. Our team feels the difficulty of breaking into the film industry is a significant problem. Our approach towards a solution is a series of machine learning models that predicts a movie's box office success using features such as ratings, budgets, genres, runtime, and popularity that will bring foresight to independent film companies.

Personally, each team member enjoys watching movies, and we ourselves were motivated to learn the biggest drivers of a movie's success, as well as to explore if these drivers changed throughout the history of the box office. Our big-picture goal, however, is to provide a machine learning project that will allow those working in the film industry to leverage data to understand how they can produce successful and popular movies without the expensive and time-consuming process of trial-and-error.

Through our machine learning implementation and analysis, we specifically explored two different measures of a movie's success: box-office revenue and rating. We chose to create a linear regression model and neural network in order to make predictions on revenue, and KNN and decision tree models in order to make predictions on movie ratings. Together, these models provide assessments of several factors filmmakers should consider when producing a movie.

Dataset and Exploratory Data Analysis

The dataset was sourced from The Movies Dataset, an open-source dataset on Kaggle that included information on 45,000 movies posted by Rounak Banik, a Data Science Fellow at McKinsey & Company. The dataset includes several spreadsheets with various metadata regarding each film released from the late 1800s to 2017. The columns include cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDB vote counts, and vote averages.

After inspecting the dataset, we conclude that the quality is relatively high. There are some null values that are common for a dataset of this size, but most of the data is of its correct data type. However, there is one inconsistency of a movie poster's filename being listed as a budget which was later addressed during our data preparation. Before any data cleaning, however, our data frame appeared exactly as these two examples of data points below.

Data Point #1: Toy Story

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_id	original_language	original_title	overvi
0	False	{'id': 10194, 'name': 'Toy Story Collection', ...	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	http://toystory.disney.com/toy-story	862	tt0114709	en	Toy Story	Led by Woo Andy's toys happily in his

view	...	release_date	revenue	runtime	spoken_languages	status	tagline	title	video	vote_average	vote_count
ody, s live ds ...	...	1995-10-30	373554033.0	81.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	NaN	Toy Story	False	7.7	5415.0

Data Point #2: Father of the Bridge Part 2

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_id	original_language	original_title	overvi
4	False	{'id': 96871, 'name': 'Father of the Bride Col...	0	[{'id': 35, 'name': 'Comedy'}]		NaN	11862	tt0113041	en	Father of the Bride Part II	Just wh George Ban has recove from his

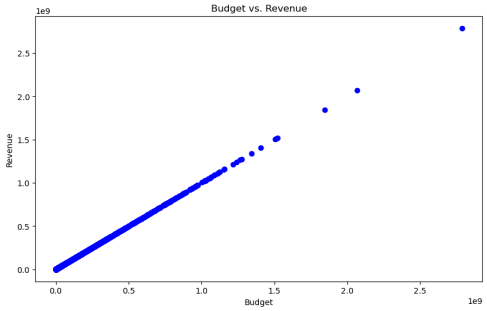
release_date	revenue	runtime	spoken_languages	status	tagline	title	video	vote_average	vote_count
1995-02-10	76578911.0	106.0	[{'iso_639_1': 'en', 'name': 'English'}]	Released	Just When His World Is Back To Normal... He's ...	Father of the Bride Part II	False	5.7	173.0

In order to clean our data, there were several techniques applied to the data being used across all of our models and others that were model-specific.

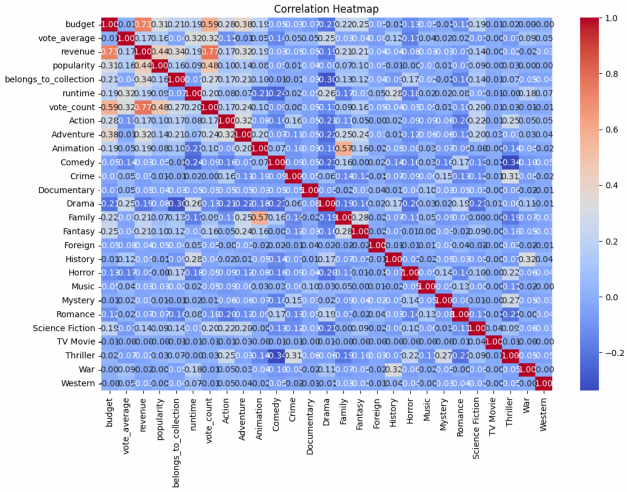
Across all models, we used one-hot encoding in order to encode categorical columns such as genre, collection, adult, original language, and status depending on the necessary features. We also handled null values based on the stylistic preferences of the model's implementer. For example, we dropped the rows of null values for the data in our regression and neural network, while performing imputation on the data by the most frequent value for the data in our classification algorithms (KNN and Decision Tree). Lastly, we handled the particular case of a movie poster's filename in the budget column by dropping the row altogether. There were some features that didn't appear useful for any of our machine learning models, such as the homepage or imbd_id, so we decided to not include these as features.

One more particular piece of data preparation was the addition of a rating classification for our classification algorithms. We placed a rating into one of three classes (low, medium, or high-quality rating) using bins and separated them using quantiles.

One high-level trend we explored was the correlation between budget and revenue. We expected that these two columns would be directly, and positively correlated given our intuition of a movie with more production resources resulting in greater financial return. However, we were unable to effectively analyze this relationship until we had cleaned the data. By plotting a scatter diagram, we confirmed our original expectation of a direct, positive relationship.



Additionally, we also plot the below heatmap to visualize the relationship between various features of the dataset and the target variable, movie revenue. The color-coded matrix provides immediate insight into which features are positively or negatively related to one another. For example, the budget has a significant positive association with revenue, suggesting that as the budget for a film increases, so does its potential revenue. Vote count, another strongly connected aspect with revenue, may show the impact of a film's popularity on its financial success.



Although this relationship seemed to be a simple linear one, we were still interested in exploring the nuances and more complex relationships between different features and a

movie's performance. Therefore, we were eager to proceed with creating our machine-learning models.

Approach and Methodology

Box Office Revenue Predictor

The data was prepared as explained in the section above and then was split into training and testing datasets using an 80% 20% split. After splitting the dataset, 2 models were trained using the training data.

With the dataset we have collected, we needed to identify the features that pertain to the influence of Box Office Revenue. For our features, we selected the following to run through our models:
belongs_to_collection: this indicates whether the movie is part of a franchise - we felt this was important since we understand the heightened market value of a sequel franchise and we are interested in understanding the correlation between this feature and box-office revenue
budget: film's total budget or production value spent
popularity: measure the film's overall popularity
runtime: total time of film
vote_average: voting score collected from IMDB, which is an online movie rating platform where users can post reviews and rate movies based on their preferences
vote_count: total number of votes
genre: indicates the genre or category of a movie - this category can have multiple genres so we encoded our data to label our films with multiple genres
The target variable that we are measuring for this is *revenue*. This leads us to consider the model that best suits our data processing needs.

1. **Linear Regression**

Based on the data exploratory section, there seems to be a linear relationship between the features. Therefore, linear regression is suitable for estimating how budget increases or changes in other features translate to an increase in revenue. Linear regression is a simple regression model that serves as a baseline for other models. It does not require hyperparameter tuning.

Using the scikit-learn library, we successfully build a model that captures the linear relationship between movie features with the target variable of movie revenue. The result will be discussed in the next section.

2. **Neural Network:**

For our study, we chose to build a Feed-Forward Neural Network Architecture to predict a movie's revenue for this particular regression problem. For this problem, the relationship between the input variables and the output variable is usually nonlinear. In this

case, the feed-forward network is best at capturing complex non-linear models to catch intricate relationships. Our previous model Linear Regression proposed a simplified model that was not able to pick strong relationships between its features and target. With this Neural Network model, we set out to uncover hidden patterns within the data, unlike the traditional linear model.

For this network, we implemented using PyTorch by leveraging the 'nn.Module' class for constructing the neural network layers. We have employed an input layer, one hidden layer, and an output layer. Each layer is passed through a Rectified Linear Unit (ReLU) activation function to enhance the model's capacity to learn. For the training procedure of this model, we utilized the Mean Square Error as our criterion for this model, and we used the Adam function for our optimizer. As for selecting our hyperparameter, we had several parameters we chose to run with this model including hidden layer size, learning rate, batch size, and number of epochs. For the hidden layer, we chose a size of 64. The learning rate parameter is used for the Adam optimizer to help determine the step size during the training period. For this, we chose a learning rate of 0.001. The batch size defines the number of samples processed before updating the parameters. For this, we chose a batch size of 64. The number of epochs defines the number of times the dataset is put through the training process and is passed forward and backward. For this, we chose 50 epochs.

Movie Rating Predictor

For our features, we selected the following to run through both of our classification models:
budget: film's total budget or production value spent
original_language: the original_language in which the movie was filmed
runtime: total time of film
status: whether the film has been released or not

The response variable that we are measuring for our KNN and Decision Tree models is *rating quality*. These two models are unique from the others in the sense that they are classifiers. We decided to use classification as our movie rating predictor is inspired by recommendation systems common on streaming services. Therefore, our end goal in future implementation is to determine whether a viewer will enjoy a movie or not. As stated in our data preparation overview, we are classifying a movie as having a poor, medium, or high-quality rating.

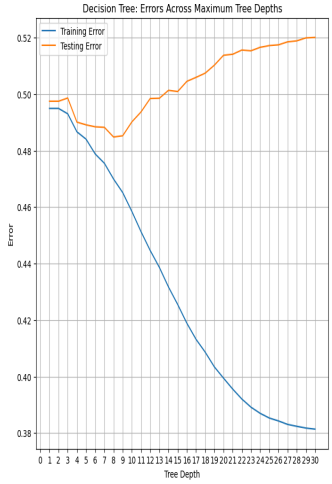
1. K-Nearest-Neighbor:

We decided to implement a KNN classifier in order to predict movie rating quality due to our future goal of creating a movie recommendation system, because it utilizes similarities between different data points in order to make predictions. Therefore, if a viewer enjoys a movie, it is likely they will enjoy a movie with similar qualities. KNN is also a simple algorithm, and based on Occam's Razor, we decided that it would be important to explore simple models in addition to the more complex ones such as neural networks.

We used sci-kit learn packages in order to generate our KNN model. We created a for loop in order to perform cross-validation in order to find the best value of k, which was 7 when analyzing recall specifically. We decided to use recall in order to choose our best value of k due to the fact that we'd rather recommend a wider net of movies as opposed to a stricter one.

2. Decision Tree:

Next, we implemented a decision tree due to its interpretability and its ability to handle both categorical and numerical values. Given we implemented some more complex models such as neural networks in which the processes were harder to make sense of visually, we decided to include a model in which the process of classifying data points was more intuitive and digestible. We wanted to optimize our decision tree in order to find a balance between bias and variance. We used sci-kit learn packages again to create our decision tree. Using a for loop to iterate through maximum depths, we found an optimal depth of 8.



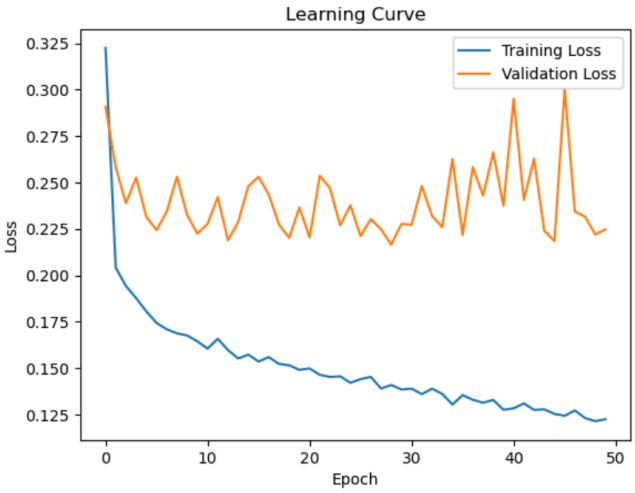
Metrics & Evaluation Methodology

Linear Regression:

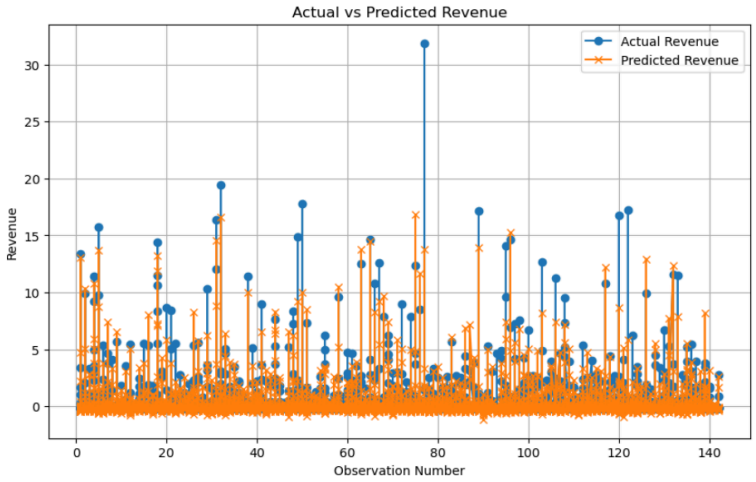
The model's performance on training data yielded an R^2 value of 0.750, indicating that it explains nearly 75% of the variance in the dataset. The model retained high predictive ability and explained almost 70% of the variance when applied to testing data (R^2 = 0.698). The slight decline in R^2 from training to testing indicates that, while the model may have some degree of overfitting, the model generalizes pretty well to unseen data. The model's robustness on the testing set supports the initial assumption of a linear relationship and shows the model's ability to predict movie revenue.

Neural Network:

The model's performance on training data yielded an R^2 value of 0.814, showing a strong performance in terms of the fit of its data. Along with this, we were able to get a Mean Squared Error of 0.225, which is a relatively low score. The figure below shows a chart of a learning curve where we have Training Loss and Validation Loss. The training error, also known as training loss, is a measure of how well the model is performing on the training data during each iteration of the training process. The validation loss is a measure of how well the model generalizes to unseen data, typically evaluated on a separate validation dataset. With each iteration, our training loss was able to exponentially decrease through each epoch showing the data given indicates a strong correlation to our results.

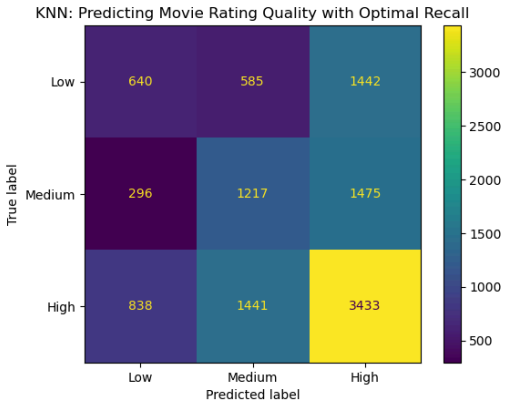


Our next visualization below shows our results of Actual vs Predicted Revenue. We can see that the visual indicates several outliers within our predictions which show that there may be other factors that we might not be considering in the model that require a deeper dive into. However, our model's predictions for the most part are relatively close to the actual box office revenue results.



K-Nearest Neighbor:

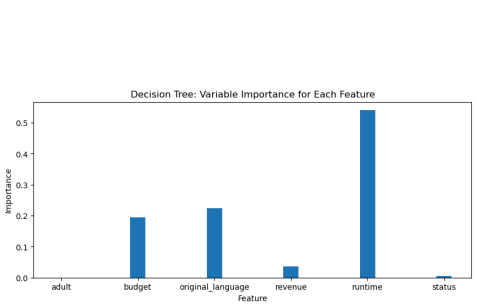
In order to evaluate our KNN model's performance, we used a confusion matrix and observed the distribution of predictions and actual values. After displaying this matrix, we observed that the most common prediction and actual value was a high-quality movie rating, followed by medium-quality, followed by low-quality rating. While the most common prediction is a high-quality rating regardless of the actual prediction, we see that the model is generally effective at distinguishing an actual medium-quality rating from a potential low-quality one, whereas it is not as effective as distinguishing an actual low-quality movie rating from a potential medium-quality one. Although most actual values are high-quality ratings, the model clearly favors high-quality predictions in general.

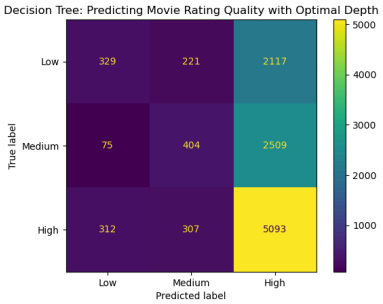


Decision Tree:

We used a confusion matrix in order to evaluate the Decision Tree's performance, and found that this model was even more biased in the sense that its

most common prediction was a high-quality rating. Similarly to the KNN confusion matrix, the model is fairly effective at distinguishing a true medium-quality rating from a potential low-quality rating, but not as effective at distinguishing a true low-quality rating from a potential medium-quality rating. In addition to the confusion matrix, we plotted the importance of each feature, and found that the most important feature was runtime, while budget and original_language were also important. One possible explanation for runtime's importance is the fact that early movies (1800s) tended to have long runtime's, and also hold high movie ratings.





Discussion and Result Interpretation

Box Office Revenue Predictor:

For our Revenue Predictor, we look into two models: Linear Regression and Neural Network. Linear Regression is a relatively less complex model compared to Neural Network. Considering this, our Neural Network was capable of predicting at a higher r-squared compared to Linear Regression. This is due to the non-linearity that Neural Networks introduce into a prediction model, whereas on the other Linear Regression focuses on drawing a linear boundary between its features and its target. We expected the Neural Network to have better accuracy than the Linear Regression model. However, the Linear Regression model performed very closely on par with our Neural Network with a difference of just 0.06 which was a very surprising result to us.

Movie Rating Predictor:

Analyzing our confusion matrices, we conclude that our KNN model performs better than our Decision Tree model. While both models seemed to be sensitive to the fact that most movie ratings are of the highest-quality, the Decision Tree model seemed to be the most sensitive, and made a disproportionate amount of predictions towards high-quality ratings. Therefore, the simpler KNN model did outperform the more complex decision tree with a maximum depth of 8, confirming our consideration of Occam's Razor. Neither model

performed as well as we had hoped, which could indicate there are many more factors not found in our dataset that would be helpful in predicting a movie's rating.

Future Work

There are several elements of our project we feel we can develop further in order to provide even more insight to filmmakers as to how they can create movies that are both well-liked and successful.

Firstly, we hope to use News Sentiment Analysis from both new sources and social media in order to capture the qualitative components of how movie critics and viewers engage with a movie. In doing this, we can explore user thoughts of movies beyond a single rating. For example, a movie may be very popular, but not well-liked by most viewers or vice-versa. In all, sentiment analysis will provide an opportunity for qualitative considerations which are especially important when it comes to entertainment.

Next, we hope to expand on our movie rating predictor findings by combining our dataset with viewer data. Having access to how viewer rates particular movies or their favorite genres would allow us to predict whether they would enjoy a particular movie and whether it should be presented to them in an ad or on a streaming service.

Lastly, we hope to test our models on 2024 movie data in order to predict the box office for the current year. While this is a much simpler work compared to using news sentiment analysis or making recommendations to users, we feel it would be worthwhile to explore recently-released movie performances, and which factors in particular impact these performances.

Conclusion

To summarize, we set out our goal to evaluate the film industry and ask the question: how do we create a successful movie? This is an ongoing question within the film industry being an extremely volatile field. We wanted to evaluate films with various metrics and with our results we were able to have several takeaways. We were able to conclude that there is a blend of linear and nonlinear relationships within movie data. Many different factors contribute to a movie's success that we are not taking into consideration. Genre is a huge contributing factor to the response variables compared to other features. These takeaways are just a starting point for exploring many more factors that help a movie's success whether it is box office revenue or a movie's rating.

References

Banik, R. (2017, November 10). *The movies dataset*. Kaggle. https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset

Moore, S. (2019, January 3). *Most films lose money!*. Forbes. https://www.forbes.com/sites/schuylermoore/2019/01/03/most-films-lose-money/?sh=165ac89739f2