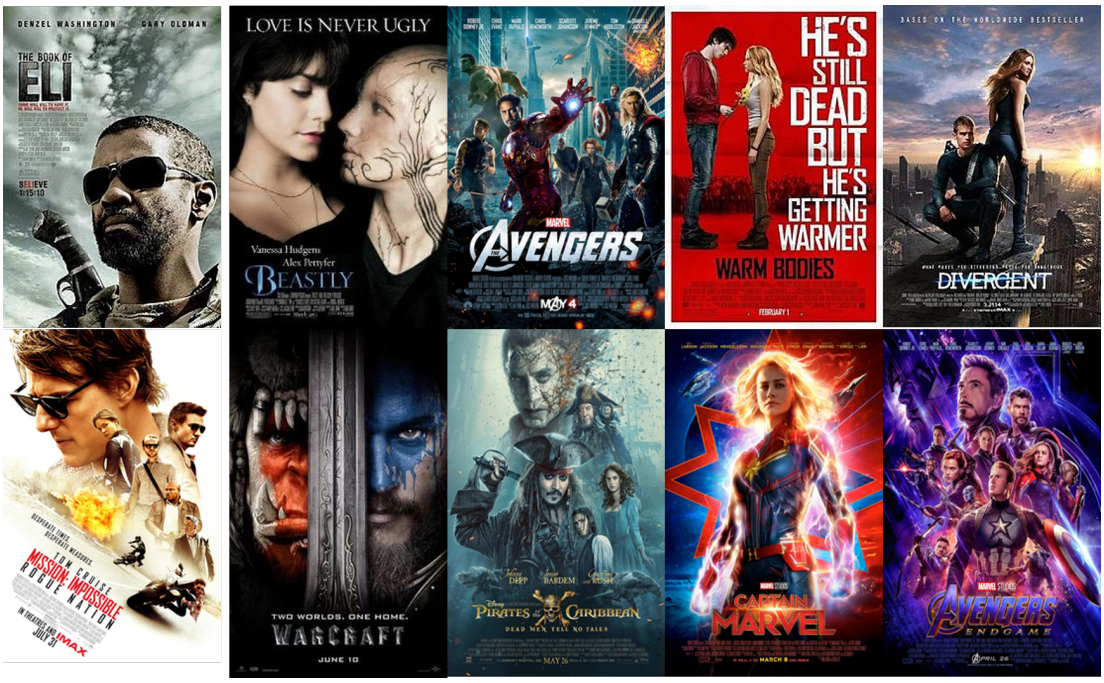

What can Hollywood Learn from YouTube?

John Jack Lewis
Department of Computer Science
Florida Southern College
Lakeland, Florida
jklewis99@gmail.com

***Abstract*—Though most of a film's marketing budget is directed toward television advertisements, the rise in video-sharing platforms provides an additional outlet for promotion. Arguably, the most notable of these video-sharing platforms is YouTube. This paper addresses how the data of a film trailer released on YouTube – like number of views, comments, likes, and dislikes – help predict the generated revenue for a film, additionally exploring how these features expand on the effect of other features in the filmmaking process. The data used to evaluate these features is generated from films released in the 2010s in The Movie Database (TMDb) and video trailer statistics from the YouTube Data API. Specifically, these features are used to train machine learning models to evaluate which models are most successful in revealing connections between these features and revenue. This work closely examines the impact of the statistics of a film trailer from YouTube to help producers better understand the impact of their investments.**

***Index Terms*—machine learning, decision trees, neural networks, CART, regression, film**

I. Introduction

In the 2010s, 8,319 films were released with a total box office gross over $109 billion [1]. Of the highest grossing films from each of the ten years in the decade, Disney owns the production rights of eight. Understanding how Disney and other big production companies are able to acquire so much profit from films is a growing question for filmmakers and producers across the world.

One method for garnering a greater understanding comes with the use of machine learning. Machine learning is a type of artificial intelligence that bases decision-making on patterns in data, without requiring explicitly programmed instructions. The simplest example of a machine learning model is the "line of best fit", or $y = mx + b$, which uses trends in data to form a line that follows the path of linear data. However, machine learning often stretches the complexity of what is observable by humans, finding intricate nonlinear correlations and multidimensional relationships. For machine learning models to be effective, they require sufficient data, but in the 21st century, that is less of an issue than ever before. Today, digital content is being consumed like never before. By 2005, humans had created only 130 exabytes (EB) of data, but by 2018, that number grew to 33,000 EB, which is greater than a 25,000% increase in only 13 years, and it is expected to grow to 175,000 EB by 2025 [2]. To download all of that data, at 25 megabytes per second (the average connection speed), it would take 1.8 billion years. With such a vast amount of data generated every day, companies and organizations continue to rely on artificial intelligence to solve business needs that

cannot be effectively performed by humans. Machine learning helps Tesla develop its self-driving cars, Apple to prevent other fingerprints or faces from accessing your personal information, Netflix to find out what you want to watch next or what they should make next, and much more. These companies are able to improve their success because of access to data.

Data in the film industry is continuing to expand. Social media, in particular, has had a huge impact on film. A study from Twitter and Nielsen showed that 87% of users made the decision to see a movie based on tweets [3]. Applications and services like the Internet Movie Database (IMDb) [4], The Movie Database (TMDb) [5], Metacritic [6], Rotten Tomatoes [7], and Letterboxd [8] allow users to see how critics and non-critics alike feel about a movie, to connect with users who share similar tastes in film, to write reviews and rate movies, and even to respond to other reviews. Often, these applications can connect with other social media applications to share individual movie reviews. Additionally, production companies leverage user presence on social media to offer exclusive deals to promote their films [9].

One of the most notable platforms used by films is YouTube. YouTube is effective for film promotion in a number of ways: advertisements, trailers, clips, and user content. As marketing begins to shift funds from television to online digital media, production companies invest in YouTube advertisements which play before videos, but they also release trailers on their own channels [10] [11]. Additionally, popular channels like Movieclips Trailers [12] and Movieclips [13], both run by Fandango, which have 15.1 million and 46.8 million subscribers, respectively, are able to expand the impact of movies. YouTube has also become a platform for movie reviews and video essays, which may help production companies reach an even greater audience.

As audience demographics change, and the medium from which movies are consumed continues to shift, the film industry has struggled to make confident decisions about how to approach the future [14]. For centuries, movie theaters have had exclusive rights to release films within the first 90 days of official release. However, as some films begin to move toward streaming platform releases instead of sole theatrical releases – especially since the COVID-19 pandemic – production companies and investors are relying on confident decision-making to help improve their generated income. Still, most studios want to keep their releases in theaters, especially due to the pay-per-view income, and consequently, they need to adapt to the changing culture to keep users coming to the theaters [15].

To address these evolving concerns, this work evaluates the efficacy of machine learning models to predict total generated revenue. Using data from the MovieLens 25M dataset [16], the YouTube Data API [17] and the TMDb API [18], total revenue is predicted with regression models; specifically, linear regression, support vector regression, random forest regression, and neural networks. The best of these models yields predictions with R^2 values of 0.733, 0.841, and 0.773 for baseline features, baseline features with trailer data, and pre-release features, respectively. Film production companies can leverage this machine learning resource to determine targeted digital content strategies that can help increase revenue.

II. Literature Review

Sinha and Pan showed that movie income distribution has a power-law tail with Pareto exponent $\alpha = 2$ [19]. This trend is independent of total and opening gross. The movies that flopped opening weekend generally do not leave that trend of being a bomb/flop. Movies defined by "blockbuster" openings still follow a similar trend in which income declines in weeks following opening weekend. Moreover, movies defined by "flop" openings also showed a decline in box office after the opening weekend. However, movies categorized as "sleepers" are the rare trend, where income increases or remains steady for much longer than the time in theaters. These are films who develop a following long after its opening, often generating momentum late after initial release which drives movie DVD sales or ticket sales in theaters months to years after opening. For general purposes, this research showed the box office revenue trend over time. Aside from sleepers, movies that opened with higher box office revenue yielded higher gross profits, while movies that opened to lower box office revenue yielded lower gross profits. Still, profits for both types of movies trail after the first few weeks of opening.

Inspired by other approaches to estimate opening box office from tweets, Mestyan et al. have shown that Wikipedia can be a useful resource for predicting opening weekend box office income [20]. They used a multivariate linear regression model to estimate opening weekend box office revenue using data from the Wikipedia page corresponding to the movie. Revenue and number of theaters showing the movie was gathered from IMDb's Box Office Mojo, and Wikipedia features included the number of users who contributed to the page, the number of edits made by human users on the article, the collaborative rigor (number of edits made by the same user counts as one edit), and the number of views a given page is viewed from inception to time t. They tested their model using subsets of the ensemble of features, but the best coefficient of determination, or R^2 value, came out of the model which included all features; specifically, for Wikipedia features, the number of page views, the number of users who contributed to the page, the rigor, and the number of edits on the page, and for Box Office Mojo features, the number of theaters. This all-inclusive model achieved an R^2 value of 0.77, but it is not stated whether this was achieved on the entire dataset of 312 movies or from a test set. They compare their model to the model based on Twitter features with the same training and test set, and achieve an R^2 value of 0.94, compared to the R^2 value of 0.98 for the twitter model [21]. The evaluation of a model using the R^2 squared value is the most useful outcome in regards to the work that follows in this paper. Achieving an R^2 value greater than 0.9 shows strong accuracy of a model, but with a total dataset comprising only 24 films, this accuracy is difficult to generalize.

Quader et al. worked to predict movie success using 15 features, and predicting a class of profit split into five categories, spanning from "flop" to "blockbuster" [22]. The classes include: (1) profit less than or equal to \$0.5 million (M), (2) profit between \$0.5 M and \$1 M, (3) profit between \$1 M and \$40 M, (4) profit between \$40 M and \$150 M, and (5) profit greater than \$150 M. Features were split into two categories: pre-released features and post-released features. Examples of post-release features included user IMDb ratings and Metacritic Metascore. They predicted box office success with support vector machines (SVM) and neural networks using only pre-released features and with all features in both categories. In evaluation, they monitored exact classification and one-away classification, which is defined as predicting the class, or range, of profit that was next closest to profit range. For SVMs, the best exact classification accuracy on pre-released features was 49.54% (using a Radial Basis Function kernel) and on all features was 56.16% (using a linear kernel). For one-away classification accuracy with SVMs, the best accuracy using pre-released features was 83.44% (using a polynomial kernel) and the best accuracy using all features was 88.87% (using a linear kernel). For neural networks, exact match accuracy with pre-released features was 48.41% and with all features was 58.41%. For neural networks evaluated on one-away accuracy, the model achieved 84.1% accuracy on pre-released features and 89.27% on all features. They determined that the most relevant features for predicting success are budget, number of screens for pre-release features, and IMDb votes. Release month generally leads to a higher class, as does the number of screens showing the movie. This work shows that machine learning models can be successful in predicting range of profit fairly closely, but it is no guarantee that the features used to train these networks alone were sufficient to scale for real-world applications.

Sentiment analysis is another approach to predicting movie success. This involves evaluating the sentiment, whether positive or negative, of a piece of text and assigning a score to it. Sentiment analysis was used on YouTube movie trailer comments by Timani et al for inputs into a simple linear regression model to predict opening revenue, achieving R^2 values of 0.7343, 0.722, and 0.7205 on multiple sentiment analysis index methods [23]. This was also used by Joshi et al. to predict box office revenue based on pre-release reviews, using mean absolute error as a metric for evaluation [24]. Vasu Jain developed a simple metric called PT-NT ratio which represents the total number of positive tweets about a film divided by the total number of negative tweets about a film, and Jain show that this PT-NT ratio has the same trend as the profit ratio [25]. However, Jain's methods were tested on eight films, only half of which were accurately classified by the model. Although these models did not have tremendous accuracy, they reveal that sentiment analysis of reviews may be an important feature moving forward with research to predict revenue.

Evidently, many researchers have approached predicting film success in a variety of ways. However, none have looked at the impact of film trailers on YouTube, one of the most popular platforms for sharing content. This research seeks to fill that vacancy.

III. Methods

A. Data

The data used to train models comes from a variety of sources. First, the MovieLens 25M dataset was downloaded, which is a comma-separated values (CSV) file consisting of 62,423 movies. This data was cleaned to only include films released between the years 2010 and 2019, inclusive, and films with a known *TMDb ID*. Following this sampling and expansion of features, 20,401 films remained. The *TMDb ID* is a data parameter that represents the unique identification number for a film on The Movie Database (TMDb) [5]. TMDb has an associated applications programming interface (API). Using this API, a number of features for a movie can be extracted, including, but not limited to, actors, crew, budget, and reviews. To acquire more information, the TMDb API was used to extract budget, title, vote count, vote average, revenue, runtime, popularity, and overview. The *TMDb ID* from the cleaned Movie Lens 25M dataset was sent as the *movie_id* parameter to the request to the TMDb API and the specified features were extracted. The dependent variable for the models that will be prepared is *revenue*, so any films that did not have a value for revenue were removed from the dataset. Moreover, because of later steps of acquiring movie trailers and limitations to computer hardware, only films that had a vote count greater than or equal to the 80th percentile of films with a revenue parameter present were kept, as this percentile suggested the greatest probability for trailer presence on YouTube. This last cleaning step reduced the number of movies to 677. Then, another request was sent to the TMDb API to get information on release date, crew, and actors. The next step was to acquire information about the movie trailers on YouTube. For this, the YouTube API was used. The YouTube API limits quota cost to 10,000 units per day, and each query costs 101 units. Therefore, queries to the YouTube API were spread across the week of October 18th and October 24th, 2020. Defining accurate query responses can be an ambiguous and difficult task, so a custom similarity score metric was used, specifically, an equally weighted Jaccard index on the video title compared to "{movie name} official trailer" and the Sorensen-Dice formula of the video description compared to the overview provided by TMDb. However, after further evaluation of the query responses from the YouTube API, any video was included in the response, unless it was determined to be a clip from the film, a gameplay trailer, a music video, or an "Honest Trailer" (which is a specific type of popular content from the channel Screen Junkies [26]). Finally, any video that was released more than a year before the movie release date or any video released after the film release date was removed from the dataset. After all of these cleaning steps, the final dataset comprised 650 videos. A visualization of the span of how many films were included from each year is shown in Fig. 1.

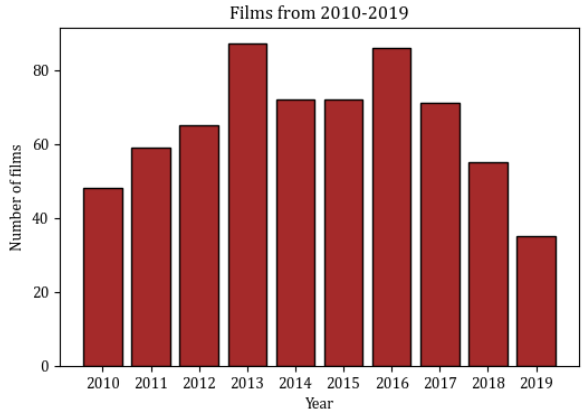

Fig. 1: An overview of the number of films from each year in the decade

The following features were considered for explanatory variables: budget, vote count, vote average, runtime, popularity, view count, like count, dislike count, comment count, and genres. Table I describes what these features represent. Available genres in the dataset include film-noir, mystery, action, imax, sci-fi, romance, musical, comedy, war, crime, children, fantasy, horror, drama, western, adventure, animation, and thriller. Each of these genres was binary encoded in a new column, meaning, for each genre, a value of 1 was present if the film was considered to be classified by this genre according to MovieLens 25M dataset, and a 0 if was not. A distribution of these genres is shown in Figure 2. Different sets of these features were used to predict the revenue of the film, as shown in Table II.

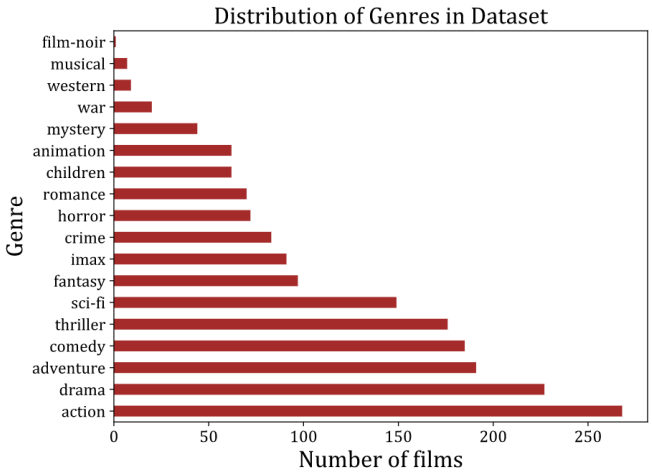

Fig. 2: The distribution of each genre in the dataset

B. Models

The following types of models were compared: linear regression, support vector regression, random forest regression, and neural networks. Each model was trained on every subset of features, as defined in Table II. The details of these models are specified below.

TABLE I: Description of each input feature

Feature	Description
Budget	The movie's budget
Vote Count	Number of ratings for the movie on TMDb
Vote Average	Average vote from all user votes on TMDb for the movie
Runtime	the length, in minutes, of the movie
Popularity	numerical calculation of the movie's popularity on TMDb
View Count	Sum of the number of views from all trailers for the movie
Like Count	Sum of the number of likes from all trailers for the movie
Dislike Count	Sum of the number of dislikes from all trailers for the movie
Comment Count	Sum of the number of comments from all trailers for the movie
Genres	Broken into independent features labeled as film-noir, mystery, action, imax, sci-fi, romance, musical, comedy, war, crime, children, fantasy, horror, drama, western, adventure, animation, and thriller. Denoted by a 1 if the film is labeled as that genre and a 0 if it is not.

TABLE II: Features included in each feature subset

Subset Label	Features Included
Baseline	Budget, Vote Count, Vote Average, Runtime, Popularity, Genres
Trailers	View Count, Like Count, Dislike Count, Comment Count, Budget, Vote Count, Vote Average, Runtime, Popularity, Genres
Pre-Release	Budget, View Count, Like Count, Dislike Count, Comment Count, Runtime, Genres

1) Linear Regression: A linear regression model has the benefit of not being an extremely complex network that can be the simplest method for predicting a continuous value. It is an expansion of the slope-intercept equation, $y = mx + b$, where m and x are arrays instead of scalars. Additionally, because each feature has an associated coefficient, normalization is not required, as the coefficients will scale the features accordingly. However, this model will only be accurate when the relationship between the inputs and the outputs is linear. Ordinary least squares (OLS) was the method used to define the linear regression model.

2) Support Vector Regression: While support vector machines [27] are generally used for classification, support vector regression (SVR) is used for the prediction of continuous real values. For any linear relationship between an output and features, SVR can be understood as a line of best fit, like linear regression, but with a buffer region, called an epsilon-insensitive tube. However, for nonlinear relationships, input data is mapped from the feature space to an implicit higher dimensional space, called the kernel space. It is not an explicit mapping, which requires the calculation of raw feature coordinates in the actual higher dimensional space which becomes computationally insufficient as dimensionality

increases, but it is instead an implicit mapping based on a similarity function, or predefined kernel, between pairs of data. Four kernels were used to train the model: linear, polynomial, sigmoid, and the radial basis function (RBF).

3) Random Forest Regression: Random forest regression (RFR) is an ensemble method that combines the predictions of a specified number of weak learners, decision trees, with the goal of making more accurate predictions and to prevent overfitting. A decision tree is a model in which data is split at each level of the tree into a specified number (often 2) of children nodes. This happens recursively until a node matches a certain criterion, such as max depth or purity. Additionally, decision trees randomly sample the data at each node to define the split at that node, which can help prevent the model from overfitting. However, the decision tree, independently, is considered a weak learner often for at least one of two reasons: because it is designed to be shallow or because it is likely to have too high a variance, meaning the tree is sensitive to the data on which it was trained. Fortunately, combining many of these weak learners together in a forest of decision trees allows the model to be more robust and less prone to overfitting. RFR leverages a technique called bagging, which averages the output of a "bag" of trees and produces an output prediction. The data to train each tree in the forest is generated by bootstrapping, which takes a specified number of samples from the dataset, with replacement, and trains a tree on that data. Under the assumption that the number of samples is an approximation of the distribution, and the data within each sample is not overly correlated, this method can help the ensemble perform very well. Another approach to help achieve similar accuracy in both training and testing is to adjust the depth of each tree. This can prevent each model from learning specifics of the dataset, i.e. overfitting, and make the model more generalizable. However, the optimal depth of a tree is highly dependent on the problem, so many depths are tested to learn how to best generalize data.

For training, the bootstrapping parameter for sample size was varied to analyze its effect on the accuracy of each model. Moreover, changing the max depth of each tree tended to yield different results. In total, a random forest was built from a number of trees in the range [20, 95] with a step size of 5 trees. Mean squared error was used to measure the quality at each splitting node for every tree. The exact parameters values for RFR model are detailed in Section IV.

4) Neural Networks: Neural networks, or multilayer perceptrons (MLP), are one of the most popular and high-performing machine learning models because they can learn any continuous function and work remarkably well on data with nonlinear relationships. A MLP represents an artificial network similar to the structure of human neural activity, as first introduced by [28]. To generate an accurate artificial neural network, a model is first built with a predefined structure: an input layer with nodes equal to the number of features, hidden layers with a specified number of nodes in each layer, and an output layer. Connections between nodes in successive layers are known as weights, and learning occurs when the network adjusts these weights in order to better represent the underlying mapping from inputs to output. The most common MLP structure is a fully connected network, in which every node in a layer is connected to every node in the next layer. For the models that were tested, the input layer always consisted of 27 nodes, which is equal to the number of input features, and the output layer always had one node for regression. Network weights were initialized randomly using a normal distribution with zero mean and standard deviation equal to 0.05.

Training a MLP is a two-step process: (1) the input data is fed forward through the neural network, where each internal node computes the weighted sum of its inputs and then applies an activation function; once the output is reached, the loss is calculated; and (2) the loss is used to update all of the weights in the network via a method called backpropagation. This process is repeated until a stopping criterion is reached, such as a specified number of epochs, a stagnation in performance improvement, or an indication of the onset of overfitting.

For the MLPs that were trained on this problem, each model can be described as follows:

- All weights are initially sampled from a normal distribution with mean 0 and standard deviation 0.05
- All hidden layers use the rectified linear unit (ReLU) activation function
- The final output layer activation function is linear
- The Adam optimizer is used as the loss function with learning rate 0.001
- Training batch size is 50 samples
- Training epochs is 500
- Best weights, as defined by the smallest loss on validation data, were saved for each model

In total, 24 models were trained, 19 models with normalized inputs and outputs and five models without normalized inputs and outputs. The structure of the 5 models without normalized data was defined by the engineer. The structure of one of the models with normalized data was defined by the engineer, and the rest of the models were pseudo-randomly generated, where the number of hidden layers was sampled from the range [2, 6] and the number of nodes at each layer was sampled from the range [10, 5000] with a step size of 10.

C. Training

The entire dataset of 650 movies was split into training and testing subsets in an 80-20 split, meaning 80% of films (520) were used to train each model, and 20% of films (130) were used to test the model. This split was randomly generated with a constant seed, so all of the testing and training data was consistent across each network.

D. Normalization

For SVR and MLP models, data was normalized. Normalization is required for SVR so that features with larger values (e.g. budget, vote count) do not trump smaller values like genre or day of week. For MLPs, normalization is not technically necessary, and models could still be effectively trained on simpler models, but training becomes extremely difficult on

TABLE III: R^2 Values Results for each features subset per model

Model	Baseline	Trailers	Pre-Release
SVR (Linear)	0.710	0.760	0.691
SVR (RBF)	0.715	0.735	0.634
Linear Regression	0.701	0.823	0.549
Random Forest (25 Trees)	0.695	0.824	**0.773**
Random Forest (40 Trees)	0.709	**0.841**	0.760
Random Forest (95 Trees)	**0.733**	0.833	0.768
Neural Network	0.693	0.818	0.680

larger values. This is due to the random initialization of all weights, which are initialized at very small values. Without normalization, the increased complexity of the model will lead to output values so much larger than the intended output that the loss function begins to diverge because certain parameters are given unequal priority in gradient descent, meaning the network is unable to effectively "learn." The normalization approach that is used for each model is the Standard Scalar, which transforms the mean to 0 and scales to the unit variance. The standard scalar is applied to all input features and another standard scalar is applied to the output. In order to maintain the validity of the testing data, the fitting of the scalar is only performed on the training data. However, data transformation is performed on both the training and testing data based on the standard scalar for inputs and the standard scalar for outputs.

IV. Results

This section details the results of the regression models on each subset. Each model is evaluated based on the coefficient of determination, or R^2. This metric evaluates how closely predicted values match the actual value. If the R^2 value is 1, that means the network perfectly predicted the data. Model results are shown in Table III.

A. Linear Regression

Results suggest that there is indeed a linear correlation between each subset of features and revenue. The R^2 value on a linear regression model for feature subsets of baseline, trailers, and pre-release was 0.701, 0.823, and 0.549, respectively. While results for the baseline and trailer subsets suggest stronger correlation, pre-release features suggest that revenue predictions does not follow as strict a linear trend.

B. Support Vector Regression

SVR did not appear to be an substantially effective model for predicting revenue based on the features in the dataset. Using a linear kernel, the R^2 values of baseline, trailers, and pre-release were 0.710, 0.760, and 0.691, respectively. The success of this kernel reflects the results from linear regression, though linear regression still under-performs on the pre-release subset of features. Using the RBF kernel, the R^2 values were 0.715, 0.735, and 0.634, respectively. However, the use of a polynomial kernel and sigmoid kernel resulted in worse predictions than any model at all. The R^2 values of both of these kernels were negative, and after further investigation, these models appear very susceptible to outliers, specifically the film *Ghostbusters (2016)*, whose YouTube trailer data consists of unreasonably high view count for a film that only made $229 Million. Evidently, there are features in the dataset that skew the accuracy of the model, but the kernels also may not be effective in representing relationships in data in higher dimensional space.

C. Random Forest Regression

RFR outperforms all other models in all feature subsets, but only by a limited fraction. Most R^2 values circulated the R^2 value of 0.7 for baseline features, but the best performing forest on this feature subset came from a forest of 95 trees with an R^2 value of 0.733. A plot of this model's predictions is shown in Figure 3. The best R^2 for the trailers subset was 0.841 with a forest of 40 trees (Figure 4). Finally, on the pre-release feature subset, the best forest consisted of 25 trees with an R^2 value of 0.773 (Figure 5). These trends are likely due to the bootstrap sample size and max depth parameters that trained and defined each tree in the forest. For each of the top-performing forests, 80% of training data was bootstrapped, meaning each tree was trained on a total 416 movie samples. Moreover, in forests with fewer trees, a max depth was set for each tree with the goal of preventing overfitting. A max depth of 8 was set for the 25 tree forest that yielded the best results for pre-release features. Forests trained on other feature subset performed best when trees were not limited by depth.

D. Neural Networks

Random testing showed that the model learns fairly quickly in training. Of the 19 total models, the most successful models consisted of 4 hidden layers. They achieved R^2 values of 0.693, 0.818, and 0.680 on the baseline, trailers, and pre-release feature subsets, respectively. However, each model overfit after only a few epochs, so the best weights were saved around only 30 epochs. One issue with the neural network models is that some predictions trend below 0. This reflects the same trend for linear regression and SVR, but predicting impossible values is less than ideal in real-world scenarios.

Neural networks were also used for revenue class predictions, similar to the approach in [22], but results varied very little comparatively. Using a small network with an output layer of 5 neurons, the model achieved 54% class accuracy and 85% one-away accuracy, comparable to the the model used in [22] at 48.41% and 84.10% in the same respective categories. However, exact class accuracy is more important for production, so perfect class prediction achieved on only about half of the films in testing does little to aid production decisions.

V. Limitations

First, it cannot be concluded that popularity of a particular movie contributed to the success of the movie trailers. This limitation is due to the COVID-19 pandemic, which prevented data collection through the year of 2020, specifically, the view-, like-, dislike- and comment count of YouTube trailers.

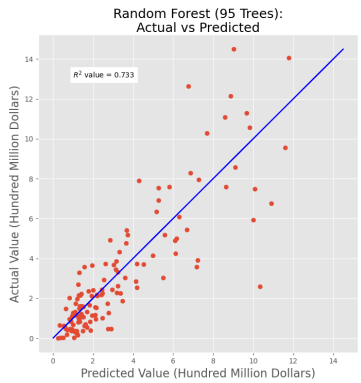

Fig. 3: Results of the random forest regression model on baseline features with 95 trees.

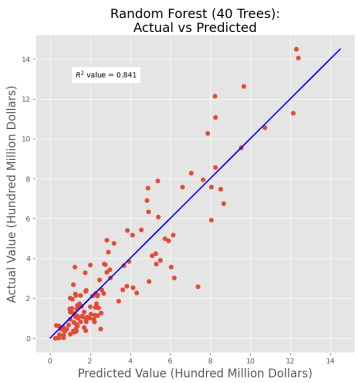

Fig. 4: Results of the random forest regression model on trailer features with 40 trees.

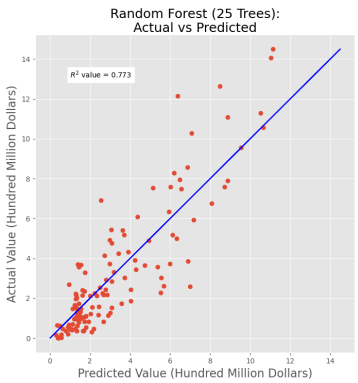

Fig. 5: Results of the random forest regression model on pre-release features with 25 trees.

Many movies that were expected to be released in 2020 were rescheduled or released on streaming services up front, so collecting opening box-office data and the YouTube data directly at various intervals before the release of a film was impractical and at times impossible. Further research can be more confident in results if these original intentions and data collection processes are pursued. Second, though YouTube data is shown to improve the accuracy of predictions, these results may not significantly help production companies direct their funds, as the most significant improvements result from post-release features. Third, by use of the ordinary least squares model, multicollinearity is not eliminated. The feature collinearity heatmap is shown in Figure 6. This heatmap shows that YouTube features view count, like count, and comment count, are often "describing" the same information. This is not an issue for SVRs, RVRs, and MLPs, as the models presumably "learn" that these features describe the same information in training, but in more practical applications, it suggests that not all of these features are necessary to store in memory for training. Finally, a more hands-on approach to video trailer data collection can confirm the consistency of the data itself. This research had no opportunities for cross-referencing the data collected from the automated process in collection from the YouTube Data API beyond string similarity metrics Jaccard index and Sorensen-Dice formulae, which as mentioned, were eventually removed from consideration for inclusion of videos. It could also be argued that the biases present in this data suggest skewed results, and resampling from different production companies and lower budget films may provide improved confidence in the applications of these models.

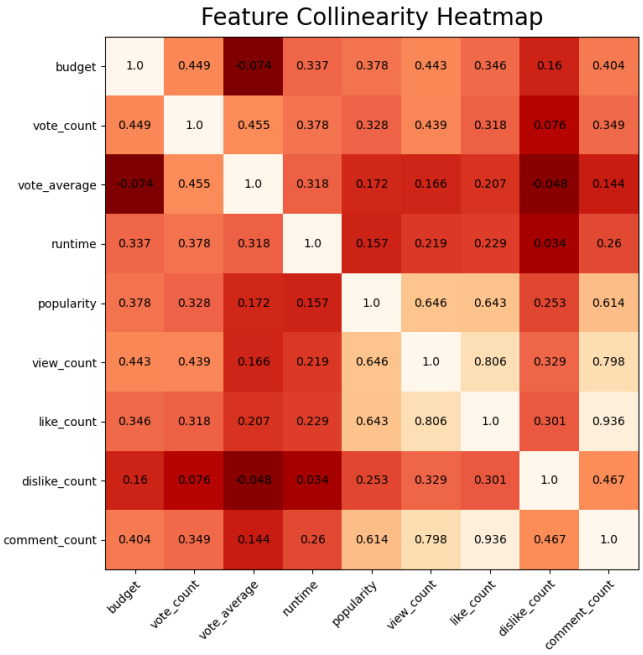

Fig. 6: Heatmap of the relationships between features

VI. Conclusion

Results show there is strong correlation between subsets of movie features in this custom dataset and film revenue. Four different models were used to predict revenue on these subsets: linear regression, support vector regression, random forest regression, and neural networks. Of these models, random forest regression yielded the best performance. On baseline features, the best R^2 value achieved was 0.733. On trailer features, the best R^2 value achieved was 0.841. Finally, on pre-release features, the best R^2 value achieved was 0.773.

The best model for each subset of features is a random forest with 25 trees for pre-release features, 40 trees for the baseline data with inclusion of YouTube trailer data, and 95 trees for the baseline features. Clearly, the addition of trailer data improves predictions, increasing the R^2 value by nearly 15 percent. However, when considered within the context of strictly pre-release features, the addition of YouTube trailers matches the standards of other research using Wikipedia [20], and better than research using YouTube comments [23].Though these

results show promise, issues are still present as mentioned in Section V.

Ethical Considerations

Though this research strictly identifies correlation and not causation, the link between trailers and revenue success can often be misunderstood, and inappropriate misunderstandings may have significant consequences for the future of the medium of film. For research carrying forward, it is important to balance the business aspirations with the cultural dynamics that are ingrained in film and moviegoers. Should research rely strictly on data that appeals to certain demographics, AI runs the risk of proliferating unethical and discriminatory practices. For example, should research investigate which actors to include in an upcoming film to increase the probability of Oscar nominations, it is important to evaluate the data on which a particular algorithm is trained. The Academy of Motion Picture Arts and Sciences, though moving in the right direction, is known to lack significant diversity in representation in nominations and wins for Oscars [29]. If the last 40 years were used as training data to evaluate what kind of actor should be included, the results are most certainly to favor white males. Recognizing the ethical implications of this work is more important than any business benefit an algorithm may bring.

ACKNOWLEDGMENTS

I want to thank my advisor, Dr. Eicholtz, for continuing encouragement and support throughout this exploration. I want to also thank the Honors Program and the Department of Computer Science at Florida Southern College for the countless learning opportunities and supportive environment.

REFERENCES

[1] "Domestic Yearly Box Office." *Box Office Mojo*. Available: https://www.boxofficemojo.com/year/. Accessed: Nov. 22, 2020.
[2] D. Reinsel, J. Gantz, and J. Rydning, "The Digitization of the World from Edge to Core," p. 28, Nov. 2018.
[3] T. Spangler and T. Spangler, "Twitter Users Hit Theaters More Than Average Moviegoers: Study," Apr. 2015.
[4] "IMDb." https://www.imdb.com.
[5] "The Movie Database." https://www.themoviedb.org.
[6] "Metacritic." https://www.metacritic.com.
[7] "Rotten Tomatoes." https://www.rottentomatoes.com.
[8] "Letterboxd." https://letterboxd.com/.
[9] Dr. Pepper [@drpepper]. What makes the @Marvel's @Avengers a one of a kind team? Use #OneofaKindAvengers to unlock an Age of Ultron video., *Twitter*, Apr. 8, 2015 [Online]. Available: https://twitter.com/drpepper/status/585857736673513473, Accessed Nov. 25, 2020.
[10] Warner Bros. Pictures. *YouTube*. Available: https://www.youtube.com/user/WarnerBrosPictures. Accessed: Nov. 25, 2020.
[11] Pixar. *YouTube*. Available: https://www.youtube.com/user/DisneyPixar. Accessed: Nov. 25, 2020.
[12] Movieclips Trailers. *YouTube*. Available: https://www.youtube.com/user/movieclipsTRAILERS. Accessed: Nov. 25, 2020.
[13] Movieclips. *YouTube*. Available: https://www.youtube.com/user/movieclips. Accessed: Nov. 25, 2020.
[14] B. Lang and B. Lang, "The Reckoning: Why the Movie Business Is in Big Trouble," Mar. 2017.
[15] S. Whitten, "Why Hollywood is sticking with movie theaters and only a few films are heading to streaming," Apr. 2020. Section: Entertainment.
[16] "MovieLens," Sept. 2013. https://grouplens.org/datasets/movielens/.
[17] "YouTube Data API | Google Developers." https://developers.google.com/youtube/v3.
[18] "The Movie Database API." https://developers.themoviedb.org/3.
[19] S. Sinha and R. K. Pan, "Blockbusters, Bombs and Sleepers: The Income Distribution of Movies," Apr. 2005.
[20] M. Mestyán, T. Yasseri, and J. Kertész, "Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data," *PloS one*, vol. 8, p. e71226, Aug. 2013.
[21] S. Asur and B. A. Huberman, "Predicting the future with social media," in *In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '10*, pp. 492–499, IEEE Computer Society, 2010.
[22] N. Quader, M. O. Gani, D. Chaki, and M. H. Ali, "A machine learning approach to predict movie box-office success," in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pp. 1–7, Dec. 2017.
[23] H. Timani, P. Shah, and M. Joshi, "Predicting Success of a Movie from Youtube Trailer Comments using Sentiment Analysis," in *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 584–586, Mar. 2019.
[24] M. Joshi, D. Das, K. Gimpel, and N. A. Smith, "Movie Reviews and Revenues: An Experiment in Text Regression," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (Los Angeles, California), pp. 293–296, Association for Computational Linguistics, June 2010.
[25] V. Jain, "Prediction of Movie Success using Sentiment Analysis of Tweets," *JSCSE*, vol. 3, no. 3, pp. 308–313, 2013.
[26] Screen Junkies. *YouTube*. Available: https://www.youtube.com/channel/UCOpcACMWblDls9Z6GERVi1A. Accessed: Nov. 25, 2020.
[27] W. S. Noble, "What is a support vector machine?," *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.
[28] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, Dec. 1943.
[29] J. Sarkisian, "A complete breakdown of the Oscars diversity problem over the past decade." Feb. 2, 2021. Accessed on: April 20, 2020 [Online]. Available: https://www.insider.com/oscars-diversity-problem-race-gender-numbers-academy-awards-2021-2.