Movie Analysis: Recommendation Engine and Revenue Predictions Report

Dharini Baskaran[*]
Vignesh Kumar Karthikeyan[*]
Sabina Adhikari[*]
dharini.baskaran@colorado.edu
vigneshKumar.karthikeyan@colorado.edu
sabina.adhikari@colorado.edu
University of Colorado Boulder
Boulder, Colorado, USA

1 ABSTRACT

In this work, we explore two datasets from GroupLens; one to build a recommendation engine, illustrating the algorithms used by many digital companies and the second to build a prediction model of revenue generation. To build a recommendation engine, we focus on a few attributes of movies, namely, the title, movieId and genre, and ratings of the movies. A recommendation system based on collaborative filtering which takes account of activities of other users and a recommendation system based on hybrid approach that takes account of the user's history and genre preference are developed. To build a prediction model, we focus on the second dataset, which has information about attributes such as revenue, budget, popularity, vote_counts, and vote_average. A subset of dataset is used as a training set to fit a linear regression model, which is later used as the prediction model for revenues. The performance of the linear regression model is compared with the performance of other regressor models such as kNeighbors regressor and Random Forest regressor.

2 INTRODUCTION AND MOTIVATION

With the abundance of data collection, data mining methods and machine learning have been handy to make predictions and recommendations, which filter out useful information relevant for users and various stakeholders. Recommendation system is a type of content filtering system, which suggests items related to search items and history of the items. Recommendation system sorts out relevant information, in the presence of overwhelming data these days. Similarly, prediction methods help to make forecasts based on historical data.

Both prediction and recommendation methods have applications in many industries like e-commerce, entertainment and banking. Digital platforms like Netflix, Prime Video, Spotify, online advertisements, shopping or job recommendation sites follow recommendation systems to make suggestions. This recommendation system could be useful for viewers to understand the choices being offered by different companies. On the other hand, recommendation methods are significant to companies to promote carefully catered products to its customers based on their previous usage and preferences. Prediction methods are appropriate and effective for any financial companies, health companies and entertainment businesses to forecast earnings or sales or number of customers. In addition, prediction method could be instrumental to identify factors or attributes needed to be modified to achieve certain goals or to support various business decision-making activities.

As our group loves movies and there is vast dataset on movies, we explored two datasets from GroupLens and built a recommendation systems and prediction methods. For the recommendation engines, we used the collaborative filtering and hybrid filtering. For the revenue prediction model, we developed a linear regression on a subset of attributes and compared its performance with other methods like Random Forest regressor and k-Neighbors regressor.

This report is organized as follows. In Sec [3], we present some ongoing work in the fields of recommendation engines and prediction models. In Sec [4], we explain the data preprocessing process for both datasets in detail and present some of the issues we ran into. In

[*] All three authors contributed equally to this work.

Sec [5], we provide a detailed description of data visualization process, recommendation engines and prediction model. In Sec [6], we present the results of data analysis, recommendation engine and prediction model, and discuss the performance of recommendation engine and prediction model. We present the conclusion and discuss the limitations of our models, and possible future work in Sec [7].

3 RELATED WORK

There have been many prior works done on both recommendation systems and prediction methods. Recommendation systems were first introduced in the 1990s. The concept of **Collaborative Filtering** was introduced in 1992 which was experimentally applied to personal emails and information filtering [3]. Personal recommendations are present everywhere, leading to growing interest in exploration of different recommendation systems and their effectiveness.

In [1], the authors have realised the bias and unfairness in existing recommendation systems, They focused on their paper to find the anomaly's origin. And this is achieved by **Soft Matrix Factorization (SoftMF)** on MovieLens dataset, which tries to balance the predictions of different types of users to reduce the present inequality.

In [4], the authors review various recommendation systems like **collaborative filtering, content-based filtering, context-based filtering** and **hybrid filtering**. The authors also present various machine learning algorithms like K-Means Clustering and Principal Component Analysis and measure the model accuracy.

In [6], the authors use the collaborative filtering with three different user similarity measures: **Cosine similarity, Correlation based similiarity** and **Euclidian similarity** to predict ratings of various movies.

In [9], the authors introduce the novel **k-clique** method on social networks to improve the efficiency of collaborative filtering.

In [8], the authors have discussed various existing methods of recommendations system in current practice. The paper discusses on improving the recommendation system's performance and agility through collaborative filtering method. To achieve this, **K-means, Content-Based recommendation** and **SVD** methods were deployed. They calculated mean and cross validation metrics to evaluate and show that their approach indeed results in increased performance.

There have been ample number of studies done on prediction methods too. In [7], the authors conduct performance of seven different machine learning methods to predict profit value of movies and conclude that **Multilayer Perceptron Neural Network** gives the best output.

In [5], the authors propose the **Support Vector Method (SVM)**-based machine learning method to use economic factors to predict movie box-revenues of China and the US. They also compare the SVM method with random forest based and neural network based machine learning method.

The paper [2], tries to find whether multi-model or single-model prediction system yields better results. This is tackled because the revenue prediction on box office have always shown conflicting results as different data is used. The main sources are either movie reviews or metadata. This experiment proves that using metadata alone, we can predict the box office revenue. This is done utilizing **EM(Expectation Maximization)** algorithm.

Hence, both recommendation methods and prediction methods are widely used to analyse the movie datasets and many other applications. Recently many works on comparison of different methods with various modifications are being explored to deal with issues like size and sparsity of datasets, and efficiency of different algorithms.

4 DATA PREPROCESSING

4.1 Dataset 1

To develop the recommendation model, we used the MovieLens dataset available at https://www.kaggle.com/datasets/shubhammehta21/movie-lens-small-latest-dataset. We focused on the files *movies.csv* and *ratings.csv*, which contains information about 9742 movies. The movies file contains data about movieId, movie's title and the genre it belongs to. The ratings file showcases information about userId, movieId, the movie's rating by the particular user and the timestamp.

During the preprocessing phase, our focus was on optimizing data quality and scalability, particularly in the "ratings.csv" file. Notably, we encountered instances of missing ratings denoted as "NaN," suggesting that certain users had not provided ratings for specific movies. To ensure robustness in our modeling efforts, we chose to replace these "NaN" values with zeros, implicitly indicating that users had not yet rated those movies. By substituting these instances with zeros, we ensured a standardized representation, explicitly indicating that users had not yet rated specific movies. This approach bolstered the integrity and consistency of our dataset, laying the groundwork for robust modeling efforts. This strategic decision aimed at maintaining data integrity and consistency, laying a solid foundation for subsequent analyses.

Another pivotal consideration revolved around the scalability of our dataset. While "movies.csv" comprised 9,742 rows, "ratings.csv" boasted 100,836 rows. Recognizing the need for efficient data retrieval in preparation for collaborative filtering, we implemented data indexing. Unlike conventional approaches, we refrained from data pruning due to the dataset's streamlined nature, containing only essential features. This choice not only streamlined our modeling process but also preserved crucial data elements, contributing to the overall efficiency of our analyses. This choice not only optimized efficiency but also retained essential features, contributing to the scalability and effectiveness of our recommendation system.

In the realm of recommendation systems, sparsity emerged as a significant concern, stemming from the imbalance between actual ratings and the vast number of potential user-movie pairs. To tackle this challenge, we embraced a hybrid model approach that seamlessly integrated user history and preferences into our recommendations. By leveraging users' viewing habits and past interactions, our recommendations gained depth and relevance. Additionally, to address sparsity directly, we tailored our collaborative filtering model to yield top-N recommendations. This adaptive strategy offered users a curated selection of potentially appealing movies, acknowledging the sparse nature of the dataset and enhancing the likelihood of providing valuable recommendations.

4.2 Dataset 2

To develop the prediction model, we could not use Dataset 1 as the file lacked information about important variables like revenue, budget, and popularity. We used the dataset available at https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset. We focused on the file *movies_metadata.csv*, which contains information about 45436 unique movies. The file contains 23 attributes; 2 binary variables: adult, video; 13 categorical variables: belongs_to_collection, genres, id, imdb_id, original_title, overview, poster_path, production_companies, production_countries, spoken_languages, status, tagline, title; 7 numerical variables: budget, popularity, release_date, vote_average, revenue, runtime, vote_count.

In the data file *m*ovies_metadata.csv, we observed many missing data and varied data format for different variables. Hence we wanted to ensure that the missing values were addressed and different attributes had information presented in a consistent and standard way. Our first step in data cleaning process was to remove all non-released movies. The status category had seven classes: "Released", "Rumored", "Post Production", "In Production", "Planned", "Canceled" and "NaN/nan". As predicting revenues of unreleased movies seemed redundant, we removed entries of such movies, leaving us with 45014 movies. We then generated a few catplots and barplots to observe the relationship and any correlation between independent variables and revenue. However as some nominal variables such as "genres" and "spoken_languages" were presented in dictionary type string format, we had to change the format to extract essential information about such variables. This step was imperative to maintain consistency and avoid missing any any pertinent information. One major issue we ran into was retrieving complete information about spoken languages and original language of the movies as languages other than English used their original script. Hence we were unable to decipher the actual name of most languages.

Next while developing a regression model, we deleted the categorical variables as we cannot find correlation between nominal variables and the numerical variables. We changed all numerical entries to floating values to maintain consistency and make sure that all numerical attributes are considered for the correlation matrix. Moreover we realized that most entries for budget and revenues were 0. Budget = 0 seemed to be an error and Revenue = 0 seemed unreasonable, hence we removed all such entries. We deemed that filling in such entries with means or medians or any similar values inappropriate, as majority of entries were 0. After removing such entries, we were left with 5377 movies. Any missing entries (usually in the form of blank space of entry NaN) in other numerical variables of remaining 5377 movies were filled in with a 0. Therefore we used these 5377 movies as the training and testing dataset for our regression model.

5 METHODOLOGY

5.1 Data Visualisation

As an integral part of our statistical analysis, we undertook a comprehensive exploration of our dataset, which allowed us to glean valuable insights. To gain a deeper understanding of the data distribution and patterns, we meticulously curated a suite of visual, graphical representations. One of the fundamental aspects of our study revolved around the determination of the average ratings for each genre by each user, thereby shedding light on the preferences and tendencies of our dataset's users. This would then lead our model to give the appropriate and relevant recommendations.

In parallel, we deduced the lowest and highest rated movies, offering insights into the extremities of the rating spectrum within our dataset. Also, studying how the ratings were spread out, helped us understand how often different rating values were used. This was visually seen through a wordCloud.

We also delved into the popularity of various genres, exploring how the user community engaged with and embraced different thematic categories. We additionally explored to see the highest rated genre on an annual basis, to capture the temporal changes in user preferences.

Further we checked the relationship between the average rating assigned to a movie and the quantity of ratings it received from individual users. This shed light on whether the popularity of a movie, as evidenced by a higher volume of ratings, actually corresponded to an elevated average rating or not.

For the prediction model, we created a few catplots and jointplots to understand any underlying relationships between the independent variables and revenue, thereby assessing the relevance of possible independent variables in the prediction model.

5.2 Recommendation Engine

5.2.1 *Collaborative Filtering.*

In our collaborative effort to develop a movie recommendation system, we harnessed the power of **Collaborative Filtering** technique to provide users with personalized movie suggestions that resonate with their viewing preferences. We began with the usage of movie and rating data from CSV files, managed by the Pandas library. Crafting a user-movie matrix, we employed the pivot function, filling in missing ratings with zeros to signify movies that users hadn't interacted with.

A critical step in optimizing computational efficiency involves the conversion of the user-movie matrix into a sparse matrix, which we accomplished through the csr-matrix function from the scipy.sparse library. This strategic choice mitigates memory consumption, particularly advantageous for large-scale datasets. The collaborative filtering model is then constructed utilizing the k-NN algorithm, specifically configured with cosine similarity as the distance metric. This method provides a quantifiable measure of similarity in user behavior, forming the foundation for effective movie recommendations.

Functionally, the get-movie-recommendation function takes a movie name as input and outputs a DataFrame containing the top movie recommendations based on user behavior similarity. We took an illustrative example involving the movie "Momento" that showcases the system's efficacy, revealing the most highly recommended movies for users who have exhibited comparable tastes.

Recommendations can be dynamically tailored to cater to user preferences, fostering a more engaging and personalized viewing experience. Anticipating future enhancements, the system could benefit from the incorporation of rigorous evaluation metrics to assess its predictive performance. Further exploration of diverse normalization techniques and similarity metrics could yield refinements, enhancing the precision of recommendations. Considerations for an improved user interface, catering to user preferences, and addressing the cold start problem for new movies or users are pivotal for sustained user satisfaction.

5.2.2 *Hybrid-Based Approach.*

Adding to the collaborative filtering model, a recommendation model with **Hybrid-Based** approach using content-filtering and content-based recommendation is also developed. Content filtering involves the categorization and organization of content based on specific attributes or characteristics. In the context of our project, this is "genre" element. This model takes up the previous history of the user and recommends him the top 10 similar movies from our database. All content-based recommendation systems leverage user preferences and historical data to suggest items that are similar to what our user has already shown interest in. This approach is particularly useful in our approach to movie recommendations as it allows for personalized suggestions that align with the individual user's tastes. The movie industry also stands to benefit significantly from these systems by fostering user engagement, increasing content consumption, and ultimately driving revenue. By tailoring recommendations to our users' specific preferences, content filtering and content-based recommendation systems contribute to a more satisfying and enjoyable viewing experience, thus leading to both user and retailer satisfaction. Additionally, our systems complement the prediction model by offering a more nuanced understanding of user preferences, enhancing the overall accuracy and relevance of recommendations. The primary motivation behind employing these systems is to create a more personalized and user-centric content discovery process, thereby fostering a stronger connection between viewers and the vast array of available movies.

This differs from the collaborative-filtering model by implementing TF-IDF method. It stands for **Term Frequency - Inverse Document Frequency**. This technique assigns weights to words based on their importance in describing a movie, considering the frequency of terms across the entire dataset. TF-IDF weighting aids in capturing the unique features of each movie and contributes to the accuracy of the recommendation system by distinguishing key elements. This practice complements our wordCloud and helps future moviemakers to come up with a revenue-generating title.

After we import our dataset of 2 files, we create a new dataframe by combining both the files such that each row represents a movieId and each column represents an userId. This dataframe is very easy to visualise the data and interpret the significance of each genre in the user's watch-list. We then calculated the tf-idf value for each user which will determine the weight of each term(genre) within each movie. Now based on the weights of each genre, our model predicts the similar movies on priority basis. Then when an userId is given as input to our model, it will process all the above steps and showcases the top 10 similar movies to watch as our recommendation list.

Our collaborative filtering model recommends movies based on the ratings from other users who have watched the similar movies. But it doesn't take into consideration of the user's interest. This is what is resolved in the next hybrid model with content-filtering and tf-idf technique.

5.3 Prediction Model

In this section, we explain the implementation of the revenue prediction model. After the initial exploration of *movies_metadata.csv* to standardize the data and understand any relationships between different attributes and revenue, we developed a revenue prediction model using multivariable linear regression model. Linear regression model is an approach to model the relationship between different variables by fitting in a linear equation to observed data. For our study, revenue is a dependent variable and attributes budget, popularity, runtime, vote_count and vote_average are the independent variables. The prediction of a dependent variable is determined by the independent variables.

To develop a prediction model, we first splitted our cleaned up data of 5377 movies into training set and testing set using the *train_test_split* function from *sklearn model selection* package. This guarantees that our data is divided into random train and test subsets without shuffle, determined by the assigned test proportion. We then created a few joint plots of our training dataset using the seaborn package to better understand the relationship between each independent variable and revenue variable and assess whether the linear regression model is appropriate for prediction.

We then fitted a *LinearRegression model* from *sklearn* package to our training dataset. This function outputs the coefficients for each independent variable and intercept, which minimize the residual sum of squares between the observed targets in the datasets and targets predicted by the linear approximation. The linear approximation model is then used as a predictor model for testing dataset, where we input the independent variables and get revenue predictions as the output. We played with different test sizes, and different subset of variables as the input variables to find the most appropriate linear regression model.

Although the focus of our prediction model is the linear regression model, we conducted a brief exploration of regression models such as *KNeighborsRegressor* and *RandomForestRegressor* algorithms on *sklearn* package to examine the performance of linear regression model in comparison with other classification models. While

KNeighborsRegressor makes predictions based on regression of closest neighbors of the testing data, *RandomForestRegressor* fits a number of classifying decision trees on various subsamples of dataset and uses averaging to improve accuracy of the predictions.

6 RESULTS

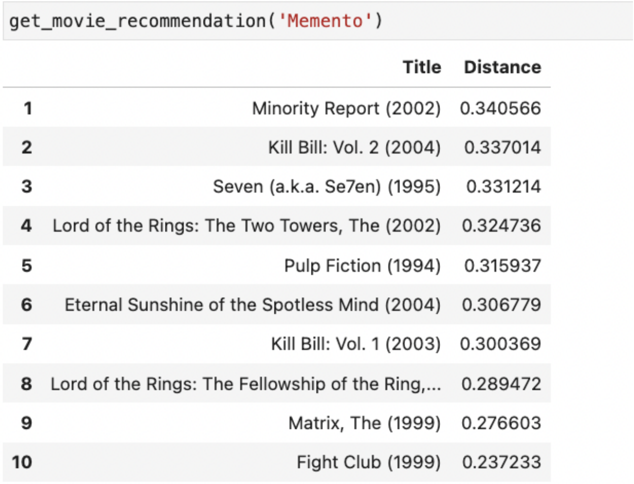
6.1 Recommendation Engine

6.1.1 Data Analysis.
As an integral part of our statistical analysis, we undertook a comprehensive exploration of our Dataset 1, which allowed us to glean valuable insights. To gain a deeper understanding of the data distribution and patterns, we meticulously curated a suite of visual, graphical representations. One of the fundamental aspects of our study revolved around the determination of the average ratings for each genre by each user, thereby shedding light on the preferences and tendencies of our dataset's users. This would then lead our model to give the appropriate and relevant recommendations.

In parallel, we deduced the lowest and highest rated movies, offering insights into the extremities of the rating spectrum within our dataset. Also, studying how the ratings were spread out, helped us understand how often different rating values were used. This was visually seen through a wordCloud.

We also delved into the popularity of various genres, exploring how the user community engaged with and embraced different thematic categories. We additionally explored to see the highest rated genre on an annual basis, to capture the temporal changes in user preferences.

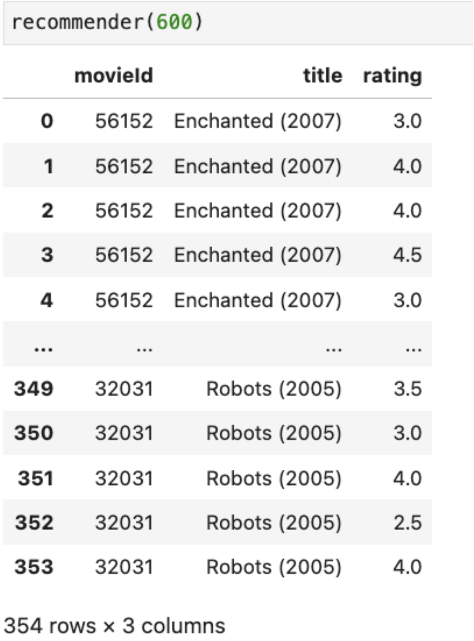
Further we checked the relationship between the average rating assigned to a movie and the quantity of ratings it received from individual users. This shed light on whether the popularity of a movie, as evidenced by a higher volume of ratings, actually corresponded to an elevated average rating or not.



```
get_movie_recommendation('Memento')
```

	Title	Distance
1	Minority Report (2002)	0.340566
2	Kill Bill: Vol. 2 (2004)	0.337014
3	Seven (a.k.a. Se7en) (1995)	0.331214
4	Lord of the Rings: The Two Towers, The (2002)	0.324736
5	Pulp Fiction (1994)	0.315937
6	Eternal Sunshine of the Spotless Mind (2004)	0.306779
7	Kill Bill: Vol. 1 (2003)	0.300369
8	Lord of the Rings: The Fellowship of the Ring,...	0.289472
9	Matrix, The (1999)	0.276603
10	Fight Club (1999)	0.237233

Figure 1: Collaborative filtering model output for "Memento"



```
recommender(600)
```

	movieId	title	rating
0	56152	Enchanted (2007)	3.0
1	56152	Enchanted (2007)	4.0
2	56152	Enchanted (2007)	4.0
3	56152	Enchanted (2007)	4.5
4	56152	Enchanted (2007)	3.0
...	...	...	...
349	32031	Robots (2005)	3.5
350	32031	Robots (2005)	3.0
351	32031	Robots (2005)	4.0
352	32031	Robots (2005)	2.5
353	32031	Robots (2005)	4.0

354 rows × 3 columns

Figure 2: Hybrid model output for User 600

6.1.2 Recommendation Results and Evaluation.
When we look at how well our movie recommendation system is doing, we need to keep in mind that regular metrics might not work perfectly for our unique method. Our system is different from usual models because we used a person's past movie-watching data to suggest movies they have not watched. The strength of our system is that it understands how users behave, finding good suggestions based on what they liked before. It's all about giving personalized ideas connected to what a person has watched, and our system does a great job of finding movies that fit a user's taste. Even if the usual metrics might not show this perfectly, the real measure of success is how happy users are, and our system is really good at making movie recommendations that people enjoy.

6.2 Prediction Model

6.2.1 Data Analysis.

The data analysis of Dataset 2 was done with a aim of understanding the relationships between the independent variables and the revenue variable. Catplots with binary attributes on the horizontal axis and revenue on the vertical axis are useful to visualize any underlying correlations. In Fig. [3], 1 on belongs_to_collections means movies belong to a franchise, whereas 0 means otherwise. We note that a few movies belonging to a collection make more revenue, but this observation does not seem significant as most movies are clustered at the bottom, implying smaller revenue generation for majority of films. We have included similar plot for attribute: adult (whether movie belongs to adult category), video video (whether the movie is a video) versus revenue on jupyter notebook, but the results were not significant to include here. On the other hand, if a movie had a tagline, it tend to make more revenue than movie not having a tagline, which is lucid in a catplot on jupyter notebook. In Fig. [4], we have a barplot of five highest average revenue with respect to the original language of movie. *En* denotes English movies, and they make more revenue on average, which could be due to English being a global language and larger market. *Te, Ta, Hi* are languages from India, although we were unable to read their names as they did not use latin alphabets. Similarly *zh* is a language from China. It is interesting to see three languages of India making the highest average revenues.

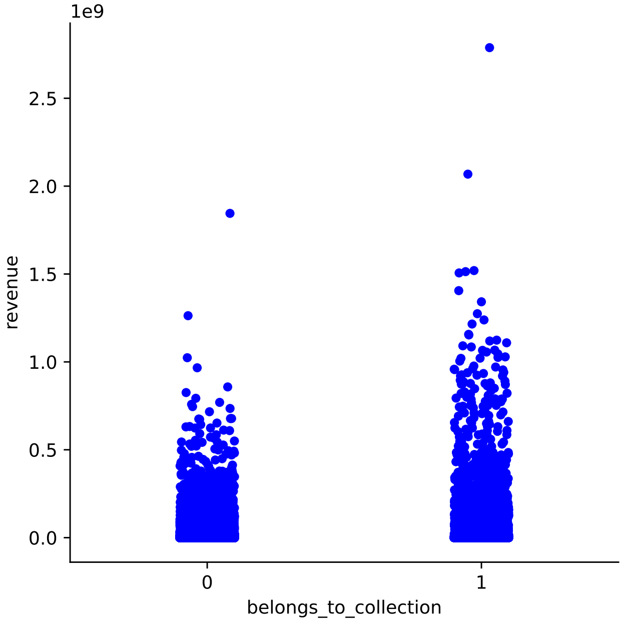


Figure 3: Catplot of whether a movie belongs to a franchise vs Revenue

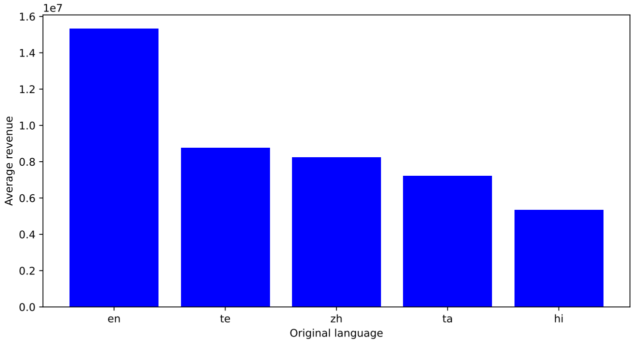


Figure 4: Barplot of original language vs Average Revenue

Similarly we explore the average revenue by genres and spoken languages in the movies. As with the original language of the movie, English is the most commonly spoken language with highest average revenue. In terms of genres, Action and Adventure had the highest average revenue, followed by Comedy, Drama, and Family. Genres like Documentary, Foreign, Western, History, Music made the slight average revenues. The plots showing these results are present in the notebook.

We did not conduct examination of production countries or production companies with respect to the highest average revenue. After the analysis of revenue with respect to all categorical and binary variable, we removed all such variables and examined the numerical attributes.

We first find the correlation matrix of all released movies as presented in Fig. [5]. All independent variables have positive correlation with the revenue, particularly vote_count, budget and popularity have very strong correlations with the revenue. This makes sense as high vote_counts and popularity imply higher number of viewings. Similarly higher budget yielding higher revenue makes sense. We then analyzed the budget and revenue of all released dataset. The budget attribute has the following statistical properties:

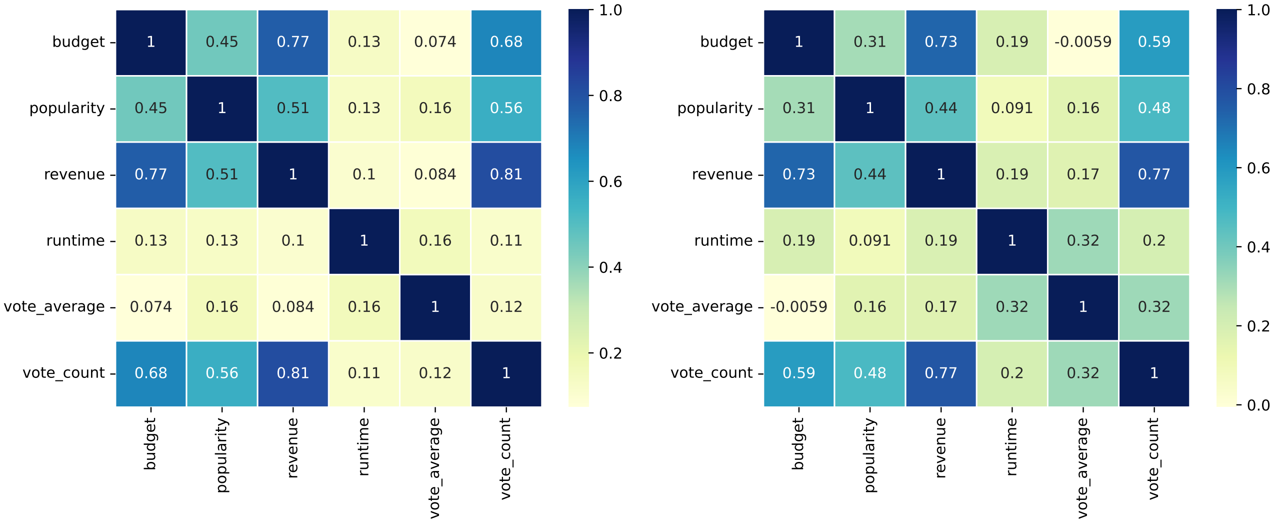


Figure 5: Left: Correlation matrix of all released movies
Right: Correlation matrix of movies with non-zero budget and revenue

total count	45014
mean	4265195
standard deviation	17505690
minimum	0
25% quartile	0
50% quartile	0
75% quartile	0
maximum	380000000

It is notable that most budget entries are 0, so we deleted all such entries. Similarly as 0 revenue does not make much sense, we removed movies with 0 revenue. The correlation matrix for the modified dataset with remaining 5377 movies is given in Fig. [5] right, which shows that original correlation between different variables is not affected much by deleting movies with 0 revenue and 0 budget. We then looked at the jointplots of each numerical attribute and revenue to understand any underlying relationship in Fig. [6]. Both budget and vote count seem to have linear positive correlation with revenue, whereas there does not seem to be present any notable relationship between runtime and revenue. Most movies are clustered at the left bottom corner of the jointplot of revenue and popularity with a few outliers with high popularity and high revenue. The jointplot for vote_average and revenue is not included here, but only in jupyter notebook, and these two variables do not seem to have any significant linear relationship. These results are consistent with the correlation matrix.

6.2.2 *Prediction Results and Evaluation.*

This section presents revenue prediction results for a case when the training and testing data are split into 80% and 20% of the modified dataset. During our initial exploration, we used all numerical attributes: budget, runtime, popularity, vote_average, and vote_count as the independent variables for the linear regression model. We get the following coefficients and intercept for our linear regression model:

budget	1.6041
popularity	250316.07
vote_count	74422.6048
runtime	-696291.88
vote_average	-2744485.56
intercept	-18144602.05

The positive coefficients of budget, popularity and vote_count are consistent with the positive correlation in Fig. [5]. However negative coefficients of vote_average and runtime fail to capture the positive correlation in the correlation matrix. In Fig. [6], we see that runtime do not have a significant linear relationship with revenue, and the coefficient fail to capture this behavior. Similarly the negative coefficient of vote_average is inconsistent with the positive correlation in Fig. [5].

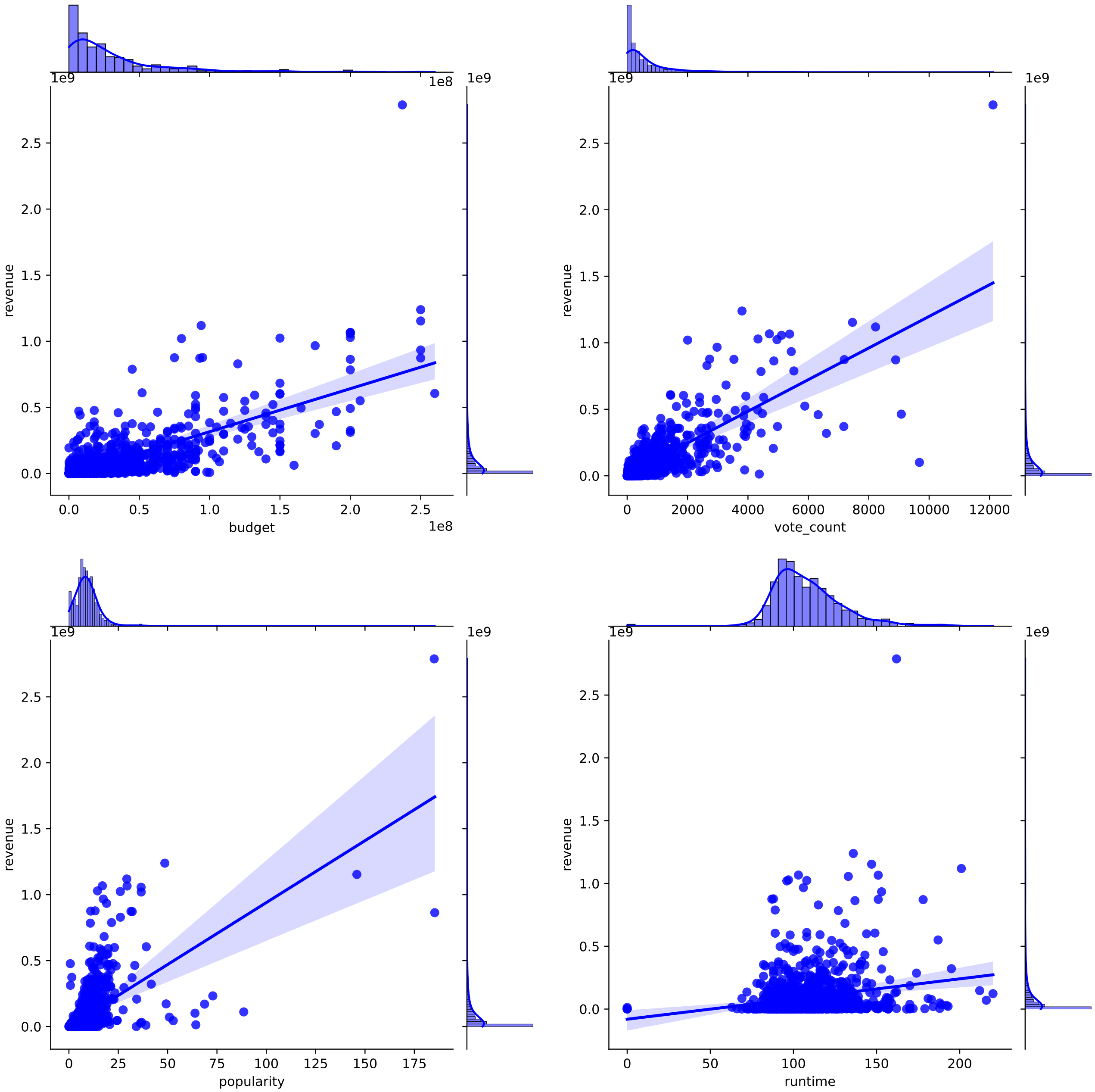


Figure 6: Top Left: Budget vs Revenue; Top Right: Vote Count vs Revenue
Bottom Left: Popularity vs Revenue; Bottom Right: Runtime vs Revenue

To evaluate our first linear regression model, we use the measures ***RMSE*** and R^2 ***statistic***, where

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

and

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

In the RMSE and R^2 equations above, n is the total test size, whereas y_i is the actual movie revenue and $\hat{y}_i$ is the predicted movie revenue. The linear regression model with all numerical variables as the independent variables had ***RMSE*** **= 92334999.363772** and $R^2 \approx$ ***0.69***. Although we can not evaluate our model independently using its own RMSE, R^2 relays important information.

Here $R^2 \approx 0.69$ implies that the 69% of the variance in revenue is explained by the variance in budget, popularity, runtime, vote_count, and vote_average.

Next, we found a linear regression model when only budget and vote_counts are independent variables. These two attributes seemed to have an evident linear relationship with revenue in the joint plots. The coefficients and the intercept are:

budget	1.645
vote_count	70178.09
intercept	-12784832.52

The positive coefficients of the independent variables are in agreement with positive correlation in Fig. [6]. The ***RMSE*** for this linear regression model is ***90237741.20***, which is smaller than for previous linear regression model, which suggests improvement in the accuracy of our prediction model. Similarly we observe higher R^2 value of ***0.7034***, suggesting improved prediction model.

For comparison, we predicted movie revenues using other regression models such as KNeighbors Regressor and RandomForest Regressor, which had ***RMSE*** of ***111536063.05*** and ***91345799.18***, respectively. The RMSE values of different regression models suggest that linear regression model with budget and vote_counts as the only independent variables is the best predictor model and the KNeighbors Regressor perform poorly in comparison with other models.

7 CONCLUSION AND LIMITATIONS

In our project, two recommendation systems were developed using GroupLens datasets. The first, employing collaborative filtering, utilized user ratings and movie attributes like title, movieId, and genre to yield personalized movie recommendations. The second, adopting a hybrid approach, considered both user history and genre preferences for refined suggestions. Despite their effectiveness, the models confront limitations. Collaborative filtering may struggle with sparse data and the cold start problem, particularly for new users or less-popular movies. The hybrid model, while nuanced, assumes consistent user preferences over time and could encounter scalability issues with larger datasets.

Notwithstanding these limitations, the recommendation systems offer valuable insights into user behavior and preferences, advancing the understanding of collaborative and hybrid filtering in movie recommendations. Future endeavors may explore advanced algorithms, adept at handling sparsity, and incorporate additional contextual information to overcome these limitations and further elevate the precision of recommendation models.

The linear regression model for revenue prediction is simple to understand and efficient to implement. The R^2 statistic close to 0.7 indicates the model's effectiveness for prediction. Moreover, the linear regression model's lower RMSE while compared to RMSE of other regression model suggests similar conclusion. However there are a few limitations of this prediction model.

First we note that some predicted revenues by the linear regression model are negative, which does not have practical interpretation in real context. Additionally, we used numerical attributes as the only independent variables for linear regression model. Categorical attributes such as production companies, language, genre, and cast do affect the revenue generation of a movie. Understanding a way to incorporate these attributes to the prediction model remains a work for future. Furthermore we were unable to understand some entries under original language and spoken languages variable due to the use of original script of the languages. Addressing these challenges and including other information such as ratings in prediction model are the direction to move forward in future.

REFERENCES

[1] Álvaro González, Fernando Ortega, Diego Pérez-López, and Santiago Alonso. 2022. Bias and Unfairness of Collaborative Filtering Based Recommender Systems in MovieLens Dataset. *IEEE Access* 10 (2022), 68429–68439. https://doi.org/10.1109/ACCESS.2022.3186719

[2] Guijia He and Soowon Lee. 2015. Multi-model or Single Model? A Study of Movie Box-Office Revenue Prediction. (2015), 321–325. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.46

[3] Dietmar Jannach, Pearl Pu, Francesco Ricci, and Markus Zanker. 2021. Recommender Systems: Past, Present, Future. *AI Magazine* 42 (2021), 3–6. Issue 3.

[4] Sambandam Jayalakshmi, Narayanan Ganesh, Robert Cep, and Janakiraman Senthil Murugan. 2022. Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions. *Sensors* (2022). https://www.mdpi.com/1424-8220/22/13/4904

[5] Dawei Li and Zhi-Ping Liu. 2022. Predicting Box-Office Markets with Machine Learning Methods. *Entropy (Basel, Switzerland)* 24 (2022). Issue 5. https://doi.org/10.3390/e24050711

[6] Rahul Pradhan, Ashish Chandra Swami, Akash Saxena, and Vikram Rajpoot. 2021. A Study on Movie Recommendations using Collaborative Filtering. *IOP Conf. Series: Materials Science and Engineering* (2021). https://iopscience.iop.org/article/10.1088/1757-899X/1119/1/012018/pdf

[7] Nahid Quader, Md. Osman Gani, and Dipankar Chaki. 2017. Performance evaluation of seven machine learning classification techniques for movie box office success prediction. *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)* (2017), 1–6. https://api.semanticscholar.org/CorpusID:25140665

[8] Mojtaba Sadeghian and Mohammad Khansari. 2018. A Recommender Systems Based on Similarity Networks: MovieLens Case Study. (2018), 705–709. https://doi.org/10.1109/ISTEL.2018.8661141

[9] Phonexay Vilakone, Doo-Soon Park, Khamphaphone Xinchang, and Fei Hao. 2018. An Efficient movie recommendation algorithm based on improved k-clique. *Human-centric Computing and Information Sciences* (2018). https://doi.org/10.1186/s13673-018-0161-6