Predicting Subscribers and Trialists for Movie Streaming Platform

Monica(Minkyung) Kim, Yena(Yunjie) Lu, Nai-Chieh Yang

Abstract

In this project, we analyzed the data extracted from the MUBI website and started with building models and comparing the performance of each model to choose the best method with the highest accuracy. The model chosen was used for conversion prediction. Afterward, a recommendation system was built given customer's ratings or preferences for trailers. Eventually, we provided suggestions, insights, and a practical algorithm for MUBI to make accurate predictions on whether one user will become a subscriber and a trial user.

1. Introduction

1.1 The MUBI platform

MUBI is a combination of a movie platform, a movie lover community, and a publication. It offers thirty handpicked movies by curators daily and provides a 7-day free trial. Moreover, MUBI users have a selection of thirty movies on a daily rotating basis. Unlike many other SVOD platforms relying on recommendation systems, MUBI provides human-curated selection and is open to nonsubscribers to leave ratings and comments.

1.2 The Problem

Founded in 2007, the same year the Netflix platform was launched, MUBI wasn’t able to grow as much as Netflix did. In 2015, MUBI had 100k subscribers, on the contrary, Netflix had 75 million subscribers. The reason for the huge gap was that MUBI streams art and indie movies, and recommends hand-picks movies. However, the post-Covid 19 era is a growth opportunity for movie streaming platforms(Westcott et al., 2020). Therefore, MUBI should seize the opportunity by identifying critical attributes for turning non-paying users into subscribers and predicting potential subscribers to boost growth.

1.3 The Hypothesis

(1) A user actively rating movies and leaving reviews is more likely to become a subscriber.
(2) The number of movies rated during the trial period has a high impact on conversion.
(3) Hand-curated films play a big role in turning free or trial users into paying users.

2. Data Exploratory Analysis

Three datasets among five datasets were retrieved from Kaggle: MUBI lists data, MUBI movie data, and MUBI rating data (Msika, 2020). In the MUBI lists dataset, 6 variables are used and aggregated for further analysis. In MUBI movies datasets, 6 variables are used and aggregated for further analysis. In the MUBI rating dataset, 9 variables are used and aggregated. We joined the three datasets and created a new dataset called 'dfuser' with 21 variables which were derived and aggregated from the three datasets. (See Table 1)

2.1 MUBI Lists Dataset - listdf

In the lists dataset, 23,118 unique users created 80,311 movie lists. Each list includes 37 movies on average. The max number of movies on the lists is 10,915. User ID 61596227 has 34,805 movies in the lists, 46,415 followers in total. Additionally, the user received 3,438 comments in the lists. According to the correlation of the variables result in the lists dataset, two sets of variables have a slightly strong relation. The number of lists has a positive relationship with the number of movies in the lists. Also, the number of followers has a positive relationship with the number of comments.

2.2 MUBI Movies Datasets - moviedf

In the movies dataset, there are 226,575 unique movies and 95,645 unique directors in total. The movie popularity is the number of times users click the 'Like' button on the website. In terms of movie popularity, its score ranges from 0 to 13,989. When the movies are listed from high to low popularity, the top ten movies are No.161, 303, 488, 147, 204, 405, 1537, 92, 148, and 315 released from 1960 to 2004. In addition, 5401 users rated all the 10 movies in the top ten movies by popularity.

2.3 MUBI Rating Dataset - ratingdf

In the rating dataset, there are 15.5 million rating records for 142,698 movies. Compared with the number of movies in the movie dataset, 38% of the movies on the MUBI website are not rated. The average rating for the movies is 3.59. Meanwhile, 5,271,049 movie ratings are at a score of 4.0. MovieID 133187 has the greatest number of ratings. 38% of users who rated have subscribed to the service on the website. When the movies are listed from the greatest number of ratings to the fewest number of ratings, the top ten movies are No.161, 147, 918, 92,405, 1731, 488, 303, 320, 204. The results of the top 10 movies in the movies dataset and ratings dataset are similar. 7 movies in the top 10 by ratings are the same as the ones in the top 10 by popularity and Pulp Fiction ranks in first place for both lists. It is worth noting that users love old movies. After we categorized movies by its released year, we found that although the number and popularity of old movies are lower than the recent movies, the average rating is higher at 3.93 compared with 3.77 and 3.37. (See Table 1)

We discovered that the number of ratings and unique users' ratings daily increased dramatically after Covid-19. Covid-19 is an opportunity for MUBI to attract more users to watch movies online instead of going to the theaters. Also, MUBI is confident that more users will get into the habit of watching movies from online platforms and subscribing to the service. Additionally, a spike happened on January 24th, 2019 when a large number of users rated movies. Some unknown event might have happened as there are suspicious users who rated more than 100 movies on that day (See Figure 1).

2.4 New Created User Dataset - dfuser

A new user dataset, dfuser, is derived from the three datasets. Each row contains data for each user. There are 451,757 unique users who rated a movie at least one time. Among the users, 23% of the users are subscribers and 8% of the users are trialists. Interestingly, there are no trialists that did not become subscribers after the trial period, meaning that MUBI has a high conversion rate. The remaining 347,521 users who are non subscribers and non trialists are MUBI's potential customers. Each user rated 34 movies on average and 5.1% of the users created a movie list. One interesting fact we discovered was that User ID 19983 is not a subscriber or a trialist but rated 20,000 movies with a 1.39 average rating score which is much lower than the average rating score, and rated 14 movies daily for 1428 days which is almost 4 years. It is obvious that a person watching or rating 14 movies per day is against common sense. Eventually, we defined User ID 19983 as an outlier and dropped its records.

3. Model Evaluation

The two binary classification problems of this project are predicting subscribers and predicting trialists. The final data set consisted of 21 variables and 451k rows. When we selected a model, the number of variables and the characteristics of each variable were in consideration. Gaussian Naïve Bayes model was not applicable as most of the continuous variables were not normally distributed. Also, Bernoulli Naive Bayes model was not applicable as there were few binary variables. Besides, some variables are calculated from other variables which are against the assumption of Naive Bayes: predictors are independent. Moreover, KNN was dropped from the list of models due to the number of dimensions. Finally, we have chosen Logistic Regression, Random Forest, XGBoost, and LightGBM to test and compare. Test AUC has been used as a criterion when we evaluate the four models.

3.1 Predicting Subscribers

For Logistic Regression, we selected 10 features with PCA which explained 91% of the variance, and elastic net regularization was chosen to penalize overfitting. The model had a 0.7780 test AUC. Random Forest with GridSearchCV returned a test AUC of 0.8725. XGBoost with Hyperopt optimization resulted in a

0.9007 test AUC. LightGBM with Hyperopt ran faster compared to XGBoost but had a 0.8657 test AUC. XGBoost performed the best among the models for predicting subscribers. (See Table 3)

Next, we decided to use SHAP to understand the variable importance and impact. The top three variables for predicting subscribers are the average movie popularity of rated movies, whether a user was a trialist or not, and movies rated per day. Based on the SHAP result, low movie popularity value tends to have a positive impact on becoming a subscriber. As the platform is for art movies, we may assume that the users are coming to MUBI to discover indie movies. Next, trial users have a higher tendency of becoming a subscriber. With this finding, MUBI can put more marketing effort to turn non-subscribers to try out the platform for a limited time for free. Finally, the rated movies per day variable is complicated to interpret. A high number of rated movies can either have a negative or no impact on the output. (See Figure 2)

3.2 Predicting Trialists

After we selected 10 features with PCA which was explaining 93% of the variance, Logistic Regression with elastic net regularization returned 0.6369 test AUC. Random Forest after GridSearchCV resulted in 0.6803 test AUC. XGBoost with Hyperopt optimization resulted in a 0.7795 test AUC. The train AUC was 0.8061 so the model was slightly overfitting but in the acceptable range. LightGBM with Hyperopt had a 0.7515 test AUC. Overall, XGBoost performed the best among the four models when it came to predicting trialists. (See Table 4)

With the SHAP results, we can interpret the variable importance and influence much easier. The top three variables of the XGBoost model predicting trialists were the average movie popularity of rated movies(movie_popularity) which was also important for predicting subscribers, movies rated per day(ratedperday), and the number of unique directors of rated movies(directorcount). High movie popularity tends to have a negative impact on becoming a trialist while low movie popularity can bring different results. This is an important finding for recommending movies to trialists. Also, users tend to become trialists when they rated a higher number of movies per day but smaller numbers of movies per day have a mixed impact on the result. Additionally, a small number of directors of rated movies have a positive impact on model output but this may be due to having 7 to 30 days on the platform as a trialist. (See Figure 3)

4. Conclusion and Recommendations

In the final recommendation, we would like to propose the best model and its implementation and key factors that could contribute to the sales increase to MUBI.

4.1 Model Recommendation

The best model for predicting both subscribers and trialists is XGBoost. The model evaluation process was the following:

(1) Import our XGBoost model. Set “user_subscriber”, “user_trialist” as dependent variables. Set variables such as “comments”, “movie_popularity” attributes as independent variables.
(2) Find the best parameter by setting the goal to look for the best loss (the best loss was 0.10).
(3) Run training set and testing set AUC and check out the overfitting/underfitting problem.
(4) Utilize the SHAP to see the importance of each variable.

4.2 Hypothesis Verification

(1) Hypothesis 1: “A user who actively rates movies and leaves reviews is more likely to be a subscriber.” was false. The SHAP graph showed that actively rating has no impact or a negative impact on a user being a subscriber. Moreover, leaving more reviews is less likely to influence a user to be a subscriber. Therefore, we can conclude that the potential subscribers are not proactively interacting with the website.
(2) Hypothesis 2: “The number of movies rated by users during the trial period having a high impact on conversion.” was true. Being a trialist has a positive impact on becoming a subscriber. This could be echoed back to our findings that all the trialists become subscribers. The user experience of being a trialist is so good that makes the conversion rate so high.
(3) Hypothesis 3: “Hand-curated films playing a big role in turning free or trial users into paying users.” was true. According to the SHAP graph, “movie_popularity” is the variable with the most impact but the low average popularity of movies rated has a positive impact. The less popular movie that was listed in the hand-curated list attracted the potential subscriber more.

4.3 Final Recommendation

(1) Data Collection Recommendation:

MUBI should try to avoid outlier records. For example, there are two abnormal records in the dataset, one is the sharp peak on 01/24/2019 and the other is the person that rated over 20k movies within 1,400 days. Thus, we suggest that MUBI should build a stronger cybersecurity system that could block the users that are acting abnormal or the extremely high number of ratings/comments in a short time. Also, MUBI should take more variables into consideration. For example, it could add variables such as “viewed movie list”, “average stay time”, “movie category”, etc. Moreover, MUBI could utilize marketing tools like Google Analytics and combine data for precise analysis and better understanding of the customer journey.

(2) Marketing/Operation Recommendation:

In terms of the result, MUBI has very niche subscribers and trialists who don’t like to go with the flow as the less popular movie has a better impact on users transferring into subscribers. MUBI should launch a discovery section on the website and promote older or rare movies for art movie lovers. The new function can be built to target the new potential boomer users that started to use streaming services in the post-COVID era.

Since everyone can leave comments and ratings on the website, users' judgment will be influenced by non-subscribers. We suggest that MUBI separate the website into two: one for movie discussion and the other for watching movies.

Although the hand-curated movie list has been worked out well, we suggest MUBI to improve the customer experience by using a recommendation system. Based on the XGBoost-SHAP results, high average rating scores across rated movies have a neutral impact on turning users into subscribers but a low average rating score has a great negative impact. Therefore, a recommendation system should recommend movies that a user is predicted to give high ratings.

Reference List

Bergstra, J., Yamins, D., Cox, D. D. (2013). Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms. *Proceedings of the 12th Python in Science Conference*

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95.

Jain, A. (2016). XGBoost Parameters: XGBoost Parameter Tuning. Retrieved October 15, 2020, from http://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

Lundberg, S. M. & Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. *31st Conference on Neural Information Processing Systems.*

McKinney, W., & others. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., … Liu, T. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *31st Conference on Neural Information Processing Systems.*

Msika, C. (2020, June). MUBI SVOD Platform Database for Movie Lovers (Version 2) [Data set]. Retrieved September 21, 2020 from https://www.kaggle.com/clementmsika/mubi-sqlite-database-for-movie-lovers.

Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1). Trelgol Publishing USA.

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Waskom, M., Botvinnik, O., O’Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., … Qalieh, A. (2017). *mwaskom/seaborn: v0.8.1 (September 2017)*. Zenodo. https://doi.org/10.5281/zenodo.883859

Westcott, K., Loucks, J., Downs, K., Arkenberg, C., Jarvis D. (2020). Digital media trends survey, 14th edition. *Deloitte Center for Technology, Media & Telecommunications*. Retrieved from Deloitte.

Appendix

Table 1: Final dataset variable

Variable	Derived from	Explanation	Variable	Derived from	Explanation
ratedmovies	ratingdf	The number of movies rated	user_trialist	ratingdf	1 = Trialist, 0 = Non-trialist
Creation_to_update_time	listdf	Time difference of first and last list update	directorcount	moviedf	Number of unique directors of rated movies
comments_sum	listdf	Number of comments	rating_time	ratingdf	Time difference of first and last rated movie
followers_sum	listdf	Number of followers	top10count	ratingdf/moviedf	Number rated movies among the top 10 movies
list_count	listdf	Number of lists	1.0	ratingdf	Number of movies rated 1
movie_popularity	moviedf	Number of users who click 'like' button	2.0	ratingdf	Number of movies rated 2
rating _score	ratingdf	Average rating score	3.0	ratingdf	Nnumber of movies rated 3
critic_comments	ratingdf	Number of comments in the critic	4.0	ratingdf	Number of movies rated 4
critic_likes	ratingdf	Number of likes for the critic	5.0	ratingdf	Number of movies rated 5
movie_in_list	listdf	Number of movies in the list	ratedperday	ratingdf	Number of rated movies per day
user_subscriber	ratingdf	1 = Subscriber, 0 = Non-subscriber			

Table 2: Movie count, popularity, and average rating by release year group

Year	Movie Count	Total Movie Popularity	Average Rating
1878~1950	10,980	945,853	3.93
1950~2000	55305	7,185,597	3.77
2000~2020	76358	7,388,457	3.37

Table 3: Test and Train AUC of four models for predicting subscribers

	XGBoost	LightGBM	Random Forest	Logistic Regression
Test AUC	0.9007	0.8657	0.8756	0.7780
Train AUC	0.9145	0.8685	0.8725	0.7790

Table 4: Test and Train AUC of four models for predicting trialists

	XGBoost	LightGBM	Random Forest	Logistic Regression
Test AUC	0.7795	0.7515	0.6803	0.6369
Train AUC	0.8061	0.7516	0.6828	0.6371

Figure 1: Number of ratings per day from Jan 2018 to April 2020

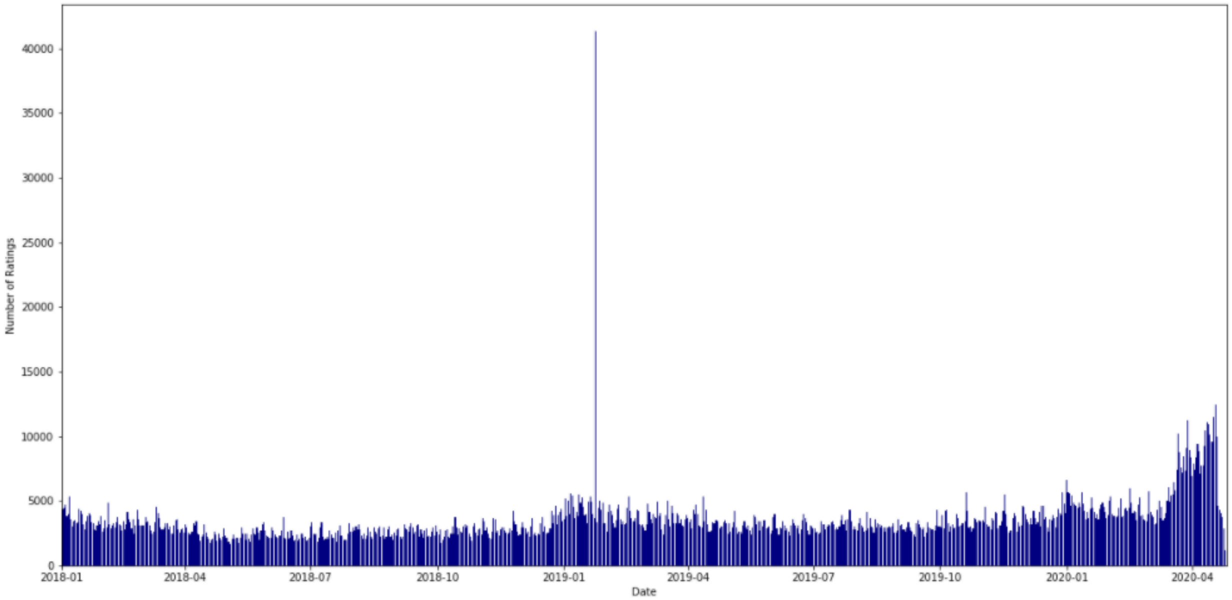


Figure 2: SHAP result for predicting subscribers with XGBoost

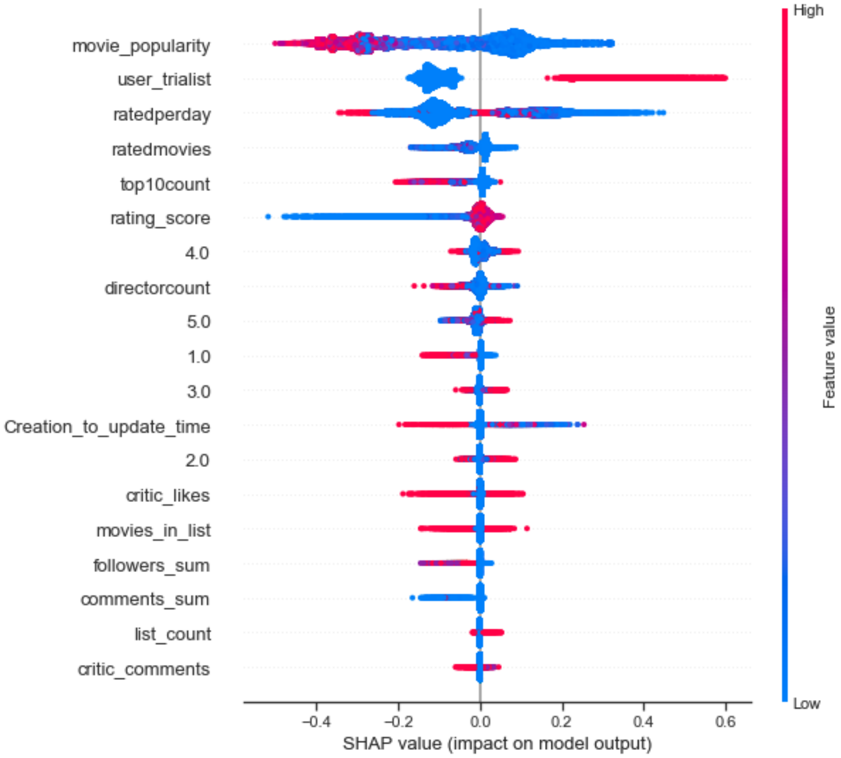
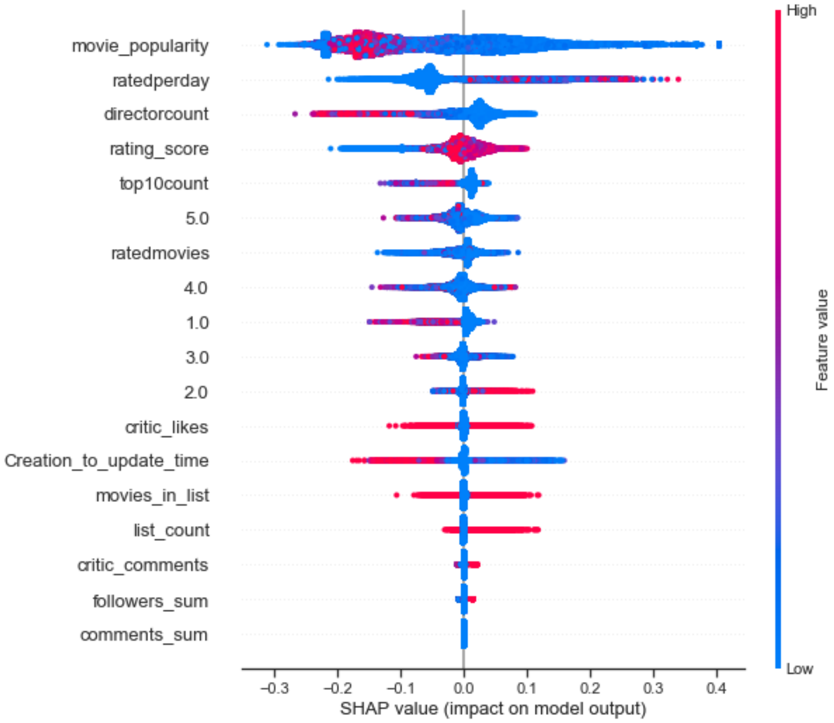


Figure 3: SHAP result for predicting trialist with XGBoost


movie_popularity
ratedperday
directorcount
rating_score
top10count
ratedmovies
critic_likes
Creation_to_update_time
movies_in_list
list_count
critic_comments
followers_sum
comments_sum
SHAP value (impact on model output)
High
Feature value
Low