Recommendation on Twitch: Predicting Users' Tendencies to Watched a Streamer

Abstract

For live-streaming platforms like Twitch, one essential aspect of marketing is to recommend streamers that appeal to users so that the platforms and streamers can earn more revenue with more clicks and gifts, while the users also having a more satisfying experience. The most widely used concept is to analyze the users' watching patterns and habits in hope to predict whether a streamer would draw the users' attention and make them want to click. Therefore, we are doing a task to predict using the interactive metadata that we have in hand to predict whether a user will watch a streamer that the user has not encountered before.

We selected the Twitch user-streamer dataset, which contains following fields:

1. User ID (Anonymized into 1 ~ 100000)
2. Stream ID
3. Streamer Name (Unique)
4. Start Time when the user watches the stream (unit in 10 mins)
5. End Time when the user finished watching the stream (metric in 10 mins)

We would learn about streamers that a user watched and the users that watched a streamer using the correspondence between "User ID" and "Streamer Name" fields which are unique to each user and each streamer, we would also use the start time and end time to do several predicative tasks involving thresholds.

1 Exploratory Analysis

The statistic and properties we focus on is the number of times that each streamer is watched, in other words, the watch frequency for each streamer. Twitch streaming services provide a typical dataset that is skewed to the right, with a large portion of streamers being viewed for fewer than a dozen of time while only a certain few are subscribed and watched by tens of thousands. By plotting density-based distribution of proportion of streamers against frequency of streamers being watched (Fig 1), we spotted a characteristic of bilateral distribution in the dataset. Across the samples, the vast majority of streamers possess a watch frequency of less than 3000, and a very minuscule fraction of the entirety shares the major influx of audiences, with watch frequency over 20,000. The frequency that lies in between, as a result, embodies scarcity. As a result, the distribution of streamers in terms of "popularity", metricized as frequency, is extremely tilted.

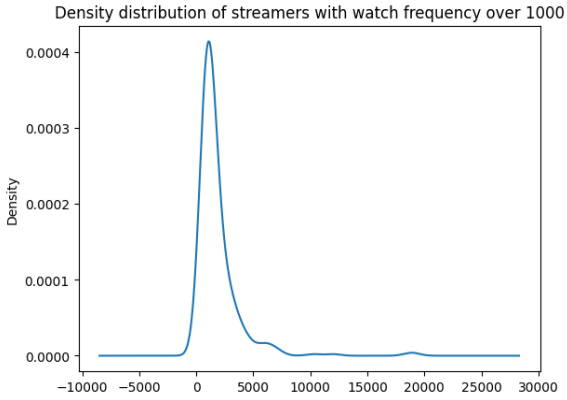

Fig 1

Similar tendencies are exposed in a scatter plot attempting to depict the density distribution of streamers based on their total live watch length (Fig 2). We discovered that the vast majority of streamers have no more than 20000 (10 min) unit time of watch lengths, and as total watch length increases, the distribution of streamers gets increasingly sparse.

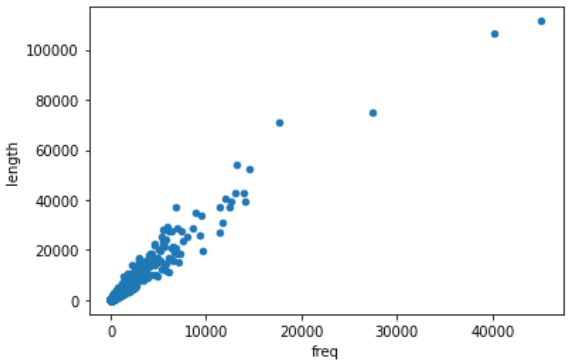

Fig 2

Interesting exploration of the data is drawn from such distribution tendencies. That is, suppose the watch frequency and total watch duration distributions are highly tilted, and that certain outliers with enormously high frequency/watch duration length and low occurrences. It suggested a pattern unlike early datasets we have examined. With disturbance of these outliers that consume the majority of stream shares, can we devise a model that can accurately predict the watch history and preference of users on the platform, and distinguish whether or not they will watch a certain streamer in the future? Thus, we decided our predictive task in the next section based on this exploration, and designed multiple pathways in achieving it.

2 Predictive Task: Predict whether the user would watch the streamer's stream or not

Based on the property of the dataset, and its similarity to our prior homework datasets, we devised a similar task, predicting whether or not the user would watch the streamer provided. In realistic scenarios, this prediction can be introduced as a supportive index in identifying whether the user is likely to watch this streamer in the future. If applied to the incorporation of Twitch's online recommender system, this supportive index can not only offer feedbacks on maintenance and parametric adjustment in terms of the site's recommendation process, but also can offer insight in personalized recommendation strategy with the prediction on user's watch likelihood or preference over certain types of streamers.

The task can essentially be converted into the "Read Predict" task we did in previous assignments, where we are also given a user and an item that the user has never interacted with and predict the opinion of the user. From the assignment, we know that there are two approaches to the problem: popularity of the streamer; similarity of the streamer compared to other streamers that the user has watched.

The baseline in the predictive task is based on popularity ranking using total watch duration for the streamer, regardless of their stream id, where we add streamers to a set of popular streamers until the sum of their streaming time exceeds 75% of total streaming time. We would improve upon it by taking similarity into account.

The validation of the predictions is based on the train-test split strategy, where we generate a negative set on the validation dataset to assess the prediction accuracy.

The features we used in the final model are Jaccard similarity and popularity. We created feature vectors based on the popularity and similarity between user/streamer set, where the approaches are adopted from the predictive tasks on collected beer and book reviews datasets. In order to calculate similarity, we stored user/streamer collections for each user-streamer pair in a user-streamer dictionary. To support the popularity-based model, we formulated pipelines of extracting the most popular streamers.

3 Model Description: Jaccard Similarity and Popularity Based Model

Our model is primarily based on Jaccard similarities. We gather the data of watch histories of users, and the audience sets of streamers, and use Jaccard similarity to calculate their intersections and likelihood of similarity in a user and a potential streamer, thus predicting the user's interest in watching the streamer's live. We adopted a parameter of 0.013 from the workbook as a threshold of whether the similarity is significant enough.

To optimize the model, we retained the popularity-based model from the baseline model. We adopted a parameter of frequency of 40 from the workbook as a threshold of whether the streamer is popular enough. Using two weighing factors, either when the examined streamer has the significant enough maximum similarity with the user's historical viewings, or when the popularity of the examined streamer is high enough, we would consider the user's interest in watching the streamer's live positive.

The issues due to scaling and overfitting lies in the efficiency of validation. Our algorithm in data pre-processing took O(n^2) time to run, so it will take significantly longer to run on larger datasets. Also, if we adjust our data input used for predictions over the entire dataset, it will overfit and encounter difficulty predicting parts of the data. To resolve the issue, we take only 0.67 to form a smaller dataset and split it into 50:25:25 for train, test and validation, leaving the rest for further testing. With this modification, we can finish data pre-processing within seconds and get similar accuracies within different test inputs.

Multiple other models are adopted. We used the baseline model and an improved version of baseline model that increases the cutoff of most popular streamers (allowing more streamers to be in the set). We also attempted Ridge Classifier to conduct the predictive task, and converted similarities and streamer ranking with respect to each user\streamer pair as features.

Unsuccessful attempts are models that involve a threshold. In accordance with the nature of our dataset, we designed a threshold in watch time for each record, filtering out watch histories with durations less than 10 minutes to eliminate outliers and put them in a negative set. We applied the approach on baseline models, Ridge Classifier, and Jaccard Similarity and Popularity Based Model. The accuracy turns out to be much lower when we use the threshold.

Strengths and weaknesses of models are as such:
Our baseline adopted mere popularity for predictive tasks. This means it would be fast to produce results at a cost of lower accuracy.
Our Ridge Classifier model and the improved baseline model both combine popularity with similarity. The models run fast but don't have an improvement in accuracy over the baseline. Our most successful attempt using Jaccard Similarity and Popularity Based Model has its strength in its high accuracy. However, it runs much slower than the other approaches, running around 4 to 6 minutes for a validation set of size around 500,000 to 700,000.

4 Literature and Dataset

Our dataset is based on Twitch live stream data. The data is gathered and published from Prof. McCauley's CSE 158 repos, which consisted of over 3,000,000 data points with each entry having fields of masked user ids, their stream ids, as well as the live streamers they watched with the start and end time in 10 minutes. The dataset consists of user data from 100,000 anonymized users and their watch

histories on the platform. With different streamer ids, the streamers stored in each user's records are unique, thus enabling the predictive tasks. By applying this model to the system, we can try to predict whether or not the user would watch the streamers' live.

Other similar datasets include:
Welcome! | Million Song Dataset: This dataset consists of metadata of various online songs and lyrics incorporating sub-datasets of users who have listened to each song, relevant ratings, as well as their preferences as song lists, etc.
Book-Crossing Dataset (uni-freiburg.de): This dataset consists of review data fetched from the Book-Crossing community which incorporates 278,858 anonymized users and their reviews over 271,379 book items containing a total amount of 1,149,780 reviews.

These datasets are similar to the one we chose in a way that they both consisted of reviewer/item pairs which can provide similarity-based analysis possibilities. Similar approaches that we adopted on our own dataset can be applied in a similar manner to the two sample datasets identified above.

The state-of-the-art method we found in terms of conducting predictive tasks over similar datasets comes from an essay published by prof. Juan McCauley on performing predictive tasks over live streaming platform data extracted from Twitch in July 2019. Prof. McCauley and his colleagues utilized repeated consumptions to construct feature vectors from users-streamer data based on repeated consumptions and interactions, With construction of self-attention matrix over sequences of users over certain streamers extracted from dot product rankings, they can predict and recommend streamers to users in a flow.

5 Results and Discussion

Ridge Classifier, as well as the previous baseline attempts, achieved similar accuracy around 0.52.

The performance of Jaccard Similarity and Popularity Based Model is the best. After tuning on hyperparameters in the model, the accuracy in watch prediction reached an accuracy of 0.88.

For features and parameters, this model is based on user-streamer pair dictionaries and Jaccard-calculated similarities among user-streamer pairs in the validation process. We use the streamers set from the user and users set from the streamer as features in calculation of similarities, and popularity of the streamer based on its ranking and the time being watched from the training set. Instead of fitting a classifier, we utilized our combined approach out of consideration that the formatting was simple, and that fitting our classifier into a dataset with highly heated data points due to some top-tier streamers would result in skewed predictions.

For other models, the baseline and the improved version of the baseline model both reached accuracy of 0.82. The improved baseline model doesn't have an improvement over the original model possibly because of the lack of adjustable parameters for tuning, making the model too inflexible.
The classifier approach merely reached 0.82 accuracy in our trial as well. This might be because our labeling of popularity index in feature extraction for the classifier might have also distracted the model, as there are some significant differences in rankings of the most popular streamers.
As we adopted a threshold in calculating the similarities, all models perform poorly compared to when we don't use it, having accuracies from 0.49 to 0.52 depending on the model, possibly because the heuristic choice of watch time threshold is not validable and therefore prone to inaccuracy.

In conclusion, for a model to successfully predict watching, it requires the popularity ranking and similarity features to work together. It is also evident that the freedom of fine tuning hyperparameters in the model can help, and the dependence on heuristic with no justification can lead to worsening results.

Reference

[1] Jérémie Rappaz, Julian McAuley and Karl Aberer. RecSys, 2021. Recommendation on Live-Streaming Platforms: Dynamic Availability and Repeat Consumption
[2] Julian McAuley. CSE 158/258 Homeowrk Solutions
[3] Julian McAuley. CSE 158/258 Assignment 2