An Ensemble Approach to Streaming Service Churn Prediction

WSDM Cup 2018 Churn Prediction Challenge

Hang Li
Data Science Team, Hulu LLC
Los Angeles, USA
hangli@hulu.com

Quang Hieu Vu
Data Science Group, Zalora
Ho Chi Minh City, Vietnam
quanghieu.vu@zalora.com

Thanh Lam Pham
DataLab, VNG
Ho Chi Minh City, Vietnam
lampt@vng.com.vn

Tam T. Nguyen
LS3 Lab, Ryerson University
Toronto, Canada
nthanhtam@gmail.com

Song Chen
American International Group
New York City, USA
song.chen@aig.com

Jeong-Yoon Lee
Microsoft
Los Angeles, USA
jeol@microsoft.com

ABSTRACT

Churn prediction plays the central role in all customer retention strategies that aim to keep customers with the company. However, there is not much work on churn prediction for music streaming services where the data consists of user activity logs (service usage) and subscription transactions (registration and subscription history). How to leverage this heterogeneous data set to have the best churn prediction models still needs further investigation. This paper presents an ensemble learning approach to predict the likelihood of customer churn in a music streaming service. The proposed approach will be discussed from two main aspects: (1) what are important features from membership registration, subscription transactions and historical service usage to construct single models and (2) how to take advantages of emerging machine learning algorithms to build the best ensemble model. The testing results show that our solution has the log-loss score of 0.10076 and 0.10187 in the public and private leader boards, respectively.

KEYWORDS

Churn prediction, streaming service, churn prediction challenge

ACM Reference format:
Hang Li, Quang Hieu Vu, Thanh Lam Pham, Tam T. Nguyen, Song Chen, and Jeong-Yoon Lee. 2018. An Ensemble Approach to Streaming Service Churn Prediction. In *Proceedings of WSDM conference, US, February 2018 (WSDM'18),* 8 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The customer is the heart of any company and no company can grow and prosper without having customers. In fact, without customers, you do not have a company at all. Given this important role of customers, customer relationship management (CRM) is an important part of all companies. This part is even more important for subscription-based companies whose major source of revenue comes from the subscription fees of customers. In CRM, an important task is to acquire and keep customers with the company. The experience from customer-centric, however, states that customer acquisition is often much more expensive than customer retention. It is because the cost and efforts putting into advertising for getting new customers are high. In general, the cost of retaining an existing customer is only from one tenth to at most one fifth of the cost of getting a new customer. As a result, several companies have a retention strategy to keep customers at all times. To implement a good retention strategy, however, churn prediction plays the central part.

With the development of multimedia compression technology and the popularization of the high-speed internet, streaming service is becoming a very important part of the entertainment industry and becoming an integral part of life. Several well-known streaming services are subscription only or have subscription product, e.g. Netflix, Hulu, Amazon Instant Video, Apple Music, Google Play Music/YouTube Red, Pandora, Spotify, etc. It will not be surprised that these companies are working on their own churn prediction internally as parts of their customer retention strategy.

In KKBOX's Churn Prediction Challenge[1], participants were challenged to build an algorithm that predicts whether a user will churn after their subscription expires based on membership information, user logs, and transaction history. This problem attracts a lot of attention in Kaggle, finally there were 575 teams participated this challenge. To address this challenge, we proposed a two-layer ensemble approach with intensively carrying out feature engineering based on all available data. Our approach achieved 0.10076 log-loss in the private leader board that is one of top 10 teams in the competition. In summary, our contributions are as follows:

- A thoughtful analysis and discussion of important features generated from membership registration, subscription transactions and historical service usage to build single churn prediction model.
- A strategy to leverage advantages of single models to build a better-performed ensemble model.

The remainder of this paper is organized as follows. In Section 2, we present related work. In Section 3, we present all features we used. In Section 4, we introduce our proposed ensemble model. In Section 5, we show experimental study. Finally, in Section 6, we

WSDM'18, February 2018, US

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] https://www.kaggle.com/c/kkbox-churn-prediction-challenge

draw some concluding remarks and discuss potential improvements for the future.

2 RELATED WORK

In this section, we briefly introduce the machine learning techniques used in our model: Deep Learning, XGBoost and Ensemble model.

2.1 Deep Learning

Deep Learning (DL) refers to a class of machine learning techniques and architectures, where many layers of non-linear information processing stages in hierarchical architectures are exploited for representation learning. Particularly, a DL network represents a multi-layer neural network with the deeper structures compared to the shallow models like Support Vector Machines and a specific method where the data is processed at and in between layers. Even though the concept of DL was introduced long time ago, it has only gained popularity recently due to the lower cost of computing hardware, the increased speed of chip processing, and recent advances in DL algorithms. DL has been successfully employed for graphical modeling, optimization, pattern recognition, signal processing, and natural language processing [1].

2.2 XGBoost and LightGBM

Boosting is a powerful meta-algorithm used to reduce prediction bias. The basic idea of boosting algorithm is to first produce a series of individually average performing models trained on the subsets of original data and then boost their performance by combining them together using various aggregation functions like majority vote or weighted average [6]. Gradient boosting is a version of boosting method that manages to achieve deeper performance gains compared to state-of-the-art predictors and is commonly used to solve regression and classification problems. XGBoost [2] is an implementation of the gradient boosted decision trees based on the extreme gradient boosting model [4]. Recently, XGBoost and later LightGBM [5] have gained increased popularity and attention due to their advantages of fast processing speed and high prediction performance. In particular, these models have been used to win top prizes in many machine learning competitions hosted in Kaggle[2], the largest data science community in the world.

2.3 Ensemble Method

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of their predictions [3]. Similar to XGBoost, ensemble method is a popular technique employed by winning teams in Kaggle's machine learning competitions. In our work, we employ a 2 layers ensemble approach. The 1^{st} layer is using lightGBM and simple average ensemble 10 different models. The 2^{nd} layer is a simple average ensemble method from 2 different 1st layer ensemble models.

3 FEATURE ENGINEERING

In this section, we will present our method to generate features that consist of membership, user log, transaction, and churn-related features. The whole feature set including feature description is listed in all tables in Appendix A. Team members generate different feature sets from the whole feature set.

[2] http://www.kaggle.com

Table 1: Cut-off Date of Training and Testing Data

File	Meaning	Cut-off Date
train.csv	Churn label in Feb 2017	2017-02-01
train_v2.csv	Churn label in Mar 2017	2017-03-01
sample_submission_v2.csv	Churn or not in Apr 2017	2017-04-01

3.1 Pre-processing & Cut-off Date

As user log data is huge, i.e. 30GB on disk, it is not trivial to process it directly. We use several different ways to process this data. The proposed approach is solely based on data warehouse methods as follows:

- **Time-based partitioning.** We split the data into many parts based on time. We then process each part independently. We apply this method when we extract time-related features.
- **Hashing partitioning.** We use a hashing function to split the data into many sub-sets. For instance, we hash 'msno' and then index the data by this hashing value. This technique is used when we generate features for each 'msno'.
- **Incremental processing.** Process "user_logs" by chunk then merge stats result from each chunk. This is a similar idea as "map-reduce", and can be parallelized to map-reduce easily.

In this challenge, transactions and user logs are time-related behaviors to avoid time travel when generating features. We set a cut-off date for each training file and testing file. When generating transaction or user log related features, we cut off and only use the data before the cut-off date. The cut-off date for the training and testing data can be found in Table 1.

3.2 Membership Features

We use all features of the membership data. Details of the features are shown in Table 6. We directly use membership data such as city, age of user, registered channel, etc. as features. Moreover, we also decompose registration date into month, day, and year. We then use them as date features.

3.3 User Log features

For user logs and transaction data, we generate features using various time windows as follows:

- Entire history: from the beginning to cut-off date
- Last month: from 1 month before cut-off date to cut-off date
- Day 7: from 7 days before cut-off date to cut-off date
- Day 7-14: from 14 days before cut-off date to 7 days before cut-off date
- Day 14-21: from 21 days before cut-off date to 14 days before cut-off date
- Day 21-28: from 28 days before cut-off date to 28 days before cut-off date
- Day 7-28: from 28 days before cut-off date to 7 days before cut-off date
- Week 4-8: from 8 weeks before cut-off date to 4 weeks before cut-off date

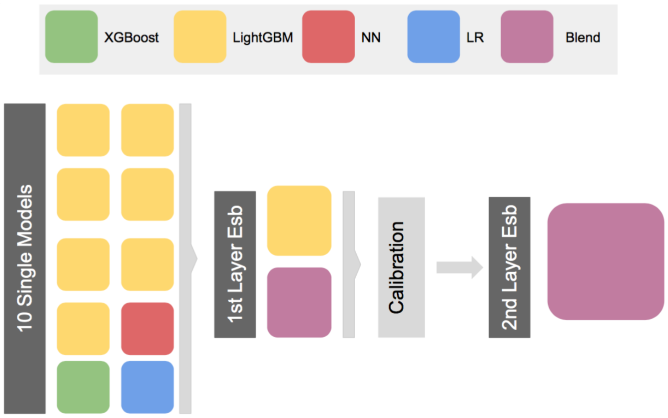

Figure 1: Modelling Architecture.

- Eight weeks - 5 months: from 5 months days before cut-off date to 8 weeks before cut-off date

Daily user logs describe the listening behavior of a user. This is the biggest data generated in streaming service. We generate user's active days and total consumption hours and seconds during different time periods to reflect user's overall activity in KKBOX in Table 7. Then we aggregate number of songs played in user logs to have more user log features. Details are described in Table 10.

3.4 Transaction Features

Transaction is the billing information of all user in the whole dataset. We generate the most common value of transactions during a given time period. The detailed features are shown in Table 8. We next calculate the ratio of auto renew and cancel transactions during a given time window. These kinds of features are shown in Table 13. We then aggregate transaction data to have statistical features in the time window as shown in Table 11. Table 12 shows our final feature set based on subscription service.

3.5 Churn Related Features

We also generate monthly churn or not-churn features from the transactions using the same logic provided by organizer. These features are shown in Table 9.

4 MODELLING METHODOLOGY

In this section, we will respectively introduce our local validation method, single models, and ensemble models for churn prediction. Modelling architecture can be found in Figure 1.

4.1 Local Validation

The organizer uses *log-loss* as final leader board metric. To validate model in local, we merge train.csv and train_v2.csv and use stratified 5 fold cross-validation to evaluate the performance of our models.

4.2 Base Models

For different feature sets, we build multiple models including LightGBM, XGBoost, Logistic regression and Neural Networks with different parameters. In order to make prediction, we split the training data into 5 folds and train 5 models. The final prediction is the average of 5 predictions of 5 models. This is the fold-based bagging technique mentioned in [7]. LB performance result of 10 based models which are used in final ensemble models are shown in Table 2 is the prediction after applying calibration which is mentioned in Section 4.4.

Table 2: Base Models

Model #	Model	Feature Set	Private LB	Public LB
01	LightGBM_v1	F1	0.10592	0.10716
02	LightGBM_v2	F2	0.10580	0.10716
03	LightGBM_v2	F3	0.10332	0.10462
04	Neural Network	F3	N/A	N/A
05	LightGBM_v3	F4	0.10351	0.10439
06	XGBoost	F5	N/A	N/A
07	Logistic Regression	F6	N/A	N/A
08	LightGBM_v4	F6	N/A	N/A
09	LightGBM_v5	F7	0.10190	0.10320
10	LightGBM_v6	F8	0.11359	0.11416

Table 3: Ensemble Models

Model #	Model Name	Base Models	Private Leader Board	Public Leader Board
11	LightGBM	#1 - #9	0.10200	0.10309
12	average blend	#6, #10	0.10105	0.10214
13	average blend	#11, #12	0.10076	0.10187

4.3 Ensemble Models

From all base models we choose diverse type, diverse feature set, and high local validation performance models into ensemble models. Using 10 base models, we build two 1^{st} layer ensemble models using LightGBM and average blend. We then calculate mean of the ensemble models in the 2^{nd} layer of ensemble model. This averaged prediction is used as the final prediction of our proposed approach.

4.4 Calibration

From private leader board, we can estimate the churn ratio of test set (Apr 2017) is 3.57%. We calibrate our final prediction probability to be mean as 3.57% through simple scaling method as shown in Equation 1.

$$Ratio = \frac{0.0357}{\sum_{i \in test\ set} Prob_i}$$
$$Prob_{calibration} = Ratio * Prob_{original} \tag{1}$$

5 PERFORMANCE EVALUATION

5.1 Performance Result

Table 2 shows the performance of our base models on the leader board (LB). LightGBM outperforms other algorithms. It has the log-log of 0.1019 and 0.1032 on the private and public LBs, respectively. For models with NA results, we have not submitted them to check the leader board score.

The performance result of ensemble models is shown in Table 3. We build an ensemble LightGBM model and a average blend model using the predictions of base models as the input. Their performance is better than base models in public LB. Mean of two 1^{st} layer

Table 4: Feature Ranking List

Feature #	Name	Gain %
TR6	auto renew ratio week 8 - 5 months	0.150925
TR5	auto renew ratio week 4-8	0.084758
TR4	auto renew ratio day 7-28	0.062273
TR10	cancel ratio day 7-28	0.032292
M7	registration months	0.028751
S25	tran_date_duration week 8 - 5 months	0.026572
UU3	num_unq day 7	0.025188
TR9	cancel ratio day 7	0.023437
TR3	auto renew ratio day 7	0.022739
UH7	total_seconds day 7	0.018946
S35	days_before_expiration_max day 7-28	0.018603
UN39	num_100 day 7	0.014604
S33	expire_date_duration week 8 - 5 months	0.012745
S39	days_before_cut-off_min day 7-28	0.012607
UU7	num_unq day 7-28	0.012496
TS47	plan_list_price_min week 4-8	0.012304
S8	tran_frequency week 8 - 5 months	0.011449
S34	days_before_expiration_max day 7	0.011447
S44	days_before_expiration_max week 4-8	0.010778
S43	days_before_expiration_max day 7-28	0.010754

Table 5: Model Performance with Top Features

Features	CV Log-loss	CV AUC
Top 10 of subset	0.158699107078	0.864959026377
Top 20 of subset	0.148143412525	0.886655015795
Top 30 of subset	0.146019232689	0.890673475636
Entire subset (121 features)	0.1448001201	0.891853544445

ensemble models achieves log-loss 0.10076 and 0.10187 in private and public LBs, respectively. Its performance is better than 1^{st} layer ensemble models and it is our final score.

5.2 Feature Importance Study

A feature importance analysis on a subset of the whole features through LightGBM. It is generated by one of our team member during competition. Top 20 most important features are shown in Table 4 where gain is percentage of gains of splits which use the feature of all features. In these top features, auto renew features are the most important. They are a good indicator whether or not a user will be a churner. It is reasonable because if a user select auto renew option, she/he will be using the service in the near future.

We use 5 fold cross validation to test the prediction power of top features. The performance result is shown in Table 5 where LightGBM works very well on the first top 30 features with the log-loss of 0.1460. Comparing to the entire subset(121 features), its log-loss drops only 0.0028 that is not significant. Therefore, in practice, the recommended number of features should be about 30.

6 FINAL THOUGHT

In this paper, we presented a two-layer ensemble approach with intensively carrying out feature engineering based on membership, user log and transaction data. There exists other techniques that we think they are useful for solving this problem but we have not tried. They could be included in the future work.

- Representation learning, using denoising autoencoder to have a better representation than raw features.
- Retrain all baseline models through best feature set.
- Random forest, extra-trees and other tree-based model.
- Generate more historical training through target labeling code.

Moreover, we think that leveraging more data could improve the performance of churn prediction. The following data internally could be helpful.

- Device usage information.
- Time of day usage and pattern.
- Preference of users.
- Quality of service.

In subscription based service company, the final goal is to improve customer's retention, the challenges can be listed as follows:

- **Identify high risk users.** This can be solved through supervised learning approach.
- **Understand why user will churn.** Model interpretation, causal inference, controlled experiment and other techniques can be used to tackle this.
- **Understand the impact of different influence channel.** There are several different channels can be used to influence user to make decision in company's favor.

There will be a long journey to go after implementing a churn prediction model.

REFERENCES

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (Aug. 2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[2] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[3] Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems (MCS '00)*. Springer-Verlag, London, UK, UK, 1–15. http://dl.acm.org/citation.cfm?id=648054.743935

[4] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Ann. Statist.* 29, 5 (10 2001), 1189–1232. https://doi.org/10.1214/aos/1013203451

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3149–3157. http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

[6] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. 1999. Boosting Algorithms As Gradient Descent. In *Proceedings of the 12th International Conference on Neural Information Processing Systems (NIPS'99)*. MIT Press, Cambridge, MA, USA, 512–518. http://dl.acm.org/citation.cfm?id=3009657.3009730

[7] Tam T. Nguyen, Hossein Fani, Ebrahim Bagheri, and Gilberto Titericz. 2017. Bagging Model for Product Title Quality with Noise. In *Proceedings of the International Conference on Information and Knowledge Discovery (CIKM'17)*.

A LIST OF ALL FEATURES FOR STREAMING SERVICE CHURN PREDICTION

Table 6: Membership Related Features

Feature #	Name	Description
M1	city	city of the user
M2	bd	age of the user
M3	gender	gender of the user
M4	registered_via	channel where the user registered
M5	registration days	days from the time the user registered to the cut-off date
M7	registration months	months from the time the user registered to the cut-off date
M8	year of registration init time	year of user registered
M9	month of registration init time	month of user registered
M10	day of registration init time	day of month of user registered

Table 7: Overall Consumption Related Features

Feature #	Name Time Period	Description
UA1	active days entire history	total days of user using the service during entire history
UA2	active days recently 1 month	total days of user using the service during recently 1 month
UA3	active days day 7	total days of user using the service during recently 7 days
UA4	active days day 7-14	total days of user using the service between day 7 and 14
UA5	active days day 14-21	total days of user using the service between day 14 and 21
UA6	active days day 21-28	total days of user using the service between day 21 and 28
UH1	total_hours entire history	total hours of using the service during entire history
UH2	total_hours recently 1 month	total hours of using the service during recently 1 month
UH3	total_hours day 7	total hours of using the service during recently 7 days
UH4	total_hours day 7-14	total hours of using the service between day 7 and 14
UH5	total_hours day 14-21	total hours of using the service between day 14 and 21
UH6	total_hours day 21-28	total hours of using the service between day 21 and 28
UH7	total_seconds entire history	total seconds of using the service during entire history
UH8	total_seconds recently 1 month	total seconds of using the service during recently 1 month
UH9	total_seconds day 7	total seconds of using the service during recently 7 days
UH10	total_seconds day 7-28	total seconds of using the service between day 7 and 28
UH11	total_seconds week 4-8	total seconds of using the service between week 4 and 8
UH12	total_seconds 8 weeks - 5 months	total seconds of using the service between 8 weeks and 5 months
UH13	total_seconds last transaction	total seconds of using the service during the last transaction
UH14	total_seconds_mean entire history	the average total seconds of using the service during entire history
UH15	total_seconds_std entire history	the standard deviation total seconds of using the service during entire history

Table 8: Most Common Value in Transaction Features

Feature #	Name Time Period	Description
TC1	most common payment_method_id entire history	most common payment method during entire history
TC2	most common payment_method_id recently 1 month	most common payment method during recently 1 month
TC3	most common payment_plan_days entire history	most common length of membership plan in days during entire history
TC4	most common payment_plan_days recently 1 month	most common length of membership plan in days during recently 1 month
TC5	most common plan_list_price entire history	most common plan list price during entire history
TC6	most common plan_list_price recently 1 month	most common plan list price during recently 1 month
TC7	most common actual_amount_paid entire history	most common actual amount paid during entire history
TC8	most common actual_amount_paid recently 1 month	most common actual amount paid during recently 1 month
TC9	most common auto renew entire history	most common auto renew or not during entire history
TC10	most common auto renew recently 1 month	most common auto renew or not during recently 1 month
TC11	most common cancel entire history	most common cancel or not during entire history
TC12	most common cancel recently 1 month	most common cancel or not during recently 1 month

Table 9: Lag Churn Features

Feature #	Name Time Period	Description
C1	prev_churn1m	churn or not in last month
C2	prev_churn2m	churn or not between 1 and 2 months
C3	churn encode city	the churn ratio of city
C4	churn encode registered_via	the churn ratio of channel
C5	time_since_first_suspension	number of months since the first suspension(churn) in history
C6	time_since_last_suspension	number of months since the last suspension(churn) in history

Table 10: User Log Related Features

Feature #	Name Time Period	Description
UU1	num_unq entire history	sum of daily unique songs played during entire history
UU2	num_unq recently 1 month	sum of daily unique songs played during recently 1 month
UU3	num_unq day 7	sum of daily unique songs played during recently 7 days
UU4	num_unq day 7-14	sum of daily unique songs played between day 7 and 14
UU5	num_unq day 14-21	sum of daily unique songs played between day 14 and 21
UU6	num_unq day 21-28	sum of daily unique songs played between day 21 and 28
UU7	num_unq day 7-28	sum of daily unique songs played between day 7 and 28
UU8	num_unq week 4-8	sum of daily unique songs played between week 4 and 8
UU9	num_unq 8 weeks - 5 months	sum of daily unique songs played between 8 weeks and 5 months
UN1	num_25 entire history	sum of songs played less than 25% of the song length during entire history
UN2	num_25 recently 1 month	sum of songs played less than 25% of the song length during recently 1 month
UN3	num_25 day 7	sum of songs played less than 25% of the song length during recently 7 days
UN4	num_25 day 7-14	sum of songs played less than 25% of the song length between day 7 and 14
UN5	num_25 day 14-21	sum of songs played less than 25% of the song length between day 14 and 21
UN6	num_25 day 21-28	sum of songs played less than 25% of the song length between day 21 and 28
UN7	num_25 day 7-28	sum of songs played less than 25% of the song length between day 7 and 28
UN8	num_25 week 4-8	sum of songs played less than 25% of the song length between week 4 and 8
UN9	num_25 8 weeks - 5 months	sum of songs played less than 25% of the song length between 8 weeks and 5 months
UN10	num_50 entire history	sum of songs played between 25% to 50% of the song length during entire history
UN11	num_50 recently 1 month	sum of songs played between 25% to 50% of the song length during recently 1 month
UN12	num_50 day 7	sum of songs played between 25% to 50% of the song length during recently 7 days
UN13	num_50 day 7-14	sum of songs played between 25% to 50% of the song length between day 7 and 14
UN14	num_50 day 14-21	sum of songs played between 25% to 50% of the song length between day 14 and 21
UN15	num_50 day 21-28	sum of songs played between 25% to 50% of the song length between day 21 and 28
UN16	num_50 day 7-28	sum of songs played between 25% to 50% of the song length between day 7 and 28
UN17	num_50 week 4-8	sum of songs played between 25% to 50% of the song length between week 4 and 8
UN18	num_50 8 weeks - 5 months	sum of songs played between 25% to 50% of the song length between 8 weeks and 5 months
UN19	num_75 entire history	sum of songs played between 50% to 75% of the song length during entire history
UN20	num_75 recently 1 month	sum of songs played between 50% to 75% of the song length during recently 1 month
UN21	num_75 day 7	sum of songs played between 50% to 75% of the song length during recently 7 days
UN22	num_75 day 7-14	sum of songs played between 50% to 75% of the song length between day 7 and 14
UN23	num_75 day 14-21	sum of songs played between 50% to 75% of the song length between day 14 and 21
UN24	num_75 day 21-28	sum of songs played between 50% to 75% of the song length between day 21 and 28
UN25	num_75 day 7-28	sum of songs played between 50% to 75% of the song length between day 7 and 28
UN26	num_75 week 4-8	sum of songs played between 50% to 75% of the song length between week 4 and 8
UN27	num_75 8 weeks - 5 months	sum of songs played between 50% to 75% of the song length between 8 weeks and 5 months
UN28	num_985 entire history	sum of songs played between 75% to 98.5% of the song length during entire history
UN29	num_985 recently 1 month	sum of songs played between 75% to 98.5% of the song length during recently 1 month
UN30	num_985 day 7	sum of songs played between 75% to 98.5% of the song length during recently 7 days
UN31	num_985 day 7-14	sum of songs played between 75% to 98.5% of the song length between day 7 and 14
UN32	num_985 day 14-21	sum of songs played between 75% to 98.5% of the song length between day 14 and 21
UN33	num_985 day 21-28	sum of songs played between 75% to 98.5% of the song length between day 21 and 28
UN34	num_985 day 7-28	sum of songs played between 75% to 98.5% of the song length between day 7 and 28
UN35	num_985 week 4-8	sum of songs played between 75% to 98.5% of the song length between week 4 and 8
UN36	num_985 8 weeks - 5 months	sum of songs played between 75% to 98.5% of the song length between 8 weeks and 5 months
UN37	num_100 entire history	sum of songs played over 98.5% of the song length during entire history
UN38	num_100 recently 1 month	sum of songs played over 98.5% of the song length during recently 1 month
UN39	num_100 day 7	sum of songs played over 98.5% of the song length during recently 7 days
UN40	num_100 day 7-14	sum of songs played over 98.5% of the song length between day 7 and 14
UN41	num_100 day 14-21	sum of songs played over 98.5% of the song length between day 14 and 21
UN42	num_100 day 21-28	sum of songs played over 98.5% of the song length between day 21 and 28
UN43	num_100 day 7-28	sum of songs played over 98.5% of the song length between day 7 and 28
UN44	num_100 week 4-8	sum of songs played over 98.5% of the song length between week 4 and 8
UN45	num_100 8 weeks - 5 months	sum of songs played over 98.5% of the song length between 8 weeks and 5 months
UF1	userlog_first	total days from the first record date to cut-off date
UL1	userlog_last	total days from the last record date to cut-off date
UI1	userlog_interval_min entire history	the shortest duration days between two consequence record dates during entire history
UI2	userlog_interval_max entire history	the largest duration days between two consequence record dates during entire history
UI3	userlog_interval_mean entire history	the average duration days between two consequence record dates during entire history
US1	repeat_songs entire history	sum of songs played minus sum of unique during entire history
US2	repeat_songs 1 month	sum of songs played minus sum of unique during recently 1 month
US3	avg_repeat_songs entire history	the average of replayed songs per day during entire history
US4	avg_repeat_songs 1 month	the average of replayed songs per day during recently 1 month
US5	quick_scan entire history	the average of songs played less than 25% per day during entire history
US6	quick_scan 1 month	the average of songs played less than 25% per day during recently 1 month
US7	playlist_usage entire history	sum of songs played over 75% of the song length during entire history
US8	playlist_usage 1 month	sum of songs played over 75% of the song length during recently 1 month
US9	avg_playlist_usage entire history	the average of songs played over 75% of the song length per day during entire history
US10	avg_playlist_usage 1 month	the average of songs played over 75% of the song length per day during recently 1 month

Table 11: Statistical Transaction Features

Feature #	Name Time Period	Description
TS1	payment_plan_days_mean entire history	the average payment plan days during entire history
TS2	payment_plan_days_mean 1 month	the average payment plan days during recently 1 month
TS3	payment_plan_days_median entire history	the median payment plan days during entire history
TS4	payment_plan_days_median 1 month	the median payment plan days during recently 1 month
TS5	payment_plan_days_min entire history	the shortest payment plan days during entire history
TS6	payment_plan_days_min 1 month	the shortest payment plan days during recently 1 month
TS7	payment_plan_days_max entire history	the longest payment plan days during entire history
TS8	payment_plan_days_max 1 month	the longest payment plan days during recently 1 month
TS9	plan_list_price_mean entire history	the average plan list price during entire history
TS10	plan_list_price_mean 1 month	the average plan list price during recently 1 month
TS11	plan_list_price_median entire history	the median plan list price during entire history
TS12	plan_list_price_median 1 month	the median plan list price during recently 1 month
TS13	plan_list_price_min entire history	the minimum plan list price during entire history
TS14	plan_list_price_min 1 month	the minimum plan list price during recently 1 month
TS15	plan_list_price_max entire history	the maximum plan list price during entire history
TS16	plan_list_price_max 1 month	the maximum plan list price during recently 1 month
TS17	actual_amount_paid_mean entire history	the average actual amount paid during entire history
TS18	actual_amount_paid_mean 1 month	the average actual amount paid during recently 1 month
TS19	actual_amount_paid_median entire history	the median actual amount paid during entire history
TS20	actual_amount_paid_median 1 month	the median actual amount paid during recently 1 month
TS21	actual_amount_paid_min entire history	the minimum actual amount paid during entire history
TS22	actual_amount_paid_min 1 month	the minimum actual amount paid during recently 1 month
TS23	actual_amount_paid_max entire history	the maximum actual amount paid during entire history
TS24	actual_amount_paid_max 1 month	the maximum actual amount paid during recently 1 month
TS25	payment_plan_days_median day 7	the median payment plan days during recently 7 days
TS26	payment_plan_days_median day 7-28	the median payment plan days between day 7 and 28
TS27	payment_plan_days_median week 4-8	the median payment plan days between week 4 and 8
TS28	payment_plan_days_median week 8 - 5 months	the median payment plan days between 8 weeks and 5 months
TS29	payment_plan_days_min day 7	the minimum payment plan days during recently 7 days
TS30	payment_plan_days_min day 7-28	the minimum payment plan days between day 7 and 28
TS31	payment_plan_days_min week 4-8	the minimum payment plan days between week 4 and 8
TS32	payment_plan_days_min week 8 - 5 months	the minimum payment plan days between 8 weeks and 5 months
TS33	payment_plan_days_max day 7	the maximum payment plan days during recently 7 days
TS34	payment_plan_days_max day 7-28	the maximum payment plan days between day 7 and 28
TS35	payment_plan_days_max week 4-8	the maximum payment plan days between week 4 and 8
TS36	payment_plan_days_max week 8 - 5 months	the maximum payment plan days between 8 weeks and 5 months
TS37	payment_plan_days_std day 7	the standard deviation payment plan days during recently 7 days
TS38	payment_plan_days_std day 7-28	the standard deviation payment plan days between day 7 and 28
TS39	payment_plan_days_std week 4-8	the standard deviation payment plan days between week 4 and 8
TS40	payment_plan_days_std week 8 - 5 months	the standard deviation payment plan days between 8 weeks and 5 months
TS41	plan_list_price_median day 7	the median plan list price during recently 7 days
TS42	plan_list_price_median day 7-28	the median plan list price between day 7 and 28
TS43	plan_list_price_median week 4-8	the median plan list price between week 4 and 8
TS44	plan_list_price_median week 8 - 5 months	the median plan list price between 8 weeks and 5 months
TS45	plan_list_price_min day 7	the minimum plan list price during recently 7 days
TS46	plan_list_price_min day 7-28	the minimum plan list price between day 7 and 28
TS47	plan_list_price_min week 4-8	the minimum plan list price between week 4 and 8
TS48	plan_list_price_min week 8 - 5 months	the minimum plan list price between 8 weeks and 5 months
TS49	plan_list_price_max day 7	the maximum plan list price during recently 7 days
TS50	plan_list_price_max day 7-28	the maximum plan list price between day 7 and 28
TS51	plan_list_price_max week 4-8	the maximum plan list price between week 4 and 8
TS52	plan_list_price_max week 8 - 5 months	the maximum plan list price between 8 weeks and 5 months
TS53	plan_list_price_std day 7	the standard deviation plan list price during recently 7 days
TS54	plan_list_price_std day 7-28	the standard deviation plan list price days between day 7 and 28
TS55	plan_list_price_std week 4-8	the standard deviation plan list price days between week 4 and 8
TS56	plan_list_price_std week 8 - 5 months	the standard deviation plan list price between 8 weeks and 5 months
TS57	actual_amount_paid_median day 7	the median actual amount paid during recently 7 days
TS58	actual_amount_paid_median day 7-28	the median actual amount paid between day 7 and 28
TS59	actual_amount_paid_median week 4-8	the median actual amount paid between week 4 and 8
TS60	actual_amount_paid_median week 8 - 5 months	the median actual amount paid between 8 weeks and 5 months
TS61	actual_amount_paid_min day 7	the minimum actual amount paid during recently 7 days
TS62	actual_amount_paid_min day 7-28	the minimum actual amount paid between day 7 and 28
TS63	actual_amount_paid_min week 4-8	the minimum actual amount paid between week 4 and 8
TS64	actual_amount_paid_min week 8 - 5 months	the minimum actual amount paid between 8 weeks and 5 months
TS65	actual_amount_paid_max day 7	the maximum actual amount paid during recently 7 days
TS66	actual_amount_paid_max day 7-28	the maximum actual amount paid between day 7 and 28
TS67	actual_amount_paid_max week 4-8	the maximum actual amount paid between week 4 and 8
TS68	actual_amount_paid_max week 8 - 5 months	the maximum actual amount paid between 8 weeks and 5 months
TS69	actual_amount_paid_std day 7	the standard deviation actual amount paid during entire history
TS70	actual_amount_paid_std day 7	the standard deviation actual amount paid during entire history
TS71	actual_amount_paid_std day 7-28	the standard deviation actual amount paid days between day 7 and 28
TS72	actual_amount_paid_std week 4-8	the standard deviation actual amount paid days between week 4 and 8
TS73	actual_amount_paid_std week 8 - 5 months	the standard deviation actual amount paid between 8 weeks and 5 months

Table 12: Subscription based Service Related Features

Feature #	Name Time Period	Description
S1	tran_first	days from the first transaction date to cut-off date, this can also be considered as user's tenure
S2	tran_last	days from the last transaction date to cut-off date
S3	tran_last_expired	days from the last transaction's expired data to cut-off date
S4	tran_frequency	total number of transactions
S5	tran_frequency day 7	total number of transactions during recently 7 days
S6	tran_frequency day 7-28	total number of transactions between day 7 and 28
S7	tran_frequency week 4-8	total number of transactions between week 4 and 8
S8	tran_frequency week 8 - 5 months	total number of transactions between 8 weeks and 5 months
S9	unique_tran_method day 7	total number of unique transaction method during recently 7 days
S10	unique_tran_method day 7-28	total number of unique transaction method between day 7 and 28
S11	unique_tran_method week 4-8	total number of unique transaction method between week 4 and 8
S12	unique_tran_method week 8 - 5 months	total number of unique transaction method between 8 weeks and 5 months
S13	tran_total_days	total number of subscribed days from transactions
S14	tran_total_paids	total fee that has been paid for all subscribed transaction
S15	tran_total_discounts	total discount that has been received from subscribed transactions
S16	tran_last_payment	payment method of the latest (newest) transaction
S17	tran_last_renew	whether the latest transaction has auto renew option
S18	tran_last_cancel	whether the latest transaction was canceled
S19	product count 1 month	how many different products(unique plan list price) user had during recently 1 month
S20	payment status 1 month	what is users payment status: free trail, convert to pay or paid during recently 1 month
S21	pay ratio 1 month	ratio of actual payment and list price during recently 1 month
S22	tran_date_duration day 7	days between 1st transaction and last transaction during recently 7 days
S23	tran_date_duration day 7-28	days between 1st transaction and last transaction between day 7 and 28
S24	tran_date_duration week 4-8	days between 1st transaction and last transaction between week 4 and 8
S25	tran_date_duration week 8 - 5 months	days between 1st transaction and last transaction between 8 weeks and 5 months
S26	expire_date_duration day 7	days between 1st expiration date and last expiration date of transactions during recently 7 days
S27	expire_date_duration day 7-28	days between 1st expiration date and last expiration date of transactions between day 7 and 28
S28	expire_date_duration week 4-8	days between 1st expiration date and last expiration date of transactions between week 4 and 8
S29	expire_date_duration week 8 - 5 months	days between 1st expiration date and last expiration date of transactions between 8 weeks and 5 months
S30	days_before_expiration_min day 7	the min of days between transaction date and expiration date during recently 7 days
S31	days_before_expiration_min day 7-28	the min of days between transaction date and expiration date between day 7 and 28
S32	days_before_expiration_min week 4-8	the min of days between transaction date and expiration date between week 4 and 8
S33	days_before_expiration_min week 8 - 5 months	the min of days between transaction date and expiration date between 8 weeks and 5 months
S34	days_before_expiration_max day 7	the max of days between transaction date and expiration date during recently 7 days
S35	days_before_expiration_max day 7-28	the max of days between transaction date and expiration date between day 7 and 28
S36	days_before_expiration_max week 4-8	the max of days between transaction date and expiration date between week 4 and 8
S37	days_before_expiration_max week 8 - 5 months	the max of days between transaction date and expiration date between 8 weeks and 5 months
S38	days_before_cut-off_min day 7	the min of days between transaction date and cut-off date during recently 7 days
S39	days_before_cut-off_min day 7-28	the min of days between transaction date and cut-off date between day 7 and 28
S40	days_before_cut-off_min week 4-8	the min of days between transaction date and cut-off date between week 4 and 8
S41	days_before_cut-off_min week 8 - 5 months	the min of days between transaction date and cut-off date between 8 weeks and 5 months
S42	days_before_cut-off_max day 7	the max of days between transaction date and cut-off date during recently 7 days
S43	days_before_cut-off_max day 7-28	the max of days between transaction date and cut-off date between day 7 and 28
S44	days_before_cut-off_max week 4-8	the max of days between transaction date and cut-off date between week 4 and 8
S45	days_before_cut-off_max week 8 - 5 months	the max of days between transaction date and cut-off date between 8 weeks and 5 months
S46	first_tran_length	length of the 1st transaction
S47	last_tran_length	length of the last transaction

Table 13: Ratio of Transaction Related Features

Feature #	Name Time Period	Description
TR1	auto renew ratio entire history	auto renew transaction ratio of all transactions during entire history
TR2	auto renew ratio recently 1 month	auto renew transaction ratio of all transactions during recently 1 month
TR3	auto renew ratio day 7	auto renew transaction ratio of all transactions during recently 7 days
TR4	auto renew ratio day 7-28	auto renew transaction ratio of all transactions between day 7 and 28
TR5	auto renew ratio week 4-8	auto renew transaction ratio of all transactions between week 4 and 8
TR6	auto renew ratio week 8 - 5 months	auto renew transaction ratio of all transactions between 8 weeks and 5 months
TR7	cancel ratio entire history	cancel transaction ratio of all transactions during entire history
TR8	cancel ratio recently 1 month	cancel transaction ratio of all transactions during recently 1 month
TR9	cancel ratio day 7	cancel transaction ratio of all transactions during recently 7 days
TR10	cancel ratio day 7-28	cancel transaction ratio of all transactions between day 7 and 28
TR12	cancel ratio week 8 - 5 months	cancel transaction ratio of all transactions between 8 weeks and 5 months