A Work Project, presented as part of the requirements for the Award of a Master's degree in Business Analytics from the Nova School of Business and Economics.

Deep Learning Frameworks for Enhanced User Engagement Prediction

Zhanshuo Guo

Work project carried out under the supervision of:

Yufei Shen

17/12/2025

Abstract

This work demonstrates the individual contribution of Zhanshuo Guo in the field lab project "How Early Can We Predict Churn? Short-Window Engagement Forecasting in a Hybrid UGC Music Platform". While chapters 1-3 summarize the collective effort of the group, chapter 4 dives into the deep learning approaches of the project.

Using impression-level data from NetEase Cloud Music recently launched UGC module "Cloud Village", this work aims at providing a scalable solution for predicting user engagement from short-term behaviors, which can help the management team to handle high user mobility issue and design win-back strategies. We employ both machine and deep learning models across two targets: binary churn and multiclass engagement. The cross design provides complementary perspectives on user behaviors, enabling performance indicators from one task to enrich the other. The result contributes to a more robust understanding of engagement prediction and operational values.

Keywords

Deep Learning, Churn Prediction, Long Short-Term Memory, Sequence Processing

This work was funded by Fundação para a Ciência e a Tecnologia (UID/00124/2025, UID/PRR/124/2025, Nova School of Business and Economics) and LISBOA2030 (DataLab2030 - LISBOA2030-FEDER-01314200).

Table of Contents

1. Introduction 6

1.1 The Shift to the Attention Economy 6

1.2 The Latency Problem in Churn Prediction 7

1.3 Research Objectives and Questions 8

1.4 Scope and Significance 8

2. Research Context and Data 10

2.1 Music Streaming Platform: NetEase Cloud Music 10

2.2 Data Description 11

2.3 Sampling Method and Key Definitions 11

2.4 Measuring User Engagement 14

2.5 Research Design 17

3. Results & Conclusion 18

3.1 Limitations 20

3.2 Future Research Directions 21

4. Deep Learning Approaches 23

4.1 Data Preprocessing 23

4.1.1 Sessions 23

4.1.2 Session Preparation 24

4.1.3 User Demographics 26

4.2 Long Short-Term Memory (LSTM) 26

4.2.1 Model Structure and Setup 26

4.2.2 Binary Classification Results 28

4.2.3 Multiclass Classification Result 30

4.3 Hybrid LSTM 31

4.3.1 Attention Module 31

4.3.2 MLP Module 32

4.3.3 Model Structure and Setup 33

4.3.4 Binary Classification Results 34

4.3.5 Multiclass Classification Result 35

4.4 Summary 36

List of Figures

Figure 1: the Dynamic-starting Sampling 12

Figure 2: Distribution of Engagement Scores 16

Figure 3: Proportion of Target Classes 17

Figure 4: Structure of Memory Block 27

Figure 5: Structure of Basic LSTM 28

Figure 6: Confusion Matrix of Basic LSTM 29

Figure 7: Class-wise Metrics of Basic LSTM 31

Figure 8: Structure of Hybrid LSTM 33

Figure 9: Confusion Matrix of Hybrid LSTM 35

Figure 10: Class-wise Metrics of Basic LSTM 36

1. Introduction

1.1 The Shift to the Attention Economy

Music streaming services compete in an increasingly crowded marketplace with one of the most valuable resources being the user's attention. The early years of music streaming focused on providing the largest possible library of music and then expanding accessibility. With libraries now large enough that every major artist and label is available on all streaming services, competition has shifted toward how well a service retains users and encourages them to continue using the service. In other words, the question that each service needs to answer is no longer “Who will subscribe?” But rather “Who will stay, who will actively engage, and who will leave?”

This project is located at the nexus of behavioral analytics and Machine Learning, but the setting of the problem is unique in that it involves music streaming services and user generated content (UGC). Unlike typical e-commerce sites or contractually obligated software-as-a-service (SaaS), music streaming services exist in a hyper-competitive space where there are very low barriers to switching. As a result, a user may slowly withdraw from the site (a concept known as soft churn) before they even decide to formally close their account (Hadiji et al. 2014). This phenomenon is particularly difficult for UGC-based services to address since users do not simply consume professional content but also create their own content in order to engage with others. Creating a social ecosystem based upon commenting, sharing and creating content drives the network effects of a service like NetEase Cloud Music (Katona et al. 2011; Huang et al. 2019). Therefore, understanding the lifecycle of users of such services, particularly during the post-adoption period, is crucial (Bhattacherjee 2001; Limayem et al. 2007). Studies indicate that the first few days of interaction between a user and a service contain the latent signals required to predict long-term patterns of behavior (Hoang and Cam

2025). The key to successfully decoding those signals is the difference between a successful, self-sustaining community and a “leaky bucket” business model.

1.2 The Latency Problem in Churn Prediction

One of the greatest challenges facing Customer Relationship Management (CRM) and Retention Analytics is the time delay between when a user decides psychologically to disengage from a service and when the service identifies that decision (Buckinx and Van den Poel 2005; Verbeke et al. 2012). Most models of churn use lagging indicators, such as a sudden drop off in Monthly Active Users (MAU) or a missed payment, to flag a potentially churned user (Buckinx and Van den Poel 2005). By the time these indicators are detected, however, the user has typically already broken his habit loop, preventing the effectiveness of any retention efforts (Limayem et al. 2007).

This research seeks to solve the early detection problem by focusing on the fourteen days of user activity, which is usually considered as a short period in business practices, to find behavioral signatures that predict retention in the subsequent weeks (i.e., activity in the third week). A “short observation, mid-term prediction” framework is operationally critical. It enables service providers to implement interventions (e.g., algorithmic boosts, push notifications, personalized incentives) while the user is still in the confirmation phase of the Information Systems Lifecycle, i.e., during the interval between initial trial and habitual use (Bhattacherjee 2001). This effectively closes the gap between trial and habitual use, intervening with users when they are assessing whether the service has sufficient utility to warrant continued use.

The early detection of users at risk is urgent as industry data indicates that early user behavior is significantly more predictive of long-term outcomes than later user behavior. For example, some studies suggest that up to 60% of users abandon mobile apps after just a few interactions (Hoang and Cam 2025). This high rate of abandonment in the early stages of use provides a "window of opportunity" to intervene with users when the cost of doing so is minimal but the potential Customer Lifetime Value (CLV) saved is substantial (Bhattacharjee et al. 2023). Unfortunately, identifying users at risk of abandoning a service requires more than simply examining aggregate measures of user behavior; instead, sophisticated models capable of interpreting the nuances of sequential behavior are needed (Gao et al. 2023; Chen et al. 2012).

1.3 Research Objectives and Questions

The main goal of this thesis is to develop and test a predictive model architecture that uses fine-grained, short-term behavioral data to forecast user retention on a music streaming/UGC service. Specifically, the goals of this project include comparing the performance of traditional aggregate feature models to sequential deep learning models, particularly with respect to the trade-offs among model complexity, explainability, and the timeliness of predictions.

In light of the proposed research methodology and the existing gaps in the research literature, this study is guided by the following Research Question:

How can early user interactions, such as clicks, likes, and shares, be used to predict future user activity and engagement on the platform?

1.4 Scope and Significance

This research utilizes a massive dataset from NetEase Cloud Music containing more than 57 million impressions and approximately 2 million users. The boundaries of this project are limited to the "Cloud Village" tab of the app (a specific tab of user-generated content) which represents an excellent proxy for modern social-streaming hybrid environments.

The significance of this work lies in its potential to operationalize “Habit Formation Theory” (Limayem et al. 2007) using Deep Learning. By transforming abstract concepts of user engagement into tangible tensor representations in a Long Short-Term Memory (LSTM) network, this research provides a methodological link between Behavioral Science and Data Engineering. Furthermore, the emphasis on both accuracy and explainability directly responds to the industry's need for actionable insights (Bauer et al. 2023; Wang et al. 2022; Rudin 2019). Knowing why a user is at risk of churning is as important as knowing that they are at risk, allowing the move from simple risk-scoring to “LIFT” modeling and targeted intervention (Ascarza 2018; Lemmens and Gupta 2020).

2. Research Context and Data

2.1 Music Streaming Platform: NetEase Cloud Music

NetEase Cloud Music (NCM) is an online music streaming platform launched in April 2013 by Chinese company NetEase Inc., which gained immediate popularity in the following years and rose into top four online music platforms along with Ali Music, Tencent Music Entertainment (TME), and Tai He Music Group. In April 2017, it was valued at around 8 billion CNY (1.14 billion USD). Besides a freemium-based music streaming service, the company has been consistently exploring new possibilities. In August 2019, a social entertainment module named "Cloud Village" where users could view and post short videos known as "Mlog", was officially started on the mobile applications. The entry of this module is placed as the fourth icon on the bottom tab, indicating a similar level of importance as the music playing function from the perspective of user interface design. After entering this module, by default a series of content cards will be shown to users in the format of an infinite scrolling timeline, recognized as "Discovery" sub-tab; Also, users can manually switch to "Follow" sub-tab to check latest content posted by creators they have followed.

As the "Cloud Village" serves a purpose completely different from music playback, it might suffer from high user mobility and user retention remains a critical concern for NCM's management group. Therefore, the goal of this research is to develop a scalable predictive system that is capable of capturing churn signals from users' early-on-actions within a limited time period and delivering an estimate of user's activity intensity in the next seven days, which can contribute to the business value of the platform.

2.2 Data Description

The publicly available data set provided by Zhang et al. (2020) serves as the basis for this work. The highlight of this data set is that it provides fifty-seven million impression-level data exclusively from previously mentioned "Discovery" sub-tab in the Cloud Village module, associated with 2,085,533 users from November 1 to 30, 2019. As described earlier in the module's user interface, an impression in this context refers to a content card shown to a given user within his own timeline. Thus, a record of impression is uniquely determined by a timestamp together with the user and the content card's identifiers, describing whether the user has certain kinds of interaction with given card through binary fields such as *isClick*, *isLike*, *isShare*, and total time spent on this card in seconds through numeric field *mlogViewTime*. This core impression data is further complemented by other data sets including demographics as well as daily statistics for users, creators, and content on the platform. Preserving a structure that remains as close as possible to the raw data stored in online warehouses provides us with considerable flexibility for applying task-specific transformations in the modeling stage. For example, via the content card's identifier, each impression can be linked to the corresponding card's metadata. This relationship enables the extraction of a user's preferred content categories, creator types, and other relevant preference signals. Moreover, because the data are available at the impression level, we then have more freedom in selecting appropriate analytical granularity and in constructing industry-relevant KPIs.

A detailed overview of the original dataset is presented in *Appendix 1*.

2.3 Sampling Method and Key Definitions

Due to the limited sampling period of the original dataset, the length of observable activity varies across users, depending on when they first appear in the data. For example, a user who first appeared on November 20 has only 10 days of observable records. To guarantee that the users included in our study have data of comparable length and extended coverage, we require at least 21 days of available data, consisting of a 14-day observation window and a 7-day label period. Therefore, we then decided to include only a subset of total 2,085,533 users, whose first appearance occurred within the first seven days of the 30-day dataset. Despite the restriction, this still provides 841,670 users for model training and evaluation. The definition of observation window and label period is presented below and visualized in *Figure 1*:

- **Observation Window:** 14 days following a user's first appearance (e.g., November 7–20 for a user first observed on November 7).
- **Label Period:** The subsequent 7 days used to assess continued activity (e.g., November 21–27 for the same user).

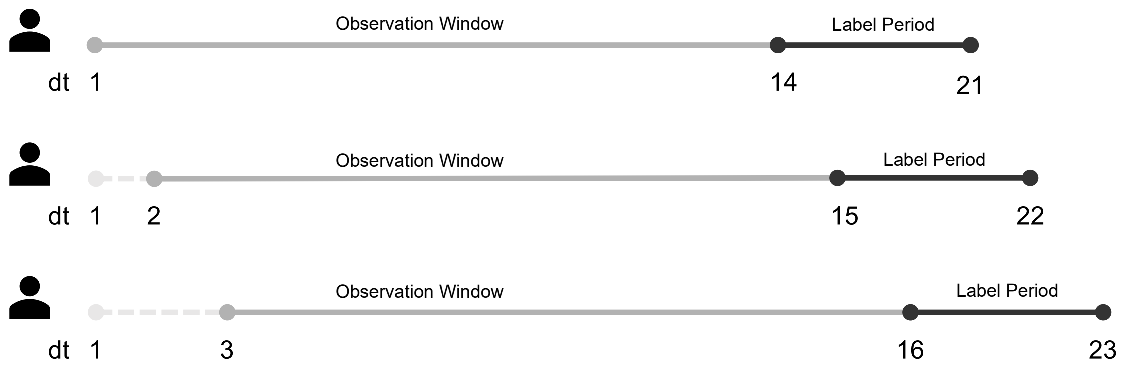

Figure 1: the Dynamic-starting Sampling

Choosing an appropriate standard that could distinguish active and inactive users is usually a challenge in the churn-prediction problems. Unlike cancelling subscriptions in the music streaming service that NCM operates, this module lacks a deciding factor that marks a user's churn. Therefore, our definition of being inactive follows the practice of Shamik (2022): A full week with zero impressions. Such a condition may indicate either complete churn from the overall NCM service or continued use of other functions on the platform with diminished interest in the Cloud Village module. Regarding the length of the period used to determine user being inactive, both overly short and overly long windows introduce noise. A short window provides insufficient evidence to affirm a user's activity status, whereas an extended window may include users who 'revive' after being inactive for a long time due to external events. In industrial practices, seven days of inactive (having zero impression in our scenario) are typically regarded as "light" churn, and fourteen days as "hard" churn. Therefore, we believe our choice for label period length is reasonable and practical. Besides, it balances the proportion of active and inactive users. Under this standard, there are 518,363 (61.5%) inactive users and 323,307 (38.5%) active users in respective label periods. As more restricted standards such as having impressions greater than a certain threshold (e.g., at least five impressions) or having at least one click contribute to the imbalance in the proportion, we eventually adhered to this definition of being inactive.

Nevertheless, we notice that the bias of encoding users' activity status into binary representation: Unable to separate highly engaged users from those who opens the module once in a week because both of them are encoded as "active" and this might limit the business value of this predictive system. For this reason, we then plan to build a user engagement scoring framework which eventually can produce a continuous rating from 0 to 1 based on various metrics that evaluate users' activities in the 7-day label period. Note that this score does not directly serve as the target to predict, and instead, five engagement levels are derived from this score; they are: **Churned**, **At-Risk**, **Regular, Engaged** and **Champion**. A detailed implementation of this rating system will be presented in the Exploratory Data Analysis and Feature Engineering.

2.4 Measuring User Engagement

As outlined in the research design, an engagement scoring framework was developed before the process of modeling, corresponding to the necessity to form multiclass labels based on their engagement on the “Cloud Village” module. Based on available behavioral data, three main perspectives are evaluated for sampled users during the label period: **Frequency**, **Intensity**, **Quality**.

Frequency measures how consistently a user accesses the platform, through the proportion of active days in the label period. This metric differentiates users who log in regularly from those who open the module only occasionally.

$$\text{Frequency} = \frac{\text{Number of active days}}{7}$$

Intensity reflects user’s overall level of content consumption usage in the module through total number of impressions during the label period, since high number of impressions indicates that a user is persistently exploring new content. This metric is normalized using 95th percentile and extreme values are capped at 1.

$$\text{Intensity} = min\left(\frac{\text{Number of impressions}}{\text{95th percentile}}, 1\right)$$

Quality measures whether a user has strong intention to meaningfully interact with content viewed, through number of key interactions such as like, share and comment. First, a weighted interaction ratio is computed based on clicks, likes, shares, and comments, which have an increasing level of feedback from the perspective of a creator. This metric is also normalized through 95th percentile.

$$\text{Quality}_{\text{raw}} = \frac{\text{Clicks} + 2 \times \text{Likes} + 3 \times \text{Shares} + 4 \times \text{Comments}}{\text{Number of impressions}}$$

$$\text{Quality} = min\left(\frac{\text{Quality}_{\text{raw}}}{\text{95th percentile}}, 1\right)$$

Through the introduction of these three dimensions, the heterogeneity in behavioral patterns among users remaining active can be preserved. For instance, a user who logs in frequently but consumes limited content will receive high score on Frequency but low on Intensity, while a user with fewer active days but heavy usage will have opposite scores. To summarize these differences into a single score, we construct the formula presented below through a weighted combination of the three metrics:

$$\text{Engagement Score} = 0.3 \times \text{Frequency} + 0.4 \times \text{Intensity} + 0.3 \times \text{Quality}$$

A higher weight is assigned to Intensity because it is the most direct representation of user's dependency on the module. All sampled users receive an engagement score based on their raw impression data in the label period following this formula. Users without any impressions are automatically assigned a score of zero. Finally, sampled users are segmented into five engagement levels using this engagement score as an intermediary metric, their definitions and score ranges are presented below in *Table 1*:

Name	**Score Range**	**Description**
Churned	0	User with no activity recorded during the label period. This group of users might have either completely churned from the platform or stopped the use of "Cloud Village" module.
At-Risk	0-0.3	Users with minimal engagement. They may open the application infrequently, consume extremely limited content and may not have valid interactions recorded during their browses.
Regular	0.3-0.5	Users with moderate and stable engagement. They tend to sustainably log in to the module, generate higher number of impressions, and have interactions from time to time.

Engaged	0.5-0.7	Users with high engagement. They usually perform well in all of the three perspectives: being active on most days, having significant content consumption and leaving feedback habitually during their browses.
Champion	0.7-1	Users with exceptionally high engagement. Such high scores indicate that they access the module frequently, consume content at volumes close to the upper bound of the active population and demonstrate strong interaction intent.

Table 1: Engagement Levels and Definitions

Figure 9 illustrates the distribution of users regarding the Engagement Score, exhibiting a clear characteristic of the long-tail effect. Such distribution also results in decreasing proportion as the engagement level goes up, shown in *Figure 10*

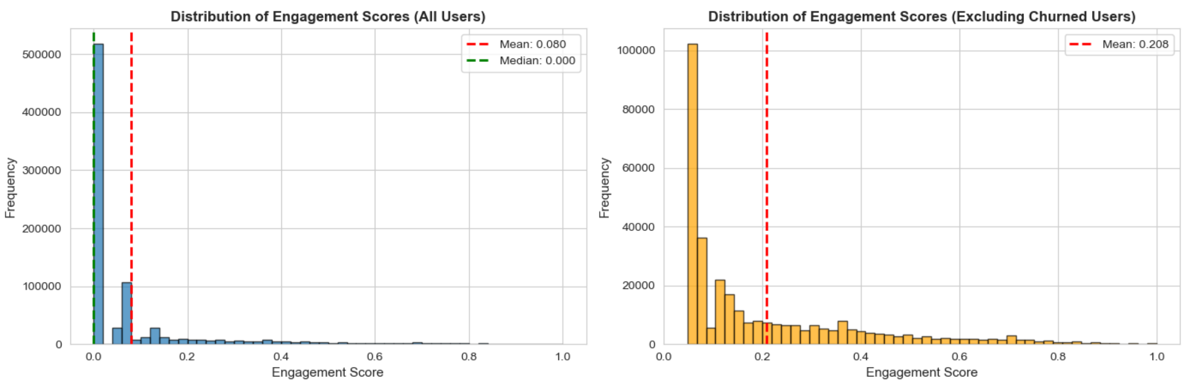

Figure 2: Distribution of Engagement Scores

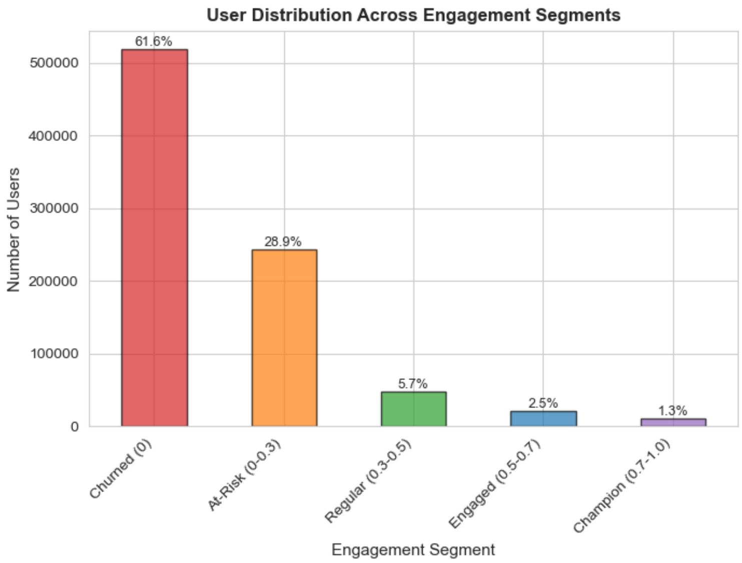

Figure 3: Proportion of Target Classes

2.5 Research Design

Aimed at delivering a feasible solution for churn predicting system on NCM platform, this study follows a commonly used prediction framework in retention focused research: Implement a fixed length time window to collect users' early behavioral patterns and then measure same group of users' subsequent activity status in a later period. Based on this design, we formulated two predictive tasks: a **Binary Churn Prediction** and a **Multi-class Engagement-level Classification**, where the latter is inherently more difficult.

To compare the effectiveness of different implementation approaches, we adopt two methodological paths: traditional machine learning models based on manually extracted KPIs, and deep learning models that can learn directly temporal patterns in the sequential data.

A detailed description of the observation window, outcome definitions, modeling framework and evaluation metrics will be provided later in this work.

3. Results & Conclusion

In the binary classification tasks, the results are presented in *Table 2*.

Classifier	ROC-AUC	Accuracy	Precision	Recall	F1
Logistic Regression	0.7305	0.7092	0.7136	0.6509	0.6529
XGBoost	0.7479	0.7156	0.7112	0.6665	0.6714
LSTM	0.7733	0.7400	0.7297	**0.7067**	**0.7129**
Hybrid LSTM	**0.7821**	**0.7441**	**0.7393**	0.7048	0.7122

Table 2: Performance of All Models on Binary Classification Task

In the multiclass classification tasks, the results are presented in *Table 3*.

Classifier	AUC (OvR)	Accuracy	Precision	Recall	F1
Logistic Regression	0.7838	0.6549	0.4574	0.3191	0.3387
XGBoost	0.8031	0.6596	0.4867	0.3555	0.3752
LSTM	0.8234	0.6803	0.5247	0.3906	0.4199
Hybrid LSTM	**0.8333**	**0.6859**	**0.5380**	**0.4118**	**0.4436**

Table 3: Performance of All Models on Multiclass Classification Task

The objective of this dissertation was to evaluate if early use by users on hybrid User Generated Content (UGC) platforms can serve as an indicator of long-term engagement and churn. Utilizing impression level data from NetEase Cloud Music's "Cloud Village" module (including 57M impressions generated by 841,670 users) we created and evaluated both machine learning and deep learning methods for both binary churn prediction and multi-class engagement classification. Overall, the findings demonstrate significant empirical evidence to support the hypothesis that the digital footprint produced by users within their first two weeks will include enough information to allow for accurate forecasting of future retention.

There were three key findings in our empirical evaluation. First, the deep learning methods utilized to model sequential data consistently demonstrated greater performance than traditional machine learning methods; the Hybrid LSTM method had a ROC-AUC of 0.7821 for binary classification and 0.8333 (One-Vs-Rest) for multi-class classification, demonstrating improved performance when compared with XGBoost (0.7479 and 0.8031) and logistic regression baselines. Second, we were able to demonstrate that a 14-day time frame is adequate to generate actionable predictions based upon behavioral data, supporting the idea that the "habits-forming" window discussed in academic literature can be implemented into retention systems. Finally, we found that the explicit modeling of temporal changes in behavior, rather than aggregating static features into a single metric, is the primary factor driving the predictive power of the LSTM model. It is capable of identifying decreasing trends in user activity and changing user interaction patterns that are obscured by aggregating metrics.

Our results have implications for both academia and practitioners. From an academic viewpoint, the results validate the central tenet of Habit Formation Theory that early-stage behavioral characteristics solidify into stable long-term trajectories. Additionally, the multi-dimensional engagement framework that combines frequency, intensity and quality of engagement supports the distinction between passive and engaged users and demonstrates that user-generated content that requires effort (e.g., sharing, commenting) has higher predictive value than passively consuming content. From a practitioner's viewpoint, the five engagement classes (Churned, At Risk, Regular, Engaged and Champion) provide a segmentation mechanism that can be embedded into CRM systems to facilitate differentiated intervention strategies during the confirmation phase of the user life cycle.

3.1 Limitations

There are several important limitations that need to be acknowledged. Firstly, due to the publicity of the dataset, the providers were very cautious in introducing information related to other modules in the whole NCM platform. According to the description, a user's total usage of the application in a day is not available, which can potentially carry strong signals to differentiate user those who stay active at minimum level because of being a heavy user of other functions in the platform, for example, music streaming. Also, the issue of distinguishing between Module Churn and Platform Churn is a common problem in analyzing sub-modules and prevents a full understanding of overall user value. Under current circumstance, people who continue to engage with the platform through music streaming but cease engagement with the social feed module are categorized as "churned" in our models, since we are only able to focus on their usage of "Cloud Village" module. With information about usage in other modules, it is possible to expand the scope of the predictive system to the platform in general.

Secondly, the limited sample size of a single month (November 2019) results in a comparatively conservative choice on the length of observation window, meaning we cannot examine the existence of seasonal patterns, long term habit formation or behavioral changes throughout longer periods. While the forecast horizon of this study (third week) was established as a valid measure of retention it is no indicator of whether a user will survive to six months or one year. Although the 14-day observation period works well in general terms, it is based on a one-size fits all approach which may not work for every type of user; some users may develop habits during their third week ("late bloomers") whereas other users may have ceased to use the platform by the end of their third day ("fast churners").

Lastly, the significant class imbalance in the multiclass scenario present in the data, particularly for the Champion segment (1.3% of users), has limited the ability to precisely categorize users to their engagement levels. Additionally, LSTM architecture lack interpretability when compared to tree-based models, presenting challenges for stakeholders seeking transparent explanations for individual predictions.

3.2 Future Research Directions

Future research could extend this research along several dimensions. Methodologically, the use of transformer-based architectures such as BERT4Rec or SASRec may capture longer-range dependencies while also providing the benefits of an attention-based model for interpretation purposes; thus, addressing both the performance and interpretability limitations of previous models. Additionally, dynamic observation windows implemented through time-to-event models or sliding window architectures, can provide a means for continuous risk scoring as opposed to predicting at a specific point in time. From the data perspective, incorporating cross-module signal (streaming history, search queries) would allow for the analysis of spillovers between social behavior and consumption behavior, resolving the module vs platform churn ambiguity.

Perhaps most importantly, future research should focus on moving past predictive models and developing prescriptive models. Randomized Controlled Trials (RCTs) for segment specific interventions would allow researchers to estimate the Heterogeneous Treatment Effects (HTE) of different types of interventions, thereby allowing the transition from risk-based targeting to lift-based targeting of users that respond best to each type of intervention. This causal inference would convert the Engagement Score from a Diagnostic Metric into a Decision Support Tool with Measurable ROI.

The methodological framework provided by this study, combining session level sequential modeling with a Multi-Dimensional Engagement Score, provides a transferable blueprint for non-contractual digital services in the Attention Economy. As Hybrid UGC Platforms become increasingly complex systems and user attention becomes the rarest resource, the ability to decode early behavioral patterns and intervene during the Habit Formation Window will continue to be a major driver of Competitive Advantage.

4. Deep Learning Approaches

In this chapter, we will break down how we tackle two kinds of classification tasks through Deep Learning (DL) models and interpret the results in a more detailed way. One important aspect to distinguish selected DL models from traditional ML models is how user behavior is represented. In traditional ML models, each user is typically represented by a single feature vector, composed of metrics calculated over the observation window. This process of extracting features from original data is usually considered lossy, due to the inevitable loss in temporal information which may be essential in predicting users' active status.

For instance, a user who showed a clear downward trend in activity intensity may end up with similar aggregate values, such as total impressions or average impressions per day, compared to a user who maintained a stable level of activity on the module. Although the problem can be alleviated via introducing more features, it is still considered as a main challenge in traditional ML approaches. To reduce this loss of potential temporal patterns, the DL models in this chapter will preserve the sequential structure of original data as much as possible, where one user is represented by a time series made of different numbers of vectors. Therefore, this allows the models to read temporal dynamics which aggregated features cannot reflect.

4.1 Data Preprocessing

4.1.1 Sessions

Building models on sequential input does not mean without any form of aggregation. Among our sampled users, the number of impressions within the 14-day window can easily pass one thousand and the highest can even reach fourteen thousand (14,832). Such long dependencies not only require strong computation power in modeling, limiting the model's scalability, but also bring noise, which may affect the final performance. Therefore, choosing an appropriate aggregation level is essential before entering modeling stage.

The aggregation level of sequential data in this study is eventually settled on Session, a level between impression and day, which is also a commonly used concept in Web analytics. According to Arlitt (2000), a general and less technical definition of session is a sequence of requests made by a single end-user during a visit to a particular site. For instance, when a user enters the “Cloud Village” module and has at least one impression, then he starts a new session. Mehrzadi and Feitelson (2012) summarized that a common practice in the industry is to set 30 minutes as the threshold to divide sessions, and we decided to follow this rule. If the interval between impression A and impression B is shorter than 30 minutes, then they belong to one session; otherwise, impression B marks the start of a new session. It’s important to note that we did not set constraints on the duration of sessions, meaning that one session can last for several hours if a user consistently has new impressions. Under this definition, the max number of sessions within 14 days is 192 and the max number of sessions within one day is 19, which significantly reduces the length of sequential data but still retains more details than aggregation based on day level.

4.1.2 Session Preparation

The core step in matching impressions with their corresponding sessions was conducted through database operations in DuckDB. First, each impression was annotated with a binary field *is_new_session*, indicating whether it started a new session within a user's chronological sequence.

Based on this marker, two types of session indexes were generated to describe the position of each session within a user's activity history: **global session index** and **daily session index**.

The global session index assigns every session a unique, incremental identifier across the entire sequence for a given user, recording absolute position of a session unit. It ranges from 1 up to the total number of sessions observed for that user. On the other hand, the daily session index records the relative position of a session within each calendar day. For each user and each day, sessions are renumbered starting from 1, allowing the model to reason about session order at a daily granularity as well.

Regarding one session unit, features such as *total_impressions*/*total_clicks*/*total_likes* etc. were aligned with the feature engineering in traditional ML. However, we explicitly introduced four positional and temporal indicators to capture temporal regularities and recency:

- *day_num*: The N-th day of the observation window.
- *week_day*: The day of a week based on a sessions' starting time.
- *hour_of_day*: The hour at which the session begins, representing intraday periodicity.
- *session_gap*: The time gap between current session and a user's last session (in minute); the first session of a user is marked as -1.

Note that periodical features are further broken down to sin/cos representations, resulting in a dimensionality of 24 features per session in the sequential input (detailed in Appendix 2). Finally, to ensure consistent tensor shapes, sequences are padded to a maximum length of 192 sessions (padding sessions receive a zero vector and a corresponding mask is used to avoid influencing the model output); and all numeric features are standardized using z-score normalization.

4.1.3 User Demographics

The user demographics features serve as a static input to complement information about individual differences between users. Numeric features such as *gender* (in one-hot representation), *age*, *registered_months* were preprocessed and inherited from traditional ML pipeline, but were then combined with additional features that provide an overview of activity intensity over the observation window particularly related to the "session" concept we introduced in previous section. For example, mean duration of a user's sessions through feature *avg_session_duration* and average number of sessions initiated per day through feature *avg_daily_sessions*. These features provide a compact summary of each user's overall engagement style and intensity

All static features are concatenated into an 18-dimensional vector, standardized via z-score normalization. The complete set of static attributes is listed in *Appendix 2*.

4.2 Long Short-Term Memory (LSTM)

4.2.1 Model Structure and Setup

The Long-Short Term Memory (LSTM) model is a refined neural network system derived from the Recurrent Neural Network (RNN) to solve the exploding or vanishing gradients problems. This ability to learn from long-term sequences enables LSTM to be very helpful and gain popularity in different fields. It has long been regarded as one of the most advanced models to process temporal sequences.

A typical LSTM architecture is composed of a series of recurrently connected components, which are recognized as memory blocks. The key to prevent exploding or vanishing gradients is the gating mechanism inside each memory block, formed by forget gate, input gate and output gate. (Van Houdt, Mosquera, & Nápoles, 2020). *Figure 4* displays the internal structure of a memory block, helping to provide a brief explanation of how the model works.

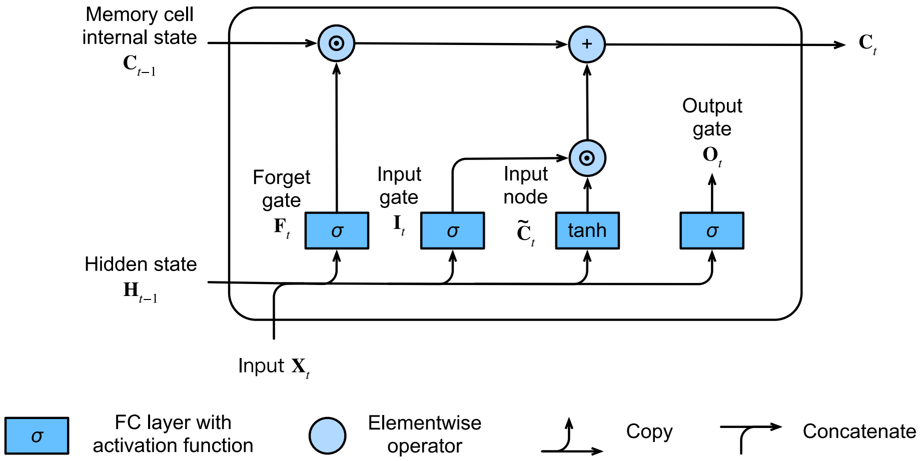

Figure 4: Structure of Memory Block

When processing step *t*, **Cell State** C_{t-1} and **Hidden State** H_{t-1} are inherited from block *t-1*:

The input node processes X_t, H_{t-1} into cell input $\hat{C}_t$ through Tanh activation function.

The input gate determines how much of the newly computed information should be written into the cell. It takes current input X_t, hidden state from last block H_{t-1} to produce a weight I_t between 0-1. This weight generated from input gate will now be applied to previously mentioned $\hat{C}_t$.

- The forget gate decides how much of the previous cell state should be preserved. Similarly, it receives X_t, H_{t-1} and produces a weight F_t, which is exclusively applied to C_{t-1}.
- The output gate determines how much of the updated cell state C_t (computed by point-wise multiplying $\hat{C}_t$ with I_t, then adding C_{t-1} multiplied by F_t) should be revealed as the hidden output for this time step. It also takes X_t, H_{t-1} and produces a weight O_t
- The output of step *t*, known as H_t, is then cell state C_t weighted by O_t. These gating mechanisms allow LSTM to maintain a stable memory over long sequences while

partially updating information.

For this reason, we would like to choose a simple LSTM model mainly composed of an LSTM layer and a Full-Connected classification head as a starting point predicting the target variable only through time series data without users' profile. This sets the foundation for testing how future enhancements such as adding complexity and incorporation of static numeric features from user's profile can help us achieve higher performance. *Figure 5* is an illustration of the model, and it is important to note that the classifier also applies normalization and dropout to the output of LSTM

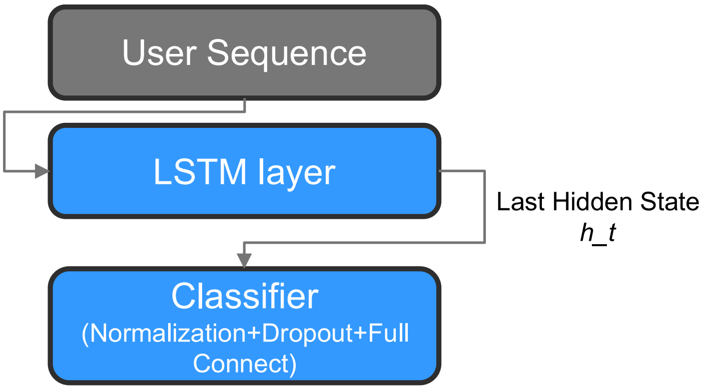

Figure 5: Structure of Basic LSTM

To provide a baseline evaluation of the LSTM's ability to capture sequential patterns, this model is trained with single layer within LSTM, a modest hidden size of 128, a dropout rate of 0.2, and 20 epochs with a batch size of 128, reflecting deliberately simple settings in terms of model complexity and training configuration. The learning rate is also set to a static value of 0.001.

4.2.2 Binary Classification Results

On the binary classification problem, this baseline LSTM model exhibited great predictive performance, achieving an AUC of 0.773 and accuracy of 0.74 on the test set. No significant difference between results on test set and validation set was observed, ruling out the probability of overfitting. Such results indicate that users' session sequences alone carry strong predictive information and it was successfully interpreted by the model.

From the examination of the classification reports, a stronger performance in inactive class was noticed, with both higher precision (0.76) and recall (0.85) leading to an F1-score of 0.8; however, in the active class while the precision (0.70) remained high, the recall (0.56) was significantly lower, dropping the F1-score to 0.62. Although the imbalance in performance can be attributed to the skewed distribution of samples (60% inactive users), it can also be interpreted that active users have more heterogeneous and less easily captured temporal patterns. The confusion matrix of prediction results on test set is presented in *Figure 6.*

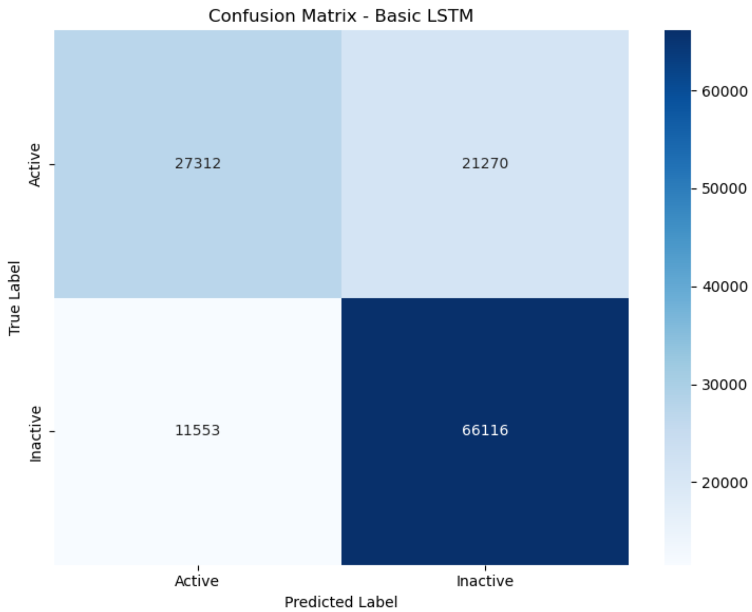

Figure 6: Confusion Matrix of Basic LSTM

Overall, the prediction results can be considered credible for achieving high precision in both classes, and the model performs well in capturing inactive users, while the ability to capture active users still has potential to be improved. These results show that the simple LSTM structure is effective in extracting meaning signals from user sequences made of our previously defined session units, setting a solid foundation for other supportive enhancements.

4.2.3 Multiclass Classification Result

The multiclass classification model followed same setup as the binary one, while being fitted to a different target set composed of five engagement levels. As expected, the performance declined due to the increased number of categories. This model achieved an average OvR AUC of 0.824 and an overall accuracy of 0.68. However, compared to the binary classification model, its F1-score dropped to 0.42.

The influence of class imbalance was magnified. Still, the dominant Class 0 (inactive) showed superior performance compared to others, with a precision of 0.72 and recall of 0.92 but the separability between other four subdivided classes was not explicitly demonstrated. Among active users, the class with comparatively best performance was Class 4 (champion), with precision of 0.52 and recall of 0.41 resulting in an F1-score of 0.46. This class represents users who are heavy users of the module during label period, and we noticed the OvR AUC of this class even reached 0.948, implying that this user group tend to have easily learnable behavioral patterns.

On the contrary, Class 2 (regular) showed the weakest performance, with recall is dropping to 0.09 due to often being categorized into class 0 and 1. This might reflect that users in this group usually have strong overlap in behavioral patterns with moderately active users. *Figure 7* presents the recall, precision and F1-score of each class in bar chart.

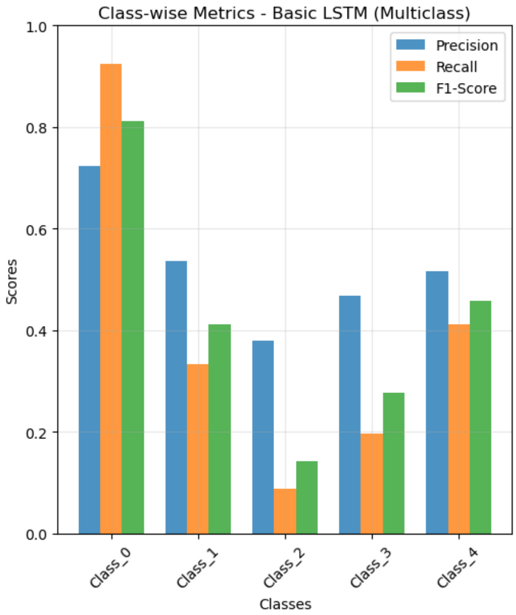

Figure 7: Class-wise Metrics of Basic LSTM

Taken together with the results of binary classification, the low discrimination among active users suggests that although LSTM structure alone has good efficiency in separating active and inactive users, the sequential data provided might not contain sufficient discriminatory signals to further categorize users into fined engagement levels.

4.3 Hybrid LSTM

In this section, we will introduce methods to further contribute to the performance of LSTM-based models, including Attention mechanism and Multiple Layer Perceptron (MLP) processor for incorporating static numeric features. We then evaluate how these additions influence predictive performance through quantitative comparison.

4.3.1 Attention Module

The baseline LSTM structure can still be improved because it relies solely on the last hidden state as the input to the classifier module. When the sequence is long, this design causes earlier time steps to have limited influence on the final output due to the gating mechanisms. In our dataset, the sequences can have a maximum length of 192. Although sessions near the end of the observation window may often play a more decisive role in predicting churn, the sessions occurring in the middle could also mark important shifting points in a user's activity intensity that the model should not overlook.

To address this limitation, we introduced the attention mechanism that provides a way for the model to automatically learn which time steps have higher or lower importance on the final output. It introduces a learnable weight matrix to compute a similarity score between the last hidden state, known as the query, and the hidden states from all previous time steps, known as the keys and values. Finally, these scores are normalized through a Softmax function and used to calculate a weighted sum of all time steps, forming a context vector that serves as the input to the classifier. This ensures that the model makes use of information across the entire sequence.

Compared to using a bidirectional structure or switching to a more complex sequence-based model, the attention mechanism offers a lighter solution that is particularly designed for more efficiently processing temporal patterns without significantly increasing model complexity.

4.3.2 MLP Module

Since the static numerical features are separated from the sequential input in our DL framework, the LSTM cannot directly utilize static user attributes, treating all individuals identically when processing the sequences. Moreover, relying solely on the sequence encoder may limit the model's ability to incorporate aggregate information that summarizes the full observation window. To address this limitation, we introduced a lightweight MLP module that serves as a simple feature extractor for the user demographics data. This latent vector produced by this module is then concatenated with the sequence representation produced by the LSTM before going into the classifier, which allows it to learn individual differences and make predictions based on both dynamic behavioral patterns and stable profile characteristics.

4.3.3 Model Structure and Setup

Combining the attention and MLP module, *Figure 8* displays an illustration of our hybrid LSTM model.

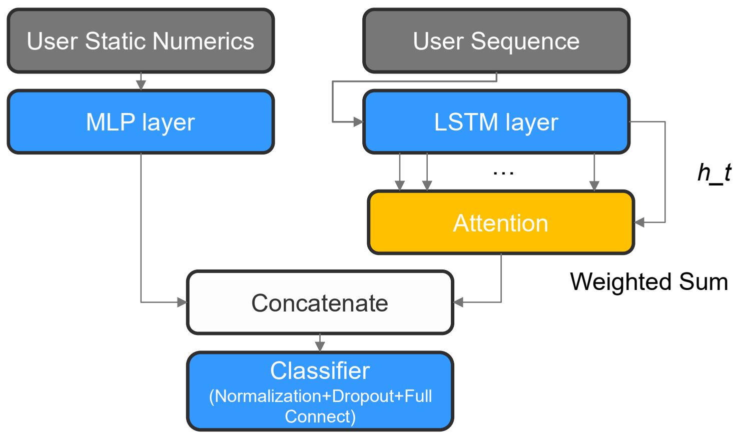

Figure 8: Structure of Hybrid LSTM

The MLP branch on the left contains two fully connected layers with ReLU activation, transforming the static numerical features into a 32-dimensional latent vector. The additional attention module on the right then produces a context vector using a weighted sum of all hidden states, providing a more accurate sequence representation. Notably, although the incorporation of results from both sides is completed through concatenation, the normalization layer inside classifier helps balance the contribution of each side.

Intending to further extend this model's performance, this time the number of layers within LSTM was set to 2, with hidden size maintained at 128 and drop rate at 0.2. In the training process, we applied learning rate scheduler to ensure that the model fully converged: Starting at a higher value of 0.001, the rate is halved if validation loss fails to improve by 0.0001 after four consecutive epochs. Moreover, total epochs were set to 50 with early-stopping mechanism. If the validation loss does not improve by at least 0.0001 after seven consecutive epochs, training terminates and best-performing checkpoint is saved based on validation loss.

4.3.4 Binary Classification Results

For binary classification, the training terminated at epoch 28. We then again measured its performance on same test set. Regarding the core metric AUC score, it reached 0.782 (+0.009); and the accuracy reached 0.744 (+0.004). Although the numerical improvement over the basic LSTM is modest, the enhanced model had clearer decision boundaries and greater reliability. From the classification report, we found that the ability to detect inactive users was strengthened, with recall increasing from 0.85 to 0.88 (+0.03) and precision maintaining at 0.75. The difference suggests that incorporating user demographics helps the model identify more users who turned inactive during label period, possibly because their static features contain more distinguishable information (e.g., as missing age and gender). However, the imbalance continued to exist, the active class presented a trade-off between improved precision of 0.73 (+0.03) and slightly decreased recall of 0.53 (-0.03). In the confusion matrix shown in *Figure 9*, it is observed that less actual inactive users were misclassified into active, explaining the increase in precision of active class and recall of inactive class.

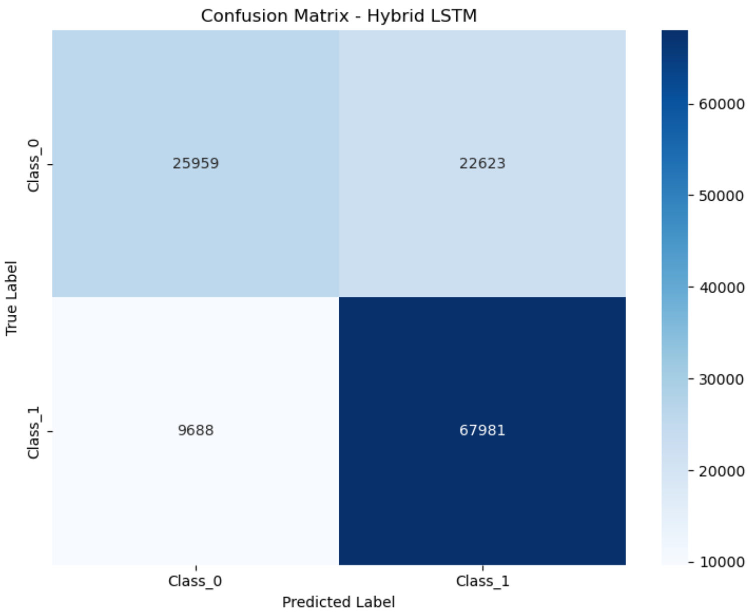

Figure 9: Confusion Matrix of Hybrid LSTM

Overall, the hybrid LSTM did not have dramatic increase in main performance metrics. The main predictive power still lies in the temporal patterns, yet the attention and static inputs play a supplementary role in refining prediction behavior.

4.3.5 Multiclass Classification Result

Similarly, the additional modules achieved modest increase on OvR AUC, from 0.823 to 0.8333 (+0.01) and also accuracy, from 0.68 to 0.685 (+0.05) in the multiclass setting. Besides, the macro-averaged F1 score increased from 0.42 to 0.443 (+0.023), indicating that the model benefited from incorporating static features and the attention mechanism. *Figure 10* presents a side-by-side comparison of class-wise metrics between the basic and hybrid models.

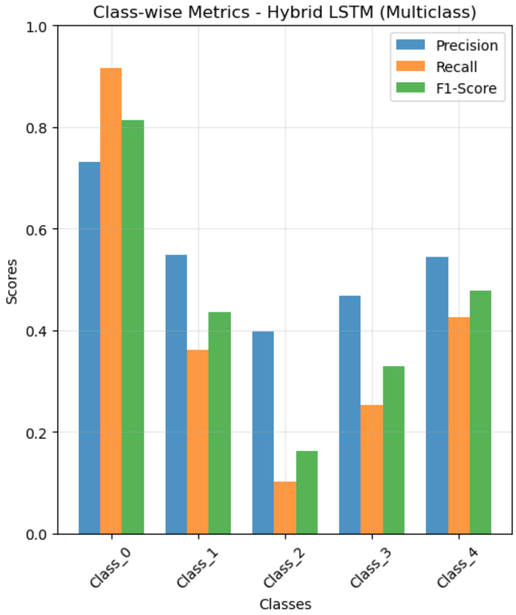

Figure 10: Class-wise Metrics of Basic LSTM

The effect was mainly reflected in users that remained active (Class 1 to Class 4). These classes gained increases both in precision and recall, resulting in slightly higher F1 scores. For example, in Class 1 F1 score improved from 0.41 to 0.44, and in Class 3 from 0.28 to 0.33. However, the relative ranking among these classes remained unchanged: Class 2 (regular) continued to exhibit the weakest performance, while Class 4 (Champion) remained the strongest.

Compared with the binary classification results, although the additional modules still play a supporting role, their contribution in the multiclass setting is more evident, as all classes benefited simultaneously and no "tradeoff" phenomenon was observed. Meanwhile, the unchanged performance hierarchy across Class 1–4 might indicate that static demographic features alone are insufficient to resolve the inherent overlap within these user groups.

4.4 Summary

Although improvement is moderate, in both classification problems the Hybrid LSTM model enhanced by attention and MLP performed better. We believe that these enhancements will systematically complement the model's robustness when implemented on large scale for their unique theoretical background without significantly increase in computation power required. For the binary classification problem, the final model presented an AUC score of 0.782 and an accuracy of 0.744, offering great flexibility in choosing appropriate threshold value in implementation. Since the model did not suffer from serious imbalance proportion in data, precision in both classes passed 0.7, which indicates good credibility of classification results. The staring part is that it shows strong ability to capture users turning inactive for a recall close to 0.9, while presenting the average ability on user remaining active. Based on these characteristics, we consider the model to be applicable, offering feasible instructions on platform's operations. However, the design of deriving strategies can be influenced by relatively lower recall rate in the active class, which will be discussed in the later part of this work.

For the multiclass classification problem, although the final model presented an OvR AUC of 0.8333 and accuracy of 0.685, its performance is heavily affected by amplified imbalance in proportion. Due to subdivision based on engagement scores, each class of users remaining active only occupied limited proportion in the dataset. The higher engagement scores, the lower number of users, which makes it very challenging for the model to fully absorb their difference. Through analysis of confusion matrix, it is found that users with low engagement scores, especially in Class 1 and 2, are often misclassified as fully inactive, suggesting strong overlap in their temporal patterns. The optimization on these classes in the final model, though not substantial, still indicates that supplementing the model with external information along with temporal patterns is a potential direction if aiming for further improvement. Overall, the performance on multiclass classification does not support the model to have an central role in decision making.

Finally, this chapter demonstrates that while sequential modeling is effective for broad churn prediction, achieving reliable segmentation across multiple engagement levels likely requires integrating richer behavioral signals or alternative modeling strategies beyond LSTM-based architectures.

References

Arlitt, M. (2000). Characterizing Web user sessions. *ACM SIGMETRICS Performance Evaluation Review*, 28(2), 50–63. https://doi.org/10.1145/362883.362920

Ascarza, Eva. 2018. "Retention Futility: Targeting High-Risk Customers Might Be Ineffective." *Journal of Marketing Research* 55 (1): 80–98. https://doi.org/10.1509/jmr.16.0163.

Bauer, Kevin, Moritz von Zahn, and Oliver Hinz. 2023. "Expl(AI)Ned: The Impact of Explainable Artificial Intelligence on Users' Information Processing." *Information Systems Research* 34 (4): 1582–602. https://doi.org/10.1287/isre.2023.1199.

Bhattacharjee, Shamik, Utkarsh Thukral, and Nilesh Patil. 2023. *Early Churn Prediction from Large Scale User-Product Interaction Time Series*. September 25. http://arxiv.org/abs/2309.14390.

Bhattacherjee, Anol. 2001. "Understanding Information Systems Continuance: An Expectation-Confirmation Model." *MIS Quarterly* 25 (3): 351–70. https://doi.org/10.2307/3250921.

Buckinx, Wouter, and Dirk Van den Poel. 2005. "Customer Base Analysis: Partial Defection of Behaviourally Loyal Clients in a Non-Contractual FMCG Retail Setting." *European Journal of Operational Research* 164 (1): 252–68. https://doi.org/10.1016/j.ejor.2003.12.010.

Chen, Zhen-Yu, Zhi-Ping Fan, and Minghe Sun. 2012. "A Hierarchical Multiple Kernel Support Vector Machine for Customer Churn Prediction Using Longitudinal Behavioral Data." *European Journal of Operational Research* 223 (2): 461–72. https://doi.org/10.1016/j.ejor.2012.06.040.

Gao, Ge, Hongyan Liu, and Kang Zhao. 2023. "Live Streaming Recommendations Based on Dynamic Representation Learning." *Decision Support Systems* 169 (June): 113957.

https://doi.org/10.1016/j.dss.2023.113957.

Hadiji, Fabian, Rafet Sifa, Anders Drachen, Christian Thurau, Kristian Kersting, and Christian Bauckhage. 2014. "Predicting Player Churn in the Wild." *2014 IEEE Conference on Computational Intelligence and Games*, August, 1–8. https://doi.org/10.1109/CIG.2014.6932876.

Hoang, Ha Dang, and Nguyen Tan Cam. 2025. "Do They like Your Game? Early-Stage Churn Prediction Using a Two-Phase Neural Network System." *Engineering Applications of Artificial Intelligence* 144 (March): 110102. https://doi.org/10.1016/j.engappai.2025.110102.

Huang, Ni, Gordon Burtch, Bin Gu, et al. 2019. "Motivating User-Generated Content with Performance Feedback: Evidence from Randomized Field Experiments." *Management Science* 65 (1): 327–45. https://doi.org/10.1287/mnsc.2017.2944.

Katona, Zsolt, Peter Pal Zubcsek, and Miklos Sarvary. 2011. "Network Effects and Personal Influences: The Diffusion of an Online Social Network." *Journal of Marketing Research* 48 (3): 425–43. https://doi.org/10.1509/jmkr.48.3.425.

Lemmens, Aurélie, and Sunil Gupta. 2020. "Managing Churn to Maximize Profits." *Marketing Science* 39 (5): 956–73. https://doi.org/10.1287/mksc.2020.1229.

Limayem, Moez, Sabine Gabriele Hirt, and Christy M. K. Cheung. 2007. "How Habit Limits the Predictive Power of Intention: The Case of Information Systems Continuance." *MIS Quarterly* 31 (4): 705–37. https://doi.org/10.2307/25148817.

Mehrzadi, D., & Feitelson, D. G. (2012). On extracting session data from activity logs. *SYSTOR '12: Proceedings of the 5th Annual International Systems and Storage Conference*, 1–7. https://doi.org/10.1145/2367589.2367592

Mienye, I. D., & Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, 10, 99129–99149.

Dhttps://doi.org/10.1109/access.2022.3207287

Rudin, Cynthia. 2019. "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead." *Nature Machine Intelligence* 1 (5): 206–15. https://doi.org/10.1038/s42256-019-0048-x.

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, 53(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Verbeke, Wouter, Karel Dejaeger, David Martens, Joon Hur, and Bart Baesens. 2012. "New Insights into Churn Prediction in the Telecommunication Sector: A Profit Driven Data Mining Approach." *European Journal of Operational Research* 218 (1): 211–29. https://doi.org/10.1016/j.ejor.2011.09.031.

Wang, Tong, Cheng He, Fujie Jin, and Yu Jeffrey Hu. 2022. "Evaluating the Effectiveness of Marketing Campaigns for Malls Using a Novel Interpretable Machine Learning Model." *Information Systems Research* 33 (2): 659–77. https://doi.org/10.1287/isre.2021.1078.

Zhang, D., Hu, M., Liu, X., Wu, Y., & Li, Y. (2020). NetEase Cloud Music Data. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3554826

Appendix 1: Dataset provided by Zhang et al. (2020)

Table Name: impression_data.csv		
Field Name	**Data Type**	**Description**
userId	string	The unique identifier of each user in the data set
dt	numeric	The number of days from the start of the sample period
mlogId	string	The unique identifier of each card in the data set
impressTime	numeric	The epoch time of the impression
impressPosition	numeric	The position of impression in the feed
isClick	binary	1 if the user clicks on the card, 0 otherwise
isComment	binary	1 if the user comments on the card, 0 otherwise
isLike	binary	1 if the user likes the card, 0 otherwise
isShare	binary	1 if the user shares the card, 0 otherwise
isViewComment	binary	1 if the user views comments from other users on the card, 0 otherwise
isIntoPersonalHomepage	binary	1 if the user enters the creator's homepage through the card, 0 otherwise
mlogViewTime	numeric	The number of seconds that the user has spent on the card
detailMlogInfoList	string	JSON file contains all cards that the user sees if s/he swipes down

Table Name: user_demographics.csv		
Field Name	**Data Type**	**Description**
userId	string	The unique identifier of each user in the data set
province	string	The province in Pinyin that this user comes from
age	numeric	The predicted age of the user
gender	string	The predicted gender of the user
registeredMonthCnt	numeric	The number of months between a user's registration time and December 1st, 2019
followCnt	numeric	The number of people a user has followed till December 1st, 2019
level	numeric	The activity intensity level of a user ranging from 0 to 10

Table Name: mlog_demographics.csv		
Field Name	**Data Type**	**Description**
mlogId	string	The unique identifier of each card in the data set.
songId	string	The unique identifier of each song in the data set.
artistId	string	The unique identifier of each artist of a song in the data set.
creatorId	string	The unique identifier of each creator of a card in the data set.
publishTime	numeric	The number of days when the card is published till December 1st, 2019
type	binary	1 if the card contains a set of images and text with background music, 2 if the card contains a music video
contentId	numeric	The anonymized type of a card's content with 122 unique levels
talkId	numeric	The anonymized topic of a card with 9; 914 unique levels

Table Name: mlog_stats.csv		
Field Name	**Data Type**	**Description**
mlogId	string	The unique identifier of each card in the data set.
dt	numeric	The number of days from the start of the sample period
userImpressionCount	numeric	The number of unique users the card was shown to for a given date
userClickCount	numeric	The number of unique users who clicked on the card for a given date
userLikeCount	numeric	The number of unique users who liked on the card for a given date
userCommentCount	numeric	The number of unique users who commented on the card for a given date
userViewCommentCount	numeric	The number of unique users who viewed others' comments on the card for a given date
userShareCount	numeric	The number of unique users who shared the card for a given date
userIntoPersonalHomepageCount	numeric	The number of unique users who entered the creator's homepage from this card for a given date
userFollowCreatorCount	numeric	The number of unique users who followed the creator from the card for a given date

Table Name: creator_demographics.csv		
Field Name	**Data Type**	**Description**
creatorId	string	The unique identifier of each creator of a card in the data set.
gender	string	The predicted gender of the creator, which can be unknown or NA.
registeredMonthCnt	numeric	The number of months between this creator's registration time and December 1st, 2019
follows	numeric	The number of people a creator has followed on November 1st, 2019
followeds	numeric	The number of followers a creator has on November 1st, 2019
creatorType	numeric	The anonymized type of a creator with 10 levels
level	numeric	The activity intensity level of a creator ranging from 0 to 10

Table Name: creator_stats.csv		
Field Name	**Data Type**	**Description**
creatorId	string	The unique identifier of each creator of a card in the data set.
dt	numeric	The number of days from the start of the sample period
PublishMlogCnt	numeric	The number of cards this creator has created for a given date

Appendix 2: Feature Engineering – Deep Leaning

Table Name: dl_time_series_sessions_14d.csv		
Field Name	**Data Type**	**Description**
userId	string	The unique identifier of each user in the data set.
global_session_idx	numeric	The indicator of a session's position among a user's all sessions (ranging from 1 to 192).
daily_session_idx	numeric	The indicator of a session's position among a user's sessions within a day (ranging from 1 to 19).
day_num	numeric	N-th day (1-14) in the observation window.
is_bew_day	binary	The marker of a new day.
session_duration	numeric	The duration of a session in minutes.
session_gap	numeric	The time gap between current session and a user's last session (in minutes), for the first session of a user it is marked as -1.
hour_of_day	numeric	The hour part of a session's starting time.
week_day	numeric	The day in a week according to the session's starting time.
max_impress_poisition	numeric	The max position of an impression within a session.
total_impression	numeric	The total number of impressions during a session.
total_clicks	numeric	The total number of clicks during a session.
total_likes	numeric	The total number of likes during a session.
total_shares	numeric	The total number of shares during a session.
total_comments	numeric	The total number of comments during a session.
total_into_personal_pages	numeric	The total number of going into personal homepages during a session.
total_view_comments	numeric	The total number of viewing comments during a session.
ctr	numeric	The impression Click-Through Rate during a session.

interaction_rate	numeric	The proportion of impressions that triggered at least one type of action besides click during a session.
like_rate	numeric	The proportion of impressions that triggered like among **clicked** impressions during a session.
share_rate	numeric	The proportion of impressions that triggered share among **clicked** impressions during a session.
comment_rate	numeric	The proportion of impressions that triggered comment among **clicked** impressions during a session.

Table Name: dl_static_numeric_14d.csv		
Field Name	**Data Type**	**Description**
userId	string	The unique identifier of each user in the data set
province	string	The province in Pinyin that this user comes from
age	numeric	The predicted age of the user
is_age_missing	binary	The indicator of missing age information through 0 and 1
gender	string	The predicted gender of the user
registeredMonthCnt	numeric	The number of months between a user's registration time and December 1st, 2019
followCnt	numeric	The number of people a user has followed till December 1st, 2019
active_days	numeric	The total number of days that the user is active (has at least one impression) during the input time range.
total_sessions	numeric	The total number of sessions, equal to the length of a user's sequence.
avg_daily_sessions	numeric	The average number of sessions per active day.
avg_session_impressions	numeric	The average number of impressions per session.
avg_session_duration	numeric	The average session duration in minutes.
avg_session_clicks	numeric	The average number of clicks per session.

avg_session_mlog_watchtime	numeric	The average time spent on watching Mlog per session in minutes.
avg_session_likes	numeric	The average number of likes per session.
avg_session_shares	numeric	The average number of shares per session.
avg_session_comments	numeric	The average number of comments per session.
user_ctr	numeric	The global Click-Through Rate for a user during the input time period.