Imbalanced User Churn Prediction in Music Streaming Platforms: An MLP-Based Modeling Approach

Abstract

For music streaming platforms, accurately identifying users at risk of churn is critical, as they rely heavily on subscriptions. In this study, I develop a Multi-Layer Perceptron (MLP) to predict churn based on user logs, transactions and demographic features. To improve generalisation, I apply a semi-supervised learning (SSL) approach using pseudo-labelling on unlabelled data. Compared to benchmark models, MLP performs similarly to LightGBM, and substantially outperforms GLM in terms of F1-score and AUC. When incorporating SSL, the model shows a notable improvement in recall and F1-score for churn prediction, despite the overall AUC remaining similar. These results highlight the potential of neural networks for churn prediction, while also underscoring the practical challenges of applying semi-supervised learning in real-world imbalanced classification tasks.

1 Introduction

Churn prediction is a critical business problem for music streaming platforms such as KKBOX, Spotify, and Apple Music. Even a small fluctuation in user churn can significantly affect their revenue income, as these platforms rely heavily on paid subscriptions and sustained user engagement. Consequently, it is essential for them to accurately identify users who are likely to stop subscriptions, enabling the implementation of proactive strategies such as targeted promotions and personalised retention efforts.

This project develops a machine learning pipeline for user churn prediction using real-world data from Kaggle. The most critical challenge in this dataset is the severe class imbalance, where users who churn account for only a small fraction of the overall user base. This skews supervised models toward the majority class, leading to deceptively high accuracy while missing true churn cases—those of greatest business interest. Moreover, the high dimensionality of the feature increases the risk of overfitting, while the presence of numerous missing demographic values may introduce bias—particularly if the missingness is not random.

To address these challenges, I start with preprocessing, including feature engineering, categorical encoding, and standardization. The baseline MLP suffered from severe class imbalance, which was solved through a combination of resampling techniques and focal loss. Then, a Multi-Layer Perceptron (MLP) was trained and fine-tuned as the primary predictive model. To enhance generalization, I incorporated a semi-supervised learning (SSL) strategy based on unlabelled user data.

The performance of the MLP models was compared against two benchmarks: an interpretable logistic regression model (GLM) and the widely adopted LightGBM. A range of performance metrics—including AUC, F1 score, recall, precision, and accuracy—was used to provide a comprehensive evaluation of predictive effectiveness.

The remainder of this paper is organized as follows. Section 2 describes the dataset, including data sources, preprocessing steps, and feature construction procedures. Section 3 presents the overall methodology and implementation details, covering MLP, SSL and benchmark models. The construction of benchmark models is also included in this section. Section 4 reports the experimental results for all models. Section 5 concludes with a discussion of key findings, limitations, and directions for future work.

2 Literature Review

Customer churn prediction has been extensively studied across industries, particularly in telecommunications, where retaining existing users is both cost-effective and critical to profitability (Ahmed & Linen, 2017). Traditional approaches such as logistic regression, decision trees, and support vector machines have been widely applied (Prabadevi et al., 2023), but recent studies suggest that neural networks and hybrid models offer superior predictive accuracy.

MLP models have been shown to be effective for churn prediction tasks. Adwan et al. (2014) explored MLPs with backpropagation in the Jordanian telecom sector, and identifed the most influential churn factors. Similarly, Ismail et al. (2015) demonstrated the superiority of MLPs over classical statistical models in a Malaysian telecom dataset, achieving 91.28% prediction accuracy.

SSL methods also shows promise, especially when labeled data is limited. Rani and Kant (2020) proposed a pseudo-labeling approach that improves baseline classifiers with large volumes of unlabeled data, achieving accuracy as high as 99.62%. Liu et al. (2018) extended SSL into the mobile gaming domain, developing an embedding model that jointly learns user-app interaction representations and churn probabilities from large-scale behavioural data.

Together, these studies highlight the evolving landscape of churn prediction research and point to the importance of algorithmic design in building robust prediction systems.

3 Data

3.1 Data Source

This study is based on the KKBox churn prediction[1] dataset, originally released for the WSDM Cup. It contains detailed user logs, subscription transaction history, and demographic information from KKBox, a subscription-based music streaming platform.

For the purpose of semi-supervised learning (SSL), I redefine the dataset splits as follows: users from the original training files—whose subscriptions expired in February 2017—are used as labelled data. In contrast, users in the original test set, whose subscriptions expired in March 2017, serve as the unlabelled set.

In this context, churn is defined as a user not initiating a new subscription within 30 days after their membership expiration date.

3.2 Data Preprocessing

The dataset includes a large number of features of diverse aspects. While offering valuable signals for understanding user churn, it also increases the risk of overfitting and computational costs. Besides, due to user privacy choices or incomplete data capture, there are a lot of missing values related to user demographics. If left unaddressed, these missing values can introduce bias into the learning process—particularly if the missingness is not random.

Prior to model development, I transformed the high-dimensional raw data into a set of interpretable features. For daily user log data, I computed aggregated statistics such as average and total listening time, play counts across completion thresholds, and user engagement frequency within a defined time window. In the transaction data, I derived indicators that reflect users' subscription stability and pricing sensitivity, such as changes in subscription plans, and discrepancies between list prices and actual charges.

To improve the quality of demographic information, I addressed both outliers and missing values. Implausible age entries were removed and replaced with the population mean. Missing categorical variables were imputed either with a placeholder category (`Unknown`) or the mode within the corresponding age group.

This process reduced redundancy, mitigated sparsity, and improved model interpretability.

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

3.3 Final Dataset Summary

After preprocessing, the final dataset consists of 68 features, with 970,960 labelled users and 907,471 unlabelled users.

4 Methodology

To address the churn prediction task, I adopt a Multi-Layer Perceptron (MLP) as the core predictive model, improved with a semi-supervised learning (SSL) pipeline. I also compare model performance against two benchmarks: Generalized Linear Model (GLM) and LightGBM.

4.1 Multi-Layer Perceptron Pipeline

MLP is a feedforward neural network that transforms input features through multiple layers to make predictions. It captures complex patterns using hidden layers and outputs a probability through a sigmoid function for binary classification tasks like churn prediction.

In this study MLP is implemented using PyTorch, with a full workflow consisting of several modular steps designed.

4.1.1 Data Preparation

Input data were preprocessed through categorical encoding and numerical standardization, followed by conversion to PyTorch tensors. A stratified split was applied, resulting in a training set of 776,768 users and a test set of 194,192.

4.1.2 MLP Implementation and Training

The MLP architecture was implemented using a custom `MLPClassifier`class, which allows flexible configuration of hidden layers, dropout rates, and ReLU activation functions. The model was trained using binary cross-entropy loss and the Adam optimizer. To mitigate overfitting, early stopping was applied based on the validation loss. The best-performing model on the validation set was restored after training.

In this study, four variants of MLP are explored to find the most effective configuration.

a. Baseline MLP

The baseline model was trained on the original imbalanced dataset without any resampling or loss adjustment. It used the `BCEWithLogitsLoss` as the objective function and served as a reference point for evaluating subsequent improvements.

b. Resampled MLP

This model incorporated a hybrid resampling strategy, which generates synthetic minority class samples (via SMOTE) while randomly removing majority class samples. It can address class imbalance and

reduce overfitting risk. Several sampling configurations were tested (`smote_ratio` $\in \{0.3, 0.5, 0.7\}$ and `target_ratio` $\in \{0.6, 0.8, 1.0\}$), and the best performance was observed with `smote_ratio = 0.7` and `target_ratio = 0.8`.

c. Focal Loss MLP

Instead of modifying the data distribution, this variant adjusted the loss function to `Focal Loss`. **Focal Loss** down-weights easy examples and focuses learning on hard, misclassified cases, particularly effective for handling class imbalance during training. A grid search over $\alpha \in \{0.25, 0.5, 1.0\}$ and $\gamma \in \{1, 2, 3\}$ was conducted, and the best combination was found to be $\alpha = 0.5$, $\gamma = 1$.

d. Combined MLP

This variant combined moderate resampling with Focal Loss, aiming to reduce noise from oversampling while still improving minority class recall via the customised loss function.

According to 1, the combined model showed promising performance and was selected for further fine-tuning. Although the test set was used instead of a validation set here, the potential risk of information leakage is minimal due to the further tuning of multiple hyperparameters.

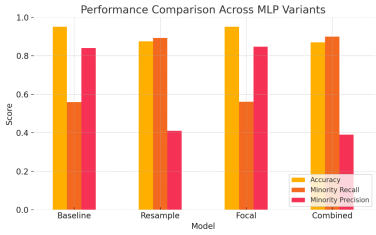

Figure 1: The performance across MLP variants

4.1.3 Fine-Tuning and Final Training

To tune the hyperparameters, a randomized grid search was conducted using stratified 3-fold cross-validation, which balances the class distribution across folds. The search was run 12 times over a parameter grid covering the number of hidden units in each layer, the dropout rate, and the learning rate.

Each configuration was trained for up to 20 epochs with a batch size of 256. Early stopping was employed with a patience of 5 epochs based on validation loss. The F1-score was used as the primary evaluation metric in the training data. All experiments were performed on CPU.

The search identified the optimal configuration as `hidden_dims` $= [256, 128, 64]$, `lr` $= 0.0005$, and `dropout_rate` $= 0.1$.

I retrained the model on the full training dataset using these optimal values. To further improve classification performance, the decision threshold is optimized based on the F1-score. The optimal threshold was determined to be 0.68, for it balances precision and recall under class imbalance, as illustrated in Figure 2. The final_model with optimal threshold is saved for later evaluation and comparison.

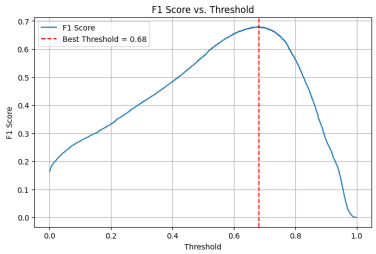

Figure 2: F1-score as a function of decision threshold

4.2 Semi-Supervised Learning Pipeline

To enhance the generalization capability of the MLP model, I adopt a semi-supervised learning approach based on pseudo-labelling.

This process starts with inferring churn probabilities on the unlabeled dataset using the previously trained MLP model. Only samples with high prediction confidence (probability ≥ 0.9 or ≤ 0.1) were selected, under the assumption that these predictions were more likely to be correct.

The selected pseudo-labelled samples were then appended to the original labelled training set to form an expanded training dataset. The same MLP architecture was subsequently retrained on this augmented dataset. Specifically, the size of the unlabeled data was 907,471, from which 452,092 pseudo-labeled samples were selected. The final augmented training set thus contained a total of 1,228,860 instances.

4.3 Benchmark Models Pipeline

To assess the performance of the MLP and SSL-MLP models, I employed two benchmark machine learning models. Logistic regression was used due to its simplicity and interpretability. LightGBM, a gradient boosting framework based on decision trees, was selected for its ability to handle large-scale, high-dimensional data. It has been widely adopted in the industry for churn prediction and similar classification tasks. All models were trained on the same preprocessed and balanced dataset as the MLP to ensure consistency in input distributions.

4.4 Evaluate

Model performance was evaluated on a held-out test set using a consistent set of metrics. Specifically, F1-score, area under the ROC curve (AUC), precision, and recall are reported, which together provide a comprehensive assessment

of each model's classification capability across different decision thresholds.

5 Result

Table 1 summarizes the overall performance of all models on the test set. MLP and SSL-MLP perform similarly, with accuracy scores around 94%, and AUC values around 0.95, suggesting that pseudo-labeling had a marginal impact under the current configuration.

Table 1: Overall model performance on the test set.

Model	Accuracy	Macro F1	AUC
MLP	0.95	0.82	0.9553
SSL-MLP	0.93	0.81	0.9545
GLM	0.82	0.67	0.9101
LightGBM	0.94	0.82	0.9530

Tables 2 and 3 present class-specific metrics. For class 0 (non-churn), all models attain high precision and recall.

Table 2: Performance on class 0 (non-churn users).

Model	Precision	Recall	F1-score
MLP	0.96	0.98	0.9700
SSL-MLP	0.98	0.94	0.9600
GLM	0.98	0.81	0.8900
LightGBM	0.96	0.97	0.9700

For class 1 (churn), performance varies more substantially. LightGBM and MLP achieve the highest F1-score of 0.67, followed by SSL-MLP with 0.65, reflecting a strong balance between precision and recall. GLM shows the highest recall (0.86), but its precision (0.31) is too low for reliable predictions.

Table 3: Performance on class 1 (churn users).

Model	Precision	Recall	F1-score
MLP	0.73	0.62	0.67
SSL-MLP	0.57	0.77	0.65
GLM	0.31	0.86	0.46
LightGBM	0.71	0.63	0.67

Figure 3 intuitively illustrates the classification performance of different models. The left panel shows the Precision-Recall curves, where both the MLP and SSL-MLP models achieve the highest area under the curve of 0.761. LightGBM follows closely, while GLM performs significantly worse.

The right panel presents the ROC curves, which assess overall classification performance. Here, both MLP and SSL-MLP again attain the highest AUC, closely followed by LightGBM. These results show the robustness of MLP-based models, while also highlighting the effectiveness of LightGBM. The similarity between the MLP and SSL-MLP curves suggests that pseudo-labeled data did not significantly alter the model's decision boundary, but effectively preserving its overall performance.

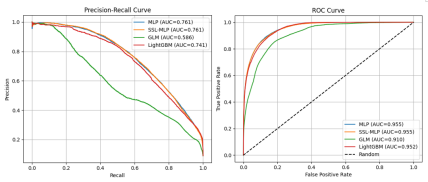

Figure 3: Comparison of model performance using Precision-Recall and ROC curves

Overall, MLP-based models demonstrate strong performance across both classes. LightGBM performs competitively, while GLM suffers from minority class prediction. Notably, the SSL-MLP model improves minority class recall without severely harming overall performance, validating the value of pseudo-labeling.

6 Discussion

This study demonstrates that deep learning models, particularly MLPs, outperform linear models in predicting user churn, while LightGBM remains a strong benchmark for structured user data.

However, several aspects did not perform as expected. The SSL-enhanced MLP showed limited improvement over the supervised benchmarks, likely due to the small volume of unlabeled data and noise introduced by pseudo-labels. In practice, semi-supervised learning tends to yield better results when supported by large-scale unlabeled datasets. Additionally, the current resampling strategy may have been less effective. The model's ability to distinguish between the majority and minority classes remains imbalanced, suggesting that applying SMOTE alone does not provide sufficient information to effectively mitigate class imbalance.

Looking ahead, several enhancements may improve model performance. Expanding the unlabeled dataset would allow for more effective use of SSL techniques. Furthermore, incorporating temporal models such as RNNs could help capture sequential patterns in user behavior that static features fail to represent.

References

Adwan, O., Faris, H., Jaradat, K., Harfoushi, O., and Ghatasheh, N. Predicting customer churn in telecom industry using multilayer preceptron neural networks: Modeling and analysis. *Life Science Journal*, 11(3):75 – 81, 2014. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84893260534&partnerID=40&md5=d09b6734f7f8b8c7793488183b59b5b3. Cited by: 56.

Ahmed, A. and Linen, D. M. A review and analysis of churn prediction methods for customer retention in telecom industries. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1–7, 2017. doi: 10.1109/ICACCS.2017.8014605.

Howard, A., Chiu, A., McDonald, M., msla, Kan, W., and Yianchen. Wsdm - kkbox's churn prediction challenge. https://kaggle.com/competitions/kkbox-churn-prediction-challenge, 2017. Kaggle.

Ismail, M., Awang, M. K., Abdul Rahman, M. N., and Makhtar, M. A multi-layer perceptron approach for customer churn prediction. *International Journal of Multimedia and Ubiquitous Engineering*, 10:213–222, 07 2015. doi: 10.14257/ijmue.2015.10.7.22.

Liu, X., Xie, M., Wen, X., Chen, R., Ge, Y., Duffield, N., and Wang, N. A semi-supervised and inductive embedding model for churn prediction of large-scale mobile games. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 277–286, 2018. doi: 10.1109/ICDM.2018.00043.

Prabadevi, B., Shalini, R., and Kavitha, B. Customer churning analysis using machine learning algorithms. *International Journal of Intelligent Networks*, 4:145–154, 2023. ISSN 2666-6030. doi: https://doi.org/10.1016/j.ijin.2023.05.005. URL https://www.sciencedirect.com/science/article/pii/S2666603023000143.

Rani, B. J. B. and Kant, S. Semi-supervised learning approach to improve machine learning algorithms for churn analysis in telecommunication. In *International Journal of Computer Information Systems and Industrial Management Applications*, 2020. URL https://api.semanticscholar.org/CorpusID:235668131.