informatics

MDPI

Article

Temporal Churn Prediction in Mobile Game with Fuzzy Behavioral Augmentation and SHAP Explainability

Kaan Arik [1,*], Burak Aggul [2], Hüseyin Gökal [3] and Bilal İnan [4]

[1] Department of Computer Technologies, Vocational School of Information Technologies, Sakarya University of Applied Sciences, Sakarya 54050, Turkey
[2] Department of Database, Network Design and Management, Plato Vocational School, Istanbul Topkapı University, Istanbul 34020, Turkey; burakaggul@topkapi.edu.tr
[3] Department of Computer Technologies, Plato Vocational School, Istanbul Topkapı University, Istanbul 34020, Turkey; huseyingokal@topkapi.edu.tr
[4] Department of Computer Technologies, Niksar Vocational School, Tokat Gaziosmanpaşa University, Tokat 60600, Turkey; bilal.inan@gop.edu.tr

Abstract

The entertainment industry covers numerous fields of business, such as telecommunication, television, music, video games, and event management. Out of these markets, the gaming industry has been growing significantly due to its high revenues, a substantial number of players, and overall popularity among customers worldwide. Player churn prediction may inform retention decisions, although predictive performance alone does not establish that an intervention improves retention. This study uses fuzzy behavioral modeling and SHAP-based explanations to examine temporal churn prediction in a single mobile word puzzle game. We examine whether fuzzy representations of player activity improve the prediction of future inactivity in a mobile word puzzle game. The dataset contains 2602 users, 46,132 session identifiers, and 50,014 session-level observations collected over 86.41 days. At each prediction date, eligible players have been active during the prior 14 days; churn is defined as no recorded activity in the following 14 days. Historical training observations accumulate across dates, and all preprocessing is fitted on training data. Six ML classifiers are evaluated using 183 base features, 199 fuzzy-augmented features, and filtered versions of both sets. Filtering retains 85 base or 91 augmented features for the original test set. Two historical validation transitions are followed by test periods containing 144 and 101 players, respectively, with 245 observations from 204 distinct players in the combined assessment. The Logistic Regression configuration selected by historical validation achieves original-test balanced accuracy of 0.608, AUC of 0.700, and an F1-score of 0.643. Its combined balanced accuracy is 0.620 (95% player-cluster bootstrap interval 0.543–0.692). Paired ablations show that fuzzy augmentation does not provide a consistent advantage across classifiers and evaluation settings; the effects of feature reduction also vary. These performance estimates are specific to this dataset, its recently active players, and the stated temporal protocol. They do not establish general predictive performance or the usefulness of fuzzy augmentation in other games, populations, or observation periods.

Keywords: churn prediction; temporal validation; fuzzy features; feature selection; mobile games

Check for updates

Academic Editors: Ashkan Ebadi and Mani Abedini

Received: 29 June 2026
Revised: 24 September 2026
Accepted: 29 September 2026
Published: 30 September 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Predicting when an active player will stop returning is a practical problem for mobile game developers. The available evidence depends on when a prediction is made: a model used immediately after onboarding has a different information set from one used after several weeks of play. Hoang and Cam studied the first setting using a two-stage system built from early gameplay records [1]. In a mobile-game classification study, Emre and Çotul examined compact behavioral summaries such as level count and average play duration [2]. Arık et al. mapped the game customer churn literature through bibliometric analysis and investigated the indicators affecting customer churn in MMORPGs using machine-learning models [3,4]. In this study, we focus on recently active players and ask whether their recorded histories predict inactivity over the next 14 days.

A useful evaluation must reproduce the information available at that decision point. Apicella et al. demonstrated how preprocessing and data splitting can lead to leakage even when the classifier itself is applied correctly [5]. Understanding these risks is particularly important when a player appears at several prediction dates. Earlier observations may be legitimate training data, but their outcome windows must be complete before the next prediction. We therefore define eligibility, feature histories, and future outcomes separately and fit every transformation on historical training observations.

Fuzzy features offer a simple way to express gradual changes in engagement. Their usefulness remains an empirical question because they transform existing measurements rather than introduce new information. Research on fuzzy classification has examined feature selection as part of the learning procedure, including iterative selection within rule-based classifiers [6]. Our approach is simpler and applies fixed membership functions transform four behavioral measurements, and the resulting features are supplied to conventional classifiers.

We evaluate six ML classifiers with and without a set of 16 fuzzy features, both before and after training-only redundancy filtering, with historical observations accumulating across prediction dates and a latest-fold-only analysis to measure the effect of that training policy. Two later periods provide separate and combined evaluations, with paired intervals for differences between feature conditions. We document a temporal workflow and assess fuzzy feature engineering in one mobile word puzzle game. Our study does not introduce a new classifier or evaluate performance across games.

2. Related Work

2.1. Behavioral Features and Game Churn

Recent game-churn studies differ substantially in the information they use and the stage of the player lifecycle they address. As previously noted, Hoang and Cam combined neural and conventional models for early prediction after onboarding [1]. Wu et al. examined behavioral variability in social casino games and distinguished the histories of monetizing and non-monetizing players [7]. These studies motivate attention to the observation window and the target population; neither provides a directly comparable benchmark for our active-player, 14-day outcome.

Social relationships offer information beyond individual activity summaries. Han et al. combined graph convolution and label-propagation methods to model social activity in an MMORPG [8]. Guo and Liu represented friendship and teammate relationships as separate network layers for churn prediction [9]. The dataset used in this study does not contain information on social relationships or interaction networks; therefore, the feature set is limited to each player's recorded actions and temporal behavior.

The interpretation of inactivity also depends on how missing records are handled. Lee et al. distinguished different kinds of non-login periods and evaluated imputation strategies

for game-churn prediction [10]. In our dataset, missing activity values are represented as zero. This may make it difficult to distinguish between no activity and missing data. We state this assumption explicitly, as the original logging process does not allow us to verify this distinction.

2.2. Representation and Explanation

Feature-based representations are one of the existing ways of gameplay description. Melo et al. compared the game-provenance graph representation to manually defined gameplay metrics in terms of win and hit prediction [11]. While their tasks are different from ours (churn prediction), the comparison is useful in that it puts our feature set of features in the context of the game-analytics literature. On the other hand, denoised player-behavior representation learning was proposed by Li et al. as a way of handling potentially noisy behavioral signals [12]. We keep the simple and interpretable and evaluate its contribution directly through ablation.

Interpretability is also a well-known issue in churn analysis. In this regard, Wang et al. proposed to explicitly consider both explicit and implicit features to make game-churn prediction more interpretable and consequently actionable [13]. Jung et al. studied the problem of explainable prediction in the context of personalized retention promotions [14]. Our SHAP analysis aims to characterize the fitted model, however, we did not explicitly test interpretability-informed interventions. In general, the problem of model interpretation, particularly in the context of SHAP/LIME frameworks, was discussed in detail by Salih et al., who showed that feature collinearity, as well as the fitted model itself, may significantly bias the results of SHAP/LIME analyses [15]. Therefore, we rely on the SHAP framework for interpreting the model behavior, while the ablation study allows us to estimate the contribution of individual features to the model's performance.

2.3. Temporal Evaluation and Fuzzy Classification

A leakage assessment should consider both the implemented data pipeline and the proposed experimental design. Alturayeif and Hassine evaluated the automatic detection of leakage in machine-learning code [16]. The REFORMS recommendations also highlight the importance of reporting study design, evaluation, and reproducibility choices explicitly [17]. We therefore report the split dates, fitted transformations, selected columns, and model hyperparameters together with the computational materials.

Chronological splitting does not account for shifting demographics in the player population. Arora et al. discuss the challenges of detecting and adapting to distribution shifts that occur during training [18]. This motivates our choice to report the results of our two later temporal splits separately, even if the short length of the series prevents us from characterizing seasonal trends or concept drift. Reducing the number of features may have improved the estimation accuracy, but the subset chosen should be stable and relevant to the classifier. Comparison studies performed by Barbieri et al. Have measured the predictive accuracy, redundancy, stability and efficiency of feature-selection algorithms as reported in their study [19]. It is also shown by Lohrmann et al. That individual and ensemble filters can remove features from fuzzy nearest-neighbor classifiers without consistent reduction in performance across the benchmark datasets that they consider [20]. This indicates that the relative usefulness of feature selection should be measured rather than assumed to be an advantage.

Our variance filter used here avoids the need to search across feature subsets and is used independently at each training origin. Additionally, the fuzzy layer applied here uses piecewise-linear memberships and a fixed arithmetic aggregation. There is no ruleset learning phase, or any iterative rule learning algorithm explored by Li et al. [6]. Instead,

its design space is deliberately limited to investigate if encoded recency and activity in a stepwise fashion can improve the performance of the same classifiers with the same temporal protocol.

3. Methodology

Figure 1 summarizes the methodology of the temporal evaluation. Player's activity before each cutoff date determines the training data as well as the target (future inactivity). Features were generated, selected, scaled, and SMOTE-ed using only the training partitions. The resulting model was then evaluated on future (test) partitions using classifiers trained in the same fashion.

Pre-cutoff data and active-player selection → Past outcome periods → Cumulative training observations → Fuzzy feature generation and optional feature selection using training data only → Scaling and SMOTE using training data only → Classifier training → Later-period prediction and paired evaluation

Figure 1. Temporal evaluation workflow. Every fitted transformation uses historical training observations only. Labels are defined by subsequent inactivity rather than fuzzy outputs.

3.1. Dataset and Observation Unit

The dataset was obtained from a commercial mobile word puzzle game available on iOS and Android. The users table contains 3224 registration records. The actions table contains 1,706,884 event records associated with 2602 users, 46,132 distinct 'act_id' session identifiers, and 37 action types. The observation period extends from 30 August 2024 at 20:09:55 to 25 November 2024 at 05:59:53, covering approximately 86 days, as shown in Figure 2.

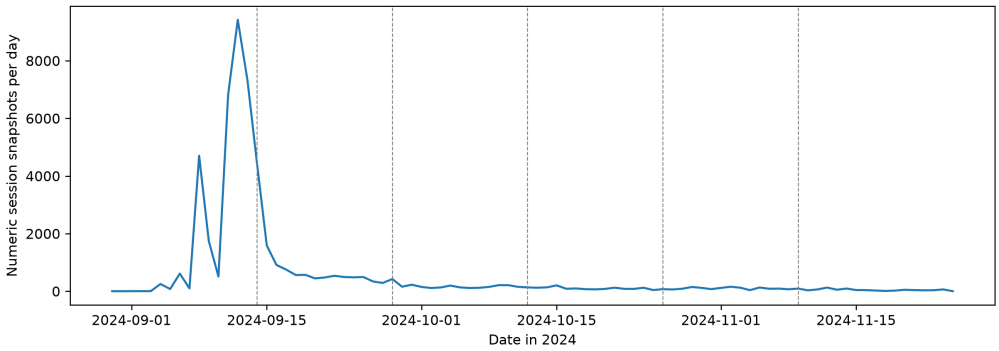


Figure 2. Daily numeric session-snapshot counts. Vertical lines identify the five prediction cutoffs. The raw observation interval extends beyond the last complete outcome window.

Each event has an identifier, action type, value, timestamp, user identifier, and session identifier. A session identifier is not a unique timestamp: 4028 session identifiers occur at multiple timestamps, while none is shared by multiple users. Consequently, a pivot indexed by `(act_id, user, date_at)` produces session snapshots, not one row per distinct session. Numeric conversion followed by this pivot yields 50,014 rows and 40 columns (37 action columns and three indexing columns). The raw data contain 50,161 distinct session–user–timestamp keys, of which 147 contain no numeric value. The numeric pivot retains 46,132 distinct session identifiers and 3882 additional timestamp rows beyond one row per retained identifier.

There are no duplicate records for the full `(act_id, user, date_at, action)` key. Nonnumeric values are coerced to missing before the numeric pivot. Missing numeric behavioral entries are represented as zero, preserving the original implementation's absence-

as-zero assumption. This assumption may conflate an unrecorded event with a true zero and is a limitation of the available telemetry. For consistency with the prior experiment, the variable named `total_sessions` counts numeric session snapshots; all session-frequency and gap features use this same unit. We do not reinterpret 50,014 snapshots as 50,014 independent sessions.

The 33 behavioral inputs exclude `Email`, `LanguageLevelStr`, `Link`, and `RemoveAds`. Email is an account/contact field with mixed numeric and textual values, while LanguageLevelStr contains structured strings that are not directly comparable scalar measurements. According to the data owner, Link records arrival through an advertisement click, and RemoveAds records watching and dismissing an advertisement. The latter two were absent from the original preselected gameplay-feature list; their exclusion is not evidence that they are uninformative. We preserve the original 33-action definition for the primary ablation and additionally restore both advertising actions in a sensitivity analysis. These exclusions define the original feature set, rather than a test-label-driven selection. An inventory of all 37 action types is provided with the computational materials.

3.2. Outcome and Active-User Population

At cutoff t, a player is eligible if at least one numeric session snapshot occurs in $[t - 14 \text{ days}, t)$. Predictors use only snapshots strictly before t. The outcome is

$$y_i(t) = \mathbb{1}\{\text{no snapshot for player } i \text{ in } [t, t + 14 \text{ days})\}. \quad (1)$$

Our dataset label records a 14-day episode of future inactivity; it does not establish permanent churn. We retain this operational horizon from the our dataset. The half-open intervals avoid assigning an event at a shared boundary to two outcome windows. Eligibility uses past activity only and is independent of the subsequent outcome.

3.3. Temporal Evaluation Design

Three historical cutoffs provide two forward validation transitions before the original test cutoff: train on F_1 and validate on F_2; then train on $F_1 \cup F_2$ and validate on F_3. For the original test T_1, train on $F_1 \cup F_2 \cup F_3$. For the additional test T_2, add the now-observed T_1 training snapshots and outcomes. In every case, the latest training outcome window ends no later than the evaluation cutoff. The additional test window ends on 23 November, before the last recorded timestamp as seen in Table 1 and Figure 3.

The training data accumulate across prediction dates, with previous user–cutoff observations kept as new dates are added. For the original test period, the training data contain 3167 user–cutoff observations from 2512 distinct users, and the additional test period, 3311 observations. The same user can contribute observations at multiple cutoff dates. User identifiers are excluded from all predictors, and the focus is on predicting later behavior of existing players; therefore, repeated observations of the same user are not considered independent players. As part of a sensitivity analysis, we also evaluate a latest-fold-only setting, where only the most recent historical period is used for training.

Table 1. Prediction snapshots and half-open 14-day outcome windows in 2024. All feature histories begin at the first recorded event and end strictly before the cutoff.

Snapshot	Cutoff	Outcome End	n	Churn	Retained	Churn (%)
F_1	14 Sep	28 Sep	2120	1640	480	77.4
F_2	28 Sep	12 Oct	833	695	138	83.4
F_3	12 Oct	26 Oct	214	159	55	74.3
T_1	26 Oct	09 Nov	144	103	41	71.5
T_2	09 Nov	23 Nov	101	75	26	74.3

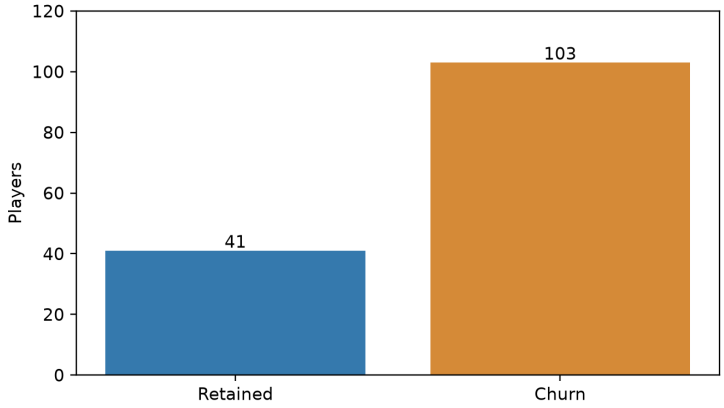


Figure 3. Original-test outcome distribution among 144 eligible players: 103 churned and 41 retained. The all-churn accuracy baseline is 103/144.

We use two validation periods, T_1, T_2, and an out-of-time test period. The available observation period is limited by the need to construct the features, which requires a sufficient history of observations, a full 14-day outcome period, and later periods for testing. We therefore use two pre-test validation periods and reserve the remaining full period for additional out-of-time evaluation. The two validation periods only provide limited insight into temporal variation, which is taken into consideration while interpreting the findings. The predictions of T_1 and T_2 are created before observing their corresponding outcome periods, so neither validation period uses the information of their outcomes while producing the predictions.

In feature engineering, for feature definitions, our base vector contains 183 features, and the fuzzy layer adds 16, for 199 before selection. The count is

$$11 + 165 + 4 + 2 + 1 + 16 = 199. \tag{2}$$

The disjoint components are:

- 11 temporal features: snapshot count, lifespan, active-day count, snapshots per active day, mean/standard deviation/maximum/minimum inter-snapshot gaps, weekend ratio, average play hour, and night-play ratio.
- 5 aggregates (mean, maximum, sum, standard deviation, and last value) for each of 33 numeric behavioral inputs: 165 features.
- 4 least-squares slopes over snapshot index for Score, PlayTime, CoinEarned, and WrongMove. Slopes are zero when fewer than three snapshots are available.
- 2 features for last-snapshot engagement diversity and an indicator of positive recorded MoneySpent.
- One recency feature, $r_i = (t - t_i^{\text{last}})/(1\text{ day})$, expressed in days. Recency is counted only here, not among the eleven temporal features.
- 16 fuzzy features as specified below.

The exact ordered feature names are saved for each fit. Behavior aggregates describe recorded values; without a complete telemetry dictionary, they should not automatically be interpreted as independent event frequencies or transaction totals.

3.4. Training-Only Fuzzy Transformation

The four inputs are recency, active days, mean PlayTime, and snapshot count. They encode recency, frequency, intensity, and volume. For each training collection separately, estimate q_{25}, q_{50}, and q_{75} without using outcomes or evaluation observations. Freeze these quantiles before transforming the next period.

We use trapezoidal shoulder functions for low and high values and a triangular function for intermediate values. The plateau in a shoulder represents similarly low or high engagement over a range, while the triangle gives maximal medium membership at the training median and decreases towards the adjacent quartiles. These choices give rise to transparent, low-complexity piecewise-linear encodings with no optimized shape parameters. They are a modelling choice appropriate to the small dataset, not evidence that triangles and trapezoids are better than Gaussian or other membership families. To claim such superiority, one would have to perform a separate nested comparison.

For nondegenerate quantiles, define

$$L(v) = \begin{cases} 1, & v \le q_{25}, \\ (q_{50} - v)/(q_{50} - q_{25}), & q_{25} < v < q_{50}, \\ 0, & v \ge q_{50}, \end{cases} \tag{3}$$

$$M(v) = \max\left(0, \min\left\{\frac{v - q_{25}}{q_{50} - q_{25}}, \frac{q_{75} - v}{q_{75} - q_{50}}\right\}\right), \tag{4}$$

$$H(v) = \begin{cases} 0, & v \le q_{50}, \\ (v - q_{50})/(q_{75} - q_{50}), & q_{50} < v < q_{75}, \\ 1, & v \ge q_{75}. \end{cases} \tag{5}$$

Terminal fuzzy sets saturate outside the training range as seen in Table 2 and Figure 4. For tied quantiles, zero-width slopes use a denominator floor of 10^{-12} and the median receives medium membership one; low/high plateaus retain their boundary definitions. This avoids the division-by-zero problem in formulas that use equal trapezoid endpoints. The same deterministic convention applies to every model and evaluation period.

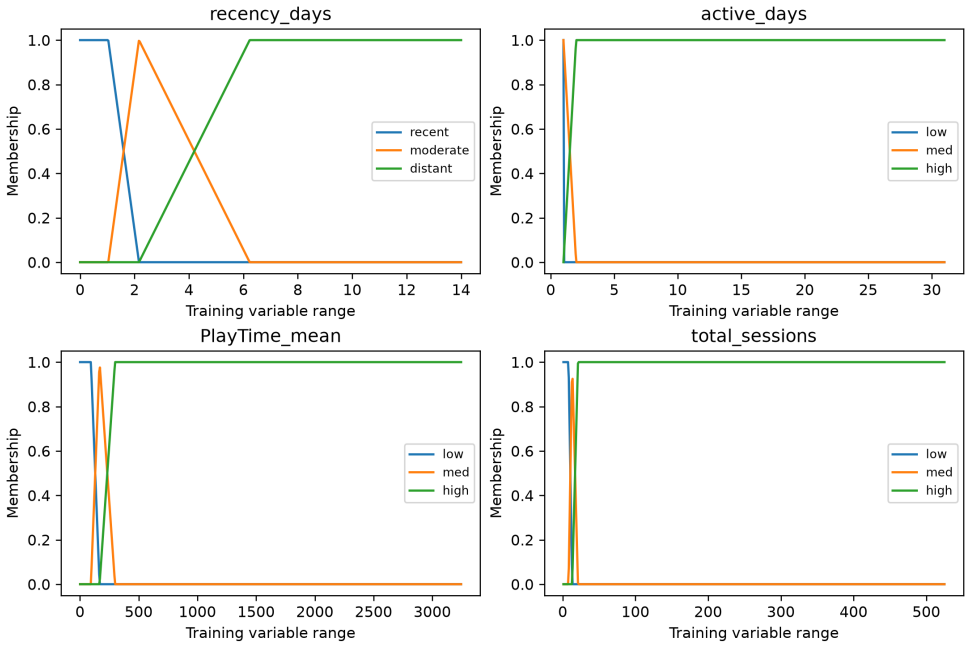


Figure 4. Membership functions calibrated on accumulated historical training observations for the original test. The extreme memberships remain saturated outside the displayed training range.

Recency uses the labels recent, moderate, and distant for L, M, H, respectively. For the other three inputs, the labels are low, medium, and high. Average the distant-recency membership and the three low positive-engagement memberships to obtain e_L; average the

four medium memberships to obtain e_M; and average recent recency with the three high memberships to obtain e_H. The scalar engagement score is

$$E = \frac{15e_L + 50e_M + 85e_H}{e_L + e_M + e_H + 10^{-10}}. \quad (6)$$

Sixteen outputs comprise twelve input memberships, three aggregate memberships, and this score. All weights and score anchors are fixed. The layer does not generate the churn label, learn rules, or claim psychological measurement validity.

Table 2. Fuzzy breakpoints estimated on accumulated historical training observations for the original test. Minima and maxima describe the training range; shoulder memberships saturate outside it.

Input	Min	q_{25}	q_{50}	q_{75}	Max
recency_days	0.000	1.035	2.164	6.231	14.000
active_days	1.000	1.000	1.000	2.000	31.000
PlayTime_mean	1.000	93.025	166.852	298.572	3245.474
total_sessions	1.000	8.000	13.000	21.000	525.000

3.5. Training-Only Feature Selection and Noise Control

We compare four conditions: base features (Raw), base plus fuzzy features (Fuzzy), selected base features (Raw selected), and selection after fuzzy augmentation (Fuzzy selected). Selection first removes zero-variance columns using training observations only. It then computes absolute Spearman correlations between remaining training features. In the deterministic original feature order, a feature is retained unless its absolute correlation with an already retained feature exceeds 0.90. The threshold is fixed, not optimized against either test period. The resulting column list is frozen for evaluation. This simple filter addresses exact constants and monotonic redundancy without estimating hundreds of outcome-dependent feature weights from a small fold.

Feature selection is performed prior to scaling and synthetic oversampling, reducing the number of features but not necessarily eliminating measurement noise or overfitting. We retain all four conditions, therefore, to assess their impact on the model performance. Robust scaling, limited tree complexity, regularization, and the latest-fold sensitivity analysis offer further opportunities for examining the model robustness.

3.6. Models and Hyperparameters

Six model families retain the principal settings of the supplied implementation. Logistic Regression uses L2 regularization, $C = 1$, balanced class weights, and at most 2000 iterations. Random Forest uses 200 trees, maximum depth 8, minimum leaf size 5, and balanced class weights. XGBoost uses 150 estimators, depth 5, learning rate 0.1, and minimum child weight 3. LightGBM uses 150 estimators, depth 6, learning rate 0.1, minimum child size 10, and balanced class weights. CatBoost uses 150 iterations, depth 5, learning rate 0.1, automatic balanced class weights, and minimum leaf-data setting 5. The MLP uses hidden layers of 64 and 32 units, regularization $\alpha = 0.01$, and at most 500 iterations.

No grid, random, or Bayesian hyperparameter search is performed. These are fixed settings inherited from the existing analysis, not claimed to be optimal. The random seed is set to 42 for reproducibility. Parameters not listed retain the versions' defaults, and exact software versions are locked in the computational materials. MLP's internal random early-stopping holdout is disabled: splitting an already oversampled dataset internally could mix synthetic relatives across fitting and validation. Training-loss convergence can still terminate optimization, and any convergence warnings are recorded. Differences

from the prior run consequently reflect the corrected protocol and environment as well as accumulated training data; the new tables supersede the earlier values.

For each fit, the training-only fuzzy transformation and optional selector are followed by RobustScaler fitted on training observations, synthetic minority oversampling (SMOTE) on training observations only, and model fitting. SMOTE uses $k = \min(5, n_{\min} - 1)$. Evaluation data are transformed but never oversampled. SMOTE and class weights are retained together where specified in the source configuration, and their interaction is a limitation rather than an optimized balancing strategy. Predictions use a fixed probability threshold of 0.5; the threshold is not tuned on test outcomes.

To make the models more compactly presented, the feature conditions selected are those with the highest mean balanced accuracy across the 2 historical validation transitions (deterministic alphabetical ties) that are held fixed across test periods. All other conditions are provided in the ablation and computational tables, avoiding suppression of bad test results through the selection of test conditions.

3.7. Metrics and Uncertainty

Churn is the positive class. We report accuracy, churn-class F1-score, ROC AUC, recall, retained-user specificity, and balanced accuracy, which is the mean of recall and specificity. For the always-churn baseline, specificity is zero, while balanced accuracy and AUC equal 0.5, and its accuracy and F1-score depend on the prevalence of churn in each period.

We estimate percentile 95% confidence intervals (CI) using 1000 paired bootstrap replications with a fixed random seed, for the accuracy, balanced accuracy, F1-score, AUC, and differences within a model. The data are resampled at the user level: all test-period observations from a user resampled in a bootstrap are included. The same set of users is used to compare different feature conditions. For a single period, this implements a user-level bootstrap; for two periods, the dependence within a user across periods is preserved. All intervals should be interpreted as uncertainty conditional on the fitted models and the observed periods for the resampled users. They do not reflect uncertainty due to refitting or the generalizability to other games or periods. Similarly, comparisons of different conditions are exploratory and not corrected for multiple hypothesis testing; the intervals do not account for post-selection inference [21].

The selection of feature conditions is made using historical validation, while the maximum test value within a condition is reported descriptively. An interval spanning zero is therefore not treated as evidence of a reliable improvement.

3.8. Correlation and Interpretability Analysis

Pearson correlation measures linear association and is sensitive to extreme values. Spearman correlation uses ranks, accommodates skewed counts and monotonic nonlinear relationships, and is therefore used in the redundancy filter. Distance correlation additionally captures non-monotonic dependence; its distance-based construction is discussed by Raymaekers and Rousseeuw [22]. It is nonnegative and does not encode direction. These quantities are descriptive association measures, not causal effects.

Full Pearson and Spearman matrices use accumulated training data for the original test. For a directly comparable three-panel figure, all three measures are computed on the same deterministic sample of 512 training user–cutoff observations and ten prespecified interpretable features. This bounds the quadratic cost of distance calculations. The empirical distance correlation uses double-centered pairwise Euclidean distance matrices; its finite-sample value can be positive even under independence, so no independence test is inferred from the heatmap. Evaluation features and outcomes are not used to select

columns or compute these training diagnostics. SHAP is separately calculated on the original test for the fitted full-feature LightGBM model, solely as a post hoc explanation.

4. Results

4.1. Validation and Feature Condition Selection

As shown in Table 3, the largest mean validation balanced accuracy is 0.709 for Logistic Regression with raw selected features. Feature conditions were selected using these historical transitions, not the test metrics. Complete results for all four conditions and both training policies are retained in the computational materials.

Table 3. Historical validation results for the feature condition selected separately for each model by mean balanced accuracy. Standard deviations summarize only two periods and are not confidence intervals.

Model	Condition	Balanced Accuracy	F1	AUC
LR	Raw selected	0.709 ± 0.001	0.802 ± 0.101	0.788 ± 0.011
RF	Fuzzy selected	0.691 ± 0.029	0.805 ± 0.036	0.775 ± 0.047
XGB	Raw selected	0.675 ± 0.014	0.874 ± 0.049	0.756 ± 0.009
LGBM	Raw selected	0.682 ± 0.031	0.883 ± 0.042	0.751 ± 0.007
CB	Raw selected	0.674 ± 0.045	0.864 ± 0.033	0.762 ± 0.037
MLP	Fuzzy	0.677 ± 0.000	0.777 ± 0.028	0.699 ± 0.055

Although XGB, LGBM, and CB achieve higher mean F1-scores, their lower balanced accuracies suggest that this performance may be concentrated more strongly on the prevalent churn class. The model-specific selections—raw selected features for four classifiers, fuzzy selected features for RF, and unfiltered fuzzy features for MLP—also indicate that fuzzy augmentation does not provide a uniform advantage in these historical validation transitions. Its observed effects are conditional on the classifier, feature condition, and temporal split in this dataset. Because these summaries are based on only two transitions, the small standard deviations should not be interpreted as evidence of broad temporal stability.

4.2. Original and Additional Out-of-Time Tests

As seen in Tables 4 and 5, original all-churn baseline has accuracy 0.715 and an F1-score of 0.834. Some revised conditions exceed its point accuracy; the highest across the reported conditions is 0.771 for XGBoost with fuzzy features. This is a descriptive maximum across conditions, not a test-based deployment selection. Statistical uncertainty and changes in preprocessing prevent attributing the difference to a single correction. For example, full-feature XGBoost has accuracy 0.771, an F1-score of 0.845, and balanced accuracy 0.693. Feature conditions for the primary comparison remain those chosen by historical validation.

Overall, Figures 5–8 show that threshold-independent discrimination and classification at the fixed 0.5 threshold provide complementary evidence. Most ROC and precision–recall curves remain above their respective reference baselines, although MLP exhibits the weakest separation, while the confusion matrices reveal materially different error profiles: LR preserves the largest number of retained players but misses more churners, whereas the boosting models identify more churners at the cost of additional false churn alarms. Within this original-test sample and the fixed full fuzzy-feature condition, XGB has the highest observed accuracy, F1-score, and balanced accuracy, while RF has the highest AUC. These descriptive rankings are specific to this dataset and temporal protocol; they establish neither a stable ranking beyond the evaluated sample nor an advantage of fuzzy augmentation over the corresponding base features. This pattern also explains why high accuracy or F1-score should not be interpreted independently of class-specific errors and threshold-free

discrimination in an imbalanced churn setting. Simulation evidence further cautions that class-imbalance corrections can harm predictive performance and should not be assumed to be beneficial [23].

Table 4. Original test on 144 users (103 churned and 41 retained). Brackets contain user-bootstrap 95% intervals. Feature conditions were chosen using validation only.

Model	Condition	Accuracy	F1	Specificity	Balanced Accuracy [CI]	AUC [CI]
LR	Raw selected	0.576	0.643	0.683	0.608 [0.522, 0.699]	0.700 [0.599, 0.808]
RF	Fuzzy selected	0.708	0.790	0.561	0.664 [0.581, 0.750]	0.734 [0.643, 0.823]
XGB	Raw selected	0.701	0.790	0.488	0.637 [0.551, 0.724]	0.711 [0.620, 0.807]
LGBM	Raw selected	0.681	0.772	0.488	0.623 [0.537, 0.715]	0.683 [0.583, 0.782]
CB	Raw selected	0.681	0.785	0.341	0.578 [0.496, 0.664]	0.683 [0.584, 0.780]
MLP	Fuzzy	0.674	0.777	0.366	0.581 [0.503, 0.667]	0.587 [0.484, 0.696]
All churn	—	0.715	0.834	0.000	0.500	0.500

Table 5. Additional test (101 users) and combined out-of-time assessment (245 user–cutoff observations from 204 distinct users). Combined intervals resample users jointly across both periods.

Model	T_2 Accuracy	T_2 bal. acc.	T_2 AUC	Combined bal. acc. [CI]	Combined AUC [CI]
LR	0.624	0.634	0.719	0.620 [0.543, 0.692]	0.706 [0.608, 0.792]
RF	0.733	0.707	0.769	0.681 [0.611, 0.748]	0.746 [0.669, 0.817]
XGB	0.802	0.728	0.771	0.674 [0.604, 0.742]	0.734 [0.656, 0.809]
LGBM	0.802	0.703	0.789	0.656 [0.585, 0.726]	0.722 [0.638, 0.798]
CB	0.802	0.728	0.794	0.638 [0.566, 0.701]	0.730 [0.650, 0.801]
MLP	0.663	0.535	0.558	0.563 [0.489, 0.635]	0.579 [0.494, 0.668]
All churn	0.743	0.500	0.500	0.500	0.500

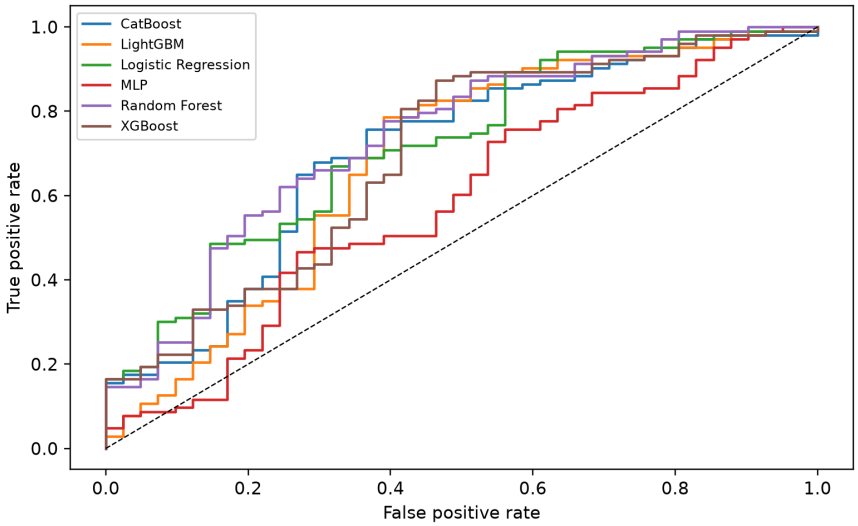


Figure 5. Original-test ROC curves for the six full fuzzy-feature models under accumulated training. These fixed-condition curves complement the validation-selected results and are not used for model selection.

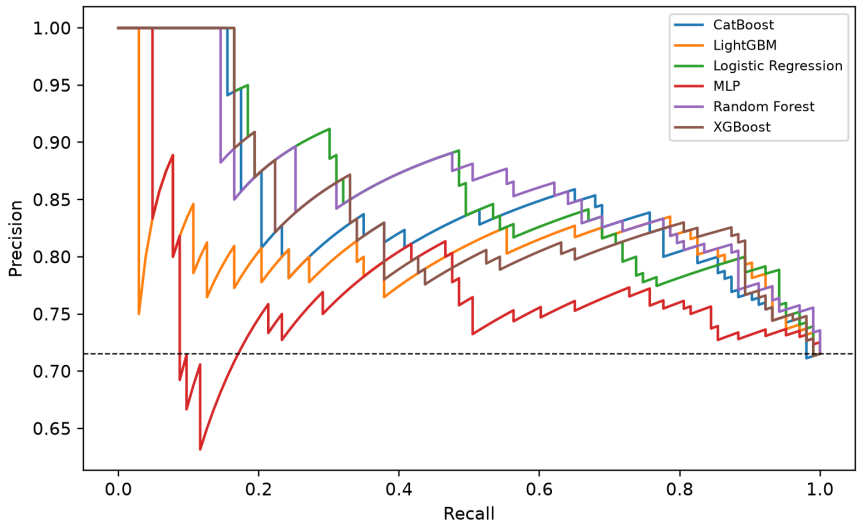


Figure 6. Original-test precision–recall curves for the full fuzzy-feature models. The horizontal baseline is the churn prevalence, 103/144.

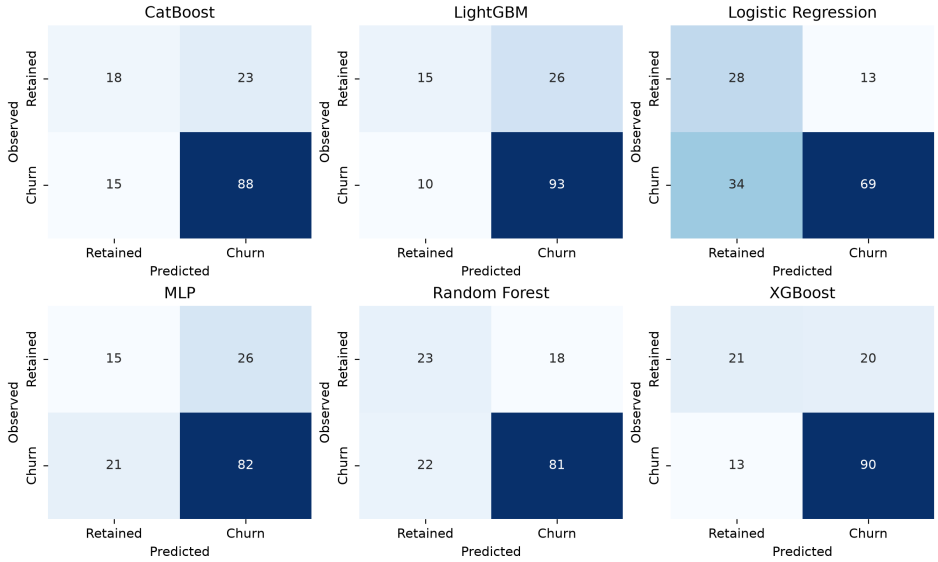


Figure 7. Original-test confusion matrices for all six full fuzzy-feature classifiers trained on accumulated history. Rows are observed outcomes; columns are predictions at threshold 0.5.

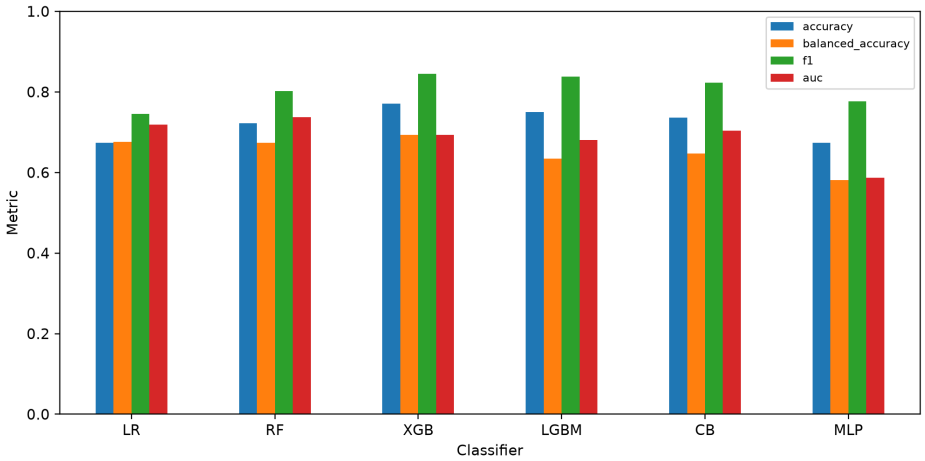


Figure 8. Original-test metric comparison for the full fuzzy-feature condition under accumulated training. Model abbreviations follow the validation table.

4.3. Fuzzy-Layer Ablation

In Table 6 and Figure 9, the effect of the fuzzy layer differs across models. For the original test period, 2 of the 6 paired balanced-accuracy intervals are entirely above zero, while none are entirely below zero; the other intervals include zero. Taken together with the other feature conditions and test period, these results do not demonstrate a consistent advantage for fuzzy augmentation across classifiers and evaluation settings. The positive paired comparisons are conditional evidence from this dataset and protocol, rather than evidence of general superiority. For LightGBM, the AUC difference is −0.026, while for MLP, the F1-score difference is −0.009. These differences should be considered together with their intervals and compared with the results from the additional test period.

Table 6. Original-test fuzzy ablation before feature selection: the same classifier and training protocol with 183 base versus 199 augmented features. Positive differences favor fuzzy augmentation; intervals are paired user-bootstrap 95% intervals.

Model	Raw bal. acc.	Fuzzy bal. acc.	Δ bal. acc. [CI]	Δ AUC [CI]
LR	0.608	0.676	0.068 [0.021, 0.120]	0.038 [0.002, 0.075]
RF	0.650	0.674	0.024 [−0.042, 0.087]	−0.002 [−0.065, 0.054]
XGB	0.610	0.693	0.083 [0.010, 0.155]	−0.007 [−0.059, 0.044]
LGBM	0.644	0.634	−0.010 [−0.086, 0.053]	−0.026 [−0.087, 0.035]
CB	0.676	0.647	−0.029 [−0.076, 0.006]	−0.007 [−0.051, 0.036]
MLP	0.625	0.581	−0.044 [−0.130, 0.037]	−0.063 [−0.169, 0.031]

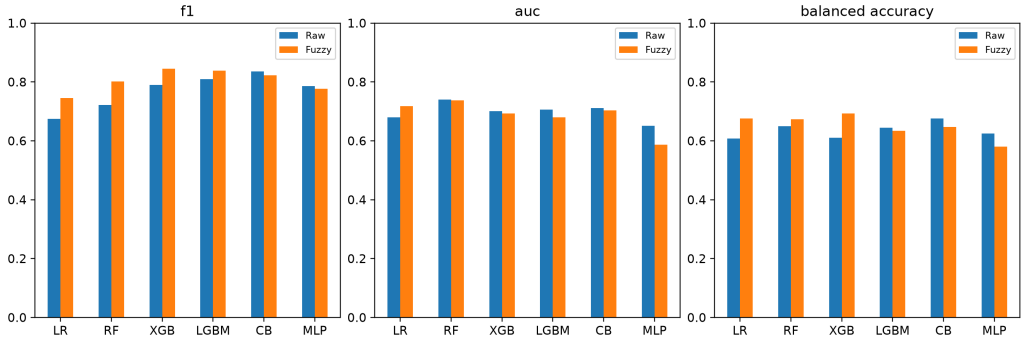


Figure 9. Direct original-test comparison of 183 raw features and 199 fuzzy-augmented features under the same accumulated-training protocol. Paired uncertainty is reported in Table 6.

4.4. Feature Reduction and Training-Sample Sensitivity

In Tables 7 and 8, the nominal full-feature ratio decreases from 199/214 in the earlier latest-fold design to 199/3167 in accumulated training, but the 3167 observations represent 2512 distinct users rather than 3167 independent users. The selector retains 85 base features or 91 augmented features for the original test. Selection does not consistently improve performance: it removes redundancy, but may also remove useful signals. MLP balanced accuracy changes from 0.625 to 0.683 for base features, and from 0.581 to 0.603 for augmented features.

As a sensitivity analysis, we added 5 additional aggregate features for each of the Link and RemoveAds actions. This adds 10 base features, providing a total of 193 raw features, with 209 features following fuzzy augmentation, before feature selection. Across all six ML classifiers, four feature conditions, and two test periods, the 48 additional model fits change balanced accuracy by −0.112 to +0.063 compared with the original action set. The primary configurations selected during validation were not changed. This analysis considers how the exclusion of advertising-related actions might affect the results, but

makes no assumption that such actions provide no useful information. The full model-level differences are provided in the result table in the computational materials.

Table 7. Original-test balanced accuracy before and after training-only feature filtering. Raw features reduce from 183 to 85, and augmented features from 199 to 91.

Model	Raw	Raw Selected	Fuzzy	Fuzzy Selected	Δ Selected Fuzzy [CI]
LR	0.608	0.608	0.676	0.657	−0.019 [−0.087, 0.046]
RF	0.650	0.642	0.674	0.664	−0.010 [−0.053, 0.032]
XGB	0.610	0.637	0.693	0.666	−0.027 [−0.105, 0.060]
LGBM	0.644	0.623	0.634	0.678	0.044 [−0.021, 0.117]
CB	0.676	0.578	0.647	0.649	0.002 [−0.042, 0.051]
MLP	0.625	0.683	0.581	0.603	0.022 [−0.063, 0.114]

Table 8. Original-test sensitivity to training policy, holding the revised model and preprocessing settings fixed. Latest-fold training has 214 observations; accumulated training has 3167.

Model	Latest Full	Latest Selected	Accumulated Full	Accumulated Selected
LR	0.591	0.572	0.676	0.657
RF	0.635	0.632	0.674	0.664
XGB	0.649	0.676	0.693	0.666
LGBM	0.615	0.598	0.634	0.678
CB	0.632	0.625	0.647	0.649
MLP	0.623	0.610	0.581	0.603

4.5. Model Correlation and SHAP-Based Explainability

In Figures 10–12, accumulated training data, active days and snapshot count have Pearson correlation 0.686 and Spearman correlation 0.601. Their sample distance correlation is 0.644. Differences between the coefficients reflect different notions of dependence and, for distance correlation, the specified subsample. The three heatmaps use a common training subsample for a fair visual comparison. Redundancy is expected because fuzzy outputs are derived from existing inputs; association does not demonstrate additional predictive information.

In Figures 13 and 14, leading mean absolute SHAP features for the full-feature LightGBM model are `fuzzy_recency_recent`, `recency_days`, `night_play_ratio`. This ranking explains the fitted model in the original test population. It does not indicate causal effects, measure stability across retraining, or replace the incremental-value comparison. Correlated original and transformed predictors can redistribute SHAP attribution.

Figure 14 provides a more detailed ranking of the features used by the full-feature LightGBM model. The prominence of `fuzzy_recency_recent`, `recency_days`, and `night_play_ratio` indicates that the fitted model relies strongly on temporal recency and selected activity patterns. Because the original recency variable and its fuzzy transformation overlap by design, their SHAP attributions may be redistributed across correlated representations. Consequently, this ranking describes the fitted model but does not establish independent predictive value, causal influence, or an incremental benefit from fuzzy augmentation; the latter is evaluated through the paired ablation analysis.

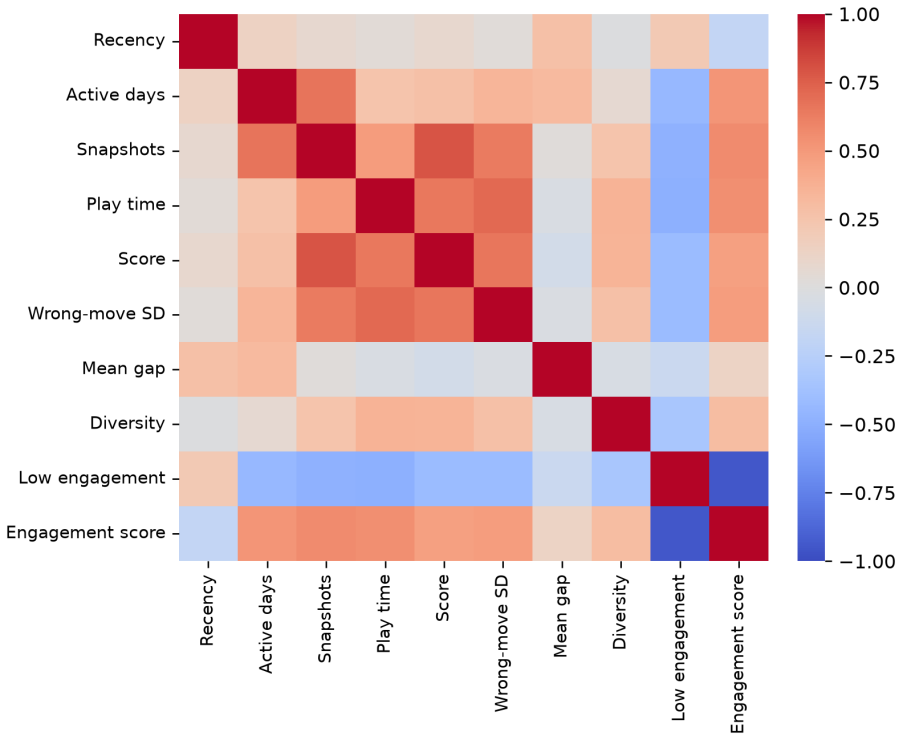


Figure 10. Pearson linear correlations for ten prespecified features on the same sample of 512 historical training observations. No test observations or outcomes enter this diagnostic.

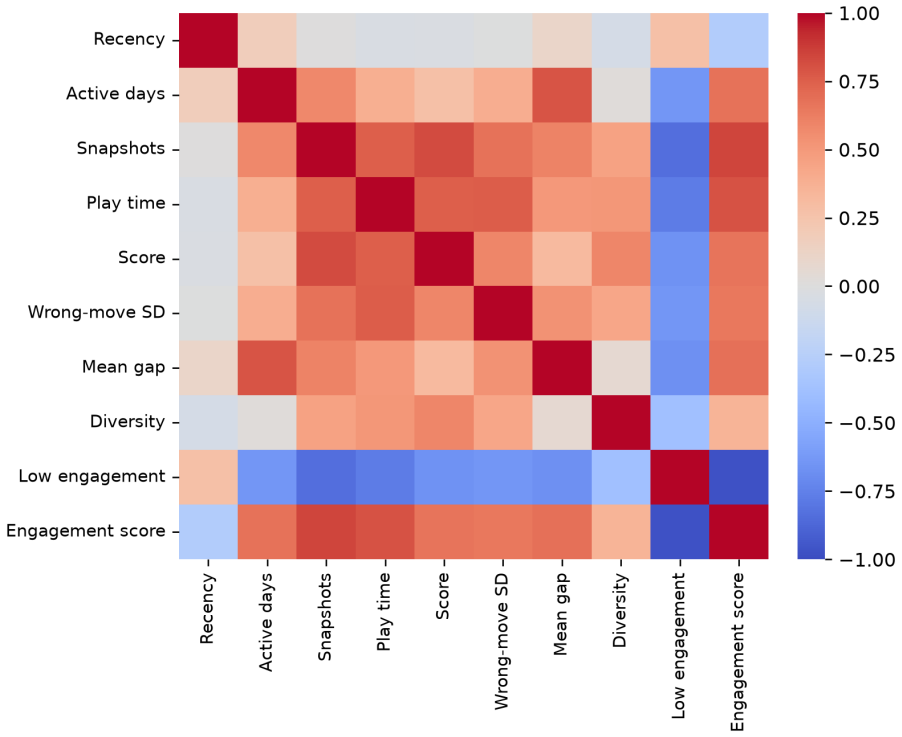


Figure 11. Spearman rank correlations for ten prespecified features on the same sample of 512 historical training observations. No test observations or outcomes enter this diagnostic.

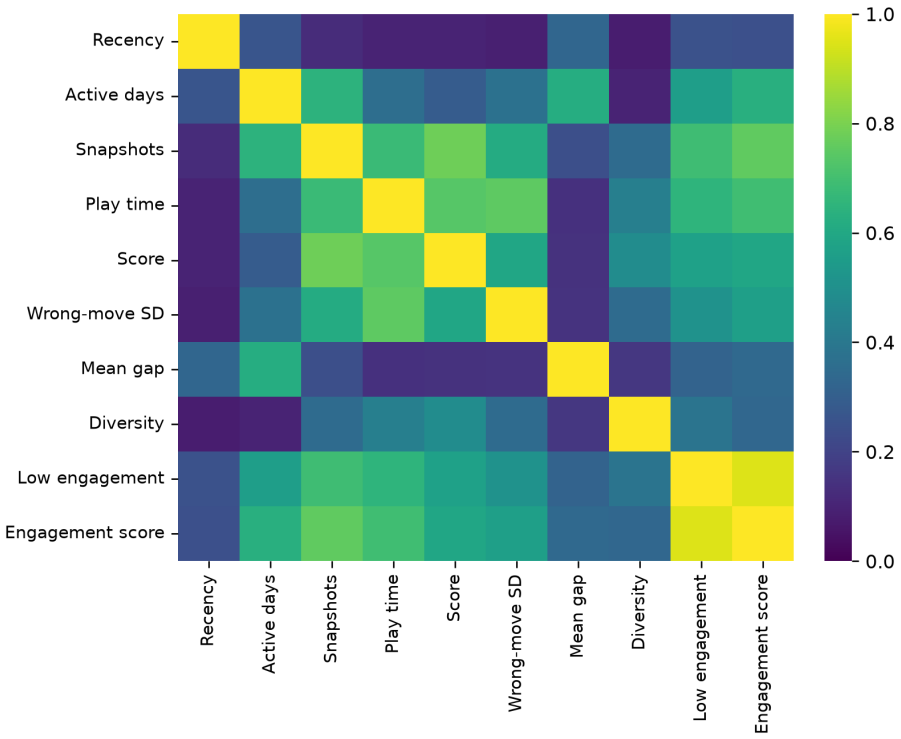


Figure 12. Distance correlations for ten prespecified features on the same sample of 512 historical training observations. No test observations or outcomes enter this diagnostic.

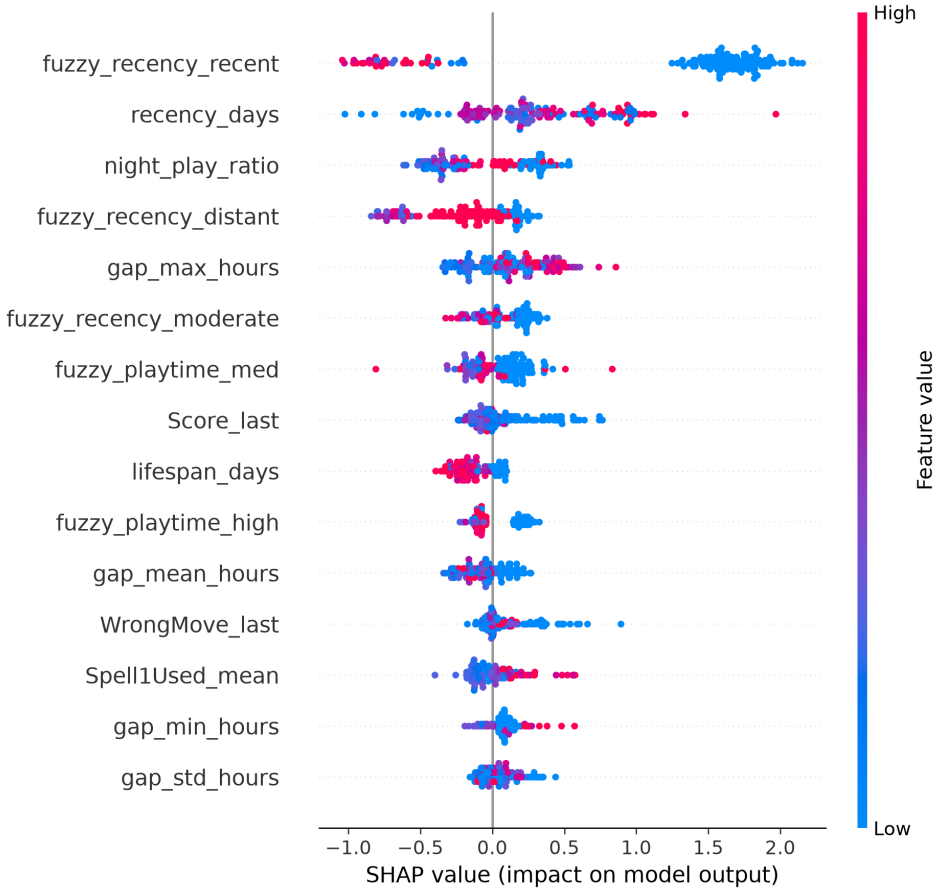


Figure 13. SHAP explanation of the full-feature LightGBM model on the original test. Positive values increase the model output toward churn; colors represent feature values after the training-fitted scaling.

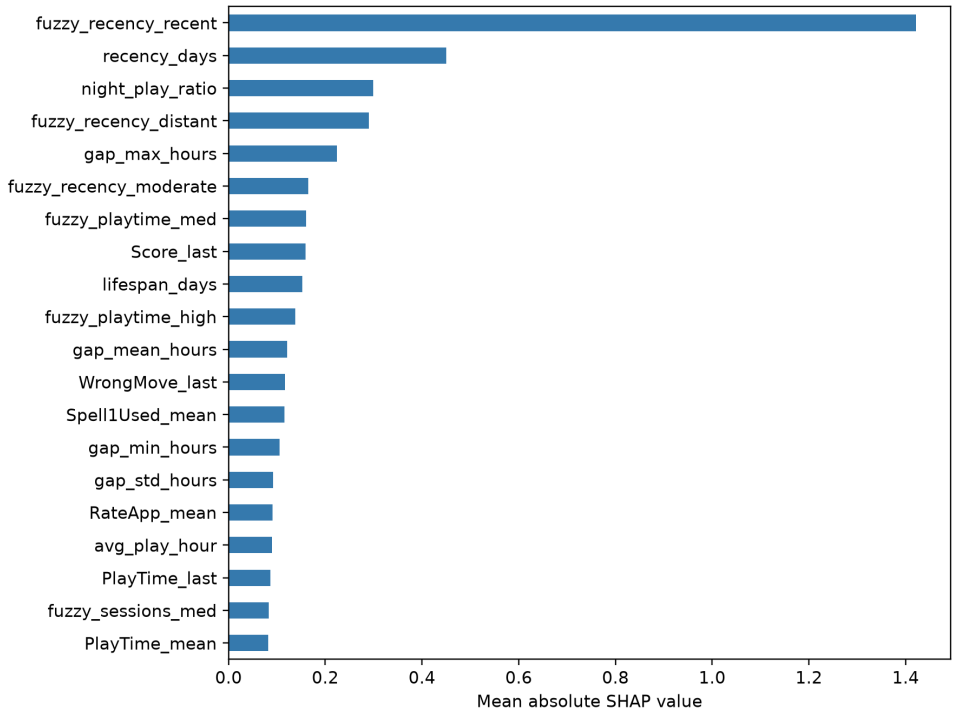


Figure 14. Twenty highest mean absolute SHAP attributions for the same accumulated-training LightGBM fit and original-test population.

4.6. Descriptive Distributions and Training Diagnostics

Learning curves use only the first historical transition, $F_1 \rightarrow F_2$. Five nested random subsets of the historical players (seed 42) are used; every subset independently fits fuzzy calibration, scaling, SMOTE, and each full-feature classifier. Training F1 is evaluated on the original subset, not synthetic observations. Later validation always uses F_2; neither test period is used. These curves describe one training seed and do not measure variability across repeated fits.

Table 9 and Figures 15–17 jointly indicate that additional historical data generally narrow the training–validation gap, most clearly for Logistic Regression and Random Forest, whereas the persistent gaps for several boosted models suggest residual variance or overfitting. Across the two temporal transitions, however, the direction and magnitude of performance changes differ by classifier—particularly for AUC—showing that a favorable learning curve within one transition does not guarantee stability over time. Because only two validation transitions are available, these patterns are diagnostic rather than conclusive evidence of temporal robustness.

Table 9. Exploratory two-sided Mann–Whitney tests on original-test features (103 churned, 41 retained). Unadjusted *p*-values describe associations and do not test incremental fuzzy predictive value.

Feature	U	Unadjusted p
recency_days	2708.0	0.00833
active_days	1243.0	0.0001036
total_sessions	1399.0	0.001616
PlayTime_mean	1396.0	0.00155
fuzzy_engagement_score	1157.5	2.386×10^{-5}
Score_mean	1457.0	0.003789

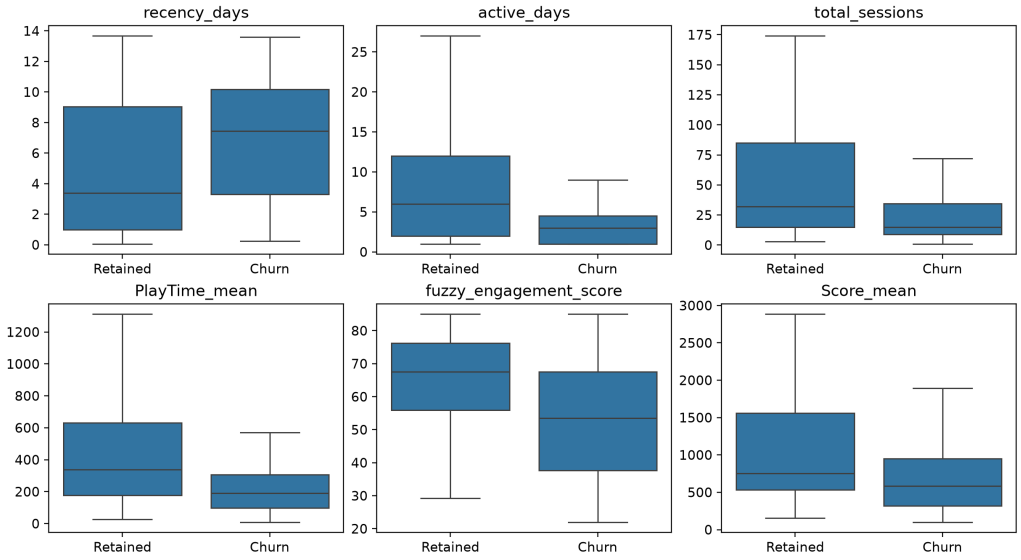


Figure 15. Original-test feature distributions by future-inactivity outcome. Boxes show quartiles; outlier points are omitted for legibility. The engagement score uses accumulated-training calibration. These held-out descriptions do not enter feature selection.

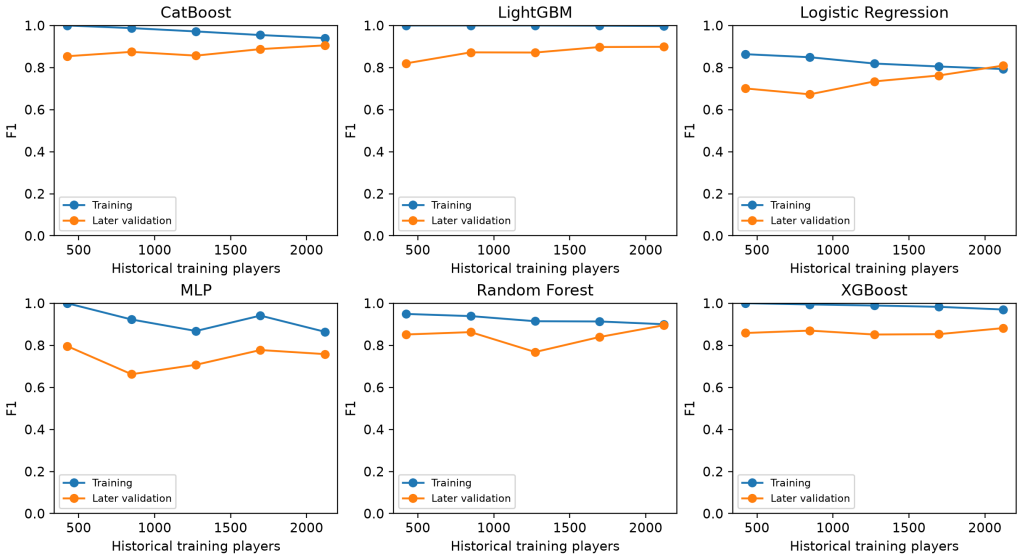


Figure 16. Learning curves for all six full fuzzy-feature classifiers. Each point refits preprocessing and the classifier within the indicated historical training subset; validation is the later F_2 period.

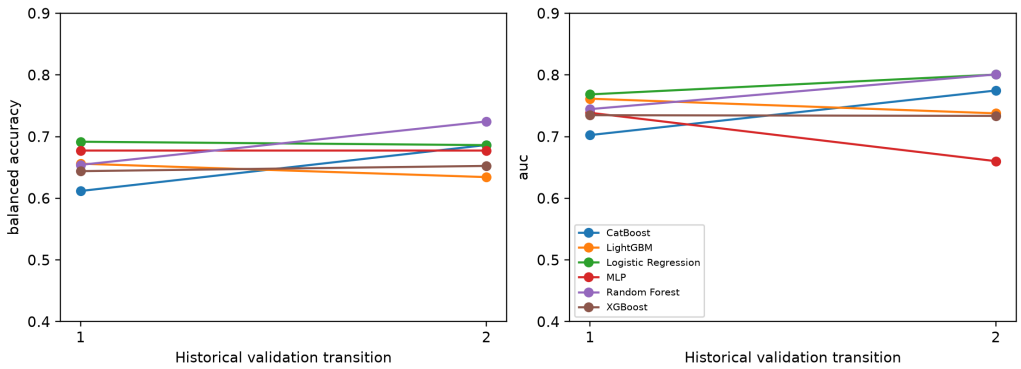


Figure 17. Balanced accuracy and AUC across the two accumulating historical validation transitions for the full fuzzy-feature condition. Two periods provide limited evidence about temporal stability.

5. Discussion

All reported performance estimates are conditional on this single-game dataset, its 86.41-day observation period, and the specified temporal protocol: eligibility requires activity in the preceding 14 days, and the outcome is inactivity in the following 14 days. They do not establish performance for other games, player populations, observation windows, prediction horizons, or validation protocols. Accumulating historical observations increases the number of training observations, but not the number of independent players proportionally. Older observations may also become less representative under temporal drift.

The fuzzy transformation provides interpretable encodings of four existing predictors. The paired ablations show mixed effects across classifiers, feature-selection conditions, metrics, and evaluation periods, with uncertainty intervals that often include zero. Accordingly, the results do not establish a consistent benefit or justify recommending fuzzy augmentation as a default feature representation. A SHAP ranking does not establish incremental predictive value, because correlated raw and transformed features can share or redistribute attribution. Any apparent benefit remains conditional on the evaluated dataset and temporal protocol. Model and feature choices for deployment require prospective validation; the highest test value is not itself a selection rule.

In addition to these interpretations, following limitations should be considered when evaluating the findings of this study:

- Our dataset covers a single game and approximately 86 days; therefore, the findings cannot be directly generalized to other games, genres, countries, or longer observation periods.
- The original test set contains only 144 users, while the additional test period is also relatively small, limiting the generalizability of the results to broader user populations and time periods.
- Both test periods originate from the same telemetry collection, and, therefore, the second period should not be considered fully independent external validation.
- Feature filtering and data augmentation do not eliminate the risk of overfitting. Moreover, the use of fixed hyperparameters and a single training seed limits the assessment of model variability.
- Synthetic oversampling may not fully capture actual player behavior and may influence the calibration of predicted probabilities.
- The absence of a complete event-semantics dictionary, the potential presence of cumulative or state-dependent variables, and the use of zero-filling assumptions limit the behavioral interpretation of the telemetry features.
- The 14-day label represents an operational inactivity episode rather than permanent churn; therefore, longer-term return behavior cannot be assessed with the available data.
- Raw data cannot be publicly redistributed without permission from the data owner. Consequently, although the release of analysis code and aggregate results improves auditability, full independent reproducibility cannot be guaranteed.

6. Conclusions

In this mobile word puzzle dataset, fuzzy augmentation produced both positive and negative point-estimate differences, depending on the classifier, metric, feature-selection condition, and evaluation period. Together with the reported uncertainty intervals, these findings do not demonstrate a consistent advantage for fuzzy features across the evaluated settings. Training-only filtering removed substantial redundancy, but its effect on prediction also varied across models. The validation-leading configuration, Logistic Regression with raw selected features, achieved balanced accuracy of 0.608 and AUC of 0.700 on the

original test. Its combined balanced accuracy was 0.620 [0.543, 0.692] across 245 user–cutoff observations from 204 players. Accumulating historical observations provides a larger training sample while preserving the timing of features and outcomes. These comparisons describe performance only for the available telemetry under the specified temporal protocol; they establish neither general model performance nor a general advantage for fuzzy augmentation. Longer prospective studies in independent games are needed to assess transferability, and controlled interventions are needed to determine whether the predictions can improve retention.

The reported performance is limited to this single game, the 86.41-day collection period, and players eligible under the 14-day activity and 14-day outcome definition. The additional test period comes from the same collection and does not constitute external validation. No claim of predictive effectiveness in other datasets or temporal protocols follows from these results. The SHAP analysis further shows that recency-related variables are prominent in the fitted LightGBM model, but these explanations should be considered descriptive rather than causal. Future research should therefore focus on longer observation periods, independent game datasets, repeated prospective evaluations, and controlled retention experiments to determine whether the observed predictive patterns remain stable and whether they can support effective player-retention interventions.

Author Contributions: Conceptualization, K.A. and B.A.; methodology, K.A., H.G. and B.İ.; software, K.A. and B.A.; validation, H.G. and B.İ.; formal analysis, K.A., H.G. and B.A.; investigation, K.A., H.G., B.A. and B.İ.; resources, B.A. and B.İ.; data curation, K.A. and B.A.; writing—original draft preparation, K.A. and B.A.; writing—review and editing, H.G., B.A. and B.İ.; visualization, K.A. and B.A.; supervision, H.G. and B.İ.; project administration, K.A. and H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset is owned by a commercial company and cannot be publicly shared or redistributed. Access to the raw data is subject to the data owner's permission.

Acknowledgments: The authors would like to express their sincere thanks to the editor and the anonymous reviewers for their helpful comments and suggestions. During the preparation of this manuscript, authors used ChatGPT (GPT-6) solely to assist with English-language academic writing and language editing.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hoang, H.D.; Cam, N.T. Do they like your game? Early-stage churn prediction using a two-phase neural network system. *Eng. Appl. Artif. Intell.* **2025**, *144*, 110102. [CrossRef]
2. Emre, İ.E.; Çotul, S.E. Prediction of Player Churn in Mobile Games Using Classification Algorithms. *J. Data Appl.* **2026**, *4*, 20–29. [CrossRef]
3. Arık, K.; Gezer, M.; Tayalı, S.T. Bibliometric Analysis of Scientific Studies Published on Game Customer Churn Analysis between 2008 and 2022. *J. Politics Econ. Manag.* **2022**, *5*, 55–75.
4. Arık, K.; Gezer, M.; Tolun Tayalı, S. The Study of Indicators Affecting Customer Churn in MMORPG Games with Machine Learning Models. *Upravlenets* **2022**, *13*, 70–85. [CrossRef]
5. Apicella, A.; Isgrò, F.; Prevete, R. Don't push the button! Exploring data leakage risks in machine learning and transfer learning. *Artif. Intell. Rev.* **2025**, *58*, 339. [CrossRef]
6. Li, H.; Wang, C.; Huang, Q. Employing Iterative Feature Selection in Fuzzy Rule-Based Binary Classification. *IEEE Trans. Fuzzy Syst.* **2024**, *32*, 5109–5121. [CrossRef]

7. Wu, R.Y.; Hu, Y.H.; Chou, E.Y. Predicting the churn patterns of monetizers and non-monetizers: Exploring the influence of behavioral variability in churn prediction. *Internet Res.* **2026**, *36*, 438–463. [CrossRef]
8. Han, Y.J.; Moon, J.; Woo, J. Prediction of Churning Game Users Based on Social Activity and Churn Graph Neural Networks. *IEEE Access* **2024**, *12*, 101971–101984. [CrossRef]
9. Guo, M.; Liu, Y. Online game player churn prediction based on multiplex social influence through network embedding. *PLoS ONE* **2026**, *21*, e0356200. [CrossRef] [PubMed]
10. Lee, J.; Rerkjirattikal, P.; Nam, S. A Data Imputation Strategy to Enhance Online Game Churn Prediction, Considering Non-Login Periods. *Data* **2025**, *10*, 96. [CrossRef]
11. Melo, S.; Thurler, L.; Paes, A.; Clua, E. Game provenance graph-based representation learning vs metrics-based machine learning: An empirical comparison on predictive game analytics tasks. *Entertain. Comput.* **2025**, *52*, 100755. [CrossRef]
12. Li, W.; Yao, D.; Xu, Z.; Gong, C.; Jing, Q.; Zhao, S.; Wu, R.; Tan, H.; Hu, Z.; Lv, T.; et al. DPBL: Denoised Player Behavior Representation Learning. *IEEE Trans. Games* **2026**, *18*, 44–55. [CrossRef]
13. Wang, X.; Xie, L.; Wang, H.; Xing, X.; Wan, W.; Wu, Z.; Ma, X.; Li, Q. Deciphering Explicit and Implicit Features for Reliable, Interpretable, and Actionable User Churn Prediction in Online Video Games. *IEEE Trans. Vis. Comput. Graph.* **2025**, *31*, 5990–6007. [CrossRef] [PubMed]
14. Jung, Y.; Jang, H.; Kim, S. Explainable AI for Player Retention: Hyperpersonalized Promotions to Prevent Game User Churn. *IEEE Trans. Games* **2026**, *18*, 369–382. [CrossRef]
15. Salih, A.M.; Raisi-Estabragh, Z.; Galazzo, I.B.; Radeva, P.; Petersen, S.E.; Lekadir, K.; Menegaz, G. A Perspective on Explainable Artificial Intelligence Methods: SHAP and LIME. *Adv. Intell. Syst.* **2025**, *7*, 2400304. [CrossRef]
16. Alturayeif, N.; Hassine, J. Data leakage detection in machine learning code: Transfer learning, active learning, or low-shot prompting? *PeerJ Comput. Sci.* **2025**, *11*, e2730. [CrossRef] [PubMed]
17. Kapoor, S.; Cantrell, E.M.; Peng, K.; Pham, T.H.; Bail, C.A.; Gundersen, O.E.; Hofman, J.M.; Hullman, J.; Lones, M.A.; Malik, M.M.; et al. REFORMS: Consensus-based Recommendations for Machine-learning-based Science. *Sci. Adv.* **2024**, *10*, eadk3452. [CrossRef] [PubMed]
18. Arora, S.; Rani, R.; Saxena, N. A systematic review on detection and adaptation of concept drift in streaming data using machine learning techniques. *WIREs Data Min. Knowl. Discov.* **2024**, *14*, e1536. [CrossRef]
19. Barbieri, M.C.; Grisci, B.I.; Dorn, M. Analysis and comparison of feature selection methods towards performance and stability. *Expert Syst. Appl.* **2024**, *249*, 123667. [CrossRef]
20. Lohrmann, C.; Lohrmann, A.; Kumbure, M.M. On the benefit of feature selection and ensemble feature selection for fuzzy k-nearest neighbor classification. *Appl. Soft Comput.* **2025**, *171*, 112784. [CrossRef]
21. Rink, P.; Brannath, W. Post-selection confidence bounds for prediction performance. *Mach. Learn.* **2025**, *114*, 82. [CrossRef]
22. Raymaekers, J.; Rousseeuw, P.J. Distance Covariance, Independence, and Pairwise Differences. *Am. Stat.* **2025**, *79*, 122–128. [CrossRef]
23. Carriero, A.; Luijken, K.; de Hond, A.; Moons, K.G.M.; van Calster, B.; van Smeden, M. The Harms of Class Imbalance Corrections for Machine Learning Based Prediction Models: A Simulation Study. *Stat. Med.* **2025**, *44*, e10320. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.