

Leakage-controlled Crowdfunding Success Prediction: Do Feature Selection and SMOTE Improve Boosting-Based Models?

Adellia Nirma Azzahra¹, Riska Dhenabayu², Dhiyan Septa Wihara³, Muhammad Jiwa
 Akmal Sani Arman Rachman Liga⁴

^{1,2,3}Digital Business Study Program, faculty of Economics and Business, Universitas Negeri Surabaya,
 Surabaya, Indonesia

⁴Diploma of Digital Transformation, Eve College, Sydney, Australia

Received:

June 16, 2026

Revised:

July 15, 2026

Accepted:

September 3, 2026

Published:

October 24, 2026

Corresponding Author:

Author Name*:

Riska Dhenabayu

Email*:

riskadhenabayu@unesa.ac.id

DOI:

10.63158/journalisi.v8i5.1760

© 2026 Journal of
 Information Systems and
 Informatics. This open
 access article is distributed
 under a (CC-BY License)



Abstract Decision-making in digital financial information systems plays an important role in predicting crowdfunding success for investors, platform providers, and campaign creators. Ignoring data imbalance and the potential for information leaks in the pursuit of high accuracy actually backfires. This kind of approach creates the illusion of performance that looks perfect on paper but is actually unreliable in practice. This study proposes a leakage-controlled framework that combines Hybrid Feature Selection, RFECV, SMOTE, and boosting algorithms using a public Turkish crowdfunding dataset. The dataset contains 76.9% unsuccessful and 23.1% successful campaigns. Pearson Correlation and Mutual Information were used for feature selection, followed by RFECV for feature optimization. The data splitting stage was intentionally completed before feature filtering, and class balancing via SMOTE was applied only to the training data set to ensure that the evaluation remained objective without data leakage. Separate data testing showed that the XGB configuration, supported by GBM-RFECV and SMOTE, achieved the best results Precision, Recall, F1-score, ROC-AUC, and PR_AUC of 0.871, 0.906, 0.888, 0.975, 0.923, along side minority successful campaign. The results suggest that leakage-controlled feature selection and imbalance handling can provide a more realistic assessment of model performance for live crowdfunding campaign monitoring.

Keywords: Crowdfunding Success Prediction; Hybrid Feature Selection; Class Imbalance; Boosting Ensemble; Crowdfunding Analytics

1. INTRODUCTION

The presence of crowdfunding has opened up a path for entrepreneurs, the SME sector, and creative initiatives to secure internet-based financial support, especially when traditional sources of capital remain difficult to access [1]. The emergence of the digital ecosystem and high public participation in crowdfunding platforms have opened the door to faster, more efficient, and cost-effective fundraising compared to traditional methods [2]. However, in reality, many campaigns stall before their targets are met, leaving the success of these projects a major question mark for both initiators and investors [3]. A number of studies indicate that this success rate is influenced by a complex network of factors such as the characteristics of the initiative, the credibility of the campaign details, the initiator's communication skills, and investor behavior which form a complex data set that is too intricate to evaluate manually [4]. Therefore, a machine learning-based approach is needed that can identify key patterns in crowdfunding data providing a baseline for campaign-monitoring analytics, while noting that real-world operational decision-support deployment requires future multi-platform external validation. In general, predictive models in crowdfunding platforms serve three essential functions: validating a project's feasibility in the early stages using static data, monitoring the progress of ongoing campaigns in real time, and analyzing the final results after the fundraising campaign as part of a comprehensive evaluation.

Various previous studies have actively applied artificial intelligence systems to estimate the success of crowdfunding projects in order to support decision-making by stakeholders. Various previous studies have tested many methods, including filtering out important data, using a combination of intelligent models, making prediction results more plausible, and comparing the performance of each available model. Table 1 compares representative crowdfunding prediction studies in terms of their methods, contributions, and remaining limitations, thereby highlighting the research gaps addressed in this study. While the accuracy scores in previous studies do show high values, few have outlined a complete process from start to finish for feature selection, addressing data imbalance, and ensuring that test data remains confidential.

Table 1. Comparison of Previous Crowdfunding Prediction Studies

Study	Focus	Main Method	Main Contribution	Limitations
Kılınç & Aydın [4]	Turkish Crowdfunding	Pearson Correlation + Wrapper Feature Selection + Gradient Boosting	Demonstrated the effectiveness of hybrid feature selection for crowdfunding prediction.	Did not address class imbalance or feature-selection leakage.
Ionica et al [2]	Research Project Crowdfunding	Ensemble Machine Learning	Showed that ensemble learning improves crowdfunding readiness prediction.	No feature optimization or leakage-controlled validation.
Gregoriades & Themistocleous [3]	Crowdfunding Decision Support	Explainable Artificial Intelligence (XAI)	Improved interpretability of crowdfunding prediction models.	Did not investigate feature optimization or imbalance handling.
Avvad [5]	Turkish Crowdfunding	Comparative Machine Learning	Compared multiple machine learning classifiers for crowdfunding success prediction.	Focused mainly on predictive performance without integrating leakage control.

Study	Focus	Main Method	Main Contribution	Limitations
This Study	Turkish Crowdfunding	Hybrid Feature Selection + RFECV + SMOTE + GBM/XGBoost/CatBoost	Proposes a leakage-controlled prediction framework integrating hybrid feature selection, wrapper optimization, class imbalance handling, and boosting ensemble comparison.	The study is limited by the use of dynamic campaign features, a single crowdfunding platform and market, potential limitations of SMOTE-generated samples, and the absence of instance-level explainability.

In addition to predictive performance challenges, data leakage has emerged as an important issue in machine learning research. Data leakage is a significant issue in addition to predictive performance; in this study, data leakage is categorized into three types: (1) training-test data contamination, (2) leakage during variable selection, and (3) time-based or output-based bias. Data leakage in training and testing often occurs when the available data influences the model development process, resulting in overly optimistic performance results. Meanwhile, overfitting during feature selection occurs when the data used has not yet been split into training and test sets, so that information from the entire dataset influences the selection of predictors. Temporal or outcome-related leakage occurs when predictors contain information that becomes available only after a crowdfunding campaign has progressed, thereby exposing partial outcome signals that may not be accessible under certain prediction scenarios. According to [6],

performing feature selection before separating training and testing data may introduce information leakage, resulting in overly optimistic performance estimates and reduced model generalizability. Despite growing awareness of this issue, limited studies on crowdfunding success prediction have explicitly incorporated leakage-controlled procedures into the model development process. Consequently, the reliability and robustness of many existing crowdfunding prediction models remain uncertain when applied to unseen data. Moreover, although previous studies have demonstrated the effectiveness of feature selection and boosting algorithms for crowdfunding prediction [4], the potential benefits of combining hybrid feature selection, wrapper-based optimization, class imbalance handling, and leakage-controlled validation have not been extensively investigated.

Based on these limitations, this study proposes a Leakage-controlled Crowdfunding Success Prediction framework that integrates Hybrid Feature Selection, wrapper-based feature optimization, class imbalance handling, and boosting ensemble algorithms within a unified modeling pipeline. Pearson Correlation and Mutual Information are employed as complementary filter-based feature selection methods to identify both linear and non-linear relationships between features and the target variable. The selected predictors, including `number_of_supporters`, `number_of_awards`, and `updates`, are appropriate for campaign monitoring prediction because they represent information that becomes available during an active crowdfunding campaign. These variables are intentionally incorporated to capture campaign progress and are therefore suitable for monitoring-based prediction rather than pre-launch prediction. Accordingly, this study does not aim to perform pre-launch or early-stage prediction but rather to estimate campaign success using campaign-related information that evolves over time. The selected features are subsequently optimized using RFECV based on GBM and XGBoost estimators. To minimize the risk of information leakage, all feature selection and optimization procedures are performed exclusively on the training data after train-test splitting, while class imbalance is addressed using the SMOTE. The proposed framework is evaluated using GBM, XGBoost, and CatBoost classifiers. The novelty and primary contribution of this study lie in addressing these research gaps through a unified leakage-controlled framework that incorporates: (1) hybrid linear and non-linear feature selection, (2) wrapper-based RFECV optimization, (3) strict post-split SMOTE placement to prevent data contamination, and (4) a rigorous comparative evaluation of modern

gradient boosting classifiers under controlled validation. Beyond predictive performance, the framework emphasizes methodological reliability by separating the training and test data before feature selection and resampling. The comparison of GBM, XGBoost, and CatBoost also evaluates the performance of boosting classifiers under leakage-controlled validation.

2. METHODS

2.1. Dataset

This study utilizes the Turkish Crowdfunding Startups Dataset introduced by Kiliç and Aydın (2023) [4]. Collected from the Fongogo platform in 2022, the original dataset comprises 1,628 projects observations spanning campaign characteristics, backer activity, project content, user interactions, and promotional media. Following the removal of non-informative identifiers and leaky post-campaign variables during initial preparation, the refined dataset retains 30 attributes, consisting of 29 predictor features and one binary objective target variable, `success_status`. All attributes contain complete values without missing data, providing an ideal benchmark of mixed numerical and machine learning algorithms in crowdfunding analytics [4], [5].

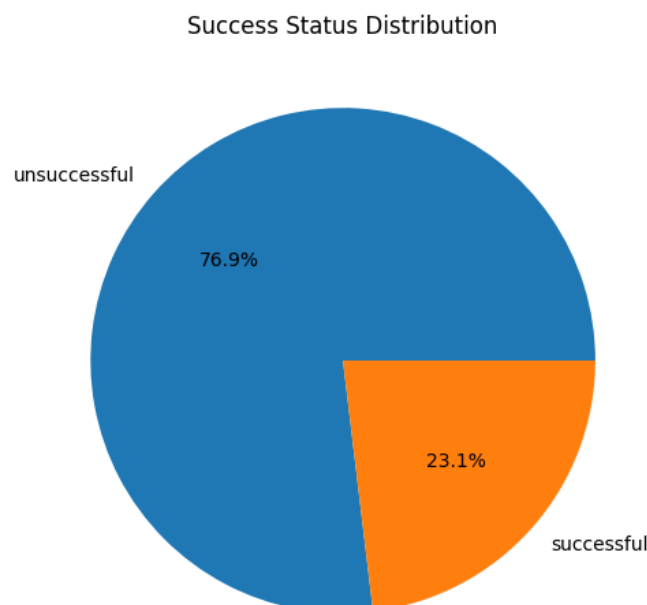


Figure 1. Distribution of `success_status` Classes

To prepare the data for unbiased model evaluation while preventing distribution distortion, the dataset was partitioned into an 80% training set (1,302 instances) and a 20% testing set (326 instances) using stratified sampling. The proportional partition across both classes is detailed in Table 2.

Table 2. Training and Testing Data Distribution

No	Data Type	Successful	Unsuccessful	Total
1	Training (80%)	301	1001	1302
2	Testing (20%)	75	251	326
Total		376	1252	1628

As reported in Table 4, the stratified split successfully preserved the original class ratio across both subsets. This partition serves as the foundation across all subsequent pipeline stages: preprocessing, hybrid feature selection, wrapper-based optimization, post-split SMOTE resampling, and gradient boosting evaluation.

2.2. Research Framework

This study incorporates the Leakage-controlled Hybrid Feature Selection approach, which combines Pearson Correlation, Mutual Information, GBM-based RFECV and XGBoost, as well as SMOTE, into a single integrated prediction pipeline for feature selection and machine learning. In contrast to earlier research that typically concentrated on feature selection or classification models independently. Using preprocessing, hybrid feature selection, wrapper feature optimization, addressing data imbalance, and classification using boosting techniques, the dataset served as the foundation for this study's model to forecast crowdfunding success [4], [6]. A train-test split, feature selection using filters and wrappers, data balancing using SMOTE, classification using GBM, XGBoost, and CatBoost, and performance evaluation using Accuracy, Precision, Recall, F1-Score, and ROC-AUC were the steps used to create the system. Figure 2, which depicts the study workflow from data preprocessing to model evaluation, displays the suggested research framework.

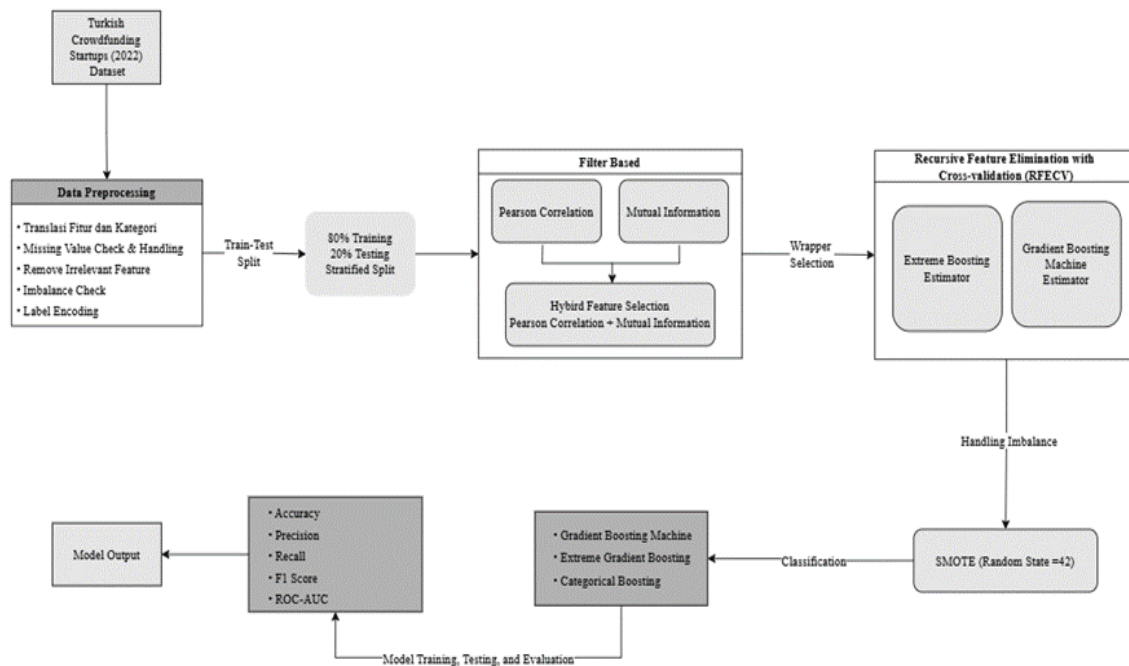


Figure 2. Research Framework

2.3. Data Preprocessing and Leakage-controlled Split

Preprocessing is used before feature selection. The steps involved include translating attribute names into English, standardizing data types, verifying missing values, and removing features that contain identifiers, redundancies, or are suspected of causing data leaks. The target variable, `success_status`, was designated as the binary objective. Table 3 details the retention status and exclusion criteria across all 38 original attributes. Categorical encoding transforms nominal variables into numeric indicators. To strictly prevent train-test leakage, all `LabelEncoders` were fitted exclusively on the training set X_{train} and subsequently transformed onto the testing set X_{test} . While Pearson Correlation primarily detects linear relationships across numeric and monotonic encoded levels, its known theoretical limitation on nominal encodings was explicitly addressed and mitigated by integrating Mutual Information (MI), which is non-parametric and model-free, capable of capturing arbitrary non-linear associations and categorical dependencies without assuming ordinality.

Table 3. Original Attributes and Preprocessing Status

No	Original Attributes	Status	Reason
1	id	Removed	Identifier attribute that does not contribute to prediction.
2	platform_name	Retained	Represents crowdfunding platform characteristics.
3	crowdfunding_type	Retained	Represents the crowdfunding mechanism used.
4	category	Retained	Provides information about the project domain.
5	fund_type	Retained	Describes the funding scheme of campaigns.
6	project_name	Retained	Represents project-related textual information.
7	project_owner	Removed	High-cardinality identifier that may reduce model generalizability.
8	project_owner_gender	Retained	Represents demographic characteristics of project owners.
9	number_of_supported_projects	Retained	Represents prior crowdfunding participation experience.
10	number_of_subscribed_projects	Removed	Removed due to limited relevance and redundancy with supported project information.
11	number_of_projects_owners	Retained	Represents project ownership characteristics.
12	number_of_team_members_in_projects	Retained	Represents project team composition.
13	location	Retained	Represents campaign geographical information.
14	region	Retained	Represents broader geographical information

No	Original Attributes	Status	Reason
			(missing values were imputed using the mode).
15	year	Retained	Captures temporal characteristics of campaigns.
16	project_start_date	Removed	Replaced by the derived duration feature (number_of_days).
17	project_end_date	Removed	Replaced by the derived duration feature (number_of_days).
18	number_of_days	Retained	Represents campaign duration.
19	promotion_video	Retained	Indicates the availability of promotional videos.
20	video_length	Retained	Represents promotional video duration.
21	number_of_images	Retained	Represents the amount of visual campaign information.
22	faq	Retained	Indicates the availability of frequently asked questions information.
23	updates	Retained	Represents campaign communication activities.
24	comments	Retained	Represents supporter engagement during campaigns.
25	number_of_supporters	Retained	Represents supporter participation.
26	number_of_awards	Retained	Represents campaign incentives offered to supporters.
27	number_of_team_members	Retained	Represents project team size.
28	website	Retained	Indicates the availability of an external website.

No	Original Attributes	Status	Reason
29	social_media	Retained	Indicates the availability of social media accounts.
30	number_of_social_media_accounts	Retained	Represents the extent of social media utilization.
31	social_media_followers	Retained	Represents the social media reach of campaigns.
32	number_of_tags	Retained	Represents campaign metadata information.
33	number_of_words_in_content	Retained	Represents the amount of textual campaign information.
34	project_description	Retained	Represents project description information.
35	target_amount	Retained	Represents the fundraising goal specified by campaigns.
36	amount_collected	Removed	Potentially introduces target leakage because it directly reflects fundraising outcomes.
37	support_rate	Removed	Available only after campaign completion and may introduce information leakage.
38	success_status	Retained	Used as the prediction target variable.

Following preprocessing, the dataset was partitioned into training (80%) and testing (20%) subsets using stratified sampling to preserve class proportions. Unlike workflows that perform feature selection globally, this study implements a strict three-tier leakage-control protocol to prevent optimistic evaluation bias and ensure real-world validity [6]:

- 1) Train-Test Isolation: Partitioning was executed prior to any feature selection, resampling, or model fitting, keeping the test set entirely unseen until final evaluation.
- 2) Feature-Selection Isolation: All downstream selection stages including Pearson Correlation, Mutual Information, and RFECV wrapper optimization were fitted

strictly on the training subset, ensuring no test-distribution parameters influenced feature ranking or elimination.

- 3) Temporal Scope Alignment: The objective is framed explicitly as an active campaign-monitoring task rather than pre-launch forecasting. Consequently, dynamic progression metrics `number_of_supporters`, `updates`, `number_of_awards` are observable during live fundraising and serve as legitimate monitoring indicators rather than sources of target leakage.

2.4. Hybrid Feature Selection

1) Pearson Correlation

Pearson Correlation Coefficient (PCC) was applied to evaluate the linear dependence between each candidate feature and the target variable, `success_status`. The correlation coefficient r ranges from -1 to +1, where values near ± 1 indicate strong positive or negative linear relationship, and values near 0 signify the absence of linear association. Equation (1) can be used to compute Pearson correlation mathematically.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \quad (1)$$

Where $\bar{x}$ is the mean of the feature. $\bar{y}$ is the mean of the targets, r is the correlation coefficient, x_i is the feature value, and y_i is the target. Due to its computational efficiency and interpretability, PCC effectively identifies direct linear predictors [4]. In this framework, PCC was computed strictly on the training subset following data partitioning to eliminate train-test leakage. A permissive threshold of $r > 0.05$ was established to retain features demonstrating at least weak linear association with campaign success. This conservative cutoff prevents the premature exclusion of marginally correlated variables that may provide complementary predictive value when combined non-linearly in downstream boosting models.

2) Mutual Information

Mutual Information (MI) was employed to evaluate feature relevance based on information theory principles [7]. Unlike Pearson correlation, which captures only linear dependencies, MI measures the shared information between variables without parametric constraints, enabling the detection of both linear and complex non-linear relationships

with the target variable [7], [8]. Equation (2) is used to determine the mutual information value.

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \quad (2)$$

Where $p(x, y)$ is the joint probability distribution, $p(x)$ and $p(y)$ denote the respective marginal probability distributions. Higher value of $I(X, Y)$ signify greater dependency and predictive capacity regarding the outcome class [8]. In this framework, MI was computed exclusively on the training partition to uphold leakage-control protocols. To curate a focused subset and constrain downstream computational complexity, the top 10 ranked features by MI score were selected. This strategy ensures the inclusion of highly informative, non-linear predictors that standard correlation analysis would otherwise overlook before proceeding to wrapper optimization.

3) Hybrid Mechanism

Hybrid feature selection integrates Pearson Correlation Coefficient (PCC) and Mutual Information (MI) to capture both linear associations and complex non-linear dependencies relative to the target variable, `success_status` [4], [7]. While PCC detects monotonic linear relationships, MI extracts non-parametric dependencies, mitigating the limitations of correlation-based filtering on encoded categorical features[8]. Equation (3) provides a mathematical expression for the feature combination process.

$$F_{\text{Hybrid}} = F_{\text{Pearson}} \cup F_{\text{MI}} \quad (3)$$

Where F_{MI} represents the set of features chosen using mutual information and F_{Pearson} represents the set of features chosen using Pearson correlation. Recursive Feature Elimination with Cross Validation (RFECV) was employed in the feature optimization step when all features chosen by both approaches were combined into a single feature subset. In this study, both methods were solely applied to the training data.

Linear filtering is performed using Pearson correlation, while mutual information is used to capture nonlinear relationships among variables. The results from both filtering processes are then subjected to a hybrid feature selection process to mitigate the

limitations arising from reliance on the categorical attributes that have been encoded in the Pearson correlation.

2.5. Wrapper Feature Optimization

1) GBM-RFECV

A feature selection wrapper technique called RFECV works by progressively removing features according to their significance until an ideal subset of features is found. The RFE principle is extended through the RFECV approach, in which less relevant features are recursively removed based on their importance as determined by the model, while cross-validation is used to determine the optimal number of features to retain. The RFE approach was originally introduced as a model-based feature selection strategy [9]. In this study, GBM was selected as one of the estimators. In order to minimize the loss function at each iteration, GBM constructs models iteratively by adding a new weak learner. In general, the GBM optimization process can be described using equation (4).

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \quad (4)$$

Where $h_m(x)$ is the additional weak learner, γ_m is the contribution weight of the new model, $F_m(x)$ is the model at the m-th iteration, and $F_{m-1}(x)$ is the prior model. A more relevant feature subset can be produced by using GBM as an estimator in RFECV, which enables the feature selection process to take into account intricate correlations between features and the goal [4]. Using feature selection before training a model can improve prediction accuracy and reduce model complexity [10]. In this study, features with poor contribution are iteratively removed from the feature subset generated from Hybrid Feature Selection using GBM-RFECV until an ideal feature combination is found based on cross-validation outcomes. In the RFECV procedure, feature elimination was performed iteratively with a step size of one feature using accuracy as the evaluation metric. Stratified 5-fold cross-validation (StratifiedKFold) with shuffle=True and random_state=42 was employed to preserve class distributions across folds. The GBM estimator was configured with n_estimators=200, max_depth=3, learning_rate=0.1, and random_state=42. The minimum number of selected features was set to one in accordance with the default RFECV configuration.

2) XGB-RFECV

XGBoost is used as an estimator in the feature evaluation process of XGB-RFECV, an extension of the RFECV approach. The use of a stepwise boosting process in XGB can improve prediction accuracy by combining multiple decision trees to produce a reliable model [10]. This model provides better predictions than conventional statistical models because it is capable of learning complex data patterns. [11]. To balance predictive accuracy and model simplicity, XGBoost minimizes a regularized objective function at each iteration t , which combines a specific loss function with a regularization term to prevent overfitting [12], [2], [13], [14]. The objective function is formulated in equation (5).

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (5)$$

Where $\mathcal{L}^{(t)}$ represents the object function at the t -th iteration, l is the differentiable convex loss function measuring the error between the actual label y_i and the predicted value, $f_t(x_i)$ denotes the t -th decision tree structure, and $\Omega(f_t)$ represents the regularization term that penalizes model complexity. In the RFECV process, XGBoost is used to assign feature importance scores, which are then used as the basis for gradually eliminating features until the optimal feature subset is obtained. XGBoost's ability to produce stable predictions across various levels of data imbalance makes it an effective estimator for the feature evaluation process [14]. Combining feature selection with XGBoost can improve model accuracy while maintaining the interpretability of prediction results [13]. In this study, XGB-RFECV was applied to the same Hybrid Feature Selection results as GBM-RFECV, allowing for a direct comparison of the effectiveness of both estimators in generating an optimal feature subset for predicting crowdfunding success. The selection of GBM and XGBoost as RFECV estimators was motivated by their ability to capture complex non-linear relationships and feature interactions commonly found in crowdfunding datasets. Furthermore, both algorithms have demonstrated strong predictive performance in classification tasks involving tabular data and provide feature importance measures that can effectively guide the recursive feature elimination process [13], [14]. The use of these two estimators in feature optimization is used to evaluate and identify which of the two estimators produces the most effective features for predicting crowdfunding success. The XGB-RFECV implementation employed an XGBClassifier estimator with `n_estimators=200`, `max_depth=5`, `learning_rate=0.1`,

eval_metric='logloss', and random_state=42. Similar to GBM-RFECV, recursive feature elimination was conducted using a step size of one feature and accuracy as the scoring metric under Stratified 5-fold cross-validation with shuffle=True and random_state=42. The minimum number of features is defined as at least one informative feature remaining after the elimination process.

2.6. Imbalanced Data-Handling using SMOTE

When one class has much more samples than the others, the model favors the dominant class and ignores the minority class, a phenomenon known as class imbalance. The SMOTE, an oversampling technique proposed by Chawla et al. [15], that creates new synthetic samples depending on the proximity between samples within the minority class, is used in this study to address this problem. Equation (6) can be used to express the synthetic sample generation process in SMOTE [15]:

$$x_{\text{new}} = x_i + \lambda(x_{nn} - x_i) \quad (6)$$

Where λ is a random number in the interval [0,1], x_i is a minority class sample, and x_{nn} is one of x_i 's closest neighbors. By increasing the diversity of data in the minority class, SMOTE helps lower the danger of overfitting, in contrast to random oversampling, which merely duplicates the data [16]. Only after the Hybrid Feature Selection procedure and feature optimization using RFECV were finished was SMOTE applied to the training data in this study. The goal of applying SMOTE exclusively to the training data after feature selection and train-test splitting is to prevent information leakage while preserving the original distribution of the testing data for unbiased model evaluation [17]. In this study, SMOTE was applied exclusively to the final feature subset obtained after Hybrid Feature Selection and RFECV had been completed. The oversampling process was performed only on the training dataset (X_train_final and y_train), whereas the testing dataset remained untouched throughout both feature selection and imbalance handling procedures. After the training data has been balanced in proportion for the training process, the model is tested using the original dataset so that the measured performance truly reflects the actual class distribution.

2.7. Boosting-Based Classification Models

1) GBM

GBM is a boosting-based ensemble learning method that sequentially combines weak learners to construct an additive predictive model by minimizing a specified loss function [18]. This approach enabled the model to identify the interrelationships among features and achieve competitive performance across various types of data [14]. In this study, GBM was used as a classification model after undergoing feature selection, feature optimization, and data balancing.

2) XGBoost

XGBoost extends gradient tree boosting through a regularized objective function and computational optimizations designed to improve scalability, predictive performance, and model generalization [12]. In this work, XGBoost was employed as a classification model to forecast crowdfunding success using the chosen features and as an estimator at the XGB-RFECV stage to assess feature importance. XGBoost is used in many machine learning-based prediction studies because of its capacity to handle nonlinear relationships, feature interactions, and unbalanced data [13], [14].

3) CatBoost

CatBoost is a boosting technique designed to use an ordered boosting mechanism during training to generate more robust predictive models. CatBoost incorporates ordered boosting and efficient handling of categorical features to reduce prediction shift and improve model generalization [19]. This technique can perform well on a variety of classification data types while minimizing prediction shift [3]. [19] shows that CatBoost can outperform a number of different machine learning algorithms, including Random Forest, Decision Tree, and LightGBM, in terms of accuracy. [13] This work also shows that CatBoost can perform well on health prediction datasets and is a competitive model for classification tasks. Using a feature subset obtained from optimization and data that had been balanced using SMOTE, CatBoost was employed as a classification model in this study to forecast the success of crowdfunding projects. For consistency across all classification models, CatBoost utilized the label-encoded feature representations produced during preprocessing rather than raw categorical features. Although CatBoost natively supports target-based encoding for raw categorical variables, all classifiers were trained on identical label-encoded inputs to ensure strict architectural parity and

experimental consistency across the boosting ensemble comparisons [20]. This study acknowledge that bypassing CatBoost's native categorical handling represents a controlled methodological trade-off that standardizes input feature matrices across all candidate models. Label-encoded features were intentionally used to ensure that all classifiers were trained and evaluated using the same input representation, thereby enabling a fair and consistent comparison of model performance. In order to identify the model that performs the best, the resulting prediction results were compared with GBM and XGBoost and assessed using Accuracy, Precision, Recall, F1-Score, and ROC-AUC.

2.8. Performance Evaluation Metrics

Model performance was evaluated using a confusion matrix with TP, TN, FP, and FN as reference metrics.. This study uses accuracy, precision, recall, F1-score, ROC-AUC, and PR-AUC as evaluation metrics. Equation (7) is used to calculate accuracy, which measures the total proportion of correct predictions. Equations (8) and (9) are used to determine precision and recall, which assess the model's capacity to correctly identify the positive class. Additionally, in datasets with an unbalanced class distribution, the F1-Score in Equation (10) is used to establish a balance between Precision and Recall values [19]. An evaluation of the differences between classes at various classification thresholds was performed using ROC-AUC based on the relationship between TPR and FPR, which were computed using Equations (11) and (12), consistent with the foundational ROC analysis [21]. PR-AUC was also used. Precision-Recall analysis provides a complementary evaluation to ROC analysis and is particularly informative for imbalanced classification problems because it focuses on predictive performance for the positive class [22], [23]. The model's performance was evaluated against all performance metrics after applying feature selection, feature optimization, and data balancing using SMOTE.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (10)$$

$$TPR = \frac{TP}{TP+FN} \quad (11)$$

$$FPR = \frac{FP}{FP+TN} \quad (12)$$

2.9. Leakage-controlled Validation Analysis

Two preprocessing scenarios were used to compare the effect of data leakage on predictive validity. The first leakage-prone performed feature selection prior to the train-test split, allowing global data distributions to contaminate the selection phase and introduce optimistic evaluation bias [6]. The second the proposed leakage-controlled pipeline strictly partitioned the dataset into training and testing subsets beforehand. Consequently, feature selection and RFECV optimization relied exclusively on the training subset, keeping the test set entirely unseen until final inference to preserve evaluation integrity[6]. To ensure an unbiased comparison, both workflows shared identical downstream procedures: class balancing via SMOTE applied solely to the training data, model training using the best-performing gradient boosting classifier, and evaluation via Accuracy and F1-score. Earlier feature selection has been shown to affect model performance and to evaluate the robustness of the proposed framework.

2.10. Experimental Settings

All models were evaluated using the same experimental setup. An 80:20 train-test split was applied to the dataset, with `random_state = 42` used to maintain a consistent data partition across all experiments. Feature optimization using RFECV with 5-fold cross-validation, a step size of 1, and F1-score as the evaluation metric. The GBM model used 200 estimators, a learning rate of 0.1, and a maximum tree depth of 3. For XGB, the model was set to 200 estimators, a maximum depth of 5, and a learning rate of 0.1. CatBoost was configured with 200 iterations, a tree depth of 5, and a learning rate of 0.1. These hyperparameter settings were kept fixed throughout the experiments to enable a fair comparison among the three ensemble learning models and to facilitate reproducibility.

Model stability was also evaluated using repeated stratified 5-fold cross-validation with 10 repetitions, resulting in 50 evaluation iterations. Stratification preserved the original class distribution across all folds. To prevent data leakage, data resampling via SMOTE was embedded strictly inside each individual training fold after partitioning, ensuring the

validation folds remained untouched and drawn exclusively from the empirical distribution. Furthermore, the RFECV feature mask obtained from the primary training phase was kept fixed to evaluate generalization stability. The mean and standard deviation (SD) were calculated for all validation iterations to evaluate the metrics of Accuracy, Precision, Recall, F1-score, ROC-AUC, and PR-AUC. Repeated stratified cross-validation was used to examine the stability of the model results, and 95% confidence intervals were calculated for each evaluation metric. Statistical differences between models were assessed using the Friedman test. The Nemenyi post-hoc test was used for paired comparisons when significant differences were found.

2.11. Software Environment

All experiments were conducted using Google Colaboratory on a Linux 6.6.122+ cloud runtime equipped with an Intel(R) Xeon(R) CPU @ 2.20GHz (2 vCPUs) and 12.67 GB of system RAM. The experimental workflow was implemented in Python 3.12.13 using primary computational and machine learning libraries, including Pandas 2.2.3, NumPy 2.1.3, Scikit-learn 1.6.1, XGBoost 3.4.1, CatBoost 1.2.10, Imbalanced-learn 0.14.2, and Matplotlib 3.10.0. Table 4 outlines the comprehensive hardware and software environment used in this study to ensure reproducibility

Table 4. Software Specification

Component	Specification
Platform	Google Colaboratory (Cloud Instance)
Operating System	Linux 6.6.122+
Processor	Intel(R) Xeon(R) CPU @ 2.20GHz (2 vCPUs)
System memory	12.67 GB
Programming language	Python 3.12.13
Pandas	2.2.3
NumPy	2.1.3
Scikit-learn	1.6.1
XGBoost	3.4.1
CatBoost	1.2.10
Imbalanced-learn	0.14.2
Matplotlib	3.10.0

2.12. Baseline Models and Majority Baseline

Given that the dataset exhibits intrinsic class imbalance comprising 76.9% unsuccessful $x=0$ and 23.1% successful campaigns $y=1$, a non learning majority baseline was formulated. This trivial classifier deterministically assigns all instances in the testing set to the predominant class $y=0$ regardless of predictor attributes. It serves as a strict lower-bound threshold to detect any potential accuracy paradox, verifying that subsequent model performance reflects true discriminative capability on the minority positive class rather than mere exploitation of the class prior.

The primary machine learning baselines were constructed using all 29 preprocessed features retained after attribute pruning, prior to any feature filtering or wrapper optimization. In the baseline test, the classification model was tested using the original training data without SMOTE resampling. Hyperparameters were kept strictly identical to the main experimental setup $n_estimators = 200$, $learning_rate = 0.1$ to provide an equitable point of departure. This baseline test is used as a reference point for models that have not yet undergone more complex predictive performance testing.

3. RESULTS AND DISCUSSION

3.1. Performance Evaluation

3.1.1. Hybrid Feature Selection Results

Referring to Figure 3, the feature `number_of_supporters` shows a high correlation in feature selection using both PCC and MI. This evidence indicates that this feature has a linear relationship with the campaign success. 24 features were selected using PCC, and 10 features were selected using MI.

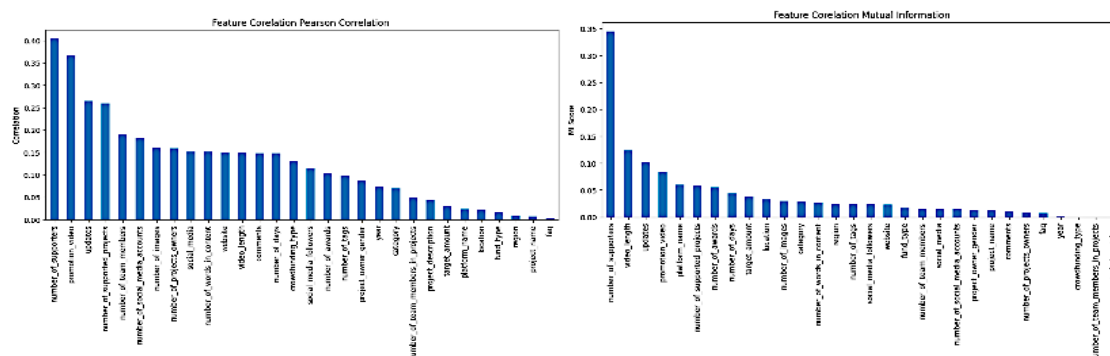


Figure 3. Comparison of Results from the Pearson Correlation and Mutual Information

The PCC and MI hybrid process yielded 25 features, as shown in Table 5. This selection filtered out less informative features from the previous stage and produced more targeted features for use in the next stage.

Table 5. Results of Selected Features

Method		Selected Feature
Filter	PCC	crowdfunding type, category, project owner gender, number of supported projects, number of projects owners, number of team members in projects, year, number of days, promotion video, video length, number of images, updates, comments, number of supporters, number of awards, number of team members, website, social media, number of social media accounts, social media followers, number of tags, number of words in content
	MI	number of supporters, video length, updates, promotion video, platform name, number of supported projects, number of awards, number of days, target amount, location
Hybrid Selection Feature	PCC + MI	category, comments, crowdfunding type, location, number of awards, number of days, number of images, number of projects owners, number of social media accounts, number of supported projects, number of supporters, number of tags, number of team members, number of team members in projects, number of words in content, platform name, project owner gender, promotion video, social media, social media followers, target amount, updates, video length, website, year
Wrapper	RFECV+XGB Estimator	number of supporters, target amount, website, number of team members in projects, number of awards, year, number of words in content, location, video length, number of days, category, platform name, promotion video, number of projects owners,

Method	Selected Feature
RFECV+GBM Estimator	comments, updates, number of tags, number of images.
	number of supporters, target amount, year, website, number of projects owners, promotion video, number of awards, video length, location, updates, number of words in content, number of supported projects, category, platform name, number of days, number of images, project owner gender, number of team members, social media followers, comments, number of tags, number of social media accounts, number of team members in projects.

3.1.2. Wrapper Feature Optimization Results

Implementation of the RFECV wrapper method, as shown in Figure 4, can optimize the results of hybrid feature selection using cross-validation, yielding 18 features for the XGB estimator and 23 features for the GBM estimator. The final features, as shown in Table 5, are then used for the classification modeling stage.

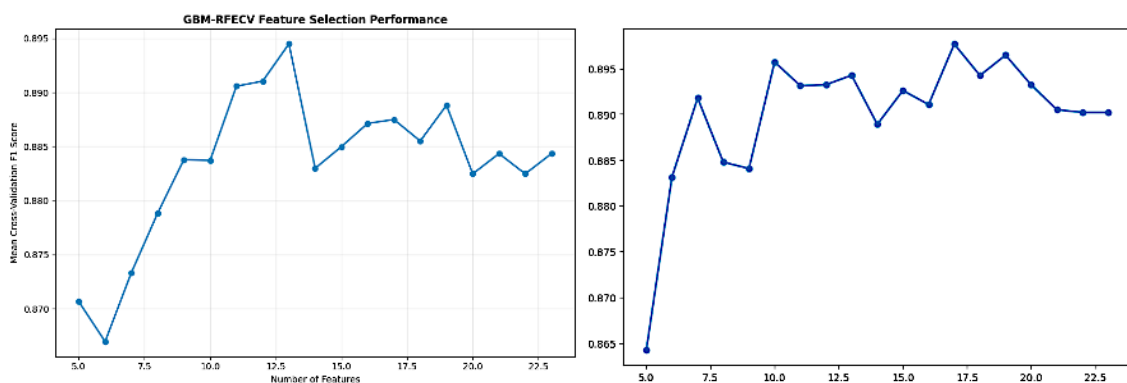


Figure 4. Mean Cross-Validation Scores of XGB-RFECV and GBM-RFECV Feature Selection

3.1.3. Class Distribution After SMOTE

The class distributions before and after SMOTE are shown in Figure 5, where “successful” is represented by 1 and “unsuccessful” by 0. Balanced data can minimize bias in the modeling process.

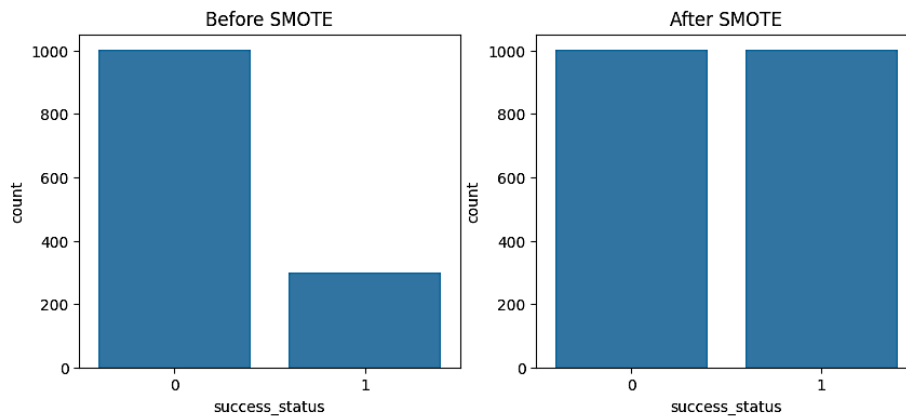


Figure 5. Class Distribution Before and After SMOTE

3.1.4. Classification Performance

Table 6 summarizes the predictive performance across all baseline configurations, ablation study variants, and the proposed leakage-controlled framework evaluated on the unseen test set. Given that the target class exhibits substantial imbalance (76.9% unsuccessful vs. 23.1% successful campaigns), all reported Precision, Recall, F1-Score, and PR-AUC metrics strictly reflect performance on the minority positive class (`success_status = 1`). The empirical results demonstrate that the trivial Majority Class Baseline yields an Accuracy of 0.7699, ROC-AUC of 0.500, and PR-AUC of 0.231, while achieving zero across Precision, Recall, and F1-score due to its total inability to detect successful campaigns. In contrast, the unoptimized ML Baselines operating on all 29 initial features establish a high raw baseline, where CatBoost and XGBoost achieved F1-scores of 0.882 and 0.874, respectively. Furthermore, the ROC curves illustrated in Figure 6 visually corroborate the strong discriminative capability across the evaluated boosting models under different feature selection setups.

Table 6. Model Comparison Results

Scenario	FS	Wrapper	Class-Imbalanced	Classifier	Accuracy	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
Proposed (Leakage-Controlled)	Hybrid PCC + MI	XGB-RFECV	SMOTE	CatBoost	0.929	0.842	0.853	0.847	0.978	0.938
				GBM	0.944	0.855	0.866	0.860	0.975	0.925
				XGB	0.935	0.864	0.853	0.859	0.982	0.942
		GBM-RFECV	SMOTE	CatBoost	0.938	0.848	0.893	0.870	0.975	0.933
				GBM	0.935	0.864	0.853	0.859	0.973	0.923
				XGB	0.947	0.871	0.906	0.888	0.975	0.923
Baseline				CatBoost	0.947	0.914	0.853	0.882	0.980	0.929

Scenario	FS	Wrapper	Class- Imbalanced	Classifier	Accuracy	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
Baseline + FS	Hybrid PCC + MI			GBM	0.950	0.864	0.853	0.859	0.979	0.925
				XGB	0.941	0.868	0.880	0.874	0.975	0.932
				CatBoost	0.932	0.863	0.840	0.851	0.977	0.927
				GBM	0.944	0.890	0.866	0.878	0.976	0.932
				XGB	0.932	0.863	0.840	0.851	0.975	0.930
Baseline + RFECV		XGB- RFECV		CatBoost	0.947	0.902	0.866	0.884	0.980	0.929
				GBM	0.950	0.904	0.880	0.891	0.981	0.935
				XGB	0.935	0.864	0.853	0.859	0.975	0.930
Baseline + SMOTE			SMOTE	CatBoost	0.941	0.868	0.880	0.874	0.976	0.942
				GBM	0.947	0.881	0.893	0.887	0.981	0.932
				XGB	0.944	0.860	0.907	0.883	0.974	0.926
Baseline + FS + RFECV	Hybrid PCC + MI	XGB- RFECV		CatBoost	0.935	0.875	0.840	0.857	0.976	0.930
				GBM	0.938	0.887	0.840	0.863	0.975	0.930
				XGB	0.932	0.863	0.840	0.851	0.973	0.923
Baseline + FS + SMOTE	Hybrid PCC + MI		SMOTE	CatBoost	0.944	0.870	0.893	0.881	0.978	0.930
				GBM	0.944	0.890	0.866	0.878	0.976	0.932
				XGB	0.944	0.870	0.893	0.881	0.975	0.923
Baseline + RFECV + SMOTE		XGB- RFECV	SMOTE	CatBoost	0.947	0.871	0.906	0.888	0.980	0.939
				GBM	0.947	0.881	0.893	0.887	0.979	0.915
				XGB	0.947	0.881	0.893	0.887	0.981	0.931
Without number_of_supporter	Hybrid PCC + MI	XGB- RFECV	SMOTE	CatBoost	0.865	0.691	0.746	0.717	0.902	0.741
				GBM	0.852	0.658	0.746	0.700	0.905	0.759
				XGB	0.865	0.686	0.760	0.721	0.904	0.728
GBM-RFECV Leakage- Prone	XGB-RFECV Leakage- Prone	GBM- RFECV	SMOTE	CatBoost	0.948	0.959	0.936	0.947	0.978	0.985
				GBM	0.938	0.946	0.928	0.937	0.977	0.983
				XGB	0.944	0.958	0.928	0.943	0.978	0.984
XGB-RFECV Leakage- Prone	XGB-RFECV Leakage- Prone	XGB- RFECV	SMOTE	CatBoost	0.944	0.958	0.928	0.943	0.979	0.984
				GBM	0.944	0.954	0.932	0.943	0.976	0.983
				XGB	0.946	0.962	0.928	0.945	0.980	0.985
Majority Class Baseline					0.7699	0.000	0.000	0	0.5	0.231

The Ablation experiments systematically reveal the modular contribution of each pipeline component:

- 1) Feature Selection Alone (Baseline + FS): Compacting the feature space to 25 features via Hybrid PCC+MI maintained competitive discrimination (GBM achieved an F1-score of 0.878), confirming that 4 redundant features could be removed without performance degradation.
- 2) Imbalance Resampling (Baseline + SMOTE): Introducing SMOTE to the baseline improved the positive-class Recall across all models (reaching up to 0.907 on XGBoost), lifting the F1-score of GBM to 0.887.

- 3) Proposed Multi-stage Pipeline: The fully integrated pipeline combining Hybrid Feature Selection, wrapper optimization, and post-split SMOTE achieved the most robust trade-off. The GBM-RFECV + SMOTE + XGBoost configuration attained an Accuracy of 0.947, Precision of 0.871, Recall of 0.906, F1-score of 0.888, ROC-AUC of 0.975, and PR-AUC of 0.923.
- 4) Feature Ablation (Without number_of_supporters): Deliberately excluding the top dynamic indicator resulted in a pronounced performance drop across all classifiers (F1-score declining to 0.700–0.721 and PR-AUC falling to 0.728–0.759). The results indicate that live backer engagement features played an important role in predicting campaign success during campaign monitoring.

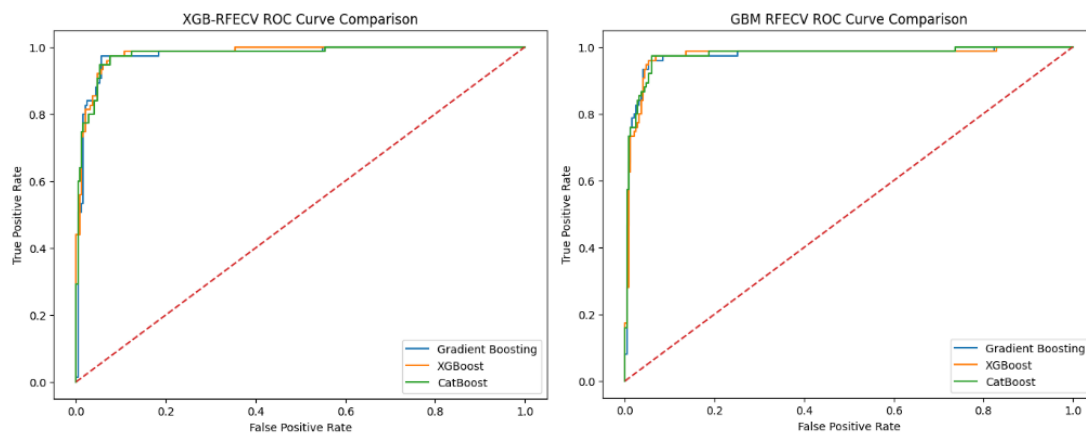


Figure 6. ROC Curve

Table 7 provides details on the predictions for the positive and negative classes. The GBM-RFECV and XGB combination performed best at detecting positive instances (TP: 68 data points) and made the fewest false negatives (FN: 7 data points). In addition, all models performed well in identifying negative instances, with TN remaining stable in the range of 239–241 data points.

Table 7. Confusion Matrix

Model	XGB-RFECV			GBM-RFECV		
	Catboost	XGB	GBM	Catboost	XGB	GBM
TN	239	241	240	239	241	241
FP	12	10	11	12	10	10
FN	11	11	10	8	7	11
TP	64	64	65	67	68	64

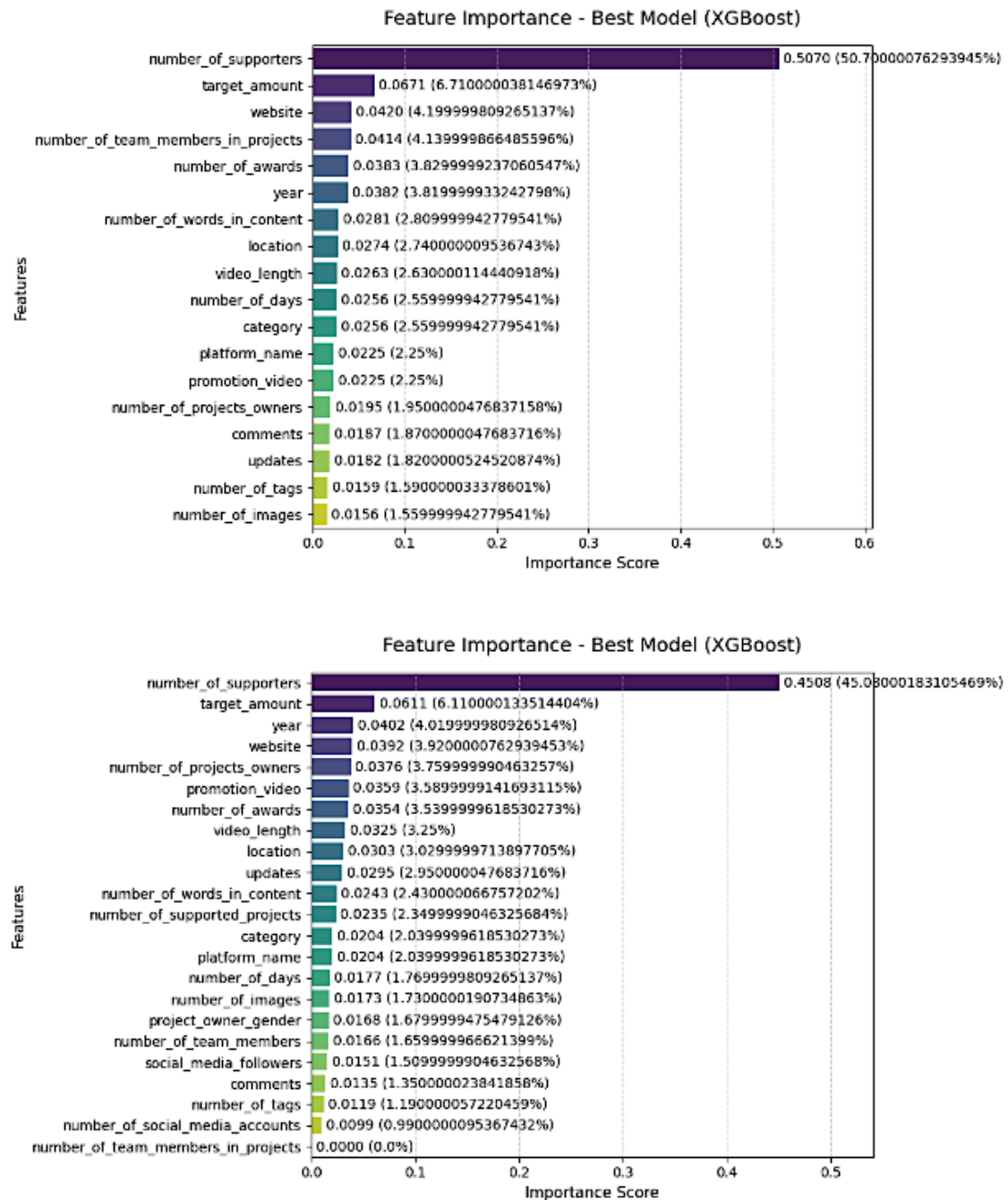


Figure 7. Feature Importance of the Best Model

The two diagrams in Figure 7, show that `number_of_supporters` is the strongest factor in XGB, with a contribution of 0.507 to XGB-RFECV and 0.450 to GBM-RFECV. These results mean that the number of supporters is the best predictor of a monitoring project's success, followed by websites, `promotion_video`, and updates.

Table 8. Class-Specific Performance

Wrapper	Classifier	Target Class	Precision	Recall	F1- Score	Support
XGB- RFECV	CatBoost	Unsuccessful	0.956	0.952	0.954	251
		Successful	0.842	0.853	0.847	75
		Macro Average	0.899	0.902	0.900	
		Weighted Average	0.929	0.929	0.939	
	XGB	Unsuccessful	0.956	0.952	0.954	251
		Successful	0.864	0.853	0.859	75
		Macro Average	0.910	0.906	0.908	
		Weighted Average	0.935	0.935	0.935	
	GBM	Unsuccessful	0.960	0.956	0.958	251
		Successful	0.855	0.866	0.860	75
		Macro Average	0.907	0.911	0.909	
		Weighted Average	0.935	0.935	0.935	
GBM- RFECV	CatBoost	Unsuccessful	0.967	0.952	0.959	251
		Successful	0.848	0.893	0.870	75
		Macro Average	0.907	0.922	0.915	
		Weighted Average	0.940	0.938	0.939	
	XGB	Unsuccessful	0.971	0.960	0.958	251
		Successful	0.871	0.906	0.888	75
		Macro Average	0.921	0.933	0.927	
		Weighted Average	0.948	0.947	0.948	
	GBM	Unsuccessful	0.956	0.960	0.958	251
		Successful	0.864	0.853	0.859	75
		Macro Average	0.910	0.906	0.908	
		Weighted Average	0.935	0.935	0.935	

Based on the specific class performance table in Table 8, the model used was able to identify the unsuccessful class with an F1-score and precision consistently above 0.950 for 251 data points. Meanwhile, for the successful project with 75 data points, the model

was able to identify them with a best F1-score of 0.888 in the GBM-RFECV and XGB scheme.

Table 9. Repeated Stratified Cross-Validation and CI 95%

Wrapper	Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
		Mean $\pm$ SD [95% CI]	Mean $\pm$ SD [95% CI]	Mean $\pm$ SD [95% CI]	Mean $\pm$ SD [95% CI]	Mean $\pm$ SD [95% CI]	Mean $\pm$ SD [95% CI]
XGB-RFECV	XGB	0.947 $\pm$ 0.012 [0.944, 0.951]	0.860 $\pm$ 0.037 [0.849, 0.871]	0.927 $\pm$ 0.030 [0.918, 0.936]	0.892 $\pm$ 0.023 [0.885, 0.898]	0.982 $\pm$ 0.006 [0.980, 0.982]	0.941 $\pm$ 0.024 [0.934, 0.948]
	CatBoost	0.944 $\pm$ 0.011 [0.941, 0.948]	0.848 $\pm$ 0.033 [0.838, 0.857]	0.930 $\pm$ 0.029 [0.921, 0.938]	0.886 $\pm$ 0.022 [0.880, 0.893]	0.982 $\pm$ 0.006 [0.981, 0.984]	0.942 $\pm$ 0.023 [0.936, 0.949]
	GBM	0.941 $\pm$ 0.011 [0.938, 0.945]	0.845 $\pm$ 0.034 [0.835, 0.855]	0.918 $\pm$ 0.034 [0.908, 0.928]	0.879 $\pm$ 0.022 [0.873, 0.886]	0.980 $\pm$ 0.006 [0.978, 0.982]	0.934 $\pm$ 0.023 [0.927, 0.940]
GBM-RFECV	XGB	0.947 $\pm$ 0.010 [0.944, 0.950]	0.856 $\pm$ 0.013 [0.847, 0.865]	0.929 $\pm$ 0.027 [0.921, 0.937]	0.890 $\pm$ 0.020 [0.884, 0.896]	0.981 $\pm$ 0.006 [0.979, 0.983]	0.940 $\pm$ 0.022 [0.933, 0.946]
	CatBoost	0.941 $\pm$ 0.011 [0.938, 0.945]	0.838 $\pm$ 0.036 [0.828, 0.849]	0.928 $\pm$ 0.029 [0.920, 0.937]	0.880 $\pm$ 0.022 [0.874, 0.887]	0.981 $\pm$ 0.006 [0.980, 0.983]	0.937 $\pm$ 0.025 [0.930, 0.944]
	GBM	0.947 $\pm$ 0.012 [0.938, 0.945]	0.841 $\pm$ 0.036 [0.831, 0.852]	0.923 $\pm$ 0.034 [0.913, 0.933]	0.880 $\pm$ 0.024 [0.873, 0.886]	0.980 $\pm$ 0.006 [0.978, 0.982]	0.935 $\pm$ 0.023 [0.928, 0.941]

Based on the result in Table 9, all classification models combined with both XGB-RFECV and GBM-RFECV wrappers demonstrated quite high and stable performance, with a mean accuracy above 0.940 and a mean ROC-AUC of 0.980–0.982. The best performing model achieved an average accuracy of 0.947, precision of 0.860, and F1-score of 0.892, obtained from the combination of XGB-RFECV and XGB. The evaluation results show that

predicting success during the monitoring process is considered more effective when supporters are actively involved.

Table 10. Friedman test for F1-Score Comparison among Classifiers

Metric	Models Compared	Friedman Statistic	p-value	Significant
F1-Score	GBM, XGBoost, CatBoost (XGB-RFECV and GBM-RFECV)	35.5441	1.17e-06	Significant

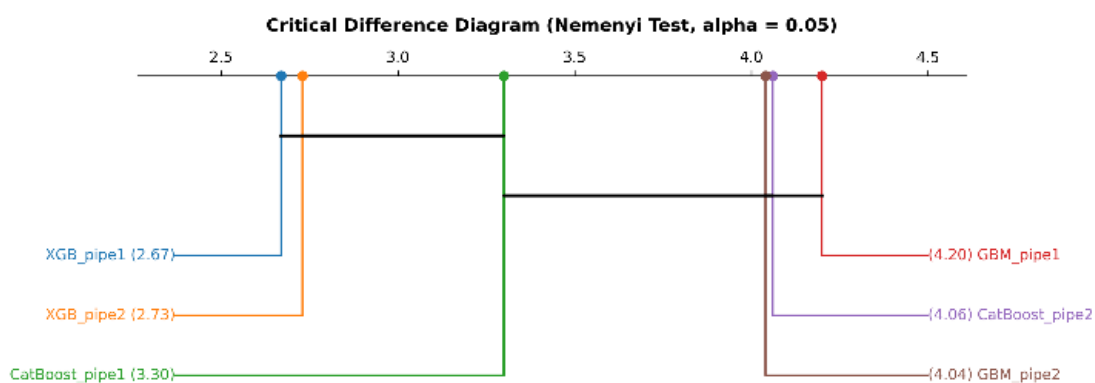


Figure 8. Critical Difference Diagram

To examine pairwise distinctions following the statistically significant Friedman test in table 10, $\chi^2 = 35.5441$, $p = 1.17 \times 10^{-6}$, a post-hoc Nemenyi test $\alpha=0.05$ was performed, as visualized in the Critical Difference diagram in figure 8. Although the Friedman test rejects the null hypothesis of global equivalence across the 50 validation evaluations, the post-hoc analysis indicates that the top-performing configurations specifically XGBoost (ranks 2.67 and 2.73) and CatBoost (rank 3.30) are connected by a critical difference bar, showing no statistically significant performance difference between them. Furthermore, the absolute empirical margin remains narrow across all ensemble variants mean F1-scores between 0.879 and 0.892 in Table 9, indicating that the choice among gradient boosting architectures yields comparable baseline efficacy under the proposed leakage-controlled framework.

3.1.5. Leakage-Control Validation Analysis

As shown in the lower section of Table 6, models trained under the leakage-prone condition (where feature selection and optimization preceded train-test splitting)

exhibited artificially inflated performance metrics, recording positive-class F1-scores up to 0.947 and PR-AUC values reaching 0.985. When identical architectures were evaluated under the strict leakage-controlled protocol, metrics adjusted to realistic levels (F1-score 0.888). This significant performance gap empirically illustrates the magnitude of optimistic evaluation bias introduced when test data contaminates feature selection.

3.2. Discussion

3.2.1. Majority Baseline and Class-Imbalance Benchmarking

In heavily skewed tabular domains such as crowdfunding, relying exclusively on standard classification accuracy creates a deceptive impression of predictive efficacy. The Majority Class Baseline established in this study demonstrates that an uninformed, zero-rule classifier predicting only unsuccessful outcomes can achieve an apparent Accuracy of 76.99%. However, such a baseline provides zero practical utility for platform operators, campaign creators, and prospective backers, as evidenced by an F1-score of 0.000, Recall of 0.000, and a PR-AUC of 0.231 that merely reflects the empirical positive class prior.

Conversely, establishing the unoptimized Machine Learning Baselines (trained on all 29 raw features without feature selection or balancing) set a considerably higher competitive benchmark, where CatBoost, GBM, and XGB achieved positive-class F1-scores of 0.882, 0.859, and 0.874, respectively. Building upon this, the fully integrated proposed framework surpassed both the majority and raw baselines across multiple operational dimensions, securing a peak Accuracy of 0.947, an F1-score of 0.888, and a PR-AUC of 0.923 on the unseen test set. This margin verifies that the ensemble boosting models did not capitalize on majority-class prevalence, but rather developed genuine discriminatory boundaries capable of isolating successful ventures amidst dominant failure distributions.

3.2.2. Hybrid Feature Selection Analysis

The effectiveness of the hybrid feature selection approach demonstrates that Pearson Correlation and Mutual Information can identify relevant features from both linear and non-linear relationships associated with crowdfunding success. Pearson Correlation is effective in detecting features that have a direct linear relationship with the target variable, while Mutual Information is capable of capturing more complex and non-linear dependencies that may not be detected through correlation analysis alone [7], [8]. The

combination of PCC and MI yielded 25 features used in the optimization process. The purpose of this combination is to reduce the risk of losing potentially relevant features; this explanation is supported by research that has already used feature selection methods to obtain more informative features [4], [8].

The feature selection results indicate that `number_of_supporters` consistently emerged as the most informative predictor of crowdfunding success. However, the number of supporters becomes more informative as a campaign progresses. The proposed framework is better suited for campaign monitoring because metrics such as the `number_of_supporters` can reflect public interest, social recognition, and collective trust in the campaign metrics that are only available once the campaign is ongoing. In addition, `updates` and `social_media_followers` were identified as important features, suggesting that communication intensity and online engagement play significant roles in attracting potential backers. Active communication through campaign updates can increase transparency and strengthen the confidence of potential supporters, while a broader social media audience may enhance campaign visibility and information dissemination [1], [3].

Even though `number_of_supporters` was identified as the most influential predictor, its inclusion relies strictly on its availability at prediction time during active campaign monitoring, rather than pre-launch settings. Under this live-monitoring setting, supporter-related dynamic metrics are observable during funding, while train-test isolation is strictly maintained across all preprocessing steps. Furthermore, the objective of the proposed framework is campaign-monitoring prediction rather than pre-launch forecasting. Under this setting, supporter-related information is considered a legitimate operational variable because it reflects campaign engagement observed during the active funding period. Therefore, the strong predictive contribution of `number_of_supporters` should be interpreted as an expected characteristic of campaign-monitoring analytics rather than evidence of information leakage.

Moreover, several content-related features, including `promotion_video`, `video_length`, and `number_of_words_in_content`, were retained during the feature optimization process. The presence of these features suggests that the quality and completeness of campaign information contribute to crowdfunding success. Effective content preparation including

a clear explanation of the project's objectives, needs, and funding outcomes can help bridge the information gap between project creators and potential supporters [1], [3]. Furthermore, the feature importance analysis of the best-performing models, Figure 7 confirms that number_of_supporters was the most influential predictor of crowdfunding success relative importance of 0.507 for XGB-RFECV and 0.450 for GBM-RFECV, followed sequentially by website, promotion_video, and updates.

3.2.3. Comparison of GBM-RFECV and XGB-RFECV

The wrapper feature selection stage demonstrated that both gradient boosting algorithms successfully refined the 25 candidate features into high-performing, compact subsets. Specifically, XGB-RFECV selected 18 features, while GBM-RFECV retained 23 features. The regularization used in XGBoost may have contributed to the smaller feature subset by controlling overfitting and reducing the influence of less informative features [13], [14].

Empirically, the two wrapper approaches exhibited complementary advantages depending on the evaluation protocol. On the single unseen test set Table 6, the GBM-RFECV subset coupled with XGBoost attained the highest individual classification metrics Accuracy 0.947, Recall 0.906, F1-score 0.888. Conversely, under Repeated Stratified Cross-Validation Table 9, the XGB-RFECV and XGB configuration demonstrated the highest mean generalization performance and stability Accuracy 0.947 ± 0.012 , Precision 0.860 ± 0.037 , F1-score 0.892 ± 0.023 .

XGB-RFECV and GBM-RFECV showed only a small difference in performance. This suggests that the feature subsets obtained from both methods provided similar predictive information for crowdfunding success. Therefore, the final performance differences are likely attributable to variations in the learning and optimization mechanisms of the boosting algorithms rather than substantial differences in feature quality. Several boosting algorithms can produce different predictive results even when trained on similar features [24]. Although the Friedman test indicates statistically significant variance across iterations $p < 0.001$, the absolute performance margin among XGBoost, GBM, and CatBoost remains narrow (mean F1-score difference < 0.015 , and post-hoc Nemenyi testing confirms that XGBoost and CatBoost do not differ significantly. This demonstrates that ensemble tree-based architectures exhibit comparable baseline

efficacy under leakage-controlled conditions. Consequently, the primary determinant of robust predictive performance lies in the systematic combination of multi-stage feature optimization and strict training-fold resampling, rather than an intrinsic statistical superiority of a single boosting algorithm

3.2.4. Effect of SMOTE on Imbalanced Crowdfunding Data

The crowdfunding dataset exhibited a severe class distribution imbalance (76.9% unsuccessful vs. 23.1% successful). Such an imbalance biases machine learning algorithms toward the majority class, leading to poor positive-class sensitivity. SMOTE was applied only to the training folds to address this issue by generating synthetic minority samples and helping the classifiers learn from a more balanced class distribution. The empirical results in Table 6 and Table 7 show that incorporating SMOTE substantially elevated positive-class Recall across all classifiers, lifting XGBoost recall from 0.880 in the unoptimized baseline to 0.906 in the proposed configuration, while limiting false negatives to 7 instances. Crucially, applying SMOTE exclusively to the training folds after data partitioning averted synthetic data contamination into the test set, ensuring that evaluation metrics reflect genuine generalization on organic data distributions [17].

3.2.5. Comparative Analysis of Baseline Variants and the Proposed Pipeline

The structured ablation configurations presented in Table 6 systematically justify the multi-stage architecture of the proposed pipeline:

- 1) Dimensionality Reduction without Performance Degradation:
 - a. Comparing the unoptimized baseline (29 features) against Baseline + FS (25 features) and Baseline + FS + RFECV (18 features) reveals that compacting the feature space did not compromise predictive accuracy. The filter-wrapper cascade effectively eliminated noisy and redundant predictors, allowing models like CatBoost and GBM to sustain F1-scores above 0.850 while reducing computational complexity.
- 2) Synergistic Multi-Stage Integration:
 - a. While individual components (FS alone, RFECV alone, or SMOTE alone) provided localized metric gains, only their full, sequential pipeline specifically Hybrid PCC+MI followed by RFECV wrapper optimization and post-split SMOTE achieved the most robust balance across Precision (0.871), Recall (0.906), and F1-score (0.888) under XGBoost.

3) Dominance of Dynamic Monitoring Signals (Feature Ablation):

The ablation experiment intentionally excluding `number_of_supporters` induced the sharpest performance drop observed across all models, with F1-scores plummeting from 0.888 to 0.721 and PR-AUC declining from 0.923 to 0.728. This step degradation confirms the empirical finding in Figure 7 that backer traction carries the primary explanatory weight in forecasting campaign completion. Methodologically, this outcome firmly establishes that the proposed architecture operates as a live-monitoring decision system, and cannot be utilized as a pure pre-launch screener without substituting dynamic signals with pre-launch social capital indicators.

3.2.6. Implications of the Proposed Leakage-controlled Framework

The comparative evaluation between the leakage-controlled and leakage-prone workflows underscores the critical role of data isolation protocols. When feature selection and oversampling were applied globally prior to train-test splitting (leakage-prone condition), metrics displayed severe artificial inflation, reaching F1-scores up to 0.947 and PR-AUC values of 0.985. Under the leakage-controlled pipeline, metrics settled at realistic, generalizable levels (F1-score: 0.888; PR-AUC: 0.923).

This divergence illustrates that data leakage introduces an optimistic evaluation bias rather than a genuine improvement in predictive utility. Enforcing strict train-test separation prior to feature filtering and SMOTE generation guarantees methodological validity, ensuring that deployment-stage performance aligns with experimental benchmarks. In an operational crowdfunding context, this model serves as an in-flight monitoring tool: by utilizing incoming engagement data (`number_of_supporters`, `updates`, `comments`) during the funding window, platforms and creators can actively track trajectory and deploy corrective marketing interventions before campaign termination.

3.2.7. Limitations of the Research

Behind these impressive accuracy results, there are still some fundamental challenges that should not be overlooked:

- 1) Operational Timeframe: Because dynamic metrics like `number_of_supporters` and `updates` accumulate during active fundraising, the framework cannot

- serve as a purely pre-launch screening tool. Future work should investigate multi-stage, time-decay models evaluated at explicit post-launch intervals.
- 2) **Platform Specificity:** The dataset is derived from a single Turkish reward-based crowdfunding platform Fangogo. Variations in investor culture, economic conditions, and platform rules suggest the need for validation across multiple platforms and market environments.
 - 3) **Synthetic Resampling Dynamics:** While SMOTE effectively mitigated class imbalance, linear interpolation in feature space may generate borderline samples that do not fully capture authentic entrepreneurial dynamics [17]. Cost-sensitive loss functions or generative models represent viable alternatives.
 - 4) **Local Model Explainability** While RFECV and tree impurity metrics established global feature contributions, future iterations should incorporate instance-level explainability frameworks such as SHAP based to provide personalized diagnostic feedback for campaign creators.

4. CONCLUSION

This study proposed a leakage-controlled crowdfunding success prediction framework that integrates Hybrid Feature Selection, RFECV-based feature optimization, SMOTE, and boosting-based classification models to improve prediction performance on imbalanced crowdfunding data. The experimental results based on the hold-out test evaluation and repeated cross-validation showed that the proposed leakage-controlled framework specifically integrating Hybrid Feature Selection PCC and MI, GBM-RFECV, SMOTE, and XGBoost achieved the most balanced performance with an Accuracy of 0.947, Precision of 0.871, Recall of 0.906, F1-score of 0.888, ROC-AUC of 0.975, and PR-AUC of 0.923. Hybrid feature selection and SMOTE were found to be effective at retaining useful features, and the model was able to distinguish well between successful and unsuccessful campaign classes. In addition, separating the training and test data before feature selection also contributed to a more reliable evaluation in this analysis. Although temporal leakage was controlled through the campaign-monitoring prediction setting, the proposed framework may not be directly applicable to early-stage or pre-launch crowdfunding prediction scenarios. The findings are based on a single Turkish crowdfunding dataset, which may limit their applicability to other platforms and geographical contexts. Several influential

predictors, such as number_of_supporters and updates, represent dynamic campaign-progress variables that reflect active engagement rather than static pre-launch attributes, reinforcing that the model is designed specifically for ongoing campaign monitoring rather than early screening. Future studies could examine time-aware prediction frameworks that assess model performance at different stages of the crowdfunding lifecycle. Overall, the proposed framework demonstrates promising predictive performance under the experimental setting. However, further external and deployment-stage validation is required before its suitability for operational decision-support applications can be established.

REFERENCES

- [1] S. Hohen, C. Hüning, and L. Schweizer, *Reward - based crowdfunding – a systematic literature review*. Springer International Publishing, 2025. doi: 10.1007/s11301-025-00544-2.
- [2] A. C. Ionica, S. Cseminski, and M. Leba, "From Concept to Market: Ensemble Predictive Model for Research Project Crowdfunding Readiness," *Systems*, vol. 12, no. 12, p. 535, 2024, doi: 10.3390/systems12120535.
- [3] A. Gregoriades and C. Themistocleous, "Improving Crowdfunding Decisions Using Explainable Artificial Intelligence," *Sustainability*, vol. 17, no. 4, pp. 1–19, 2025, doi: 10.3390/su17041361.
- [4] M. Kilinc and C. Aydin, "Electronic Commerce Research and Applications Feature selection for Turkish Crowdfunding projects with using filtering and wrapping methods," *Electron. Commer. Res. Appl.*, vol. 62, no. October, p. 101340, 2023, doi: 10.1016/j.eierap.2023.101340.
- [5] H. Avvad, "Crowdfunding Success Prediction Using Machine Learning: A Comparative Study Based on Turkiye's Campaigns," *Black Sea J. Eng. Sci.*, vol. 9, no. 2, pp. 646–663, 2026, doi: 10.34248/bsengineering.1772673.
- [6] S. Kapoor and A. Narayanan, "Article Leakage and the reproducibility crisis in machine-learning-based science Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, 2023, doi: 10.1016/j.patter.2023.100804.

- [7] N. Papaioannou, G. Myllis, A. Tsimpiris, and V. Vrana, "The Role of Mutual Information Estimator Choice in Feature Selection : An Empirical Study on mRMR," *information*, vol. 16, no. 9, pp. 1–25, 2025, doi: 10.3390/info16090724.
- [8] S. Liu and M. Motani, "Improving Mutual Information Based Feature Selection by Boosting Unique Relevance," *J. Artif. Intell. Res.*, vol. 82, pp. 1267–1292, 2025, [Online]. Available: 10.1613/jair.117219
- [9] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1–3, pp. 389–422, 2002, doi: 10.1023/A:1012487302797.
- [10] L. Gan, "XGBoost-Based E-Commerce Customer Loss Prediction," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–10, 2022, doi: 10.1155/2022/1858300.
- [11] Z. Fang, S. Yang, C. Lv, S. An, and W. Wu, "Application of a data- - driven XGBoost model for the prediction of COVID- - in the USA : a time- - series study," *BMJ Open*, vol. 12, p. e056685, 2022, doi: 10.1136/bmjopen-2021-056685.
- [12] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13–17-Aug, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.
- [13] Z. Rafie, M. S. Talab, B. E. Z. Koor, A. Garavand, C. Salehnasab, and M. Ghaderzadeh, "Leveraging XGBoost and explainable AI for accurate prediction of type 2 diabetes," *BMC Public Health*, vol. 25, p. 3688, 2025, doi: 10.1186/s12889-025-24953-w.
- [14] H. R. Imani, Mehdi; Beikmohammadi, Ali; Arabnia, "Comprehensive Analysis of Random Forest and XGBoost Performance with SMOTE , ADASYN , and GNUS Under Varying Imbalance Levels," *Technologies*, vol. 13, no. 3, p. 88, 2025, doi: 10.3390/technologies13030088.
- [15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
- [16] Y. Yang, H. A. Khorshidi, and U. Aickelin, "A review on over-sampling techniques in classification of," *Front. Digit. Heal.*, vol. 6, 2024, doi: 10.3389/fdgth.2024.1430245.
- [17] I. M. Alkhawaldeh, I. Albalkhi, and A. J. Naswhan, "Challenges and limitations of synthetic minority oversampling techniques in machine learning," *World J. Methodol.*, vol. 13, no. 5, pp. 373–378, 2023, doi: 10.5662/wjm.v13.i5.373.
- [18] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.

- [19] B. Dong, S. Ma, Z. Gao, and A. Qin, "CatBoost Improves Inversion Accuracy of Plant Water Status in Winter Wheat Using Ratio Vegetation Index," *Appl. Sci.*, vol. 15, no. 21, p. 11363, 2025, doi: 10.3390/app152111363.
- [20] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," 2018, [Online]. Available: <http://arxiv.org/abs/1810.11363>
- [21] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.
- [22] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," *ACM Int. Conf. Proceeding Ser.*, vol. 148, pp. 233–240, 2006, doi: 10.1145/1143844.1143874.
- [23] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS One*, vol. 10, no. 3, 2015, doi: 10.1371/journal.pone.0118432.
- [24] W. Wolak, A. Plichta, and H. Orlicki, "Interpretable ensemble learning for tumor-type prediction with a SHAP-based evaluation of CatBoost and voting classifiers," *Sci. Rep.*, vol. 16, p. 1401, 2025, doi: 10.1038/s41598-025-31079-x.