Improved Banking Customer Retention Prediction Based on Advanced Machine Learning Models

L W Widianti[1], A S B Karno[2], W Hastomo*[3], A N Utomo[4], D Arif[5], I S K Wardhana[6], D Strydom[7]

[1]Department of Information System; STMIK Jakarta STI&K, Jakarta, Indonesia
[2,5]Department of Information System, Faculty of Engineering, Gunadarma University, Depok, Indonesia
[3]Department of Information Technology, Ahmad Dahlan Institute of Technology and Business, Indonesia
[4]Department of Information Technology, Institut Sains dan Teknologi Nasional, Jakarta, Indonesia
[6]Department of Information System Faculty of Engineering and Computer Science, Indraprasta University PGRI
[7]Weskaap Motors corporate, 87 Main Street, Western Cape, Vredenburg 7380, South Africa

E-mail: lindawewe100@gmail.com[1], adh1t10.2@gmail.com[2], Widie.has@gmail.com[3], aryo.nurutomo@gmail.com[4], dodiarif8@gmail.com[5], indraskw@gmail.com[6], deon@wkm.co.za[7]

Abstract. The quick growth of the banking sector is reflected in the rise in the number of banks. In addition to the intense competition among banks for new customers, efforts to keep existing ones are essential to minimizing potential losses for the company. To ascertain whether customers will leave the bank or remain customers, this study will employ churn forecasts. A 1,750,036-customer demographic dataset, which includes data on bank customers who have left or are still customers, is used in the training process to compare five machine learning technology models in order to investigate the improvement of binary classification prediction accuracy. These models are Decision Tree, Random Forest, Gradient Boost, Cat Boost, and Light Gradient Boosting Machine (LGBM). According to the study's results, LGBM performs better than the other four models since it has the highest recall and accuracy and the fewest False Negatives. The LGBM model's corresponding accuracy, precision, recall, f1 score, and AUC are 0.8789, 0.8978, 0.8553, 0.8758, and 0.9694. This demonstrates that, in comparison to traditional methods, machine learning optimization can produce notable advantages in churn risk classification. This study offers compelling proof that sophisticated machine learning modeling can revolutionize banking industry client retention management.

Keywords: Customer Loyalty Forecasting; Churn Prediction; Machine Learning; Financial Customer Analytics

1. Introduction

Customer retention is crucial for the banking industry's profitability and growth, as retaining clients directly impacts long-term revenue stability and reduces operational costs [1]. However, banks confront difficulties in maintaining clients due to fierce competition and rising customer turnover rates [2]. Recent research indicates that more than 20% of bank clients switch institutions annually, driven by factors such as dissatisfaction with service quality and competitive offers [3]. With the cost of acquiring new customers estimated to be five times higher than retaining existing ones [4], banks have prioritized client retention as a strategic focus. Advanced machine learning analytics and models enable banks to accurately estimate client churn risk and design personalized retention strategies.

To further improve model performance, techniques such as tree pruning (to reduce overfitting), ensemble learning (to combine weak classifiers), and hyperparameter tuning [5] (to optimize algorithmic parameters) have proven effective in enhancing predictive accuracy [6]. Recent studies by [7] highlight that optimized machine learning models, such as gradient-boosted trees and deep neural networks, outperform classical logistic regression by 15–20% in churn prediction tasks. Previous research utilized machine learning models, including Neural Networks, Random Forest, and Gradient Boosting, to predict customer churn in financial institutions [8]. This performance gap underscores the potential of advanced algorithms to transform client retention strategies in banking [9].

This study demonstrates how optimized machine learning classification models can enhance client retention prediction. Five machine learning models were evaluated, each refined through techniques like SMOTE for class imbalance mitigation and Bayesian optimization for hyperparameter tuning [10]. Performance was assessed using metrics such as AUC-ROC (to measure class separation) [11], precision-recall curves [12] (to evaluate trade-offs in imbalanced data) [13], and F1-score (to balance precision and recall) [14].

2. Method

In terms of scope, this research method is divided into three parts, the first of which is to collect data so that it may be used in subsequent treatment processes. The first step is to load the dataset, followed by data wrangling, EDA, encoding, dataset preparation, feature engineering, scaling, and dataset splitting [15]. The second step is to create five models and train a dataset, and the third step is to evaluate the results of each model's training (Figure 1).

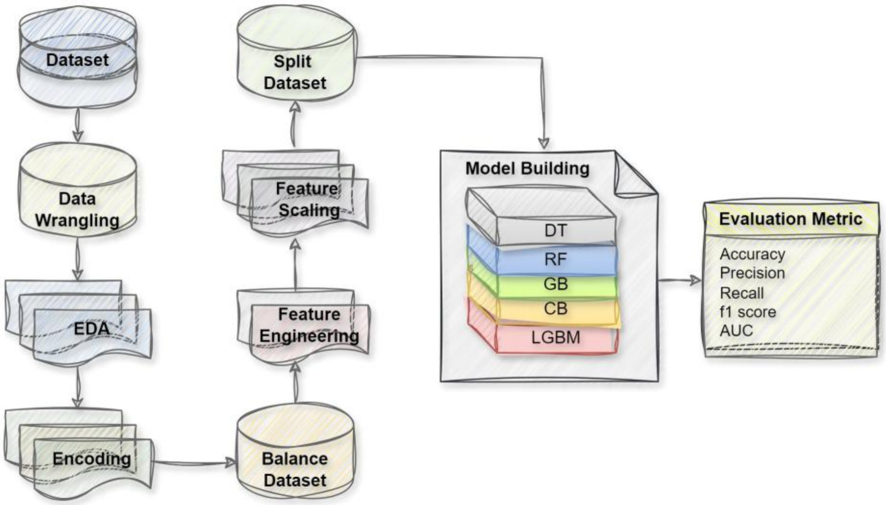


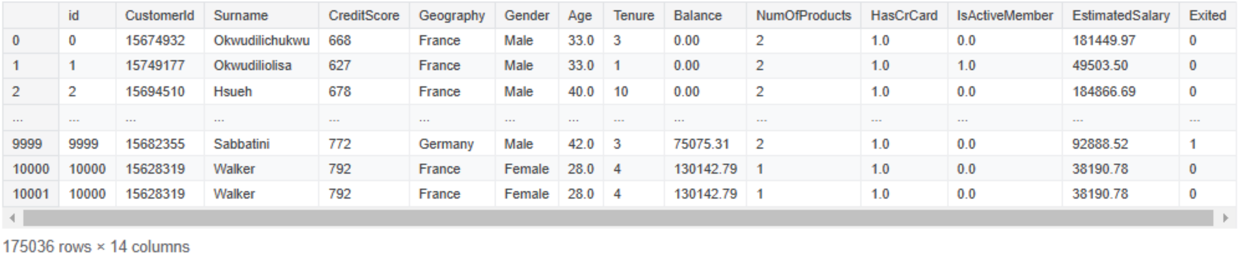
Figure 1. Research flow

2.1 Preparing Dataset

Loading Dataset

The dataset was collected by downloading a CSV file from www.kaggle.com [6]. This raw data file has 1,750,036 rows and 14 columns of data (Figure 2). Each column provides client information with the following attributes:

- Customer ID: A special number that only each client has
- Surname: Last name or surname of the client
- Credit Score: A digit that indicates a customer's credit score
- Geography: The nation in which the client resides
- Gender: The gender of the client
- Age: The age of the client.
- Tenure: The length of time a client has been a bank customer
- Balance: The total amount owed by the client
- NumOfProducts: The quantity of bank products (credit card, savings account, etc.) that the client utilizes.
- HasCrCard: Indicates if the client possesses a credit card.
- IsActiveMember: If the client is a current member
- EstimatedSalary: The customer's approximate salary
- Exited: Target Variable: Has the consumer churned?



	id	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	0	15674932	Okwudilichukwu	668	France	Male	33.0	3	0.00	2	1.0	0.0	181449.97	0
1	1	15749177	Okwudiliolisa	627	France	Male	33.0	1	0.00	2	1.0	1.0	49503.50	0
2	2	15694510	Hsueh	678	France	Male	40.0	10	0.00	2	1.0	0.0	184866.69	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9999	9999	15682355	Sabbatini	772	Germany	Male	42.0	3	75075.31	2	1.0	0.0	92888.52	1
10000	10000	15628319	Walker	792	France	Female	28.0	4	130142.79	1	1.0	0.0	38190.78	0
10001	10000	15628319	Walker	792	France	Female	28.0	4	130142.79	1	1.0	0.0	38190.78	0

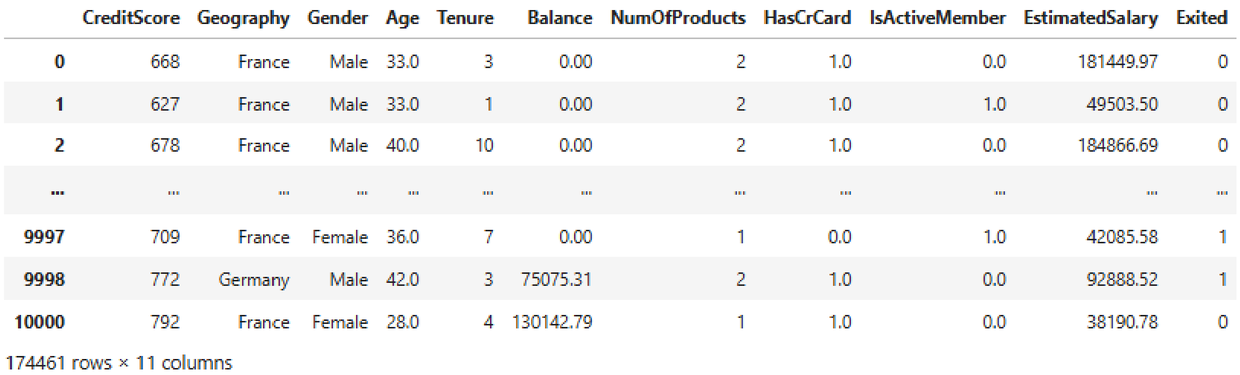
175036 rows × 14 columns

Figure 2. Dataset files

2.2 Data Wrangling

At this point, the raw data will be turned into the information required for future processing (Figure 3 is the result of this stage). This study involves several steps, including:

- Remove extraneous information, particularly columns with the characteristics id, CustomerId, and Surname.
- View and aggregate comparable data types from the current dataset, namely 9 columns with numeric data types (CreditScore, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, Exited) and 2 columns with categorical data types (Geography and Gender).
- Remove duplicate and null data.



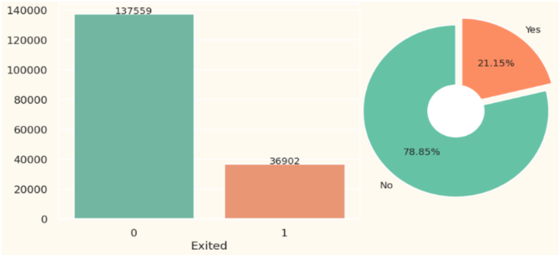
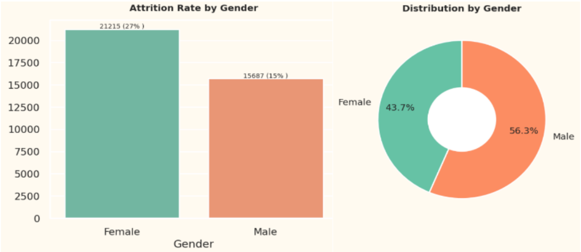
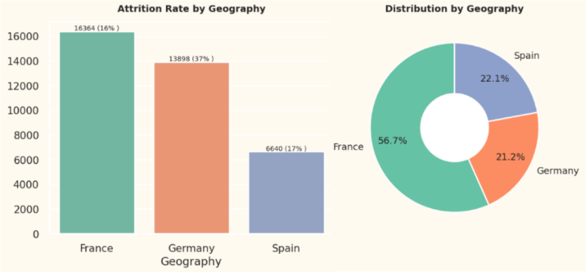
	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	668	France	Male	33.0	3	0.00	2	1.0	0.0	181449.97	0
1	627	France	Male	33.0	1	0.00	2	1.0	1.0	49503.50	0
2	678	France	Male	40.0	10	0.00	2	1.0	0.0	184866.69	0
...	...	...	...	...	...	...	...	...	...	...	...
9997	709	France	Female	36.0	7	0.00	1	0.0	1.0	42085.58	1
9998	772	Germany	Male	42.0	3	75075.31	2	1.0	0.0	92888.52	1
10000	792	France	Female	28.0	4	130142.79	1	1.0	0.0	38190.78	0

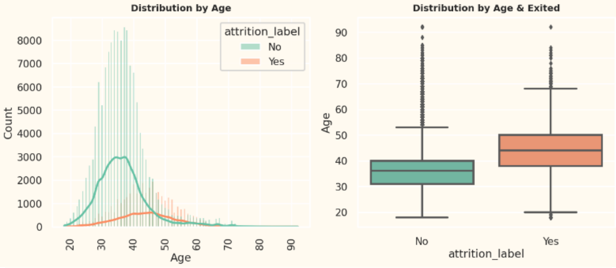
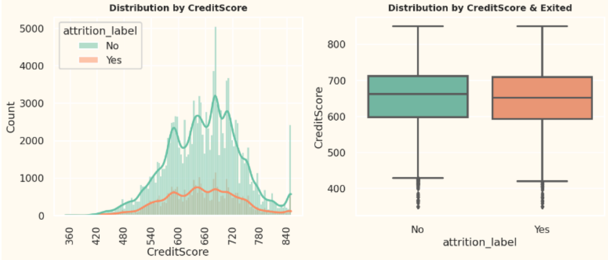
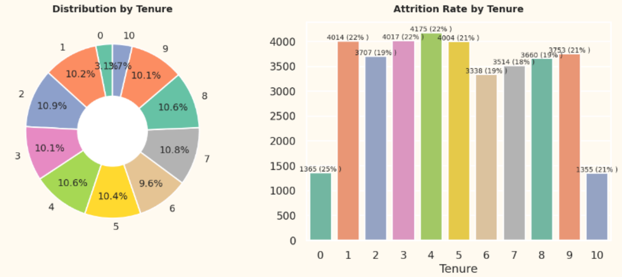
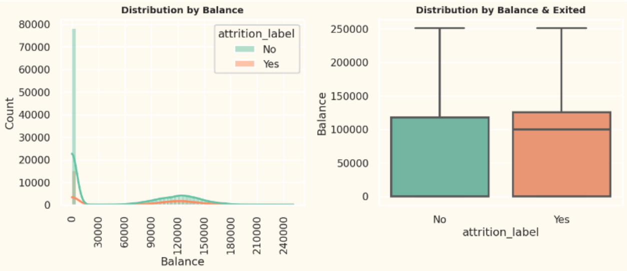
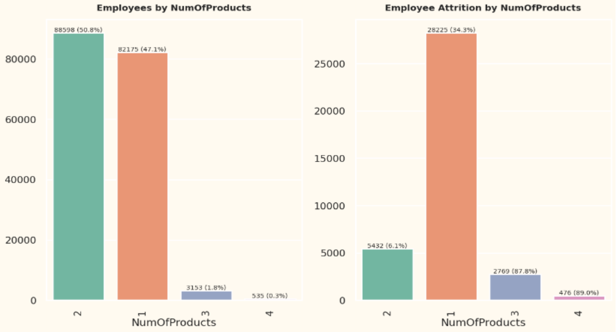
174461 rows × 11 columns

Figure 3. The result of data wrangling

2.3 Exploratory Data Analysis (EDA)

To avoid the problem of excessive page use, the 11 results of the investigation process are collected in one table containing charts and conclusions of information obtained from each chart (Figure 4).

	Inference	EDA
1	Average visual customer attrition is 21.15%, and the data appears imbalanced.	
2	An analysis of customer attrition by gender of 43.7% women and 56.3% men reveals that more women (27%) quit the bank than males (15%).	
3	Analysis of customer attrition by location reveals that France has the most customers (56.7%) but the lowest attrition rates (16%). Germany has the greatest attrition rates (37%), although having just 21.2% of all consumers.	

	Inference	EDA
4	Customers' ages range from 30 to 40. It is apparent that the decreasing tendency is rising with age. Customers reduction in working customers	
5	Analysis of customers based on credit scores, there are more customers with credit scores between 600 and 700 and there is not much information here about customer attrition	
6	Analysis of customer distribution based on tenure, the average customer reduction is almost the same in each tenure category, namely between 19 and 22%, except for tenure 0 where customer reduction reaches 25%.	
7	Analysis of customer distribution based on balance, there is not much information about customer reduction in this data distribution.	
8	Analysis of customer attrition based on number of products • 50.8% Customers have 2 no of products with 6.1% Attrition rate • 47.1% Customers have 1 no of products with 34.3% Attrition rate • 1.8% Customers have 3 no of products with 88.8% attrition rate (High Attrition rate) • 0.3% Customers have 4 no of products with 89% attrition rate (High Attrition rate)	

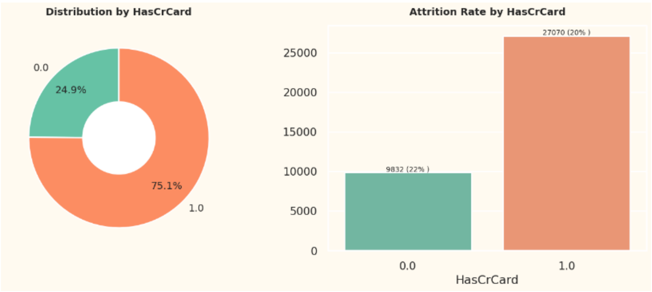
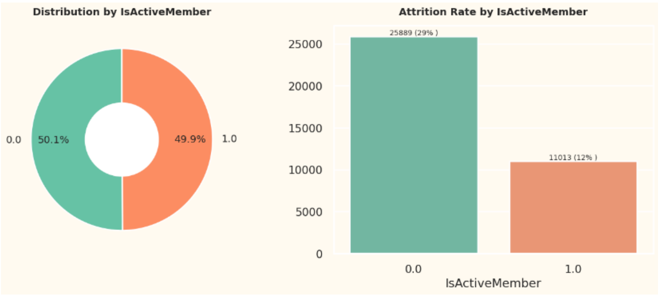
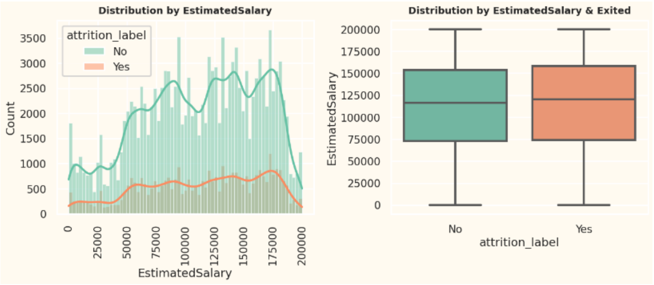
	Inference	EDA
9	Analyzing Employee Attrition by HasCrCard • 50.8% Customers have 2 no of products with 6.1% Attrition rate • 75% of customers have Credit Card • 25% of customers don't have credit cards • both classes have almost the same Attrition rate i.e. 20-22 % • No meaningfull information for attrition is seen here	
10	Analyzing Employee Attrition by IsActiveMember • 50.1% Customers are Not Active members with attrition rate 29% • 49.9% are active members with attrition rate 12% • Not Active members are most likely to be Exited	
11	Histogram on the left chart • Workers who earn more money typically stay on the job longer. • The percentage of departing employees is lower than the percentage of remaining employees. Chart on the right (box plot) • There is little difference in the median salary between departing and remaining employees. • There is no evidence that income significantly influences employee decisions to leave, and the interquartile range for salary fluctuation is very similar for both groups.	

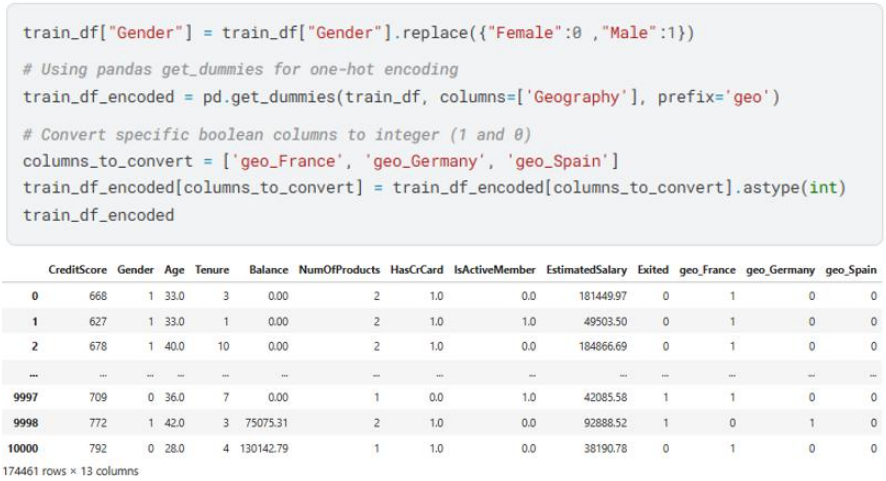
Figure 4. Results of the EDA process in the form of charts and inference

2.4 Encoding

Encoding is the process of converting categorical input into a numerical representation appropriate for machine learning models [17]. Encoding is required to represent categorical data in the form of integers with specific meanings based on the category [18]. The precise encoding depends on the data type and technique utilized. Figure 5 shows the program code and output for this encoding stage, namely:

a. Converting "Gender" to numeric (0 for Female, 1 for Male).
b. Using pd.get_dummies() to one-hot encode the "Geography" column. One-hot encoding will create three new columns:
   - geo_France → 1 if country = "France", 0 otherwise.
   - geo_Germany → 1 if country = "Germany", 0 otherwise.
   - geo_Spain → 1 if country = "Spain", 0 otherwise.

c. Making sure that the one-hot encoded columns (geo_France, geo_Germany, geo_Spain) have integer data types (0 and 1).
d. astype(int) is used because the result of pd.get_dummies() is sometimes of the bool (True/False) data type, and we need to ensure that the value is a number (1 or 0).



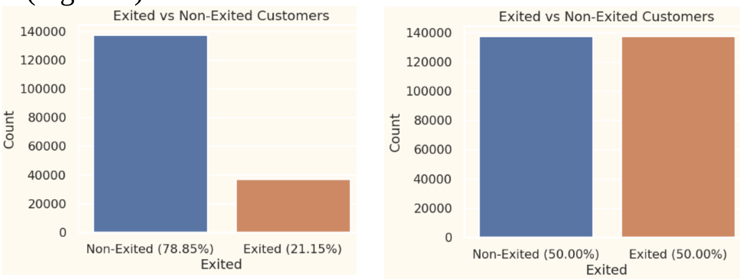
```
train_df["Gender"] = train_df["Gender"].replace({"Female":0 ,"Male":1})

# Using pandas get_dummies for one-hot encoding
train_df_encoded = pd.get_dummies(train_df, columns=['Geography'], prefix='geo')

# Convert specific boolean columns to integer (1 and 0)
columns_to_convert = ['geo_France', 'geo_Germany', 'geo_Spain']
train_df_encoded[columns_to_convert] = train_df_encoded[columns_to_convert].astype(int)
train_df_encoded
```

```
       CreditScore Gender  Age Tenure    Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary Exited geo_France geo_Germany geo_Spain
    0          668      1 33.0      3       0.00             2       1.0            0.0       181449.97      0          1           0         0
    1          627      1 33.0      1       0.00             2       1.0            1.0        49503.50      0          1           0         0
    2          678      1 40.0     10       0.00             2       1.0            0.0       184866.69      0          1           0         0
  ...          ...    ...  ...    ...        ...           ...       ...            ...             ...    ...        ...         ...       ...
 9997          709      0 36.0      7       0.00             1       0.0            1.0        42085.58      1          1           0         0
 9998          772      1 42.0      3   75075.31             2       1.0            0.0        92888.52      1          0           1         0
10000          792      0 28.0      4  130142.79             1       1.0            0.0        38190.78      0          1           0         0
174461 rows × 13 columns
```

Figure 5. Program code and results of the encoding process.

2.5 Balance dataset

In many categorization scenarios, the minority data is significantly smaller than that of the majority. This causes the model to forecast the majority class more accurately than the minority class. SMOTE (Synthetic Minority Oversampling Approach) was utilized in this work to correct for data imbalances. SMOTE is an oversampling approach designed to balance classes in an imbalanced dataset. SMOTE operates by aggregating fresh data about minority groups depending on their closest neighbours. This will boost the amount of minority data while also balancing the class ratio. SMOTE is useful in improving model performance on unbalanced datasets [19]. Data that was initially unbalanced after the SMOTE process produced balanced data (Figure 6).



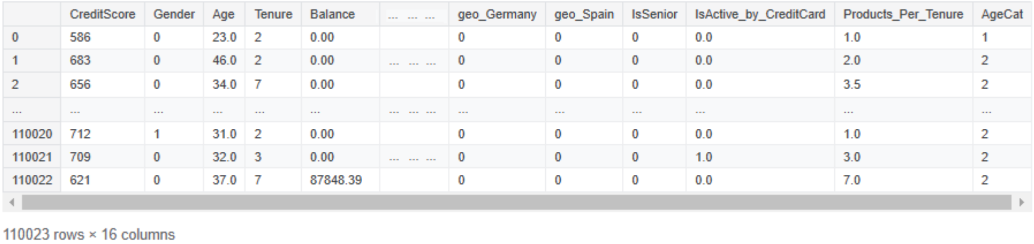
(a) Before SMOTE (b) After SMOTE

Figure 6. Target data (a) before and (b) after using SMOTE

2.6 Feature engineering

Feature engineering is the act of transforming raw data into more useful and relevant features that might boost the performance of machine learning models [20]. The primary purpose of feature engineering is to improve the representation of data so that relevant patterns and information can be recovered by modeling techniques. Feature engineering approaches involve choosing relevant features, transforming them, developing derived features, aggregating and rearranging data [21]. Feature engineering is an essential component of the machine learning process. Machine learning models will be inaccurate if they lack useful characteristics, even if powerful algorithms are used. To design the appropriate features, feature engineers must have domain expertise. In this study, feature engineering was carried out by adding new features (Figure 7 is the result of this stage), namely:

- "IsSenior" attribute for customers over 60 years old
- "IsActive_CreditCard" attribute for customers with active credit cards
- "Prod_Tenure" attribute for customers with "tenure" criteria divided by "NumProduct"
- "Age_Cat" attribute to categorize customers, namely by rounding the results of dividing age by 20



	CreditScore	Gender	Age	Tenure	Balance	... ...	geo_Germany	geo_Spain	IsSenior	IsActive_by_CreditCard	Products_Per_Tenure	AgeCat
0	586	0	23.0	2	0.00		0	0	0	0.0	1.0	1
1	683	0	46.0	2	0.00	... ...	0	0	0	0.0	2.0	2
2	656	0	34.0	7	0.00		0	0	0	0.0	3.5	2
...	...	...	...	...	...	... ...	...	...	...	...	...	...
110020	712	1	31.0	2	0.00		0	0	0	0.0	1.0	2
110021	709	0	32.0	3	0.00	... ...	0	0	0	1.0	3.0	2
110022	621	0	37.0	7	87848.39		0	0	0	0.0	7.0	2

110023 rows × 16 columns

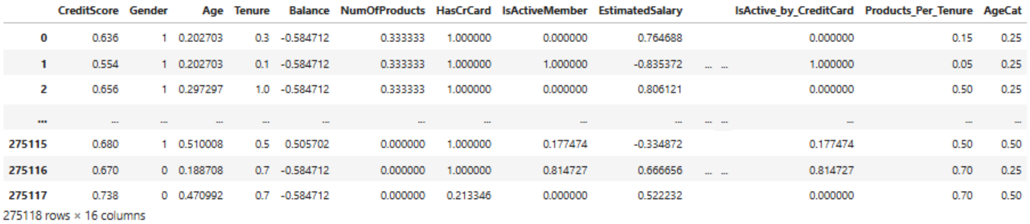
Figure 7. Results of the feature engineering stage

2.7 Feature Scaling (normalization)

Feature scaling is the process of normalizing data characteristics so that their values correspond to the same scale [22]. The primary purpose of feature scaling is to normalize the range of data feature values such that all features are similar and contribute equally to model outcomes. This is significant because if features have highly diverse value ranges, those with big values will dominate the distance or error function optimized by the machine learning algorithm [23]. In general, feature scaling is strongly advised to boost the performance and stability of various machine learning models [24].

Normalization and standardization of numeric features on the dataset performed at this stage using MinMaxScaler and RobustScaler from the sklearn library (Figure 8 is the result of this stage), namely:

- MinMaxScaler is used for features with normal distribution → (CreditScore, Age, Tenure, NumOfProducts, Products_Per_Tenure, AgeCat).
- RobustScaler is used for features with potential outliers → (Balance, EstimatedSalary).
- fit_transform() is used only on training data, while test data only uses transform() so that there is no "data leakage".



	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary		IsActive_by_CreditCard	Products_Per_Tenure	AgeCat
0	0.636	1	0.202703	0.3	-0.584712	0.333333	1.000000	0.000000	0.764688		0.000000	0.15	0.25
1	0.554	1	0.202703	0.1	-0.584712	0.333333	1.000000	1.000000	-0.835372	... ...	1.000000	0.05	0.25
2	0.656	1	0.297297	1.0	-0.584712	0.333333	1.000000	0.000000	0.806121		0.000000	0.50	0.25
...	...	...	...	...	...	...	...	...	...	... ...	...	...	...
275115	0.680	1	0.510008	0.5	0.505702	0.000000	1.000000	0.177474	-0.334872		0.177474	0.50	0.50
275116	0.670	0	0.188708	0.7	-0.584712	0.000000	1.000000	0.814727	0.666656	... ...	0.814727	0.70	0.25
275117	0.738	0	0.470992	0.7	-0.584712	0.000000	0.213346	0.000000	0.522232		0.000000	0.70	0.50

275118 rows × 16 columns

Figure 8. Results of the normalization stage

2.8 Decision Tree (DT)

Decision trees are a supervised learning approach widely used in classification and regression tasks due to their interpretability and simplicity [15]. These models create a prediction framework by learning decision rules based on data attributes, effectively splitting the dataset into smaller subsets while incrementally building a tree-like structure. The resulting tree consists of decision nodes, which test feature values and branch into potential outcomes, and leaf nodes, which represent final predictions for the target variable [25]. This hierarchical structure allows decision trees to handle both categorical and numerical data, making them versatile for various predictive modeling tasks.

Despite their advantages, decision trees are prone to overfitting, especially when the tree depth is not constrained or when the dataset contains noise [26]. Small changes in the training data can lead to significantly different tree structures, resulting in model instability. To address these limitations, ensemble methods such as random forests and gradient boosting have been developed. Random forests combine multiple decision trees to reduce variance and improve generalization, while gradient boosting sequentially builds trees to correct errors from previous iterations [27]. These ensemble techniques have been shown to outperform standalone decision trees in terms of accuracy and robustness.

The performance of decision trees is highly dependent on key parameters such as maximum depth, minimum samples per leaf, splitting criteria (e.g., Gini impurity or information gain), and pruning methods [28]. Proper tuning of these parameters is essential to balance model complexity and predictive performance. Recent studies have demonstrated that decision trees, when optimized, remain competitive for classification and regression tasks, particularly in domains requiring interpretability, such as healthcare and finance [29]. Overall, decision trees are a foundational tool in machine learning, often serving as building blocks for more advanced ensemble models.

2.9 Random Forest (RF)

Random forest is a supervised learning method widely used for classification and regression tasks due to its robustness and high accuracy [30]. It employs ensemble learning, a technique that constructs multiple decision trees independently and aggregates their predictions to improve overall performance. Each tree in a random forest is trained on a bootstrap sample of the original dataset, ensuring diversity among the trees. At each split node, only a small subset of features is randomly selected for splitting, which introduces variability and reduces the risk of overfitting [28]. This approach not only enhances generalization but also makes random forests highly effective for datasets with high dimensionality and mixed data types (categorical and numerical).

One of the key advantages of random forests is their ability to handle datasets with multiple variables and categories while maintaining high predictive accuracy [31]. Additionally, random forests can estimate the importance of features, providing insights into the relationships between predictors and the target variable [32]. This feature importance metric is particularly useful in domains such as bioinformatics and finance, where interpretability is crucial. However, the performance of random forests depends on several hyperparameters, including the number of trees, the number of features considered at each split, and the maximum depth of the trees [33]. Proper tuning of these parameters is essential to balance model complexity and computational efficiency.

Recent studies have demonstrated that random forests outperform standalone decision trees and other ensemble methods in various applications, including medical diagnosis and customer churn prediction [15]. Their ability to handle noisy data, resist overfitting, and provide interpretable results makes them a popular choice for predictive modeling tasks.

2.10 Gradient Boosting (GB)

Gradient boosting is an ensemble learning strategy that involves creating a model in stages to reduce the loss function. Gradient boosting transforms a poor learner into a strong learner by requiring each new model to fix faults in the prior model. Gradient boosting calculates the residual error gradient at each stage to decide the direction of the next improvement. The residual error is then minimized using a new model. The benefits of gradient boosting include the capacity to handle data with large cardinality features, minimize overfitting, and achieve outstanding prediction accuracy [34]. The most common gradient boosting algorithms include XGBoost, LightGBM, CatBoost, and others [35].The main parameters in gradient boosting include the number of estimators, learning rate, max depth, subsample, regularization, etc. [36]. Overall gradient boosting is very powerful and is often used for regression and classification problems.

2.11 CatBoost (CB)

CatBoost (Categorical Boosting) is a gradient boosting method that is optimized for categorical data. CatBoost uses categorial encoding to better precisely represent interactions between category features [35]. CatBoost's key benefit is that it can handle data with a large number of categorical columns without the need for preprocessing. CatBoost is also quicker than other gradient-boosting algorithms [37]. CatBoost uses symmetric tree growth and ordered boosting methods to improve efficiency. It helps to minimize computing time by leveraging categorical data structures [29]. CatBoost's key parameters are depth, learning rate, l2 leaf reg, one hot max size, and specific hyperparameters for categorical data [34]. Overall, CatBoost is ideal for predictive modeling tasks with several category variables.

2.12 Light Gradient Boosting Machine (LGBM)

LGBM is a gradient boosting solution designed for speed and efficiency. LGBM is developed using a leaf-wise decision tree technique that separates data vertically rather than horizontally [39]. The key benefit of LGBM is that it trains quicker than other gradient boosting techniques. LGBM is also better appropriate for data with categorical categories and may achieve high accuracy [40]. LGBM's important features are Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which assist decrease overfitting [34]. LGBM Classifier is ideal for classification problems. The LGBM Classifier's important parameters include num_leaves, max_depth, learning_rate, boosting type, and so on [19]. Overall, the LGBM Classifier is a strong and efficient large data classifier.

2.13 Prediction Model

In this section, the stratified K-Fold Cross-Validation technique is used to assess the performance of machine learning models. Model (the machine learning model to be assessed), x (dataset features), y (target labels), n_splits (the number of folds for cross-validation, defaulting to 5), and random_state (to ensure repeatable results) are the four primary arguments that this function takes in.

- Cross-validation using stratified K-fold.
  This function divides the dataset into n_splits folds using StratifiedKFold from the scikit-learn module. Particularly crucial for unbalanced datasets, stratification guarantees that each fold contains the same percentage of target classes as the original dataset. Before dividing the data into folds, the shuffle=True argument is used to shuffle it, and random_state makes sure that the shuffle's outcomes are repeatable.
- Data separation and training of models
  The data is divided into training data (x_train, y_train) and testing data (x_test, y_test) at each loop iteration. The training data (model.fit(x_train, y_train)) is then used to train the model, which is subsequently used to predict labels on the training and testing data (x_train_pred and x_test_pred). Additionally, predict_proba is used to compute the projected probability for the positive class (y_test_prob), which is helpful for determining metrics like the ROC-AUC score.

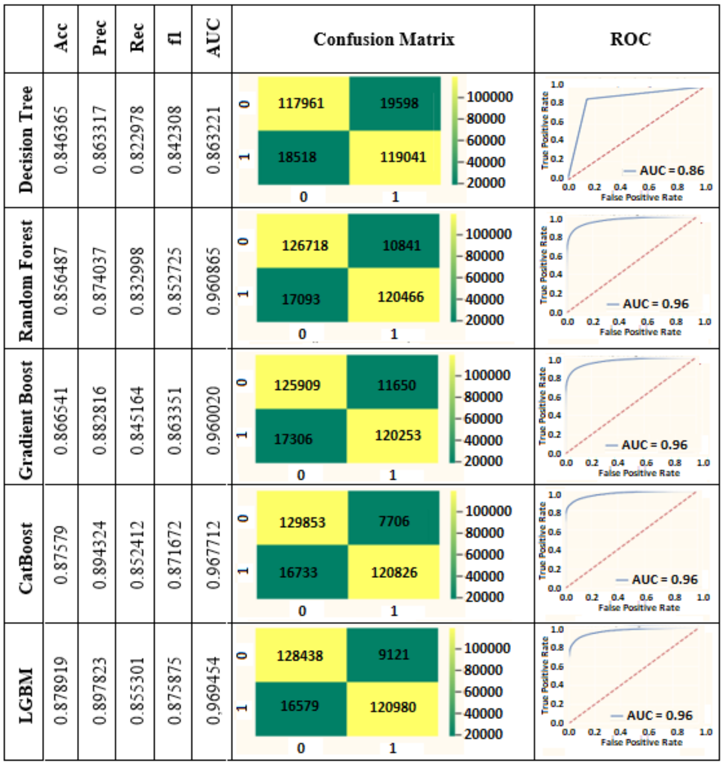
2.14 Evaluation Metric

The model is trained using training data in each iteration carried out in the preceding step, and it is assessed using test data using a number of evaluation metrics, such as accuracy, precision, recall, ROC-AUC score, and F1-score. A thorough assessment of model performance is made possible by the utilization of these diverse measures, particularly when it comes to binary or multi-class classification. The application additionally generates a confusion matrix and categorization report, both of which are displayed using a heatmap. The comprehension of model performance, such as the quantity of true positives, false positives, true negatives, and false negatives, is made easier by these visualizations.

- accuracy (ACC), which refers to the model's fraction of right predictions. This is the ratio of correct forecasts to all predictions.

- Precision (Prec) is the percentage of positive forecasts that are truly positive. This is the ratio of genuine positives to total positive forecasts.
- Recall (Rec) - The percentage of real positive cases that were accurately forecasted as positive. It is the proportion of genuine positives to the sum of true positives and false negatives.
- F1 Score (f1) is the harmonic mean of accuracy and recall. Combines both measurements into a single value.
- AUC (Area Under ROC Curve) - The ROC curve shows the ratio of true positives vs false positives. AUC measures the entire area under this curve from (0.0) to (1.1). Higher AUC indicates better classification
- Confusion Matrix - A summary of correct and incorrect predictions for each class organized in a table. Allows deeper error analysis.

3. Experimental Results and Discussion



	Acc	Prec	Rec	f1	AUC	Confusion Matrix	ROC
Decision Tree	0.846365	0.863317	0.822978	0.842308	0.863221	117961 19598 18518 119041	AUC = 0.86
Random Forest	0.856487	0.874037	0.832998	0.852725	0.960865	126718 10841 17093 120466	AUC = 0.96
Gradient Boost	0.866541	0.882816	0.845164	0.863351	0.960020	125909 11650 17306 120253	AUC = 0.96
CatBoost	0.87579	0.894324	0.852412	0.871672	0.967712	129853 7706 16733 120826	AUC = 0.96
LGBM	0.878919	0.897823	0.855301	0.875875	0.969454	128438 9121 16579 120980	AUC = 0.96

Figure 4. Evaluation results of the training process with 5 decision tree models

The training results of the dataset employing five machine learning models were evaluated using the ROC graph, confusion matrix, and five assessment tools (ACC, Prec, Rec, f1 score, and AUC). To reduce unnecessary page usage, the research findings are presented in a single table (Figure 4). Using the study results table, we analyzed each model as follows:

3.1 Decision Trees

The measurement results obtained: Acc = 0.8463, Prec = 0.8633, Rec = 0.8229, F1 = 0.8423, AUC = 0.8632. Confusion Matrix: TN = 117.961, FP = 19.598, FN = 18.518, TP = 119.041. ROC Curve: AUC = 0.86, lower than other models.

False Positive and False Negative are quite high, indicating this model tends to misclassify both classes. The lowest AUC (0.86), indicating poor discrimination ability compared to other models. Decision Tree is the model with the lowest performance in this test.

3.2 Random Forest

The measurement results obtained: Acc = 0.8565, Prec = 0.8740, Rec = 0.8329, F1 = 0.8527, AUC = 0.9608. Confusion Matrix: TN = 126.718, FP = 10.841, FN = 17.093, TP = 120.466. ROC Curve: AUC = 0.96.
Improvement compared to Decision Tree: Fewer False Positives (10.841 vs. 19.598), indicating the model is better at avoiding false detection of negative classes. Recall increases (0.8329 vs. 0.8229), meaning the model is better at capturing positive cases. AUC increased drastically to 0.96, indicating much better discrimination between classes. Random Forest is far superior to Decision Tree, with a balance between accuracy and class discrimination power.

3.3 Gradient Boost

The measurement results obtained: Acc = 0.8665, Prec = 0.8828, Rec = 0.8451, F1 = 0.8633, AUC = 0.9600. Confusion Matrix: TN = 125.909, FP = 11.650, FN = 17.306, TP = 120.253. ROC Curve: AUC = 0.96, equivalent to Random Forest.
Improvement compared to Random Forest: Precision increased (0.8828 vs. 0.8740), indicating the model is more selective in positive predictions. Recall increased slightly (0.8451 vs. 0.8329), indicating fewer positive cases were missed. AUC remains at 0.96, indicating the model still has high discrimination power. Gradient Boost outperforms Random Forest in precision and recall, making it a better choice for balanced classification.

3.4 CatBoost

The measurement results obtained: Acc = 0.8757, Prec = 0.8943, Rec = 0.8524, F1 = 0.8716, AUC = 0.9677. Confusion Matrix: TN = 129.853, FP = 7.706, FN = 16.733, TP = 120.826. ROC Curve: AUC = 0.96, , indicating very good performance.
CatBoost advantages: Lowest False Positives (7.706), indicating the model is more accurate in avoiding negative misclassification.
Highest Precision (0.8943), meaning the model is best at making correct positive predictions.
Recall is slightly lower than Gradient Boost, but still in the good range.
AUC remains high (0.96), confirming that this model is able to distinguish classes very well. CatBoost excels in Precision and a smaller number of FP errors, making it ideal if False Positives must be minimized.

3.5 LGBM

The measurement results obtained: Acc = 0.8789, Prec = 0.8978, Rec = 0.8553, F1 = 0.8758, AUC = 0.9694. Confusion Matrix: TN = 128.438, FP = 9.121, FN = 16.579, TP = 120.980. ROC Curve: AUC = 0.96, , shows very good performance, shows very good performance.

The highest accuracy (0.8789), making it the best model in overall prediction. The highest Recall (0.8553), meaning this model catches more positive cases than CatBoost. The fewest False Negatives (16,579), indicating this model is very good at detecting the positive class. The AUC remains high (0.96), comparable to other best models. LGBM excels in Recall and the lowest False Negatives, making it the best choice if positive detection must be maximized. The following variables may have an impact on model performance:
Capacity of the Model:

- Decision trees perform worse on test data because they have a tendency to overfit the training data.
- Random Forest enhances generalization and decreases overfitting through ensemble learning.

- By iteratively decreasing model error, boosting techniques are used by Gradient Boosting, CatBoost, and LGBM to increase accuracy.

Model Complexity:

- Compared to Decision Tree and Random Forest, Gradient Boosting, CatBoost, and LGBM models are more complicated, which enables them to manage non-linear interactions more well.

Model Capacity to Manage Unbalanced Data:

- High AUC models (LGBM and CatBoost) outperform low AUC models (Decision Tree) in managing class imbalance.

Hyperparameter Optimization:

- Generally speaking, models like Gradient have less optimal default hyperparameters than CatBoost and LGBM.

4. Conclusion

Based on the training results of five machine learning models Decision Tree, Random Forest, Gradient Boost, CatBoost, and LightGBM (LGBM) on a dataset consisting of 1,750,036 rows of bank customer data, various model options were identified to suit specific analytical needs. If the primary goal is to minimize False Positives, CatBoost is the best choice. To detect more positive cases, LGBM delivers superior performance. Meanwhile, if the objective is to achieve a balance between precision and recall, Gradient Boost is a strong candidate. From the overall analysis, LGBM emerged as the best-performing model, achieving the highest accuracy and recall while minimizing False Negatives. This model attained an accuracy of 0.8789, precision of 0.8978, recall of 0.8553, F1 score of 0.8758, and AUC of 0.9694, demonstrating excellent performance in predicting customer churn. For future research, this study can be extended by exploring deep learning models, such as Recurrent Neural Networks (RNN) or Transformer-based architectures, to capture more complex customer behavior patterns. Additionally, optimizing feature selection using SHAP (SHapley Additive Explanations) can provide deeper insights into the most influential variables for churn prediction. Furthermore, incorporating unsupervised learning techniques, such as clustering, could be a valuable approach to identifying high-risk customer segments, enabling financial institutions to develop more effective retention strategies.

5. References

[1] Byline, “Retail banking costs are rising on multiple fronts, increasing cost to serve,” 2024. [Online]. Available: https://www.mckinsey.com/industries/financial-services/our-insights/the-state-of-retail-banking-profitability-and-growth-in-the-era-of-digital-and-ai

[2] J. Marous, “The Importance of Primacy in Banking,” *The Financial Brand*, 2024. https://thefinancialbrand.com/news/bank-onboarding/the-importance-of-primacy-in-banking-176198

[3] Deloitte, “Kicking it up a notch Taking retail bank cross-selling to the next level.” [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-kickingitupanotch-092614.pdf

[4] T. P. Chi, N. Van Hoa, P. Van Thu, N. A. Tuan, and H. T. Nguyen, “Customer Experience Management in Retail Business : A Theoretical Debate,” vol. 4, no. 5, pp. 854–863, 2024.

[5] R. Yulianto, M. S. Rusli, A. Satyo, B. Karno, W. Hastomo, and A. R. Kardian, “Innovative UNET-Based Steel Defect Detection Using 5 Pretrained Models Innovative UNET-Based Steel Defect Detection Using 5 Pretrained Models,” vol. 10, no. 4, pp. 2365–2378, 2023.

[6] S. Gao, F. Huang, W. Cai, and H. Huang, “Network pruning via Performance Maximization,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 9266–9276, 2021, doi: 10.1109/CVPR46437.2021.00915.

[7] F. E. Usman-Hamza *et al.*, “Empirical analysis of tree-based classification models for customer

churn prediction," *Sci. African*, vol. 23, p. e02054, 2024, doi: https://doi.org/10.1016/j.sciaf.2023.e02054.

[8] E. Domingos, B. Ojeme, and O. Daramola, "Experimental Analysis of Hyperparameters for Deep Learning-Based Churn Prediction in the Banking Sector," *Computation*, vol. 9, no. 3. 2021. doi: 10.3390/computation9030034.

[9] W. Hastomo, A. S. Bayangkari Karno, N. Kalbuana, A. Meiriki, and Sutarno, "Characteristic Parameters of Epoch Deep Learning to Predict Covid-19 Data in Indonesia," *J. Phys. Conf. Ser.*, vol. 1933, no. 1, 2021, doi: 10.1088/1742-6596/1933/1/012050.

[10] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using Bayesian optimization," *Evol. Syst.*, vol. 12, no. 1, pp. 217–223, 2021, doi: 10.1007/s12530-020-09345-2.

[11] A. M. Carrington *et al.*, "Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 329–341, 2023, doi: 10.1109/TPAMI.2022.3145392.

[12] A. S. B. Karno *et al.*, "Classification of cervical spine fractures using 8 variants EfficientNet with transfer learning," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 6, pp. 7065–7077, 2023, doi: 10.11591/ijece.v13i6.pp7065-7077.

[13] S. Farhadpour, T. A. Warner, and A. E. Maxwell, "Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices," *Remote Sensing*, vol. 16, no. 3. 2024. doi: 10.3390/rs16030533.

[14] P. Gupta, A. Varshney, M. R. Khan, R. Ahmed, M. Shuaib, and S. Alam, "Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques," *Procedia Comput. Sci.*, vol. 218, pp. 2575–2584, 2023, doi: https://doi.org/10.1016/j.procs.2023.01.231.

[15] C. Vuppalapati, "Data Engineering and Exploratory Data Analysis Techniques," in *Machine Learning and Artificial Intelligence for Agricultural Economics: Prognostic Data Analytics to Serve Small Scale Farmers Worldwide*, Cham: Springer International Publishing, 2021, pp. 75–158. doi: 10.1007/978-3-030-77485-1_2.

[16] E. 1 Playground Series - Season 4, "Binary Classification with a Bank Churn Dataset," *kaggle.com*, 2024. https://kaggle.com/competitions/playground-series-s4e1 (accessed Mar. 05, 2024).

[17] K. P. N. V Satya Sree, J. Karthik, C. Niharika, P. V. V. S. Srinivas, N. Ravinder, and C. Prasad, "Optimized Conversion of Categorical and Numerical Features in Machine Learning Models," in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2021, pp. 294–299. doi: 10.1109/I-SMAC52330.2021.9640967.

[18] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," *IEEE Access*, vol. 9, pp. 114381–114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

[19] S. Dasari and R. Kaluri, "An Effective Classification of DDoS Attacks in a Distributed Network by Adopting Hierarchical Machine Learning and Hyperparameters Optimization Techniques," *IEEE Access*, vol. 12, pp. 10834–10845, 2024, doi: 10.1109/ACCESS.2024.3352281.

[20] S. Boeschoten, C. Catal, B. Tekinerdogan, A. Lommen, and M. Blokland, "The automation of the development of classification models and improvement of model quality using feature engineering techniques," *Expert Syst. Appl.*, vol. 213, p. 118912, 2023, doi: https://doi.org/10.1016/j.eswa.2022.118912.

[21] M. S. Ali, M. K. Islam, A. A. Das, D. U. S. Duranta, M. F. Haque, and M. H. Rahman, "A Novel Approach for Best Parameters Selection and Feature Engineering to Analyze and Detect Diabetes: Machine Learning Insights," *Biomed Res. Int.*, vol. 2023, no. 1, p. 8583210, Jan. 2023, doi: https://doi.org/10.1155/2023/8583210.

[22] D. Singh and B. Singh, "Feature wise normalization: An effective way of normalizing data," *Pattern Recognit.*, vol. 122, p. 108307, 2022, doi: https://doi.org/10.1016/j.patcog.2021.108307.

[23] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 3, p. 160, 2021, doi: 10.1007/s42979-021-00592-x.

[24] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Eng. Appl. Artif. Intell.*, vol. 115, p. 105151, 2022, doi: https://doi.org/10.1016/j.engappai.2022.105151.

[25] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning," *Decis. Anal. J.*, vol. 3, p. 100071, 2022, doi: https://doi.org/10.1016/j.dajour.2022.100071.

[26] D. Chicco, L. Oneto, and E. Tavazzi, "Eleven quick tips for data cleaning and feature engineering," *PLOS Comput. Biol.*, vol. 18, no. 12, p. e1010718, Dec. 2022, [Online]. Available: https://doi.org/10.1371/journal.pcbi.1010718

[27] L. Wang, M. Han, X. Li, N. Zhang, and H. Cheng, "Review of Classification Methods on Unbalanced Data Sets," *IEEE Access*, vol. 9, pp. 64606–64628, 2021, doi: 10.1109/ACCESS.2021.3074243.

[28] L. Barreñada, P. Dhiman, D. Timmerman, A.-L. Boulesteix, and B. Van Calster, "Understanding overfitting in random forest for probability estimation: a visualization and simulation study," *Diagnostic Progn. Res.*, vol. 8, no. 1, p. 14, 2024, doi: 10.1186/s41512-024-00177-1.

[29] A. V Konstantinov and L. V Utkin, "Interpretable machine learning with an ensemble of gradient boosting machines," *Knowledge-Based Syst.*, vol. 222, p. 106993, 2021, doi: https://doi.org/10.1016/j.knosys.2021.106993.

[30] M. Hajihosseinlou, A. Maghsoudi, and R. Ghezelbash, "A Novel Scheme for Mapping of MVT-Type Pb–Zn Prospectivity: LightGBM, a Highly Efficient Gradient Boosting Decision Tree Machine Learning Algorithm," *Nat. Resour. Res.*, vol. 32, no. 6, pp. 2417–2438, 2023, doi: 10.1007/s11053-023-10249-6.

[31] M. Luo *et al.*, "Combination of Feature Selection and CatBoost for Prediction: The First Application to the Estimation of Aboveground Biomass," *Forests*, vol. 12, no. 2. 2021. doi: 10.3390/f12020216.

[32] C. Li *et al.*, "Machine learning based early mortality prediction in the emergency department," *Int. J. Med. Inform.*, vol. 155, p. 104570, 2021, doi: https://doi.org/10.1016/j.ijmedinf.2021.104570.

[33] S. Lee, T. P. Vo, H.-T. Thai, J. Lee, and V. Patel, "Strength prediction of concrete-filled steel tubular columns using Categorical Gradient Boosting algorithm," *Eng. Struct.*, vol. 238, p. 112109, 2021, doi: https://doi.org/10.1016/j.engstruct.2021.112109.

[34] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems*, 2017, vol. 30. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[35] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, 2018, vol. 31. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf

[36] Z. Rahmatinejad *et al.*, "A comparative study of explainable ensemble learning and logistic regression for predicting in-hospital mortality in the emergency department," *Sci. Rep.*, vol. 14, no. 1, p. 3406, 2024, doi: 10.1038/s41598-024-54038-4.

[37] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1937–1967, 2021, doi: 10.1007/s10462-020-09896-5.

[38] S. Jeong, J. Ko, and J.-M. Yeom, "Predicting rice yield at pixel scale through synthetic use of crop

and deep learning models with satellite data in South and North Korea," *Sci. Total Environ.*, vol. 802, p. 149726, 2022, doi: https://doi.org/10.1016/j.scitotenv.2021.149726.

[39] Z. Li and D. Hu, "Forecast of the COVID-19 Epidemic Based on RF-BOA-LightGBM," *Healthcare*, vol. 9, no. 9. 2021. doi: 10.3390/healthcare9091172.

[40] H. Zhang *et al.*, "High-Resolution Vegetation Mapping Using eXtreme Gradient Boosting Based on Extensive Features," *Remote Sens.*, vol. 11, no. 12, 2019, doi: 10.3390/rs11121505.