scientific reports

OPEN

Mitigating class imbalance in churn prediction with ensemble methods and SMOTE

R. Suguna[1], J. Suriya Prakash[2], H. Aditya Pai[3], T. R. Mahesh[2], Venkatesan Vinoth Kumar[4] & Temesgen Engida Yimer[5]✉

This study examines how imbalanced datasets affect the accuracy of machine learning models, especially in predictive analytics applications such as churn prediction. When datasets are skewed towards the majority class, it can lead to biased model performance, reducing overall effectiveness. To analyze this impact, the research utilizes a churn dataset to evaluate how data imbalance influences model accuracy. The study utilized nine individual classifiers along with six homogeneous ensemble models to evaluate the effects of imbalanced data on model performance. Single classifier models struggle to identify underlying patterns in imbalanced data, while ensembles improve predictive performance by focusing on the minority class. However, when trained on unbalanced data, their accuracy remains subpar. The top six classifiers were selected for further investigation based on their performance on the imbalanced data. A SMOTE sampling technique was employed to create a balanced dataset, ensuring that all classes were adequately represented. The generated model's performance improved from 61 to 79%, indicating the removal of bias in the target data. The results showed that Adaboost, an optimal classifier, demonstrated superior performance with an F1-Score of 87.6% in identifying potential churn and assessing customer account health. The findings emphasize the importance of balanced datasets for accurate ML model predictions.

Keywords Churn prediction, Ensemble models, Imbalanced data, Machine learning, Predictive analytics, Sampling techniques, Single classifier

Machine learning (ML) approaches are becoming increasingly common for predictive analytics applications like churn prediction. With the help of these techniques, companies can predict client attrition, spot possible threats, and create proactive customer retention plans. ML models are trained on a representative sample of both the majority and minority classes, balanced datasets enable them to discover the underlying patterns and characteristics connected to each class. ML models can predict unseen data more accurately and broaden their applicability to real-world events by training on balanced data.

Churn prediction involves leveraging data insights to anticipate customer churn and intervene promptly. Customer loyalty is built upon exceptional services and exceeding customer expectations. By harnessing data and utilizing advanced analytics, financial companies can strengthen customer relationships and maintain a competitive edge in the market[1].

Customer churn is a key sector concern, influencing long-term profitability and client retention efforts[2]. Businesses that fail to foresee and control churn risk losing valued consumers. Many studies have considered strategies for accurate churn prediction to enable proactive decision-making. Existing churn prediction algorithms generally suffer from imbalanced datasets, where the dominant class (non-churned customers) greatly surpasses the minority class (churned customers). This imbalance leads to biased models that misclassify churned clients, lowering the overall effectiveness of prediction systems[3,4]. Researchers pointed out that traditional models favours the dominant class, producing good accuracy but low recall for the minority class[5]. There is a demand to address this challenge by requiring novel ways to boost model sensitivity and assure impartial representation of both classes. A survey identified 24 studies from 2003 to 2019 focusing on single,

[1]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India. [2]Department of Computer Science and Engineering, JAIN (Deemed-to-be-University), Bangaluru, Karnataka, India. [3]Department of CSE, MIT School of Computing, MIT Art, Design and Technology University, Pune 412201, India. [4]School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore 632001, India. [5]Department of Mathematics, Dilla University, Dilla, Ethiopia. ✉email: Temesgen.engida@du.edu.et

hybrid, and ensemble method design architectures, statistical analysis of classification algorithms, and datasets used in various research papers[6].

Several studies have investigated approaches to alleviate class imbalance, particularly using data resampling and advanced machine learning algorithm. Oversampling techniques such as the Synthetic Minority Oversampling Technique (SMOTE) and random oversampling have been widely utilized by the researchers to produce synthetic minority class instances, henceforth enhancing model resilience and prediction performance[7]. But they may not always generalize effectively across datasets. Some researchers have further refined SMOTE by adding additional mechanisms, such as outlier detection, as proved in the Adaptive Synthetic (ADASYN) technique. This approach has been used in credit card fraud detection and churn prediction with promising results[8]. Moreover, ensemble learning approaches, including bagging, boosting, and stacking, have been demonstrated by the researchers to enhance churn prediction by integrating several base models to reduce bias and improve accuracy[9]. For instance, earlier research has revealed that models combining Random Forest and XGBoost with SMOTE can obtain an accuracy of above 91%[10]. Similarly, hybrid approaches employing Support Vector Machines (SVM) and various resampling techniques have also proved effective in[4]. Research works such as hybrid sampling methodology, in conjunction with Random Forest, generated better results in a comparison analysis, underscoring the necessity of investigating diverse resampling procedures[11]. All existing approaches have not used balanced_accuracy metric to evaluate the models.

To obtain the best results, the choice of ensembles and resampling methodology should be customised for the specified dataset. SMOTE and other oversampling methods can increase sensitivity, but sometimes they might not produce good accuracy when compared to models trained on original datasets. Even though these techniques have the potential to enhance prediction accuracy, the inherent difficulties of class imbalance still require careful consideration and further investigation.

A balance between sensitivity and accuracy must be understood for customer retention initiatives to be successful. Churn prediction models should take into account both sensitivity and the overall impact on business outcomes. These models can be enhanced using machine learning and advanced analytics, which can help businesses market their goods. Additionally, it enhances customer happiness and fortifies the brand's image. Businesses can establish themselves as industry leaders by leveraging these technologies.

This research focuses on enhancing churn prediction models through a combination of synthetic minority oversampling (SMOTE) and ensemble learning techniques. Unlike standard models that struggle with imbalanced datasets, the proposed technique boosts accuracy, making it appropriate for financial institutions, telecoms, and subscription-based services. For instance, early detection of high-risk clients enables banks to provide customized incentives, lowering the number of account closures. Better churn prediction algorithms can also assist telecommunications companies enhance customer service interactions by providing better contract conditions before churn happens. By developing machine learning models, firms may proactively identify high-risk consumers and adopt focused retention measures, thereby reducing revenue loss. This study improves upon earlier research by integrating SMOTE with several ensemble classifiers, assessing their performance, and identifying the most robust technique for churn prediction.

Specifically, the study assesses the influence of homogeneous ensemble classifiers (such AdaBoost and Gradient Boosting) on prediction performance in conjunction with SMOTE. This work offers important insights for optimizing churn prediction algorithms for practical applications by methodically evaluating these models against current approaches.

The research contributions are as follows.

- It offers a thorough examination of how unbalanced datasets affect the functionality of several machine learning models, such as single classifiers and homogenous ensemble classifiers.
- It shows how well SMOTE balances datasets, which helps ML models better spot trends in the minority class.
- It integrates SMOTE with ensemble techniques like AdaBoost and Gradient Boosting to improve prediction performance and reduce overfitting.
- It provides useful insights by thoroughly contrasting ensemble and single classifier approaches on balanced and unbalanced datasets. The performance is assessed using Balanced Accuracy metric in the proposed work. To ensure that the majority and minority classes are adequately represented in performance evaluations, balanced accuracy is a vital criterion for assessing models on imbalanced datasets.
- It emphasizes how feature significance and hyperparameter adjustment can enhance the precision and comprehensibility of the model.
- By locating and optimizing the best classifiers, it offers a reliable method for predicting churn and evaluating the general health of client accounts.

Related work

When customers discontinue their relationship with a business or switch to competitors, that is known as customer churn. While the specific definition of churn varies across different industries, a common example is the banking sector. Here, customers are typically classified as churned if their financial activity, such as transaction volume and account balance, drops significantly — often by 30% annually[12]. The high cost of losing profitable customers has shifted business focus from acquiring new clients to retaining existing ones. Customer churn is not an isolated event; it can have a ripple effect within social groups. This is because customers often influence each other's decisions, and the departure of one customer can increase the likelihood of others leaving as well[13]. One potential strategy for managing the connection between customer satisfaction and loyalty involves restricting a customer's ability to switch providers through contractual obligations[14].

Businesses prioritize profitability when selecting predictive models and often employ metrics like Maximum Profit Criterion (MPC) and Expected Maximum Profit (EMP) for this purpose. These financial measures surpass conventional performance metrics by directly assessing a model's ability to maximize profits while retaining high-value customers[15]. Bogaert and Delaere proved that heterogeneous ensembles outperform in terms of Area under the Curve (AUC) and EMP[9]. Kurtcan and Ozcan enhanced the performance of a Support Vector Machine (SVM) model by optimizing its settings using Grey Wolf Optimization. They applied this improved SVM to predict customer churn in the telecommunications sector and compared its accuracy to other models like logistic regression, Naive Bayes, decision trees, and a standard SVM. The findings revealed that the optimized SVM consistently produced better results than the other models tested[16].

By analyzing oversampling, undersampling, and hybrid approaches on an unbalanced telecommunications dataset, Jia-Xuan et al. examine resampling techniques for customer churn prediction and discover that a hybrid SMOTE-ENN approach produces an F1 score of 95.3% and an accuracy of 96.0%[17]. SMOTE with SVM is used in churn prediction to handle imbalances in churn datasets[4,18]. The performance of SVMs is measured with the gain score. SVM with RBF kernel performs with a gain measure of 80% in 20th percentile[4]. A combination of SMOTE and ensemble learning enhances classification performance by addressing class imbalance and improves F1-Score. Different classification algorithms and voting strategies supported the efficiency of the approach[19]. Faruq et al. used various classifiers, including K-Neighbors (KNN), Random Forest, XGboost, Adaboost, and Ensemble Model, on a Kaggle dataset with imbalances. Random Oversampling was also added producing average accuracy of 97.31%[20]. SMOTE with Genetic Algorithm (GA) for feature selection are combined and applied on four different classifiers. KNN outperformed with precision and an F-measure of 96%[7].

Several research studies have employed various undersampling techniques to address the imbalance in telecommunication churn datasets[21]. An overview of sampling techniques used in churn prediction models is listed in Table 1.

An ensemble approach was explored by combining various classifiers, including neural networks, XGBoost, AdaBoost, KNN, SVM, random forests, and logistic regression. These combined models were compared to their counterparts on customer churn datasets. The results indicated that an ensemble composed of logistic regression, neural networks, and AdaBoost achieved the highest predictive accuracy[21]. SMOTE, a technique for oversampling minority classes, has been utilized in churn prediction research, as evidenced by studies[22,29,30].

Jiang proposed a novel approach for feature selection in churn prediction by combining a modified multi-objective atomic orbital search with an extreme learning machine. Inspired by quantum mechanics, this method employs atomic orbital-like structures to effectively navigate the complex landscape of multi-objective optimization[31]. Zue et al. introduced a novel ensemble method based on bagging to enhance churn prediction accuracy in imbalanced datasets[32]. This approach involves generating diverse training subsets with varying class ratios using a cost-weighted negative binomial distribution. Subsequently, cost-sensitive logistic regression with a Lasso penalty is applied to combine the predictions of multiple base classifiers. Comparative analysis with twelve state-of-the-art methods on telecommunication data demonstrated superior performance of the proposed model in terms of traditional and profit-based evaluation metrics. A heterogeneous ensemble model is suggested for churn prediction in the telecom sector, that uses distinct benefits of each base classifier along with the group knowledge to enhance prediction performance with a meta-learner[33]. Kate et al. explored the application of various Generative Adversarial Network (GAN) techniques to address class imbalance by oversampling the minority class in churn prediction[34]. Comparative analysis with other classifiers demonstrated that GAN-based oversampling significantly enhanced predictive performance.

Brito et al. employed a combined oversampling and undersampling approach to balance a financial dataset[35]. The ADASYN method was initially used to introduce 5% synthetic churn customers through oversampling, followed by the NearMiss6 technique to reduce the non-churn customer population by 65%. Pustokhina et al. employed a multi-objective rain optimization algorithm to determine the optimal sample size when applying the Synthetic Minority Oversampling Technique (SMOTE)[27]. Specifically, average account balance, customer relationship length, and transaction frequency were identified as key financial indicators of customer churn. Additionally, behavioral patterns have been recognized as crucial factors in predicting churn across various industries, including online gaming, telecommunications, and finance[36]. Mena et al. introduced a dynamic recency, frequency, and monetary value (RFM) approach utilizing deep learning for churn prediction in the financial sector[37]. While RFM remains a foundational framework, subsequent research has expanded its scope by incorporating additional variables to enhance predictive accuracy. Similarly, Smaili and Hachimi expanded

Technique	Category	Description	References
Random Oversampling	Oversampling	Increases the samples through redundancy	20
SMOTE	Oversampling	Synthetic Samples are generated	7,10,18,19,22
ADASYN	Oversampling	Focus on more challenging samples to produce synthetic instances	23
Borderline-SMOTE	Oversampling	Variant of SMOTE concentrating on border samples	24
Random Undersampling	Undersampling	Removes the samples randomly	25,26
Edited Nearest Neighbors (ENN)	Undersampling	Removes majority class samples that are misclassified by KNN	25
NearMiss	Undersampling	Removes the samples of majority class that are near to minority class.	27
Tomek Links	Undersampling	Removes one sample per class at a time	28
SMOTE-ENN	Hybrid	Combines SMOTE and ENN and applies balancing in the data	17
SMOTE-Tomek Links	Hybrid	Removes noisy majority class samples and improves minority class instances	28

Table 1. Sampling techniques employed in churn prediction.

the traditional RFM model by incorporating customer purchasing behaviour diversity, creating the RFM-D framework[38].

Amin et al. employed a genetic algorithm in conjunction with Naive Bayes to identify the most relevant features for predicting customer churn in the telecommunications sector[39]. Their method demonstrated superior performance compared to other advanced techniques. Utilizing SMOTE and ADASYN for resampling, RF achieved high accuracy and ROC-AUC scores in predicting churn[40]. Sadeghi et al. integrated Principal Component Analysis (PCA) and Particle Swarm Optimization (PSO) with the K-means algorithm to enhance clustering performance[41]. By optimizing centroid initialization and reducing data dimensionality, this approach aimed to improve clustering accuracy. The developed system exhibited sensitivity and specificity score of 75% and 99.8% respectively. A summary of various feature selection techniques employed for customer churn prediction is presented in Table 2.

Simovic et al. utilized logistic regression with a mixed penalty to mitigate overfitting[47]. Their findings indicate that incorporating a mixed penalty into logistic regression led to improved performance in classification metrics compared to the standard logistic regression model on CRM dataset. Recent studies have leveraged deep learning for customer churn prediction, achieving an impressive accuracy F1 score of 91% using CNN. However, the complex nature of these models, characterized by numerous parameters and intricate layers, hinders interpretability. As a result, understanding how input data influences model outputs becomes challenging[48]. A recent study identified service quality, customer satisfaction, subscription plan upgrade offers, and network coverage as key factors influencing customer churn within the Danish telecommunications industry[49]. Consistent with other findings, Li et al. demonstrated the superior performance of Light Gradient Boosting Machine (LightGBM) over SVM, Random Forest, and XGBoost in a financial dataset context[50,51]. To address the class imbalance issue, the authors[50] further enhanced LightGBM by incorporating a focal loss function. This loss function prioritizes misclassified minority class instances and difficult-to-classify samples, thereby improving the model's ability to handle imbalanced data. The proposed model was able to identify the churn rate of 0.94 producing an AUC score of 0.99.

Proposed methodology

Financial companies are increasingly utilizing data insights and advanced analytics techniques to predict churn and retain valuable customers. Exploratory data analysis (EDA) is used to analyze collected data, identifying outliers and potential variables. Predictive insights enable data-driven decisions and resource allocation to retain valuable customers. This strategy enables financial companies to foster long-term customer relationships, optimize business outcomes, and drive sustainable growth in a competitive market.

Exploratory data analysis and dataset nature

This research aims to develop a model that helps bank executives identify customers at risk of discontinuing their banking services. The study utilizes a dataset sourced from Kaggle, available at https://www.kaggle.com/shrutimechlearn/churn-modelling.

The dataset consists of 10,000 complete customer records with no missing values. It includes 14 attributes, with 7 related to personal customer details, 4 reflecting account status, and 2 providing insights into product usage. The target attribute is binary and represents the customer's exit status.

Visualizing variable distributions

Categorical information is transformed into numerical form. The distribution of categorical variables is depicted in Fig. 1. The distribution communicates changeable information.

Categorical Variables such as Geography, IsMale (Gender), HasCrCard, and IsActiveMember follow the Bernoulli distribution. Continuous Variables like Credit Score and Balance are distributed normally and Estimated Salary is distributed uniformly.

Technique	Category	Description	References
ANOVA and Pearson Correlation	Statistical	Relates the features to target variable	42
mRMR (Minimum Redundancy Maximum Relevance)	Wrapper	Selects related and non redundant features	43
Recursive Feature Elimination (RFE)	Wrapper	Removes very least important features	42
Hybrid Algorithms	Combined	PSO-SA (Particle Swarm Optimization - Simulated Annealing) algorithm	41,44
Archimedes optimization algorithm-based feature selection (AOAFS)	Dimensionality Reduction	Selects significant features, addresses dimensionality issues	45
L1 Regularization (Lasso)	Embedded	Selects relevant features automatically	38
Tree-Based Feature Importance	Embedded	Uses GridSearch to identify significant features	46
PCA (Principal Component Analysis)	Dimensionality Reduction	Preserving maximum variance features are transformed into uncorrelated units.	35

Table 2. Feature selection techniques employed in churn prediction.

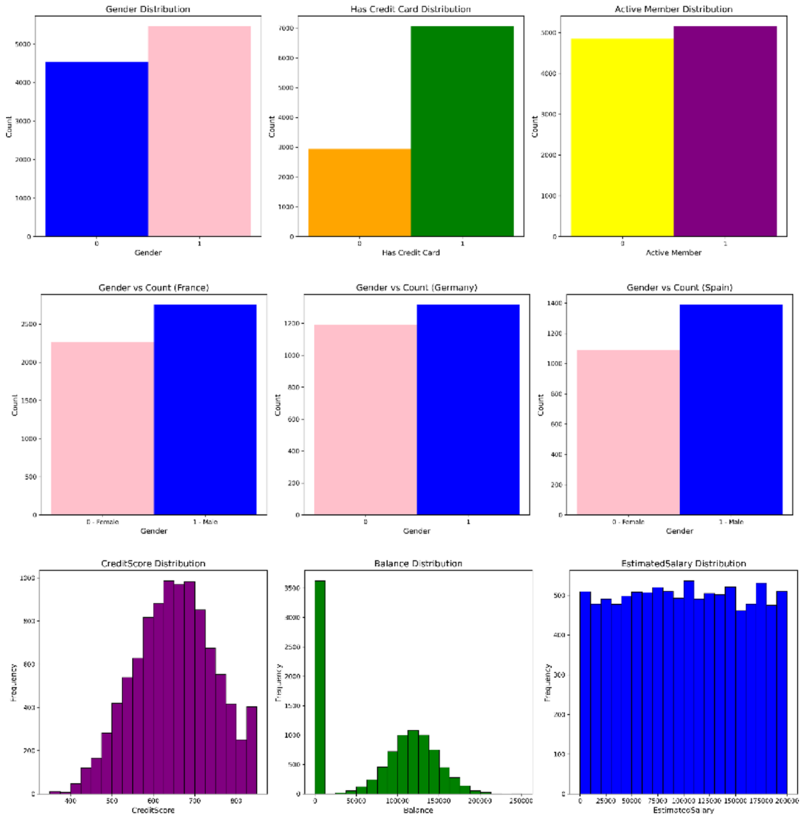

Fig. 1. Categorical and continuous variable distribution.

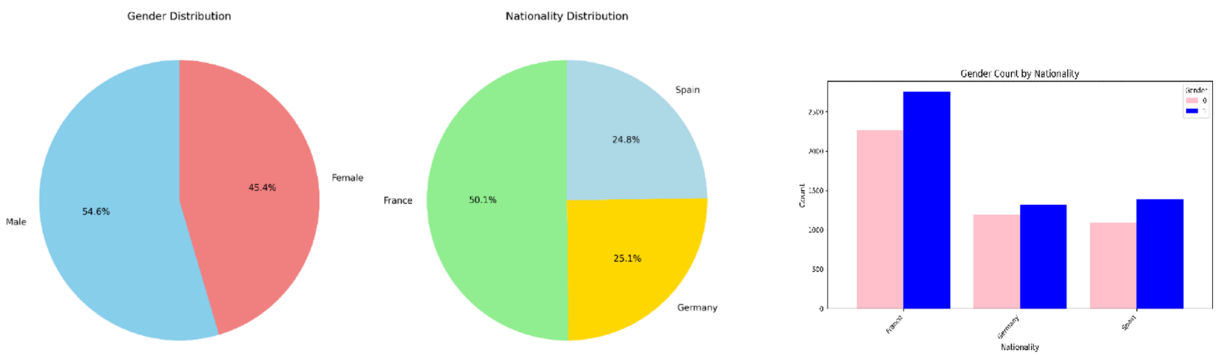

Fig. 2. Gender with demographic distribution.

Customer characteristics

The proportion of clients by nationality and gender must be established to understand client details. Figure 2 shows the gender distribution across the nation and the proportion of clients by nationality and gender overall. gender distribution across the nation and the proportion of clients by nationality and gender overall.

According to the graph, 50% of the clients are French, with the remainder split evenly between customers from Spain and Germany. When it comes to gender, men hold somewhat more accounts than women, and this is reflected in nationwide data.

The estimated pay distribution by gender is shown in Fig. 3. Their median incomes are comparable, and there isn't much of a wage gap.

There are no substantial wage variations based on age. Figure 4 illustrates that a few consumers are older citizens, with a bit of variance in salary range.

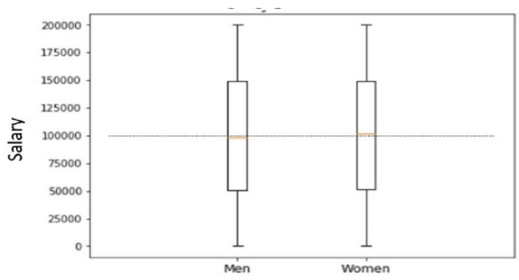

Fig. 3. Salary distribution based on gender.

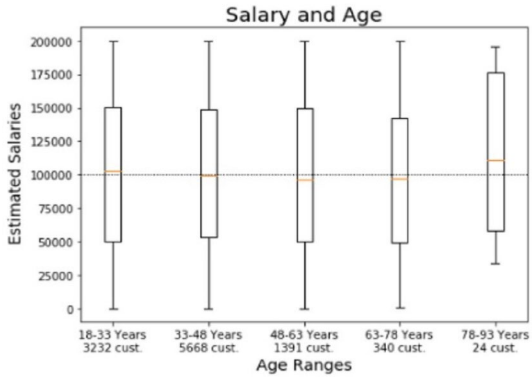

Fig. 4. Salary distribution based on age.

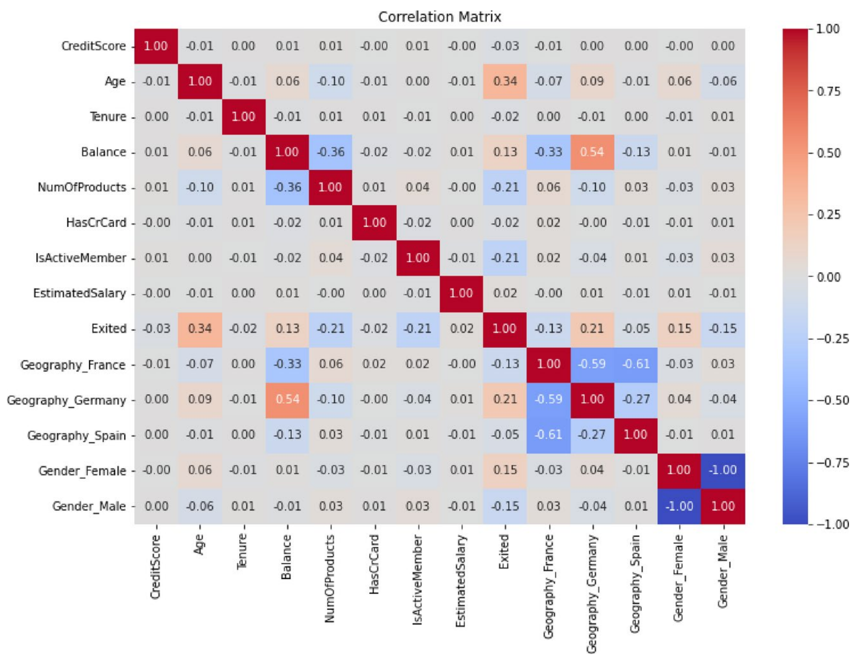

Fig. 5. Measuring the correlation between factors.

Correlation of factors

The correlation matrix can be used to determine the link between variables. The correlation map for the variables is shown in Fig. 5. There exists a strong relationship between the Nationality of Germany and account balance. Balance is correlated with age using the exit variable. Germans exhibit a higher rate of attrition.

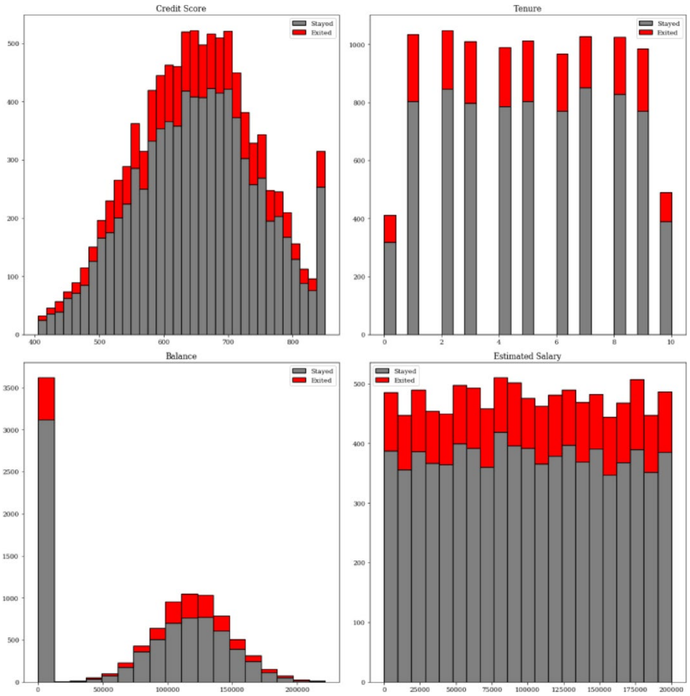

Fig. 6. Distribution of parameters related to a target variable.

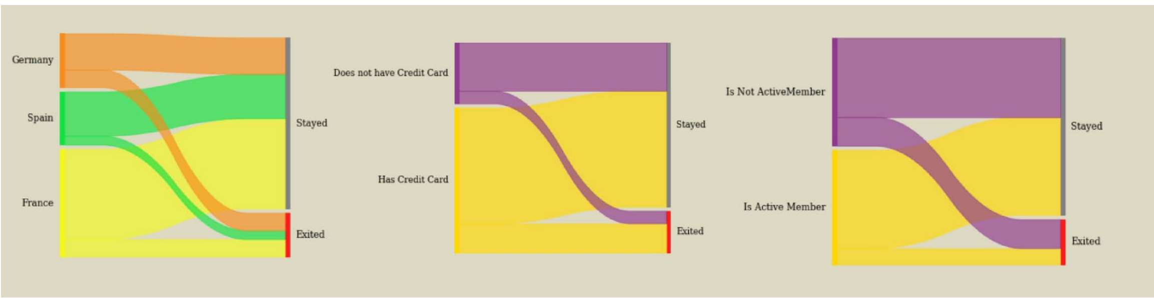

Fig. 7. Sankey diagram depicting the flow of attributes with target value.

Relationships of attributes with a target variable
The distribution of parameters related to the target variable is depicted in Fig. 6. Using a Sankey diagram depicted in Fig. 7, it can be seen how information moves from one variable to another. The width of the flow represents the ratio of quantity flow between variables.

Here are some conclusions we can draw from the charts above. In terms of gender, women are less prevalent than men, but they shut accounts at a higher rate. In Germany, the percentage of customers who have left is more significant (32%, or roughly 2x higher), compared to Spain and France, where it is lower (16% each).

Customers under the age of 40 and those over the age of 65 are more likely to maintain their accounts. The decision to remain a customer at the bank is unaffected by whether one has a credit card (20% of customers in both groups have left). Compared to active customers, non-active members are more likely to stop using a bank's service.

The initial exploratory investigation provides insights into customers and suggests a customer-focused strategy.

Building single and ensemble classifiers
Single classifiers
In machine learning, a model is trained to carry out a particular task—like classification—without aggregating the results of other models and is referred to as a single classifier. The following are the salient features of single classifiers:

- Independence: A single classifier functions on its own, generating predictions from the input data using a custom learning method.

- Simplicity: When opposed to ensemble approaches (which mix numerous models), single classifiers are typically easier to develop and comprehend.

Single Classifiers are categorized as parametric and Non-parametric.

- Parametric classifiers: Require fewer data points, have a set number of parameters, and assume a certain structure for the data distribution. Naive Bayes, logistic regression, and LDA are a few examples.
- Non-parametric classifiers: Generally, need more data points and have a variable number of parameters that increase with the data. They also do not presuppose a particular shape for the data distribution. KNN, decision trees, and non-linear SVMs are a few examples.

Table 3 shows the included classifiers for the study.

Table 4 shows the representation Single Classifiers.

Algorithm 1 explains the pseudo-code of single classifier function.

```
Input:
 - Training data: X_train (features), y_train (labels)
 - Hyperparameters: params (e.g., max_depth, learning_rate,
                                        regularization)
 - Optional: Cross-validation parameters, data preprocessing steps
 Output:
 - Trained classifier model: model

function TrainClassifier(X_train, y_train, params):
   model = InitializeModel(params)   # Initialize the classifier with hyperparameters
   model.fit(X_train, y_train)        # Train the model on the training data
   return model                            # Return the trained model
```

Algorithm 1 Pseudo-code of single classifier function.

Homogenous classifiers

A homogeneous classifier ensemble employs several instances of the same learning algorithm to create a more powerful predictive model. The secret to their effectiveness is to vary the data subsets, feature subsets, or hyperparameter settings used to create diversity among the classifiers. Known methods such as Gradient Boosting, AdaBoost, and Random Forest are instances of homogeneous classifier ensembles. Compared to individual classifiers, these ensembles enhance performance, and robustness, and lower the chance of overfitting.

Classifier	Description	Parameters
Parametric classifiers		
Logistic regressor (LR)	There is a linear connection between the target variable's log-odds and the input characteristics.	Bias and Weights
Linear discriminant analysis (LDA)	Every class has a common covariance matrix and a Gaussian distribution.	Mean vectors, shared covariance matrix, and the prior probabilities for every class
Naive Bayes (NB)	Predictors within each class are assumed to be independent.	Probabilities (for categorical data), means and variances (for continuous features).
Support vector machines (with linear kernel) (LSVM)	Assumes a linear decision boundary between classes	Bias and Weights
Quadratic discriminant analysis (QDA)	Assumes that each class in the data follows a Gaussian (normal) distribution, allowing for each class to have its covariance matrix.	Mean vectors, separate covariance matrix, and the prior probabilities for every class
Multi layer perceptron (MLP)	Neural network that consists of multiple layers of neurons: an input layer, one or more hidden layers, and an output layer.	Bias and Weights
Non parametric cassifiers		
K-nearest neighbors (KNN)	A data point is classified according to the majority class of its k nearest neighbors.	Number of neighbors k
Decision trees (DT)	Models judgments based on feature values in a manner resembling a tree.	Split points at each node; the structure and number of nodes
Support vector machines (with non-linear kernel) (KSVM)	kernel functions map input features into higher-dimensional space.	Support vectors; the number and nature

Table 3. Single classifiers under study.

Classifier	Representation	Description
Parametric classifiers		
Logistic regressor (LR)	$\widehat{y}=\sigma\left(w.x+b\right)$	$\widehat{y}$ is the predicted probability of the positive class. $\sigma\ (z)$ is the sigmoid function, defined as $\sigma\ (z)=\frac{1}{1+e^{-z}}$ w is the vector of weights. x is the input feature vector. b is the bias term.
Linear discriminant analysis (LDA)	$\delta_{\ k}\left(x\right)=x^{T}\Sigma^{\ -1}\mu_{\ k}-\frac{1}{2}\mu_{\ k}^{\ T}\Sigma^{\ -1}\mu_{\ k}+log\,\pi_{\ k}$	x is the input feature vector. $\mu_{\ k}$ is the mean vector of class Σ is the common covariance matrix. $\pi_{\ k}$ is the prior probability of class
Naive Bayes (NB)	$\widehat{y}=argmax_{k}\left(\log P\left(C_{k}\right)+\sum_{i=1}^{n}logP\left(x_{i}\middle\vert C_{k}\right)\right)$	$\widehat{y}$ is the predicted class label $P\left(C_{k}\right)$ Class Prior Probability $P\left(x_{i}\vert C_{k}\right)$ Conditional probability
Support vector machines (with linear kernel) (LSVM)	$\widehat{y}=sign(w.x+b)$	$\widehat{y}$ is the predicted class label (+1 or −1). w is the vector of weights. x is the input feature vector. b is the bias term. $sign\ (z)$ is the sign function, returns + 1 if z > 0, -1 otherwise
Quadratic discriminant analysis (QDA)	$\delta_{\ k}\left(x\right)=-\frac{1}{2}\log\vert\Sigma_{\ k}\vert-\frac{1}{2}(x-\mu_{\ k})^{T}\Sigma_{\ k}^{\ -1}(x-\mu_{\ k})\ +log\,\pi_{\ k}$	x is the input feature vector. $\mu_{\ k}$ is the mean vector of class $\Sigma_{\ k}$ is the covariance matrix of class k. $\pi_{\ k}$ is the prior probability of class $\vert\Sigma_{\ k}\vert$ is the determinant of the covariance matrix $\Sigma_{\ k}$
Multi layer perceptron (MLP)	$\widehat{y}=\ \varnothing\ (W^{(L)}h^{(L-1)}+b^{(L)})$	L- Total number of layers W – Weight of layers h – hidden layer output $\varnothing\ (.)$ Activation function
Non parametric classifiers		
K-nearest neighbors (KNN)	$\widehat{y}=mode\left(\{y_{i}\ \middle\vert\ i\,\epsilon\,KNN_{(x)}\ \}\right)$	$\widehat{y}$is the predicted class label. KNN $KNN_{(x)}$ is the set of indices of the nearest neighbors of x y_{i} are the class labels of the nearest neighbors.
Decision trees (DT)	$if\ x_{1}\ \le\ \theta_{\ 1}\ then$ $if\ x_{2}\ \le\ \theta_{\ 2}\ then\ \widehat{y}-\ y_{1}$ $else\ \widehat{y}=\ y_{2}$ $else$ $if\ x_{3}\ \le\ \theta_{\ 3}\ then\ \widehat{y}-\ y_{3}$ $else\ \widehat{y}=\ y_{4}$	x_{i} are the feature values. $\theta_{\ i}$ are the threshold values for the splits. $\widehat{y}$ is the predicted class label
Support vector machines (with non-linear kernel) (KSVM)	$\widehat{y}=sign(\sum_{i=1}^{n}\alpha_{\ i}y_{i}K\left(x_{i},x\right)+b)$	x is the input feature vector. $\alpha_{\ i}$ are the Lagrange multipliers (support vector coefficients). y_{i} are the class labels of the training examples (± 1 ± 1). $K\left(x_{i},x\right)$is the kernel function. b is the bias term.

Table 4. Single classifiers representation.

Classifier	Description	Parameters
Bagging		
Random Forest (RF)	Combines the predictions of several decision trees by averaging (for regression) or majority voting (for classification) after each tree has been trained on various bootstrap samples.	n_estimators, max_depth, max_leaf_ nodes, criterion, max_features, bootstrap
Boosting		
Gradient Boosting(GB)	Decision trees are added successively, each trained to forecast residuals of the existing model, and their combined forecasts enhance overall performance.	n_estimators, learning_rate, max_depth, min_samples_split, max_features
Ada Boosting (ADA)	A group of weak learners that are successively trained (usually decision trees). Every new classifier concentrates more on the cases that the prior classifiers misclassified.	n_estimators, learning_rate, base_estimator, algorithm, random_state
XG Boosting (XGB)	Combines the strengths of gradient boosting with optimizations for speed and performance.	learning_rate, max_depth, subsample, colsample_bytree
CAT Boosting (CAT)	Handle categorical features efficiently without the need for extensive preprocessing	learning_rate, depth, l2_leaf_reg, iterations
Light GBM (LGBM)	Gradient boosting framework designed to be distributed, efficient, and capable of handling large-scale data	learning_rate, max_depth, num_leaves, min_data_in_leaf

Table 5. Homogenous classifiers.

Classifier	Description	Parameters
Bagging		
Random Forest (RF)	$\widehat{y} = mode(T_1\,(x)\,,\ T_2\,(x) \ldots\ T_n\,(x))$	$T_1\,(x)\,,\ T_2\,(x) \ldots\ T)_n\,(x)$ – Individual Decision Trees
Boosting		
Gradient Boosting	$\widehat{y} = \sum_{m=1}^{M} \gamma\, h_m\,(x)$	M – no. of weak learners γ - learning rate $h_m\,(x)$- weak classifier
Ada Boosting	$\widehat{y} = sign(\sum_{m=1}^{M} \alpha_{\,m} h_m\,(x))$	M – no. of weak learners $\alpha_{\,m}$- weight of weak learner $h_m\,(x)$- weak classifier
XG Boosting, CAT Boosting, Light GBM	$\widehat{y}^{(m)} = \widehat{y}^{(m-1)} + \gamma\, h_m\,(x))$	γ - learning rate $h_m\,(x)$- weak classifier

Table 6. Homogenous classifiers representation.

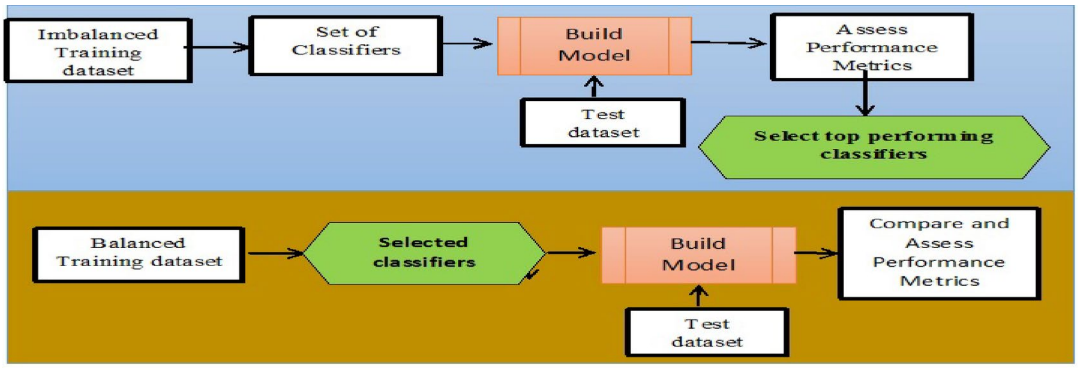

Fig. 8. Steps involved in pre-processing.

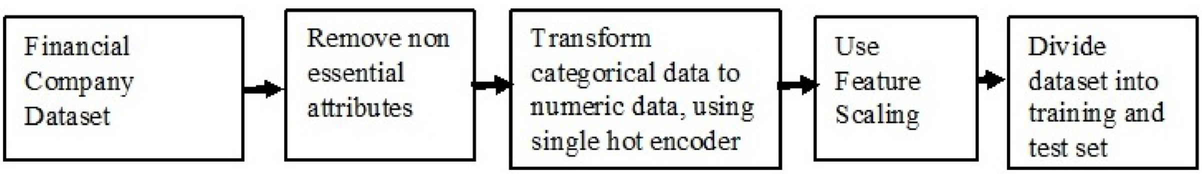

Fig. 9. Proposed work flow diagram.

Homogeneous classifiers' predictions are combined using voting, boosting, and bagging techniques. Voting determines the ultimate forecast for classification jobs while boosting combines weak learners to create strong ones. Bagging trains classifiers on bootstrap samples, averaging their predictions.

Table 5 shows the homogenous classifiers used for the study.

Table 6 shows the homogenous classifier representation and its description.

Figure 8 shows the pseudo-code representation of homogenous classifier functioning.

Proposed methodology

The problem pertains to binary classification, where data is categorized into two classes. The preprocessing step removes unnecessary attributes and converts all attributes into numerical data. Subsequently, the dataset is segmented as test sets and training sets. The training set is put into action to train multiple supervised classifiers, allowing the generation of models. The performance of these classifiers is then evaluated using appropriate metrics, with the test data serving as a benchmark. This comprehensive approach makes it possible to assess and compare the performance of classifiers, facilitating informed decision-making in binary classification. The dataset consists of 20% terminated clients and 80% non-terminated clients, highlighting an imbalance in data distribution.

The proposed flow diagram of the process is illustrated in Fig. 9.

The model-building process was conducted in two stages: using imbalanced and balanced data. With the imbalanced data, the performance of different classifiers was tested. Different classifiers, including single and homogenous classifiers, are considered for evaluating the performance. The process involved in preprocessing is shown in Fig. 8.

Data preprocessing

The dataset is examined for missing values. There are no missing values in the data set. The attributes 'RowNumber,' 'CustomerId,' and 'Surname' are removed from consideration as they are not relevant to the target attribute. The categorical attributes 'Geography' and 'Gender' are transformed into numerical data using one-hot encoding. Standard Scalar normalization is used to scale the data. Data is split into 80:20 proportions contributing to training and testing respectively.

Model building

In the first phase, a set of single classifiers was built and their performances were assessed. The performance of parametric and non-parametric classifiers is analyzed. In the second phase, homogenous classifier models were constructed and their performances were recorded. Hyperparameter tuning is done using GridSearch() method for certain single and homogenous classifiers. The GridSearch method is the hyperparameter tuning methodology that thoroughly explores a predetermined set of hyperparameters to identify the optimal combination that enhances model performance. With the suggested tuned parameters.

classifiers are constructed. The significance of features in hyper-tuned classifiers is plotted. For the churn data under study, the optimized classifier yielded appreciable performance. Best-performing classifiers are selected for further study.

Balanced dataset creation

Balanced Dataset is created using SMOTE (Synthetic Minority Over-sampling Technique). It works by generating synthetic samples of the minority class to balance the class distribution.

The steps followed in SMOTE are given below.

1. Determine which samples in the dataset belong to the minority class.
2. Choose a sample (x) at random from the minority class.
3. Using Euclidean distance or similar distance metric, find the selected sample x's k-nearest neighbors among the minority class samples (where k is usually a user-defined value).
4. Choose at random one of the k-nearest neighbors, let's say x_{nn}.
5. Using the following formula, create a synthetic sample called x_{new} along the line segment between x and x_{nn}.

$$x_{new} = x + \lambda .(x_{nn} - x)$$

6. Continue the above steps until the required quantity of synthetic samples is produced to achieve equilibrium.

The balanced dataset is subjected to selected classifiers and the metrics are compared. Balanced accuracy improved by introducing sampling techniques. In this study, balancing is done only on the training set to prevent bias in the trained model while leaving the test data.

Evaluation metric

The confusion matrix is a valuable tool for summarizing categorization results, providing clear insights into the model's performance in positive and negative scenarios. Various metrics are considered to evaluate the model's effectiveness, including accuracy, balanced accuracy, AUC, precision, recall, and F1-score. These evaluation metrics help gauge the model's performance across different dimensions, contributing to a comprehensive assessment of its classification capabilities.

When the classes in the dataset are balanced, or when each class has about the same number of occurrences, accuracy becomes a relevant indicator. When datasets are unbalanced, accuracy can be deceptive. For instance, a model could attain high accuracy by consistently predicting the majority class if one class is substantially more frequent than the other.

When working with unbalanced datasets that include underrepresented classes, balanced accuracy is especially helpful. By accounting for recall in each class, it offers a more illuminating assessment of the model's performance. In contrast to majority class dominance, balanced accuracy provides a more equitable evaluation of model performance across classes.

By averaging the recall (sensitivity) for each class, balanced accuracy corrects for unequal class distributions. It assigns equal weight to each class's performance. Balanced accuracy is computed as shown in Eq. (1).

$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right) \tag{1}$$

where $\frac{TP}{TP+FN}$—Recall for positive class and $\frac{TN}{TN+FP}$—Recall for negative class

Algorithm 2 explains the Pseudo-code of Homogenous Classifier.

```
Input:
 - Training data: X_train (features), y_train (labels)
 - Number of boosting rounds: M
 - Base learner (e.g., decision tree) parameters: base_params
 - Boosting-specific parameters (e.g., learning rate, regularization)
 Output:
 - Trained boosting classifier model: ensemble_model

function TrainBoostingClassifier(X_train, y_train, M, base_params, boosting_params):
   # Initialize weights for instances (for AdaBoost)
   InitializeWeights(y_train)

   # Initialize ensemble model
   ensemble_model = []

   # Initialize residual errors (for gradient boosting)
   residual_errors = InitializeResiduals(y_train)

   for m = 1 to M:
      # Train base learner (e.g., decision tree) on weighted data
      base_model = TrainBaseLearner(X_train, y_train, base_params)

      # Compute predictions of the base model
      predictions = base_model.predict(X_train)

      # Compute weighted error (for AdaBoost)
      weighted_error = ComputeWeightedError(y_train, predictions)

      # Compute contribution to the ensemble (weight calculation)
      alpha_m = ComputeModelWeight(weighted_error)

      # Update weights of instances (for AdaBoost)
      UpdateInstanceWeights(y_train, predictions, alpha_m)

      # Update residuals (for gradient boosting)
      UpdateResiduals(residual_errors, y_train, predictions)

      # Add base model to the ensemble with weight alpha_m
      ensemble_model.add(base_model, alpha_m)

   return ensemble_model

function Predict(ensemble_model, X_test):
   # Initialize final predictions
   final_predictions = 0

   for each model in ensemble_model:
      # Make predictions using each base learner
      predictions = model.predict(X_test)
```

Algorithm 2 Pseudo-code of homogenous classifier

Results and discussion
Single classifier performance on an imbalanced churn data
Figures 10 and 11 depict the Accuracy, AUC, precision, recall, and f1-score metrics of parametric and non-parametric classifiers under study on imbalanced churn data. According to analysis on Parametric Classifiers, the Multi-Layer Perceptron (MLP) is the top-performing model, with the highest metrics in every category,

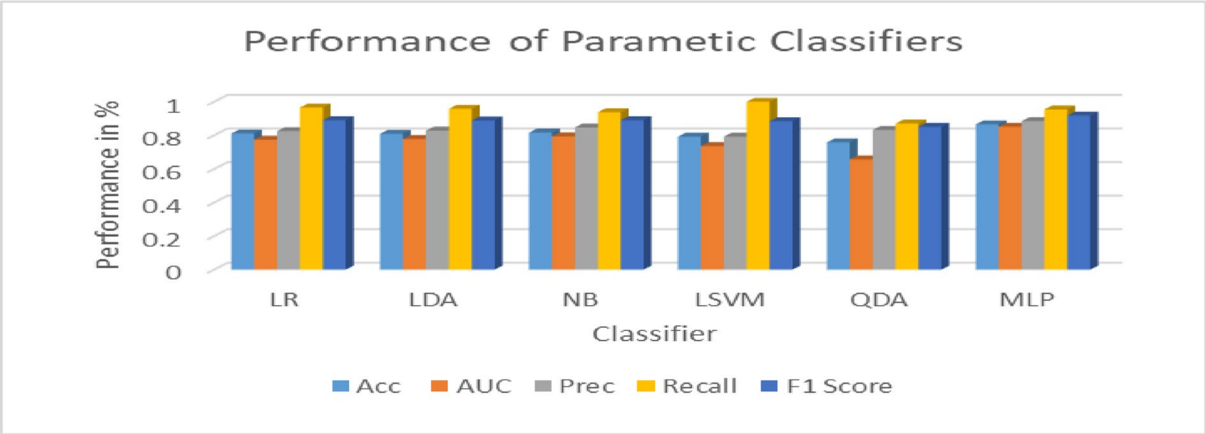

Fig. 10. Performance comparison of parametric classifiers.

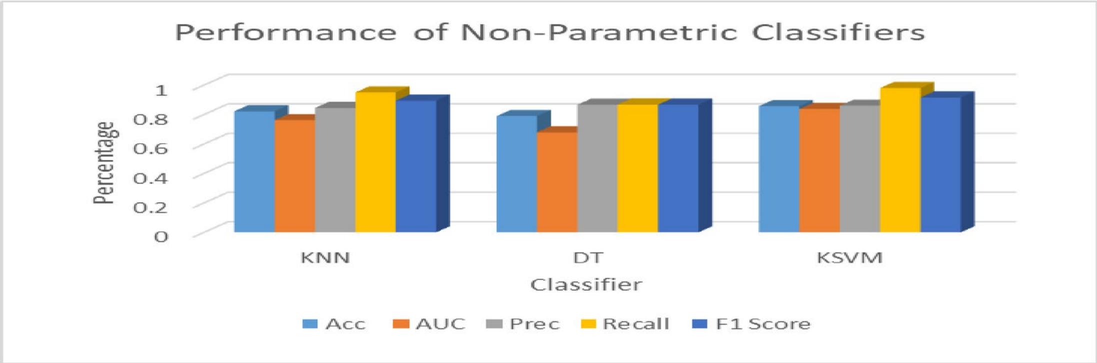

Fig. 11. Performance comparison of non-parametric classifiers.

including accuracy (86.52%), balanced accuracy (73.98%), AUC (85.09%), precision (88.44%), recall (95.45%), and F1 score (91.81%). Hence, MLP is the best deployment option, especially when efficiently managing unbalanced data. With strong balanced accuracy (64.65%), AUC (79.39%), and F1 score (88.99%), Naive Bayes (NB) also demonstrated strong performance, establishing itself as a dependable substitute for situations that need interpretability and simplicity. Both linear discriminant analysis (LDA) and logistic regression (LR) performed somewhat well, with strong recall but little capacity to correct for class imbalance. LSVM and QDA are considered as weak models as LSVM suffers from overfitting and QDA struggles in all metrics.

With the greatest accuracy (85.72%), balanced accuracy (68.51%), AUC (83.73%), and F1 score (91.57%), the Kernel Support Vector Machine (KSVM) performs better than the other models, according to the non-parametric classifier performance evaluation. It is also quite successful at accurately recognising affirmative cases while striking a good balance between precision and memory, exhibiting strong precision (85.95%) and extraordinarily high recall (97.98%). With a competitive F1 score of 89.42%, a balanced accuracy of 64.23%, and accuracy of 82.20%, the K-Nearest Neighbours (KNN) model comes in second. This suggests that the model performs reliably, especially when it comes to identifying positive cases (precision: 84.46%, recall: 95%). Although the Decision Tree (DT) model achieves balanced accuracy (67.90%) and reasonable accuracy (78.92%), it performs poorly in terms of AUC (67.90%) and F1 score (86.70%), indicating a propensity for overfitting.

Figure 12 shows the error in the classification of samples. According to the misclassification rate analysis, the Multi-Layer Perceptron (MLP) makes the most accurate predictions out of all the models, achieving the lowest rate (13.48%). With misclassification rates of 14.28% and 17.8%, respectively, Kernel Support Vector Machine (KSVM) and K-Nearest Neighbours (KNN) likewise exhibit good overall reliability. Conversely, models such as Decision Tree (DT) and Quadratic Discriminant Analysis (QDA) show greater misclassification rates of 21.08% and 24.2%, respectively, indicating their poorer predictive ability and possible overfitting problems and accuracy.

It is observed that among the parametric classifiers Multilayer Perceptron (MLP) performs well and Kernel SVM yields better results in the non-parametric category.

Table 7 shows the accuracy and balanced accuracy scores of single classifiers.

It is evident that balanced accuracy is not compatible with the accuracy score indicating the bias on the target value.

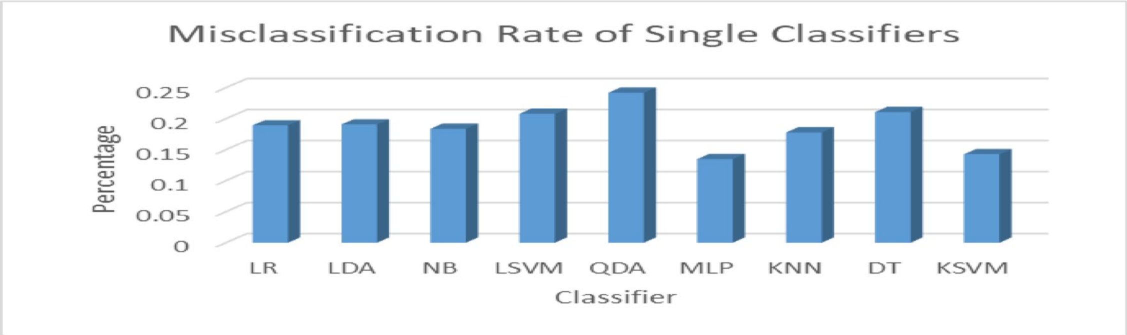

Fig. 12. The error rate of single classifiers on imbalanced churn data.

Classifier	Accuracy	Balanced accuracy
LR	0.811	0.593
LDA	0.809	0.600
NB	0.816	0.647
LSVM	0.792	0.500
QDA	0.758	0.600
MLP	0.865	0.740
KNN	0.822	0.642
DT	0.789	0.679
KSVM	0.857	0.685

Table 7. Accuracy and balanced accuracy scores of single classifiers.

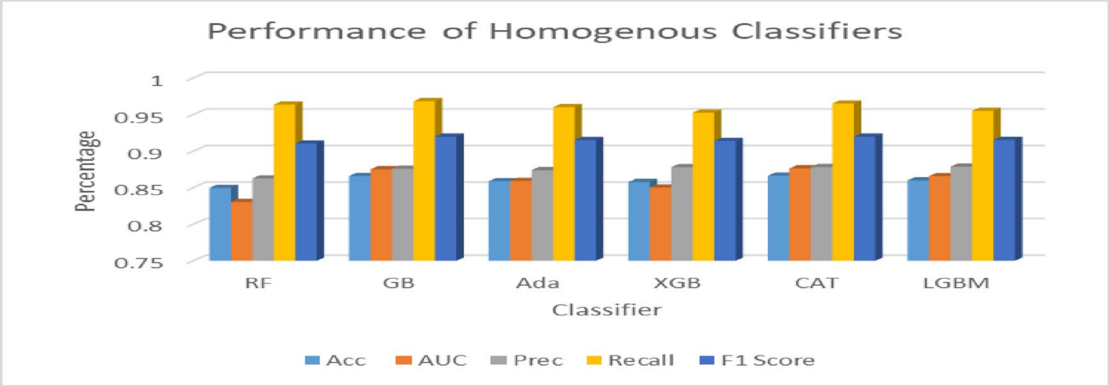

Fig. 13. Performance comparison of homogenous classifiers.

Homogenous classifier performance on an imbalanced churn data

Figure 13 depicts the Accuracy, AUC, precision, recall, and f1-score metrics of homogenous classifiers under study on imbalanced churn data. Tree-based models score well on all measures when evaluated, with CatBoost (CAT) and Gradient Boosting (GB) producing the best outcomes. The best-performing model was CAT, which had the highest balanced accuracy (72.78%), AUC (87.62%), and recall (96.52%), along with an outstanding F1 score (91.96%) and precision (87.82%). With strong predictive power demonstrated by high measures such as AUC (87.51%), recall (96.82%), and an F1 score (91.96%), GB comes in second. Strong reliability was demonstrated by models such as AdaBoost (Ada), LightGBM (LGBM), and XGBoost (XGB), which also performed well with balanced accuracy above 72% and F1 scores above 91%. Random Forest (RF) is still competitive despite having somewhat lower balanced accuracy (68.95%) and AUC (83.05%). On review, CAT and GB are the most successful types, with CAT being marginally more in classification metric.

Figure 14 shows the error in the classification of samples. CatBoost and Gradient Boosting are the most reliable homogeneous tree-based classifiers due to their low misclassification rates (13.36% and 13.4% respectively), while AdaBoost, LightGBM, and XGBoost have slightly higher error rates. Random Forest has the highest error rate of 15.04%.

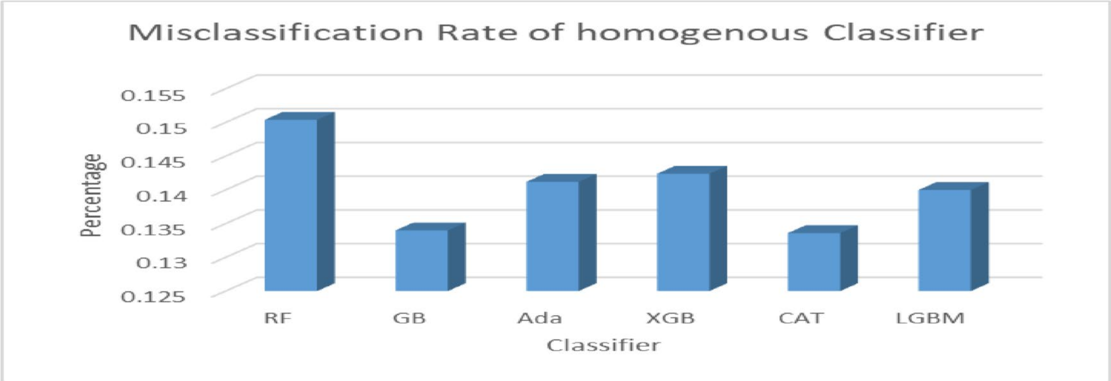

Fig. 14. The error rate of homogenous classifiers on imbalanced churn data.

Classifier	Parameters tuned
MLP	{'hidden_layer_sizes': [(s,), (s,)*2, (s,)*4, (s,)*6], 'solver': ['lbfgs', 'adam'], 'alpha': [0, 0.01, 0.1, 1, 10]}
GB	parameters = {'max_depth': [2, 3, 4, 6, 10, 15], 'n_estimators': [50, 100, 300, 500]}
XGB	(max_depth = 6, learning_rate = 0.1, n_estimators = 100, reg_lambda = 0.5, reg_alpha = 0, verbosity = 1, n_jobs = -1,).fit(train[features], train[target])
LGBM	{'num_leaves': 21, 'num_trees': 100, 'objective': 'binary', 'lambda_l1': 1, 'lambda_l2': 1, 'learning_rate': 0.1, 'seed': 1}

Table 8. Hyper parameters tuned for selected classifiers.

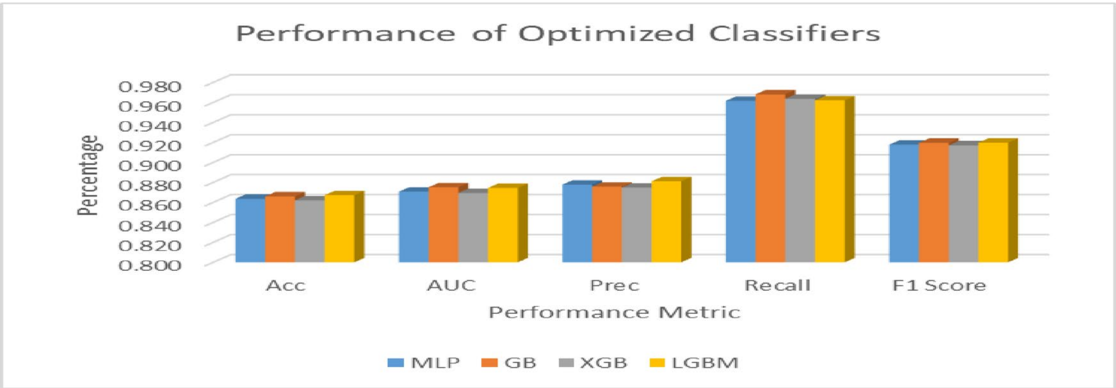

Fig. 15. Performance comparison of optimized classifiers.

Classifier performance after hyperparameter tuning on an imbalanced churn data

The performance of MLP and Gradient boosting classifiers are assessed after optimizing the parameters using the GridSearch() method. Table 8 lists the parameters tuned for the classifiers.

Figure 15 depicts AUC, precision, recall, and f1-score metrics of selected optimized classifiers under study on imbalanced churn data. All models achieve outstanding metrics, and the performance evaluation of the optimised classifiers shows very competitive results. With the highest accuracy (86.7%), balanced accuracy (73.4%), and F1 score (91.98%), as well as strong precision (88.11%) and recall (96.21%), LightGBM (LGBM) is clearly the best-performing model. These results demonstrate robust overall performance and balanced dataset handling. The next most dependable method for situations needing high sensitivity is gradient boosting (GB), which achieves high accuracy (86.6%), a slightly lower balanced accuracy (72.3%), an F1 score of 91.96%, and good recall (96.82%). With accuracy (86.4%), balanced accuracy (72.6%), and an F1 score of 91.78%, Multi-Layer Perceptron (MLP) likewise exhibits remarkable performance, demonstrating its efficacy in both precision and recall. Although it lags somewhat behind the others, XGBoost (XGB) provides competitive accuracy.

The performance metric score has improved after fine-tuning the parameters. The misclassification rate of classifiers is depicted in Fig. 16.

It is observed that LGBM has a lower error rate compared to other selected classifiers. Table 9 shows the accuracy and balanced accuracy scores of optimized classifiers.

Results show that the balanced accuracy score improved after hyperparameter tuning of classifiers. Figure 17 shows the significance of features revealed by the gradient boosting model. To determine the relevance of the

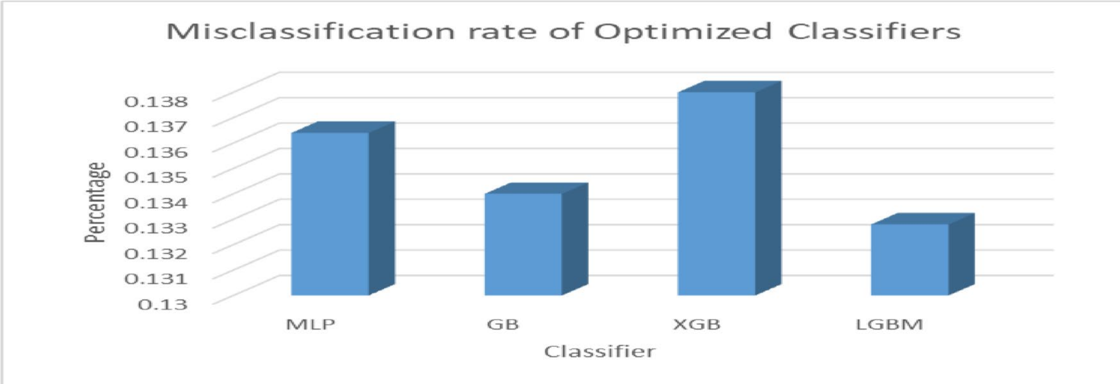

Fig. 16. The error rate of optimized classifiers on imbalanced churn data.

Classifier	Accuracy	Balanced acuracy
MLP	0.864	0.726
GB	0.866	0.723
XGB	0.862	0.719
LGBM	0.867	0.734

Table 9. Accuracy and balanced accuracy scores of optimzed classifiers.

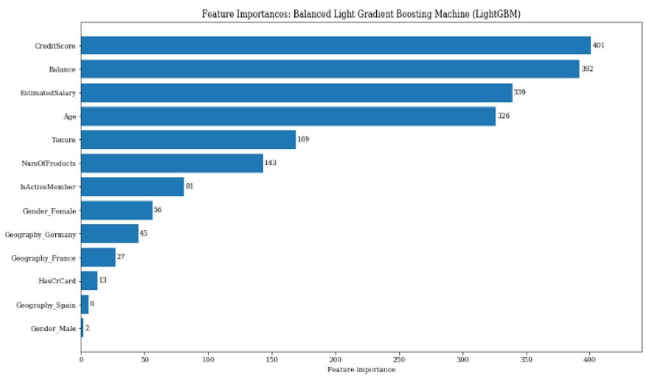

Fig. 17. Feature importance assessment on churn data.

Classifier	Parameters tuned
MLP	parameters = {'hidden_layer_sizes': [(s,), (s,)*2, (s,)*4, (s,)*6], solver': ['lbfgs', 'adam'], 'alpha': [0, 0.01, 0.1, 1, 10]}
GB	parameters = {'max_depth': [2, 3, 4, 6, 10, 15], 'n_estimators': [50, 100, 300, 500]}
Ada	{'base_estimator__max_depth':[i **for** i **in** range(2,11,2)], 'base_estimator__min_samples_leaf':[5,10], 'n_estimators':[10,50,250,1000],'learning_rate':[0.01,0.1]}
XGB	max_depth = 6, learning_rate = 0.1,n_estimators = 100, reg_lambda = 0.5, reg_alpha = 0, verbosity = 1, n_jobs = -1, tree_method = 'gpu_exact'
CAT	{ 'depth' : [4,5,6,7,8,9, 10],'learning_rate' : [0.01,0.02,0.03,0.04], 'iterations' : [10, 20,30,40,50,60,70,80,90, 100]}

Table 10. Hyper parameters tuned for selected classifiers.

feature, Gradient Boosting models use the weighted average of each feature's improvement in the loss function at each split during the tree-building process. Each feature's significance is assessed on its contribution towards lowering the performance of the model at each node. The more a feature helps, the higher its importance. Gradient Boosting identified key predictors of churn, including CreditScore, account balance, salary, and age. These insights enhance interpretability and guide targeted interventions.

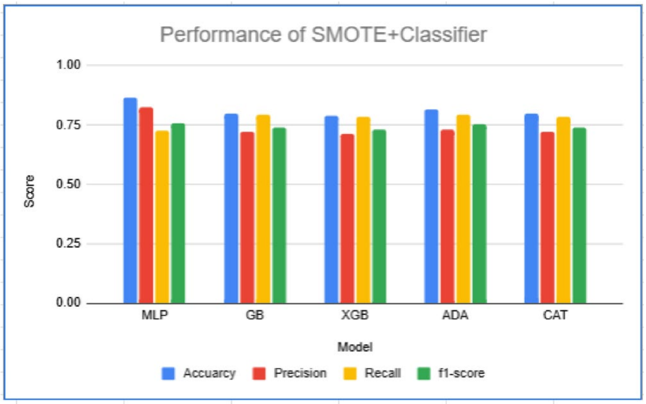

Fig. 18. Performance comparison of optimized classifiers on balanced churn data.

Classifier	Accuracy	Balanced accuracy
LR	0.801	0.650
GB	0.850	0.758
XGB	0.855	0.755
CAT	0.80	0.79
LGBM	0.804	0.755
ADA	**0.845**	**0.771**

Table 11. Accuracy and balanced accuracy scores of classifiers on balanced data.

Model	Accuracy	Balanced Accuracy	Weighted Precision	Weighted Recall	Weighted Recall
LR	0.811	0.593	0.669	0.373	0.442
LR + SMOTE	0.801	**0.650**	**0.788**	**0.801**	**0.793**
GB	0.866	0.723	0.813	0.577	0.663
GB + SMOTE	0.850	**0.758**	**0.849**	**0.850**	**0.849**
XGB	0.858	0.724	0.763	0.589	0.657
XGB + SMOTE	0.855	**0.755**	**0.850**	**0.855**	**0.755**
LGBM	0.860	0.726	0.774	0.590	0.662
LGBM + SMOTE	0.804	**0.755**	0.646	**0.804**	**0.716**
ADA	0.859	0.717	0.781	0.572	0.650
ADA + SMOTE	0.845	**0.771**	**0.850**	**0.845**	**0.847**

Table 12. Influence of SMOTE over the classifiers.

Classifier performance after hyperparameter tuning on a balanced churn data

The Churn dataset is balanced using SMOTE technique. Table 10 shows the parameters tuned using GridSearch() method. Figure 18 depicts AUC, precision, recall, and f1-score metrics of selected classifiers under study on balanced churn data.

The above comparative study selects the top-performing classifiers for further research. It is observed that the performance of boosting classifiers is appreciable. Multi-layer perceptron also performs well in churn prediction but computationally intensive. The selected classifier hyperparameters are tuned, and the classifier performance on balanced data is investigated.

Table 11 shows the accuracy and balanced accuracy scores of optimized classifiers on balanced data.

Balanced accuracy guarantees equitable assessment for all classes in imbalanced datasets, while weighted precision, recall, and F1-score offer more dependable performance metrics by taking class distribution into account. This helps prevent inaccurate inferences from unweighted measurements or standard accuracy. Table 12 shows the influence of SMOTE across the models.

The results show that, in all models, SMOTE (Synthetic Minority Over-sampling Technique) increases the balanced accuracy, weighted recall, and weighted F1-score, therefore proving its value in managing class imbalance. Although logistic regression (LR) without SMOTE performs badly in terms of balanced accuracy (0.593) and weighted recall (0.373), with SMOTE these values improve dramatically to 0.650 and 0.801, respectively. With

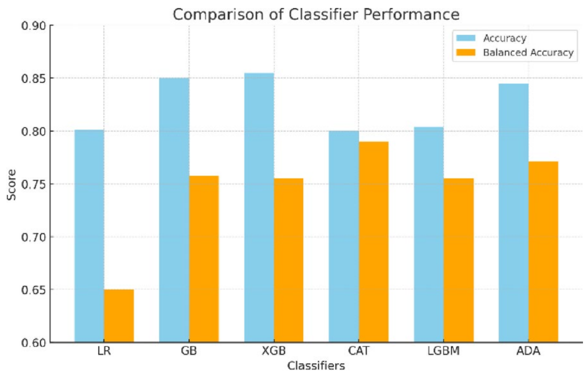

Fig. 19. Comparative analysis of performance of SMOTE+Classifiers.

GB + SMOTE the model improves the best balanced accuracy (0.758) and ADA + SMote having the highest at 0.777, Gradient Boosting (GB), XGBoost (XGB), LightGBM (LGBM), and AdaBoost (ADA) typically perform better than LR. Although weighted precision is still good across models, weighted recall is noticeably lower without SMOTE, hence underlining weak minority class predictions. Applying SMOTE produces GB + SMote and ADA + SMote as the best-performing models overall since they acquire high balanced accuracy (0.758 and 0.851) and F1-scores (0.849 and 0.847, hence more suited for unbalanced datasets.

Recommendation of optimal classifier for customer churn prediction

The results reveal that all classifiers attain relatively good accuracy, but balanced accuracy provides a clearer insight of their true performance on an imbalanced dataset. A comparative analysis of accuracy on boosting classifiers is shown in Fig. 19. While Logistic Regression (LR) has the lowest balanced accuracy (0.650), boosting-based models such as Gradient Boosting (GB), XGBoost (XGB), and AdaBoost (ADA) perform much better, with balanced accuracy values above 0.75, indicating superior handling of minority class predictions. The CatBoost (CAT) model achieves the best-balanced accuracy (0.79), implying it is the most effective at identifying both majority and minority classes fairly. LightGBM (LGBM) operates similarly to XGBoost, with a balanced accuracy of 0.755. Overall, boosting-based models beat logistic regression, and CatBoost looks to be the most balanced classifier in this scenario. However, if computational efficiency is a concern, AdaBoost is strong alternative.

AdaBoost, optimized with grid search, achieved an accuracy of 84.5% and balanced accuracy of 77.1%, making it the recommended model for churn prediction.

Conclusion

In conclusion, the growing number of private financial organizations has resulted in customers migrating away from traditional banking institutions. Therefore, financial organizations need help retaining consumers. Monitoring customer account health has become crucial in implementing preventive measures for retention. Machine learning algorithms present a promising solution for analyzing consumer status and reducing churn rates. This research conducted an extensive exploratory analysis of churn data to visualize the relationships across multiple dimensions. The study also investigated the effectiveness of classifiers of type boosting, linear, and nonlinear. The dataset used exhibited an imbalance in the churn/no churn category, which was addressed by employing SMOTE to balance the dataset. The classifiers were then put into action on the dataset on the target variable. The findings revealed a bias in the balanced dataset, which was used to assess the effect of performance measures on the imbalanced data. Notably, the Ada boosting classifier demonstrated effective performance on the balanced churn data utilized in the study. These insights highlight the importance of addressing data imbalance and applying suitable classifiers for prediction churn in the financial domain.

Future work

Future machine learning research on imbalanced datasets should investigate sophisticated sampling strategies like ADASYN and Borderline-SMOTE with hyperparameter optimisation to improve model performance. An ensemble of ensembles may provide higher forecast accuracy, and feature engineering and selection can increase efficiency. Further insights could be gained by looking into deep learning models like LSTMs and RNNs as well as cost-sensitive learning, especially in churn prediction jobs. Additionally, real-time deployment, monitoring, and enhancing model interpretability using techniques like SHAP or LIME should be prioritized. Further development in the use of balanced datasets for precise machine learning predictions may result from extending these findings to other domains such as fraud detection, medical diagnosis, or anomaly detection.

Data availability

The dataset used for the findings is available publicly at https://www.kaggle.com/shrutimechlearn/churn-modelling.

Received: 8 August 2024; Accepted: 2 May 2025
Published online: 09 May 2025

References

1. Aro, O. E. Data analytics as a driver of digital transformation in financial institutions. *World J. Adv. Res. Rev.* **24** (1), 1054–1072. https://doi.org/10.30574/wjarr.2024.24.1.3124 (2024).
2. Rajarajan., K. & Reddy, V. Priya. S, Anticipating customer churn in telecommunication using machine learning algorithms for customer retention. In *Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, 1–7. (IEEE, 2024). https://doi.org/10.1109/ICDCECE60827.2024.10548668
3. Gkonis, V. & Tsakalos, I. Deep Dive Into Churn Prediction in the Banking Sector: The Challenge of Hyperparameter Selection and Imbalanced Learning, *J. Forecast.*, 3194. https://doi.org/10.1002/for.3194 (2024).
4. Urrahman, D., Winanto, R. & Widyatama, T. Customer churn prediction in the case of telecommunication company using support vector machine (SVM) method and oversampling. *J. Stud. Res. Explor.* **2** (2), 65–79. https://doi.org/10.52465/josre.v2i2.253 (2024).
5. Mansoor, U., Sivakumar, V. & Jayabalan, M. Customer churn prediction in the banking sector on imbalance dataset. In *International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, 1–5 (IEEE, 2023). https://doi.org/10.1109/ICIICS59993.2023.10421738
6. Md, K. et al. A survey of methods for managing the classification and solution of data imbalance problem. https://doi.org/10.48550/ARXIV.2012.11870 (2020).
7. Hambali, M. A. & Andrew, I. Bank customer churn prediction using SMOTE: A comparative analysis. *Qeios* https://doi.org/10.32388/H82XTW (2024).
8. Rao, B., Xu, Y., Xiao, X., Hu, F. & Goh, M. Imbalanced customer churn classification using a new multi-strategy collaborative processing method. *Expert Syst. Appl.* **247**, 123251. https://doi.org/10.1016/j.eswa.2024.123251 (2024).
9. Bogaert, M. & Delaere, L. Ensemble methods in customer churn prediction: A comparative analysis of the State-of-the-Art. *Mathematics* **11** (5), 1137. https://doi.org/10.3390/math11051137 (2023).
10. Irawan, Y., Wahyuni, R., Ordila, R. & Herianto Comparative analysis of machine learning algorithms with SMOTE and boosting techniques in accuracy improvement. *Indones J. Comput. Sci.* **13** (5). https://doi.org/10.33022/ijcs.v13i5.4368 (2024).
11. Gurcan, D. & Soylu, A. Learning from Imbalanced Data: Integration of Advanced Resampling Techniques and Machine Learning Models for Enhanced Cancer Diagnosis and Prognosis,*Cancers*, **16**, 19, 3417, doi: https://doi.org/10.3390/cancers16193417. (2024).
12. Alizadeh, M., Zadeh, D. S., Moshiri, B. & Montazeri, A. Development of a customer churn model for banking industry based on hard and soft data fusion. *IEEE Access.* **11**, 29759–29768. https://doi.org/10.1109/ACCESS.2023.3257352 (2023).
13. Ljubičić, K., Merćep, A. & Kostanjčar, Z. Churn prediction methods based on mutual customer interdependence. *J. Comput. Sci.* **67**, 101940. https://doi.org/10.1016/j.jocs.2022.101940 (2023).
14. Ribeiro, H., Barbosa, B., Moreira, A. C. & Rodrigues, R. G. Determinants of churn in telecommunication services: a systematic literature review. *Manag Rev. Q.* **74** (3), 1327–1364. https://doi.org/10.1007/s11301-023-00335-7 (2024).
15. Manzoor, A., Atif Qureshi, M., Kidney, E. & Longo, L. A review on machine learning methods for customer churn prediction and recommendations for business practitioners. *IEEE Access.* **12**, 70434–70463. https://doi.org/10.1109/ACCESS.2024.3402092 (2024).
16. Durkaya Kurtcan, B. & Ozcan, T. Predicting customer churn using grey Wolf optimization-based support vector machine with principal component analysis. *J. Forecast.* **42** (6), 1329–1340. https://doi.org/10.1002/for.2960 (2023).
17. Ong, J. X., Tong, G. K., Khor, K. C. & Haw, S. C. Enhancing customer churn prediction with resampling: A comparative study. *TEM J.* 1927–1936. https://doi.org/10.18421/TEM133-20 (2024).
18. Harshini, A., Ramya, N. N. S., Teja, B. S., Sandeep, C. & Shareefunnisa, S. Improving customer churn prediction accuracy: A SMOTE-based approach. In *8th International Conference on Inventive Systems and Control (ICISC)*, 215–222 (IEEE, 2024). https://doi.org/10.1109/ICISC62624.2024.00044
19. Rahmi, N. A., Defit, S. Okfalisa enhancing classification performance: A study on SMOTE and ensemble learning techniques. In *International Conference on Future Technologies for Smart Society (ICFTSS)*, 63–68 (IEEE, 2024). https://doi.org/10.1109/ICFTSS61109.2024.10691339
20. Faruq, O., Ahammed, F., Mily, A. S. & Islam, A. Machine learning for enhanced churn prediction in banking: leveraging oversampling and stacking techniques. *Int. J. Sci. Res. Manag IJSRM*. **12** (09), 1434–1446. https://doi.org/10.18535/ijsrm/v12i09.ec03 (2024).
21. Soleiman-garmabaki, O. & Rezvani, M. H. Ensemble classification using balanced data to predict customer churn: a case study on the Telecom industry. *Multimed Tools Appl.* **83**, 44799–44831. https://doi.org/10.1007/s11042-023-17267-9 (2023).
22. Saha, S., Saha, C., Haque, M. M., Alam, M. G. R. & Talukder, A. ChurnNet: deep learning enhanced customer churn prediction in telecommunication industry. *IEEE Access.* **12**, 4471–4484. https://doi.org/10.1109/ACCESS.2024.3349950 (2024).
23. Ho Chi Minh City University of Economics and Finance, Vietnam & Tran, C. T. Banking customer churn prediction using random forest based on Smote and Adasyn approach. *J. Dev. Integr.* **no. 78**, 86–91. https://doi.org/10.61602/jdi.2024.78.11 (2024).
24. Feng, L. Research on customer churn intelligent prediction model based on borderline-SMOTE and random forest. In *IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 803–807 (IEEE, 2022). https://doi.org/10.1109/ICPICS55264.2022.9873702
25. Ako, R. E. et al. Effects of data resampling on predicting customer churn via a comparative Tree-based random forest and XGBoost. *J. Comput. Theor. Appl.* **2** (1), 86–101. https://doi.org/10.62411/jcta.10562 (2024).
26. Dhanawade, A., Mahapatra, B. & Bhatt, A. A smote-based churn prediction system using machine learning techniques. In *1st DMIHER International Conference on Artificial Intelligence in Education and Industry 4.0 (IDICAIEI)*, 1–7. (IEEE, 2023). https://doi.org/10.1109/IDICAIEI58380.2023.10406447
27. Pustokhina, I. V., Pustokhin, D. A., Nguyen, P. T., Elhoseny, M. & Shankar, K. Multi-objective rain optimization algorithm with WELM model for customer churn prediction in telecommunication sector. *Complex. Intell. Syst.* **9** (4), 3473–3485. https://doi.org/10.1007/s40747-021-00353-6 (2023).
28. Ly, N. N. Y. T. V. & Son, D. V. T. Churn prediction in telecommunication industry using kernel support vector machines. *PLOS ONE.* **17** (5), e0267935. https://doi.org/10.1371/journal.pone.0267935 (2022).
29. Khoh, W. H., Pang, Y. H., Ooi, S. Y., Wang, L. Y. K. & Poh, Q. W. Predictive churn modeling for sustainable business in the telecommunication industry: optimized weighted ensemble machine learning. *Sustainability* **15** (11), 8631. https://doi.org/10.3390/su15118631 (2023).
30. Haddadi, S. J., Farshidvard, A., Silva, F. D. S., Dos Reis, J. C. & Reis, M. D. S. Customer churn prediction in imbalanced datasets with resampling methods: A comparative study. *Expert Syst. Appl.* **246**, 123086. https://doi.org/10.1016/j.eswa.2023.123086 (2024).
31. Jiang, P., Liu, Z., Zhang, L. & Wang, J. Hybrid model for profit-driven churn prediction based on cost minimization and return maximization. *Expert Syst. Appl.* **228**, 120354. https://doi.org/10.1016/j.eswa.2023.120354 (2023).
32. Zhu, B., Qian, C., Vanden Broucke, S., Xiao, J. & Li, Y. A bagging-based selective ensemble model for churn prediction on imbalanced data. *Expert Syst. Appl.* **227**, 120223. https://doi.org/10.1016/j.eswa.2023.120223 (2023).

33. Tan, Y. L., Pang, Y. H., Ooi, S. Y., Khoh, W. H. & Hiew, F. S. Stacking ensemble approach for churn prediction: integrating CNN and machine learning models with catboost Meta-Learner. *J. Eng. Technol. Appl. Phys.* **5** (2), 99–107. https://doi.org/10.33093/jetap.2023.5.2.12 (2023).
34. Kate, P., Ravi, V. & Gangwar, A. FinGAN: chaotic generative adversarial network for analytical customer relationship management in banking and insurance. *Neural Comput. Appl.* **35** (8), 6015–6028. https://doi.org/10.1007/s00521-022-07968-x (2023).
35. Brito, J. B. G. et al. A framework to improve churn prediction performance in retail banking. *Financ Innov.* **10** (1), 17. https://doi.org/10.1186/s40854-023-00558-3 (2024).
36. Perišić, A. & Pahor, M. Clustering mixed-type player behavior data for churn prediction in mobile games. *Cent. Eur. J. Oper. Res.* **31** (1), 165–190. https://doi.org/10.1007/s10100-022-00802-8 (2023).
37. Mena, D., Coussement, K., De Bock, K. W., De Caigny, A. & Lessmann, S. Exploiting time-varying RFM measures for customer churn prediction with deep neural networks. *Ann. Oper. Res.* **339**, 1–2. https://doi.org/10.1007/s10479-023-05259-9 (2024).
38. Smaili, M. Y. & Hachimi, H. New RFM-D classification model for improving customer analysis and response prediction. *Ain Shams Eng. J.* **14** (12), 102254. https://doi.org/10.1016/j.asej.2023.102254 (2023).
39. Amin, A., Adnan, A. & Anwar, S. An adaptive learning approach for customer churn prediction in the telecommunication industry using evolutionary computation and Naïve Bayes. *Appl. Soft Comput.* **137**, 110103. https://doi.org/10.1016/j.asoc.2023.110103 (2023).
40. Abhinav Sudhir, T. & Ramnath, S. V. A random forest churn prediction model: an investigation of machine learning techniques for churn prediction and factor identification in the telecommunications industry. *Math. Stat. Eng. Appl.* **71** (4), 12662–12666. https://doi.org/10.17762/msea.v71i4.2434 (2022).
41. Sadeghi, M., Dehkordi, M. N., Barekatain, B. & Khani, N. Improve customer churn prediction through the proposed PCA-PSO-K means algorithm in the communication industry. *J. Supercomput.* **79** (6), 6871–6888. https://doi.org/10.1007/s11227-022-04907-4 (2023).
42. Kusnawi, K. et al. Leveraging various feature selection methods for churn prediction using various machine learning algorithms. *JOIV Int. J. Inf. Vis.* **8** (2), 897. https://doi.org/10.62527/joiv.8.2.2453 (2024).
43. Yüzer, D., Tinaz, Z. S., Yurtbaş, E. & Ayata, D. Advanced customer churn prediction using machine learning. In *32nd Signal Processing and Communications Applications* Conference *(SIU)*, 1–4 (IEEE, 2024). https://doi.org/10.1109/SIU61531.2024.10600858
44. Hao, M. & Research on customer churn prediction based on PSO-SA feature selection algorithm. In *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, 1055–1059 (IEEE, 2024). https://doi.org/10.1109/AINIT61980.2024.10581724
45. Mengash, D. A. et al. Archimedes optimization Algorithm-Based feature selection with hybrid Deep-Learning-Based churn prediction in telecom industries. *Biomimetics* **9** (1), 1. https://doi.org/10.3390/biomimetics9010001 (2023).
46. Ou, L. Customer Churn Prediction Based on Interpretable Machine Learning Algorithms in Telecom Industry. In *International Conference on Computer Simulation and Modeling, Information* Security *(CSMIS)*, 644–647. (IEEE, 2023). https://doi.org/10.1109/CSMIS60634.2023.00120
47. Šimović, P. P., Chen, C. Y. T. & Sun, E. W. Classifying the variety of customers' online engagement for churn prediction with a Mixed-Penalty logistic regression. *Comput. Econ.* **61** (1), 451–485. https://doi.org/10.1007/s10614-022-10275-1 (2023).
48. Seema & Gupta, G. Development of fading channel patch based convolutional neural network models for customer churn prediction. *Int. J. Syst. Assur. Eng. Manag.* **15** (1), 391–411. https://doi.org/10.1007/s13198-022-01759-2 (2024).
49. Saberi, Z., Hussain, O. K. & Saberi, M. Data-driven personalized assortment optimization by considering customers' value and their risk of churning: case of online grocery shopping. *Comput. Ind. Eng.* **182**, 109328. https://doi.org/10.1016/j.cie.2023.109328 (2023).
50. Li, D., Bai, X., Xu, Q. & Yang, D. Identification of customer churn considering difficult case mining. *Systems* **11** (7), 325. https://doi.org/10.3390/systems11070325 (2023).
51. Khaliq, A., Ajaz, S., Ali, A., Shakir, D. & Baig, K. From data to decisions: predictive machine learning models for customer retention in banking. *Asian Bull. Big Data Manag.* **4** (3). https://doi.org/10.62019/abbdm.v4i3.206 (2024).

Author contributions

S.R and S.P.J have taken care of Conceptualization, methodology. A.P.H and M.T.R did initial drafting. V.K.V has done the formal analysis, investigation and gathering resources. S.P.J and T.E.Y contributed to methodology and formal analysis. T.E.Y has done data curation and original draft preparation. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to T.E.Y.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.