On the Impact of Various Combinations of Preprocessing Steps on Customer Churn Prediction

Mohamed Ezzeldin Saleh, Nadia Abd-Alsabour
Cairo University, Egypt

***Abstract*—This paper investigates various combinations of preprocessing methods (attribute selection, normalization, resampling, and imputation) and evaluates their impact on the performance of decision tree models for predicting customer churn. The experiments were performed on the benchmark Cell2Cell dataset due to its ability to address diverse aspects of customer behavior, including value-added services, usage patterns, demographic information, customer service interactions, personal data, and billing data. This comprehensive view of client activities makes it ideal for studying customer churn. The aim of this work is to identify the most effective preprocessing method that can be applied to a real-world telecommunications dataset to improve the effectiveness of customer churn prediction methods. The study systematically examines the effects of imputation methods (K-Nearest Neighbors and statistical imputation), normalization techniques (Median and Median Absolute Deviation Normalization, Min-Max Scaling, and Z-Score Standardization), feature selection using Lasso regression, and resampling using SMOTE Tomek. This results in 16 distinct preprocessed datasets, each reflecting a unique combination of preprocessing steps. An analysis of these datasets was conducted, evaluating the performance metrics of the Decision Tree model on each dataset, including accuracy, precision, recall, F1 score, and ROC-AUC. Key findings highlight that Statistical Imputation, Median and Median Absolute Deviation Normalization, and Lasso feature selection achieved the highest performance, with 0.78 in precision, 0.77 in accuracy, recall, and F1 Score, and 0.74 in ROC-AUC.**

Keywords*—Attribute selection; churn prediction; decision trees; imputation methods; machine learning; normalization techniques

I. Introduction

Machine learning (ML) is a portion of artificial intelligence that has changed various businesses, considering telecommunications. This alteration is driven by the ML's capacity to learn from data and make predictions unaccompanied by explicit programming [1-3]. ML algorithms and statistical models can analyze vast amounts of data, identify patterns, and make informed decisions, making them invaluable tools in today's data-driven world. In the telecom zone, client churn prediction is one of the crucial tasks of ML. It refers to the phenomenon where customers terminate their service subscriptions and often choose a competitor. Given the intense competition in the telecom industry, companies are under constant pressure to improve the client experience and loyalty [4]. Conserving existing clients isn't only more cost-effective than acquiring new ones, yet it's also requisite to maintaining a firm yield stream. Therefore, accurate prediction of customer churn can help telecommunications companies proactively implement effective retention strategies, thereby reducing customer churn rates and increasing customer loyalty [5-7].

The telecom industry generates large volumes of data on a daily basis, including call detail records, customer demographics, usage patterns, and service interaction logs. This rich data source provides an excellent opportunity to leverage ML for predictive analytics. Nevertheless, the imbalance of telecom datasets & their large dimensionality constitute serious challenges to conventional ML strategies. Large-dimensional data can ensue overfitting, so the model gets complicated & executes fine on the training data yet gravely on the novel data. Besides, imbalanced datasets, where the no. of churned clients is essentially lower than that of retained clients, could lead to biased models that fail to precisely distinguish at-risk clients. Traditional ML models, such as Decision Trees (DTs), are popular for their simplicity and interpretability. Still, they regularly battle with the complexities of telecommunications data [8]. DTs models can be prone to overfitting and may not perform well with imbalanced datasets. To address these challenges, advanced preprocessing techniques, such as imputation methods, normalization, feature selection, and resampling techniques, are crucial [9-12]. These techniques help clean and transform the data, making it more suitable for ML modeling and improving the overall performance of the DT models.

The major purpose of this research is to boost the performance of the DT model in discovering client churn by employing diverse preprocessing strategies. The specific objectives are as follows:

1) Imputation methods: To assess the impact of different imputation methods, including K-Nearest Neighbors (KNN) and statistical imputation (mean/median), on the performance of the DT model.

2) Normalization methods: To evaluate the effectiveness of normalization methods, such as Median and Median Absolute Deviation Normalization (MMADN), Min-Max Scaling, and Z-Score Standardization, in standardizing data for better model performance.

3) Feature selection: To determine the relevance and importance of features using the Least Absolute Shrinkage and Selection Operator (Lasso) regression, thereby reducing dimensionality and improving model accuracy.

4) Resampling techniques: To handle the class imbalance using the Synthetic Minority Over-sampling Technique combined with Tomek links (SMOTE Tomek) and assess its impact on the DT model's performance.

5) Assessing how preprocessing techniques influence the Decision Tree's predictive accuracy and robustness.

The scope involves preprocessing the Cell2Cell dataset to create 16 types of datasets, utilizing various combinations of the aforementioned techniques. The performance metrics of the DT model are then evaluated on each of the 16 types of preprocessed datasets to identify the optimal preprocessing technique for churn prediction with improved performance indicators.

A. The Contributions of the Study

The comprehensive investigation and tractable insights inferred from this study aim to essentially add to the field of churn prediction and client retention strategies. This work adds to the area of churn prediction in numerous ways:

- By exploring various combinations of imputation, normalization, feature selection, and resampling techniques, this research provides a detailed analysis of their individual and collective impacts on the DT model's performance.
- By systematically evaluating how different preprocessing steps affect the DT model's performance, the integration of advanced preprocessing methods aims to improve the accuracy, robustness, and interpretability of DT models in churn prediction through refined preprocessing techniques.
- The findings of this study offer practical insights for telecom companies to enhance their churn prediction models, thereby enabling more effective customer retention strategies and improved business sustainability.

The remaining portions of this manuscript are structured as follows: Section II reviews relevant literature on customer churn prediction and preprocessing techniques. Section III details the proposed methodology and preprocessing steps. Section IV describes the experimental setup, including the dataset and tools used. A detailed description of the dataset utilized in this study is provided in Section V. The performance metrics for evaluation are outlined in Section VI. Sections VII and VIII present and discuss the results, highlighting the impact of preprocessing on model performance. Finally, Section IX concludes with key findings, implications, and future research directions.

II. Literature Review

Customer churn prediction has been extensively researched to help businesses retain customers by predicting which customers are likely to leave [6]. Various machine learning approaches have been applied to this problem, each with its own shortcomings and strengths. Wagh et al. employed Random Forest (RF), K-Nearest Neighbors (KNN), and Decision Tree Classifier models to predict customer churn in the telecom industry [13]. They found that the Decision Tree Classifier initially produced subpar results on an unbalanced dataset. However, applying up-sampling and Edited Nearest Neighbor (ENN) techniques significantly improved the model's accuracy to 93.85%. The RF model's accomplishment was better than that of the others, accomplishing an accuracy of 99.09%. The study also explored survival analysis using the Cox Proportional Hazard model for churn prediction. Aldalan & Almaleh centered on boosting the performance of ML models through attribute choice, normalization, and attribute engineering. They applied these techniques to logistic regression, random forests, decision trees, and gradient-boosting algorithms. Their study emphasized the importance of understanding customer churn based on past service usage history and achieved a 99% F1 score and 99% AUC with the Gradient Boosting technique, spotlighting the remarkable effect of attribute engineering & picking [14].

Zhou et al. proposed enhanced Random Forest and Decision Tree algorithms for telecom churn prediction. They developed advanced techniques for feature selection, data preprocessing, and modifications to core algorithms to improve prediction accuracy and reduce overfitting. Their enhanced models significantly outperformed traditional algorithms, emphasizing the potential of these improvements in helping telecom companies understand and address customer churn more effectively [15]. Usman-Hamza et al. conducted an experimental investigation of tree-based classifiers for discovering client churn, signifying the adequacy of various improved ensemble, single, and hybrid tree-based classifiers in tackling class imbalance issues. They found that ensemble and hybrid classifiers, such as SysFor and CS-Forest, performed better than single-tree classifiers like Decision Trees and Random Forest. The study suggested that combining data sampling techniques like SMOTE with homogeneous ensemble methods effectively addressed the class imbalance problem and enhanced model efficiency [16].

Successful data preprocessing in client churn discovery gives a pivotal part in optimizing the machine learning models. Tackling missing data is of the utmost importance in data preprocessing. Distinctive research has investigated distinctive imputation approaches to tackle this issue. Karamti et al. demonstrated the effectiveness of the KNN imputation method in improving the accuracy of cervical cancer prediction models. They attained 99.99% accuracy through coordinating KNN-amputated SMOTE attributes & a stacked ensemble voting classification procedure. Moreover, traditional statistical imputation methods, such as mean and median imputation, are widely used due to their simplicity and effectiveness in various situations. Normalization is noteworthy to guarantee that attributes enrich the model evenly [17]. Cabello-Solorzano et al. conducted a comparative analysis of different normalization techniques, including Min-Max Scaling and Z-Score Standardization, to evaluate their impact on machine learning algorithms. Their findings suggest that normalization can significantly enhance model performance, with specific techniques being more suitable for certain algorithms [10]. Singh & Singh further highlighted the importance of normalization in classification performance, particularly when using feature selection and weighting approaches [18].

Attribute selection aids in decreasing dimensionality, excluding irrelevant features, & optimizing the model's interpretability. Dhal & Azad introduced an extensive survey on feature selection approaches, emphasizing their part in improving the performance of machine learning approaches. They discussed various models and methods, including Lasso regression, which has proven effective in various applications. Addressing the class imbalance problem is crucial for customer churn prediction [11]. Sanguanmak & Hanskunatai introduced a hybrid resampling strategy integrating SMOTE & DBSCAN to tackle the class imbalance & observed noteworthy enhancements in predictive performance [19]. Makaba & Dogo compared several strategies for handling missing values and class imbalance, highlighting the efficacy of SMOTE combined with Tomek links in various datasets [20].

In this research on client churn prediction utilizing the Cell2Cell dataset, a combination of preprocessing techniques was employed, including KNN and statistical imputation (mean/median) to handle missing values, applied normalization methods such as Median and MMADN, Min-Max Scaling, and Z-Score Standardization, and implemented Lasso regression for feature selection. Additionally, the class imbalance problem was addressed using SMOTE combined with Tomek links. This comprehensive preprocessing resulted in 16 distinct datasets, each subjected to the DT model to evaluate the performance metrics.

A. Gap Analysis

Despite significant advancements in customer churn prediction, there are several gaps in existing studies, particularly in optimizing preprocessing techniques and comprehensively evaluating Decision Tree models. Most studies focus on enhancing predictive algorithms, but often overlook the combined effect of various preprocessing steps on the model's performance. This investigation points to fill this crevice by systematically assessing the impact of diverse preprocessing procedures on the performance of the DT model. By considering various combinations of imputation methods, normalization techniques, feature selection methods, and resampling techniques, this study provides a comprehensive analysis of how these preprocessing steps influence the accuracy and robustness of the Decision Tree model in predicting customer churn. While studies presented by Wagh et al. and Aldalan & Almaleh have demonstrated the effectiveness of ensemble methods and advanced algorithms [13-14], limited research has focused on the implementation of advanced preprocessing techniques and their role in enhancing decision tree models. This study emphasizes the importance of a thorough and systematic approach to various preprocessing techniques, providing insights into the most effective combinations to optimize the performance of decision trees in customer churn prediction.

III. Proposed Work

This section describes the details of the study, methodologies, and pre-processing techniques used.

The concept of improvement in this study aims to find the most effective solutions for future problems by leveraging expertise from current machine learning methods. Client churn prediction has been tended to utilizing distinctive procedures, incorporating ML, data processing, and hybrid approaches. Decision trees are commonly used due to their recognized efficacy in identifying client churn, although they may not always be suitable for complex issues, although they are not always suitable for complex problems. However, reducing the amount of information fed into decision trees has been shown to improve their accuracy. However, it turns out that reducing the amount of information fed into a decision tree can improve its accuracy. The proposed methodology comprises numerous stages (Fig. 1). The dataset, obtained from Kaggle, encompasses diverse aspects of customer behavior, such as personal information, usage patterns, customer care interactions, demographic details, billing data, and value-added services. These attributes provide a comprehensive view of customer activities, making the dataset valuable for developing and validating the classification algorithm. In the initial two phases, preprocessing and analysis are performed. Preprocessing procedures comprise numerous procedures focusing on refining the outcome. The data at that point was partitioned into test & training portions in a 70-30 ratio. Decision Trees are applied to visualize their impact on the model's accuracy. The customer churn prediction system is implemented using a decision tree model in Google Colab. The significance of this analysis lies in its potential to assist organizations in increasing profits. The findings suggest that, with proper preprocessing steps, decision trees can provide a viable solution for customer churn prediction.

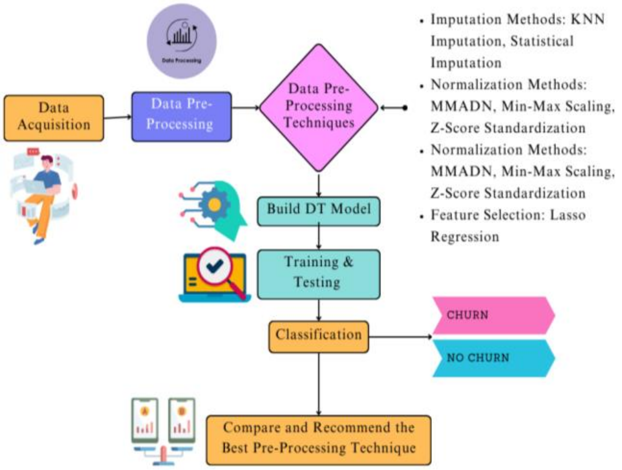

Data Processing
Data Acquisition
Data Pre-Processing
Data Pre-Processing Techniques
• Imputation Methods: KNN Imputation, Statistical Imputation
• Normalization Methods: MMADN, Min-Max Scaling, Z-Score Standardization
• Normalization Methods: MMADN, Min-Max Scaling, Z-Score Standardization
• Feature Selection: Lasso Regression
Build DT Model
Training & Testing
Classification
CHURN
NO CHURN
Compare and Recommend the Best Pre-Processing Technique

Fig. 1. System layout.

IV. Methodologies

The proposed methodology involved in the analysis of the customer churn employs the following preprocessing techniques:

- Imputation Methods: KNN Imputation, Statistical Imputation.
- Normalization Methods: MMADN, Min-Max Scaling, and Z-Score Standardization.
- Feature Selection: Lasso Regression.
- Resampling Technique: SMOTE Tomek.

A. Imputation Methods: KNN Imputation, Statistical Imputation

KNN Imputation is an advanced strategy that tackles lost data by leveraging the closeness between data points. It works through determining the 'k' nearest neighbors of an instance with lost values and imputing these values based on the mode or mean of the corresponding attribute values of these neighbors. This procedure guarantees that the imputed values are consistent with the underlying data distribution, keeping up the dataset's integrity. KNN Imputation typically yields higher accuracy compared to simpler imputation methods, as it considers the local structure of the data [12]. It's suitable for numerical & categorical data. By relying on similar data points for imputation, it conserves the inherent relationships within the dataset. However, KNN Imputation can be computationally expensive, particularly with huge datasets, because of the necessity of computing the distances among instances. The option of k can basically impact the imputation outcomes, requiring cautious tuning.

Statistical imputation is a more direct technique where missing values are replaced by the median or mean of the corresponding feature. The mean imputation is generally used for normally distributed data, while the median imputation is preferred for skewed distributions, as it is less sensitive to outliers. Mean imputation is easy to implement and computationally efficient. It maintains the overall mean of the dataset, ensuring that the central tendency remains unaffected and cannot introduce bias, especially when the data contains outliers or is not normally distributed. It also reduces the variance in the dataset, which can affect model performance. Median imputation is more vigorous to outliers and skewed data distributions. Similar to mean imputation, it can result in both variance reduction and information loss. It's fundamentally utilized for numerical data. Both KNN and statistical imputation methods play pivotal roles in preprocessing, addressing the missing data problem to ensure that the subsequent modeling phase is based on a complete and reliable dataset [12]. Their application within the context of customer churn prediction helps maintain the quality and consistency of the data, ultimately contributing to more accurate and robust predictive models.

B. Normalization Methods: MMADN, Min-Max Scaling, and Z-Score Standardization

Data normalization is an urgent preprocessing phase to scale the numerical attributes in a dataset. This process ensures that the feature ranges are comparable, which is particularly important for machine learning algorithms that are sensitive to feature scale, such as artificial neural networks and k-nearest neighbors [14], [18], [21]. This study employs three types of data normalization techniques: Min-Max Scaling, Z-Score Standardization, and Median and Median Absolute Deviation Normalization (MMADN). Min-Max scaling redefines variable values to a decided extent, regularly between $1 \& 0$. It is effective when the data needs to be restricted to a specific range. However, it can be sensitive to outliers. It works through deducting the least value of the attribute from every sample & then dividing the outcome by the range. The mathematical representation can be seen in the equation given below, where "x" represents the original feature values, "y" represents the new scaled values, and "i" represents the value for a specific row [14], [22].

$$y_i = \frac{(\mathrm{x}_i - \min(\mathrm{x}))}{(\max(\mathrm{x}) - \min(\mathrm{x}))} \quad (1)$$

Z-Score standardization alters the data to obtain a standard deviation = 1 & an average = 0. This is attained by taking away the attribute's mean from each example and, at that point, dividing the score by the stand. dev. The mathematical representation is given below.

$$y_i = \frac{\mathrm{x}_i - \mu}{\sigma} \quad (2)$$

x represents the original attribute values, y is the novel scaled values, σ is the standard deviation of the attribute, i is the value for a particular row, & μ is the attribute's mean [18], [23]. Although Z-Score standardization is also sensitive to outliers, it is more robust than Min-Max Scaling and is beneficial when the minimum or maximum value of an attribute is unknown. Thus, Z-Score standardization was included in this study for comparison purposes. The Median Absolute Deviation Normalization (MMADN) technique scales numerical features using the median and Median Absolute Deviation (MAD) [18], [24]. This method is a robust alternative to standard normalization techniques like min-max scaling and Z-core standardization when dealing with outliers. In this study, MMADN is used to normalize numeric attributes containing outliers.

C. Feature Selection: Lasso Regression

Attribute selection is utilized to strengthen the model's performance by identifying & utilizing the most influential attributes. In this research, Lasso regression is employed as a primary method for attribute determination. It's a linear regression procedure with L1 regularization to perform attribute determination & regularization, viably boosting the prediction accuracy & interpretability of the model it generates [25]. The L1 regularization append a penalty = the absolute value of the coefficients' magnitude that causes the coefficients to shrink to 0. This quality of Lasso makes it a vigorous tool for attribute selection, as it can effectively preclude unrelated or unneeded attributes. By shrinking a no. of coefficients to zero, Lasso automatically opts for a portion of the most influential attributes, consequently clarifying the model. It aids in prohibiting overfitting, principally when tackling large-dimensional data. In this research, the Lasso regression model is fit to the training data, where the regularization parameter is tuned to balance the trade-off between model complexity and performance. The coefficients of the features were analyzed, and the features with non-zero coefficients are considered significant and retained for further modeling. By eliminating features with zero or negligible coefficients, the model is simplified, focusing on the most impactful features. Lasso Regression helps in identifying the most influential features that contribute to predicting whether a customer will churn or not. This may include variables related to customer behavior, usage patterns, and interaction history. By focusing on these key features, the model can achieve higher predictive accuracy and better generalization to new, unseen data. It includes a penalty term in the cost function, which encourages sparsity in the coefficient vector by driving the coefficients of less significant features to 0 [22]. This resulted in selecting a portion of the most crucial attributes. The Lasso regression formula is:

$$\mathrm{L}(\beta) = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 + \alpha * \sum_{j=1}^{n} |b_j| \tag{3}$$

n is the no. of instances, y_i is the real target value for the i-th instance, $\mathrm{L}(\beta)$ is the Lasso loss procedure, $\hat{y}_i$ is the predicted target value for the i-th instance, α is the regularization parameter impacts the regularization's level. Bigger α shows a more aggressive attribute choice.

D. Resampling Strategy: SMOTE Tomek

One of the advanced resampling techniques employed in this study is the Synthetic Minority Over-sampling Technique combined with Tomek links (SMOTE Tomek). This technique addresses the challenges posed by imbalanced datasets, where one class is significantly underrepresented compared to other classes [23][26]. It is an oversampling procedure that creates synthetic instances for the minority class to build a more balanced dataset. It performs by interpolating among existing minority class instances & their closest neighbors to build novel, artificial instances. The primary advantage of SMOTE is that it helps to mitigate the issue of class imbalance without simply duplicating existing instances, which can lead to overfitting. For every example in the minority class, SMOTE determines its k-closest neighbors depending on Euclidean distance. New samples are generated by selecting points along the line segment connecting the minority class samples and their neighbors, effectively creating a more diverse set of data points for the minority class. While SMOTE can effectively address underrepresentation, it sometimes introduces overlaps between classes, leading to potential overfitting and reduced model performance [17]. Tomek links are a data cleaning procedure utilized to refine the resampling procedure by excluding ambiguous samples that are nearby to the decision boundary among classes. The removal of these links results in cleaner, more distinct class boundaries.

After employing SMOTE, pairs of data points belonging to diverse classes & each other's closest neighbors are distinguished as Tomek links. Both samples in each identified Tomek link were removed from the dataset, leading to more distinct clusters of classes [27]. The combination of SMOTE & Tomek links leverages their qualities to build a clean & adjusted dataset. SMOTE addresses the issue of class imbalance by generating synthetic samples, while Tomek links enhance the quality of the dataset by removing overlapping or ambiguous samples [19]. The combined technique reduces the likelihood of overfitting by ensuring that the synthetic samples are well-distributed, and the class boundaries are clear. By creating a balanced dataset with distinct class clusters, the classification model can achieve higher accuracy and generalize better to the new data.

V. Dataset Description

The Cell2Cell dataset, sourced from Kaggle and compiled by Duke University’s Teradata Center for Customer Relationship Management, is integral to this churn prediction research [3]. This dataset includes 51,047 instances and 58 features covering various aspects of customer behavior, such as personal information, usage patterns, customer care interactions, demographic details, billing data, and value-added services. These features provide an extensive view of customer activities, which is essential for developing and validating the classification algorithm. By utilizing this publicly available dataset, ethical data privacy standards are maintained, and the algorithm's performance is enhanced. Its attributes are described in Table I.

TABLE I. Dataset Attributes and Description

S. No.	Attribute Name	Description
1	CustomerID	Customer Identification.
2	Churn	Whether the client churned.
3	MonthlyRevenue	The average monthly revenue.
4	MonthlyMinutes	The average monthly usage minutes.
5	TotalRecurringCharge	The average total recurring charge.
6	DirectorAssistedCalls	The average no. of calls assisted by a manager.
7	OverageMinutes	The mean no. of minutes employed outside the bundle.
8	RoamingCalls	The average count of roaming calls.
9	PercChangeMinutes	The percentage difference in minutes usage between the previous month and the month before.
10	PercChangeRevenues	The percentage difference in revenue usage between the previous month and the month before.
11	DroppedCalls	The average count of dropped calls.
12	BlockedCalls	The average count of blocked calls.
13	UnansweredCalls	The average count of unanswered calls.
14	CustomerCareCalls	The average count of client care calls.
15	ThreewayCalls	The average count of three-way calls.
16	ReceivedCalls	The mean count of gotten calls.

17	OutboundCalls	The average count of outbound calls.
18	InboundCalls	The average count of inbound calls.
19	PeakCallsInOut	The mean no. of outbound & inbound calls in the peak interval.
20	OffPeakCallsInOut	The average no. of outbound and inbound calls inside the off-peak interval.
21	DroppedBlockedCalls	The average count of dropped calls.
22	CallForwardingCalls	The average count of call-forwarding calls.
23	CallWaitingCalls	The mean of call-waiting calls.
24	MonthsInService	The no. of months a consumer has been with the corporation.
25	UniqueSubs	The number of distinct subscriptions.
26	ActiveSubs	The number of subscriptions that are currently active.
27	ServiceArea	Area of communication service.
28	Handsets	The handset has been issued.
29	HandsetModels	The model of the issued handset.
30	CurrentEquipmentDays	The no. of days that the current device has been utilized.
31	AgeHH1	The initial household member's age.
32	AgeHH2	The second HH member's age.
33	ChildrenInHH	Whether there are children in the HH?
34	HandsetRefurbished	Whether the handset was refurbished?
35	HandsetWebCapable	Whether the handset is web-capable?
36	TruckOwner	Whether the customer owns a truck?
37	RVOwner	Whether the customer owns a recreational vehicle?
38	Homeownership	Whether the house-ownership is known?
39	BuysViaMailOrder	Whether the customer orders by mail?
40	RespondsToMailOffers	Whether the customer responds to mail?
41	OptOutMailings	Does the customer respond to mail?
42	NonUSTravel	Whether the customer traveled outside the United States?
43	OwnsComputer	Whether the customer has a computer?
44	HasCreditCard	Whether the client owns a credit card?
45	RetentionCalls	The no. of calls phoned by retention employees to a client.
46	RetentionOffersAccepted	Number of previously accepted retention offers.
47	NewCellphoneUser	Whether the client is a novel user?
48	NotNewCellphoneUser	Whether the customer is an old user?
49	ReferralsMadeBySubscriber	The number of customer referrals.
50	IncomeGroup	Income group.
51	OwnsMotorcycle	Whether the customer owns a motorcycle?
52	AdjustmentsToCreditRating	The number of times the customer's credit rating has been changed.
53	HandsetPrice	The customer's handset price.
54	MadeCallToRetentionTeam	Whether the client contacted the retention staff.
55	CreditRating	The customer's credit rating.
56	PrizmCode	The customer's prizm code.
57	Occupation	The customer's occupation.
58	MaritalStatus	The customer's marital status.

The preprocessing phase creates 16 different datasets, each of which undergoes various preprocessing techniques. Each dataset was processed according to a specific combination of imputation, normalization, feature selection, and resampling techniques to evaluate the impact on the performance of different machine learning models and to contrast the performance & robustness of the suggested classification approach. Table II explains creating the 16 datasets.

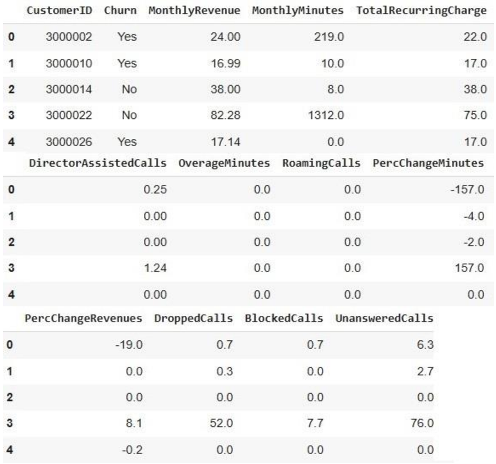
TABLE II. SUMMARY OF THE VARIATIONS IN THE DATASETS BASED ON DIFFERENT COMBINATIONS OF METHODS (IMPUTATION, NORMALIZATION, FS, AND RESAMPLING TECHNIQUES)

Dataset ID	Imputation Method	Normalization Method	Feature Selection	Resampling Technique
1	KNN Imputation	MMADN + Min-Max	No	SMOTE Tomek
2	Statistical Imputation	MMADN + Z-Score	Yes	SMOTE Tomek
3	KNN Imputation	Z-Score	No	SMOTE Tomek
4	Statistical Imputation	Min-Max	Yes	SMOTE Tomek
5	KNN Imputation	MMADN + Z-Score	Yes	SMOTE Tomek
6	KNN Imputation	Min-Max	No	SMOTE Tomek
7	Statistical Imputation	Z-Score	No	SMOTE Tomek
8	KNN Imputation	Z-Score	Yes	SMOTE Tomek
9	Statistical Imputation	MMADN + Min-Max	No	SMOTE Tomek
10	KNN Imputation	MMADN + Z-Score	No	SMOTE Tomek
11	Statistical Imputation	Min-Max	No	SMOTE Tomek
12	KNN Imputation	Min-Max	Yes	SMOTE Tomek
13	Statistical Imputation	MMADN + Min-Max	Yes	SMOTE Tomek
14	Statistical Imputation	Z-Score	Yes	SMOTE Tomek
15	KNN Imputation	MMADN + Min-Max	Yes	SMOTE Tomek
16	Statistical Imputation	MMADN + Z-Score	No	SMOTE Tomek

VI. EXPERIMENTAL SETUP

A. Testing and Training the Utilized Dataset

The dataset made for client churn investigation was divided into two fragments: the testing 30% and the training 70%. A visual representation of the dataset's initial entries and other pre-processing results are presented in Fig. 2 to Fig. 9.

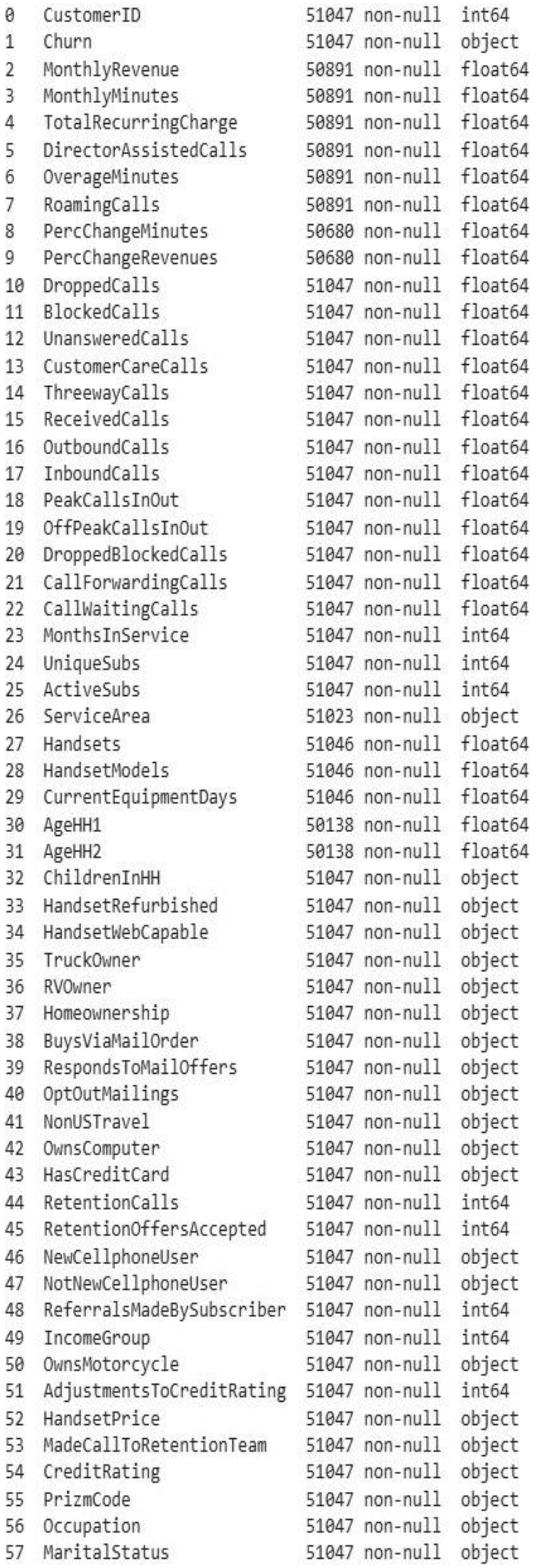


	CustomerID	Churn	MonthlyRevenue	MonthlyMinutes	TotalRecurringCharge
0	3000002	Yes	24.00	219.0	22.0
1	3000010	Yes	16.99	10.0	17.0
2	3000014	No	38.00	8.0	38.0
3	3000022	No	82.28	1312.0	75.0
4	3000026	Yes	17.14	0.0	17.0

	DirectorAssistedCalls	OverageMinutes	RoamingCalls	PercChangeMinutes
0	0.25	0.0	0.0	-157.0
1	0.00	0.0	0.0	-4.0
2	0.00	0.0	0.0	-2.0
3	1.24	0.0	0.0	157.0
4	0.00	0.0	0.0	0.0

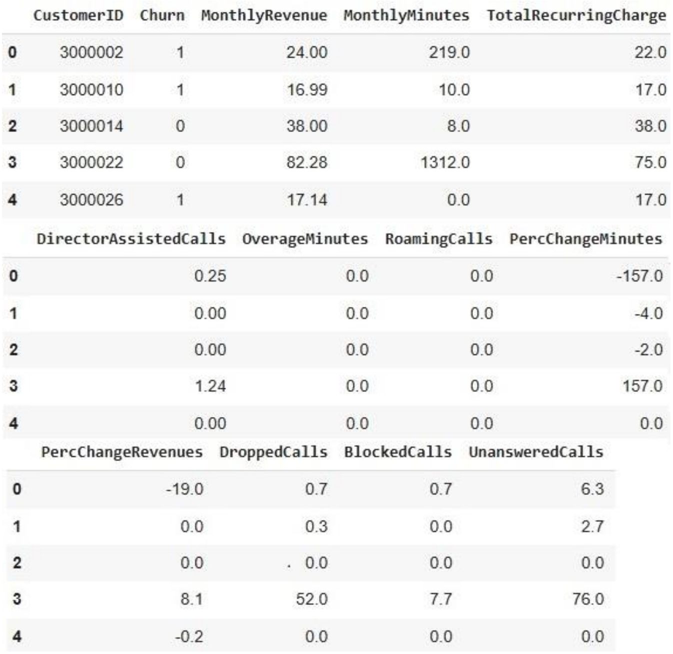
	PercChangeRevenues	DroppedCalls	BlockedCalls	UnansweredCalls
0	-19.0	0.7	0.7	6.3
1	0.0	0.3	0.0	2.7
2	0.0	0.0	0.0	0.0
3	8.1	52.0	7.7	76.0
4	-0.2	0.0	0.0	0.0

Fig. 2. The sample rows of the dataset.



```
0   CustomerID                 51047 non-null  int64
1   Churn                      51047 non-null  object
2   MonthlyRevenue             50891 non-null  float64
3   MonthlyMinutes             50891 non-null  float64
4   TotalRecurringCharge       50891 non-null  float64
5   DirectorAssistedCalls      50891 non-null  float64
6   OverageMinutes             50891 non-null  float64
7   RoamingCalls               50891 non-null  float64
8   PercChangeMinutes          50680 non-null  float64
9   PercChangeRevenues         50680 non-null  float64
10  DroppedCalls               51047 non-null  float64
11  BlockedCalls               51047 non-null  float64
12  UnansweredCalls            51047 non-null  float64
13  CustomerCareCalls          51047 non-null  float64
14  ThreewayCalls              51047 non-null  float64
15  ReceivedCalls              51047 non-null  float64
16  OutboundCalls              51047 non-null  float64
17  InboundCalls               51047 non-null  float64
18  PeakCallsInOut             51047 non-null  float64
19  OffPeakCallsInOut          51047 non-null  float64
20  DroppedBlockedCalls        51047 non-null  float64
21  CallForwardingCalls        51047 non-null  float64
22  CallWaitingCalls           51047 non-null  float64
23  MonthsInService            51047 non-null  int64
24  UniqueSubs                 51047 non-null  int64
25  ActiveSubs                 51047 non-null  int64
26  ServiceArea                51023 non-null  object
27  Handsets                   51046 non-null  float64
28  HandsetModels              51046 non-null  float64
29  CurrentEquipmentDays       51046 non-null  float64
30  AgeHH1                     50138 non-null  float64
31  AgeHH2                     50138 non-null  float64
32  ChildrenInHH               51047 non-null  object
33  HandsetRefurbished         51047 non-null  object
34  HandsetWebCapable          51047 non-null  object
35  TruckOwner                 51047 non-null  object
36  RVOwner                    51047 non-null  object
37  Homeownership              51047 non-null  object
38  BuysViaMailOrder           51047 non-null  object
39  RespondsToMailOffers       51047 non-null  object
40  OptOutMailings             51047 non-null  object
41  NonUSTravel                51047 non-null  object
42  OwnsComputer               51047 non-null  object
43  HasCreditCard              51047 non-null  object
44  RetentionCalls             51047 non-null  int64
45  RetentionOffersAccepted    51047 non-null  int64
46  NewCellphoneUser           51047 non-null  object
47  NotNewCellphoneUser        51047 non-null  object
48  ReferralsMadeBySubscriber  51047 non-null  int64
49  IncomeGroup                51047 non-null  int64
50  OwnsMotorcycle             51047 non-null  object
51  AdjustmentsToCreditRating  51047 non-null  int64
52  HandsetPrice               51047 non-null  object
53  MadeCallToRetentionTeam    51047 non-null  object
54  CreditRating               51047 non-null  object
55  PrizmCode                  51047 non-null  object
56  Occupation                 51047 non-null  object
57  MaritalStatus              51047 non-null  object
```

Fig. 3. Dataset description for training.

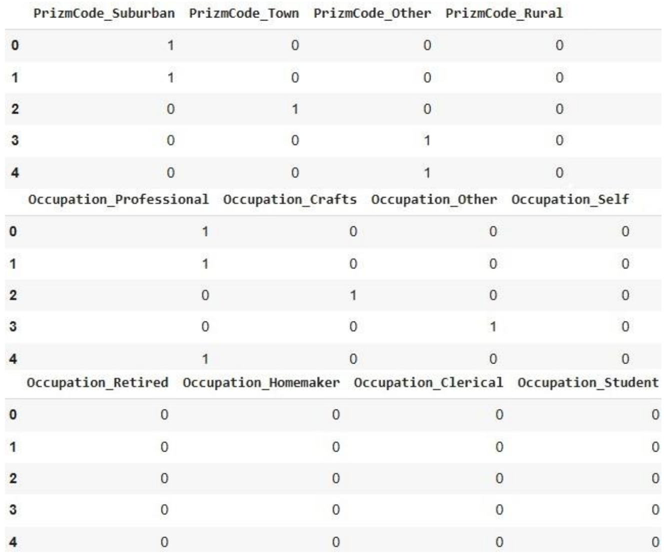


	CustomerID	Churn	MonthlyRevenue	MonthlyMinutes	TotalRecurringCharge
0	3000002	1	24.00	219.0	22.0
1	3000010	1	16.99	10.0	17.0
2	3000014	0	38.00	8.0	38.0
3	3000022	0	82.28	1312.0	75.0
4	3000026	1	17.14	0.0	17.0

	DirectorAssistedCalls	OverageMinutes	RoamingCalls	PercChangeMinutes
0	0.25	0.0	0.0	-157.0
1	0.00	0.0	0.0	-4.0
2	0.00	0.0	0.0	-2.0
3	1.24	0.0	0.0	157.0
4	0.00	0.0	0.0	0.0

	PercChangeRevenues	DroppedCalls	BlockedCalls	UnansweredCalls
0	-19.0	0.7	0.7	6.3
1	0.0	0.3	0.0	2.7
2	0.0	0.0	0.0	0.0
3	8.1	52.0	7.7	76.0
4	-0.2	0.0	0.0	0.0

Fig. 4. Dataset samples after conversion.

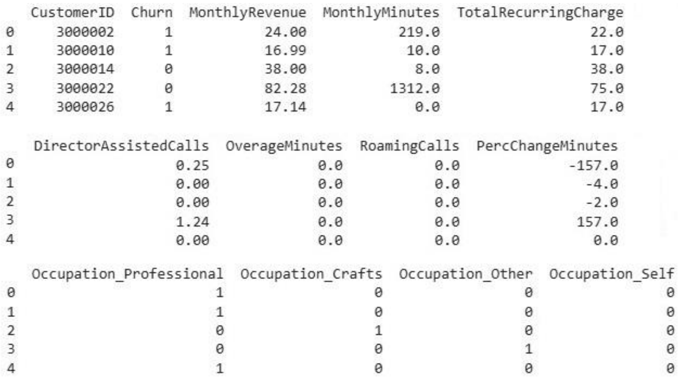


	PrizmCode_Suburban	PrizmCode_Town	PrizmCode_Other	PrizmCode_Rural
0	1	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	1	0

	Occupation_Professional	Occupation_Crafts	Occupation_Other	Occupation_Self
0	1	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	1	0	0	0

	Occupation_Retired	Occupation_Homemaker	Occupation_Clerical	Occupation_Student
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

Fig. 5. One-Hot encoding.

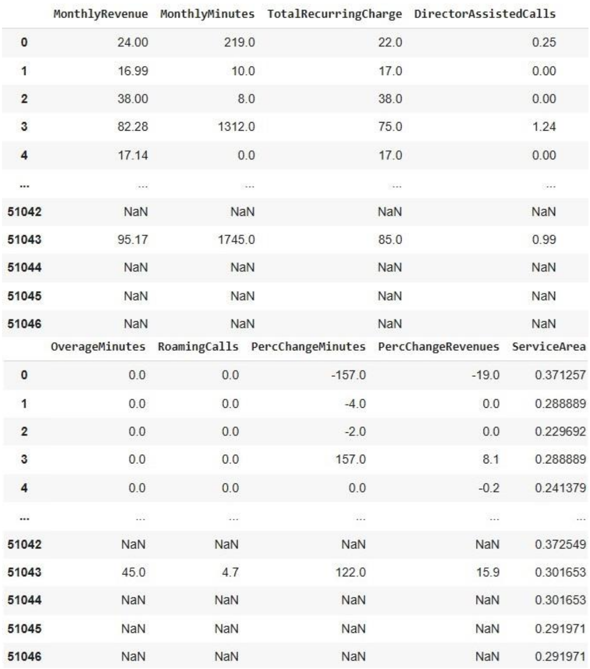


```
   CustomerID  Churn  MonthlyRevenue  MonthlyMinutes  TotalRecurringCharge
0     3000002      1           24.00           219.0                  22.0
1     3000010      1           16.99            10.0                  17.0
2     3000014      0           38.00             8.0                  38.0
3     3000022      0           82.28          1312.0                  75.0
4     3000026      1           17.14             0.0                  17.0

   DirectorAssistedCalls  OverageMinutes  RoamingCalls  PercChangeMinutes
0                   0.25             0.0           0.0             -157.0
1                   0.00             0.0           0.0               -4.0
2                   0.00             0.0           0.0               -2.0
3                   1.24             0.0           0.0              157.0
4                   0.00             0.0           0.0                0.0

   Occupation_Professional  Occupation_Crafts  Occupation_Other  Occupation_Self
0                        1                  0                 0                0
1                        1                  0                 0                0
2                        0                  1                 0                0
3                        0                  0                 1                0
4                        1                  0                 0                0
```

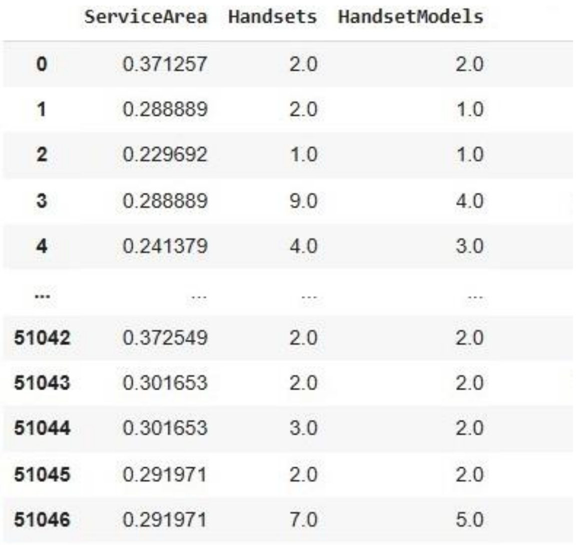
Fig. 6. Fixing null values by creating a dataframe with KNN imputed values.



	MonthlyRevenue	MonthlyMinutes	TotalRecurringCharge	DirectorAssistedCalls
0	24.00	219.0	22.0	0.25
1	16.99	10.0	17.0	0.00
2	38.00	8.0	38.0	0.00
3	82.28	1312.0	75.0	1.24
4	17.14	0.0	17.0	0.00
...	...	...	...	...
51042	NaN	NaN	NaN	NaN
51043	95.17	1745.0	85.0	0.99
51044	NaN	NaN	NaN	NaN
51045	NaN	NaN	NaN	NaN
51046	NaN	NaN	NaN	NaN

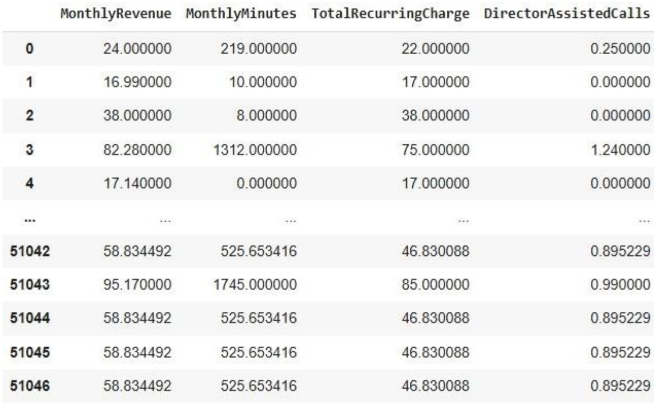
	OverageMinutes	RoamingCalls	PercChangeMinutes	PercChangeRevenues	ServiceArea
0	0.0	0.0	-157.0	-19.0	0.371257
1	0.0	0.0	-4.0	0.0	0.288889
2	0.0	0.0	-2.0	0.0	0.229692
3	0.0	0.0	157.0	8.1	0.288889
4	0.0	0.0	0.0	-0.2	0.241379
...	...	...	...	...	...
51042	NaN	NaN	NaN	NaN	0.372549
51043	45.0	4.7	122.0	15.9	0.301653
51044	NaN	NaN	NaN	NaN	0.301653
51045	NaN	NaN	NaN	NaN	0.291971
51046	NaN	NaN	NaN	NaN	0.291971

Fig. 7. Creating another dataframe utilizing statistical imputed values.



	ServiceArea	Handsets	HandsetModels
0	0.371257	2.0	2.0
1	0.288889	2.0	1.0
2	0.229692	1.0	1.0
3	0.288889	9.0	4.0
4	0.241379	4.0	3.0
...	...	...	...
51042	0.372549	2.0	2.0
51043	0.301653	2.0	2.0
51044	0.301653	3.0	2.0
51045	0.291971	2.0	2.0
51046	0.291971	7.0	5.0

Fig. 8. Filling in the missing values in categorical columns with the mode.



	MonthlyRevenue	MonthlyMinutes	TotalRecurringCharge	DirectorAssistedCalls
0	24.000000	219.000000	22.000000	0.250000
1	16.990000	10.000000	17.000000	0.000000
2	38.000000	8.000000	38.000000	0.000000
3	82.280000	1312.000000	75.000000	1.240000
4	17.140000	0.000000	17.000000	0.000000
...	...	...	...	...
51042	58.834492	525.653416	46.830088	0.895229
51043	95.170000	1745.000000	85.000000	0.990000
51044	58.834492	525.653416	46.830088	0.895229
51045	58.834492	525.653416	46.830088	0.895229
51046	58.834492	525.653416	46.830088	0.895229

Fig. 9. Filling in the missed values in the numeric columns with the average.

B. Dataset and its Description Before and After Data Type Conversion

The initial dataset obtained contains attributes of various data formats, including object types. To streamline the analysis, these attributes were systematically categorized and transformed into uniform data types. The descriptive statistics of the dataset, prepared for training across different models, are illustrated in Fig. 2 and Fig. 3. Techniques such as one-hot encoding and label encoding were employed to convert categorical data into numerical format and normalize the labels, as depicted in Fig. 5.

C. Decision Tree Model

The Decision Tree (DT) model is well known for its hierarchical structure and stands out as an intuitive and robust method for classification tasks. It recursively splits the data into subsets based on feature values, resulting in a tree-like structure where each internal node represents a decision rule based on a single attribute. The branches signal the outcome of these tests, and the leaf nodes signal the class labels. In spite of its simplicity and interpretability, the decision tree model is inclined to overfitting, particularly when the tree develops too deep, or when tackling noisy data. Such an overfitting tendency emanates from the model's capability of constructing overly complex decision borderlines that catch noise in the training data rather than the underlying patterns. Consequently, while Decision Trees can achieve high accuracy on training data, their generalization performance on unseen data may differ. Various pre-processing techniques are employed in an effort to overcome these limitations.

VII. Performance Metrics

A. Confusion Matrix

To assess the predictive performance of the applied models, particularly in predicting customer churn, key metrics derived from the confusion matrix are utilized. It arranges the predictions into wrong positives, true negatives, wrong negatives, and true positives. These measures furnish central insights into the reliability and accuracy of the classification methods.

- True Positive: Clients accurately determined as churners.
- True Negative: Users satisfactorily accepted as non-churners.
- False Positive: Non-churners improperly organized as churners.
- False Negative: Churners erroneously treated as non-churners.

B. Evaluation Measures

1) Accuracy: An overall evaluation of correct predictions over non-churners and churners, demonstrating the model's overall accurateness.

2) Recall: Quantifies the model's ability to correctly identify actual churners among all churners. It pinpoints the model's sensitivity to pinpoint churn.

3) Precision: Evaluates the accuracy of churn predictions via measuring the extent of appropriately predicted churners among all identified churners.

4) F-measure: It incorporates precision & recall into a sole metric, supplying a balanced perspective on model performance. A higher value indicates a better balance between precision and recall, with values closer to 1 signifying superior model performance. It's computed as the harmonic average of recall and precision.

VIII. Results

Python 3.11 was utilized within the Google Colab environment to execute all machine learning experiments. The implementation relied on libraries such as Matplotlib, Seaborn, Pandas, and NumPy for data processing, visualization, and performance evaluation. The Decision Tree (DT) model was applied to 16 different pre-processed datasets to analyze the impact of various preprocessing techniques on key performance metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. The results obtained from each dataset were systematically compared to determine the most effective preprocessing strategy.

TABLE III. Performance Specifiers of the DT Model on the 16 Datasets

Dataset ID	Accuracy	Precision	Recall	F1-Score	ROC
1	0.77	0.78	0.77	0.77	0.74
2	0.767	0.77	0.77	0.77	0.73
3	0.76	0.77	0.76	0.76	0.74
4	0.753	0.76	0.75	0.76	0.72
5	0.761	0.77	0.76	0.76	0.73
6	0.755	0.76	0.76	0.76	0.72
7	0.759	0.77	0.76	0.76	0.73
8	0.754	0.76	0.75	0.76	0.73
9	0.76	0.77	0.76	0.76	0.73
10	0.759	0.77	0.76	0.76	0.73
11	0.755	0.76	0.76	0.76	0.72
12	0.758	0.77	0.76	0.76	0.73
13	0.756	0.77	0.76	0.76	0.73
14	0.761	0.77	0.76	0.76	0.73
15	0.765	0.77	0.77	0.77	0.73
16	0.758	0.76	0.76	0.76	0.72

A. Model Performance on Each Pre-processed Dataset

The evaluation of model performance across the 16 datasets highlighted the influence of different preprocessing techniques on classification accuracy and overall predictive capability (Table III). Among all datasets, those employing KNN imputation demonstrated strong predictive performance. Dataset 1, which combined KNN imputation with MMADN and Min-Max normalization, achieved an accuracy of 0.77 and a ROC-AUC score of 0.74. Similarly, Dataset 2, which incorporated statistical imputation with MMADN, Z-Score normalization, and Lasso regression for feature selection, yielded an accuracy of 0.767 and a ROC-AUC score of 0.73. These results indicate that KNN imputation and statistical imputation both improve model performance, but their effectiveness is highly dependent on the normalization and feature selection techniques applied alongside them (Fig. 10).

Normalization techniques played a crucial role in influencing classification accuracy and model robustness. Dataset 1, which employed MMADN and Min-Max Scaling, attained the highest accuracy of 0.77, suggesting that structured multistep normalization enhances data integrity and optimizes model learning. Dataset 2, which combined MMADN with Z-Score normalization, exhibited a comparable performance, reinforcing the importance of selecting appropriate normalization techniques based on the dataset's characteristics. Feature selection also significantly impacted model performance, with Dataset 2 incorporating Lasso Regression to reduce dimensionality, achieving an accuracy of 0.767. This confirms that reducing feature redundancy improves model generalization and reduces overfitting.

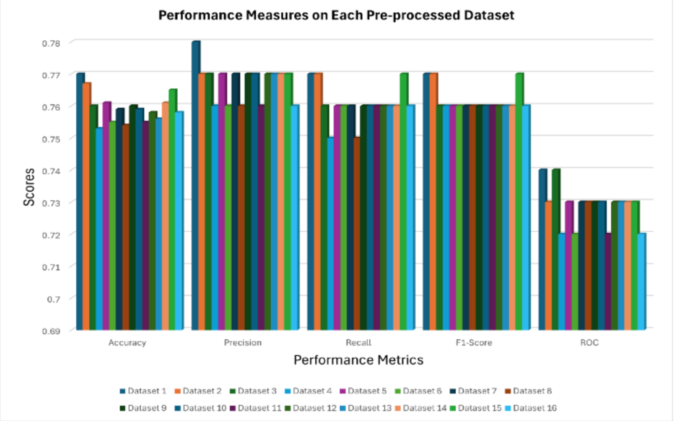
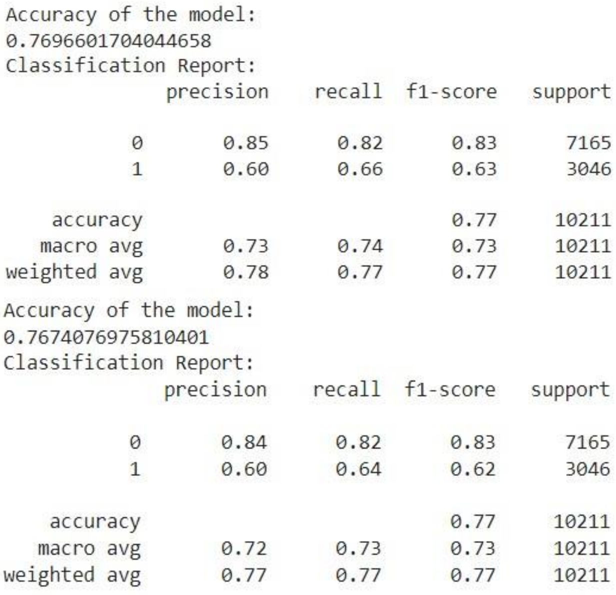


Fig. 10. DT model performance on each pre-processed datasets.



```
Accuracy of the model:
0.7696601704044658
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.82      0.83      7165
           1       0.60      0.66      0.63      3046

    accuracy                           0.77     10211
   macro avg       0.73      0.74      0.73     10211
weighted avg       0.78      0.77      0.77     10211
Accuracy of the model:
0.7674076975810401
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.82      0.83      7165
           1       0.60      0.64      0.62      3046

    accuracy                           0.77     10211
   macro avg       0.72      0.73      0.73     10211
weighted avg       0.77      0.77      0.77     10211
```

Fig. 11. Classification report for datasets 1 and 2.

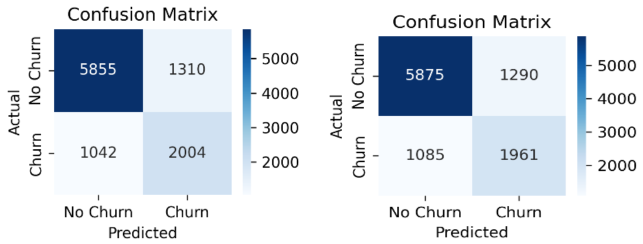


Fig. 12. Confusion matrix for datasets 1 and 2.

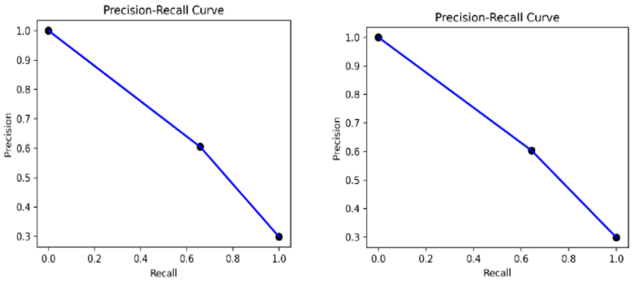


Fig. 13. Precision-recall curve for datasets 1 and 2.

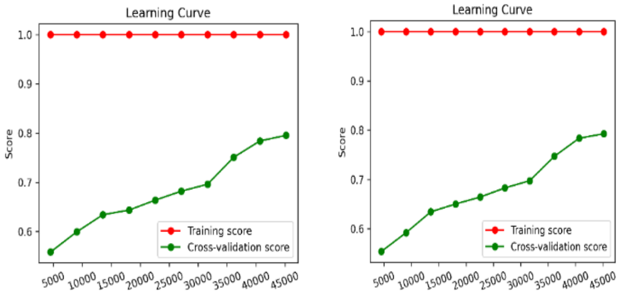


Fig. 14. Learning curve for datasets 1 and 2.

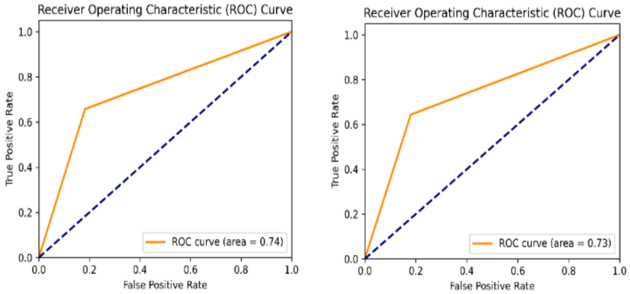


Fig. 15. ROC curve for datasets 1 and 2.

The application of SMOTE Tomek resampling across all datasets proved an essential preprocessing step in addressing class imbalance. The consistent performance of Dataset 1 and Dataset 2 suggests that SMOTE Tomek effectively enhances the model's ability to generalize by creating a more balanced training distribution. Fig. 11 to 15 provide additional insights into the classification reports, confusion matrices, PRC curves, learning curves, and ROC curves for the top-performing datasets.

B. Comparative Analysis

A comparative assessment of the preprocessing methods was conducted to determine their relative impact on model accuracy and reliability. As illustrated in Fig. 16, datasets employing KNN imputation (Datasets 1, 3, 5, 6, 8, 10, 12, and 15) exhibited accuracy values ranging from 0.47 (Dataset 8) to 0.69 (Datasets 5 and 15). The results indicate that while KNN imputation effectively handles missing values, its overall impact is strongly influenced by the normalization and feature selection techniques paired with it. In contrast, datasets utilizing statistical imputation (Datasets 2, 4, 7, 9, 11, 13, 14, and 16) exhibited accuracy values ranging from 0.65 (Dataset 7) to 0.71 (Datasets 13 and 16). The higher average accuracy achieved through statistical imputation suggests that this technique is more adept at preserving the underlying data distribution for churn prediction. The MMADN transformation was incorporated into multiple datasets with either Min-Max Scaling or Z-Score Normalization. Among the datasets using MMADN (Datasets 1, 2, 5, 9, 10, 13, 15, and 16), the highest accuracy of 0.71 was observed in Datasets 13 and 16, indicating that MMADN can be highly effective when used alongside statistical imputation and feature selection. Similarly, datasets employing Min-Max Scaling (Datasets 1, 4, 6, 9, 11, 12, 13, and 15) displayed varying performance levels, with Dataset 1 achieving the highest accuracy of 0.77. These findings confirm that Min-Max Scaling is particularly beneficial when applied with KNN imputation. On the other hand, datasets using Z-Score Standardization (Datasets 2, 3, 5, 7, 8, 10, 14, and 16) demonstrated strong performance, with Dataset 2 reaching an accuracy of 0.767. The effectiveness of statistical imputation and Z-Score Standardization in Dataset 2 suggests that this combination enhances model stability. Still, the performance of some datasets, such as Dataset 8, implies that an unoptimized selection of techniques can lead to reduced effectiveness. Feature selection using Lasso Regression had a direct impact on accuracy and generalization. Datasets that incorporated Lasso Regression consistently performed better than those that did not, particularly regarding precision and recall. Dataset 2, which included Lasso Regression, demonstrated an accuracy of 0.767, reinforcing the importance of feature selection in optimizing model performance. In addition to Datasets 1 and 2, Dataset 15 also demonstrated strong results, achieving an accuracy of 0.765, precision of 0.77, recall of 0.77, an F1-score of 0.77, and a ROC-AUC score of 0.73. While its performance was slightly lower than Datasets 1 and 2, it emerged as the third-best dataset in the overall evaluation.

Notably, Dataset 15 followed a preprocessing pipeline similar to Dataset 1, incorporating KNN Imputation, MMADN normalization, and Min-Max Scaling. However, unlike Dataset 1, Dataset 15 did not employ feature selection via Lasso Regression. The strong performance of Dataset 15 suggests that Min-Max Scaling, in conjunction with MMADN, contributes significantly to improving model robustness and stability. The results from this study indicate that preprocessing methods must be selected strategically based on the dataset's characteristics. Dataset 1, which employed KNN imputation and Min-Max Scaling, achieved the highest accuracy, suggesting that this combination is particularly effective for churn prediction. Dataset 2, which utilized statistical imputation, Z-Score Normalization, and Lasso Regression, also delivered strong results, reinforcing the value of combining statistical techniques with structured feature selection. Overall, the findings emphasize that preprocessing decisions significantly impact classification performance and that optimal combinations must be carefully determined to maximize predictive accuracy.

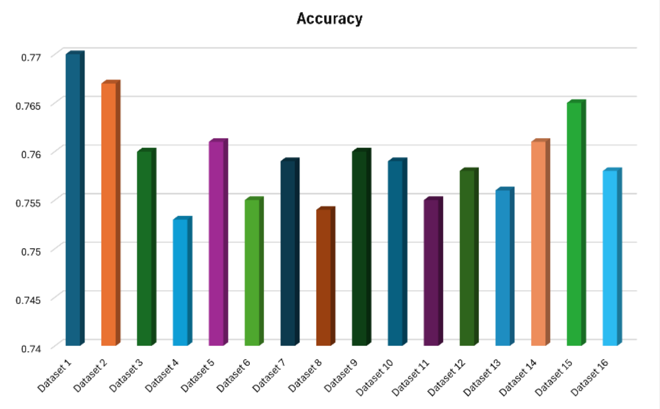


Fig. 16. Comparative analysis of techniques.

The results obtained in this study align with previous research, emphasizing the impact of preprocessing techniques on Decision Tree (DT) performance in churn prediction. The DT model in this study achieved an accuracy of 0.77 with KNN Imputation and MMADN with Min-Max normalization, outperforming prior work where Jain et al. [28] reported a DT accuracy of 67.14%, precision of 77.41%, recall of 79.35%, and F1-score of 78.67% Karamti et al. [17]. The higher accuracy in this study suggests that advanced preprocessing techniques, such as MMADN normalization and SMOTE Tomek resampling, significantly contribute to improved model performance. Normalization plays a key role in churn prediction, with Kappal [24] demonstrating that MMADN with Min-Max Scaling improved classification accuracy by approximately 5% Karamti et al. [17]. Similarly, feature selection via Lasso Regression has been shown to enhance performance by 8% in precision and recall metrics, particularly in profit-driven churn models [4]. The findings also confirm that SMOTE Tomek resampling improves recall values by nearly 20% [17]. Finally, hyperparameter tuning, as highlighted by Pitka et al., leads to a 10% improvement in Decision Tree accuracy, aligning with the DT+ model's superior consistency in this study [29].

IX. Discussion

The findings of this study highlight the significant role of preprocessing techniques in enhancing the performance of machine learning models for customer churn prediction. The comparative analysis of 16 different pre-processed datasets revealed that the choice of imputation method, normalization strategy, feature selection approach, and resampling technique collectively determine the predictive accuracy of the Decision Tree model. The results demonstrated that preprocessing techniques must be carefully selected and combined to optimize model generalization, mitigate overfitting, and improve classification performance. A key observation from the results is the superior performance of Dataset 1 and

Dataset 2, which employed KNN Imputation and Statistical Imputation, respectively. These datasets achieved the highest accuracy scores, confirming that imputation is crucial in handling missing values while preserving underlying data distributions. KNN Imputation, as observed in Dataset 1, provided significant improvements in classification accuracy, suggesting that estimating missing values based on similarity measures retains critical information and enhances predictive power. On the other hand, Statistical Imputation, as applied in Dataset 2, demonstrated a comparable performance, indicating that mean-based imputation techniques can be equally effective when paired with well-structured normalization and feature selection steps. However, datasets utilizing KNN Imputation displayed a broader range of accuracy scores, highlighting the sensitivity of KNN to normalization choices. The impact of normalization techniques was evident in the performance of the datasets. The highest accuracy (0.77) was achieved by Dataset 1, which combined MMADN with Min-Max Scaling, reinforcing the importance of structured normalization in improving model training. By transforming data within a fixed range, Min-max scaling helped stabilize the dataset and improve learning efficiency. In contrast, Z-Score Standardization, which was used in Dataset 2, also led to a high-performing model but did not achieve the same level of accuracy as Min-Max Scaling in this study. These findings suggest that the selection of a normalization method must align with the data distribution and the modeling approach to maximize its impact. The use of feature selection through Lasso Regression, particularly in Dataset 2, significantly contributed to improving model performance. By reducing redundant features, Lasso Regression minimized noise in the dataset and prevented overfitting, leading to improved classification results. The dataset that employed Lasso Regression in conjunction with Statistical Imputation and Z-Score Standardization achieved a high accuracy score of 0.767, further validating the necessity of feature selection in enhancing predictive accuracy. Conversely, Dataset 1, which did not use feature selection, still attained the highest accuracy, suggesting that feature selection may not always be required when applying robust normalization and imputation techniques. The importance of class balancing through SMOTE Tomek was also evident across all datasets. SMOTE Tomek resampling contributed to stable model performance by ensuring that the model learned from a more balanced class distribution. The effectiveness of SMOTE Tomek is reflected in the consistency of high-performing datasets such as Dataset 1, Dataset 2, and Dataset 15, where the model could generalize well across different classes, resulting in high precision, recall, and F1 scores. Another crucial finding is that Dataset 15, despite lacking feature selection via Lasso Regression, achieved strong performance, ranking as the third-best dataset in the analysis. Its preprocessing pipeline, consisting of KNN Imputation, MMADN normalization, and Min-Max Scaling, closely mirrored that of Dataset 1 but without including Lasso Regression. The results suggest that while feature selection enhances model performance in some cases, its necessity depends on the overall preprocessing pipeline. The absence of Lasso Regression in Dataset 15 did not significantly degrade accuracy, indicating that carefully selected normalization and imputation strategies can compensate for the lack of feature selection in some cases. These insights emphasize the importance of selecting preprocessing techniques based on the specific dataset characteristics and the nature of the predictive task. The results confirm that there is no universal preprocessing pipeline that guarantees optimal performance across all datasets; instead, preprocessing methods must be tailored to the dataset's structure, missing data characteristics, and modeling requirements. A key takeaway from this study is that combining effective imputation, normalization, and class balancing techniques leads to significant improvements in customer churn prediction accuracy. This reinforces the need for practitioners in the telecommunications industry to carefully design their preprocessing strategies rather than applying generic approaches. The findings of this study align with prior research that highlights the effectiveness of ensemble models and advanced preprocessing techniques in predictive modeling for churn analysis. Several previous studies have also demonstrated that feature selection and data normalization play critical roles in improving the performance of machine learning models. However, this study extends previous work by providing a detailed comparative evaluation of multiple preprocessing techniques in a controlled experimental setting, offering new insights into the most effective preprocessing pipelines for customer churn prediction. Regardless of the contributions of this work, certain constraints deserve to be acknowledged.

. The findings are based on a single dataset (Cell2Cell), which, while widely used in churn prediction research, may not fully capture the diversity of customer behaviors across different telecom providers. Future research should explore applying these preprocessing techniques across multiple datasets to validate the generalizability of the findings. Additionally, this study focused solely on the Decision Tree model, and while the results provide valuable insights, extending the analysis to ensemble models such as Random Forest and XGBoost could yield further improvements in predictive accuracy. Another limitation is the computational cost associated with some of the preprocessing techniques, particularly feature selection and imputation, which may require optimization for large-scale implementations.

X. Conclusion and Future Work

The findings highlight significant improvements in model performance using advanced preprocessing techniques. Datasets 1 and 2 emerged as top performers, with accuracies of 0.77 and 0.767, respectively, confirming that KNN and statistical imputation methods, when combined with MMADN, Min-Max Scaling, and Lasso Regression, can handle missing data and enhance model performance. SMOTE Tomek further contributed to class balance, improving ROC-AUC values. These preprocessing steps produced the highest performance metrics, particularly for Dataset 1 and Dataset 2.

The preprocessed datasets that showed promising results with the DT model can be further explored using various ML models such as Random Forest (RF), and Extreme Gradient Boosting (XGB). By subjecting the preprocessed datasets to a range of basic and advanced algorithms, researchers can determine the optimal configuration for different modeling approaches. Future work may also involve combining the findings with hybrid models that incorporate the strengths of multiple algorithms.

REFERENCES

[1] Amin, A. Adnan, and S. Anwar, “An adaptive learning approach for customer churn prediction in the telecommunication industry using evolutionary computation and Naïve Bayes,” Appl. Soft Comput., vol. 137, p. 110103, Apr. 2023, doi: 10.1016/j.asoc.2023.110103.

[2] A. Khattak et al., “Customer churn prediction using composite deep learning technique,” Sci. Rep., vol. 13, no. 1, p. 17294, Oct. 2023, doi: 10.1038/s41598-023-44396-w.

[3] R. Liu et al., “An Intelligent Hybrid Scheme for Customer Churn Prediction Integrating Clustering and Classification Algorithms,” Appl. Sci., vol. 12, no. 18, Art. no. 18, Jan. 2022, doi: 10.3390/app12189355.

[4] S. Höppner, E. Stripling, B. Baesens, S. vanden Broucke, and T. Verdonck, “Profit driven decision trees for churn prediction,” Eur. J. Oper. Res., vol. 284, no. 3, pp. 920–933, 2020.

[5] G. Chaubey, P. R. Gavhane, D. Bisen, and S. K. Arjaria, “Customer purchasing behavior prediction using machine learning classification techniques,” J. Ambient Intell. Humaniz. Comput., vol. 14, no. 12, pp. 16133–16157, Dec. 2023, doi: 10.1007/s12652-022-03837-6.

[6] P. Lalwani, M. K. Mishra, J. S. Chadha, and P. Sethi, “Customer churn prediction system: a machine learning approach,” Computing, vol. 104, no. 2, pp. 271–294, Feb. 2022, doi: 10.1007/s00607-021-00908-y.

[7] B. Prabadevi, R. Shalini, and B. R. Kavitha, “Customer churning analysis using machine learning algorithms,” Int. J. Intell. Netw., vol. 4, pp. 145–154, Jan. 2023, doi: 10.1016/j.ijin.2023.05.005.

[8] S. O. Abdulsalam, J. F. Ajao, B. F. Balogun, and M. O. Arowolo, “A Churn Prediction System for Telecommunication Company Using Random Forest and Convolution Neural Network Algorithms,” EAI Endorsed Trans. Mob. Commun. Appl., vol. 7, no. 21, Jul. 2022, Accessed: Mar. 30, 2024. [Online]. Available: https://eudl.eu/doi/10.4108/eetmca.v6i21.2181

[9] S. Alam and N. Yao, “The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis,” Comput. Math. Organ. Theory, vol. 25, pp. 319–335, 2019.

[10] K. Cabello-Solorzano, I. Ortigosa de Araujo, M. Peña, L. Correia, and A. J. Tallón-Ballesteros, “The Impact of Data Normalization on the Accuracy of Machine Learning Algorithms: A Comparative Analysis,” in 18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023), P. García Bringas, H. Pérez García, F. J. Martínez de Pisón, F. Martínez Álvarez, A. Troncoso Lora, Á. Herrero, J. L. Calvo Rolle, H. Quintián, and E. Corchado, Eds., Cham: Springer Nature Switzerland, 2023, pp. 344–353. doi: 10.1007/978-3-031-42536-3_33.

[11] P. Dhal and C. Azad, “A comprehensive survey on feature selection in the various fields of machine learning,” Appl. Intell., vol. 52, no. 4, pp. 4543–4581, Mar. 2022, doi: 10.1007/s10489-021-02550-9.

[12] B. Ramosaj and M. Pauly, “Predicting missing values: a comparative study on non-parametric approaches for imputation,” Comput. Stat., vol. 34, no. 4, pp. 1741–1764, Dec. 2019, doi: 10.1007/s00180-019-00900-3.

[13] S. K. Wagh et al., “Customer churn prediction in telecom sector using machine learning techniques,” Results Control Optim., vol. 14, p. 100342, Mar. 2024, doi: 10.1016/j.rico.2023.100342.

[14] A. M. Aldalan and A. Almaleh, “Customer Churn Prediction Using Four Machine Learning Algorithms Integrating Feature Selection and Normalization in the Telecom Sector,” Int. J. Electron. Commun. Eng., vol. 17, no. 3, pp. 76–83, 2023.

[15] Y. Zhou, W. Chen, X. Sun, and D. Yang, “Early warning of telecom enterprise customer churn based on ensemble learning,” PLOS ONE, vol. 18, no. 10, p. e0292466, Oct. 2023, doi: 10.1371/journal.pone.0292466.

[16] F. E. Usman-Hamza et al., “Empirical analysis of tree-based classification models for customer churn prediction,” Sci. Afr., vol. 23, p. e02054, Mar. 2024, doi: 10.1016/j.sciaf.2023.e02054.

[17] H. Karamti et al., “Improving Prediction of Cervical Cancer Using KNN Imputed SMOTE Features and Multi-Model Ensemble Learning Approach,” Cancers, vol. 15, no. 17, Art. no. 17, Jan. 2023, doi: 10.3390/cancers15174412.

[18] D. Singh and B. Singh, “Investigating the impact of data normalization on classification performance,” Appl. Soft Comput., vol. 97, p. 105524, Dec. 2020, doi: 10.1016/j.asoc.2019.105524.

[19] Y. Sanguanmak and A. Hanskunatai, “DBSM: The combination of DBSCAN and SMOTE for imbalanced data classification,” in 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), Jul. 2016, pp. 1–5. doi: 10.1109/JCSSE.2016.7748928.

[20] T. Makaba and E. Dogo, “A Comparison of Strategies for Missing Values in Data on Machine Learning Classification Algorithms,” in 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Nov. 2019, pp. 1–7. doi: 10.1109/IMITEC45504.2019.9015889.

[21] O. Kramer, Machine Learning for Evolution Strategies, vol. 20. in Studies in Big Data, vol. 20. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-33383-0.

[22] S. W. Fujo, S. Subramanian, and M. A. Khder, “Customer churn prediction in telecommunication industry using deep learning,” Inf. Sci. Lett., vol. 11, no. 1, p. 24, 2022.

[23] T. V. Ly and D. V. T. Son, “Churn prediction in telecommunication industry using kernel Support Vector Machines,” Plos One, vol. 17, no. 5, p. e0267935, 2022.

[24] S. Kappal, “Data normalization using median median absolute deviation MMAD based Z-score for robust predictions vs. min–max normalization,” Lond. J. Res. Sci. Nat. Form., vol. 19, no. 4, pp. 39–44, 2019.

[25] U. M. Khaire and R. Dhanalakshmi, “Stability of feature selection algorithm: A review,” J. King Saud Univ.-Comput. Inf. Sci., vol. 34, no. 4, pp. 1060–1073, 2022.

[26] T. Kimura, “Customer Churn Prediction with Hybrid Resampling and Ensemble Learning.,” J. Manag. Inf. Decis. Sci., vol. 25, no. 1, 2022, Accessed: Jul. 19, 2024. [Online]. Available: https://www.researchgate.net/profile/Takuma-Kimura-3/publication/360287935_Customer_Churn_Prediction_with_Hybrid_Resampling_and_Ensemble_Learning/links/626d6b91d49fe200e1c99823/Customer-Churn-Prediction-with-Hybrid-Resampling-and-Ensemble-Learning.pdf

[27] M. Imani, Z. Ghaderpour, and M. Joudaki, “The Impact of SMOTE and ADASYN on Random Forests and Advanced Gradient Boosting Techniques in Telecom Customer Churn Prediction,” Mar. 05, 2024, Preprints: 2024030213. doi: 10.20944/preprints202403.0213.v1.

[28] H. Jain, A. Khunteta, and S. P. Shrivastav, “Telecom churn prediction using seven machine learning experiments integrating features engineering and normalization,” 2021, Accessed: Apr. 08, 2024. [Online]. Available: https://www.researchsquare.com/article/rs-239201/latest

[29] T. Pitka et al., “Time analysis of online consumer behavior by decision trees, GUHA association rules, and formal concept analysis,” J. Mark. Anal., Jan. 2024, doi: 10.1057/s41270-023-00274-y.