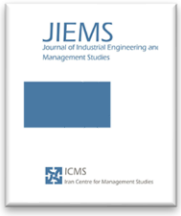
Vol. 10, No.1, 2023, Pages 141-157

JIEMS
Journal of Industrial Engineering and Management Studies

journal homepage: www.jiems.icms.ac.ir



An interpretable machine learning framework for customer churn prediction: a case study in the telecommunications industry

Mohammad Javad Jafari [1], Mohammad Jafar Tarokh *[2], Paria Soleimani [3]

[1] Department of Information Technology Management, Science and Research Branch, Islamic Azad University, Tehran, Iran.

[2] Department of Industrial Engineering, Khajeh Nasir University of Technology, Tehran, Iran.

[3] Department of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran.

Received: Jan 2023-02/ Revised: April 2023-22/ Accepted: July 2023-06

Abstract

Customer churn prediction has been gaining significant attention due to the increasing competition among mobile service providers. Machine learning algorithms are commonly used to predict churn; however, their performance can still be improved due to the complexity of customer data structure. Additionally, the lack of interpretability in their results leads to a lack of trust among managers. In this study, a step-by-step framework consisting of three layers is proposed to predict customer churn with high interpretability. The first layer utilizes data preprocessing techniques, the second layer proposes a novel classification model based on supervised and unsupervised algorithms, and the third layer uses evaluation criteria to improve interpretability. The proposed model outperforms existing models in both predictive and descriptive scores. The novelties of this paper lie in proposing a hybrid machine learning model for customer churn prediction and evaluating its interpretability using extracted indicators. Results demonstrate the superiority of clustered dataset versions of models over non-clustered versions, with KNN achieving a recall score of almost 99% for the first layer and the cluster decision tree achieving a 96% recall score for the second layer. Additionally, parameter sensitivity and stability are found to be effective interpretability evaluation metrics.

Keywords: Machine Learning; Customer Churn Prediction; Interpretability; Clustering; Classification.

Paper Type: Original Research

1. Introduction

Customer is the most crucial business asset in an exceedingly dynamic and competitive environment (Coussement et al., 2017). They leave their current service providers because of dissatisfaction with the service or receiving better offers from other service providers, a phenomenon known as customer churn. Thus, predicting customer churn is vital to various businesses as it can help predict changes in revenue and to address the issue by proposing more attractive products in hope of reducing the churn rate (Vahidi Farashah et al., 2021). It makes sense to try to retain current customers as it is usually a lot cheaper than the cost of acquiring new ones. So, identifying customers who were most likely to leave businesses is vital; so that accommodations would be proposed to prevent this behavior. In telecommunications industry proposing customer retention plans is a common way to achieve this goal. Therefore, companies employ intelligent methods to anticipate and provide customer retention solutions. A growing body of the literature indicates that these solutions and suggestions lead to lower companies' costs (Mardani et al., 2015). Machine learning is one of the most commonly used methods for predicting customer churn (Umayaparvathi & Iyakutti, 2016). Achieving improved performance frequently entails escalating model complexity, leading to a reduction in interpretability. Indeed, several studies have pointed out the lack of interpretability of current machine learning-based methods for customer churn management (Došilović et al., 2018; Moosavi-Dezfooli et al., 2017; Nguyen et al., 2015; Szegedy et al., 2013). They noted that the complexity of some machine learning algorithms can make it difficult to understand the reasoning behind their predictions. These findings underline the importance of developing an interpretable machine learning framework for customer churn management to improve the transparency and reliability of these models.

*Corresponding Author: mjtarokh@kntu.ac.ir

The concept of machine learning interpretability pertains to the capacity to comprehend and explain the reasoning behind the judgments rendered by a machine learning model. In other words, interpretability of a model is critical in ensuring that its predictions are reliable and trustworthy. Interpretability is particularly important in areas such as health, marketing, or automated driving, where ethical issues and justice are naturally raised (Lipton, 2016). The aim of this study is to explore the issue of machine learning interpretability and provide insights into how this aspect of algorithm design can be optimized. To achieve this, two primary research questions are investigated: Firstly, the indicators of machine learning interpretability that are currently in use are identified and examined. Secondly, A method for improving the interpretability of case study machine learning model was proposed, building on the existing literature and empirical research. The use of machine learning algorithms in customer churn prediction is not new. However, most studies focus on improving the performance of classification algorithms without considering the model's interpretability. This paper addresses this gap by proposing a novel interpretable machine learning framework for customer churn prediction. The framework's interpretability ensures that managers can make informed decisions based on the model's predictions. Additionally, the framework's use of clustering techniques is expected to enhance the predictive power of customer churn prediction models, leading to better customer retention strategies. The objectives of this study are twofold. Firstly, we aim to identify the metrics that are currently used to evaluate the interpretability of machine learning models. This will involve a comprehensive review of the existing literature to determine the most commonly used metrics in practice. Secondly, we seek to propose and implement a machine learning framework with high interpretability that can be applied to real-world business problems. To achieve this objective, we will employ supervised and unsupervised machine learning techniques and conduct experiments on a case study dataset containing 7044 customer records using 21 attributes related to their churn decision. Through this study, we hope to contribute to the ongoing conversation on interpretability in machine learning and provide a practical solution for businesses looking to improve the transparency and interpretability of their predictive models. As the concern for ML models' interpretability is somewhat new, very few studies have provided a comprehensive evaluation framework for interpretability metrics. Our study stands out by proposing improvements for the ML models' metrics. Our approach is unique as it combines supervised and unsupervised machine learning techniques and employs a case study dataset to demonstrate the effectiveness of our approach in improving interpretability. By boosting the transparency and interpretability of ML business models our framework has the potential to benefit various industries that rely on machine learning models, such as finance, healthcare, and marketing. Overall, our study provides valuable insights into how interpretability can be optimized. This paper is organized as follows: In Section 2, we conduct a thorough literature review of machine learning interpretability and identify key indicators for measuring interpretability. Building on this review, in Section 3 we propose a novel three-layer framework for enhancing the interpretability of machine learning models. The first layer, called pre-modeling, involves pre-processing techniques applied to the research dataset to prepare it for analysis. In the second layer, called modeling, we develop a novel hybrid machine learning model consisting of both supervised and unsupervised models. In the third layer, called post-modeling, we evaluate the interpretability of the resulting model using the indicators identified in the literature review. The most interpretable model is then selected for further analysis. The results of our study demonstrate that the proposed framework outperforms previous approaches to machine learning interpretability. Finally, in Conclusions, we summarize the main findings of our study, highlight its contributions to the field, discuss its limitations, and suggest directions for future research. Figure 1 illustrates the conceptual framework of this study.

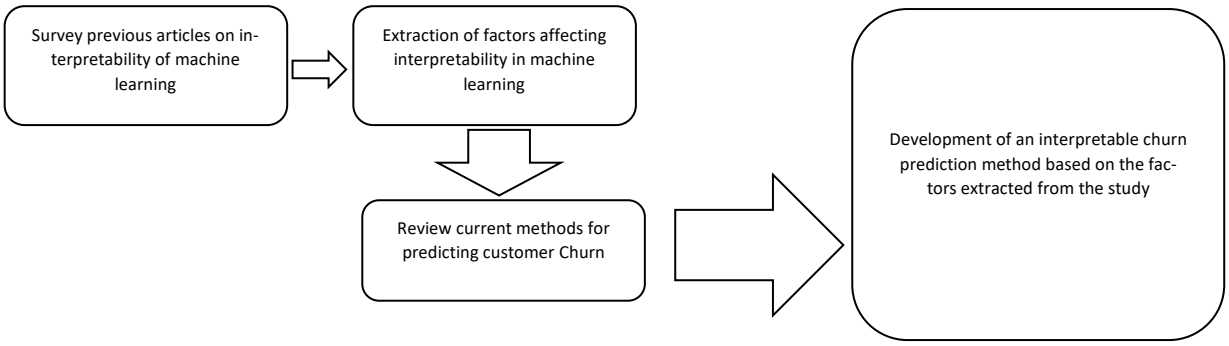


Figure 1:Conceptual framework

2. Literature review

In this section, a survey is conducted on machine learning algorithms and their interpretability. In a review of previous research on customer churn prediction based on machine learning algorithms, it was observed that most researchers intended to improve the performance of classification algorithms. Several researchers aimed to discover the factors influencing churn. Also, some studies have considered the customer churn prediction problem as a binary classification so that the customers could be placed in both churner and non-churner categories. This

classification process was performed using a single machine learning algorithm (Dolatabadi & Keynia, 2017) (Ying et al., 2008). In addition, some researchers have focused on statistical issues and improved algorithms such as Random Forest (RF) Decision Tree (DT), Artificial Neural Network (ANN), Support Vector Machine (SVM), and Logistic Regression (LR). Other researchers have improved the prediction performance of different models using hybrid or ensemble algorithms (Brânduşoiu et al., 2016; Coussement & Van den Poel, 2008; Dahiya & Bhatia, 2015; Gürsoy, 2010; Vafeiadis et al., 2015) For interpretability, the most famous articles that studied machine learning interpretability were reviewed. These articles were selected based on the delivery of measurable indicators for interpretability. This review examines the possibility of using these indicators to provide the required research framework. Since the user trust and local explanations of ML models are critical items in customer churn prediction models, it is necessary to consider the interpretability of these models. For example, a comprehensive interpretation of a neural network depends on the user's knowledge, experience, and desires. In an interpretable system, explanations must be simple and sufficiently understandable to the users. This interpretation is achieved by having a system that behave predictably in various circumstances. Doshi-Velez and Kim stated that a single metric, such as classification accuracy, is an incomplete description of most real-world task which needs humans trust (Doshi-Velez & Kim, 2017). It is hard to trust systems whose decisions cannot be well interpreted, especially in the areas such as health, marketing, or automated driving, where ethical issues and justice are naturally raised. This situation confirms the need for reliable, transparent, and high-performance models for real-world applications, resulting in interpretable machine learning models (Gunning & Aha, 2019). In other words, applying a highly interpretable framework helps managers make fundamental decisions. Nowadays, the volume of research on machine learning interpretability is rapidly growing. This sub-section investigates the studies that provide indicators for evaluating the interpretability of machine learning algorithms. Researchers have addressed machine learning interpretability through ontology and semantics. Most of these studies were focused on purely subjective perceptions of interpretability (Du et al., 2019). Hermann represented the necessity of paying more attention to the interpretability of systems that use human evaluations. The author showed a specific and robust tendency toward more straightforward descriptions and warned that simply paying attention to them could make the system not to be interpretable and complete but to be convincing. Also, he described an ethical challenge: to what extent can an interpretation be ethically manipulated to persuade users? In addition, different people may have various tendencies regarding the interpretability of a model. This situation makes it challenging to determine the interpretability of a model (Herman, 2017). Lage et al. showed that the subjective evaluation of the interpretability of algorithms developed for intelligent automobiles provides algorithms with faster reaction times than more accuracy (Lage et al., 2019).

Velez and Kim introduced three strategies (i.e., application-grounded, human-grounded, and function-grounded) for assessing interpretability. The application-grounded analysis involves conducting human experiments under real-world scenarios. For example, a doctor can decide through the disease diagnosis results. The human-grounded analysis is linked to the realization of additional simple human-subject experiences that maintain the essence of the target application. Although the functionally-grounded analysis does not require human experience, it uses a formal definition of interpretability as a proxy for explanatory quality (Doshi-Velez & Kim, 2017).

Murdoch et al. introduced (PDR) framework to discuss ML interpretation. Their framework provides three overarching requirements for evaluation: predictive accuracy, descriptive accuracy, and relevance judged relative to a human audience. Also, they categorized the criteria into two groups, including model-based and post-hoc (Murdoch et al., 2019). Yang Do et al. divided the ability to interpret machine learning into two dimensions: scale and method. In the scale dimension, there are two interpretations (i.e., local and global). The global interpretation refers to the overall model structure, and the local interpretation refers to the model behavior in a particular instance or state of the problem. Interpretation methods are divided into two categories, including model-based and post-hoc interpretations. They used three criteria (i.e., generalization, transparency, and persuasiveness) to evaluate interpretations. The interpretability evaluation metric proposed in this study included three layers. The generalization, persuasiveness, and transparency were the first, second, and third layers, respectively. Also, the structure of this framework was hierarchical (Yang et al., 2019). Moraffah et al decomposed the interpretability concept of machine learning algorithms into two categories. The first category was related to the essence of the model. It comprises the models whose training and decision-making processes were intuitive. The authors introduced decision tree models, rule-based models, linear regression, and belief networks as inherently interpretable models. The second category, post-hoc interpretability, was related to the ability to interpret models after their decision. It refers to creating explanations for interpreting an existing model using an auxiliary model. Also, the example-based models fall into this category (Moraffah et al., 2020). Approximations make a case to represent that more complex models result in a specific prediction by fitting a less complicated model, either regionally (approximate just for one or a couple of data points) or globally (approximate the behavior of the whole model). In recent years, many believed that local approximation was the most scientific and promising method to explain how complex models behave (Mittelstadt et al., 2019). However, local models can solely be a particular domain of a model, and thus they can be deceptive and inaccurate outside the domain. The main characteristics of machine learning interpretability reported in mentioned studies are listed in Table 1.

Table 1. Machine learning interpretability in literature

Post-hoc	Model based	framework	scope	Reference
1.Dataset-Level Interpretation 2.Prediction-Level Interpretation	1.sparsity 2.Simulatability 3. Modularity 4.Domain Based Feature Engineering 5.Model-Based Feature Engineering	1.Predictive accuracy(modeling) 2.Descriptive accuracy (post hoc) 3.Relevancy (post hoc)	1.Modeling 2.Post Hoc	(Murdoch et al., 2019)
Fidelity	1.Local and global 2.Generalizability 3.Model performance	1.generalizability 2.fidelity 3.persuasibility	1.Intrinsic local and global 2.Post hoc local and global	(Yang et al., 2019)
1. natural language explanations 2. visualizations approximations	An algorithm is simple enough to be examined all at once by a human		transparency post-hoc interpretation	(Lipton, 2018)

3. Methodology

In this section, we will present the step-by-step framework proposed to predict customer churn with high interpretability. The framework design consists of three layers that work together to preprocess and classify data, and to evaluate and improve the interpretability of the prediction model. The first layer involves data preprocessing techniques such as cleaning, filtering, and transforming the data to make it suitable for analysis. The second layer proposes a novel classification model based on supervised and unsupervised algorithms to predict customer churn. Finally, the third layer uses evaluation criteria to improve the interpretability of the prediction model. This section will describe the framework's components in detail and explain how they work together to predict customer churn with high accuracy and interpretability.The literature survey indicated that machine learning interpretability was associated with three steps (Lage et al., 2019). There are regional and global interpretability techniques before constructing a prediction model, within the modeling process, and after modeling.The framework of the present study consists of three modules: pre-model, model-based, and post-hoc. The pre-model module uses model-independent techniques that can only be utilized independently for the data. In the model-based module a hybrid classification model with clustering and classification used, finally in post-modeling module interpretability of predictions were evaluated.The first step of CCP is associated with data collection. Indeed, different aspects of data collection can affect interpretability. In the second step, a prediction model is developed. Interpretability considerations at this step are often related to selection between simpler and easier models for interpreting and black-box models with more prediction performance. After training the model, the user analyzes the results and answers how the model performed the results. The interpretability is maximized by clustering customers based on behavioral characteristics. In clustering, data points with similar characteristics are placed in the same group so that the members of each group have the most similarity. Also, they have the slightest similarity with the members of other groups. Since the group members are homogenous, the prediction accuracy of classification algorithms can be improved (Sivasankar & Vijaya, 2019). With this effect, the readability of the input data and the prediction process become understandable to the user, and the user can observe the prediction for separate behavioral groups. The clustering methods are divided into four categories: segregated algorithms, hierarchical, density-based, and model-based clustering algorithms.

3.1. Pre-modeling

Different aspects of the data-collection process can affect the interpretation pipeline (Murdoch et al., 2019). This type of interpretation usually takes place before choosing a model. Indeed, it is vital to have a good understanding of the data before selecting a model. A meaningful display of features and showing them with higher importance using EDA can increase interpretability (Komorowski et al., 2016). These methods include traditional data exploration such as principal component analysis (PCA) (Howley et al., 2005) and t-SNE (Hung, 2017) or newer methods such as K-Means and MMD.

3.2. Model-based

This module focuses on constructing models that quickly provide information for relationships they have learned. In this case, the model-based interpretability techniques usually require simpler models, leading to predictive performance reduction. Therefore, model-based interpretability is the best option when the underlying relationship is sufficiently straightforward to permit the model-based techniques to attain affordable predictive performance.

The literature survey revealed that most classification algorithms were evaluated under predictive and descriptive performance to predict customer churn. Besides, the logistic Regression, Support Vector Machine, Decision Tree, Random Forest, Ada-boost, and Multilayer Perceptron algorithms were applied before and after hyperparameter tuning to determine the best configuration for high predictive performance. The model generalization is the same as the prediction (Lakkaraju et al., 2016; Letham et al., 2015). Also, the post-hoc generalization considers the same metrics on test data (Che et al., 2015; Frosst & Hinton, 2017). Although the transparency factor assessment is not precisely defined for the model, the essence of most of the developed models is transparent.

3.3. Post-hoc interpretability

In post-hoc interpretation methods, the model is first trained. Then, information about relationships between the features and target is extracted from what the model has learned. These are helpful when the high-dimensional complicated dataset forces modelers to use complex or black-box models to achieve high predictive performance, which sometimes leads to low descriptive performance. In these cases, descriptive proxies with a different structure can be used. In this regard, introducing perturbations and ablations to data have been proposed (Fong & Vedaldi, 2017; Nguyen et al., 2015). The main objective of these methods is to change the algorithm input to observe their effect on prediction. Another aspect of post-hoc interpretation approaches involves separating learning tasks and the model explanation (Adadi & Berrada, 2018). These explanations may be expressed using visualizations, natural language or text, rules, examples, and other formats (Adadi & Berrada, 2018; Escalante et al., 2018). Also, these approaches can be categorized into two groups, including model-specific and model-agnostic approaches (Došilović et al., 2018). The model-based methods can only be used for specific models. They rely on the idiosyncrasies of their internal mechanisms. These explanations may target local and global levels, but the global explanations tend to be more frequent. The model-agnostic explanation approaches are not related to any model or algorithm. In other words, they consider the original model as a black-box system. Also, they only analyze the inputs and outputs of the model and then explain the model behavior. In addition, these approaches provide explanations at the global and instant levels. The instance-level explanations are more common than the global levels. But their completeness level is lower than other approaches. Besides, they can typically provide high comprehensibility and offer the attractive advantage of being generalizable. More specifically, the model-agnostic post-hoc explanation approaches provide general explanation formats that allow customization to fit user information needs, enable comparisons of different models, and facilitate the switching out process of a model in a deployed ML system.

3.4. Evaluation criteria

True positive (TP), true negative (TN), false positive (FP) and false negative (FN) measures are generally considered to evaluate the performance of a machine learning classification model. These are described by a confusion matrix, which has classification label frequencies in four variables. These variables include accuracy, sensitivity (recall), specificity, precision, error rate, F-score, and Area Under the Receiver Operator Characteristic (ROC) (AUC) derived from the Confusion Matrix. The present study considered F1, recall, accuracy, and AUC using Equations (1-4).

$$Precision = TP/(TP + FP) \tag{1}$$

$$Recall = TP/(TP + FN) \tag{2}$$

$$F1 = 2 * (Precision * Recall)\ /(Precision + Recall) \tag{3}$$

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{4}$$

The interpretability evaluation indexes are defined in the study framework

Table ***1***. Also, the clustering and classification algorithm hyperparameters were tuned for the recall score. Then, prediction model stability and its parameter sensitivity were evaluated. The recall score was chosen because the true positive rate in telecom churn prediction is more valuable than other metrics. Indeed, the cost of losing a true churner is more than its retention. However, other metrics are important while evaluating a classification algorithm.

Table 1. Indexes for interpretability evaluation

Pre-modeling	Model-base	Post-hoc1	Post-hoc2
Missing value	Hyperparameter tuning	Intrinsic interpretability	Stability
Feature selection	Training score		Generalizability
Balancing			Parameter sensitivity
Normalization and scaling			

1-model specific 2-model-intrinsict

The current article summarizes the main characteristics of machine learning interpretability reported in these studies, such as the three strategies for assessing interpretability introduced by Doshi-Velez and Kim, the framework for discussing ML interpretation proposed by Murdoch et al., and the ability to interpret machine learning divided into two dimensions proposed by Yang Do et al. Indicators identified during this review has enabled us to propose an interpretable ML framework for customer churn prediction.

4. Implementation

The proposed framework has been implemented using the IBM telecommunication dataset. The data used is a dataset that was updated in 2021. This data is used because it has a variety of features and no missing pieces of data, it is very suitable to be used as a model for machine learning. this dataset provides 7043 customers with 21 features. Python programming language was used to implement clustering, classification, dimensionality reduction, visualization techniques, and data preprocessing

4.1. Data preprocessing

The raw data must be preprocessed for application in machine learning algorithms. The dataset includes information about Customers who left within the last month – the column is called Churn, Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device -protection, tech support, and streaming TV and movies, Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges Demographic info about customers – gender, age range, and if they have partners and dependents. Table 3 shows other features.

Table 2.Costumer main features in IBM dataset

Attribute	Definition	Attribute	Definition
CustomerID	A unique ID that identifies each customer	**Number of Dependents**	the number of dependents that live with the customer.
Gender	The customer's gender: Male, Female	**Phone Service**	if the customer subscribes to home phone service with the company: Yes, No
Senior Citizen	Indicates if the customer is 65 or older: Yes, No	**Multiple Lines**	f the customer subscribes to multiple telephone lines with the company: Yes, No
Married (Partner)	Indicates if the customer is married: Yes, No	**Internet Service**	Indicates if the customer subscribes to Internet service with the company: No, DSL, Fiber Optic, Cable.
Dependents	if the customer lives with any dependents: Yes, No. Dependents could be children, parents, grandparents, etc.	**Online Security**	Indicates if the customer subscribes to an additional online security service provided by the company: Yes, No
Online Backup	if the customer subscribes to an additional online backup service provided by the company: Yes, No	**Device Protection Plan**	if the customer subscribes to an additional device protection plan for their Internet equipment provided by the company: Yes, No
Premium Tech Support	if the customer subscribes to an additional technical support plan from the company with reduced wait times: Yes, No	**Streaming TV**	if the customer uses their Internet service to stream television programing from a third-party provider: Yes, No. The company does not charge an additional fee for this service.
Streaming Movies:	Indicates if the customer uses their Internet service to stream movies from a third-party provider: Yes, No. The company does not charge an additional fee for this service.	**Contract**	Indicates the customer's current contract type: Month-to-Month, One Year, Two Year.
Paperless Billing	if the customer has chosen paperless billing: Yes, No	**Payment Method**	how the customer pays their bill: Bank Withdrawal, Credit Card, Mailed Check
Monthly Charge	the customer's current total monthly charge for all their services from the company.	**Total Charges**	Indicates the customer's total charges, calculated to the end of the quarter specified above.
Tenure	the total amount of months that the customer has been with the company.		

It is important to deal with null or missing values in the dataset and to check the dataset for imbalanced class distributions, which has been one of the emerging problems of machine learning (García et al., 2009). The problem of imbalanced dataset can be solved through re-sampling techniques (Qureshi et al., 2013) or by enhancing

evaluation metrics (Burez & Van den Poel, 2009), etc. The number of churners is often less than non-churners as in the dataset used in this study (Gattermann-Itschert & Thonemann, 2021; Pan et al., 2020), which drives an imbalanced-data problem. The distribution of target in dataset used in this study shown in Figure 2. It can be observed that percentage of non-churners (26.5%) is significantly smaller than the percentage of churners. A balanced dataset would typically have an equal or nearly equal distribution of the target variable categories. Table 4represents how the imbalanced data problem affects interpretability (Seiffert et al., 2009).

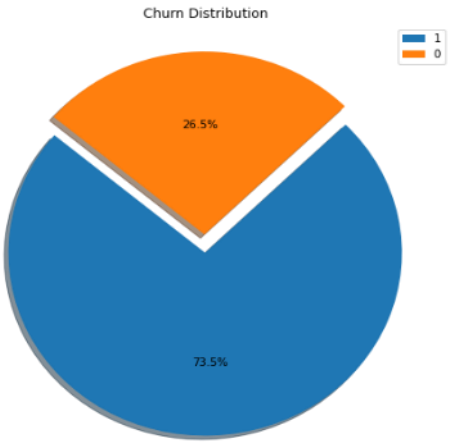


Figure 2. Churn class distribution

Table4. Imbalance dataset effects on interpretability

Interpretability	Consequence	Imbalance data problems
Decreased predictive performance	Difficulty in finding template pattern	The insignificant number for independent variable data
Decreased descriptive performance	Inaccurate evaluation of results	The insignificant number compared to other variables
Decreased predictive and descriptive accuracy	Heterogeneous distribution in k-fold or splitting methods	Data fragmentation for training and testing

The SMOTE preprocessing technique is a pioneering approach for the research community in classification problems. After its release, several developments have been made to improve its performance under various scenarios. The smote is used in most imbalanced data problems because of the popularity of this method (DeCastro-García et al., 2019). Therefore, it has been considered to balance the datasets in this study.

4.1.1. Normalization and transformation

Since most machine learning models only accept numerical variables, preprocessing the categorical variables becomes a necessary step. Label encoding and dummy encoding were used for both ordinal and nominal variables in dataset. data normalization is also an essential pre-processing step which involves the transformation of features in a common range so that higher numeric feature values would not dominate the smaller ones. The main aim is to minimize the bias of those features whose numerical contribution is higher in discriminating pattern classes. A study showed that data scaling techniques such as MinMax normalization and standardization have also significant effects on data analysis (Ambarwari et al., 2020). Amin et al for LR, KNN, SVM, DT and RF in their study are affected with using data scaling (Amin et al., 2019). In this study we applied standard scalar technique to scale dataset.

4.1.2. Identification of most suitable data

Most classification problems (e.g., CCP) include many features, leading to lower predictive performance. The post-hoc interpretation is also affected by many low-importance features, which reduces the descriptive performance of the models. The correlation matrix of features is employed to select features. The Pearson (product-moment) correlation coefficient is a measure to determine the linear relationship between two features Equation (5).

$$r = \sum_i \Big((x_i - mean(x))(y_i - mean(y))\Big) \Big(\sqrt{\sum_i (x_i - mean(x))^2} \sqrt{\sum_i (y_i - mean(y))^2}^2 \tag{5}$$

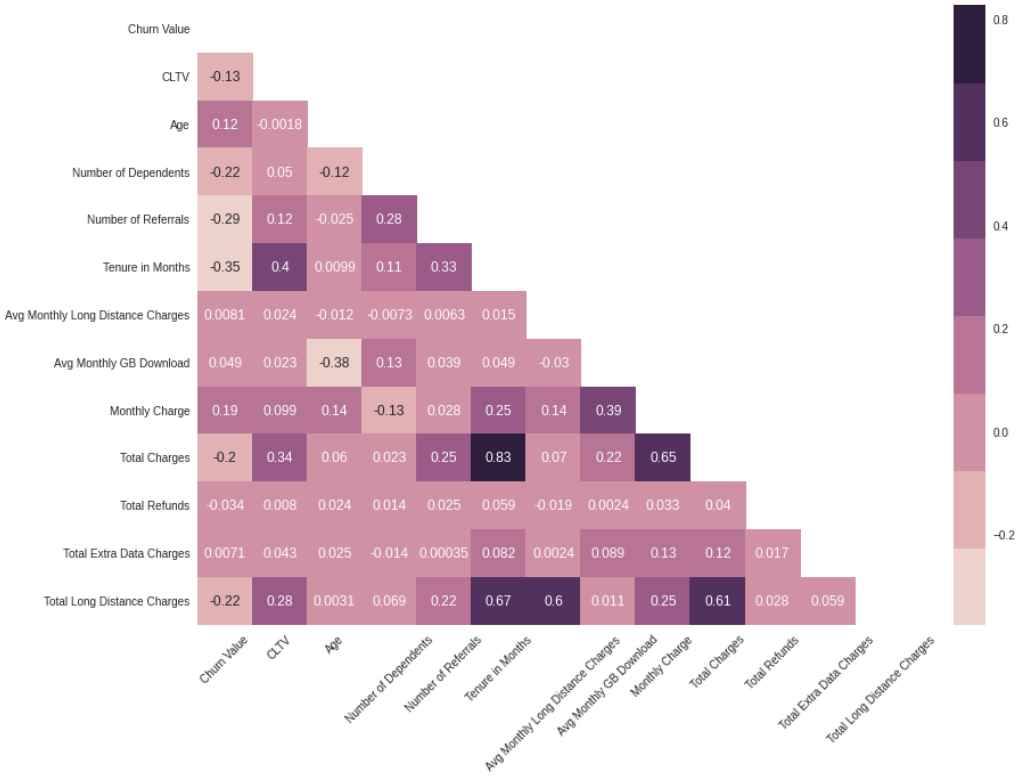


Figure 3. Correlation matrix of attributes

The r denotes a value between -1 and 1 in which the higher number shows a stronger correlation. Correlation matrix of attributes (ratio of the covariance of x and y to the product of their standard deviations) was considered to find the correlation between input features in this study. In this study, the independency of features was maintained by omitting features with a correlation of more than 80. This threshold was chosen based on previous research indicating that highly correlated features can lead to overfitting, increased computational costs, and reduced interpretability of the resulting model. Figure 3 displays the correlation scores between all pairs of features, with values ranging from -1 to 1. Negative scores indicate a negative correlation, while positive scores indicate a positive correlation. The diagonal of the matrix is always 1, as it represents the correlation of each feature with itself.

4.2. Hybrid prediction model

The preprocessed data were given to the clustering algorithm in the model-based module. Here, the numeric features of Tenure in Months, Avg Monthly Long-Distance Charges CLTV, Monthly Charge, and Avg Monthly GB Download were considered for clustering customers. In order to determine the appropriate number of clusters, the Elbow method was used. This a visual method to test the consistency of the best number of clusters by comparing the difference of the sum of square error (SSE) of each cluster, the most extreme difference forming the angle of the elbow shows the best cluster number. Table5 compares the performance scores of different clustering algorithms with a fixed number of clusters equal to 2, which was determined to be the optimal number of clusters for this dataset using the Elbow method. The table includes four columns: name of clustering algorithm, Silhouette score, Calinski-Harabasz score, and Davies-Bouldin score. The 'Silhouette', 'Calinski-Harabasz', and 'Davies-Bouldin' columns provide performance scores for each algorithm. The Silhouette score measures how well each data point is clustered with its own group compared to other groups, while the Calinski-Harabasz score measures the ratio of between-cluster variance to within-cluster variance. The Davies-Bouldin score measures the average similarity between each cluster and its most similar cluster, considering the distance between their centroids. Table 3 shows the performance of various clustering algorithms on the dataset used in this study, as measured by the Silhouette, Calinski-Harabasz, and Davies-Bouldin indices. K-means achieved the highest Silhouette and Calinski-Harabasz scores, indicating that it produced clusters with high intra-cluster similarity and low inter-cluster similarity. The Davies-Bouldin index for K-means was also relatively low, suggesting that the clusters produced by this algorithm were well-separated. Results for Agglomerative clustering (hclust) and Birch were similar, with moderate Silhouette and Calinski-Harabasz scores, and low Davies-Bouldin scores. These algorithms are known for their ability to handle large datasets and produce hierarchical cluster structures. DBSCAN and Kmods produced the lowest

Silhouette scores, indicating that the clusters produced by these algorithms had low intra-cluster similarity. However, DBSCAN achieved a low Davies-Bouldin score, suggesting that the clusters it produced were well-separated. Kmods, on the other hand, had a high Davies-Bouldin score, indicating that the clusters produced by this algorithm were not well-separated.Overall, K-means performed the best among the tested algorithms, followed by hclust and Birch. DBSCAN and Kmods had relatively poor performance, but may be useful in certain scenarios where well-separated clusters are desired, or when dealing with high-dimensional data.The K-means clustering algorithm is an unsupervised technique that splits a large set of elements based on their features and characteristics into K groups. In this case, each group is called a cluster. Also, the intra- and extra-cluster distances should be minimized and maximized, respectively. This issue means that the elements of one cluster are similar, while these elements are different from the elements of other clusters (Han et al., 2011).

Table 3. Clustering algorithms scores comparison with number of clusters equal to 2

Algorithm	Silhouette	Calinski-Harabasz	Davies-Bouldin
K-means	0.31*	3497*	1.32*
*ap**2***	0.17	390.2	1.38
*sc**3***	0.31	3324	1.23
*hclust**4***	0.29	2767	1.25
*dbscan**5***	0.27	27.4	1.55
Kmods	0.11	543.7	2.83
Birch	0.28	2830.6	1.34[6]

All clusters would be used as input for all chosen classification algorithms (Logistic Regression, Support Vector Machine, K-NearestNeighbor (KNN), Decision trees (DT), RandomForest(RF), Gradient Boosting(GB),XGrediant boosting(XGB)). In each execution, the Grid search method would be employed to select the best hyperparameter values and maximize the recall score. The training data will be fed to the model in cross validation method. This means training data is randomly divided into n equal parts and in each iteration one of them is used as the input for all classification algorithms (Cullen, 1993).
Trained models performing acceptably on validation data (recall >= 0.75), are chosen for processing the test data.

4.3. Results and evaluation

All models were trained on IBM dataset using training and test sets chosen by cross validation with partition types "hold-out" 20% and "k-fold" where the k value used is 10. Testing scores with and without clustering are shown in Table 4 and Table 6 respectively. Our experiment shows the best result is obtained using clustering. It also shows a significant boost in recall score in comparison with all results published in previous research (Beeharry & Tsokizep Fokone, 2022; Ebrah & Elnasir, 2019; Lian-Ying et al., 2019; Momin et al., 2020). **Error! Reference source not found.** presents the pseudo-code of the study proposed prediction algorithm.

4.3.1. Hyperparameter optimization

Hyperparameter optimization is a systematic process that helps in finding the right hyperparameter values for a machine learning algorithm. In this work, Grid Search (GS) (Syarif et al., 2016) has been used to optimize the parameters of eight classifiers.Grid search was compared to other hyperparameter optimization techniques in a study by Bergstra and Bengio (2012) and was found to be an effective and reliable method for tuning hyperparameters. The study found that grid search produced better or comparable results to more advanced techniques in many cases(Bergstra & Bengio, 2012). Therefore, grid search is a recommended method for hyperparameter tuning, especially when the hyperparameter space is small or the computational resources are limited. we have used GS to optimize the hyperparameter of all the classifiers in this study. Specifically, we have used the GridSearchCV method from the sklearn python library(Syarif et al., 2016). Table 5 provides a detailed summary of the hyperparameter tuning candidates for each algorithm used in our analysis. The table consists of three columns: 'Classifier', 'Meta-parameter', and 'Candidate settings'. The 'Classifier' column lists the name of each algorithm, while the 'Meta-parameter' column lists the hyperparameters that were tuned for each algorithm. The 'Candidate settings' column shows the range of values tested for each hyperparameter.

[2] Affinity Propagation
[3] Spectral Clustering
[4] Agglomerative Clustering
[5] Density-Based Spatial Clustering

Table 4. Classification generalization scores before clustering

	F1	Recall	ROC AUC	Accuracy
Logistic Regression	0.844515	0.868506	0.833907	0.835008
SVM	0.839969	0.873377	0.826827	0.828308
KNN	0.819549	0.88474	0.796176	0.798995
Decision Tree	0.801308	0.795455	0.796516	0.796482
Random Forest	0.868654	0.875	0.863106	0.863484
AdaBoost	0.832268	0.845779	0.823409	0.824121
GradientBoost	0.851133	0.853896	0.845633	0.845896
XGBoost	0.858521	0.866883	0.852127	0.852596

Table 5. Hyperparameter tuning candidates for each hyperparameter

Classifier	Meta-parameter	Candidate settings
Gradient Boosting	n_estimators	[100, 200, 400, 700, 1000]
	colsample_bytree	[0.5, 0.6, 0.7, 0.8]
	max_depth	[15,20,25]
	num_leaves	[25, 50, 100, 200]
	reg_alpha	[0, 1e-1, 1, 2, 5, 7, 10, 50, 100]
	reg_lambda	[0, 1e-1, 1, 2, 5, 7, 10, 50, 100]
	min_split_gain subsample	[0.3, 0.4]
		[0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
Support Vector Machine	C	[0.1,1, 10, 100]
	Gamma	[1,0.1,0.01,0.001]
	Kernel	['rbf', 'poly', 'sigmoid']
	Class_weight	['balanced', None]
K nearest neighbor	n_neighbors	[11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43]
	leaf_size	[20,40,1]
	p	[1,2]
	weights	['uniform', 'distance']
	metric	['minkowski', 'chebyshev']
Random Forest	bootstrap	[True, False]
	max_depth	[10, 20, 30, 40, 50]
	max_features	['auto', 'sqrt']
	min_samples_leaf	[1, 3, 4]
	min_samples_split	[2, 6, 10]
	n_estimators	[5, 20, 50, 100]
Logistic Regression	Penalty	['l1', 'l2', 'elasticnet', 'none']
	C	[-1,0,0.00001,0.0001,0.001,0.01,0.1,1]
	Solver	['lbfgs','newton-cg','liblinear','sag','saga']
	max_iter	[100, 1000,2500, 5000]
Decision Tree	max_depth	[2, 3, 5, 10, 20]
	min_samples_leaf	[5, 10, 20, 50, 100]
	criterion	["gini", "entropy"]
Ada boost	n_estimators	[40,50,60,70,80, 90,350]
	learning_rate	[0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2,0.7]
	algorithm	['SAMME.R', 'SAMME']

Table 6. Classification generalization scores on clustered data

	Cluster1				Cluster2			
Algorithm	**F1**	**Recall**	**ROC**	**Accuracy**	**F1**	**Recall**	**ROC**	**Accuracy**
LR	0.89	0.91	0.89	0.89	0.85	0.86	0.84	0.84
SVM	0.82	0.96	0.78	0.79	0.82	0.86	0.81	0.81
KNN	0.88	0.99	0.88	0.87	0.82	0.93	0.79	0.79
DT	0.9	0.92	0.9	0.90	0.82	0.96	0.78	0.79
RF	0.86	0.88	0.85	0.85	0.86	0.88	0.86	0.86
GB	0.91	0.92	0.91	0.91	0.81	0.80	0.80	0.8
XGB	0.95	0.94	0.94	0.95	0.88	0.88	0.87	0.87
AdaBoost	0.83	0.96	0.81	0.81	0.80	0.97	0.74	0.75

It can be observed that using clustering with grid search hyperparameter tuning generally improves scores across all algorithms. Table 7 shows top three algorithms for each of aforementioned experiments. It is shown that KNN performs great in both experiments although other algorithms like SVM performs much better using clustering. One can remark that AdaBoost outperforms other algorithms in 2nd cluster situations.

Table 7. Algorithm ranking for prediction

Experiment	First	Second	Third
Without clustering	KNN	SVM	LR
Cluster-based 1st	KNN	SVM	AdaBoost
Cluster-based 2nd	AdaBoost	DT	KNN

4.4. Parameter sensitivity

Machine learning (ML) algorithms configurations by their hyperparameters are virtually limitless. On the other hand, this choice can considerably influence the complexity, behavior, and learning speed of the model. As mentioned before, Hyperparameter optimization finds the best values for hyperparameters of a prediction algorithm to maximize the model predictive performance. A global visual representation of how one or two hyperparameter influence the predicted outcome of the model, with other features held constant is provided in

Model creation phase homogenization:
Split D in Dtr and Dval and D_{tst}
Input: (training) data
Define the number of clusters by Elbow method equal to K
Calculate initial K-means on *Dtr* spanning the total space *C*
Define subspaces Ct based on a set of Clusters K for which $C = \bigcup_{t \in K} C_t$, ∀ t≠t': $C_t \cap C_{t'} = \phi$
Assign initial values for C_1 , C_2 , …, C_K
Repeat
Assign each item D_i to the clusters which has the closet mean;
Calculate new mean for each cluster;
Until convergence criteria is met;
Output: Model M

Model creation phase classification algorithm:
Input: training data (D_{tr}) and validation data (D_{val})
Define set Classifiers = {}
For t=1 to k **do:**
For $classifire_j$ in Classifiers **do:**
Apply classifier specific for C_t
Optimize classifier with Random search on candidate spaces
Save the evaluation result of $classifire_j$ on C_t
End for
Select $argmax_{classifire}$ in result for C_t
End for

Output: best classifier for each cluster

Prediction phase:
Input: test data (D_{tst}) = $\{(Xi , Yi)\}_{i=1}^{N}$
Apply clustering rules of model M on D_{tst} spanning the total space *C*, resulting in clusters with t = 1…k
For $i = 1\ to\ k$ **do:**
Apply selected classifier specific for *Ct*
For $J = 1\ to\ n_i$ **do:**
Calculate predictions for all n_i instances in *Ct*
End For;
End For;
Combine predictions
Output: one prediction for every instance in *C*

Table 8. In this study, hyperparameter sensitivity curves show how much the model performance is changed by altering the hyperparameter values. The y-axis is recalled to be maximized, and the x-axes are different hyperparameters for each algorithm. Unlike other interpretability criteria, hyperparameter sensitivity does not have a specific preference to be chosen. However, a less sensitive model can be preferred by the less experienced users when it is easier to train with acceptable results (Lavesson & Davidsson, 2006). Table 10 gives the parameter sensitivity for the best algorithms in each cluster.

4.5. Stability

In (Mohr & van Rijn, 2022), the authors used the learning curve concept to assess the performance of a learning algorithm with respect to a certain resource, e.g. the number of training examples or the number of training iterations. Figueroa et al. (Figueroa et al., 2012) introduced the desired shape for learning curves produced by a machine learning algorithm. Based on this shape, there is a rapid performance increase, followed by a turning point with a less rapid performance increase, and eventually, the curve becomes flat. Table 10 provides a summary of the descriptive performance of the top three classification algorithms, as determined by their prediction score. To evaluate their interpretability, we examined two key indicators: How the learning curve evolves when new data points are fed to the model, assessing model's ability to generalize (stability) and how it changes when hyperparameters are changed, determining the model's parameter sensitivity. These indicators allow us to assess how the algorithms perform as their complexity changes, how robust they are to changes in hyperparameters and how they perform against new data. The table includes three columns: the name of the algorithm, the parameter sensitivity plot with its corresponding hyperparameter values, and the learning curve for each model. By examining these indicators, we can gain insight into how the algorithms function and make more informed decisions about their interpretability.

5. Conclusion

Predicting customer churn is one of the most important factors in business planning in TELCOs. To improve the churn prediction interpretability, we first extracted machine learning interpretability in literature and explored ways to make machine learning algorithm more interpretable. Second, a 3-step framework is proposed, combining best clustering techniques with eight different machine learning classifiers (K-Nearest neighbor (KNN), Logistic regression (LR), Random Forest (RF), Decision tree (DTree), Gradient boosting (GB), Gradient boosting (XGB), Support Vector Machine (SVM), Adaboost). Feature selection method was applied to remove unwanted features and grid search technique was used for hyperparameter tuning. The KNN, SVM, LR for cluster 1 and Adaboost, DT and KNN for cluster 2 are the top performing classifiers. We evaluated our methods in terms of recall score. Then the stability and parameter sensitivity of those top algorithms were evaluated. We compared our proposed models with other state-of-the-art techniques and found that the performance of our proposed model is significantly better than that of state-of-the-art techniques. Our proposed framework can be tested on the other telecom datasets to examine the generalization of our results at a larger scale. Last but not the least, work can be done to extend our approach to customer churn datasets from other business sectors to study the generalization of our claim across business domains

Figure 4. Pseudo-code of proposed Hybrid prediction mod

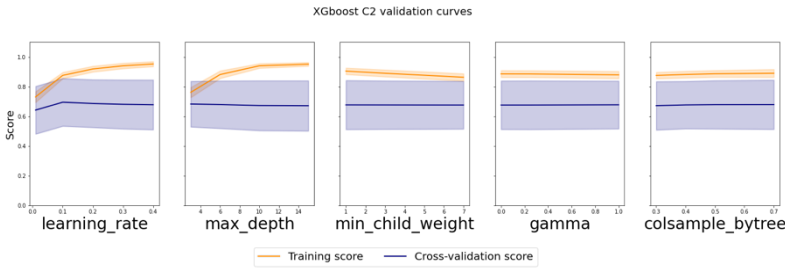
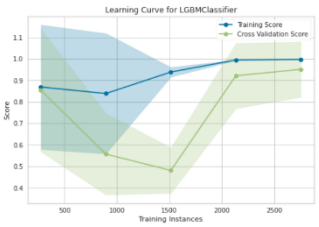
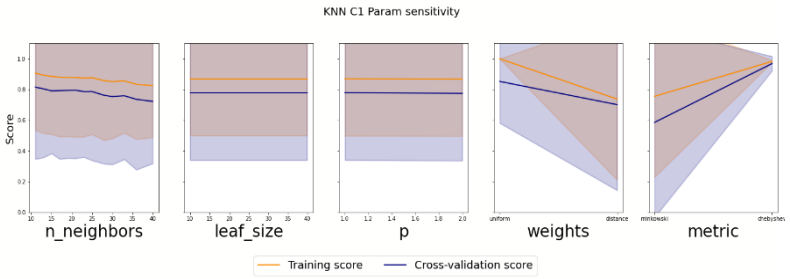
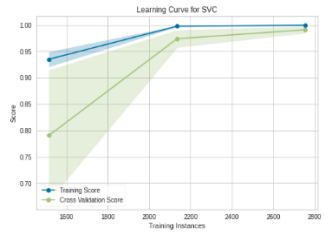
Model creation phase homogenization:
Split D in D_{tr} and D_{val} and D_{tst}
Input: (training) data
Define the number of clusters by Elbow method equal to K
Calculate initial K-means on D_{tr} spanning the total space *C*
Define subspaces Ct based on a set of Clusters K for which $C = \bigcup_{t \in K} C_t$, ∀ t≠t′: $C_t \cap C_{t'} = \phi$
Assign initial values for C_1 , C_2 , …, C_K
Repeat
Assign each item D_i to the clusters which has the closet mean;
Calculate new mean for each cluster;
Until convergence criteria is met;
Output: Model M

Model creation phase classification algorithm:
Input: training data (D_{tr}) and validation data (D_{val})
Define set Classifiers = {}
For t=1 to k **do:**
For $classifire_j$ in Classifiers **do:**
Apply classifier specific for C_t
Optimize classifier with Random search on candidate spaces
Save the evaluation result of $classifire_j$ on C_t
End for
Select $argmax_{classifire}$ in result for C_t
End for

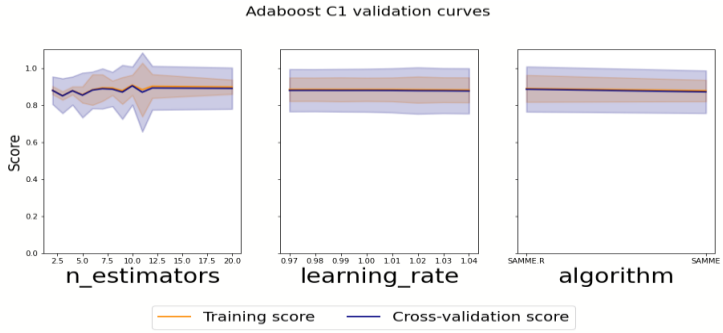
Output: best classifier for each cluster

Prediction phase:
Input: test data (D_{tst}) = $\{(Xi\ , Yi)\}_{i=1}^{N}$
Apply clustering rules of model M on D_{tst} spanning the total space *C*, resulting in clusters with t = 1…k
For $i = 1\ to\ k$ **do:**
Apply selected classifier specific for C_t
For $J = 1\ to\ n_i$ **do:**
Calculate predictions for all n_i instances in C_t
End For;
End For;
Combine predictions
Output: one prediction for every instance in *C*

Table 8:Parameter sensitivity and learning curve for algorithms with best recall score

Algorithm	Parameter Sensivity	Learning curve
Cluster1		
SVM		
	Best Parameters of GridSearchCV: **{'classifier__C': 10, 'classifier__l1_ratio': 0.0, 'classifier__penalty': 'l2'}**	
KNN		
	Best Parameters of GridSearchCV: **leaf_size=10, metric='minkowski',n_neighbors=11,p=1,weights='distance'**	

Adaboost

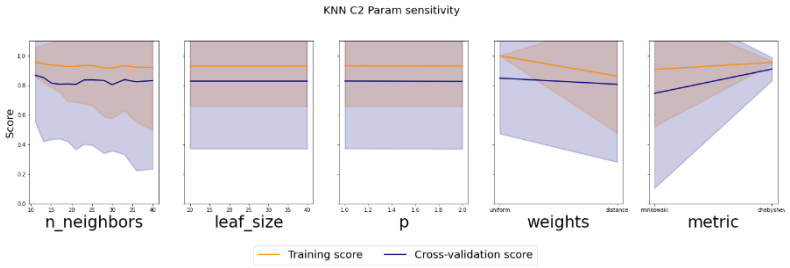


Best Parameters of GridSearchCV:

algorithm='SAMME', learning rate=1.01, estimators=10

Cluster2

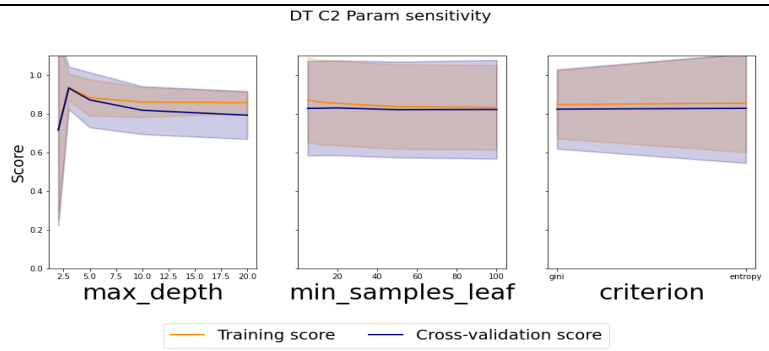
KNN

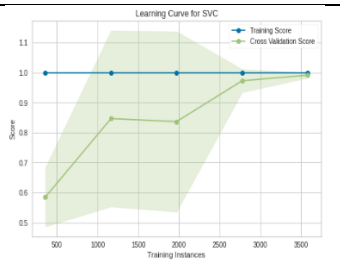


Best Parameters of GridSearchCV:

leaf_size=10, metric='minkowski',n_neighbors=11,p=1,weights='distance'

DT

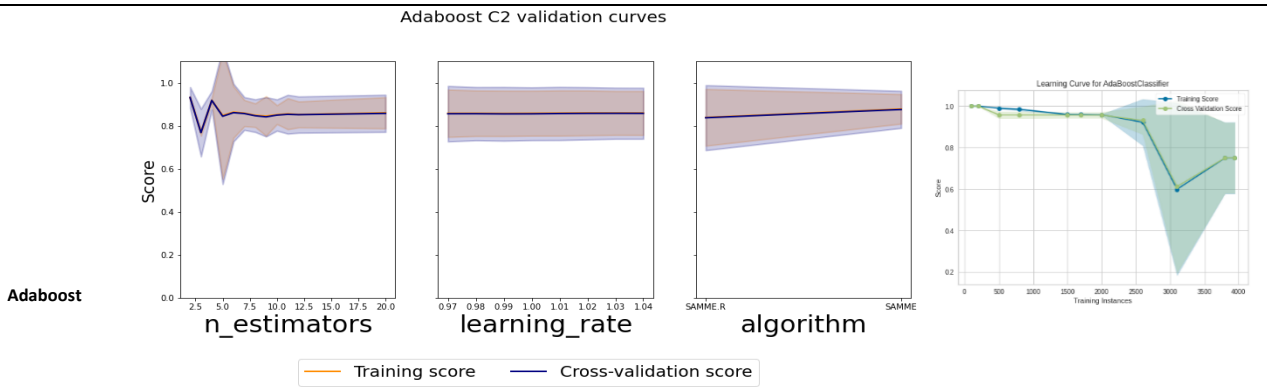




Best Parameters of GridSearchCV:

class weight='balanced', criterion='entropy', max_depth=3, min_samples_leaf=5

Adaboost



Best Parameters of GridSearchCV:

algorithm='SAMME',learning_rate=0.97,n_estimators=2

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access, 6, 52138–52160.

Ambarwari, A., Adrian, Q. J., & Herdiyeni, Y. (2020). Analysis of the effect of data scaling on the performance of the machine learning algorithm for plant identification. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 4(1), 117–122.

Amin, M. S., Chiam, Y. K., & Varathan, K. D. (2019). Identification of significant features and data mining techniques in predicting heart disease. Telematics and Informatics, 36, 82–93.

Beeharry, Y., & Tsokizep Fokone, R. (2022). Hybrid approach using machine learning algorithms for customers' churn prediction in the telecommunications industry. Concurrency and Computation: Practice and Experience, 34(4), e6627.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(2).

Brânduşoiu, I., Toderean, G., & Beleiu, H. (2016). Methods for churn prediction in the pre-paid mobile telecommunications industry. 2016 International Conference on Communications (COMM), 97–100.

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. Expert Systems with Applications, 36(3), Article 3.

Che, Z., Purushotham, S., Khemani, R., & Liu, Y. (2015). Distilling knowledge from deep networks with applications to healthcare domain. ArXiv Preprint ArXiv:1512.03542.

Coussement, K., Lessmann, S., & Verstraeten, G. (2017). A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry. Decision Support Systems, 95, 27–36.

Coussement, K., & Van den Poel, D. (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. Expert Systems with Applications, 34(1), 313–327.

Cullen, S. (1993). Selecting a classification method by cross-validation. (pp. 135–143). Machine learning 13.

Dahiya, K., & Bhatia, S. (2015). Customer churn analysis in telecom industry. 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 1–6.

DeCastro-García, N., Muñoz Castañeda, Á. L., Escudero García, D., & Carriegos, M. V. (2019). Effect of the Sampling of a Dataset in the Hyperparameter Optimization Phase over the Efficiency of a Machine Learning Algorithm. Complexity, 2019.

Dolatabadi, S. H., & Keynia, F. (2017). Designing of customer and employee churn prediction model based on data mining method and neural predictor. 74–77.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. ArXiv Preprint ArXiv:1702.08608.

Došilović, F. K., Brčić, M., & Hlupić, N. (2018). Explainable artificial intelligence: A survey. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 0210–0215.

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. Communications of the ACM, 63(1), 68–77.

Ebrah, K., & Elnasir, S. (2019). Churn prediction using machine learning and recommendations plans for telecoms. Journal of Computer and Communications, 7(11), 33–53.

Escalante, H. J., Escalera, S., Guyon, I., Baró, X., Güçlütürk, Y., Güçlü, U., van Gerven, M., & van Lier, R. (2018). Explainable and interpretable models in computer vision and machine learning. Springer.

Figueroa, R. L., Zeng-Treitler, Q., Kandula, S., & Ngo, L. H. (2012). Predicting sample size required for classification performance. BMC Medical Informatics and Decision Making, 12(1), 1–10.

Fong, R. C., & Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. Proceedings of the IEEE International Conference on Computer Vision, 3429–3437.
Frosst, N., & Hinton, G. (2017). Distilling a neural network into a soft decision tree. ArXiv Preprint ArXiv:1711.09784.
García, S., Fernández, A., & Herrera, F. (2009). Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. Applied Soft Computing, 9(4), 1304–1314.
Gattermann-Itschert, T., & Thonemann, U. W. (2021). How training on multiple time slices improves performance in churn prediction. European Journal of Operational Research, 295(2), 664–674.
Gunning, D., & Aha, D. (2019). DARPA's explainable artificial intelligence (XAI) program. AI Magazine, 40(2), 44–58.
Gürsoy, U. Ş. (2010). Customer churn analysis in telecommunication sector. İstanbul Üniversitesi İşletme Fakültesi Dergisi, 39(1), 35–49.
Han, J., Pei, J., & Kamber, M. (2011). Data mining: Concepts and techniques. Elsevier.
Herman, B. (2017). The promise and peril of human evaluation for model interpretability. ArXiv Preprint ArXiv:1711.07414, 8.
Howley, T., Madden, M. G., O'Connell, M.-L., & Ryder, A. G. (2005). The effect of principal component analysis on machine learning accuracy with high dimensional spectral data. International Conference on Innovative Techniques and Applications of Artificial Intelligence, 209–222.
Hung, C.-K. (2017). Making machine-learning tools accessible to language teachers and other non-techies: T-SNE-lab and rocanr as first examples. 2017 IEEE 8th International Conference on Awareness Science and Technology (ICAST), 355–358.
Komorowski, M., Marshall, D. C., Salciccioli, J. D., & Crutain, Y. (2016). Exploratory data analysis. Secondary Analysis of Electronic Health Records, 185–203.
Lage, I., Chen, E., He, J., Narayanan, M., Kim, B., Gershman, S., & Doshi-Velez, F. (2019). An evaluation of the human-interpretability of explanation. ArXiv Preprint ArXiv:1902.00006.
Lakkaraju, H., Bach, S. H., & Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1675–1684.
Lavesson, N., & Davidsson, P. (2006). Quantifying the impact of learning algorithm parameter tuning. AAAI, 6, 395–400.
Letham, B., Rudin, C., McCormick, T. H., & Madigan, D. (2015). Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. The Annals of Applied Statistics, 9(3), 1350–1371.
Lian-Ying, Z., Amoh, D. M., Boateng, L. K., & Okine, A. A. (2019). Combined appetency and upselling prediction scheme in telecommunication sector using support vector machines. International Journal of Modern Education and Computer Science, 10(6), 1.
Lipton, Z. C. (2018). The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue, 16(3), 31–57.
Mardani, A., Jusoh, A., Nor, K., Khalifah, Z., Zakwan, N., & Valipour, A. (2015). Multiple criteria decision-making techniques and their applications–a review of the literature from 2000 to 2014. Economic Research-Ekonomska Istraživanja, 28(1), 516–571.
Mittelstadt, B., Russell, C., & Wachter, S. (2019). Explaining explanations in AI. Proceedings of the Conference on Fairness, Accountability, and Transparency, 279–288.
Mohr, F., & van Rijn, J. N. (2022). Learning Curves for Decision Making in Supervised Machine Learning–A Survey. ArXiv Preprint ArXiv:2201.12150.
Momin, S., Bohra, T., & Raut, P. (2020). Prediction of customer churn using machine learning. EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing, 203–212.
Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., & Frossard, P. (2017). Universal adversarial perturbations. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1765–1773.
Moraffah, R., Karami, M., Guo, R., Raglin, A., & Liu, H. (2020). Causal interpretability for machine learning-problems, methods and evaluation. ACM SIGKDD Explorations Newsletter, 22(1), 18–33.
Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. Proceedings of the National Academy of Sciences, 116(44), 22071–22080.
Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 427–436.
Pan, T., Zhao, J., Wu, W., & Yang, J. (2020). Learning imbalanced datasets based on SMOTE and Gaussian distribution. Information Sciences, 512, 1214–1233.
Qureshi, S. A., Rehman, A. S., Qamar, A. M., Kamal, A., & Rehman, A. (2013). Telecommunication subscribers' churn prediction model using machine learning. 131–136.
Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2009). RUSBoost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40(1), Article 1.
Sivasankar, E., & Vijaya, J. (2019). Hybrid PPFCM-ANN model: An efficient system for customer churn prediction through probabilistic possibilistic fuzzy clustering and artificial neural network. Neural Computing and Applications, 31(11), 7181–7200.
Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). SVM parameter optimization using grid search and genetic algorithm to improve classification performance. TELKOMNIKA (Telecommunication Computing Electronics and Control), 14(4), 1502–1509.
Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. ArXiv Preprint ArXiv:1312.6199.
Umayaparvathi, V., & Iyakutti, K. (2016). Attribute selection and Customer Churn Prediction in telecom industry. 84–90.
Vafeiadis, T., Diamantaras, K. I., Sarigiannidis, G., & Chatzisavvas, K. C. (2015). A comparison of machine learning techniques for customer churn prediction. Simulation Modelling Practice and Theory, 55, 1–9.
Vahidi Farashah, M., Etebarian, A., Azmi, R., & Ebrahimzadeh Dastjerdi, R. (2021). A hybrid recommender system based-on link prediction for movie baskets analysis. Journal of Big Data, 8(1), 1–24.
Yang, F., Du, M., & Hu, X. (2019). Evaluating explanation without ground truth in interpretable machine learning. ArXiv Preprint ArXiv:1907.06831.
Ying, W., Li, X., Xie, Y., & Johnson, E. (2008). Preventing customer churn by using random forests modeling. 429–434.