Journal of Engineering and Computer Sciences
Qassim University, Vol. 15, No. 2, pp. 1-21 (July 2022)

A Hybrid Prediction Model for Customer Churn using DBSCAN and Stacking-Based Classifier

Ebrahim, Ali, Mohammed,
Department of Computer Science, FMCS, University of Gezira,
Wad Madani, Sudan, ebrahimyemen7@gmail.com,

Awadallah, M., Ahmed*
Department of Computer Science, FMCS, University of Gezira,
Wad Madani, Sudan, awadallah@uofg.edu.sd,

Gais, Alhadi, Babikir
Department of Computer Science, FMCS, University of Gezira,
Wad Madani, Sudan, gais.alhadi@uofg.edu.sd,

Mohammed A. Saleh
Department of Computer, College of Science and Arts in Ar Rass,
Qassim University, Saudi Arabia, m.saleh@qu.edu.sa,

Mukhtar, M. E., Mahmoud
Department of Information Systems, Faculty of Computer Science and Information Technology
University of Kassala, Kassala, Sudan, mukhtaredris@gmail.com,

(Received 21/02/2023; accepted for publication 26/02/2024)

Abstract: Customer churn is a key concern in numerous companies, including the telecom industry. Decision-makers and business analysts feel that retaining existing consumers is less expensive than acquiring new clients. To provide a retention solution, customer relationship management (CRM) analysts must recognize customers who intend to quit the company and understand patterns of behavior from existing churn customers' data. This paper provides a Hybrid Prediction Model (HPM) that employs classification and clustering approaches to predict client attrition. To pick the key characteristics, the proposed model uses an RFE algorithm, filter method, Boruta algorithm, and correlation matrix, as well as Density-based Spatial Clustering of Applications with Noise (DBSCAN) to discover and eliminate outlier data. In addition, it uses stacking classifier to categorize consumers, tuning threshold to handle data imbalance, and k-mean algorithm to segment churning customers—whom the stacking classifier has classified—into groups to make group-based retention offers. The Telecom dataset is not publicly available due to protect the private information of customers. Thus, the data used in this study was obtained from Openml Website. The data set contains 5000 observations and 21 variables. Moreover, the proposed model can be easily adapted and applied on larger datasets. Several metrics are used to evaluate the proposed hybrid prediction model, including accuracy, recall, precision, receiving operating characteristics (ROC) area, and f-measure. The results show that our hybrid model outperforms single techniques. Furthermore, when changing the threshold in terms of recall metrics, the model performs better.

Keywords: Data mining, K-mean clustering algorithm, DBSCAN clusters, Stacking, Churn customers.

1. INTRODUCTION

Companies across all industries are focusing on data investment to implement new policies or make strategic decisions that better serve their customers to obtain a competitive advantage and achieve success. Supermarkets, for example, remodel their layout to increase sales, while telephone companies devise new tariff structures to tempt people to make more calls. Both responsibilities necessitate the analysis of past data on client consumption habits to establish the patterns of making such strategic judgments. For many years, statisticians manually "mined" databases seeking for statistically significant patterns, but the volume

and variety of data now greatly exceed the capacity of manual analysis, leading to the application of data-mining techniques and data science principles in business [1].

Data mining is the process of acquiring knowledge by extracting and detecting hidden patterns and relationships in datasets using mathematical, statistical, machine learning, and artificial intelligence approaches [2]. Data mining employs specialized software tools that enable businesses to examine and study data to uncover data content relationships, define a comprehensive and consistent set of customer profiles to better understand customer behavior, needs, and profitability, and build models to predict future customer behavior [3].

Churn rate is an important factor in many professions and businesses, particularly in the phone service market, where mobile phone firms fight for clients, making it easier for customers to switch from one company to another. The wireless telecom industry is experiencing a major issue of customer churn, with telecommunications businesses losing 27% of their clients each year, resulting in massive financial losses and the loss of the company's reputation and brand [4]. It is also well known that getting a new customer costs five to six times as much as retaining an existing customer [5]. Poor customer service is one of the many reasons for churning. To find and keep customers, organizations must develop a more complete and accurate churn prediction model, which is less expensive than recruiting new customers. As a result, the most difficult task for CRM is to retain existing clients and lower churn rate because neglect may result in major profit reduction. The primary goal of the churn prediction model is to identify these consumers so that suitable retention efforts may be implemented, and the organization can benefit by increasing total revenue [6].

In the telecommunications industry, machine learning algorithms have demonstrated excellent effectiveness in predicting customer attrition. Furthermore, when compared to other classification models, ensemble learning techniques have proven to be effective at predicting customer attrition. However, machine learning algorithms encounter severe challenges, such as an imbalanced distribution of classes in data, where the number of churner customers is typically substantially lower than the number of non-churner customers, and outlier data, which can affect accuracy. The hybrid machine learning approaches for predicting customer churn that include clustering and classification algorithms can increase the effectiveness of a single classifier algorithm by processing data by removing outliers before completing the classification task.

The purpose of this study is to construct a predictive model that will aid telecom operators in predicting consumers that are likely to churn. Optimizing predictive model performance by removing outliers using DBSCAN-based outlier detection and resolving dataset imbalance by tuning threshold to improve predictive model performance.

The remainder of this work is structured as follows: Section 2 presents works that are related. The methodology is described in Section 3. Section 4 presents the acquired results as well as a discussion. Finally, Section 5 concludes this paper.

2. RELATED WORKS

Previous efforts in the telecoms sector have constructed churn prediction models using a single classification strategy or by incorporating clustering and classification algorithms to assist enterprises in recognizing, forecasting, and retaining churning consumers, hence assisting in decision making and CRM. Using the supplied dataset, [7] presented and assessed two prediction models based on decision trees and logistic regression. These approaches give explanations for why consumers churn, as well as a list of customers who are prone to churn.

To handle the customer attrition problem in a Chinese telecom firm with around 5.23 million consumers, He et al. [8] developed a hybrid prediction model based on RBF neural network and Analog Complexion Cluster. The RBF neural network predicts customer churn, and the AC cluster is used to separate churn consumers into different categories based on RBF neural network prediction findings to aid in marketing and related operations. The overall accuracy percentage of 91.1% was used as the standard for forecast accuracy.

Liao and Chueh [9] employed fuzzy logic to offer an effective marketing plan by analyzing prior records of marketing activity results. This marketing model assists telecom service providers in determining the appropriate marketing techniques for various client segments to efficiently reduce customer churn.

Qureshi et al. [10] used a variety of machine learning approaches and algorithms to identify clients at risk of churning, including logistic and linear regression, K-Means clustering, Artificial Neural Networks, and Decision Trees such as Exhaustive CHAID, CHAID, QUEST, and CART. This study examined a dataset of 106,000 consumers, with 94.1% of them being active users and only 5.9% churners, and several re-sampling approaches were applied to address the imbalance class problem. After developing the models, the precision, recall, and F-measure are calculated and compared. According to the results, the Exhaustive CHAID algorithm is the best.

Gaur and Dubey [11] proposed a variety of machine learning approaches and algorithms to assess customer attrition, including Logistic Regression, SVM, Random Forest, and Gradient boosted tree. According to the data, gradient boosting is the most accurate model, with an AUC value of 84.57%, Logistic regression and Random Forest are average, with AUC values of 82.86% and 81.26%, respectively, while SVM is the least accurate model, with an AUC value of 79.75%.

Pamina et al. [12] suggested three prediction models, KNN, Random Forest, and XGBoost, to improve the accuracy of customer churn prediction. In reality, a comparison study was undertaken between these models to choose the optimum model with the highest accuracy. The attributes that have the greatest

influence on customer attrition were then selected using the best classifier. The authors demonstrated that the XGBoost model outperforms the KNN and Random Forest models, and that Fiber Optic clients with higher monthly prices had a bigger influence on turnover.

Jain et al. [13] suggested two models (Logistic regression and Logit Boost) for forecasting customer turnover using a database from Orange, an American telecommunications firm, with 3333 occurrences. These models are evaluated using a variety of performance metrics. The authors revealed that there was no significant difference in the performance of the proposed models, with the logistic regression showing an accuracy of 85.2385% and the Logit Boost showing an accuracy of 85.1785%. It should be highlighted that the majority of research focuses on generating more accurate models rather than the significance of data pre-processing.

Furthermore, numerous research presented hybrid machine learning approaches to improve model performance by integrating two or more algorithms, one of which is utilized to analyse data before conducting the classification task. Some writers recommended grouping the data into numerous groups and then deleting some tiny clusters as a method of filtering out unrepresentative data, and they demonstrated considerable improvements in classification accuracy.

Hudaib et al. [14] introduced three two-phase hybrid models (the clustering phase and the prediction phase). A clustering algorithm clearly splits data into distinct categories, and the two biggest groups, which represent non-churners and churners, are combined and utilized as input to the next step, which predictions customer behavior. Small groups, on the other hand, are eliminated because they show unrepresentative behaviors and outliers. The first model combines hierarchical clustering with MLP-ANN, the second with the k-means method, and the third with self-organizing maps (SOM) and MLP-ANN. The authors demonstrated that the three hybrid models outperformed the standard single models.

To address the obstacles of predicting hybrid large-scale datasets and boost the accuracy of customer churn prediction, the authors of [15] used a hybrid model combining fuzzy K-Prototypes (FKP) and support vector machine (SVM). To cluster the mixed attributes, a fuzzy K-Prototypes technique was applied. Then, in each cluster, samples close to the cluster center are chosen as the input of SVM to tackle the problem of SVM classification accuracy for large-scale data.

Tsai and Lu [16] suggested two hybrid churn prediction models: ANN+ANN and SOM+ANN. The first algorithm is designed to remove outliers or unrepresentative data, and the outputs are utilized to train the second algorithm and develop the prediction model. In terms of prediction accuracy, Type I and Type II errors, the authors demonstrated that hybrid models outperform the baseline neural network model, and that the ANN + ANN hybrid model outperforms both the baseline model and the hybrid model SOM + ANN.

3. METHODOLOGY

This section goes into detail about the proposed churn prediction model. Fig. 1 depicts the proposed model for predicting client attrition. The proposed prediction model includes several processes, such as data pre-processing, which includes removing missing-value data and normalizing the data to transform it into an appropriate format, extracting the important attributes, detecting and cleaning outlier data based on DBSCAN, applying the stacking model to classify customers as churn or non-churn, and tuning the prediction threshold to address data imbalance. Using the k-means clustering method to divide churn-predicted customers into groups representing distinct behaviors and generating a suitable retention policy for each group to maximize firm revenues.

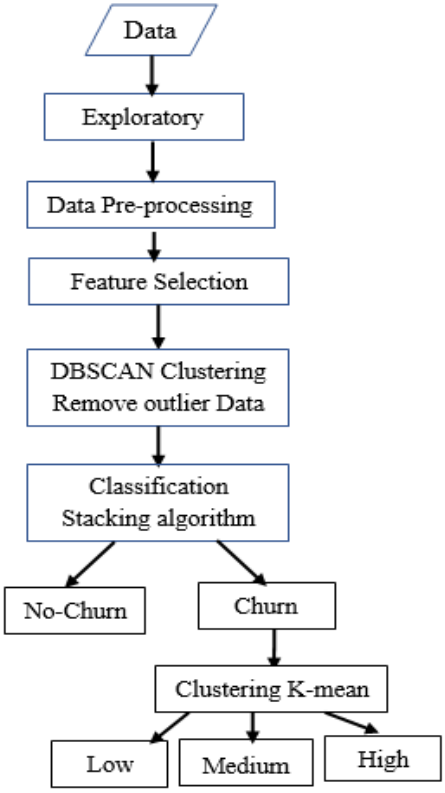


Fig. 1: The suggested model's methodology for predicting customer attrition

3.1. Data Set Selection

In this paper, we used a public dataset accessible on the Openml Website [17], which contains 5000 instances and 21 characteristics. For churned consumers, the target feature has a value of "1" while for non-churned customers, it has a value of "0." (See Table 1).

Table 1: Description of the dataset

Instances	Attributes	Target Class
5000	21	1 → Represent churn
		0 → Represent non-churn

3.2. Exploratory Data Analysis

The initial stage in developing a model is exploratory data analysis (EDA). The exploratory aspect implies that as you move, your understanding of the problem you are or may be solving alters [18]. Clearly, in this step, we explore the dataset and look at the data to understand it; to ensure the correctness of data and ready to use (ensure that the data is on the scale you expect); to understand hidden patterns within the data; to identify features with missing values; and to discover relationships between variables.

It should be mentioned that visualizations such as histograms, bar graphs, and scatter plots were utilized to explore data utilizing strong Python libraries such as Matplotlib, NumPy, Pandas, and Seaborn. It is worth noting that the dataset does not contain any categorical variables; all of the values are numerical, and there are no missing values. As a result, the values are complete, and there is no need to change their format or deal with missing data during the pre-processing step. The desired variable in our dataset is the "class" column, where about 86% of customers stay and 14% churn (See Fig. 2). As a result, it is obvious that the data set is unbalanced.

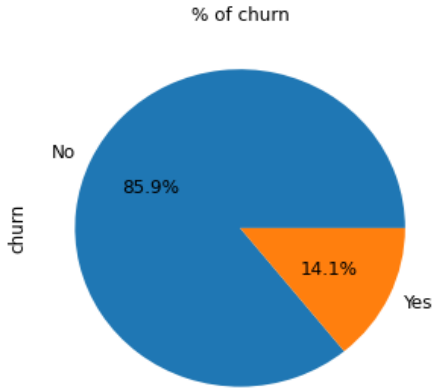


Fig. 2: The percentage of consumers who churn and do not churn

3.3 Data Pre-processing

In general, data pre-processing procedures denote the deletion, insertion, or change of training set data. Data preparation can make or break a model's prediction ability [19]. As a result, raw data is prepared and changed in this phase into a more intelligible and legible format, making it more suited for modeling. A substantial process, such as feature scaling, is used to accomplish this.

3.3.1. Feature scaling

The dataset comprises qualities with a broad variety of magnitudes, units, and ranges, and these attributes are evaluated on different scales; as a result, if the Euclidean distance computation is utilized directly, the influence of certain attributes may be completely negated by others with larger measurement scales. As a result, we must scale all of the properties to the same scale. When the input attributes have different scales, machine learning algorithms do not operate effectively [20]. Feature scaling is one of the most significant transformations we must do to data. The standardization method is used, which rescales the dataset's value distribution so that the mean of observed values is zero and the standard deviation is one. Individually,

each input variable is rescaled by subtracting the mean (to guarantee that standardized values always have a mean of zero) and dividing by the standard deviation, yielding standardized values with a common standard deviation of 1. The standardization approach is utilized, in which the value distribution of the dataset is rescaled so that the mean of observed values is zero and the standard deviation is one.

3.3.2. Feature selection

In this phase, we will remove non-informative or redundant predictors (features) from the dataset. There may be benefits to deleting predictors before training the model. To begin with, fewer predictors mean less complexity and processing time. Second, if there is a connection between two predictors, it indicates that they give the same information and that one of them should be deleted without compromising the model's performance to obtain a more parsimonious and explicable model. Third, without the problematic variables, there can be a significant improvement in model performance and/or stability [19].

It should be emphasized that three separate approaches were employed to undertake feature selection, identifying the common features obtained from these methods, and deleting remaining features to ensure the correct elimination of non-informative or redundant predictors. The correlation matrix between the selected common features was then performed to determine the correlation between them, and if any two features have a strong correlation, one of them would be deleted.

3.3.3. Elimination of the recursive feature

The Recursive Feature Elimination (RFE) method removes attributes iteratively and builds a model on those that remain. The model's accuracy is used to rank the qualities in order of relevance. In fact, the number of attributes to be chosen is a critical parameter, but we cannot determine the appropriate number of attributes to select using RFE. As a consequence, the optimal number of features is established first by selecting different numbers of features from the dataset ranging from 2 to 20 and assessing them using k-fold cross-validation with Random Forest Classifier to identify the number of features with the highest accuracy.

Fig. 3 depicts the RFE accuracy based on the number of features. As we can see, performance improves as the number of features increases, with the best accuracy at number 10 being the ideal amount of features.

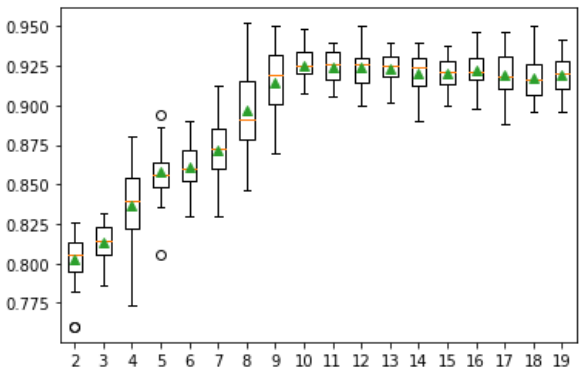


Fig. 3: Depicts the RFE accuracy based on the number of features

3.3.4. Filtering method

This approach use statistical techniques to screen and assess the association between the target variable and each input variable on the training dataset, and then returns the scores to determine the strength of the relationship [19]. These scores are used to filter and choose the variables that will be included in the model.

3.3.5. Boruta algorithm

Boruta [21] is a feature selection algorithm based on statistical foundations. The features in this algorithm do not compete with one another. Instead, they compete with shadow features, which are random copies of themselves. The concept is that a feature is selected if it outperforms the best random feature. Obviously, we will employ the Borutapy package. The most significant parameters in this package are an estimator and max iter to determine the key traits. To begin, we will utilize the random forest as an estimator, with max iter set to 20 iterations. The key features will then be saved in Boruta. Table 2 summarizes the features that were chosen for each strategy.

Table 2: Features that were selected in each method

NO	Feature	Filter	RFE	Boruta	Total
1	total_intl_minutes	True	True	True	3
2	total_eve_minutes	True	True	True	3
3	total_eve_charge	True	True	True	3
4	total_day_minutes	True	True	True	3
5	total_day_charge	True	True	True	3
6	number_vmail_messages	True	True	True	3
7	number_customer_service_calls	True	True	True	3
8	international_plan	True	True	True	3
9	voice_mail_plan	True	False	True	2
10	total_night_minutes	False	True	True	2
11	total_intl_charge	True	False	True	2
12	total_intl_calls	False	True	True	2
13	total_night_charge	False	False	True	1
14	total_night_calls	False	False	False	0
15	total_eve_calls	False	False	False	0
16	total_day_calls	False	False	False	0
17	state	False	False	False	0
18	phone_number	False	False	False	0
19	area_code	False	False	False	0

20	account_length	False	False	False	0

3.3.5. Correlation between independent variable

In this stage, we will compute the correlation matrix between independent variables to determine their correlation with any two variables that have a strong correlation, and we will delete one of them to prevent data with strongly linked predictors. In fact, redundant predictors frequently add more complexity to the model than they give.

It should be noted that the correlation matrix between independent variables was created and the correlations between them were discovered. The correlation matrix between independent variables is shown in Fig. 4, and as can be seen, there is a strong correlation between the following variables: [total day charges and total day minutes], [total eve minutes and total eve charges], [total night minutes and total night charge], and [total intl minutes and total intl charge]. As a result, the variables [total day minutes, total eve minutes, total night minutes, and total international charge] were removed from the dataset.

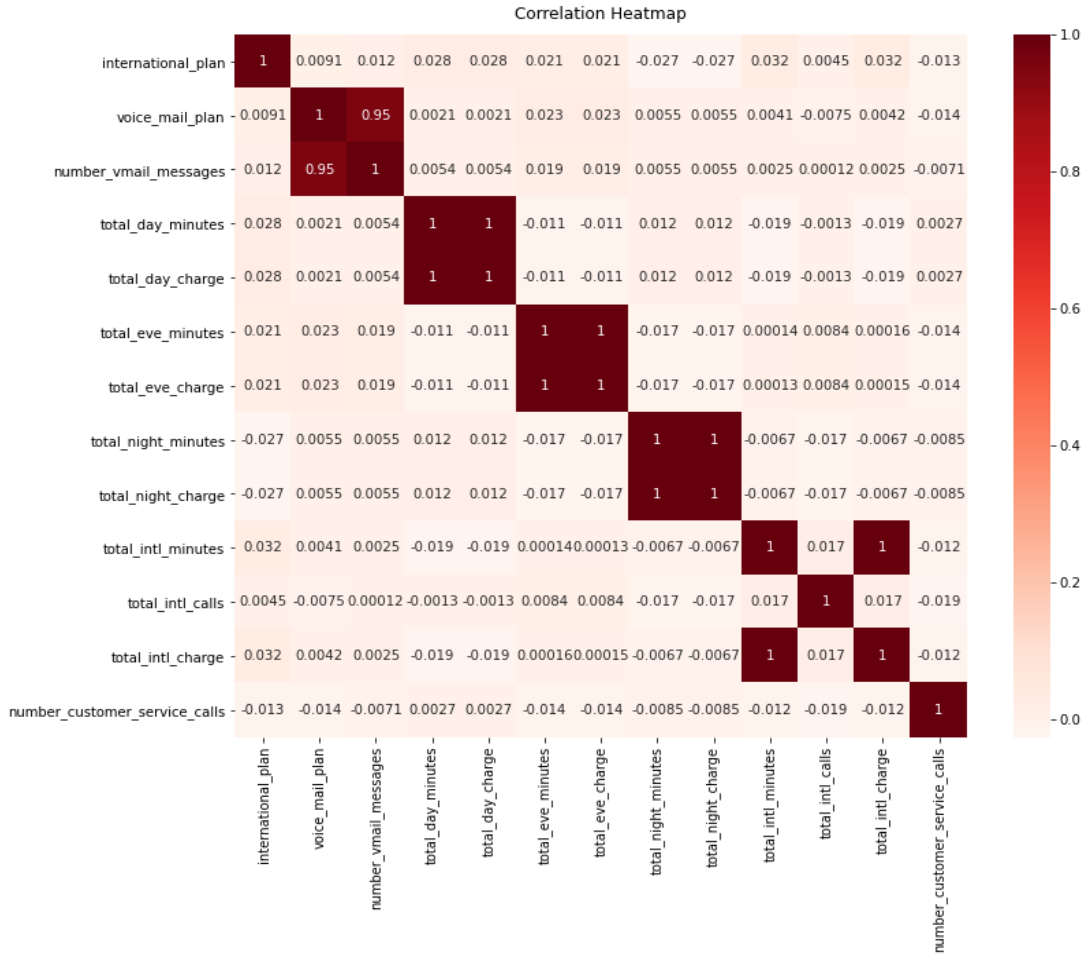


Fig. 4: Depicts the correlation matrix among independent variables

3.4 Outlier Detection Based on DBSCAN

Detecting and cleaning outlier data before to training the model is crucial to ensuring that the observations best represent the problem and, as a result, producing more accurate predictions [22]. The DBSCAN [23]

algorithm is used in this step to find and remove outlier data. The goal is to discover sites that are in "packed" zones, also known as dense regions. Outliers are points that are located outside of the dense zones. To find clusters and outliers, DBSCAN necessitates the use of two parameters: eps and minimum points (MinPts). It should be noted that the eps parameter reflects the distance utilized to establish whether or not a data point is in the same area as the other data points (setting eps to be small means a large part of points will be considered as outliers, setting eps to be large means that more points will be included into the cluster). MinPts reflect the minimal number of data points that must be grouped together for a region to be considered high-density (as min samples increase, fewer points will be labeled as core points and more points will be labeled as noise).

DBSCAN is explained in Algorithm 1. For minPts, minPts must be larger than or equal to the dataset's number of dimensions. In contrast, the dataset for eps is converted from high dimension to two dimensions using Principal Component Analysis (PCA) techniques. The distance between a data point and its nearest data points is then calculated for each data point in the dataset. Plotting these k-distances yields the "knee," which corresponds to the ideal eps parameter. The optimal value for eps is at the point of maximum curvature in the K-distance graph, which appears to be approximately 0.15 in Fig. 5. With eps set to 0.15 and min samples set to 5, there were 5 clusters and over 115 data points were considered outliers/noise. Fig. 6 depicts the outcome of grouping datasets plotted in a two-dimensional graph. DBSCAN, as seen in the graph, grouped the data points into five groups and discovered noise in the dataset. It should be noted that we deleted all of the outlier data from the dataset before training the model with the remaining data. Table 3 contains a description of the dataset once it has been prepared.

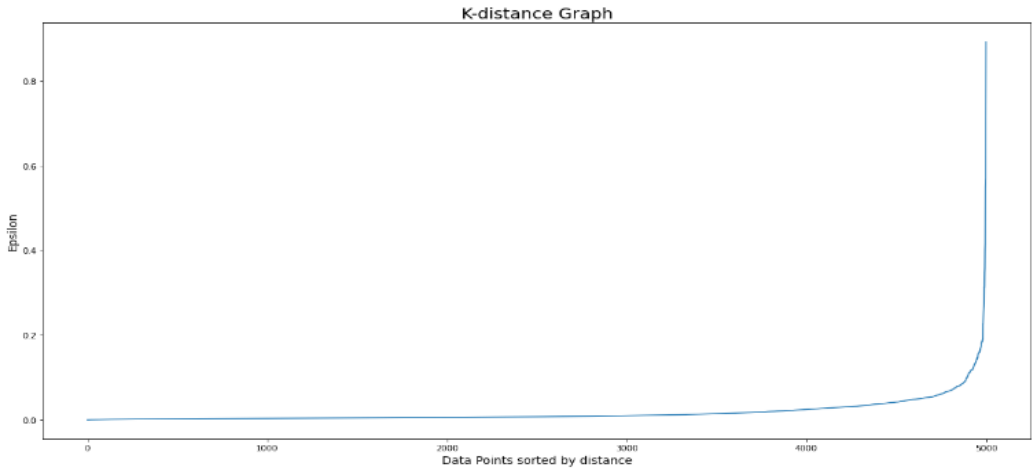


Fig. 5: Shows the K-distance Graph and optimal eps value

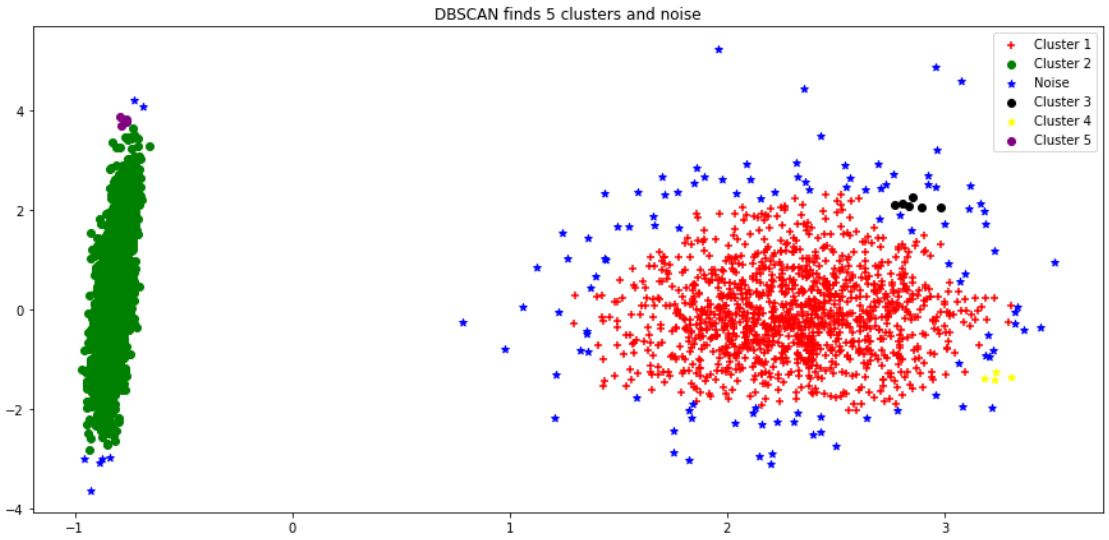


Fig. 6: The DBSCAN clusters.

Algorithm 1: The pseudo-code of the proposed technique DBSCAN to detect and remove outlier data

Input	Dataset X, minimum distance between points to be considered close "eps", number of close points needed to define a cluster "minPts".
Output	Number of clusters and outlier data
Procedure	$DBSCAN(X, eps, minPts)$ $\quad$ **for** each unvisited point x in data X **do** $\quad\quad$ mark x as visited $\quad\quad$ $nieghbrPts = GetNeighbors(x, eps)$ $\quad\quad$ **if** $sizeof(nieghbrPts) < minPts$ **then** $\quad\quad\quad$ mark x as NOISE $\quad\quad$ **end** $\quad\quad$ **else** $\quad\quad\quad$ $ExpandCluster(x, nieghbrPts, eps, minPts)$ $\quad\quad$ **end** $\quad$ **end for** $ExpandCluster(x, nieghbrPts, eps, minPts)$ $\quad$ add x to new Cluster C $\quad$ **for** each x′ in nieghbrPts **do** $\quad\quad$ **if** x′ is not visited **then** $\quad\quad\quad$ mark x′ as visited $\quad\quad\quad$ $nieghbrPts' = GetNeighbors(x', eps)$ $\quad\quad$ **if** $sizeof(nieghbrPts') >= minPts$ **then** $\quad\quad\quad$ $nieghbrPts = nieghbrPts + nieghbrPts'$ $\quad\quad$ **end** $\quad\quad$ **end** $\quad\quad$ **if** x′ is not memeber of any cluster **then** $\quad\quad\quad$ add x′ to cluster C $\quad\quad$ **end** $\quad$ **end for** $GetNeighbors(x, eps)$ **return** all points within x's $eps - neighborhood$ $(including\ x)$

Table 3: Dataset before and after pre-processing stage

	Instances	**Attributes**
Original Dataset	5000	21
Dataset after preparing Data	4885	10

3.5 Applying Stacking Classifier

The preceding step's dataset is divided into a training set and a testing set. The training set's division proportion is 80%, while the testing set's division proportion is 20%. Stacking Ensemble Machine Learning is used on the training data to learn how to identify consumers as non-churners or churners. In actuality, the stacking model has two levels: the base-model, which has two or more classifiers trained on the training dataset and whose predictions are assembled, and the meta-model, which contains a classifier that combines the base model's predictions. The major reasons for utilizing stacking are efficiency and accuracy, because the classifiers used to form the stacking model generate errors after being trained on a batch of data. However, the errors produced by different classifiers are not always the same [24]. Stacking aims to determine which classifiers are reliable by utilizing the meta learner to determine how to best mix the outputs of the base learners [25]. Algorithm 2 provides the stacking method's pseudocode.

Cross-validation is used to test the Random Forest, Logistic Regression, k-Nearest Neighbors, Decision Tree, Support Vector Machine, and Naive Bayes models on the training data [26]. High-performance models are chosen to generate the base model, and Logistic Regression is used to build the meta-model in the stack model. The stacking model is then reviewed and compared to the models that were utilized to construct it.

Due to the imbalance in the dataset, a stratified k-fold cross-validation is employed for evaluation, where it is ensured that the ratio of positive to negative cases discovered in the original distribution is respected in all folds [27]. Because the dataset is tiny and unbalanced, we will use K = 5 [28]. It should be highlighted that in the class that is underrepresented in the data sample, there is an interest in and bias toward model performance. Because typical measures such as classification accuracy or classification error are not sensitive to skewed domains, various assessment metrics like as accuracy, recall, Precision, F-Measure, ROC-AUC, and confusion matrix are employed in conjunction with stratified k-fold cross-validation. As a result, it is untrustworthy and deceptive [29].

Algorithm 2: The pseudo-code of the Stacking algorithm	
Input	Training Dataset $\{(X_1, y_1), (X_2, y_2), \dots, (X_m, y_m)\}$ Base learning algorithms $L_1, L_2, \dots, L_T$ Meta learning algorithm L

Processing	**Step 1**: Learn first − level classifiers
	for $t = 1,2,\ldots,T$ ***do***
	$h_t = L_t(D)$ # learn a base classifier h_t based on D
	end for
	***Step* 2**: Construct new data set from D
	for $i = 1,2,\ldots,m$ ***do***
	for $t = 1,2,\ldots,T$ ***do***
	$\boldsymbol{z_{it}} = h_t(x_i)$ # *Use* h_t *to classify the training example* x_i
	end for
	$\boldsymbol{D'} = \boldsymbol{D'}((\boldsymbol{z_{i1}}, \boldsymbol{z_{i2}}, \ldots, \boldsymbol{z_{iT}}), \boldsymbol{y_i})$
	end for
	***Step* 3**: Learn a second − level classifier
	$h' = L(D')$ # *train a meta classifier* h' *based on the newly constructed data* D'

Output	$H(x) = h'(h_1(x), h_2(x), \ldots, h_T(x))$

3.6 Threshold Tuning

A decision threshold parameter governs the choice to classify a client as churn or no-churn; the default value for the threshold is 0.5. Prediction probability values greater than or equal to 0.5 are allocated to class 1 (churn), whereas prediction probability values less than 0.5 are assigned to class 0. (no-churn). Using classifiers without adjusting the output threshold can be a critical mistake because the class distribution is severely skewed towards the positive class (minority), and this skewness leads to the creation of many false-negative predictions, which reduces the model's performance on the positive class (minority) compared to its performance on the majority class (negative), and the cost of one type of misclassification is more significant than another type of misclassification [30].

As more consumers are categorized as positive, lowering the threshold reduces the false-negative rate while increasing the true-positive rate. A higher threshold, on the other hand, reduces the false positive rate while increasing the actual negative rate since fewer consumers are categorized as positive (Churn) [16]. The ROC curve is then utilized to establish the ideal threshold that yields greater true positive rates, in which the model is assessed using a range of thresholds and these thresholds are plotted on the ROC curve. The G-mean measure is then applied to determine the best threshold and plot it on the ROC curve (He and Ma, 2013) [31].

3.7 Customer Profiling and Retention through Clustering

It is worth noting that decision makers are interested in identifying churner behavior and implementing a suitable retention strategy to optimize the company's earnings. As a result, in this study, we are concerned with consumers who will leave the organization, therefore churner clients will be chosen and their actions will be analyzed. The K-mean clustering technique is clearly used to separate consumers into groups based on their behavior, detecting comparable patterns and characteristics for each category to build policies to keep churn clients.

4. RESULTS AND DISCUSSION

The results will be presented in this section. It is worth noting that we used 5-fold cross-validation to assess Random Forest, Support Vector Machine, Decision Tree, k-Nearest Neighbors, Logistic Regression, and Naive Bayes models on the training dataset to create the staking model. As foundation models, we selected the classifiers with the highest accuracy. The accuracy of these classifiers is shown in Table 4. The findings showed that the Random Forest, Support Vector Machine, Decision Tree, and k-Nearest Neighbors performed the best, with accuracy of 95.6%, 93.6%, 91.8%, and 91.6%, respectively, therefore they were chosen as base models, while Logistic Regression was utilized as a meta-model.

Table 4: Average accuracy over the 5-folds of six classifiers using the selected variables and values

Model	Accuracy %
Random Forest	95.6
Support Vector Machine	93.6
Decision Tree	91.8
k-Nearest Neighbours	91.6
Logistic Regression	86.8
Naive Bayes	86.0

Furthermore, the suggested model was tested on the training dataset and compared to the classifiers employed in the base model. Table 5 details the performance of the proposed model and various classifiers. Our proposed model outperformed the other models, according to the results (i.e., Random Forest, Support Vector Machine, Decision Tree, and k-Nearest Neighbors). We can plainly observe this because we outperformed the other models with an accuracy of 95.9%, a recall of 77.8%, and an F-measure of 84.2%.

Table 5: The performance of our stacking model and other models that are used as base models with 5-fold cross-validation

Model	Accuracy %	Recall %	Precision %	F-measure %
Random Forest	95.6	74.9	93.2	83.0
Support Vector Machine	93.6	61.0	90.9	73.0
Decision Tree	91.8	74.0	70.1	72.0
k-Nearest Neighbours	91.6	48.2	86.7	61.9
Stacking	95.9	77.8	91.9	84.2

The predictions on the testing set were made to determine the prediction model's performance using the default threshold. It should be noted that, as we have shown in previous sections, there is an imbalance in the dataset, and we care about the customers who leave, since the true positive rate and recall are equal, therefore we have tweaked the threshold using the ROC curve to solve this imbalance and get a greater recall rate.

Table 6 shows the detailed performance of the stacking model using the default threshold and optimal threshold. Also, in Fig. 7 and Fig. 8, we introduce the confusion matrix of the stacking model when applied to the test dataset using the default threshold and tuning threshold. Moreover, the ROC curve with a range of thresholds (0.1 to 0.5) and with the optimal threshold, have been presented in Fig. 9 and Fig. 10. The results showed that the proposed prediction model with the optimal threshold (0.346) achieved the highest

performance (true positive rate or recall up to 91.4%), as compared to the proposed prediction model with the default threshold.

Table 6: The stacking model's performance using the default and optimal thresholds

	Model with default Threshold	**Model with best Threshold**
Accuracy %	98.3	97.9
Recall %	89.1	91.4
Precision %	97.4	92.1
F-measure %	93.1	91.8
ROC AUC	94.4	95.1

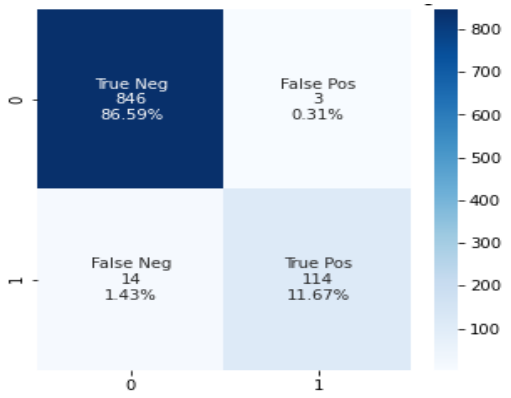


Fig. 7: The confusion matrix of the stacking model when applied to the test dataset using the default threshold

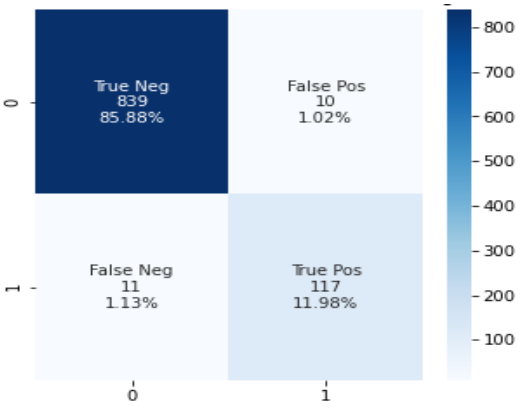


Fig. 8: The confusion matrix of the stacking model when tuning the threshold

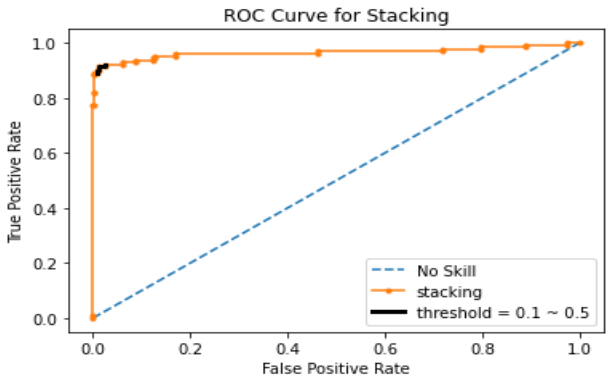


Fig. 9: The ROC curve for stacking with multi thresholds from 0.1 to 0.5

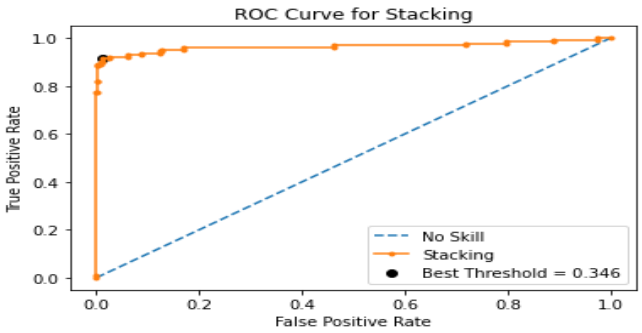


Fig. 10: The ROC curve for stacking with the best thresholds

4.1 Effect of DBSCAN and Threshold Tuning

The effect of DBSCAN-based outlier identification and threshold tweaking on the performance of the stake classifier is presented in this section. The original dataset was split into two parts: training and testing. The stacking classifier was then fitted to the training set, and the testing set was predicted. Table 7 shows how the stake classifier performed before and after using DBSCAN to find and eliminate outlier data. In addition, Fig. 11 depicts the confusion matrix of the stacking model without DBSCAN. As we can see, the accuracy, recall, precision, F-measure, and ROC AUC of the stacking classifier for the original dataset are 96.7%, 82.5%, 90.4%, 86.3%, and 90.6%, respectively. After removing the outlier data, the accuracy, recall, precision, F-measure, and ROC AUC of the stacking classifier are 97.9%, 91.4%, 92.1%, 91.8%, and 95.1%, respectively.

Table 7: Stacking model performance with and without DBSCAN

	Stacking with DBSCAN	**Stacking without DBSCAN**
Accuracy %	98.3	96.7
Recall %	89.1	82.5
Precision %	97.4	90.4
F-measure %	93.1	86.3
ROC AUC %	94.4	90.6

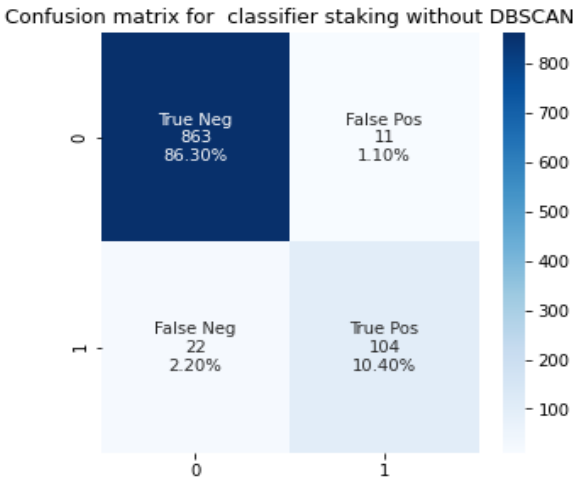
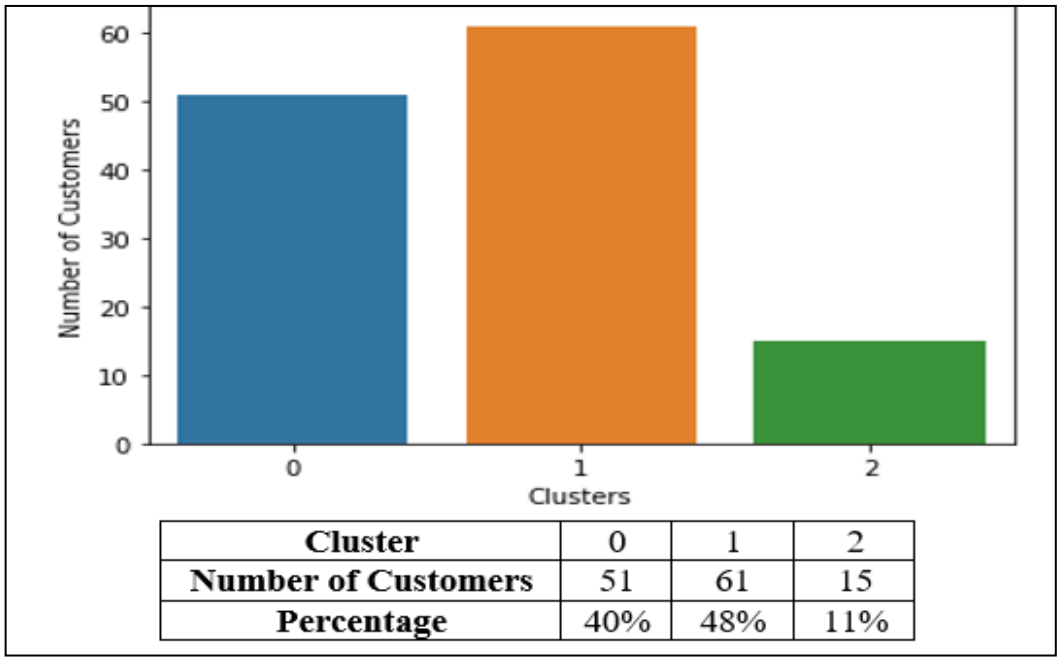


Fig. 11: The confusion matrix for stacking model without DBSCAN

In summary, we can conclude that using DBSCAN-based outlier identification on the stacking classifier will increase the model's performance. Furthermore, as demonstrated in Table 6, tweaking the threshold can improve the proposed classifier's performance in terms of recall and ROC AUC.

4.2 Clustering Output

The k-means algorithm classified customers into three categories: risky, which has the greatest percentage of churners; medium, which has a moderate percentage of churners; and low, which has a negligible percentage of churners. Figure 12 highlights the churner segmentation, demonstrating that clusters 0 and 1 have a large percentage of churners, with 40% and 48%, respectively, compared to cluster 2, which has just 11%. As a result, the corporation can keep these two groups to maximize profits. To comprehend the behavior of customers in each group, similar patterns and characteristics for each group have been discovered; Figure 13 depicts the behavior of each group with various features. The threshold value for each attribute from each clustering was retrieved to create rules for the future suggestion of only similar consumers. Win-back campaigns may be developed for each class depending on its classification, and they can give a loyalty program, a special offer, or a package as a reward to ensure client retention by targeting the appropriate class for retention.



Cluster	0	1	2
Number of Customers	51	61	15
Percentage	40%	48%	11%

Fig. 12: The segmentation of churn customers

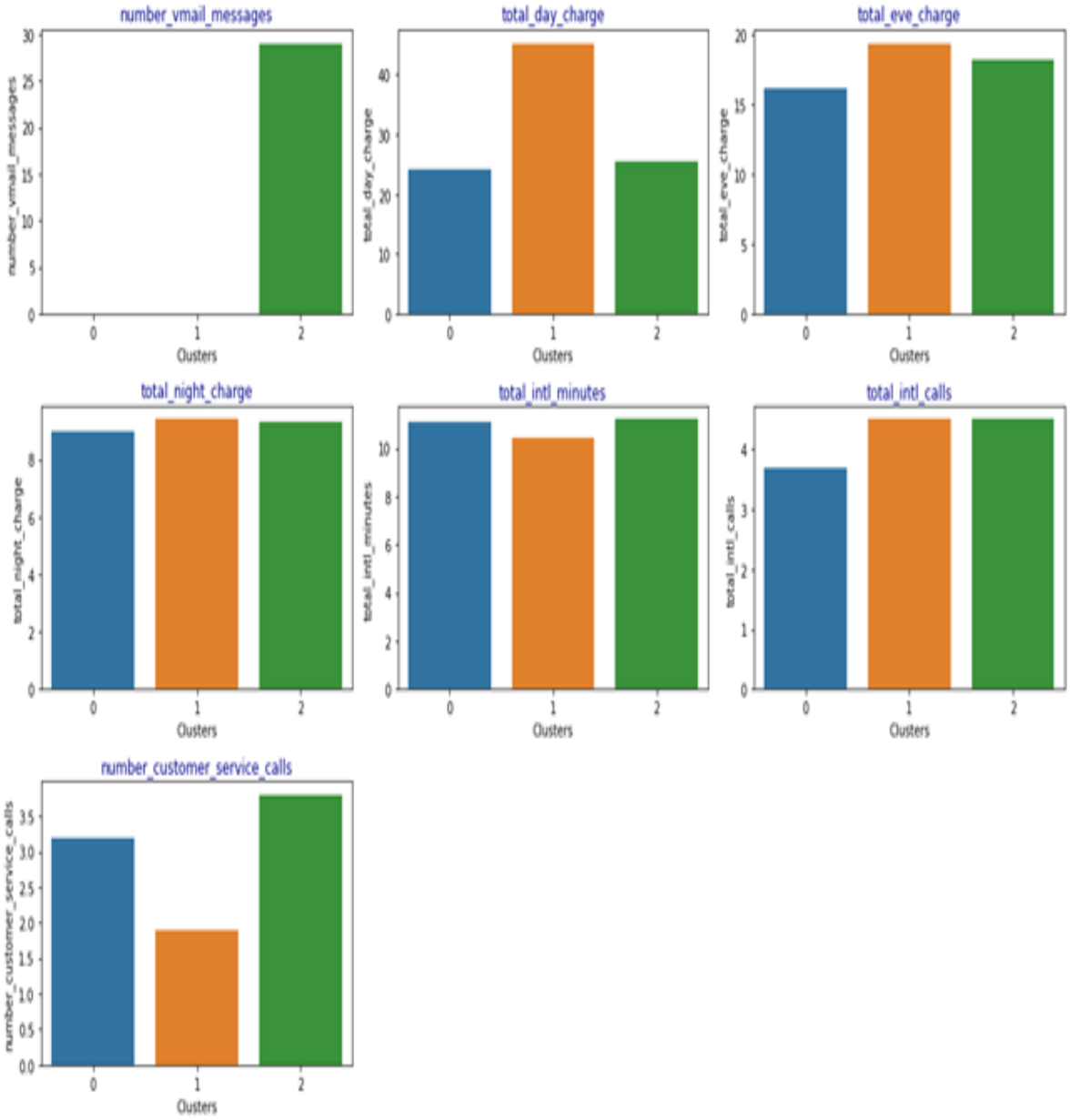


Fig. 13: Attributes behavior in each cluster

5. CONCLUSION

Since customers are telecom businesses' primary competitive advantage, customer churn prediction has become a major issue in the firm's CRM to retain customers and compete with other companies by identifying customers who are likely to quit the company and making them competitive offers. Data mining assists telecom businesses in this view by giving tools for identifying such clients and targeting retention actions. A hybrid prediction model was constructed in this study by merging three distinct approaches: DBSCAN for outlier identification, stacking classifier for classification, and threshold adjustment to address data imbalance. Then, to advise marketing management, we used the K-means clustering method to categorize turnover clients based on the prediction findings of the stacking classifier. Our hybrid model

outperformed the single models, according to the results. Furthermore, when changing the threshold in terms of recall metrics, the model performs better. Different ensemble learning approaches, such as AdaBoost and Bagging, can be used for prediction in future work. Other methods for detecting outliers include one-class SVM (OCSVM), isolation forest (IF), and local outlier factor (LOF). Other approaches, including as re-sampling methods and cost-sensitive learning, can be employed to address the imbalance in the dataset.

REFERENCES

[1] Fávero LP, Belfiore P. Data science for business and decision making. Academic Press; 2019.

[2] Yie LF, Susanto H, Setiana D. Collaborating Decision Support and Business Intelligence to Enable Government Digital Connectivity. InWeb 2.0 and Cloud Technologies for Implementing Connected Government 2021 (pp. 95-112). IGI Global.

[3] Kumar V, Reinartz W. Customer relationship management. Springer-Verlag GmbH Germany, part of Springer Nature 2006, 2012, 2018; 2018.

[4] Vo NN, Liu S, Li X, Xu G. Leveraging unstructured call log data for customer churn prediction. Knowledge-Based Systems. 2021.

[5] Farquad MA, Ravi V, Raju SB. Churn prediction using comprehensible support vector machine: An analytical CRM application. Applied Soft Computing. 2014.

[6] Bansal G, Anand A, Yadavalli VS. Predicting effective customer lifetime: an application of survival analysis for telecommunication industry. Communications in Statistics-Theory and Methods. 2020.

[7] Dalvi PK, Khandge SK, Deomore A, Bankar A, Kanade VA. "Analysis of customer churn prediction in telecom industry using decision trees and logistic regression," In symposium on colossal data analysis and networking (CDAN), Indore, India, pp. 1-4, 2016.

[8] Shabankareh MJ, Shabankareh MA, Nazarian A, Ranjbaran A, Seyyedamiri N. A stacking-based data mining solution to customer churn prediction. Journal of Relationship Marketing. 2022.

[9] Zdziebko T, Sulikowski P, Sałabun W, Przybyła-Kasperek M, Bąk I. Optimizing Customer Retention in the Telecom Industry: A Fuzzy-Based Churn Modeling with Usage Data. Electronics. 2024.

[10] Kavitha V, Kumar GH, Kumar SM, Harish M. Churn prediction of customer in telecom industry using machine learning algorithms. International Journal of Engineering Research & Technology (IJERT). 2020;9(5):181-4.

[11] Gaur A, Dubey R. "Predicting customer Churn prediction in telecom sector using various machine learning techniques," *In* 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), pp. 1-5, 2018.

[12] Pamina J, Raja B, SathyaBama S, Sruthi MS, VJ A. "An effective classifier for predicting churn in telecommunication," Jour of Adv Research in Dynamical & Control Systems, vol. 11, 2019.

[13] Jain H, Khunteta A, Srivastava S, "Churn prediction in telecommunication using logistic regression and logit boost," Procedia Computer Science, vol. 167, pp.101-12, 2020.

[14] Hudaib A, Dannoun R, Harfoushi O, Obiedat R, Faris H, "Hybrid data mining models for predicting customer churn," International Journal of Communications, Network and System Sciences. Vol. 8, no. 05 pp.91, 2015.

[15] Fa-Gui LI, ZHANG ZJ, Xin YA, "Using Combined Model Approach for Churn Prediction in Telecommunication," In 3rd Annual International Conference on Electronics, Electrical Engineering and Information Science (EEEIS 2017), Guangzhou, China, pp. 269-276, 2017.

[16] Ramesh P, Jeba Emilyn J, Vijayakumar V. Hybrid artificial neural networks using customer churn prediction. Wireless Personal Communications. 2022;124(2):1695-709.

[17] Churn Dataset, accessed 20 December 2022, https://www.openml.org/d/40701.

[18] Qamar U, Raza MS. Data Science Concepts and Techniques with Applications. Berlin/Heidelberg, Germany: Springer; 2020.

[19] Wassouf WN, Alkhatib R, Salloum K, Balloul S. Predictive analytics using big data for increased customer loyalty: Syriatel Telecom Company case study. Journal of Big Data. 2020;7:1-24.

[20] Géron A, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, Inc., 2022.

[21] Anand N, Sehgal R, Anand S, Kaushik A. Feature selection on educational data using Boruta algorithm. International Journal of Computational Intelligence Studies. 2021;10(1):27-35.

[22] Brownlee, J., "Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python," Machine Learning Mastery, 2020.

[23] Sahu RT, Verma MK, Ahmad I. Density-based spatial clustering of application with noise approach for regionalisation and its effect on hierarchical clustering. International Journal of Hydrology Science and Technology. 2023;16(3):240-69.

[24] Liu Z, Zhang X, Niu J, Dezert J. Combination of classifiers with different frames of discernment based on belief functions. IEEE Transactions on Fuzzy Systems. 2020;29(7):1764-74.
[25] González S, García S, Del Ser J, Rokach L, Herrera F. A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. Information Fusion. 2020;64:205-37.
[26] Pastore M, Rotondo P, Erba V, Gherardi M. Statistical learning theory of structured data. Physical Review E. 2020;102(3):032119.
[27] Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B. and Herrera, F., "Learning from imbalanced data sets," vol. 10, pp. 978-3, Cham: Springer, 2018.
[28] Aggarwa, C.C., "Data Classification: Algorithms and Applications," Data Mining and Knowledge Discovery Series, 2015.
[29] Paula, B., Torgo, L. and Ribeiro, R., "A survey of predictive modelling under imbalanced distributions," arXiv preprint arXiv, vol. 1505, no. 01658, 2015.
[30] Tyagi S, Mittal S. Sampling approaches for imbalanced data classification problem in machine learning. InProceedings of ICRIC 2019: Recent Innovations in Computing 2020 (pp. 209-221). Springer International Publishing.
[31] Kulkarni A, Chong D, Batarseh FA. Foundations of data imbalance and solutions for a data democracy. InData democracy 2020 (pp. 83-106). Academic Press.

نموذج هجين للتنبؤ بتسرب العميل باستخدام التجميع المكاني القائم على الكثافة للتطبيقات ذات الضوضاء (DBSCAN) والمصنف القائم على التراص

إبراهيم علي محمد
قسم علوم الحاسوب، كلية العلوم الرياضية والحاسوب
جامعة الجزيرة، السودان
ebrahimyemen7@gmail.com

عوض الله محمد أحمد
قسم علوم الحاسوب، كلية العلوم الرياضية والحاسوب
جامعة الجزيرة، السودان
awadallah@uofg.edu.sd

قيس الهادي بابكر
قسم علوم الحاسوب، كلية العلوم الرياضية والحاسوب
جامعة الجزيرة، السودان
gais.alhadi@uofg.edu.sd

مختار محمد إدريس محمود
قسم نظم المعلومات، كلية علوم الحاسوب وتقانة المعلومات
جامعة كسلا، السودان
mukhtaredris@gmail.com

محمد عباس الأمين صالح
قسم الحاسب الآلي، كلية العلوم والآداب بالرس
جامعة القصيم، المملكة العربية السعودية
m.saleh@qu.edu.sa

ملخص البحث: يعتبر تسرب العملاء مصدر قلق رئيسي للعديد من الشركات، بما في ذلك شركات صناعة الاتصالات. يشعر صانعو القرار ومحللو الأعمال أن الاحتفاظ بالمستهلكين الحاليين أقل تكلفة من اكتساب عملاء جدد. من أجل توفير حل الاحتفاظ، يجب على محللي إدارة علاقات العملاء (CRM) التعرف على العملاء الذين يعتزمون ترك الشركة وفهم أنماط سلوكهم من بيانات العملاء الحالية. يقدم هذا البحث نموذج التنبؤ الهجين (HPM) الذي يستخدم أساليب التصنيف والتجميع للتنبؤ بتناقص عدد العملاء. ولاختيار الخصائص الرئيسية، يستخدم النموذج المقترح خوارزمية RFE، وطريقة التصفية، وخوارزمية Boruta، ومصفوفة الارتباط، بالإضافة إلى التجميع المكاني القائم على الكثافة للتطبيقات ذات الضوضاء (DBSCAN) لاكتشاف البيانات الخارجية والقضاء عليها. بالإضافة إلى ذلك، يستخدم النموذج المقترح مصنف التراص لتصنيف المستهلكين، وضبط العتبة للتعامل مع عدم توازن البيانات، وخوارزمية k-mean لتقسيم العملاء المتخبطين - الذين صنفهم مصنف التراص - إلى مجموعات لتقديم عروض احتفاظ قائمة على المجموعة. في هذا البحث تم استخدام العديد من المقاييس لتقييم نموذج التنبؤ الهجين المقترح، بما في ذلك الدقة، والاسترجاع، والدقة، واستقبال خصائص التشغيل (ROC)، والقياس f. أظهرت النتائج أن النموذج الهجين المقترح يتفوق على التقنيات الفردية. علاوة على ذلك، عند تغيير العتبة من حيث مقاييس الاستدعاء، يعمل النموذج بشكل أفضل.

كلمات مفتاحية: تسرب العملاء، تنقيب البيانات، خوارزمية التجميع k-mean، تجميع DBSCAN، التراص.