http://communications.science.ankara.edu.tr

Commun.Fac.Sci.Univ.Ank.Ser. A1 Math. Stat.
Volume 74, Number 2, Pages 294–317 (2025)
https://doi.org/10.31801/cfsuasmas.1502723
ISSN 1303-5991 E-ISSN 2618-6470

Research Article; Received: June 19, 2024; Accepted: February 24, 2025

A proposed classification method approach for binary variable data using Boolean algebra and an application to digital advertising

Haydar EKELİK[1] and Mustafa TEKİN[2]

[1,2]Department of Econometrics, İstanbul University, İstanbul, TÜRKİYE

Abstract. In this paper, Boolean decision table (BDT) approach is proposed as a new classification technique for binary variables using Boolean algebra. Since the proposed BDT approach is similar to the decision tree methods used in classification analysis, the performance of the BDT approach is compared with the widely used decision tree methods in the literature: classification and regression tree (CART), random forest (RF), and extreme gradient boost (XGBoost) algorithms. While making the comparison, attention was paid to the classification performance of the models (classification accuracy, ROC, and PR curve) as well as the interpretability of the results obtained. The benefits and drawbacks of the proposed BDT approach were analyzed using real data from digital ads of an e-commerce company. The results of the analysis show that the BDT approach outperforms RF and CART algorithms in classification and is close to the XGBoost algorithm. The BDT approach has demonstrated greater validity in the digital advertising industry because, in comparison to the XGBoost algorithm, its results are more interpretable. Furthermore, classification performance was also compared using a future dataset from the same e-commerce company that is not included in the training or test datasets. Important target audiences were identified in addition to classification performance because target audiences are crucial to digital advertising. A multi-criteria decision-making technique called TOPSIS was used to ascertain the relative importance of the target audiences. Both the proposal of the BDT approach and the evaluation of the results of the classification algorithms using the TOPSIS method are considered to contribute to the literature in this field.

2020 Mathematics Subject Classification. 03G05, 62P25, 60G25.
Keywords. Boolean algebra, classification and regression tree, random forest, extreme gradient boost.

1. Introduction

George Boole developed Boolean algebra, which forms the basis of binary mathematics and Boolean logic. It began with his initial publications in the subject of logic, "The Mathematical Analysis of Logic" published in 1848 and "An Investigation of the Laws of Thought" published in 1854 [25]. Boole created an algebra that could assess the validity or falsity of statements made with terms like "and," "or," and "not." Boolean algebra has a binary number system, since it is based on two-valued logic. Variables in boolean algebra are represented numerically as 1 and 0 and orally as "true" and "false," respectively. Based on specific logical procedures, a binary (boolean) output can be generated given an input with k variables using a boolean function with k variables, represented as $B : \{0,1\}^k \to \{0,1\}$ [7]. A truth table can be used to illustrate a Boolean function. In the table, each row corresponds to a combination of the values of the Boolean variables and the corresponding output value. The truth table of a Boolean function with k variables consists of 2^k rows and $k+1$ columns, which include the outcome variable.

[1] haydar.ekelik@istanbul.edu.tr -Corresponding author; 0000-0002-0661-4164.
[2] mustafatek@istanbul.edu.tr; 0000-0002-1169-1463.

Research on Boolean algebra is mainly found in the fields of electrical engineering [19,36,46], computer engineering [32,55], biology [59,61], and mathematics [41,63]. In social sciences, the qualitative comparative analysis (QCA) developed by Charles Ragin is often used, which is based on Boolean algebra and set theory [52]. Studies using qualitative comparative analysis can also be found in the literature [5,6,24,60].

There are a few studies that use Boolean algebra to determine the probabilities of specific statistical events, notwithstanding the paucity of research on statistical and classification analyses with Boolean algebra [49,57]. Studies utilizing Boolean algebra in classification primarily focus on low-observation and medical applications. It has been discovered that procedures based on Boolean rules were more accurate than alternative approaches [37,47]. In a different classification study, a Boolean algebra-based algorithm was suggested that performed better than existing approaches by using feature differences to quickly classify huge datasets and data stream systems [62].

There are also studies in the literature on classification analysis using digital advertising data [1,13, 16,34,42]. In this paper, a classification technique for digital advertising data based on Boolean algebras was proposed. The proposed approach was similar to the decision table algorithm of Ogihara *et al.* [47] and Lu and Liu Grouping and Counting (GAC) Lu and Liu [43], but differs in terms of the technique used to classify the observations and the use of multi-criteria decision methods. The accuracy of the classification in the Ogihara *et al.* investigation was determined exclusively by frequency counts. The accuracy of classification in this study was assessed using the ROC and PR curves. In addition, the best feature (variable, attribute) combinations in the dataset was determined using the multi-criteria decision-making technique known as the technique for order of preference by similarity to ideal solution (TOPSIS) analysis. Along with prediction accuracy, classification model interpretability is also quite important. Interpretability entails concentrating on the classifier model's outputs and providing a more comprehensible interpretation of the classification rules that emerge from the classification analysis [21]. Artificial neural networks, support vector machines, and ensemble learning methods are so-called "black-box" models that evaluate classification accuracy [44]. Furthermore, understanding classification models and being able to interpret combinations of variables gives more confidence that the model identifies the correct patterns. This also helps to deal with the problem of dataset shifting, which occurs when future data has a different distribution than past data [21,51].

In the study that would analyze binary-valued features (variables), the classification tree created with Boolean algebra (Boolean decision table) was compared with other decision tree algorithms in terms of classification performance. The decision to use this method was made because the features in the dataset used in the application were represented in accordance with Boolean algebra logic and the obtained Boolean rules were more interpretable compared to other classification algorithms [21].

The rest of the paper was organized as follows: Section 2 provides a brief introduction to Boolean algebra and decision tree techniques, including the CART, Random Forest, and XGBoost algorithms. Using a sample dataset, the proposed Boolean decision table approach was presented and illustrated. Techniques for evaluating model performance were also briefly covered. Section 3 presents the findings of applying the suggested Boolean decision table approach to decision tree methods with Google Analytics data from an e-commerce firm. The conclusion section in Section 4 provides a general assessment of the research. In this study, variable and feature are used interchangeably.

2. Methods

2.1. **Boolean Algebra, Boolean Functions and Minterms.** Boolean algebra variables are expressed verbally as "true" and "false," and numerically as 1 and 0, respectively. A Boolean function with k variables, denoted as $B : \{0,1\}^k \to \{0,1\}$, is defined, and this function allows a two-valued (Boolean) output to be produced for an input with k variables based on specific logical operations [7]. In another representation, Boolean functions can also be found in the notations of algebraic set operations. A Boolean function $F = f(A_1, A_2, \cdots, A_{(n-1)}, A_n)$, where $A_i, i = 1, \cdots, n$ are sets, is obtained through algebraic set operations such as union (∪, or), intersection (∩, and), and complement (not, c). For example, there are two Boolean functions for sets A, B, and C:

$$F = f_1(A,B,C) = [A \cup (B \cup C^c)]^c$$

$$G = f_2(A,B,C) = (A \cup A^c B)^c$$

In the creation of functions, the superscript "c" expression represents the complement, and A^cB while represents the intersection of sets $A^c \cap B$ as described [49].

Partitions of Sets and Minterms: Let $D = \{A, B, C\}$ denote the class of sets, under the assumptions that sets are not covered by each other and their intersections are non-empty. Based on this assumption, the following set D^0 is constructed.

$$D^0 = \{m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7\}$$

$$\begin{array}{cc} m_0 = A^cB^cC^c \sim 000 & m_4 = AB^cC^c \sim 100 \\ m_1 = A^cB^cC \sim 001 & m_5 = AB^cC \sim 101 \\ m_2 = A^cBC^c \sim 010 & m_6 = ABC^c \sim 110 \\ m_3 = A^cBC \sim 011 & m_7 = ABC \sim 111 \end{array}$$

m_i are called minterms, and the intersection of minterms in the set D^0 is the empty set. When constructing minterms, the complement of the sets is represented by 0 and the sets themselves by 1. This representation is called the binary designator of the minterm m_i. Using the binary designator, the minterms are converted to decimal, and the index of the corresponding minterm is obtained. The binary designator is the binary writing of index values in decimal base. For example, the binary indicator of the minterm m_6 is 110. This is the equivalent of the binary indicator in the decimal system;

$$110 = 1.2^2 + 1.2^1 + 0.2^0 = 6.$$

The resulting number 6 corresponds to the m_6 minterm. Since there may only be a maximum of seven indices, three bases 2^2, 2^1, and 2^1 are employed. In binary notation, the number 7 is represented by at least three base values. Also, the fact that the given example contains 3 clusters determines the base value. If these operations are applied to the cluster class with finite n clusters, the resulting number of minterms is 2^n. Minterms can also be thought of as the intersection of all input sets [49].

Table 1. Binary representation of sets

A ~	1	1	1	1	0	0	0	0
B ~	1	1	0	0	1	1	0	0
C ~	1	0	1	0	1	0	1	0
A^c ~	0	0	0	0	1	1	1	1
B^c ~	0	0	1	1	0	0	1	1
C^c ~	0	1	0	1	0	1	0	1

Table 2 below displays the complements and intersections of the sets whose binary representations are provided in Table 1.

Table 2. Set intersections and minterms

ABC ~	1	0	0	0	0	0	0	0
ABC^c ~	0	1	0	0	0	0	0	0
AB^cC ~	0	0	1	0	0	0	0	0
AB^cC^c ~	0	0	0	1	0	0	0	0
A^cBC ~	0	0	0	0	1	0	0	0
A^cBC^c ~	0	0	0	0	0	1	0	0
A^cB^cC ~	0	0	0	0	0	0	1	0
$A^cB^cC^c$ ~	0	0	0	0	0	0	0	1

In Table 2, a value of 1 in the minterm column $m_i, i = 1, 2, 3, 4, 5, 6, 7$ represents the relevant minterm. For example, for the intersection set ABC, since the m_7 minterm column is 1, this intersection set is represented by the minterm m_7. Here, intersection operations in binary notation are taken into account when creating the clusters.

2.2. **Classification and Regression Tree (CART) Algorithm.** The CART algorithm adopts a greedy top-down iterative divide-and-conquer approach, which is common to decision trees. In most algorithms for decision trees, the training set is iteratively split into smaller subsets [26]. Each split creates two new branches down the tree. At each stage of the tree-building process, the best split is made at that stage, rather than looking ahead and choosing a split that will lead to a better tree at some future step [31].

Leo Breiman created the CART (Classification and Regression Trees) technique using binary tree architectures [9]. Every node in a binary tree structure contains two branches. The Gini index or towing criteria are used for partitioning. Post-pruning is used to modify model complexity if needed. When it is determined that there is no more information to be gained or when the stopping criteria are satisfied, branching in the CART method ceases [53].

2.3. **Bagging – Random Forest.** Bagging and random forests are called ensemble learning algorithms. The goal of ensemble learning is to combine individual classifiers to obtain new classifiers with better performance (higher classification accuracy, lower classification error) [54]. Bagging (bootstrap aggregating-bagging) was proposed by Leo Brieman in 1996 [8]. For every data set acquired through bootstrap sampling, a decision tree is constructed. There is no dependence between these decision trees; they operate in parallel. In an effort to increase accuracy, bagging aggregates the results of all decision trees into a single prediction to produce an ensemble classifier. To classify a new sample, each classifier generates its own class prediction, and the bagged classifier returns the most predicted class as the result (voting method) [54]. Bagging can be applied to a wide range of algorithms, such as linear regression, logistic regression, and discriminant analysis. What makes the difference here is to create different datasets by resampling the observations in the training set and applying the chosen algorithm to these datasets [44].

The application of the bagging method to the CART algorithm is also called the random forest algorithm [10]. However, the difference between random forest and bagging is related to feature selection. If there are m features in the dataset, p of these variables are randomly selected to form trees [31]. Random variable selection is called the random subspace method proposed by Ho [28]. Thus, the main step of the random forest algorithm can be considered as the proposed random subspace method. In the random forest algorithm, the generated decision trees are not pruned, the best variable among p randomly selected variables is determined, and the root node in the tree is formed. The classification of a new instance is determined by the voting method [54].

Another ensemble learning method, boosting, is a method that tries to improve classification accuracy by combining multiple models by generating different classifiers, as in bagging. In both learning methods, voting is used to classify a new example. In boosting, unlike bagging, each classifier depends on the errors made in the previous classifier, and a new classifier is created by taking these errors into account. In bagging, each classifier is independent of other classifiers, and the errors of other classifiers are not taken into account [54].

2.4. **Gradient Boosting Decision Tree (GBDT) and Extreme Gradient Boosting (XGBoost) Algorithms.** Gradient boosting is a gradient-based approximation method for iteratively training a boosting classifier. The approximation to the function f is computed using linear combinations of the functions $f : R^n \to R$ as the base function and $h : R^n \to R$ as the weak learners [56]. The mathematical expression of this situation is shown in Eq. (1)

$$f(x) = \sum_{j=1}^{M} \alpha_j h_j(x; \theta_j) \tag{1}$$

Where $x \in R^n$ is the input vector (feature, variable), $\alpha_j \in R$, and M is the number of weak models.

$\{(x_i, y_i)\,|x_i \in R^n \ ve \ y_i \in R\ \}_{i=1:N}$ being the training dataset, $f\,(.)$ is constructed by iteratively selecting the parameters α_j and θ_j shown in Eq. (1) to minimize an augmented loss function. The loss function L shown in Eq. (2) is a function used to measure the difference between the true value and the predicted value.

$$L = \sum_{i=1}^{N} l(y_i, f(x_i)) \tag{2}$$

The Gradient Boosting Decision Tree (GBDT) is a machine learning algorithm widely used for its classification accuracy and efficiency. GBDT is an ensemble model of sequentially trained decision trees [22]. GBDT often uses regression trees as weak models [56]. At each iteration, it builds decision trees using negative gradients (also called errors) [33].

The steps of the GBDT algorithm are given below [22,27]. $\{(x_i, y_i)\}_{i=1}^{n}$ dataset and $l\left(y_i, f\left(x_i\right)\right)$ is the differentiable loss function;

Table 3. Gradient boosting algorithm

1. $f_0\left(x\right) = argmin_{\gamma} \sum_{i=1}^{N} l\left(y_i, \gamma\right)$
2. For $m = 1 \; to \; M$: For $i = 1,\ 2,\ \ldots,\ n$

$$r_{im} = -\left[\frac{\partial l\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)}\right]_{f=f_{m-1}}$$

3. A regression tree is constructed where the terminal nodes (leaves) are denoted by $R_{jm},\ j = 1, 2, \ldots,\ J_m$ and the dependent variable is r_{im}. For $j = 1, 2, \ldots,\ J_m$

$$\gamma_{jm} = argmin_{\gamma} \sum_{x_i \in R_{jm}} l\left(y_i, f_{m-1}\left(x_i\right) + \gamma\right)$$

4. Update $f_m\left(x\right) = f_{m-1}\left(x\right) + \sum_{j=1}^{J_m} \gamma_{jm} I\left(x \in R_{jm}\right)$
5. Output: $\hat{f}\left(x\right) = f_M\left(x\right)$

The first line of the algorithm in Table 3 starts with an optimal fixed decision tree model with only one terminal node. The negative gradient components computed in row 2 are called generalized or pseudo residuals r_{im}, where i is the observation order and m is the number of trees. In 3, a regression tree is constructed, where R_{jm} corresponds to the leaf nodes in the trees, m is the number of trees, and j is the j_{th} leaf node in the m_{th} tree. In 2 and 4, the number of trees is iterated M times, resulting in the aggregated model $f_M\left(x\right)$. The algorithm has two main tuning parameters: the number of iterations (also called the number of trees) M and the number of terminal nodes (number of leaves) J_m in each of the constituent trees.

In Step 4, the gradient boosting model is regularized by adding a regularization parameter such that $f_m\left(x\right) = f_{m-1}\left(x\right) + v \sum_{j=1}^{J_m} \gamma_{jm} I\left(x \in R_{jm}\right)$ $0 < v \leq 1$. v is also called the learning rate, and adding the learning rate avoids the overfitting problem Friedman (2001) [22]. In short, overfitting can be considered as the classification accuracy that is high in the training set but not high enough in the test set [38]. In addition, the learning rate improves the prediction performance by reducing the individual effect of each tree [23].

In classification problems, the loss function is usually used $l = e^{-y_i f(x_i)}$ and $l = log\left(1 + e^{-2y_i f(x_i)}\right)$ [4]. Eq. (3) is also used for n observation values, which is obtained by using logistic regression. This study makes use of Eq. (3).

$$l = \sum_{i=1}^{n} y_i ln\left(p\right) + \left(1 - y_i\right) ln\left(1 - p\right) \tag{3}$$

XGBoost can be considered a generalized version of the gradient boosting algorithm with additional parameter settings. It has a high prediction performance and has a multicore and parallel (distributed) machine implementation [45]. XGBoost algorithm innovatively introduces a new tree learning algorithm for sparse data, determines sample weights in tree learning with weighted quantile sketch, and makes learning faster with parallel information processing [11].

To prevent overfitting, XGBoost employs a regularization term in addition to the differentiable loss function. $\{(x_i, y_i)\}_{i=1}^{n}$ dataset and $l\left(y_i, \widehat{y_i}\right)$ is the differentiable loss function;

$$L\left(\varphi\right) = \sum_{i} l\left(y_i, \widehat{y_i}\right) + \sum_{k} \Omega\left(f_k\right) \tag{4}$$

$$\Omega\left(f_k\right) = \gamma T + \frac{1}{2}\lambda w^2$$

Eq. (4) is obtained. $\Omega\left(f_k\right)$ is the complexity of the f_k model (decision tree), and this term prevents overfitting. A zero value for this term corresponds to the gradient boosting algorithm [11]. In the function $\Omega\left(f_k\right)$, T is the number of leaves in the tree f_k, w is the leaf weight and is calculated using the predicted values in the leaves. γT generates a fixed penalty score for each additional tree leaf, and λw^2 penalizes overweights. γ and λ are practitioner-specified parameters [45]. The XGBoost algorithm also uses a shrinkage parameter, similar to the learning rate used in gradient boosting, and column (feature) subsampling to prevent overfitting. The column subsampling is similar to the random subspace method in the random forest (RF) algorithm. The tuning parameter reduces the individual influence of each tree to make the model predictions consistent, as in the GBDT algorithm [11,23].

2.5. **Decision Table and Boolean Decision Table (BDT) Approach.** Decision tables are tabular representations of knowledge in which a set of conditions is used in combination and outcomes are determined according to these conditions. When used as classification models, the conditions are the values of attributes (arguments), and each attribute-value set (table cell) is associated with a class prediction [21,40].

The model structure of the decision table is a simple relational table. Each row in the table represents all combinations for each value of the features (variables) used. There is a column for each attribute and a column containing the probability values of the dependent variable corresponding to the respective attribute combinations. The probabilities in this last column give the proportion of observations in the class label [3]. Each row in the decision table represents a classification rule [43]. An example representation of a decision table with 3 binary-valued features x_1, x_2, x_3, and one binary-valued class label y is as follows.

Table 4. Sample decision table

x_1	x_2	x_3	y Number of Observations n_i	y Probability Values	
				Class 1	Class 2
1	0	1	10	0.40	0.60
0	1	0	20	0.30	0.70
0	0	1	5	0.80	0.20

In Table 4, there are 10 observations with a combination of variables in the first row. Of these observations, 40% is in the first class and 60% in the second class. In the second row, there are 20 observations with combinations of variables, and 30% of these observations are in the first class and 70% in the second class. In the third row there are five observations with the combination of variables, and 80% of these observations are in the first class and 20% in the second class. If a new observation is to be classified in the decision table, the matching rows in the table are examined, and assignment is made according to the majority class, as is usually done in decision trees. If there is no matching row, the observation is assigned to the predetermined default class (the majority class in the dataset) [43].

Lu and Liu (2000) developed a Grouping and Counting (GAC) decision table algorithm [43]. Since this algorithm is similar to the proposed Boolean decision table approach, it will first be introduced, and then the proposed algorithm will be described.

In the GAC algorithm; $(a_1,\ a_2,\ a_3,\ \cdots,\ c_k)$, where a_i denotes the values of attributes $A_i \in \{A_1,\ A_2,\ \cdots,\ A_n\}$ and $c_k \in \{c_1,\ c_2,\ \cdots,\ c_m\}$ denotes the class values. In the GAC algorithm, the decision table consists of $n+3$ columns; $(A_1,\ A_2,\ \cdots,\ A_n,\ Class,\ Sup,\ Conf)$.

Each row in the table shows the classification rule $(a_{1i},\ a_{2i},\cdots,\ a_{2i},\cdots,\ a_{ni},\ c_i,\ sup_i,\ conf_i)$; if $A_1 = a_{1i}$ and $A_2 = a_{2i}$ and $\cdots$ and $A_n = a_{ni}$ then $Class = c_i\ (sup_i,\ conf_i)$ where sup_i, $conf_i$ are the support and confidence values of the classification rule. The support value of the rule (sup) is the number of observations covered in the data set divided by the total number of observations, while the confidence value ($conf$) is the conditional probability value $P(class = c_i \mid (A_1 = a_{1i}) \cap (A_2 = a_{2i}) \cap \cdots \cap (A_n = a_{ni}))$ in other words, the class distribution of the observations covered by the rule. Each row in the table is called a decision table because it represents the rule that determines the class of an instance with given variable values. After appropriate grouping and counting, statistical information about the attribute values and class distribution in a candidate decision table is obtained, and the rows in the candidate table are pruned according to certain criteria (rows with confidence value below conf 0.5, etc.) to obtain the final decision table.

In the BDT approach; $(x_1,\ x_2,\ x_3,\ \cdots,\ x_k)$, where x_i denotes the values of features $X_i \in \{X_1,\ X_2,\ \cdots,\ X_n\}$ and $y \in \{0,1\}$ denotes the class values. The minterm variable (M) is created, and the variable order is important when creating (M); $M = x_k.2^0 + x_{k-1}.2^1 + \cdots + x_2.2^{k-2} + x_1.2^{k-1}$.

In the BDT approach, the decision table consists of 6 columns; (M_i, *Frequency-0*, *Frequency-1*, *Total Frequency*, *Normalized Probability-0*, *Normalized Probability-1*). *Total Frequency* corresponds to the sum of *Frequency-0* and *Frequency-1*; *Total Frequency* = *Frequency-0*+ *Frequency-1*. Column M corresponds to unique combinations of variables in the dataset. *Frequency-0* and *Frequency-1* represent the class variable frequencies corresponding to the minterms, while *Normalized Probability-0* and *Normalized Probability-1* represent the conditional probability values corresponding to the frequencies. Classification is performed according to the normalized probability values corresponding to the values in the minterm column. Afterwards, the minterms are ranked according to their importance with the TOPSIS method, taking into account the *Frequency-0*, *Frequency-1* values. The implementation steps of the BDT approach are explained in detail under the heading Application on Sample Data Set.

2.6. **Application on Sample Dataset.** In the proposed Boolean decision table approach, minterms in Boolean algebra are used since the features are binary. Minterms represent combinations of features, and the classification process is performed using the frequency numbers of these feature combinations and conditional probability values as in the GAC algorithm. Unlike the GAC algorithm, all features (variables) in the dataset are used after the selection of important features (feature selection). Instead of the support value in the GAC algorithm, pruning is done by considering the class frequency counts. In addition, the best classification rules in the dataset are determined by TOPSIS method, which is one of the multi-criteria decision-making methods, taking into account the frequency counts. Other multi-criteria decision-making methods can also be used, but this method was chosen because the application-specific ranking of alternatives is required and TOPSIS method is more suitable for ranking [2]. In addition, TOPSIS is more appropriate when the number of alternatives and criteria is high and the data is quantitative [48]. In a different study, other multi-criteria methods can be used to compare the results. Although a method similar to the algorithm we will use is proposed in Ogihara et al. [47], it differs in terms of the technique used in observation classification and the use of multi-criteria methods. In Ogihara et al.'s study, the classification process and classification accuracy were performed by considering only the frequency counts, while in this study, the classification accuracy was determined according to the ROC and PR curves and the best classification rules were tried to be determined with the TOPSIS method, which is one of the multi-criteria decision-making methods. This is how, the best feature combinations were identified in the dataset.

Let's try to explain the proposed approach on a sample dataset; Table 5 shows the dataset for binary classification consisting of 1173 observations where the independent feature (attribute, variable) x_1, x_2, x_3 is a Boolean attribute taking 3 binary values. The aim is also to determine the best combinations of features along with the classification analysis. The class label Y is a nominal variable with two classes (boolean). 1083 observations belong to the category represented by 0 (zero), and 90 observations belong to the category represented by 1. 92% of the observations therefore belong to the 0 category and 8% to the 1 category.

The minterm column in Table 5 was obtained using features (independent variables) $x_1,\ x_2,\ x_3$. Since these variables take binary values, they were converted to the decimal number system as used in the Boolean algebra method, and the minterm column was obtained. The rightmost variable x_3 corresponds to $\mathbf{2^0}$, x_2 to $\mathbf{2^1}$ and x_1 to $\mathbf{2^2}$. Unique values are obtained by multiplying the observation values of the variables by the corresponding values in the binary base. For observation 1, this column value is calculated as $\mathbf{1.2^2 + 1.2^1 + 0.2^0 = 6}$. Decimal values of other observations are found in a similar way. Since there are three independent variables that take binary values, there is a unique value (called minterm in this application) such that $\mathbf{2^3 = 8}$. Depending on the dataset, some minterm values may have no observations. In the example, minterm 0, 1, 2, 2, 4, 5, 6, resulting in a total of 6 unique values

Table 5. Sample dataset

n	$x_1(2^2)$	$x_2(2^1)$	$x_3(2^0)$	*Y (Class Label)*	*Minterm*
1	1	1	0	0	6
2	1	0	1	1	5
3	0	0	1	0	1
4	0	1	0	0	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1173	0	0	0	0	0

(minterm). 1173 observations can be represented by 6 minterm. There are no observations corresponding to the 3_{rd} and 7_{th} minterms. The 3_{rd} and 7_{th} minterms are not included in Table 6 below.

Table 6. Decision table for the sample dataset

				Y Class Label)						
$x_1(2^2)$	$x_2(2^1)$	$x_3(2^0)$	*Minterm*	Frequency-0	Frequency-1	Total Minterm Frequency	Total Probability-0	Total Probability-1	Normalized Probability-0	Normalized Probability-1
1	1	0	6	22	3	25	22/1173	3/1173	22/25	3/25
1	0	1	5	174	21	195	174/1173	21/1173	174/195	21/195
1	0	0	4	46	4	50	46/1173	4/1173	46/50	4/50
0	1	0	2	80	10	90	80/1173	10/1173	80/90	10/90
0	0	1	1	604	46	650	604/1173	46/1173	604/650	46/650
0	0	0	0	157	6	163	157/1173	6/1173	157/163	6/163
Total Number of Observations				1083	90	1173				

Table 6 shows the minterms corresponding to the variable combinations and the corresponding class variable frequency counts and conditional probabilities. The minterms in Table 6 represent a rule that can be used to determine the class of a sample with variable values. Frequency-0 is the number of frequencies belonging to the class coded with 0 in the class variable and Frequency-1 is the number of frequencies belonging to the class coded with 1. Here, two probability values are calculated: "Total Probability" and "Normalized Probability". "Total Probability" is the probability value calculated with the total number of observations 1173. For the 6_{th} minterm, there are 3 observations belonging to class 1 and 22 observations belonging to class 0 (zero), and the probabilities are $\frac{22}{1173} = 0.02$ and $\frac{3}{1173} = 0.003$. In the "Normalized Probability" values, there are 25 observations in total for the 6_{th} minterm and the normalized probability values are calculated as $\frac{22}{25} = 0.88$ and $\frac{3}{25} = 0.12$ considering the number 25. The sum of "Normalized Probability" values is $0.12 + 0.88 = 1$. The classification process will be carried out by considering these "Normalized Probability" values.

Table 7. Summary decision table for the sample dataset

Minterm	Frequency-0	Frequency-1	Total Minterm Frequency	Normalized Probability-0	Normalized Probability-1
6	22	3	25	22/25	3/25
5	174	21	195	174/195	21/195
4	46	4	50	46/50	4/50
2	80	10	90	80/90	10/90
1	604	46	650	604/650	46/650
0	157	6	163	157/163	6/163

The minterms in Table 7 can be thought of as leaves in a CART algorithm. Although the initial probability of belonging to class 1 was 8%, conditional probabilities greater than 8% were obtained here. With this method, the best combinations of variables are obtained. When we look only at the probability values, it is the 6_{th} minterm that has the maximum probability of belonging to the 1-class. The 110 encoding, which is the binary equivalent of the 6_{th} minterm, can be interpreted as the best combination of independent variables in this dataset. The combination where x_1 is 1, x_2 is 1, and x_3 is 0 has the highest classification success. TOPSIS method is used to decide on the best or best of the minterm values. The reason for choosing the TOPSIS method is that the data is quantitative, and the ranking of alternatives is important. It is more advantageous to use the TOPSIS method in ranking alternatives [2].

Table 8. Ideal points for TOPSIS

	Y (Class Label)	
	Frequency-0	Frequency-1
Ideal point	22	46
Non-ideal point	604	3

In Table 8, as optimal values; Frequency-0 is the negative criterion and Frequency-1 is the positive criterion. Positive and negative criteria vary according to the application area. The minimum value for the negative criterion and the maximum value for the positive criterion are taken into consideration. The TOPSIS method is used to determine the best minterm and minterms. Thus, the minterms that give the most successful classification in the data set and the feature combinations accordingly are determined. Since there are 7 alternatives here, the best ranking is not done. The ranking phase will be explained in more detail in the application section.

2.7. **Model Evaluation - Classification Tables, ROC and PR Curves.** The performance of a model is related to its predictive success on independent test data. Evaluating this performance provides a measure of the quality of the model [27]. Model evaluation provides metrics to assess how good the observations are at predicting the class label [26]. The error matrix is a table of m rows and columns for a given problem with m $(m > 2)$ classes. The i_{th} row and j_{th} column of the error matrix correspond to the number of observations that actually belong to class j and are predicted by the classification model in class i. For a classification model to have good accuracy, it is desirable that most of the observations lie on the prime diagonal and are close to zero off the prime diagonal. The error matrix for a two-class classification problem is shown in Table 9. Observations are labeled as positive and negative. P represents the number of positive observations in the dataset, N represents the number of negative observations, and P' and N' represent the number of positive and negative observations predicted by the model [26].

Table 9. An example classification table

		Actual Class		
		0	1	Total
Predicted Class	0	*TN*	*FN*	N'
	1	*FP*	*TP*	P'
	Total	N	P	N+P

True Positive (TP): Refers to positive observations correctly labeled by the classifier. TP corresponds to the number of correct positive observations.

True Negative (TN): Refers to negative observations correctly labeled by the classifier. TN corresponds to the number of correct negative observations.

False Positive (FP): The label value refers to observations that are actually negative but labeled positive by the classifier. FP corresponds to the number of false positive observations.

False Negative (FN): The label value refers to observations that are actually positive but labeled as negative by the classifier. FN corresponds to the number of false negative observations.

In problems with class imbalance, the classification model correctly classifies the observations of the majority class. However, in this case, minority class observations may also be misclassified. Therefore, it is more consistent to use sensitivity and specificity measures. Sensitivity, also called true positive rate, is a measure of how many observations that are actually positive are correctly classified by the model. Specificity, on the other hand, is referred to as the true negative rate and is a measure of how many observations that are actually negative are correctly classified by the model [26]. The precision measure can be thought of as a measure of accuracy. It gives the proportion of observations labeled as positive by the model that are correctly classified [18].

In classification problems, instead of assigning a class label, it is more appropriate to give a probability distribution. This method, called scoring, can also be thought of as the probability of being assigned to a positive class. Observations with higher scores are more likely to be assigned to the positive class [18]. In this case, the choice of the threshold value will be important. Depending on the value of the threshold, an

Table 10. Evaluation criteria

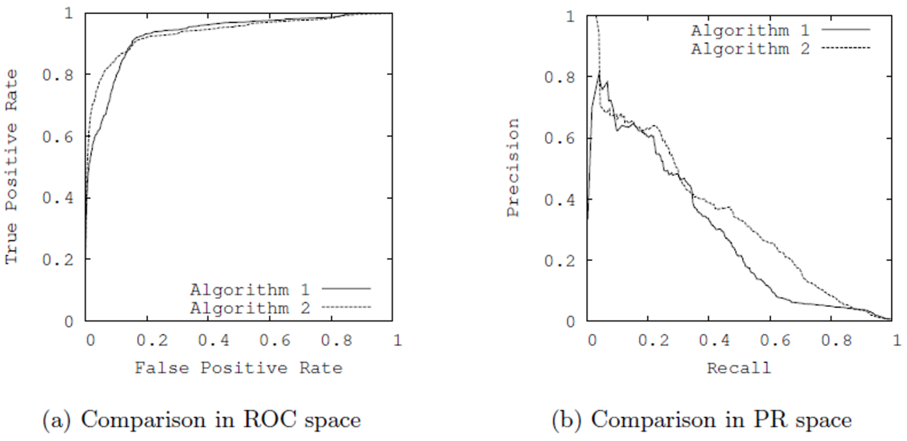
Evaluation Criteria	Formula
Accuracy	$\frac{TP+TN}{P+N}$
Misclassification rate (Error)	$\frac{FN+FP}{P+N}$
Sensitivity (True Positive Rate, Recall)	$\frac{TP}{P}$
Specificity (True Negative Rate)	$\frac{TN}{P}$
False Negative Rate	$\frac{FN}{P}$
False Positive Rate	$\frac{FP}{N}$
Precision	$\frac{TP}{P'}$
F-score	$2.\frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}$

observation can be classified as positive or negative. In binary classification (binary classification) (0-1) problems, the cut-off point is usually chosen as 0.5. The choice of threshold value may cause changes in classification errors. This is not a correct approach for imbalanced datasets [20]. Imbalanced datasets have high or low levels of observed prevalence. Prevalence is a measure of how frequent each category is in the dataset. It can also be considered as the number of observations of categories. It is divided into observed (actual) and predicted prevalence. In many applications, it is important that observed and predicted prevalence are similar. Therefore, both estimated and observed prevalence can be used as a criterion for the threshold value [14]. With a high threshold, fewer positive samples will be classified, so the false positive rate will decrease, but the false negative rate will increase. With a low threshold, more positive samples will be classified, so the false negative rate will decrease, but the false positive rate will increase. Therefore, the choice of threshold is important. All accuracy measures calculated in Table 10 are calculated based on the threshold value selection. Class labels are created with the threshold selection. Receiver operating characteristics (ROC) and Precision-Recall (PR) curves, which are methods that do not depend on threshold selection, are more objective measures used to evaluate model performance [18].

The ROC curve is a technique used to visualize, organize, and select classifiers based on their model performance [17]. The ROC curve shows the relationship between the true positive rate (sensitivity) and the false positive rate (1-specificity) [50]. In a two-class problem, the ROC curve allows us to visualize the balance between the rate at which the model correctly classifies positive observations (sensitivity) and the rate at which it incorrectly classifies negative observations as positive. The Area Under Curve (AUC) is a measure of the accuracy of the model [26]. The Area Under Curve (AUC-ROC) gives a measure of overall model performance. Good models have an AUC value close to 1, while poor models have an AUC value close to 0.5. These values are used to evaluate model performance [20].

PR curves have been proposed as an alternative to ROC curves in case of imbalance in class distribution in binary classification models [18]. The important difference between the ROC space and the PR space is the visualization of the curves. In PR curves, differences between algorithms that are not evident in ROC space can be revealed. In the PR space, there is precision on the vertical axis and sensitivity (true positive rate, recall-sensitivity) on the horizontal axis [15].

In the ROC space, the objective is to be in the upper left corner and looking at the ROC curves in Fig.1(a), they appear to be close to optimal. In the PR space, the goal is to be in the upper right corner and the PR curves in Fig.1(b) show that there is room for improvement. The performances of the algorithms seem to be close to each other in the ROC space. In the PR domain, Algorithm 2 is said to be superior to Algorithm 1. This difference is due to the fact that the number of negative samples in this domain is higher than the number of positive samples [15].



(a) Comparison in ROC space (b) Comparison in PR space

FIGURE 1. The difference between ROC and PR curves

Area Under Curve (AUC-PR) can be used in the PR curve as in the ROC curve, but AUC-PR varies according to the prevalence of the positive class and its expected value is close to the proportion of positive classes in the dataset. The lower bound for the AUC-PR value is the prevalence value of the positive class. The higher the AUC-PR value is, the better the classifier is said to be [18]. Like the AUC-ROC value, good models are expected to have an AUC-PR value close to 1.

3. APPLICATION

The data to be used in the study were obtained from the Google Analytics platform of an e-commerce firm. Google Analytics is a free Google service that can measure traffic for websites in a broad sense and keeps temporal records of this traffic.

TABLE 11. Dataset summary table

User	User Type	Device	Source	Medium	Browser	Region	Day of Week	Hour	Session Duration (seconds)	Transactions (Class Label)
1	Old	Desktop	Direct	Direct	Chrome	Istanbul	Monday	10	250	1
8	New	Mobil	Youtube	Ads	Safari	Izmir	Saturday	12	91	0
3	New	Mobil	Google	Organic	Safari	Antalya	Friday	18	75	0
4	New	Tablet	Direct	Video	Samsung	Ankara	Friday	16	61	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
116536	New	Mobil	Google	Ads	Chrome	Bursa	Wednesday	14	300	1

As seen in Table 11, the dataset contains 116536 observations, 9 features (variables, attributes), and one class label. A detailed description of the variables was given below.

Transactions: It is the class label. It is the variable that represents whether a user has made a purchase or not. It consists of 1 and 0 binary (boolean) values. 1 represents that the user has made a purchase once and 0 represents that the user has not made a purchase.

User Type: It consists of 1 and 0 binary values. 1 represents that the user has accessed the site for the first time (new user) and 0 represents that the user has accessed the site before (old user).

Device: It is the feature that shows which device users use to access the relevant site. There are three device types: mobile, desktop, and tablet. For each device, this feature was represented by 3 independent variables with one-hot encoding method [64]. Here, the device feature was represented by three features as mobile device, desktop device, and tablet device.

- ***Mobile Device*;** consists of binary values 1 and 0. 1 represents that the user accessed the relevant site using a mobile device, 0 represents that the user accessed the relevant site using another device (desktop, tablet).

- ***Desktop Device***; consists of the binary values 1 and 0. 1 represents that the user accessed the relevant site using a desktop device, 0 represents that the user accessed the relevant site using another device (mobile, tablet).
- ***Tablet Device***; consists of binary values 1 and 0. 1 represents that the user accessed the relevant site using a tablet device, 0 represents that the user accessed the relevant site using another device (mobile, desktop).

Table 12. Device feature one-hot encoding

Device	D_{Mobil}	$D_{Desktop}$	D_{Tablet}
Mobil Device	1	0	0
Desktop Device	0	1	0
Tablet Device	0	0	1

As seen in Table 12, the device feature is represented by three feature variables: D_{Mobil}, $D_{Desktop}$, D_{Tablet}.

Source; it is a feature that indicates the source through which users came to the relevant site. Direct, Youtube, and Google sources are available. Since this feature also has three categories, it was represented as three features with the one-hot encoding method.

Medium; it is the feature that shows through which tool users access the relevant site. Direct, advertising, organic, and video tools are available. Since this feature has four categories, it was represented as four features with the one-hot encoding method.

Browser; it is the feature that shows which browser users use to access the relevant site. There are four browser types: Chrome, Safari, Samsung, and Android. For each browser, it was represented as four features with one-hot encoding method.

Region; it is a feature indicating the province from which users accessed the relevant website. There are six provinces, namely Istanbul, Ankara, Izmir, Bursa, Adana, and Antalya. For each province, it was represented six features with one-hot encoding method.

Hour; it is a feature that shows the time interval in which the user came to the site. It was a feature categorized according to 24 hours. Unlike other feature, index calculation was used here and categorized as 24 hours binary.

Table 13. Hour feature discretization

Hour	Frequency-0	Frequency-1	Total Frequency	Frequency-1/Total Frequency	Binary Coding
0	6040	485	6525	485/6525=0.074	0
1	3806	238	4044	238/404 =0.059	0
2	2176	120	2296	120/2296=0.052	0
3	1342	56	1398	56/1398=0.040	0
4	856	34	890	34/890=0.038	0
⋮	⋮	⋮	⋮	⋮	⋮
23	6283	614	6897	614/6897=0.089	1

Table 13 shows the number of shoppers, non-shoppers, and total frequencies by time of day. A new column was obtained by proportioning the Frequency-1 value, which shows the number of users who shop by hours, to the total number of frequencies for the relevant time zone. For example, for time zone 0; 485/6525 = 0.074. The same method is applied for other time zones. The geometric mean of these ratios 0.078 is taken and this variable is made binary (boolean) by giving a value of 0 if the ratio of the relevant time zone is below the geometric mean value and 1 if it is above it. As can be seen in Table 13, the hours starting from 8 a.m. are given a value of 1 and the hours before 8 a.m. are given a value of 0 (zero).

Day of week; it is the feature that shows on which day of the week users access the relevant website. It was represented by 7 independent variables with one-hot encoding method as Monday, Tuesday, Wednesday, Thursday, Thursday, Friday, Saturday, and Sunday.

Session Duration; It is a continuous feature that calculates how many seconds a user spends on the site. Because it is continuous, it is different from the other variables in the dataset. This variable was made discrete with the discretization method. It was discretized according to the quartile values using the equal frequency discretization method. It was divided into 4 intervals according to values less than 1st quartile, between 1st quartile and 2nd quartile, between 2nd quartile and 3rd quartile and greater than 3rd quartile and represented by 4 independent variables with one-hot encoding method.

Table 14. Summary statistics for session duration feature (seconds)

Minimum	1.quartile	Median	3.quartile	Maximum
37	103	196	437	6881

Session duration.1 consists of the binary values 1 and 0. It is represented by a value of 1 if the session duration is below the 1st quartile value of 103 seconds and 0 otherwise.

Session duration.2 consists of the binary values 1 and 0. It is represented by the value 1 if the session duration is less than the 2nd quartile value of 196 seconds and greater than or equal to the 1st quartile value of 103 seconds, and 0 otherwise.

Session duration.3 consists of the binary values 1 and 0. If the session duration is less than 437 seconds, which is the 3rd quartile value, and greater than or equal to 196 seconds, which is the 2nd quartile value, it is represented by the value 1, otherwise it is represented by the value 0.

Session duration.4 consists of the binary values 1 and 0. It is represented by a value of 1 if the session duration is greater than or equal to the 3rd quartile value of 437 seconds and 0 otherwise.

The session duration variable was represented by 4 variables, separated by quartiles.

The dataset contains 116536 observations. The dataset is randomly divided into 75% training data and 25% test data using a validation set approach [31].

Table 15. Dataset class distribution

	Y-Class Variable	
Number of 0-class observations	Number of 1-class observations	Total Observations
105614 (91%)	10922 (9%)	116536 (100%)

As seen in Table 15, there are 105614 observations belonging to the category represented by 0 and 10922 observations belonging to the category represented by 1. The 0-class represents users who have not made a purchase and the 1-class represents users who have made a purchase. Proportionally, 9% of the observations in the dataset belong to the class represented by 1 and 91% to the class represented by 0.

Table 16. Training dataset class distribution

	Y-Class Variable	
Number of 0-class observations	Number of 1-class observations	Total Observations
79209 (91%)	8225 (9%)	87434 (100%)

75% of the dataset was used as training data. Table 16 shows the number and proportions of class distributions in the training set. There are 87434 user data in the training dataset.

Table 17. Test dataset class distribution

	Y-Class Variable	
Number of 0-class observations	Number of 1-class observations	Total Observations
26405 (91%)	2697 (9%)	29102 (100%)

25% of the dataset was used as test data. Table 17 shows the number and proportions of class distributions in the test set. There are a total of 29102 user data in the test dataset.

Table 18. Cross table for feature and class variable

		Class Variable		
		0	1	
i.Feature	0	$n_{00}^{(i)}$	$n_{01}^{(i)}$	
	1	$n_{10}^{(i)}$	$n_{11}^{(i)}$	
	Total	N_0	N_1	$N_0 + N_1 = N$

Table 19. Significance values of features

	Features	Odds ratio (OR)	s_j	χ^2 statistics	Z statistics
1	_User Type_	_0.377_	_0.063_	_1816.180*_	_38.363*_
2	_Mobile Device_	_0.624_	_0.336_	_363.400*_	_17.593*_
3	_Desktop Device_	_1.640_	_-0.377_	_387.570*_	_-17.948*_
4	Tablet Device	0.989	-0.960	0.010	0.140
_5_	_Direct Source_	_1.277_	_-0.760_	_46.190*_	_-6.307*_
_6_	_Youtube Source_	_0.272_	_-0.878_	_877.630*_	_44.202*_
_7_	_Google Source_	_1.844_	_0.638_	_435.790*_	_-23.910*_
_8_	_Direct Medium_	_1.277_	_-0.760_	_46.190*_	_-6.307*_
_9_	_Ads Medium_	_1.706_	_-0.090_	_529.650*_	_-22.027*_
_10_	_Organic Medium_	_0.927_	_-0.272_	_9.980*_	_3.197*_
_11_	_Video Medium_	_0.272_	_-0.878_	_877.630*_	_44.202*_
12	_Chrome Browser_	_0.746_	_-0.053_	_159.450*_	_12.621*_
13	_Safari Browser_	_1.457_	_-0.030_	_264.850*_	_-16.006*_
14	_Samsung Browser_	_0.708_	_-0.921_	_34.830*_	_6.730*_
15	_Android Browser_	_0.338_	_-0.996_	_20.500*_	_7.026*_
16	_Istanbul Province_	_1.082_	_0.063_	_11.500*_	_-3.408*_
17	Ankara Province	1.070	-0.575	5.550*	-2.333
18	_İzmir Province_	_0.865_	_-0.781_	_15.220*_	_4.099*_
19	Bursa Province	0.886	-0.899	5.230	2.417
20	_Adana Province_	_0.862_	_-0.897_	_8.000*_	_3.011*_
21	Antalya Province	0.957	-0.911	0.570	0.794
22	_Hour_	_1.220_	_0.585_	_237.270*_	_-7.339*_
23	Monday	1.022	-0.723	0.390	-0.637
24	Tuesday	0.949	-0.753	2.176	1.516
25	Wednesday	0.938	-0.733	3.482	1.919
26	_Thursday_	_0.868_	_-0.757_	_15.999*_	_4.192*_
27	_Friday_	_1.316_	_-0.620_	_84.967*_	_-8.584*_
28	_Saturday_	_0.856_	_-0.751_	_19.770*_	_4.676*_
29	Sunday	1.045	-0.664	1.962	-1.400

*: p-value<0.01

3.1. **Feature Selection.** Table 18 was organized for each feature (attribute, variable) in the dataset and the following formula values were calculated. Table 19 shows these values. Important features were determined according to these values.

The odds ratio (OR) is calculated in Eq. (5) below [35].

$$OR = \frac{n_{11}^{(i)}/n_{10}^{(i)}}{n_{01}^{(i)}/n_{00}^{(i)}} \tag{5}$$

The calculation of the OR value will take into account the increase and decrease in the probability of shopping (class variable taking the value 1) in the presence of the relevant variable value (taking the value 1). The OR value takes values between 0 and ∞. If $OR > 1$ increases the probability of shopping and $OR < 1$ decreases the probability of shopping. $OR = 1$ indicates a neutral state, and no comment can be made on whether the relevant variable increases or decreases the chance of shopping. In addition, OR values close to 1 and 1 are one of the indicators that the relevant feature is insignificant [29].

The chi-square test is a hypothesis test that determines the dependency relationship between the class variable and the feature. In addition, the significance of the calculated odds ratio will also be tested with the chi-square method. The significance of this test result will be one of the indicators that the relevant feature is important [30].

With the ratios $p_j^+ = \frac{n_{11}^{(i)}}{N_1}$ and $p_j^- = \frac{n_{01}^{(i)}}{N_1}$, the value $s_j = p_j^+ - p_j^-$ is calculated. The sign of s_j determines the role of the feature. $s_j < 0$ indicates that the feature is negatively weighted (weighted in class 0), while $s_j > 0$ indicates that the feature is positively weighted (weighted in class 1). $p_j^+ \approx p_j^-$ is one of the indicators that the relevant feature is insignificant when the two ratios are close to each other [62].

The feature is determined to be significant using the ratio test in independent samples with ratios $\hat{p}_1 = \frac{n_{10}^{(i)}}{N_0}$ and $\hat{p}_2 = \frac{n_{11}^{(i)}}{N_1}$. The calculated value $z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}}$ is calculated and the significance test of the difference between the two ratios is performed. A significant difference between two ratios is one of the indicators that the relevant feature is important [30].

The significance of the feature was determined by looking at the statistical values and p values in Table 19. p value less than 0.01 error margin indicates that the relevant feature is significant. Thus, odds ratios (OR) and s_j values are interpreted more reliably. Looking at Table 19, 7 features (Tablet Device, Ankara Province, Bursa Province, Antalya Province, Monday, Tuesday, Wednesday, Wednesday, Sunday) are said to be insignificant. The important features are italicized. Out of the remaining 22 important feature, 5 of them (User Type, Mobile Device, Google Source, Province of Istanbul, Time of Day) have positive s_j values. Significant features can also be divided according to whether the s_j value is positive or negative.

Table 20. Significance values for the discretized session duration feature by quartiles

Features	Odds ratio (OR)	s_j	χ^2 statistics	Z statistics
Session duration.1	*0.039*	*-0.971*	*3523.300**	*125.159**
Session duration.2	*0.154*	*-0.891*	*1875.700**	*73.546**
Session duration.3	*1.599*	*-0.323*	*464.250**	*-17.662**
Session duration.4	*5.350*	*0.185*	*5710.200**	*-67.533**

*: p-value<0.01

Table 20 shows that all quartile values of the discretized session duration feature are significant.

Various R packages were used to build the models. In the CART algorithm, the gini index was selected as the splitting criterion and the splitting process was performed up to 2 observations. R program rpart library was used for CART analysis [58]. In the terminal nodes (leaves), no splitting was performed after 2 observations. This can sometimes cause an overfitting problem. With the discretization of the session duration feature, no overfitting problem occurred.

For the random forest (RF) algorithm, the model was run with predefined parameters and the minimum OOB error (0.09405) was reached at the 109th tree (ntree=109). Afterwards, models were created with as many cycles as the number of feature for the model with 109 trees and it was seen that the model had the minimum OOB error when the number of random variables was 4. For random forest analysis, R program randomForest library was used [39].

For the XGBoost algorithm, the number of trees (iterations) in the model should been determined first. While determining the number of trees, the minimum log-loss value is taken into account. This value is determined separately for the training and test sets. According to the number of iterations (the number of iterations corresponds to the number of trees in the XGBoost algorithm) of the log-loss rate in the training and test sets, the log-loss rate for the test set took its minimum value at the 716 iteration. The learning rate was taken as 0.01. A small learning rate prevents overfitting. The model was first run for 1000 iterations and the minimum log-loss value for the test set was taken at iteration 716. For XGBoost analysis, R program xgboost library was used [12].

In the Boolean decision table (BDT), there is no optimization phase as described in the theoretical part. Similar to the CART algorithm, the decision table was created up to 2 observations. In the CART algorithm, there was no division after 2 observations. Similarly, the rows in the decision table were created to have at least 2 observations. The table was created so that the total number of observations in two classes, shopping (1-class) and non-shopping (0-class), is 2 or more.

Table 21. Training set results

Method	Threshold	Sensitivity	Specificity	Precision	F-Score	Error	Accuracy	AUC-ROC	AUC-PR
CART	0.09	0.85	0.72	0.24	0.37	0.27	0.73	0.85	0.36
RF	0.09	0.06	0.99	0.56	0.12	0.09	0.91	0.62	0.22
XGBoost	0.09	0.88	0.68	0.22	0.35	0.30	0.70	0.85	0.31
BDT	0.09	0.71	0.89	0.24	0.38	0.28	0.72	0.87	0.36

When we look at the results of the training set in Table 21, the measure values, except for the AUC-ROC and AUC-PR values, vary depending on the cut-off point. The cut-off point was taken as 0.09, which is the minority class prevalence value in the training set. According to this cut-off point, RF was the algorithm with the highest correct classification rate (accuracy). The correct classification rates of CART, XGBoost, and BDT algorithms were found to be close to each other. The algorithm with the highest sensitivity rate was XGBoost and the algorithm with the lowest sensitivity rate was RF. RF algorithm failed to classify the minority class. For more objective interpretations independent of the cut-off point, AUC-ROC and AUC-PR values should be considered. When we look at these values, the highest AUC-PR value belongs to BDT and AUC-PR value belongs to BDT and CART. The RF algorithm has the lowest AUC-PR and AUC-PR values. The most successful algorithms in the training set were BDT, CART, and XGBoost.

Table 22. Test set results

Method	Threshold	Sensitivity	Specificity	Precision	F-Score	Error	Accuracy	AUC-ROC	AUC-PR
CART	0.09	0.83	0.72	0.23	0.36	0.27	0.73	0.83	0.27
RF	0.09	0.04	0.99	0.35	0.08	0.09	0.91	0.59	0.16
XGBoost	0.09	0.88	0.67	0.22	0.35	0.31	0.69	0.84	0.29
BDT	0.09	0.80	0.70	0.21	0.34	0.30	0.70	0.81	0.26

Looking at the test set results in Table 22, the cut-off point is again taken as 0.09, which is the minority class prevalence value in the training set. Similar results were obtained with the training set. According to this cut-off point, RF was the algorithm with the highest correct classification rate. The correct classification rates of CART, XGBoost, and BDT algorithms were found to be close to each other. The algorithm with the highest sensitivity rate was XGBoost and the algorithm with the lowest sensitivity rate was RF. RF algorithm failed to classify the minority class. When AUC-PR and AUC-PR values are analyzed, the highest AUC-ROC and AUC-PR values belong to the XGBoost algorithm. CART and BDT algorithm results were found to be close to each other. The RF algorithm has the lowest AUC-PR and AUC-PR values. The most successful algorithms in the test set were XGBoost, CART, and BDT.

All in all, the Boolean decision table technique had the highest training success rate. The Boolean decision table yielded values close to those of the CART and XGBoost algorithms and higher than those of the RF algorithm when we looked at the results of the test dataset. The Boolean decision table does not require as much time for training as the CART, RF, and XGBoost algorithms, even when tuning the hyperparameters. Considering the time needed for training and tuning the hyperparameters, it is more advantageous to use the Boolean decision table. The TOPSIS method is used to determine the best combination of variables for the Boolean decision table with sufficient classification success. By determining the best combinations of feature in the data set, the target audience analysis in the field of digital advertising can be evaluated more objectively.

3.2. **Topsis Result.** By using the Boolean decision table method, the best of the combinations (minterm) can be determined. The TOPSIS method determines the best minterms. In this case, the number of variables involved is 25. Considering Table 16, there are 87434 observations in the training set, 79209 belonging to 0-class and 8225 belonging to 1-class. While creating the decision table, the total number of

Table 23. Decision table for training dataset (5 rows)

Minterm	Frequency-0	Frequency-1	Normalized Probability-0	Normalized Probability-1
25838120	418	48	0.90	0.10
25838104	347	78	0.82	0.18
25838152	527	7	0.99	0.01
25838216	506	1	0.99	0.01
25837864	326	87	0.79	0.21

observations belonging to 0-class and 1-class were taken as greater than or equal to 2. Thus, 3747 unique combinations were obtained for the training set.

Table 23 shows 5 of the 3747 rows. Each of the minterms in the table consists of a combination of 25 variables. What is important for the decision maker here is that the Frequency-1 and Normalized probability-1 values are high. The values of Frequency-0 and Normalized probability-0 should be low. TOPSIS analysis is performed by taking into account the Frequency-1 and Frequency-0 criteria.

Table 24. Ideal points for the training set

	Y (Class Variable)	
	Frequency - 0	Frequency - 1
Ideal point	0	114
Non-ideal point	527	0

As optimal values, Frequency-0 is the negative criterion and Frequency-1 is the positive criterion. Positive and negative criteria vary according to the application area. The minimum value for negative criteria and the maximum value for positive criteria are taken into consideration. The minterms that give the most successful classification in the dataset and the variable combinations accordingly are determined. In TOPSIS analysis, weights can be determined subjectively and using the entropy method. However, in both cases, the ranking remains unchanged.

Table 25. Criteria weights

	Frequency-0	Frequency-1
Subjective	0.1	0.9
Entropy Method	0.36	0.64

The top 5 minterms for the training set as a result of TOPSIS analysis are shown in Table 26 below.

Table 26. TOPSIS results for the training set (top 5 ranks)

Rank	Minterm	Frequency-0	Frequency-1	Normalized Probability-0	Normalized Probability-1	TOPSIS Ratio
1	21774872	299	114	0.72	0.28	0.80
2	25837848	272	102	0.73	0.27	0.77
3	9060632	201	93	0.68	0.32	0.74
4	25837864	326	87	0.79	0.21	0.68
5	9060888	207	82	0.72	0.28	0.67

Table 26 demonstrates how much greater the initial 0.09 chance of belonging to class 1 is for the minterms in this case. Feature (variable) combinations may be recognized since the minterms in this case relate to combinations of variables. Regarding the minterm column, the variable order is crucial. The variable combination in Table 27 is generated based on the minterms' variable order.

Feature combinations are calculated according to the 0-1 values of the feature. Feature combinations are called target audiences in digital advertising.

When examining the feature combination for the 21774872 minterm, it is interpreted that the probability of shopping is 0.28 for users whose user type is new, who access the website after 20:00, whose session duration is above the 3rd quartile value of 437 seconds and who access the website using the

Table 27. Feature combinations and minterms

	Minterms				
Features	21774872	25837848	9060632	25837864	9060888
Saturday (2^0)	0	0	0	0	0
Friday (2^1)	0	0	0	0	0
Thursday (2^2)	0	0	0	0	0
Hour (2^3)	*1*	*1*	*1*	*1*	*1*
Session Duration.4 (2^4)	*1*	*1*	*1*	0	*1*
Session Duration.3 (2^5)	0	0	0	**1**	0
Session Duration.2 (2^6)	0	0	0	0	0
Session Duration.1 (2^7)	0	0	0	0	0
Ads Medium (2^8)	0	*1*	*1*	*1*	0
Organic Medium (2^9)	*1*	0	0	0	*1*
Direct Medium ($2^1$0)	0	0	0	0	0
Video Medium (2^{11})	0	0	0	0	0
Adana Province (2^{12})	0	0	0	0	0
Izmir Province (2^{13})	0	0	0	0	0
Istanbul Province (2^{14})	*1*	*1*	*1*	*1*	*1*
Android Browser (2^{15})	0	0	0	0	0
Samsung Browser (2^{16})	0	0	0	0	0
Safari Browser (2^{17})	0	*1*	*1*	*1*	*1*
Chrome Browser (2^{18})	*1*	0	0	0	0
Google Source (2^{19})	*1*	*1*	*1*	*1*	*1*
Youtube Source (2^{20})	0	0	0	0	0
Direct Source (2^{21})	0	0	0	0	0
Desktop Device (2^{22})	*1*	0	0	0	0
Mobil Device (2^{23})	0	*1*	*1*	*1*	*1*
User Type (2^{24})	*1*	*1*	0	*1*	0

Google source-organic tool from a desktop device with a Chrome browser from the province of Istanbul, as can be seen in Table 27. For this target group, Thursday, Friday, and Saturday can be excluded as the day variable is 0. This targeting can also be performed for days other than these three days.

When examining the feature combination for 25837848, it is interpreted that the shopping probability is 0.27, as can be seen in Table 27, for users whose user type is new, who access the website after 8 am, whose session duration is greater than 437 seconds, which corresponds to the 3rd quartile value, and who access the website from a mobile device with a Safari browser from the province of Istanbul.

When examining the feature combination for minterm 9060632 is examined, it is interpreted that the probability of shopping is 0.32, as can be seen in Table 27 for users whose user type is old, who access the website after 8 pm, whose session duration is greater than 437 seconds, which corresponds to the 3rd quartile value, and who access the website from a mobile device with the Safari browser from the province of Istanbul.

When examining the feature combination for minterm 25837864 is examined, it is interpreted that the probability of shopping is 0.21 as seen in Table 27 for users whose user type is new, who access the site after 8 pm, whose session duration is less than 437 seconds, which is the 3rd quartile value, and equal to or greater than 196 seconds, which is the 2nd quartile value, and who access the site from a mobile device with the safari browser from the province of Istanbul with the Google source advertising tool.

When examining the feature combination for minterm 9060888 is examined, it is interpreted that the probability of shopping is 0.28 for users whose user type is old, who access the site after 8 pm, whose session duration is greater than 437 seconds, which is the 3rd quartile value, and who access the site from a mobile device using the safari browser from Istanbul province with the Google source organic tool, as seen in Table 27.

Since the day variable Thursday, Friday, and Saturday take the value 0 in the target audience determined for these 5 minterms, these days can be excluded. This targeting can be done for other days other than these three days. In addition, a target audience is created for accessing the website, which is common for all 5 minterms, with Google resources from Istanbul province after 8 pm. For the combination of these three variables, the probability of shopping in the training set is 0.12 and the probability of shopping in the test set is 0.11. It is said that increasing the number of variables has an increasing effect on the probability.

With the TOPSIS method, the results of the best 5 minterms determined in the training set are also compared with the values in the test set. The dataset was randomly divided into 75% training data and 25% test data. Considering Table 17, there are 29102 observations in the test set, 26405 0-class and 2697 1-class observations. The results of the best 5 minterms for the training set can be compared with the values in the test set.

Table 28. Correspondence of training set minterms in the test set

Rank	Minterm	Frequency-0	Frequency-1	Normalized Probability-0	Normalized Probability-1
1	21774872	111	36	0.76	0.24
2	25837848	71	21	0.77	0.23
3	9060632	78	42	0.65	0.35
4	25837864	134	18	0.88	0.12
5	9060888	56	23	0.70	0.30

Table 28 demonstrates that for these minterms, the initial 0.09 probability of belonging to class 1 is likewise high. The top five minterms from the training set are evidently also applicable to the test set. The best minterms shared by both datasets and the optimal variable combinations for the dataset are discovered when the outcomes of the TOPSIS analysis carried out independently for the training and test sets are compared.

A separate TOPSIS analysis is also performed on the test set. The results are compared with the results of the training set and the minterms common to both sets can be interpreted as the best.

Table 29. TOPSIS results for the test dataset (top 5 ranks)

Rank	Minterm	Frequency-0	Frequency-1	Normalized Probability-0	Normalized Probability-1	TOPSIS Ratio
1	9060632	78	42	0.65	0.35	0.81
2	21774872	111	36	0.76	0.24	0.73
3	9060648	50	27	0.65	0.35	0.61
4	25838104	131	25	0.84	0.16	0.54
5	9060888	56	23	0.71	0.29	0.52

Table 29 displays the top 5 minterms from the test set TOPSIS analysis. Table 30 shows that the probability of belonging to class 1, which was originally 0.09, is significantly higher for the minimum conditions.

Table 30. Common minterms in training and test datasets

Rank	Training Set Minterm	Test Set Minterm
1	21774872	9060632
2	25837848	21774872
3	9060632	9060648
4	25837864	25838104
5	9060888	9060888

The minterms shown in Table 30 are among the top 5 minterms for both datasets (training and test). This shows that the best minterms for the dataset are 21774872, 9060632, and 9060888.

3.3. **Control Dataset.** Following the analysis, the outcomes were also contrasted with data that looked ahead. Table 31 below shows data for future dates that are not included in the dataset.

As seen in Table 31, there are 108024 observations belonging to the category represented by 0 and 3471 observations belonging to the category represented by 1. This shows that the class distribution in the control set is different from the class distribution in the training and test sets. When future data has a different distribution than past data, this is called dataset shift [51]. In the event of a dataset shift,

Table 31. Control dataset (future dated)

	Y-Class Variable	
Number of 0-class observations	Number of 1-class observations	Total Observations
108024 (97%)	3471 (3%)	111495 (100%)

Table 32. Control set results

Method	Threshold	Sensitivity	Specificity	Precision	F-Score	Error	Accuracy	AUC-ROC	AUC-PR
CART	0.09	0.79	0.77	0.10	0.17	0.23	0.77	0.83	0.12
RF	0.09	0.07	0.98	0.10	0.08	0.05	0.95	0.59	0.06
XGBoost	0.09	0.90	0.70	0.09	0.16	0.29	0.71	0.86	0.14
BDT	0.09	0.65	0.82	0.11	0.18	0.18	0.82	0.77	0.11

the classification performance of the Boolean decision table and alternative techniques (CART, RF, and XGBoost) is compared.

Table 32 shows that the algorithms can be compared with each other for different performance evaluation metrics. XGBoost algorithm has the highest AUC-ROC value, but XGBoost algorithm has the highest error in the classification table. While creating the classification table, the cut-off point training set prevalence value of 0.09 was used. By selecting different cut-off points, the metrics that depend on the classification table will change. The highest accuracy and specificity values for the classification table belong to the Random forest algorithm. AUC-PR values were almost the same for CART, XGBoost, BDT. The difference of XGBoost and RF algorithm from the other two methods is that it is an ensemble learning algorithm. The aim here is to improve the classification accuracy. CART and BDT, on the other hand, produce interpretable results as well as classification accuracy compared to ensemble learning algorithms. The disadvantage of the CART algorithm is that the interpretability feature becomes difficult when the number of variables or the number of leaves in the decision tree is high [44]. In the BDT, the number of variables does not affect interpretability since the variables are represented by minterms. Boolean decision table results, whose classification accuracy is close to XGBoost and CART algorithms, can also be interpreted with confidence. For this, TOPSIS method is used to compare the training, test, and control set results.

With the TOPSIS method, the results of the best 5 minterms determined in the training set are compared with the values in the control set. The results of the best 5 minterms determined for the training set can be compared with the values in the control set and comments can be made.

Table 33. Correspondence of training set minterms in the control set

Rank	Minterm	Frequency-0	Frequency-1	Normalized Probability-0	Normalized Probability-1
1	21774872	260	53	0.83	0.17
2	25837848	116	20	0.85	0.15
3	9060632	35	8	0.81	0.19
4	25837864	173	10	0.95	0.05
5	9060888	138	19	0.88	0.12

Table 33 shows the performance of the top 5 minterms in the training set on the control set. The probability of belonging to class 1, which was 0.03 in the control set, was found to be higher in the minterms here. The analysis is said to be successful in the dataset with high classification performance.

A separate TOPSIS analysis is performed on the control set. The results here are compared with the results in the training set and the test set, and the minterms common to all three sets can be interpreted as the best. While creating the decision table, the total number of observations for 0-class and 1-class are taken as equal to or greater than 2. Thus, 3104 unique combinations were obtained for the control set.

Table 34 displays the top 5 minterms from the control set TOPSIS analysis. Table 34 shows that the probability of belonging to class 1, which was initially 0.03, is much higher for the minterms here. Since the minterms here correspond to independent variable combinations, variable combinations can be determined. The variable order is important when creating the minterm column. As in the control

Table 34. TOPSIS results for the control dataset (top 5 ranks)

Rank	Minterm	Frequency-0	Frequency-1	Normalized Probability-0	Normalized Probability-1	TOPSIS Ratio
1	25838104	517	72	0.88	0.12	0.74
2	21774872	260	53	0.83	0.17	0.65
3	21758488	332	45	0.88	0.12	0.57
4	9175576	270	42	0.87	0.13	0.55
5	25952792	1004	44	0.95	0.05	0.54

set, training and test set, when the minterms are written in binary base, combinations are determined according to the values of the variables and target group analysis is performed.

When the results of the TOPSIS analysis performed separately for the training, test, and control sets are compared, the best minterms common to all three datasets are determined and the best variable combinations for the dataset are found.

Table 35. Common minterms in training, test, and control datasets

Rank	Training Set Minterm	Test Set Minterm	Control Set Minterm
1	21774872	9060632	25838104
2	25837848	21774872	21774872
3	9060632	9060648	21758488
4	25837864	25838104	9175576
5	9060888	9060888	25952792

As seen table 35, 21774872 minterms rank in the top five of each of the three datasets. This indicates that 21774872 is the best minterm. For the implemented e-commerce firm, 21774872 is the ideal minterm. The feature combination for 21774872 minterm is as seen in the table 27; it is interpreted that the probability of shopping is 0.28 for users whose user type is new, who access the website after 20:00, whose session duration is above the 3rd quartile value of 437 seconds and who access the website using the Google source-organic tool from a desktop device with a Chrome browser from the province of Istanbul, as can be seen. In this target audience, Thursday, Friday, and Saturday can be excluded because the day variable takes the value 0. This targeting can be done for days other than these three days. The best minterm for the e-commerce company that is implemented is thus determined. In subsequent advertising campaigns, target audiences are determined by taking into account the variable combinations corresponding to this minterm.

4. Conclusions

In this paper, the Boolean decision table (BDT) approach, a classification technique that used Boolean algebra to analyze binary-valued features, and compare its classification performance with CART, Random Forest, and XGBoost algorithms, which were widely used decision tree methods in the literature. When comparing, consideration was given to the models' interpretability as well as their classification performance (classification accuracy, ROC, and PR curve). Furthermore, future data that was not included in the training or test data sets was acquired, and this data set was also used to compare performance. Since the probability distribution of future data differs from that of past data, the test data prediction may not hold up. This situation is called dataset shift. The BDT and other methods were also compared according to the dataset shift situation. Thus, a more objective interpretation of the validity of the obtained results is made. The algorithm with the highest training success in the results was the Boolean decision table (BDT). Examining the test dataset's results, the BDT approach outperformed the RF algorithm and came in close the CART and XGBoost algorithms in terms of values. With hyperparameter optimization included, the CART, RF, and XGBoost algorithms require a significant amount of training time. With the Boolean decision table, this is not the case. Taking into account the training duration and hyperparameter optimization procedure, employing the Boolean decision table becomes beneficial.

One of the most important advantages of proposing and using BDT is that the results obtained are interpretable. Interpretability is the ability to focus on the rules that emerge from the classification analysis rather than just the classifier model's performance [21]. Interpretable models attempt to build connections between the results obtained in addition to concentrating on classification accuracy. Results obtained with the proposed BDT method are easier to interpret. The most effective classification rules were identified using BDT. Furthermore, the interpretability of BDT's results increases confidence that the model recognizes the right patterns, which helps address the issue of dataset shifting. Using TOPSIS, the optimal variable combinations were also ascertained. Target audiences that are significant for the digital marketing industry are identified based on the combinations of variables. For example, the feature combination for minterm 21774872; it is interpreted that the probability of shopping is 0.28 for users whose user type is new, who access the website after 20:00, whose session duration is above the 3rd quartile value of 437 seconds and who access the website using the Google source-organic tool from a desktop device with a Chrome browser from the province of Istanbul. Considering the defined probability of shopping of 0.09 in the dataset, as a result of the analysis, variable combinations that increase this probability to 0.28 were found. Variable combinations can also be determined with CART analysis, but the large number of observations and variables makes such interpretations difficult. Another feature of CART analysis is that it can also be interpreted visually, but again, the high number of observations and variables makes visual interpretation difficult. RF and XGBoost algorithms, which are ensemble learning methods, focus only on classification accuracy and cannot be interpreted according to variable combinations. The most important feature of using and proposing BDT is that both the classification accuracy is close to other methods and the results obtained are interpretable. The high classification accuracy makes the interpretation of the results more reliable. In this way, the decision maker evaluates the results more objectively. In the case of digital advertising, the decision maker shapes future advertising strategies by determining appropriate target audiences within the framework of these evaluations. As a result of these evaluations, the decision maker can minimize advertising costs and get more return.

Real-world data unique to digital advertising was examined using the suggested BDT technique, yielding domain-specific outcomes. The results' interpretability boosts the model's confidence in identifying the right patterns because the suggested BDT method makes the classification results more interpretable. Finding the appropriate patterns is crucial in the realm of digital advertising in order to create target audiences that are pertinent. The suggested approach is better understood thanks to the application in this field. In future studies, the results of the proposed method can be compared by using simulation studies with real world data, artificial data, and different class distribution scenarios, which are frequently used in the literature. Necessary modifications for the method can be identified and suggested.

Author Contribution Statements All authors contributed equally in writing this article.

Declaration of Competing Interests No potential conflict of interest was reported by the author.

Acknowledgements The authors would like to thank the editor and anonymous reviewers for their constructive comments which led to the improvement of the paper.

References

[1] Aarthi, G., Karthikha, R., Sankar, S., Priya, S. S., Jamal, D. N., Banu, W. A., Application of Machine Learning in Customer Services and E-commerce. Data Management, Data Management, Analytics and Innovation, Springer, Singapore, 2023.

[2] Aruldoss, M., Lakshmi, T. M., Venkatesan, V. P., A Survey on multi criteria decision making methods and its applications, *American Journal of Information Systems*, 1(1) (2013), 31-43.

[3] Becker, B. G., Visualizing decision table classifiers, In Proceedings IEEE Symposium on Information Visualization, (1998).

[4] Becker, C., Rigamonti, R., Lepetit, V., Fua, P., Supervised Feature Learning for Curvilinear Structure Segmentation, In Medical Image Computing and Computer-Assisted Intervention - MICCAI, Springer Berlin, 2013.

[5] Bianchi, F., Garnett, E., Dorsel, C., Aveyard, P., Jebb, S. A., Restructuring physical micro-environments to reduce the demand for meat: a systematic review and qualitative comparative analysis, *The Lancet Planetary Health*, 2(9) (2018), e384–e397.

[6] Blackman, T., Dunstan, K., Qualitative comparative analysis and health inequalities: investigating reasons for differential progress with narrowing local gaps in mortality, *Journal of Social Policy*, 39(3) (2010), 359–373.

[7] Boole, G., The Calculus of Logic, (1848).

[8] Breiman, L., Bagging Predictors, *Machine Learning*, 24 (1996), 123-140.

[9] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., Classification and Regression Trees, Routledge, New York, 1984.

[10] Breiman, L., Random Forests, *Machine Learning*, 45(1) (2001), 5-32.

[11] Chen, T., Guestrin, C., XGBoost: a scalable tree boosting system, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2016), 785–794.

[12] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Junyuan Xie, Lin, M., Geng, Y., Li, Y., Xgboost: Extreme Gradient Boosting, R package version 1.4.1.1, (2021).

[13] Chiu, C., Ku, Y., Lie, T., Chen, Y., Internet auction fraud detection using social network analysis and classification tree approaches, *International Journal of Electronic Commerce*, 15(3) (2011), 123–147.

[14] Cramer, J. S., Logit Models from Economics and Other Fields, Cambridge University Press, 2003.

[15] Davis, J., Goadrich, M., The relationship between precision-recall and ROC curves, Proceedings of the 23rd International Conference on Machine learning, (2006), 233–240.

[16] Ekelik, H., Emir, Ş., A comparison of machine learning classifiers for evaluation of remarketing audiences in e-commerce, *Eskişehir Osmangazi University Journal of Economics and Administrative Sciences*, 16(2) (2021), 341–359.

[17] Fawcett, T., An introduction to ROC analysis, *Pattern Recognition Letters*, 27(8) (2006), 861–874.

[18] Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., Herrera, F., Learning from Imbalanced Data Sets, Springer Nature, Switzerland, 2018.

[19] Fratta, L., Montanari, U., A Boolean algebra method for computing the terminal reliability in a communication network, *IEEE Transactions on Circuit Theory*, 20(3) (1973), 203–211.

[20] Freeman, E. A., Moisen, G. G., A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa, *Ecological Modelling*, 217(1) (2008), 48-58.

[21] Freitas, A. A., Comprehensible classification models: a position paper, *ACM SIGKDD Explorations Newsletter*, 15(1) (2014), 1–10.

[22] Friedman, J. H., Greedy function approximation: a gradient boosting machine, *Annals of statistics*, (2001), 1189–1232.

[23] Friedman, J. H., Stochastic gradient boosting, *Computational Statistics & Data Analysis*, 38(4) (2002), 367–378.

[24] Gran, B., Aliberti, D., The office of the children's ombudsperson: children's rights and social-policy innovation, *International Journal of the Sociology of Law*, 31(2) (2003), 89–106.

[25] Hailperin, T., Boole's algebra isn't Boolean algebra, *Mathematics Magazine*, 54(4) (1981), 173–184.

[26] Han, J., Kamber, M., Pei, J., Data Mining: Concepts and Techniques, 3rd ed., Morgan Kaufmann, 2012.

[27] Hastie, T., Tisbshirani, R., Friedman, J., The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, New York, 2018.

[28] Ho, T. K., The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8) (1998), 832–844.

[29] Hosmer, D. W., Lemeshow, S., Sturdivant, R. X., Applied Logistic Regression, 3rd ed., John Wiley & Sons, 2013.

[30] Huang, S. H., Supervised feature selection: A tutorial, *Artificial Intelligence Research*, 4(2) (2015), 22-37.

[31] James, G., Witten, D., Hastie, T., Tibshirani, R., An Introduction to Statistical Learning: with Applications in R, Springer, New York, 2013.

[32] Jiménez-Hernández, E. M., Oktaba, H., Díaz-Barriga, F., Piattini, M., Using web-based gamified software to learn Boolean algebra simplification in a blended learning setting, *Computer Applications in Engineering Education*, 28(6) (2020), 1591–1611.

[33] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T. Y., LightGBM: a highly efficient gradient boosting decision tree, Proceedings of the 31st International Conference on Neural Information Processing Systems, (2017), 3149–3157.

[34] Kim, J. W., Lee, B. H., Shaw, M. J., Chang, H. L., Nelson, M., Application of decision-tree induction techniques to personalized advertisements on internet storefronts, *International Journal of Electronic Commerce*, 5(3) (2001), 45–62.

[35] Kleinbaum, D. G., Klein, M., Logistic Regression A Self-Learning Text, 3rd ed., Springer New York, 2010.

[36] Kumar, P., A Study on interconnection between Boolean algebra and binary tree, *Globus an International Journal of Management & IT*, 9(2) (2018), 1–2.

[37] Kumar, R., Lawrance, R., Boolean Rule Based Classification for Microarray Gene Expression Data, International Journal of Recent Technology and Engineering, 2019.

[38] Larose, D. T., Larose, C. D., Discovering Knowledge in Data: an Introduction to Data Mining, 2nd ed., John Wiley & Sons, 2014.

[39] Liaw, A., Wiener, M., Classification and regression by randomForest, *R News*, 2(3) (2002), 18-22.

[40] Lima, E., Mues, C., Baesens, B., Domain knowledge integration in data mining using decision tables: case studies in churn prediction, *Journal of the Operational Research Society*, 60(8) (2009), 1096–1106.

[41] Lindaman, R., A theorem for deriving majority-logic networks within an augmented Boolean algebra, *IRE Transactions on Electronic Computers*, 3 (1960), 338–342.

[42] Liu, C. J., Huang, T. S., Ho, P. T., Huang, J. C., Hsieh, C. T., Machine learning-based e-commerce platform repurchase customer prediction model, *PLOS ONE*, 15(12) (2020), e0243105.

[43] Lu, H., Liu, H., Decision tables: scalable classification exploring RDBMS capabilities, Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt, 2000.

[44] Maimon, O., Rokach, L., Data Mining and Knowledge Discovery Handbook, 2nd ed., Springer, 2010.

[45] Mitchell, R., Frank, E., Accelerating the XGBoost algorithm using GPU computing, *PeerJ Computer Science*, 3 (2017), e127.

[46] Muller, D. E., Application of Boolean algebra to switching circuit design and to error detection, *Transactions of the IRE Professional Group on Electronic Computers*, 3 (1954), 6–12.

[47] Ogihara, H., Fujita, Y., Hamamoto, Y., Iizuka, N., Oka, M., Classification based on boolean algebra and its application to the prediction of recurrence of liver cancer, 2nd IAPR Asian Conference on Pattern, 2013.

[48] Özcan, T., Çelebi, N., Esnaf, Ş., Comparative analysis of multi-criteria decision making methodologies and implementation of a warehouse location selection problem, *Expert Systems with Applications*, 38(8) (2011), 9773-9779.

[49] Phiffer, P. E., Concepts of Probability Theory, Second Revised Edition, Dover Publications, New York, 1978.

[50] Provost, F., Fawcett, T., Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions, Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, (1997), 43–48.

[51] Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N. D., Dataset Shift in Machine Learning, The MIT Press, 2008.

[52] Ragin, C. C., The Comparative Method, University of California Press, 2014.

[53] Rokach, L., Maimon, O., Data Mining with Decision Trees: Theory and Applications, 2nd ed., World Scientific Publishing, 2015.

[54] Rokach, L., Pattern Classification Using Ensemble Methods, Singapore, World Scientific Publishing, 2010.

[55] Rushdi, A. M., Zagzoog, S. S., Balamesh, A. S., Design of a hardware circuit for integer factorization using a big Boolean algebra, *Journal of Advances in Mathematics and Computer Science*, (2018), 1–25.

[56] Son, J., Jung, I., Park, K., Han, B., Tracking-by-Segmentation with Online Gradient Boosting Decision Tree, IEEE International Conference on Computer Vision (ICCV), 2015.

[57] Tekin, M., Calculation of Probabilities of Some Statistical Events with the Help of Boolean Algebra, Unpublished Master's Thesis, Istanbul University, Institute of Social Sciences, 1989.

[58] Therneau, T., Atkinson, B., rpart: Recursive Partitioning and Regression Trees, R package version 4.1-15, (2019).

[59] Thomas, R., Boolean formalization of genetic control circuits, *Journal of Theoretical Biology*, 42(3) (1973), 563–585.

[60] Vis, B., Under which conditions does spending on active labor market policies increase? An fsQCA analysis of 53 governments between 1985 and 2003, *European Political Science Review*, 3(2) (2011), 229–252.

[61] Wang, R. S., Saadatpour, A., Albert, R., Boolean modeling in systems biology: an overview of methodology and applications, *Physical Biology*, 9(5) (2012), 055001.

[62] Xiao, Y., Mehrotra, K. G., Mohan, C. K., Efficient classification of binary data stream with concept drifting using conjunction rule based boolean classifier, In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2015.

[63] Zhang, Q., Li, Z., Boolean algebra of two-dimensional continua with arbitrarily complex topology, *Mathematics of Computation*, 89(325) (2020), 2333–2364.

[64] Zheng, A., Casari, A., Feature Engineering for Machine Learning, O'Reilly Media, California, 2018.