Middle East Journal of Science

https://dergipark.org.tr/mejs

MEJS

e-ISSN:2618-6136

Research Article

MACHINE LEARNING APPROACH TOWARD TELEMARKETING ESTIMATION

Mehmet SALTİ[1] ***Evrim Ersin KANGAL[2]*** ***Bilgin ZENGIN[3]***

[1]Department of Business Information Management, Graduate School of Social Sciences, Mersin University, Mersin, Türkiye
[2]Computer Technology and Information Systems, School of Applied Technology and Management of Erdemli, Mersin University, Mersin, Türkiye
[3]Department of Electrical and Electronic Engineering, Faculty of Engineering, Munzur University, Tunceli, Türkiye
Corresponding author;evrimersin@gmail.com

Abstract: *Machine learning empowers us to extract insights from large datasets beyond human capacity. It involves training computers to identify patterns within data, enabling them to glean valuable information and apply it to novel tasks. This study focuses on analyzing a specific telemarketing dataset using various machine learning algorithms to determine if accurate predictions can be made to support company decision-making. The findings highlight that customer "Age" and "Product ID" are the primary factors influencing "Sales" numbers, indicating their significance in the predictive model.*

Keywords: *Data mining, machine learning, kNN, telemarketing.*

Received: January 28, 2024 *Accepted: April 3, 2024*

1. Introduction

Marketing campaigns are constructed via a typical strategy to enhance business capacity, which is a measure of the amount of work that may be overcome by a company within a fixed amount of time [1]. Contacting different types of customers to meet specific targets, companies use direct strategies while communicating with them. On the way to this goal, companies prefer to centralize their remote interactions with customers in a contact center to facilitate the operational management of campaigns. Such contact centers communicate with potential customers through various channels: telephone (landline or mobile) is the most widely used tool in this context. Marketing carried out through a contact center is called telemarketing in literature because of its remoteness feature [2]. Depending on which factor (customer or contact center) triggers, people are divided into two groups inbound and outbound. It is worth emphasizing here that each situation presents different challenges (for instance, outgoing calls are often seen as annoying). Technology requires us to rethink marketing methods and aim to increase the value of a customer by addressing information and demands from different perspectives, thus it enables us to establish longer and tighter relationships in line with business demands [3].
Developing technology has led to an exponential increase in the global data volume. One of the sources from which intensive scientific data is produced today is the large hadron collider (LHC) at the European Organization for Nuclear Research (CERN) [4]. The LHC produces a continuous data stream of 40 Tb/s [5]. Such datasets are called “Big Data” [6] in literature. As expected, the number of recorded features of samples also increases due to an aggressive enlargement in the data volume. On the other hand, the

increase in the recording area causes “the curse of dimensionality” [7]. Marron and Maine [8] thought that this problem would be overcome with the data compression method, but in the following years, it was revealed that the data compression causes some information to be lost [9-11]. Data analysis through artificial intelligence (AI) has gained considerable popularity in recent years. AI is generally taken into account in literature to identify the artificial creation of human brains that can learn, plan, perceive, or process [12] and it is basically the development of computer systems capable of performing tasks (for instance visual perception, speech recognition, decision-making, and language translation) that need human intelligence [13]. ML, Neural Networks, and Deep Learning (DL) are sometimes used interchangeably, but there are some nuances among them. All of these concepts are actually sub-fields of the AI: the DL is a sub-branch of Neural Networks while a neural network approach is a sub-branch of the ML. The AI approach has noteworthy potential to increase the level of our knowledge in diverse fields such as finance [14], business management [15], meteorology [16], neuroscience [17], demoscopic analysis [18], quantum chemistry [19], spacecraft engineering [20], astrophysics [21-24], cosmology [25-27], etc.

The ML algorithms are divided into two main parts: the supervised architectures and the unsupervised ones [28,29]. The supervised learning mechanisms try to establish a relationship, which is organized by the classification or regression algorithms, between the input data and the corresponding output. On the contrary, in the case of unsupervised learning, there is no use of any training set and data, which are divided into different branches according to similarity criteria involving data features. In addition, an ordinary decision support system (DSS) focuses on information technology to support the managerial decision-making phase and there are various DSS types, such as personal and intelligent DSSs [1]. A personal DSS takes an effective role in relatively small-scale situations (for example, a manager's decision-making task), while an intelligent DSS uses AI algorithms to support decisions [30]. Business intelligence (BI) is another important DSS concept. The BI is generally accepted as an umbrella term that supports decision-making tasks by making use of business data and consists of information technologies such as data warehouses and data mining [31]. Classification algorithms used commonly in the field can be listed as follows [32-36]: the Decision Trees (DTs), Linear Regression (LiR), Logistic Regression (LoR), Support Vector Machines (SVMs), Naive Bayes Classifier (NBC), Neural Networks, Instance-based Learning (IBL) and the k-Nearest Neighbors (kNN).

Telemarketing is a sort of direct marketing where salespersons communicate with prospective customers to acquire new customers, vend their products, and provide their services over the phone [2]. Since only the direct marketing dataset helps us to reach the database of prospective customers, it is very significant for a company to estimate successfully the group of potential clients with the highest prospect to admit the sales and/or offer according to their individual characteristics or attitude while shopping [37]. Therefore, many companies have recently started to focus on data mining methods for customer classification while many scientists have taken diverse ML applications into account for the different telemarketing databases. Tekouabou et al. [38] introduced a new data modeling approach to optimize the estimation of telemarketing target calls for selling bank long-term deposits. Keles and Keles [39] are interested in constructing an Intelligent Bank Market Management System based on the ML approaches to operate marketing campaigns efficiently. Kocoglu and Esnaf [40] developed a model to classify the success of telemarketing with various ML approaches, such as the NBC, C5.0, Extreme Learning Machine (ELM), and the DL. Considering a telemarketing dataset, Halim et al. [41] concluded that the Neural Networks approach, which is supported with data cleaning algorithms such as the Missing Common and the Tomek Links, indicates a better conclusion compared to the Ignore Missing mechanism. Shashidhara et al. [42] have investigated the most appropriate model for the analysis of marketing data in the banking sector with ML architectures.

In the present study, the focus is aimed at the wrapper-based kNN algorithm for conducting data analysis in a telemarketing case. The outline of this paper is as follows. In the second section, the methodology and the selected data pool are introduced. In the third section, the ML architecture is constructed, and the selected performance testing method is introduced. Next, in the fourth section, attention is given to preprocessing and visualization of the dataset to decipher some hidden features of the case. The constructed wrapper-based kNN algorithm is tuned in the fifth section. In the sixth section, the architecture is run. The final section is devoted to final remarks. It is noted that Python Anaconda 3.7 version is used for coding all algorithms and as an interpreter. Furthermore, the Pandas, Seaborne, Scikit, Numpy, and Matplotlib libraries are taken into account extensively in the phase of architecture implementation.

2. Methodology and Dataset

Increasing campaign costs and low marketing-response rates drive companies to model customer behavior to obtain successful sales statistics, but this requirement often forces them to work with complex methods. It is known that predicting a customer's reaction to marketing may provide a significant advantage to a company before the campaign. Motivated by this situation, the problem of telemarketing success classification is discussed within the scope of our study. It is carried out the investigation in accordance with the well-known methodology CRoss Industry Standard Process for Data Mining (CRISP-DM), which has six sequential phases [43,44]:

- Understanding the problem - What the need of business is?
- Deciphering the data - What data do we have? What do we need?
- Data preparation - Is it clean? How do we organize it before modeling?
- Developing the appropriate techniques - Which models can we apply?
- Evaluating performance of the approach - Which algorithm best meets the goals?
- Application - How do stakeholders achieve the goals?

In this study, we focus on an anonymous telemarketing case, which is taken from the Kaggle platform and includes high-quality datasets[45]. The selected data pool is represented by a table that includes 12 feature columns and 100000 rows of customer information. A part of the aforementioned dataset is presented in Table 1 as an example.

Table 1. A sample is extracted from the dataset. [45]

Call ID	Sales	Agent ID	Age	Product ID	Time Zone	Phone Code	First Name	Last Name	Area Code	Gender	Call Count
9545434	False	5265	42	147	2	37	Jk	Jk	2302	Male	1
9211206	False	5226	74	146	2	37	Em	Sh	1501	Male	10
8873010	False	4452	35	144	2	37	BI	MI	1550	Male	9
9852034	False	5461	40	149	2	37	WT	LI	1401	Male	6
9416548	False	5298	26	147	2	37	LA	LA	125	Female	12
10189322	False	5139	33	150	2	37	Me	Is	4091	Female	1
10277850	False	4828	33	151	2	37	So	Ts	3880	Female	3
9105514	False	5292	70	145	2	37	Dy	Ma	432	Female	9
8663012	False	5044	41	143	2	37	No	Mi	4360	Female	9
10216124	True	4912	72	150	2	37	TA	MS	1983	Male	1

It would be appropriate to emphasize here that extremely serious problems may arise for the ML algorithms at the distribution classification phase. If the selected architecture is trained via the samples identified with insufficient label information, the algorithm is likely to make useless predictions. It is seen that attributes in Table 1 are represented via different data types (please check Table 2).

Table 2. Attribute types in the dataset

Attribute	Type	Attribute	Type
Call ID	Integer	Phone Code	Integer
Sales	Categorical	First Name	Categorical
Agent ID	Integer	Last Name	Categorical
Age	Integer	Area Code	Integer
Product ID	Integer	Gender	Categorical
Time Zone	Integer	Call Count	Integer

In this statistical analysis, the attribute "Sales" is selected as the target feature while all remaining attributes are taken into account as arguments. It is obvious that the “Time Zone”, Phone Code”, “First Name”, “Last Name” and “Call ID” attributes do not significantly affect the attribute “Sales”. Subsequently, in the entire table, some customers have deficiencies in some of their information and we conclude that missing information constitutes approximately 3% of the entire data pool. Accordingly, the data cleaning process to be performed in Table 1 enables faster and more consistent completion of our analyzes. Our dataset now contains 223 Agent IDs, 10 product variants, and 97205 customers. On the other hand, it is mainly to discuss whether an ML pattern can be constructed among the independent variables and the selected target parameter, thus the categorical information in Table 1 needs to be transformed into binary values with the help of dummy variables. Once the label encoder tool in Python has been applied, restructuring the dataset becomes a much more straightforward task. The reader may check Table 3 to see the assumptions we used for the numerical conversion of the categorical attributes.

Table 3. Numerical conversion of the categorical information

Feature	Gender (Male)	Gender (Female)	Gender (Non-binary)	Sales (True)	Sales (False)
Value	1	0	2	1	0

Moreover, category imbalance is another situation that needs to be carefully addressed in data science research. Most of the classification algorithms perform analysis by assuming that the presented raw dataset is balanced, but this is not the case often. Some of the categories (classes) defined in the repository may have very few elements while others may be represented by too many elements. In such cases, the selected classification algorithm yields serious issues and makes erroneous predictions, because it cannot be trained well enough for dataset elements with low characteristic information. In classification approaches, the main goal is to maximize the beneficial prediction rate, thus necessitating the balancing of attribute distributions through a suitable technique. For this purpose, in FIG. 1, it is discussed as a distribution of the results given in the "Sales" column of Table 1 and see that the number of “False (0)” cases is much higher than the “True (1)” ones. Such a situation may affect negatively the performance of our ML approach; Hence, the need to synthesize new "True (1)" cases via a suitable Python package is recognized to address this distribution problem.

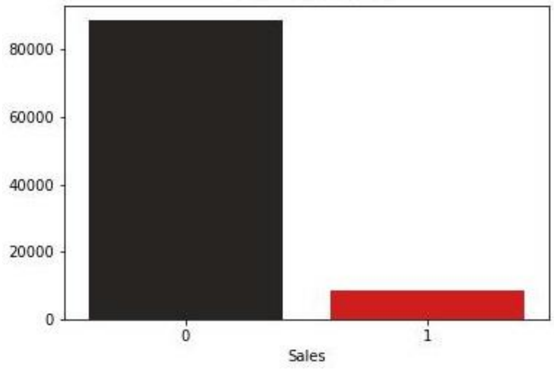

Figure 1. The number of "False (0)" and "True (1)" sales

3. Architecture

Nearest neighbor-based algorithms produce very successful results in the classification research and they treat the information obtained from the dataset in the form of specific situations or experiences. As they rely on efficient matching methods to retrieve stored information, they can be applied to new situations. The main purpose of such methods is to determine the nearest neighbors of a new data point and then place it in the correct class. Therefore, for such approaches, it is of great importance to calculate the shortest distance between two data points. We assume here that the function $d(x, y)$ basically identifies the distance between two different feature vectors $A = (x_1, x_2, \dots, x_k)$ and $B = (y_1, y_2, \dots, y_k)$ as a non-negative real number. A distance function is assumed to be a metric if it obeys a certain number of features that consist of the following conditions: positivity (or non-negativity), the identity of the inseparable, symmetry, and triangle inequality (the reader may check Refs. [46,47] for detailed information).

Example-based ML algorithms are computationally simple and generally recognized as different types of human learning [48]. The Instance-based Learning (IBL) is a simple nearest neighbor-based ML algorithm [49]. Here, the closest data point to the sample to be classified is searched according to the Euclidean formalism and then the sample is assigned to the class of that data point. On the other hand, the class of the unknown sample is determined by the majority of its neighbors. It is seen from FIG. 2 that, in the K_1 region, new data points are estimated as members of group B since the purple square data points are the dominant. Next, the majority of orange hexagonal data points in the K_2 region indicate that the new data points are members of group A.

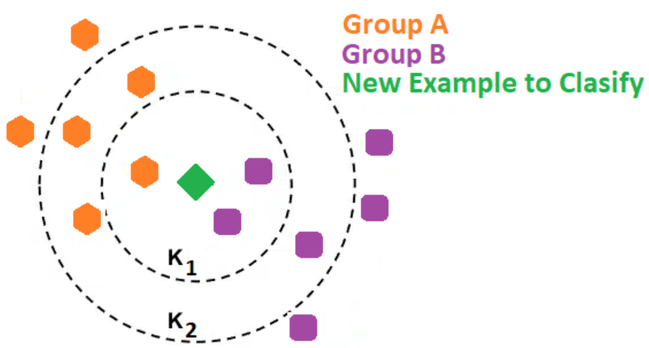

Figure 2. A representative visualization of the nearest neighbor-based classification

The related distance measurement metric used by the IBL algorithm is expressed as follows [50]:

$$d_{Euclidian}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}, \tag{1}$$

where x_i and y_i represent the i-th values of x and y samples, respectively.

The learning process is based on the selected training data also in the kNN algorithm, essentially the kNN mechanism is a more general form of the IBL approach. Here, k represents the number of nearest neighbors. Unlike the IBL approach, the distance between the two closest data points in the kNN mechanism can be measured by various methods. From this point of view, we complete our ML analysis according to kNN algorithms designed according to the 8 most used distance measurement formalisms, including the Euclidean metric. We consider the Euclidean, CityBlock (also known as the Manhattan, Taxicab, Rectilinear, or the L_1-norm), Chebyshev, Correlation, Canberra, Dice, and the Cosine measurements during the patterning phase in our research. Here, we have [50,51]

$$d_{CityBlock}(x,y) = \sum_{i=1}^{n}|x_i - y_i|, \tag{2}$$

$$d_{Chebyshev}(x,y) = \max_i |x_i - y_i|, \tag{3}$$

$$d_{Correlation}(x,y) = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}\right), \tag{4}$$

$$d_{Canberra}(x,y) = \sum_{i=1}^{n}\frac{|x_i-y_i|}{|x_i|+|y_i|}, \tag{5}$$

$$d_{Dice}(x,y) = 1 - \frac{2\sum_{i=1}^{n}x_i y_i}{\sum_{i=1}^{n}x_i^2+\sum_{i=1}^{n}y_i^2}, \tag{6}$$

$$d_{Cosine}(x,y) = 1 - \frac{\sum_{i=1}^{n}x_i y_i}{\sqrt{\sum_{i=1}^{n}x_i^2}\sqrt{\sum_{i=1}^{n}y_i^2}}, \tag{7}$$

where n represents a dimension of the feature space and the upper bar indicates the mean value of the relevant quantity.

The increase in the number of features not only negatively affects the algorithm in the decision phase and but also prolongs the compilation time of the mechanism. This issue is called “the curse of dimensionality” in literature [52-54] and can be removed by making use of the Feature Selection (FeSe) approach, which is one of the widely used solutions in the field. The essence of such methods is based on the determination of attributes that effectively influence the decision-support system. Consequently, unnecessary features are eliminated from the selected dataset and the ML pattern required for the desired performance is now formed more easily [55]. In this investigation, the wrapper-based supervised FeSe approach is considered. According to this method, feature subsets are created first, and then the results of each selected subset are compared with random selections. For this purpose, four different wrapper types are used to achieve the best results: the Sequential Backward Selection (SBS), the Sequential Forward Selection (SFS), the Sequential Backward Floating Selection (SBFS), and the Sequential Forward Floating Selection (SFFS). The algorithm creates a universal set by removing subsets one by one in the SBS approach while a universal attribute pool is created by adding the attributes one by one to the dataset in the SFS mechanism [56,57]. The main point here is that the features covered are selected randomly from the feature space. On the other hand, both the SBFS and SFFS methods were born as alternative models to the Plus-L-Take-Away-R (or the Plus-L-Minus-R) algorithm in essence [58]. Since there is no a theoretical approach to determine the values of the parameters L and R, the performance percentage of an ML algorithm becomes directly dependent on these auxiliary parameters, which means

the choice of L and R is the key to success of this algorithm [59]. In order to eliminate this dependence, the floating (or moving) L and R values are usually created instead of the fixed ones [60].
The workflow of our architecture is schematically illustrated in FIG. 3 and one can check here the main procedure of the selected ML algorithm.

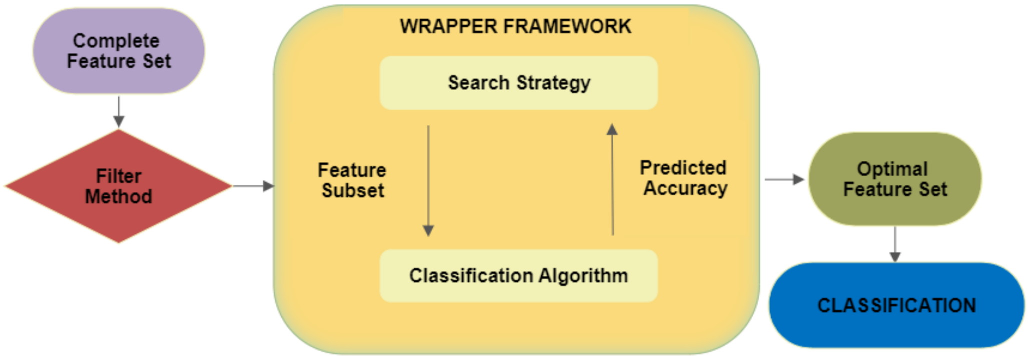

Figure 3. Flow chart of the wrapper-based kNN mechanism

There are 8 different distance measurement metrics, 4 types of wrappers and different *k* values and it is aimed to build a successful kNN algorithm. For this reason, there is a need for a performance test method that can allow us to compare different algorithm combinations.

		Estimated Data			Totals
		Positive (1)	**Negative (0)**		
Actual Data	**Positive (1)**	***TP***	***FN***	Sensitivity (Positive Recall) $\frac{TP}{TP+FN}$	Actual Positives $P = TP + FN$
	Negative (0)	***FP***	***TN***	Specificity (Negative Recall) $\frac{TN}{FP+TN}$	Actual Negatives $N = FP + TN$
		Positive Precision $\frac{TP}{TP+FP}$	Negative Precision $\frac{TN}{TN+FN}$	Accuracy $\frac{TP+TN}{TP+FN+FP+TN}$	
	Totals	Estimated Positives $P = TP + FN$	Estimated Negatives $N = FN + TN$		

Figure 4. A general representation of confusion matrix

Performance evaluation of an ML processes can be made by means of a number of metric criteria [61]. The confusion matrix is among the well-known criteria and it is structured in a way that we can obtain True and False values as a result of classification. In FIG. 4, we present a general representation of confusion matrix. Here, rows correspond to actual data while columns imply predicted data. Also, the abbreviations TP, FN, FP and TN mean respectively True Positive (we have actual positivity in the data, which has been estimated correctly as positive by the selected model), False Negative (there is an actual negativity in the data, which has been predicted accurately as negative by the algorithm), False Positive (we have actual negativity in data, but the selected approach has predicted it as positive) and True Negative (there is an actual positivity in data, but the model has estimated it as negative). In the light of these definitions, various mathematical metrics such as the ACC (Automatically determining the Cluster Centers) [62], RAC (Relative Angle Correction) [63] and F-Measure (or F1-score) [64] can also be used to test the performance of an ML approach.

4. Preprocessing and Visualization of Data

An unbalanced dataset may be a serious source of inconsistency when trying to use the selected ML algorithm in the decision-support phase. There are two main ways to eliminate this issue: under-sampling and over-sampling. On this purpose, one can use the SMOTE (Synthetic Minority Oversampling Technique), NCL (Neighborhood Cleaning Rule), OSS (One-Sided Selection) and the BootsOS (Bootstrap-based Over Sampling) approach [65-69]. In FIG. 5, it is illustrated as the process of creating a synthetic dataset via the under-sampling and over-sampling approaches. Here, the red group (A) is the minority class while the black one (B) is the majority class.

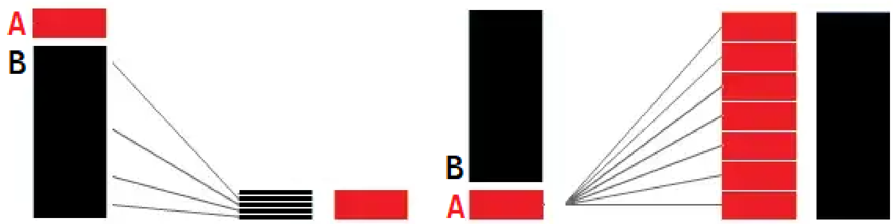

Figure 5. Under-sampling (left) and over-sampling (right) mechanisms

Based on the situation shown in FIG. 1, Sufficient synthetic "True (1)" cases for the attribute "Sales" are generated with the help of SMOTE. After this task, FIG. 6 is obtained, which indicates that the number of “True (1)” and “False (0)” sales are now balanced.

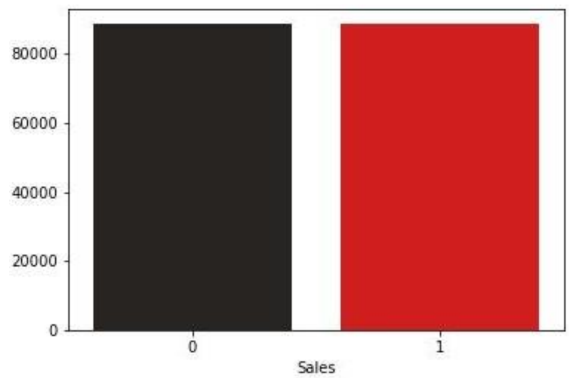

Figure 6. The number of “False Sales” (Black) and “True Sales” (Red) after the SMOTE

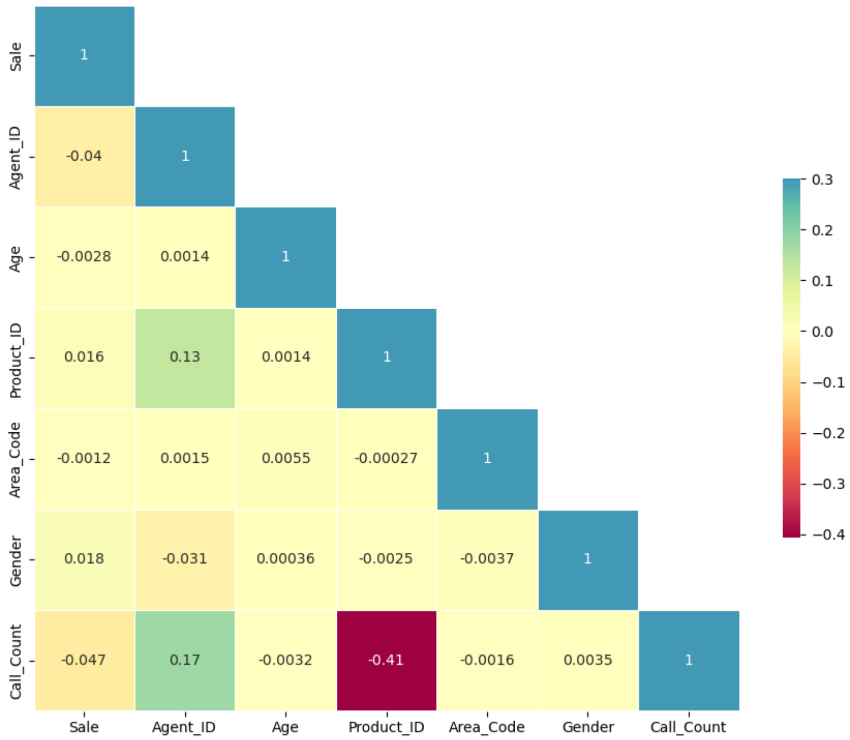

Figure 7. Correlation analysis of the attributes

Another important point that can negatively affect a ML analysis is the correlation between attributes. A correlation table includes columns and rows that represent variables of the selected dataset. A negative correlation between any two features is indicated when the corresponding correlation value falls within the range of [-1, -0.5]. On the other hand, there is a positive correlation between any two attributes if the correlation coefficient is in the range of [0.5, 1]. The main diagonal line that includes 1.00s implies that each variable always perfectly correlated with itself and other correlation values indicate no-correlation cases. FIG. 7 is created to reveal the positive and negative correlations among the attributes of our dataset clearly. It is seen that there is no issue that may adversely affect the performance of our ML analysis, since there is no connectivity problem among the attributes.

Over-sampling process may have changed other properties of the repository as well, hence controlling these changes can allow us to access additional information about the selected tele-marketing case. In FIG. 8, we check the gender distribution in the dataset. Subsequently, in FIG. 9, we aim to discuss how the number of customers is distributed by the attribute “Age” before and after the SMOTE. In our investigation, the SMOTE algorithm focused on the available raw data while synthesizing the attribute labeled "True Sales", thus an increase in the number of female customers was observed. This indicates that the positive marketing situations in the raw data set are mostly observed in the female customer profile.

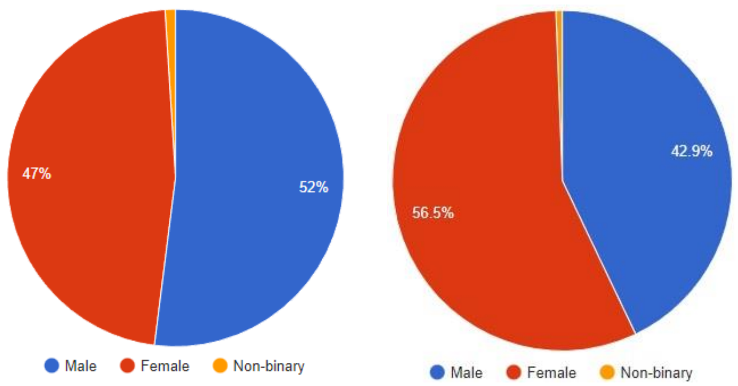

Figure 8. “Gender” distribution before (left) and after (right) the SMOTE

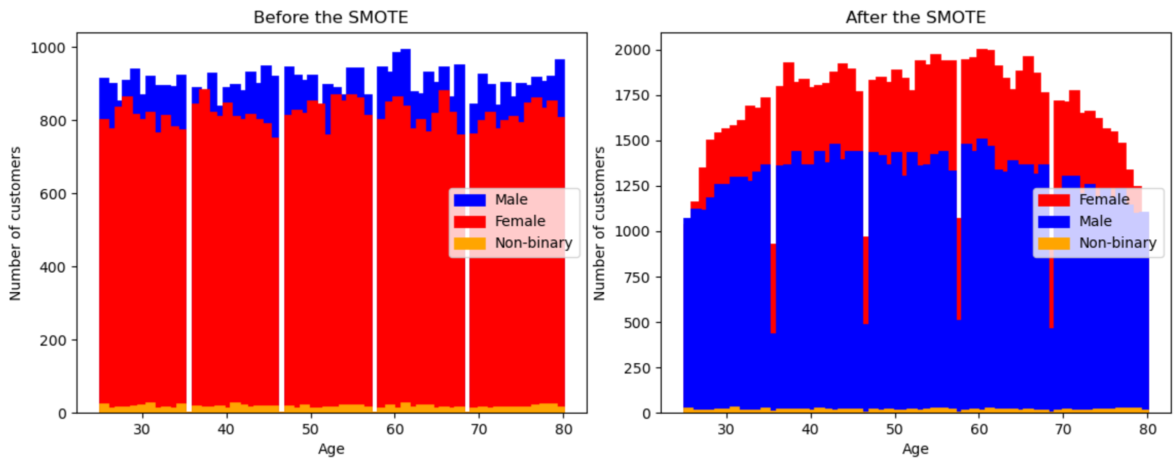

Figure 9. Distribution of the number of customers according to their ages

At FIGs. 10 and 11, it is discussed the distribution of the number of “Sales” according to the attributes “Agent ID” and “Product ID”. It is understood that some agents put more effort in reaching customers and some products are marketed more than others. In addition, it is seen that the SMOTE algorithm preserves characteristic features of the raw data. This is important because it tells us that the estimated results obtained as a result of the ML application can be trusted.

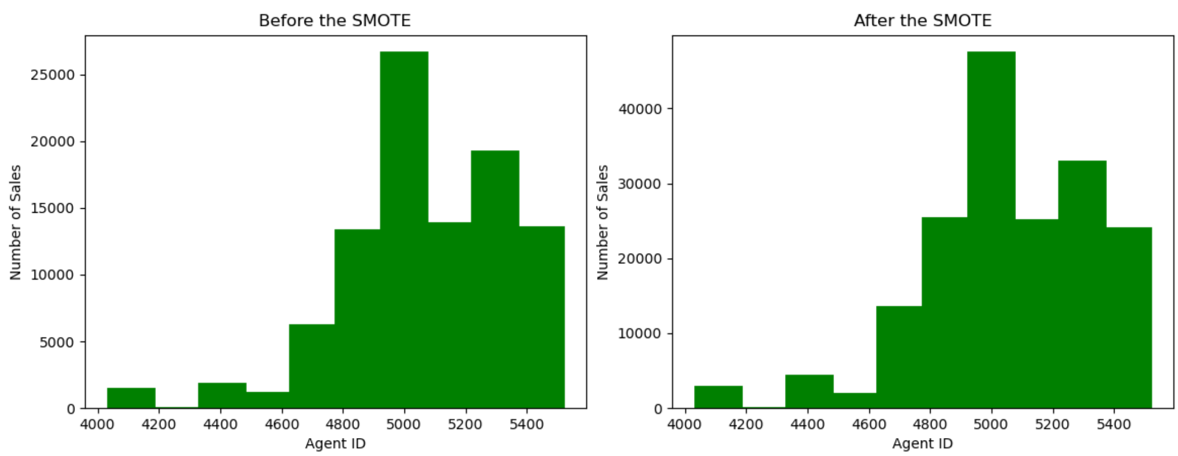

Figure 10. Number of “Sales” vs “Agent ID”

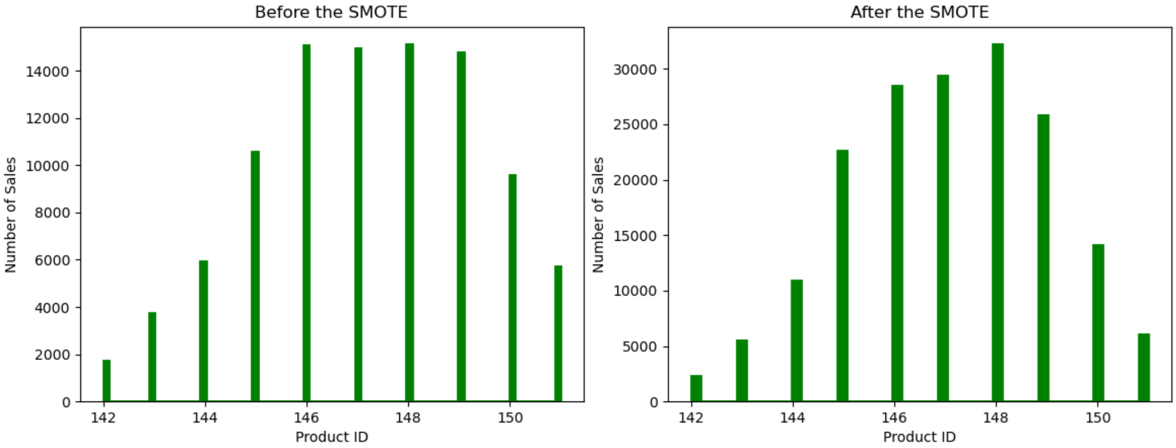

Figure 11. Number of "Sales" vs "Product ID"

5. Tune the Algorithm

A hyperparameter (the prefix "hyper" indicates here that such a parameter is top-level quantity) is a vital ML part that is explicitly defined by the user to control the learning process. For example, batch size, number of epochs, pooling size, learning rates, number of hidden layers, and the number of neighbors are some of the well-known hyperparameters [70-72].

In this part of the study, we do not only aim to determine the most reasonable value of the hyperparameter *k*, which plays an important role in the wrapper-based kNN algorithm, but also want to choose the appropriate "distance function" and "wrapper" formulations. In other words, the main idea is to figure out the best combination of algorithm parts. The general method is to use some of the available data as a tuning group in different algorithm combinations. Consequently, the various kNN algorithms were run by considering the train-test split ratio of 70-30, different values of *k*, seven kinds of distance function and four different wrapper types.

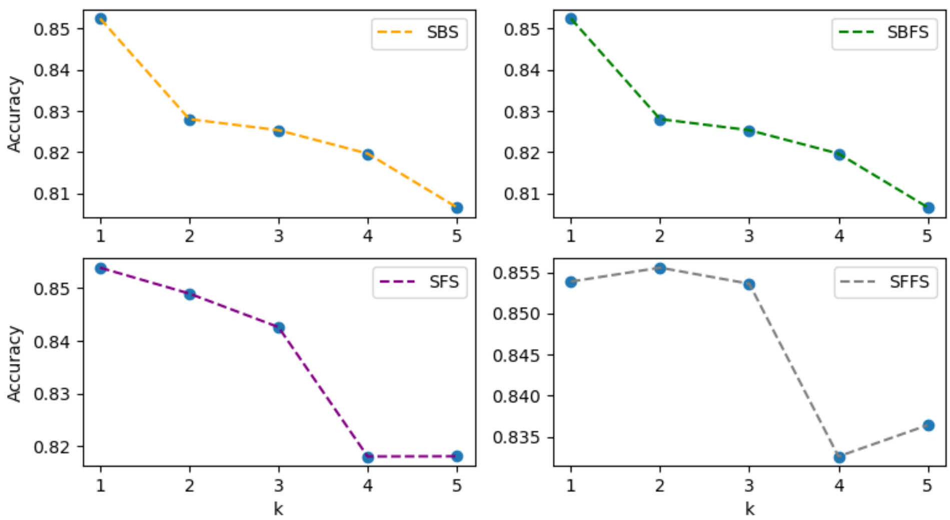

Figure 12. Performance analysis of the CityBlock measurement

Table 4. Accuracies obtained in the CityBlock analysis

k	SBS	SBFS	SFS	SFFS
1	0.852453	0.852453	0.853901	0.853901
2	0.828050	0.828050	0.848994	0.855593
3	0.825362	0.825362	0.842621	0.853638
4	0.819684	0.819684	0.818067	0.832581
5	0.806655	0.806655	0.818105	0.836398

Table 5. Tuning results according to the highest accuracy values

Distance Type	*k*	Wrapper	Accuracy (%)	Process Time (≈)
CityBlock	2	SFFS	85.6	13 minutes
Chebyshev	2	SFFS	85.1	1 hour
Euclidean	2	SFFS	85.3	45 minutes
Correlation	1	SFFS	87.5	7 hours
Canberra	2	SBS, SBFS, SFFS	91.6	9 hours
Dice	1	SFS, SFFS	62.6	11 hours
	5	SBS, SBFS, SFS, SFFS		
Cosine	2	SFS, SFFS	87.9	10 hours

For the CityBlock distance measurement function, we reach the results presented in FIG. 12 and Table 4. Subsequently, the same procedure is followed for the Chebyshev, Euclidean, Correlation, Canberra, Dice and the Cosine type distance measurements. It can be summarized the obtained noteworthy results in Table 5. In conclusion, the most appropriate combination of the architecture components and the best value of the hyper parameter *k* are now clearly revealed. The most significant differences among the performance of various distance measurement expressions are the maximum accuracy and the process time. Although the Correlation, Canberra and the Cosine formulations offer relatively higher prediction accuracy, it is understood that they require a remarkably long analysis time and a powerful set of computer hardware. Moreover, all types of kNN architecture give their best results while working with the SFFS type wrapper.

6. Run the kNN architectures

The ML analysis is now sought to be deepened to explore the implications of different kNN architectures on the attribute "Sales" in our telemarketing case. To achieve this goal, it is assumed that the train-test split ratio of the selected dataset is 70-30 again, and the kNN architectures consist of the SFFS type wrapper.

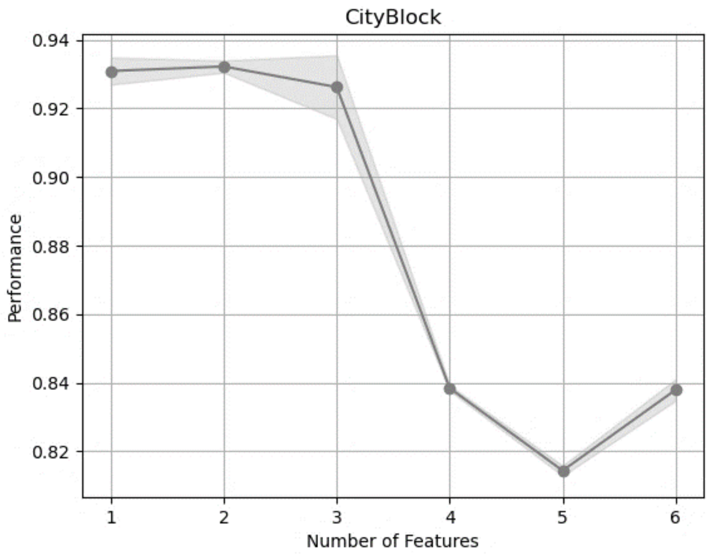

Figure 13. Performance analysis of the CityBlock_SFFS_kNN_vk2 algorithm

First, the performance of the CityBlock measurement based SFFS-kNN algorithm (we name this architecture CityBlock_SFFS_kNN_vk2) is obtained as illustrated in Figure 13. Subsequently, we give a detailed report of this 1-minute analysis in Table 6. It is basically concluded that the success rate of the CityBlock-based algorithm decreases as the number of features affecting the decision increases. According to the CityBlock_SFFS_kNN_vk2 architecture, the most important features affecting the attribute "Sales" are the "Age" of customer and the "Product ID".

Table 6. Detailed performance analysis of the CityBlock_SFFS_kNN_vk2 algorithm

Number of Features	**1**	**2**	**3**	**4**	**5**	**6**
Accuracy (%)	**93.1**	**93.2**	**92.6**	**83.8**	**81.4**	**83.8**
Agent ID				X	X	X
Age	X	X				X
Product ID		X	X	X	X	X
Area Code					X	X
Gender			X	X	X	X
Call Count			X	X	X	X

Next, for the CityBlock_SFFS_kNN_vk2 algorithm, the components of the confusion matrix are obtained as given below

$$C_{CityBlock} = \begin{pmatrix} TN & FN \\ TP & FP \end{pmatrix} = \begin{pmatrix} 26383 & 269 \\ 3249 & 23289 \end{pmatrix}. \quad (8)$$

Consequently, the performance values are calculated as presented in Table 7.

Table 7. Performance report of the CityBlock_SFFS_kNN_vk2 model

Weighted Mode (0,1)	Precision	Recall	F1-score	Support
0	0.89	0.99	0.94	26652
1	0.99	0.88	0.93	26538
Total				
Accuracy			0.93	53190
Macro Average	0.94	0.93	0.93	53190
Weighted Average	0.94	0.93	0.93	53190

The same evaluation steps are followed respectively also for the Chebyshev, Euclidean, Correlation, Canberra, Dice and the Cosine distance models. In conclusion, the main results are summarized in Table 8.

Table 8. Comparing all types of the kNN architecture

Architecture	Accuracy Value	Performance Rate	Effective Attributes	Process Time (≈)
CityBlock_SFFS_kNN_vk2	0.93	0.9322	Age, Product ID	1 minute
ChebyShev_SFFS_kNN_vk2	0.93	0.9327	Age, Product ID	1 minute
Euclidean_SFFS_kNN_vk2	0.93	0.9336	Age, Product ID	1 minute
Correlation_SFFS_kNN_vk1	0.90	0.9032	Age, Gender, Product ID	47 minutes
Canberra_SFFS_kNN_vk2	0.94	0.9329	Age, Gender	40 minutes
Dice_SFFS_kNN_vk1	0.63	0.5983	Gender, Call Count	75 minutes
Dice_SFFS_kNN_vk5	0.63	0.6225	Product ID, Gender	65 minutes
Cosine_SFFS_kNN_vk2	0.94	0.9322	Product ID, Gender, Call Count	70 minutes

7. Final Remarks

Human expectations and technological developments take vital roles in increasing the significance of data and knowledge nowadays. In this direction, the data gathered from various sources are kept in different data storage environments. Consequently, various approaches are introduced in literature to process the data stacks, which are increasing in size and changing in structure. The ML approach, which is about focusing on mathematical equations representing real-world scenarios, is among these methods. The key tasks of ML are regression, classification, clustering, transcription, machine translation, anomaly detection, synthesis & sampling, estimation of probability density, prediction of probability mass function, similarity matching, so-occurrence grouping, causal modeling and link profiling. On the other hand, the main development phases that have been used to overcome the aforementioned ML tasks are exploratory data analysis, data preprocessing, feature engineering (feature creation/extraction, feature selection, feature dimensionality reduction), architecture selection, training algorithms, testing & matching, approach monitoring and model retraining. Within the scope of this research, the analysis of a telemarketing dataset of company X through a classification-capable ML architecture is intended, and its methodology is illustrated in Figure 14.

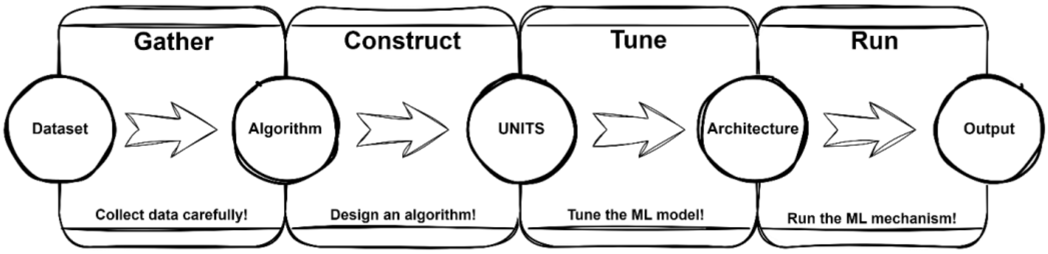

Figure 14. An illustration for our methodology

An ordinary ML architecture consists of various subunits and critical hyperparameters, therefore choosing the most appropriate combination of components is a critical step for the success of the algorithm. Unfortunately, no algorithm can learn the most appropriate subunits and determine the best value of hyperparameters. In the first phase of our research, the dataset has been cleaned, arranged and balanced. These tasks can be done with a series of commands including various packages of the Python. The second important phase in our research was the design of the most appropriate ML architecture. This task generally dominates a significant portion of an ML research in terms of research time spent. Various subunits and different hyperparameter values have been considered in our research, and then some suitable kNN architectures have been constructed to analyze the selected telemarketing case. In the last step of our investigation, we have run these suitable forms of the kNN architecture. A remarkable conclusion has emerged from the results presented in Table 8: the “Age” of customer and “Product ID” are the most important attributes that affect the “Sales” column in Table 1. If a preference is given to an analysis completed quickly, the classification success rate of the architecture relatively decreases. Conversely, if the aim is to achieve the most successful forecasting results, time must be sacrificed.

Figure 15. Best combination of the kNN architecture in our investigation

As a matter of fact, the most appropriate architectural design clearly comes to the fore. According to the highest accuracy rate and performance value, the best kNN algorithm should be as in FIG. 15. The architecture above shows that the most important factors affecting the "Sales" attribute are the "Age" and "Gender" information. So, considering the whole of our research, it would be appropriate to add "Product ID" information next to these two effective attributes.

Ethical statement

The data is sourced from an open-access database, so there is no need for an ethics committee’s evaluation.

Acknowledgment

We are grateful to the anonymous referees for substantial suggestions that improved the quality of our study. Also, we would like to thank Assoc. Professor Mustafa SALTI for their valuable financial support.

Conflict of interest

The authors declare no competing interests.

Authors' Contributions

M. Salti: Software, Formal analysis, Data curation. E.E. Kangal: Supervision, Conceptualization, Methodology, Writing-original draft, Writing - review & editing B. Zengin: Validation, Investigation.

References

[1] Moro, S., Cortez, P., Rita, P., "A data-driven approach to predict the success of bank telemarketing", *Decision Support Systems*, 62, 22-31, 2014.

[2] Kotler, P., Keller, K.L., "*Framework for Marketing Management (6th edition)*", Pearson, London-UK, 2015.

[3] Rust, R.T., Moorman, C., Bhalla, G., "*Rethinking Marketing*", Harvard Business Review, 1, 1, 2010.

[4] The reader can find detailed information about the European Organization for Nuclear Research (CERN) at https://www.home.cern.

[5] R. Krawczyk, Colombo, T., Neufeld, N., Pisani, F., Valat, S., "Ethernet for high-throughput computing at CERN", *IEEE Transactions on Parallel and Distributed Systems*, 33, 3640-3650, 2022.

[6] Buhl, H.U., Roglinger, M., Moser, F., Heidemann, J., "Big Data", *Business & Information Systems Engineering*, 5, 65-69, 2013.

[7] Verleysen, M., Francois, D., "The Curse of Dimensionality in Data Mining and Time Series Prediction", *Computational Intelligence and Bioinspired Systems*, 3512, 758-770, 2005.

[8] Marron, B. A., de Maine, P. A. D., "Communications of the ACM", *Communications of the ACM*, 10, 711-715, 1967.

[9] Heavens, A. F., Jimenez, R., Lahav, O., "Massive lossless data compression and multiple parameter estimation from galaxy spectra", *Monthly Notices of the Royal Astronomical Society*, 317, 965-972, 2000.

[10] Zhaoping, L., "Theoretical understanding of the early visual processes by data compression and data selection", *Network: Computation in Neural Systems*, 17, 301-334, 2006.

[11] Suarjaya, I.M.A.D., "A New Algorithm for Data Compression Optimization", *Int. J. Adv. Comp. Sci. and Appl.*, 3, 14-17, 2012.

[12] Adek, R.T., Ula, M., "A Survey on The Accuracy of Machine Learning Techniques for Intrusion and Anomaly Detection on Public Data Sets", *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, 19-27, 2020.

[13] Thomas, R.N., Gupta, R., "A Survey on Machine Learning Approaches and Its Techniques" *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 1-6, 2020.

[14] Dixon, M.F., Halperin, I., Bilokon, P., "*Machine Learning in Finance from Theory to Practice*", Springer Nature Press, 2020.

[15] Bose, I., Mahapatra, R.K., "Business data mining a machine learning perspective.", *Inf. Management*, 39, 211-225, 2001.

[16] Stirnberg, R., Cermak, J., Kotthaus, S., Haeffelin, M., Andersen, H., Fuchs, J., Kim, M., Petit, J.E., Favez, O.,, "Meteorology-driven variability of air pollution (PM1) revealed with explainable machine learning", *Atmos. Chem. Phys.*, 21, 3919–3948, 2021.

[17] Vogt, N., "Machine learning in neuroscience", *Nature Methods*, 15, 33, 2018.

[18] Bektas, J., Bektas, Y., Kangal, E.E., "Integrating a novel SRCRN network for segmentation with representative batch-mode experiments for detecting melanoma", *Biomedical Signal Processing and Control*, 71, 103218, 2022.

[19] Ramakrishnan, R., von Lilienfeld, O.A., "*Chapter 5 in Reviews in Computational Chemistry*", Wiley-VCH, Weinheim, Germany, 225–256, 2017.

[20] Ibrahim, S.K., Ahmed, A., Zeidan, M.A.E., Ziedan, I.E., "Machine Learning Methods for Spacecraft Telemetry Mining", *IEEE Trans. Aerosp. Electron. Syst.*, 55, 1816-1827, 2019.

[21] Caldeira, J., Wu, W.L.K., Nord, B., Avestruz, C., Trivedi, S., Story, K.T., "DeepCMB: Lensing reconstruction of the cosmic microwave background with deep neural networks.", *Astron. Comput*. 28, 100307, 2019.

[22] Ntampaka, M. Trac, H., Sutherland, D. J., Battaglia, N., Póczos, B., Schneider, J., "A Machine Learning Approach for Dynamical Mass Measurements of Galaxy Clusters.", *Astrophys. J.*, 803, 50, 2015.

[23] Salti, M., Kangal, E.E., Aydogdu, O., "Evolution of CMB temperature in a Chaplygin gas model from deep learning perspective", *Astronomy and Computing*, 37, 100504, 2021.

[24] Salti, M., Kangal, E.E., "Deep learning of CMB radiation temperature", *Annals of Physics* 439, 168799, 2022.

[25] Kangal, E.E., Salti, M., Aydogdu, O., "Machine learning algorithm in a caloric view point of cosmology", *Phys. Dark Univ.*, 26, 100369, 2019.

[26] Escamilla-Rivera, C., Quintero, M.A.C., Capoziello, S., "A deep learning approach to cosmological dark energy models.", *JCAP*, 03, 008, 2020.

[27] Tilaver, H., Salti, M., Aydogdu, O., Kangal, E.E., "Deep learning approach to Hubble parameter.", *Comp. Phys. Commun.*, 261, 107809, 2021.

[28] Donalek, C., "*Supervised and unsupervised learning*", Astronomy Colloquia, California Institute of Technology, USA, 2011.

[29] Benvenuto, F., Piana, M., Campi, C., A. M. Massone, "A Hybrid Supervised/Unsupervised Machine Learning Approach to Solar Flare Prediction", *ApJ*, 853, 90, 2018.

[30] Arnott, D., Pervan, G., "Eight key issues for the decision support systems discipline", Decision Support Systems, 44, 657-672, 2008.

[31] Turban, E., Sharda, R., Delen, D., "*Decision Support and Business Intelligence Systems (9th edition)*", Pearson, London-UK, 2011.

[32] Hastie, T., Tibshirani, R., Friedman, J., "*The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd edition)*", Springer-Verlag, NY, USA, 2008.

[33] Kowsari, K., Meimandi, K.J., Heidarysafa, M., Sanjana, M., Laura, B., Brown, D., "Text Classification Algorithms: A Survey", *Information*, 10, 150, 2019.

[34] Harper, P.R., "A review and comparison of classification algorithms for medical decision making", *Health Policy*, 71, 315-331, 2005.

[35] Kumar, R., Verma, R., "Classification Algorithms for Data Mining: A Survey", *International Journal of Innovations in Engineering and Technology*, 1, 2319, 2012.

[36] Ilham, A., Khikmah, L., Ulumuddin, I., Indra I., "Long-term deposits prediction: a comparative framework of classification model for predict the success of bank telemarketing", *IOP Conf. Series - Journal of Physics: Conf. Series*, 1175, 012035, 2019.

[37] Mustapha, S.M.F.D.S., Alsufyani, A., "Application of Artificial Neural Network and information gain in building case-based reasoning for telemarketing prediction", *International Journal of Advanced Computer Science and Applications*, 10, 300-306, 2019.

[38] Tekouabou, S.C.K., Cherif, W., Silkan, H., "*A data modeling approach for classification problems: application to bank telemarketing prediction*", Proceedings of the 2nd International Conference on Networking, Information Systems & Security (NISS19), Rabat-Morocco, 2019.

[39] Keles, A., Keles, A., "IBMMS Decision Support Tool for Management of Bank Telemarketing Campaigns", *International Journal of Database Management Systems*, 17, 1, 2015.

[40] Kocoglu, F.O., Esnaf, S., "Machine Learning Approach and Model Performance Evaluation for Tele-Marketing Success Classification", *International Journal of Business Analytics*, 9, 1-18, 2022.

[41] Halim, K.N.A., Jaya, A.S.M., Fadzil, A.F.A., "Data Pre-Processing Algorithm for Neural Network Binary Classification Model in Bank Tele-Marketing", *International Journal of Innovative Technology and Exploring Engineering*, 9, 272-277, 2020.

[42] Shashidhara, B. M. Jain, S., Rao, V. D., Patil, N., Raghavendra, G.S., "*Evaluation of machine learning frameworks on bank marketing and Higgs datasets*", Proceedings of Second International Conference on Advances in Computing and Communication Engineering, Dehradun-India, 2015.

[43] Wiemer, H., Drowatzky, L., Ihlenfeldt, S., "Data Mining Methodology for Engineering Applications (DMME)-A Holistic Extension to the CRISP-DM Model", *Appl. Sci.*, 9, 2407, 2019.

[44] Jaggia, S., Kelly, A., Lertwachara, K., Chen, L., "Applying the CRISP-DM framework for teaching business analytics", *Decision Sciences Journal of Innovative Education*, 18, 612-634, 2020.

[45] Mohamed, A., "Data Analysis for Telemarketing Case", www.kaggle.com. Date of access: 04 Jan 2022.

[46] Alfeilat, H.A.A., Hassanat, A.B.A., Lasassmeh, O., Tarawneh, A.S., Alhasanat, M.B., Salman, H.S.E., Prasath, V.B.S., "Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review", *Big Data*, 7, 221-248, 2019.

[47] Deza, E., Deza, M.M., "*Encyclopedia of distances*", Springer, 2009.

[48] Hall, M.A., "Correlation-based Feature Selection for Machine Learning", PhD Thesis, The University of Waikato, Hamilton, NewZealand, 1999.

[49] Cunningham, S. J., Littin, J., Witten. I. H., "Applications of machine learning in information retrieval", Technical Report 97/6, University of Waikato, 1997.

[50] Prasatha, V. B.S., Alfeilate, H.A.A., Hassanate, A.B.A., Lasassmehe, O., Tarawnehf, A.S., Alhasanatg, M.B., Salmane, H.S.E, "Effects of Distance Measure Choice on KNN Classifier Performance - A Review", *e-Print:* 1708.04321v3, 2019.

[51] Cha, S.H., "Probabilistic, Statistical and Algorithmic Aspects of the Similarity of Texts and Application to Gospels Comparison", *International Journal of Mathematical Models and Methods in Applied Sciences*, 1, 300-307, 2007.

[52] Jain, A., Zongker, D., "Feature selection: evaluation, application, and small sample performance", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 153-158, 1997.

[53] Pavlenko, T., "On feature selection, curse-of-dimensionality and error probability in discriminant analysis", *Journal of Statistical Planning and Inference*, 115, 565-584, 2003.

[54] Myakalwar, A.K., Spegazzini, N., Zhang, C., Anubham, S.K., Dasari, R.R., Barman, I., Gundawar, M.K., "Less is more: Avoiding the LIBS dimensionality curse through judicious feature selection for explosive detection", *Scientific Reports*, 5, 13169, 2015.

[55] Elkhani, N., Muniyandi, R.C., "Membrane computing inspired feature selection model for microarray cancer data", *Intell. Data Anal.*, 21, 137-157, 2017.

[56] Last, M., Kandel, A., Maimon, O., "Information-theoretic algorithm for feature selection", Pattern Recognition Letters, 22, 799-811, 2001.

[57] Muni, D.P., Pal, N.R., Das, J., "Genetic programming for simultaneous feature selection and classifier design", *IEEE Trans. Syst. Man Cybern. Part B*, 36, 106-117, 2006.

[58] Strearns, S.D., "*On Selecting Features for Pattern Classifiers*", 3rd International Conference on Pattern Recognition", Colorado-CA, 1976.

[59] Pudil, P., Ferri, F.J., Novovicova, J., Kittler, J., "*Floating search methods for feature selection with nonmonotonic criterion functions*", Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. II-Conference B: Pattern Recognition and Neural Networks, Jerusalem-Israel, 1994.

[60] Pudil, P., Novovicova, J., Kittler, J., "Floating search methods in feature selection", *Pattern. Recogn. Lett.*, 15, 1119-1125, 1994.

[61] Caruana, R., Niculescu-Mizil, A., "*Data mining in metric space: an empirical analysis of supervised learning performance criteria*", Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle-WA, USA, 2004.

[62] Li, H., Li, H., Wei, K., "Automatic fast double KNN classification algorithm based on ACC and hierarchical clustering for big data", *Int. J. Commun. Syst.*, 31, e3488, 2018.

[63] Madray, I., Suire, J., Desforges, J., Madani, M.R., "Relative angle correction for distance estimation using K-nearest neighbors", *IEEE Sensors Journal*, 20, 8155, 2020.

[64] Snchez-Crisostomo, J., Alejo, R., López-González, E., Valdovinos, R.M., Pacheco-Sánchez, J.H., "*Empirical analysis of assessments metrics for multi-class imbalance learning on the back-propagation context*" in "*Advances in Swerm Intelligence*", Lecture Notes in Computer Science, Springer, 8795, 17-23, 2014.

[65] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, 16, 321-357, 2002.

[66] Torgo, L., Ribeiro, R. P., Pfahringer, B., Branco, P., "*SMOTE for Regression*", Progress in Artificial Intelligence, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 8154, 2013.

[67] Fernández, A., Garcia, S., Herrera, F., Chawla, N.V., "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary", *Journal of Artificial Intelligence Research*, 61 863-905, 2018.

[68] Pavithra, P., Babu, S., *International Journal of Scientific Research and Engineering Development*, 2, 86-90, 2019.

[69] Jager, M., "*Improving data imbalance using Synthetic Minority Over-sampling (SMOTE)*", www.medium.com, Date of access: 04.06.2022.

[70] Probst, P., Bischl, B., Boulesteix, A.L., "Tunability: Importance of Hyperparameters of Machine Learning Algorithms", *e-Print:* 1802.09596, 2018.

[71] Wang, B, Gong, N. Z., "*Stealing Hyperparameters in Machine Learning*", 2018 IEEE Symposium on Security and Privacy (SP), San Francisco-CA, USA, 2018.

[72] Yang, L., Shami, A., "On hyperparameter optimization of machine learning algorithms: Theory and practice", *Neurocomputing*, 415, 295-316, 2020.