Pancreatic Cancer Classification Using Missing Data Imputation And Cluster-Based Undersampling Methods: A Comparative Analysis With Multiple Machine Learning Algorithms

Wanessa L. B. Sena
Graduate Academic Department
Federal Institute of Pernambuco - IFPE
Recife, Brazil
wlbs@a.recife.ifpe.edu.br

Renata F. P. Neves
Graduate Academic Department
Federal Institute of Pernambuco - IFPE
Recife, Brazil
renatafreire@recife.ifpe.edu.br

***Abstract*—Missing values and class imbalance are issues frequently found in databases from real-world scenarios, including cancer classification. Impacts on the performance of Machine Learning (ML) models can be observed if these issues are not properly addressed prior to the analysis. In this paper, a combined solution with missing data imputation using *k*NN and cluster-based undersampling using *k-means* is proposed, focusing on pancreatic cancer classification. Different data subsets were generated by combining different preprocessing methods and the performance was analyzed using a ML analysis pipeline from a previous study. This pipeline implements ten ML classifiers, including Random Forest (RF), Support Vector Machine (SVM) and Artificial Neural Network (ANN). All data subsets presented a significant improvement ($p<0.05$ with Student's T-Test) in the performance of most ML algorithms when compared with the results obtained when the pipeline was first evaluated. Results suggest that *k*NN and *k-means* can be used in the data preprocessing phase to overcome missing values and class imbalance issues and improve the classification accuracy.**

***Index Terms*—Machine Learning, k-means clustering, kNN, undersampling, missing data imputation, classification**

I. Introduction

Pancreatic cancer is the seventh leading cause of cancer death in both genders, having similar incidence and mortality rates (495,773 new cases and 466,003 deaths registered in 2020, according to [1]). Projections indicate that pancreatic cancer will become the third leading cause of cancer death by 2025, surpassing breast cancer [1]. Apart from pancreatic cancer inherent characteristics that confer an aggressive behavior and a highly metastatic potential to these tumors, diagnosis continues to be a challenge due to the absence of sensitive methods for early detection [2].

Artificial intelligence (AI) applications in healthcare have grown significantly in the past few years, mainly driven by the progress of analytical methods and the increased availability of data associated with medical care. AI methods, including Machine Learning (ML), can provide relevant information from patients' data and help support better clinical decisions [3].

However, there are some challenges associated with the use of cancer-related real-world data in ML models. Since the number of healthy individuals is usually much higher than the number of cancer patients, this discrepancy generates an issue known in ML as class imbalance. In class imbalance, one class is represented by a large number of samples, whereas the other class (usually the class of interest) is represented by only a few samples [4]. This disparity between the two classes leads to bias toward the majority class, causing impacts on the classification performance of the minority class [5]. Also, data quality is a major concern while working with cancer registries. Since the data generally relies on patient medical records, unknown or missing values occurrences are frequent [6]. Therefore, overcoming some of these challenges becomes really important to build reliable ML models that represent the diversity and complexity of real-world data.

In the present study we aimed to evaluate different methods to solve class imbalance and missing values problems in a pancreatic cancer dataset and its impacts on the classification performance of different ML algorithms. The remainder of this paper is organized as follows: Section II presents an overview of class imbalance and missing data values issues, in addition to a summary of some ML algorithms; Section III presents the research methodology; Section IV presents the results, and Section V describes the conclusions based on our findings.

II. Theoretical Background

A. The class imbalance problem

Class imbalance is a common challenge to many real-world application areas, including healthcare [7] [8]. A practical class imbalance case is cancer classification, since the number of non-cancer cases (the majority class) is usually much higher than the number of cancer cases (the minority class). The discrepancy becomes especially evident when dealing with less frequent cancer types, such as pancreatic cancer (incident rate of 2.6%) [1]. In machine learning, the deficit in the number of the minority class can limit the model and lead to misclassification [9].

Different strategies have been applied to solve the class imbalance problem. In particular, data-level approaches, which consist of applying data preprocessing methods in order to reduce the imbalance ratio, are among the most common strategies [4]. This process can be done by either decreasing the number of majority class instances (undersampling) or increasing the number of minority class instances (oversampling) [5]. Undersampling has been reported in the literature as a better choice than oversampling since oversampling may increase the possibility of overfitting. However, depending on the method used, undersampling can also lead to underfitting [8].

In order to overcome the limitations of undersampling, cluster-based methods have been proposed to make sure useful data is not removed from the majority class [4] [5] [8]. Clustering algorithms explore the structure of the data distribution and define grouping rules for data with similar characteristics [10]. Regarding those algorithms, the *k-means* algorithm is one of the most widely used for data analysis. *K-means* clustering is an unsupervised learning method that, based on a given k, executes the following steps: 1) divide the data points into k clusters; 2) calculate the centroid of each cluster; and 3) reassign the data point based on the closest centroid [11]. The *k-means* algorithm application has been reported in different application domains [12] [13]. In the undersampling context, studies have shown the *k-means* algorithm usage as a single strategy or combined with other clustering-based techniques [4] [8].

B. The missing data values problem

Datasets with missing values are frequent and can have significant impacts on data analysis. Missing values can happen due to a series of reasons, e.g. unanswered questions in a questionnaire, data loss for unpredictable factors or high costs associated with data obtention [14]. The problems caused by missing values during the ML analysis include: increase in process time; complications while handling and analyzing the data; and bias introduction [15]. Properly addressing this issue becomes crucial when dealing with missing values in the minority class, since it is important to assure the data is representing the diversity observed in real-world use [7].

Different approaches for dealing with missing values can be applied, which includes machine learning methods, such as kNN. kNN (k-nearest neighbors) is a non-parametric method widely used for classification and regression analysis. In this method, k closest neighbors are identified based on the training data and used as a further reference for test data prediction [16].

C. ML classifiers

This subsection provides a brief description of the ML algorithms that will be explored in the subsequent sections of the present study.

Logistic Regression (LR) - LR is a statistical model widely used for binary classification problems. Based on a given set of inputs, this method uses a logistic (Sigmoid) function to model the output, returning values between 0 and 1 [17].

Decision Tree (DT) - DT is a non-parametric supervised learning algorithm applied in different areas, such as machine learning, image processing, and pattern identification [18]. A DT presents a hierarchical structure composed of nodes and branches. In general terms, the process starts with a root node, and then a divide-and-conquer strategy is conducted to identify the optimal split points and generate decision nodes. The process is recursively repeated until leaf nodes are generated, representing all the possible outcomes within the dataset [19].

Random Forest (RF) - RF is a supervised ML algorithm broadly used in classification and regression problems. Similarly to the DT method, RF is also a tree-based algorithm, which recursively splits the given dataset into two groups until a determined stopping condition is fulfilled. However, instead of generating a single tree, the algorithm averages predictions based on many individual trees [20].

Naive Bayes (NB) - NB is a probabilistic classification algorithm based on the Bayes' theorem. This method has been applied in many real-world situations and it is widely used especially for its simplicity, accuracy and good performance when compared to other methods [21].

Extreme Gradient Boosting (XGB) - XGB (also known as XGBoost) is a ML algorithm that uses gradient-boosting decision trees to make predictions. This model was first proposed by [22] and can be applied to both classification and regression problems. XGB takes advantage of the multithreading of the CPU for parallel computing, which speeds up its execution [23].

Light Gradient Boosting Machine (LGB) - LGB (also known as LGBoost or LightGBM) is a gradient-boosting decision tree algorithm, similarly to XGB. LGB is a high-precision and high-performance method used in ranking and classification problems [24].

Support Vector Machine (SVM) - SVM is a popular supervised learning algorithm that is used for classification, regression and outliers detection. The main goal of SVM is to find the choice limit (or hyperplane) in a n-dimensional space that can separate the data points in different classes [25].

Artificial Neural Network (ANN) - ANN is a ML computational model inspired by the human brain. This method is composed of several single processing elements, called neurons [25]. ANN has been applied in large scale problems, including studies related to cancer clinical practice [26] [27].

III. Methodology

The goal of the present study is to understand how we can overcome class imbalance and missing values issues in large datasets in order to select meaningful samples and increase the classification accuracy. In order to evaluate this, the methods already established by [28] were used, but the data preprocessing step was changed. Keeping the exact same methods for the other parts of the process allowed us to compare the performance of the ML models in both scenarios.

A. Dataset

PCLO (Prostate, Lung, Colorectal and Ovarian) Cancer Screening Trial was a randomized and controlled study performed by the National Cancer Institute in the United States between November 1993 and July 2001. The main goal of this study was to evaluate the efficiency of certain procedures for early detection for prostate, long, colorectal and ovarian cancers and included approximately 153,000 men and women. Self-administered questionnaires were used to collect data on healthy history, demographics and other lifestyle aspects of the patients [29].

Based on the data available, two populations were considered in the present study: the first population included people that had the pancreatic cancer diagnosis confirmed during the PCLO trial (n=807), while the second population represented healthy controls (n=100,819).

B. Feature selection

In reference to the previous study performed by [28], 19 features related to patients' health history and habits were selected to evaluate the dataset, as following: **panc_cancer** (primary case of pancreatic cancer confirmed during the PCLO trial), **cig_stat** (current cigarette smoking status), **cig_stop** (number of years since stopped smoking), **cig_years** (total number of years the participant smoked), **pack_years** (number of packs smoked during the total smoking period), **bmi_curr** (current value of body mass index), **bmi_curc** (current value of body mass index based on World Health Organization standard categorization), **diabetes_f** (ever had diabetes), **panc_fh** (pancreatic cancer family history in first-degree relatives), **fh_cancer** (cancer family history in first-degree relatives), **bmi_20** (body mass index at age 20), **bmi_50** (body mass index at age 50), **asp** (regular use of aspirin in the past 12 months), **ibup** (regular use of ibuprofen in the past 12 months), **gallblad_f** (ever had gallbladder stones or inflammation), **age** (age at trial entry), **race7** (race/ethnic background), **marital** (marital status) and **sex** (sex of the participant).

C. Data preprocessing

The algorithm used for data preprocessing was implemented using Python (version 3.10) and included the following libraries: NumPy (version 1.21.5), pandas (version 1.4.4) e scikit-learn (version 1.0.2).

Due to class imbalance present in the PCLO dataset, undersampling methods were used to rebalance the class distribution and match the healthy control population size used by [28]. The first step of the process was to remove most of the records with missing values, reducing the population by 5% (from 100,819 to 95,719). Samples related to non-smokers patients with the attribute **cig_stop** empty were the exception, since the number of years since the patient stopped smoking is not applicable in this case; these samples were kept and the attribute filled with the patient's age. After that, two methods were used to select healthy controls: 1) random selection; and 2) *k-means* clustering. In the present study, multiple clusters were generated using *k-means* (*k*=4,298) and the sample closest to the centroid was selected to represent each group.

A different approach was used when preprocessing pancreatic cancer population data. All 807 records were maintained and the *k*NN (*k*-nearest neighbors) algorithm was used to input missing values, based on the mean value from *k* nearest neighbours in the training set. In this study different data subsets were generated by changing the number of neighbors (*k*=3 and *k*=10) used to input missing values, aiming to validate the impacts of this variation on the final analysis.

After the combination of different techniques described above, 4 subsets of data were generated and used for subsequent analysis. Details are present in Table I.

TABLE I
DATA SUBSETS GENERATED BASED ON DIFFERENT PREPROCESSING METHODS

Data subset	Healthy control pop.		Pancreatic cancer pop.	
	No. of samples	**Undersampling method**	**No. of samples**	**Missing values input method**
S1	4,600	Random selection	807	kNN *k*=3
S2	4,600	k-means	807	kNN *k*=3
S3	4,600	Random selection	807	kNN *k*=10
S4	4,600	k-means	807	kNN *k*=10

D. Data analysis

In order to evaluate the impacts of the techniques described in subsection C in the final results, subsequent analyses were performed using the ML analysis pipeline created by [28]. This pipeline, which uses different Python libraries (e.g. scikit-rebate, xgboost) and runs as a Jupyter Notebook application, is publicly available at [30] and can be executed in a new dataset with small modifications.

The pipeline is composed of 4 main stages:

- Preprocessing and Feature Transformation;
- Feature Importance and Selection;
- ML Modeling;
- and Post-Analysis.

During the Preprocessing and Feature Transformation stage, an exploratory analysis is performed to assess some data aspects, including data dimensions, feature types, class imbalance and feature correlations. After that, a basic data cleaning is executed, followed by a *K*-fold Cross Validation (CV) partitioning. In *K*-fold CV, a training sample is divided into *K* smaller subsets and, while (*K* - 1) subsets are used to build the model, the remaining subset is used for validation. The process is repeated *K* times until all subsets are used for validation [31]. In this study, *K*-fold CV was used with *K* = 10.

The next step is the Feature Importance and Selection, which evaluates the feature importance prior to the ML algorithm execution and removes irrelevant features for each *k* training set when required. The last part is only important when analyzing datasets with a large number (>50) of features [28], so it is not relevant to the present study.

Once the previous steps are completed the next stage is the ML Modeling, the core of the pipeline. This ML modeling is composed of 9 different algorithms: Logistic Regression, Decision Tree, Random Forest, Naive Bayes, XGBoost, LGBoost, Support Vector Machine, Artificial Neural Network and ExSTraCS (version 2.0.2.1). ExSTraCS is a Learning Classifier System (LCS) developed to solve complex classification and prediction challenges by combining a series of heuristics [32]. This pipeline includes the analysis of ExSTraCS performance before and after applying Quick Rule Filtering (QRF), a rule-compaction procedure [33].

After each ML algorithm is executed, average CV performance is evaluated by calculating some classification metrics that includes balanced accuracy, F1-Score, precision, recall, and Receiver Operating Characteristic (ROC) area under the curve (AUC). Balanced accuracy is the average of recall and specificity, calculated based on the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), as shown in Eq. (1) and Eq. (2), respectively [32]. F1-Score is a measure highly used in different ML areas in both binary classification and multiclass cenarios, and represents the average between precision, represented in Eq. (3), and recall [34].

$$\text{recall} = \frac{\sum TP}{\sum TP + \sum FN} \tag{1}$$

$$\text{specificity} = \frac{\sum TN}{\sum TN + \sum FP} \tag{2}$$

$$\text{precision} = \frac{\sum TP}{\sum TP + \sum FP} \tag{3}$$

The Post-Analysis, the last stage of the pipeline, summarizes the metrics obtained for each algorithm and generates a series of files and graphs to facilitate the performance comparative analysis for all algorithms.

IV. Results

Tables II and III summarize average balanced accuracy and F1-Score, respectively, for each dataset and ML algorithm used in the present study, also making a comparison with the averages obtained by [28]. Averages below the results from [28] are highlighted in red, while averages above the results from [28] are highlighted in green (p<0.05 with Student's T-Test).

Results present in tables II and III show that, for all subsets, most ML algorithms had their performance significantly improved (p<0.05 with Student's T-Test) when compared with the results from [28]. Within each data subset, it was not possible to determine the top performer algorithm because most of them presented similar averages. Subsets S2 and S4, generated using *k-means* as the undersampling method,

TABLE II
BALANCED ACCURACY (WITH STANDARD DEVIATION) AVERAGE OVER 10-FOLD CROSS-VALIDATION FOR EACH DATASET AND ML ALGORITHM COMPARED WITH THE RESULTS OBTAINED BY [28]

ML Algorithm	**[28]**	**S1**	**S2**	**S3**	**S4**
LR	0.6795 (0.0328)	0.7762 (0.0191)	0.7352 (0.0276)	0.7712 (0.0263)	0.7359 (0.0259)
DT	0.6745 (0.0262)	0.7728 (0.0193)	0.7248 (0.0254)	0.7634 (0.0198)	0.7262 (0.0327)
RF	0.6798 (0.03)	0.7806 (0.016)	0.741 (0.0293)	0.7748 (0.0214)	0.7454 (0.0346)
NB	0.6053 (0.024)	0.5719 (0.0217)	0.6462 (0.0328)	0.5784 (0.0283)	0.6559 (0.0339)
XGB	0.6854 (0.0276)	0.7838 (0.017)	0.756 (0.0376)	0.7722 (0.0274)	0.7459 (0.0381)
LGB	0.6851 (0.0295)	0.7821 (0.0225)	0.7522 (0.0362)	0.7728 (0.0211)	0.7522 (0.0378)
SVM	0.6761 (0.0218)	0.7758 (0.0202)	0.7364 (0.0266)	0.7737 (0.0221)	0.7364 (0.0421)
ANN	0.5824 (0.0301)	0.7218 (0.0289)	0.7266 (0.0334)	0.7271 (0.0176)	0.7298 (0.0306)
LCS	0.6668 (0.0191)	0.7666 (0.0285)	0.7204 (0.033)	0.7644 (0.0258)	0.7201 (0.0329)
LCS with QRF	0.5579 (0.0164)	0.7067 (0.0326)	0.7126 (0.0327)	0.7139 (0.0301)	0.71 (0.0257)

TABLE III
F1-SCORE (WITH STANDARD DEVIATION) AVERAGE OVER 10-FOLD CROSS-VALIDATION FOR EACH DATASET AND ML ALGORITHM COMPARED WITH THE RESULTS OBTAINED BY [28]

ML Algorithm	**[28]**	**S1**	**S2**	**S3**	**S4**
LR	0.4221 (0.042)	0.523 (0.0233)	0.4894 (0.0348)	0.5239 (0.0336)	0.4958 (0.0286)
DT	0.4183 (0.0352)	0.4915 (0.0236)	0.5173 (0.0773)	0.4895 (0.0319)	0.5344 (0.0728)
RF	0.4272 (0.04)	0.5131 (0.0206)	0.5558 (0.0402)	0.5158 (0.0344)	0.5687 (0.047)
NB	0.3383 (0.0474)	0.2761 (0.039)	0.4241 (0.0655)	0.2874 (0.0518)	0.443 (0.0655)
XGB	0.4317 (0.0363)	0.5054 (0.0184)	0.5555 (0.0477)	0.5039 (0.044)	0.546 (0.0465)
LGB	0.4311 (0.0376)	0.5023 (0.0239)	0.5359 (0.05)	0.5067 (0.0292)	0.5469 (0.0582)
SVM	0.4231 (0.0298)	0.526 (0.0223)	0.5404 (0.0525)	0.5302 (0.0274)	0.5333 (0.0564)
ANN	0.2908 (0.0649)	0.5717 (0.0625)	0.5986 (0.0671)	0.5913 (0.0346)	0.6048 (0.0554)
LCS	0.4215 (0.027)	0.4869 (0.0339)	0.4649 (0.0263)	0.4879 (0.0257)	0.4822 (0.0513)
LCS with QRF	0.2188 (0.0486)	0.5817 (0.065)	0.594 (0.0654)	0.5953 (0.0582)	0.59 (0.0506)

performed consistently well when compared with [28] ($p<0.05$ with Student's T-Test) for all ML algorithms. However, increasing *k*NN *k* value (*k*=3 for S2 and *k*=10 for S4) did not have a significant impact on the results.

These findings are consistent with other studies that have shown that imputation methods, including *k*NN, are reliable methods for missing values estimation and can help improve the classification performance [14] [35] [36] . A study realized by [36] randomly inserted missing values in an existing dataset to evaluate *k*NN imputation accuracy and obtained a 89.5% accuracy rate using this method. Reference [14] compared different imputation techniques in a breast cancer dataset and *k*NN presented the highest accuracy averages to 4 out of 7 classifiers analyzed when compared to other imputation methods. Similarly, *k-means* algorithm usage for undersampling, alone or combined with other cluster-based techniques, has also been reported in the literature as an alternative to increase sample diversity and reduce underfitting [33] [16].

In general, subsets S1 and S3 also presented better results when compared with [28], using random selection as the undersampling method. The exception is the Naive Bayes method, which presented worse performances for both metrics when compared with [28] ($p<0.05$ with Student's T-Test). Reference [28] solved the class imbalance problem in the PCLO dataset by selecting only healthy controls with available genotyping data (n=4,298). This selection criteria led to sampling bias (e.g. 85% males, high number of smokers), which may have impacted the performance of the ML algorithms present in the pipeline. Although random selection was used in the present study, sampling bias was not observed in the populations generated using this method; also, the combination with kNN for missing values imputation may justify the increased performance overall. Random sampling has a high level of uncertainty, since it can generate clean instances and result in a highly performant model, or can potentially lead to a loss of important information, disrupting the training process and model performance [37].

Figure 1 brings a new perspective to the results by showing a comparative analysis of the ML algorithms for all data subsets based on the precision/recall curves. Similarly to what was observed with the other metrics, all subsets presented positive results for most ML algorithms in the pipeline, with similar averages between the highest performing algorithms. The exception is also the Naive Bayes method, which presented the most inferior performance when compared to the other ML algorithms ($p<0.05$ with pairwise Mann-Whitney U-test). Since Naive Bayes model assumes that features are independent [21], correlated features used for classification in this study (e.g. cigarette smoking status, number of years smoked) could have caused a negative impact on the performance.

V. Conclusions

In this study we analyzed the impacts of missing data imputation and cluster-based undersampling methods in the performance of different ML algorithms for pancreatic cancer classification. *k-means* and random selection methods were

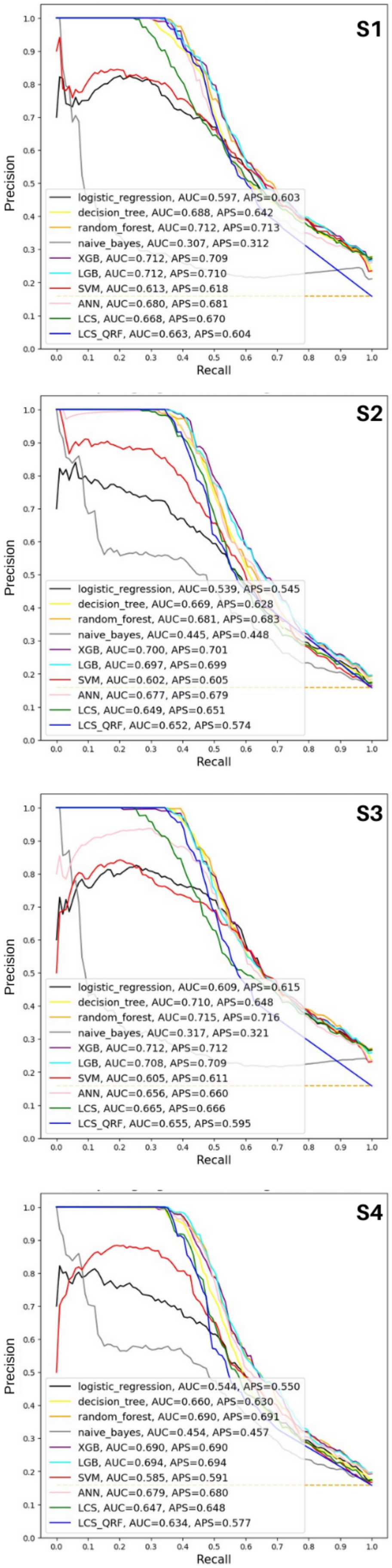

Fig. 1. Precision/recall curves showing ML algorithms performance in the data subsets S1 to S4, generated with different combinations of undersampling and missing values imputation methods. Includes precision/recall AUC and average precision score (APS).

used for healthy control population undersampling, while *k*NN was used to input missing values in the pancreatic cancer population. Performance was analyzed using the ML algorithms pipeline created by [28]. The results presented show that all 4 subsets, generated using different preprocessing methods, had significant performance improvements for most ML algorithms when compared with results from [28]. Our findings suggest that the methods explored in the present study can be a good alternative to improve the classification accuracy.

References

[1] H. Sung et al., "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries", CA Cancer J. Clin., vol. 71(3), pp. 209-249, February 2021.

[2] K. Winter et al., "Diagnostic and therapeutic recommendations in pancreatic ductal adenocarcinoma. Recommendations of the Working Group of the Polish Pancreatic Club", Przeglad gastroenterologiczny, vol. 14(1), pp. 1-18, March 2019.

[3] F. Jiang et al., "Artificial intelligence in healthcare: Past, present and future", Stroke and Vascular Neurology, v. 2(4), pp. 230–243, June 2017.

[4] L. Wei-Chao, T. Chih-Fong, H. Ya-Han, and J. Jing-Shang, "Clustering-based undersampling in class-imbalanced data", Information Sciences, vol. 409-410, pp. 17-26, October 2017.

[5] A. Guzmán-Ponce, R. M. Valdovinos, J. S. Sánchez, and J. R. Marcial-Romero. "A New Under-Sampling Method to Face Class Overlap and Imbalance", Applied Sciences, vol. 10(15), pp. 5164, July 2020.

[6] D. X. Yang et al., "Prevalence of Missing Data in the National Cancer Database and Association With Overall Survival", JAMA Netw Open., vol. 4(3):e211793, March 2021.

[7] PH. C. Chen, Y. Liu., and L. Peng, "How to develop machine learning models for healthcare", Nat. Mater., vol. 18, pp. 410–414, April 2019.

[8] J. Zhang, L. Chen, and F. Abid. "Prediction of Breast Cancer from Imbalance Respect Using Cluster-Based Undersampling Method", vol. 2019, October 2019.

[9] P. Vuttipittayamongkol, E. Elyan, and A. Petrovski, "On the class overlap problem in imbalanced data classification", Knowledge-Based Systems, vol. 212, January 2021.

[10] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation", Electronics, vol. 9(8):1295, August 2020.

[11] N. S. Hassan, A. M. Abdulazeez, D. Q. Zeebaree, and D. A. Hasan, "Medical Images Breast Cancer Segmentation Based on K-Means Clustering Algorithm: A Review", Asian Journal of Research in Computer Science, vol. 9(1), pp. 23–38, May 2021.

[12] J. Qin, W. Fu, H. Gao, and W. X. Zheng, "Distributed k-Means Algorithm and Fuzzy c-Means Algorithm for Sensor Networks Based on Multiagent Consensus Theory," IEEE Transactions on Cybernetics, vol. 47, no. 3, pp. 772-783, March 2017.

[13] L. Bai, J. Liang, and Y. Guo, "An Ensemble Clusterer of Multiple Fuzzy k-Means Clusterings to Recognize Arbitrarily Shaped Clusters," IEEE Transactions on Fuzzy Systems, vol. 26, no. 6, pp. 3524-3533, December 2018.

[14] X. Wu, H. Akbarzadeh Khorshidi, U. Aickelin, Z. Edib, and M. Peate, "Imputation techniques on missing values in breast cancer treatment and fertility data", Health Inf Sci Syst, v. 7, pp. 19, October 2019.

[15] J. Kaiser, "Dealing with Missing Values in Data", Journal of Systems Integration, vol. 5, pp. 42-51, November 2014.

[16] A. Pandey, and A. Jain, "Comparative Analysis of KNN Algorithm using Various Normalization Techniques", International Journal of Computer Network and Information Security, vol. 9, pp. 36-42, November 2017.

[17] A. Subasi, "Chapter 3 - Machine learning techniques", Practical Machine Learning for Data Analysis Using Python, Academic Press, pp. 91-202, 2020.

[18] B. Charbuty, and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning", Journal of Applied Science and Technology Trends, vol. 2(01), pp. 20-28, March 2021.

[19] X. Wei, "A Method of Enterprise Financial Risk Analysis and Early Warning Based on Decision Tree Model", Security and Communication Networks, vol. 2021, September 2021.

[20] M. Schonlau, and R. Y. Zou, "The random forest algorithm for statistical learning", The Stata Journal, vol. 20(1), pp. 3-29, March 2020.

[21] I. Wickramasinghe, and H. Kalutarage, "Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation", Soft Comput., vol. 25, pp. 2277–2293, September 2020.

[22] T. Chen, and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, pp. 785–794, August 2016.

[23] L. Wei, Y. Yanbin, Q. Xiongwen, and Z. Han, "Gene Expression Value Prediction Based on XGBoost Algorithm", Frontiers in Genetics, vol. 10, November 2019.

[24] K. M. Ghori, R. A. Abbasi, M. Awais, M. Imran, A. Ullah, and L. Szathmary, "Performance Analysis of Different Types of Machine Learning Classifiers for Non-Technical Loss Detection," IEEE Access, vol. 8, pp. 16033-16048, January 2020.

[25] A. Kurani, P. Doshi, A. Vakharia, and M. Shah, "A Comprehensive Comparative Study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on Stock Forecasting", Ann. Data. Sci., vol. 10, pp. 183–208, February 2023.

[26] J. Dheeba, N. Albert Singh, and S. Tamil Selvi, "Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach", Journal of Biomedical Informatics, v. 49, pp. 45–52, June 2014.

[27] A. Esteva et al., "Dermatologist-level classification of skin cancer with deep neural networks", Nature, v. 542, pp. 115-118, January 2017.

[28] R. Urbanowicz et al., "A rigorous machine learning analysis pipeline for biomedical binary classification: application in pancreatic cancer nested case-control studies with implications for bias assessments," ArXiv, vol. abs/2008.12829v2, September 2020.

[29] P. C. Prorok et al., "Design of the prostate, lung, colorectal and ovarian (PCLO) cancer screening trial," Control Clin. Trials, vol. 21, pp. 273S-309S, December 2000.

[30] "ExSTraCS ML Pipeline Binary Notebook", R. Urbanowicz, September 2020. [Online] Available: https://github.com/UrbsLab/ExSTraCS_ML_Pipeline_Binary_Notebook

[31] Y. Jung, "Multiple predicting K-fold cross-validation for model selection", Journal of Nonparametric Statistics, vol. 30, pp. 197-215, November 2017.

[32] R. J. Urbanowicz, and J. H. Moore. "ExSTraCS 2.0: Description and Evaluation of a Scalable Learning Classifier System", Evol. Intell., vol. 8(2), pp. 89-116, September 2015.

[33] J. Tan, J. Moore, and R. Urbanowicz, "Rapid Rule Compaction Strategies for Global Knowledge Discovery in a Supervised Learning Classifier System", ECAL 2013: The Twelfth European Conference on Artificial Life, pp. 110-117, September 2013.

[34] D. Chicco, and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation", BMC Genomics, v. 21(1):6, January 2020.

[35] W.Ksiazek, M. Hammad, P. Plawiak, U. Rajendra Acharya, and R. Tadeusiewicz, "Development of novel ensemble model using stacking learning and evolutionary computation techniques for automated hepatocellular carcinoma detection", Biocybernetics and Biomedical Engineering, vol. 40(4), pp. 1512-1524, October-December 2020.

[36] A. E. Karrar, "The Effect of Using Data Pre-Processing by Imputations in Handling Missing Values", Indonesian Journal of Electrical Engineering and Informatics, vol. 10(2), June 2022.

[37] T. Hasanin, T. M. Khoshgoftaar, J. Leevy, and N. Seliya, "Investigating Random Undersampling and Feature Selection on Bioinformatics Big Data," 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), pp. 346-356, April 2019.