Medical Appointment No-Show Prediction Using Machine Learning Techniques

Eslam Abushaaban
Department of Computer Engineering
Marmara University
Istanbul, Turkey
eslam.abushaaban@gmail.com

Mustafa Agaoglu
Department of Computer Engineering
Marmara University
Istanbul, Turkey
agaoglu@marmara.edu.tr

2022 2nd International Conference on Computing and Machine Intelligence (ICMI) | 978-1-6654-7483-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMI55296.2022.9873652

***Abstract*—Health care resources are limited and the efficient utilization of these resources is a must. No show for medical appointments is an everlasting problem that faces health care systems and is a huge waste of resources that can be a hindrance against improving health care services all over the world. This research aims to analyze the factors affecting this problem and create effective predictive models that can help solve it and reduce its social and economical ramifications on health care systems. This study suggests a methodology based on machine learning where different algorithms were utilized and compared and proposes a framework that can be utilized to achieve the best, and most robust classifier in terms of different performance metrics. The objective is to attempt to predict if a patient will attend his/her medical appointment or not.**

***Index Terms*—classification, data imbalance, health care, machine learning**

I. Introduction

Medical resources are increasing in cruciality with the outbreak of the novel coronavirus [1]. The global pandemic situation is challenging the capacity of health care systems emphasizing the need for the effective allocation of these resources. Patient no-show refers to patients not attending their scheduled medical appointments in hospitals or clinics without previous notice or canceling [2]. This causes a huge load on health care systems socially and economically as it can be a waste of time and resources for both individuals and institutions. It has a dangerous effect on patients missing their appointments as well as other patients and also medical staff. Patients who miss their appointments will have reduced medical care in terms of disease screening and prevention, on the other hand it is a waste of other patients time as it results in delayed health care provision and results in decreased satisfaction with the quality of service provided [3]. Besides, it is a huge waste of medical staff time and efforts which leads to diminished medical care quality. Also, not to mention the economical side effects on health care systems due to the inefficient use of resources [4]. The importance of the efficient allocation of medical resources have particularly increased amid COVID-19.

This problem represents human behavior and it depends on so many factors, that may be why despite all the previous studies that aimed to solve this problem, it still prevails. This work aims to better understand the problem and conduct a comprehensive study to create and compare predictive models as an attempt to provide a reliable prediction that can be used in production. The models used were supervised classification machine learning models [5] in which labeled datasets were used to train the algorithms to classify and predict outcomes. Various balancing techniques were used and compared during the model building process. A combination of under-sampling and over-sampling techniques were used. Under-sampling [6] uses various algorithms to randomly delete samples from the majority class, while over-sampling [7] uses various algorithms to randomly duplicate samples from the minority class. The aim is to nearly equalize the majority and minority classes.

For evaluation, various well known evaluation metrics were used to measure and compare the performance of classification models. The evaluation process takes into consideration the four components of the confusion matrix [8]: true positive, true negative, false positive and false negative. True positives and true negatives refer to the truly predicted samples, while false positives and false negatives refer to the samples that were predicted in the wrong category.

II. Related work

In this section, relevant work tackling the problem of no-show is discussed. Researchers tried to study different techniques used to solve this problem including building machine learning models, balancing the datasets using different balancing techniques and using evaluation metrics to measure the performance of these models.

For example, in [9], D. B. Ferro et al. used machine learning algorithms as a part of the Decision Support System (DSS). Aiming to help reduce no-show rates in a primary health care program targeting underserved communities in Bogota, Colombia between 2017 and 2018. Firstly, they applied the LASSO regression model to calculate the impact of each variable on the no-show probability. Then, used a Logistic Regression model on top of that and the results obtained were used to enhance feature selection for Random Forest and Neural Network classification. For evaluation and comparison of these models, a 10-by-10 cross validation was done and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC) [10] was used as an evaluation metric. In their work, no data sampling methods were used to overcome

the data imbalance problem, instead, they used weight class balancing for their models.

Another study was carried out by I. Mohammadi et al. [11] on data collected from a large number of federally qualified health centers (FQHC) in Indianapolis between Jan 2014 and April 2016. The dataset consisted of 599,636 appointments. The study proposed three prediction models using Logistic Regression, Artificial Neural Network and Naïve Bayes classifier. For evaluation and comparison, a 10-fold cross validation was used and the average ROC-AUCs, sensitivities and overall model accuracies were the main evaluators. Although data skewness was present, no balancing methods were used to address that issue. It was concluded that the best performing models were Logistic Regression and Naïve Bayes having ROC-AUC scores of 0.81 and 0.86 respectively which was better than the Neural Network model which scored 0.66.

A different study by L. R. Chong et al. [12] using 32,957 records from outpatient MRI appointments scheduled in the radiology department between Jan 2016 and Dec 2018 and a further holdout test set of 1,080 records from Jan 2019. The study applied and evaluated various machine learning predictive models and using ROC-AUC scores alongside F1 scores. Class weight factors or under-sampling of the majority class was performed to solve the data imbalance problem. From the experimental results, it was concluded that XGBoost [13] which is an optimized distributed gradient boosting library was the best performing model with ROC-AUC score of 0.738 on the holdout test set.

In [14], M. Nasir et al. used data acquired from kaggle.com with more than 110,000 appointment records. They proposed three machine learning models (Artificial Neural Network, Support Vector Machine and Random Forest) along with a Logistic Regression model. They also proposed four data sampling techniques to address the data imbalance problem: Random Under-sampling (RUS), SMOTE (Synthetic Minority Over-sampling Technique), ADASYN (Adaptive Synthetic Sampling approach) and MWMOTE (Majority Weighted Minority Over-sampling Technique) [15]. Then, they evaluated and compared all the model variation using a 5-fold cross validation with evaluation metrics such as AUC, sensitivity, specificity and accuracy. Although Logistic Regression with Random Under-sampling had the highest sensitivity score, they concluded that the Random Forest model was the best because it had the highest AUC score which indicated a good balance between sensitivity and specificity.

III. Data And Methodology

This study used real-life data obtained from kaggle.com [16] containing 110,527 appointment records collected from several medical centers in Vitória, Brazil during three months in the year 2016. The data consists of 14 variables as follows: PatientID: identification of a patient, AppointmentID: identification of an appointment, Gender: whether male or female, ScheduledDay: the day someone called and registered an appointment, AppointmentDay: the day of the actual appointment, Age: the age of a patient, Neighborhood: the neighborhood where the appointment took place, Scholarship: a social welfare program of the Brazilian government providing financial aid to poor families, Hypertension: whether a patient has increased blood pressure or not, Diabetes: whether a patient has increased blood sugar levels or not, Alcoholism: whether a patient is an alcoholic or not, Handicap: whether a patient has a disability or not, SMSReceived: whether a patient received an sms message about the appointment or not and NoShow: whether a patient attended the appointment or not and this is the target variable. The data contained 22,319 positive samples i.e. patients did not show at their appointments (NoShow = Yes) and 88,208 negative samples (NoShow = No).

A. Data Preprocessing

Data preprocessing is a crucial part of data analysis and is very important to ensure the best practice to create a good performing model [17]. To start with preprocessing, missing and null values were checked and the data was found to contain none. When checking the data, one record was found to have a negative value for the age variable, so it was considered as noise and was dropped. Then duplicate entries were checked and 618 duplicated entries were found and dropped. With further data checking, 5 entries were found to have the appointment date preceding the reservation date, so these entries were also dropped. Besides, variable renaming and type adjustment were performed in order to make the data ready for further processing. After ensuring that the data is clean and ready, feature engineering was conducted in order to extract all the possible useful variables in the data.

B. Feature Engineering

Aiming to enhance the predictive signals within the dataset, a number of derived variables were included in this study. A variable that indicates the number of days between the reservation of the appointment and the actual date, namely WaitingDays was created. The DayOfWeek variable which indicates the day of the appointment itself was created by extracting the day value from the AppointmentDay variable. The WaitingCategory variable is also proposed to subdivide the number of waiting days into categories. Each category corresponds to a range of the days count a patient had to wait before the actual appointment. It was calculated from the derived WaitingDays variable. Upon doing so, this categorization was further checked to see its impact on the no-show problem. It was observed that appointments on the same day of scheduling had the least no-show rate among all the categories. OnSameDay variable was created to explicitly capture this feature. Besides, the VisitCount variable which indicates the total number of appointments for a patient was created along with the AgeGroups variable which subdivides the patients into different age groups. Also, the SameDayAppCount variable which indicates the number of appointments for a patient in one day was created. The last two created variables were related to the patient's no-show as in the PastNoShow variable which for a patient's current appointment it indicates whether

that patient attended his/her previous appointment or not and the NoShowRate variable which indicates the rate of a patient's appointment missing in relation to all of his/her previous appointments. Table I shows the description of all the derived variables.

TABLE I
DESCRIPTION OF THE DERIVED VARIABLES

Derived Variables	Description
WaitingDays	Indicates the number of days between the reservation of the appointment and the actual date
DayOfWeek	Indicates the day of the appointment itself
WaitingCategory	Subdivides the number of waiting days into categories
OnSameDay	Indicates appointments on the same day of scheduling
VisitCount	Indicates the total number of appointments for a patient
AgeGroups	Subdivides the patients into different age groups
SameDayAppCount	Indicates the number of appointments for a patient in one day
PastNoShow	For a patient's current appointment it indicates whether that patient attended his/her previous appointment or not
NoShowRate	Indicates the rate of a patient's appointment missing in relation to all of his/her previous appointments

C. Data Preparation

After completing the feature engineering process, the data must be configured to be suitable for machine learning models. First data shuffling was performed to ensure the random state of the data that will be introduced to model training. Then, the class label (NoShow variable) was converted into a numerical binary variable (0 for patients who attended their appointments and 1 for patients who did not attend their appointments). PatientID, ScheduleDay and AppointmentDay variables were dropped as they will not be of any use while building and training the models. After that, one hot encoding was conducted to convert all the categorical variables into numerical variables which are the type of variables needed in model building and training. Furthermore, the data was normalized using MinMax normalization to help models converge faster. For the final step, the data was split into training and testing datasets with 80 to 20 ratio.

D. Handling Data Imbalance

The data acquired for this study was highly imbalanced which means that one class is extremely dominant over the other [18], as the patients who attended their appointments are obviously four times more than those who did not (Show: 87,803 records (79.9%), NoShow: 22,100 records (20.1%)). To handle this problem, there are three categories of balancing techniques that can be used:

- **Data sampling:** which can be either under-sampling of the majority class or over-sampling of the minority class or a hybrid method that combines both [19].
- **Algorithmic modification:** which improves the learning capability of the classification algorithms to accommodate more to the imbalance issues [20].
- **Cost sensitive learning:** which can be on the data level, algorithmic level or both and aim to assign more cost to the misclassification of minority class in order to minimize the cost errors [21].

This study attempts to incorporate most of the balancing techniques available in order to come up with the most suitable solution to the imbalance problem for model creation. Different under-sampling techniques were used such as: Random Under-sampling (RUS) that randomly removes data from the majority class, AllKNN that applies a nearest-neighbors algorithm and edit the dataset by removing samples which do not agree enough with their neighborhood [22], NearMiss that selects the majority class samples for which the average distance to the N closest samples of the minority class is the smallest [23] and TomekLinks which detects the so-called Tomek's links which exist if the two samples are the nearest neighbors of each other [24].

Also, different over-sampling techniques were used like: Random Over-sampling (ROS) that randomly duplicates data from the minority class, SMOTE (Synthetic Minority Over-sampling Technique) which generates new samples in the minority class by interpolation [25] and ADASYN (Adaptive Synthetic Sampling) which acts similar to SMOTE but focuses on generating samples next to the original samples which are wrongly classified using a k-Nearest Neighbors classifier [26].

Finally, combinations of under-sampling and over-sampling techniques were used as: SMOTEENN which combines over-sampling using SMOTE with under-sampling using Edited Nearest Neighbor [27] and SMOTETomek that combines over-sampling using SMOTE with under-sampling using Tomek-Links [27]. All of the previously mentioned methods were used and compared within the model creation and evaluation process.

On the other hand, a number of algorithmic modification techniques were also used like Balanced Random Forest classifier [28] which is a modified Random Forest classifier, as well as ensemble methods that were also modified in order to better address the imbalance problem like EasyEnsemble classifier which is bag of balanced boosted learners where the classifier is an ensemble of AdaBoost learners trained on different balanced bootstrap samples and the balancing technique is Random Under-sampling [29], BalancedBagging classifier which is a bagging classifier with additional balancing [30] and RUSBoost classifier where Random Under-sampling is integrated in each iteration of the boosting algorithm [31]. Cost sensitive learning was also applied, in the form of adjusting class weight. The weights were calculated using the following formula to automatically adjust weights inversely proportional to class frequencies.

$$\frac{No.ofSamples}{No.ofClasses \times BinCount(target)} \quad (1)$$

E. Machine Learning Algorithms

This study aims to provide a comprehensive overview about different models used to solve the no-show problem. To do this a total of five models were created and compared and these models are as follows: Decision Tree classifier (DT) using gini impurity, Random Forest classifier (RF) using gini impurity, Gaussian Naïve Bayes classifier (NB), linear Support Vector Machine classifier (SVM) and Artificial Neural Network classifier (ANN). Some other modified models were used to address the imbalanced nature of the data like Balanced Random Forest classifier, and ensemble methods like bagging and boosting. Python packages such as scikit-learn [32] and imbalanced-learn [33] were used to build all of the mentioned models.

F. Model Evaluation

A 10-fold cross validation was performed in order to ensure models predictability and robustness. The training dataset (80% of the data) was used and divided into 10 splits where each split was used for testing once while the other 9 were used in training the models for each fold. Then for each fold the evaluation metrics were calculated and the average of each metric was reported to come up with the best model. Finally, the best performing model out of the 10 folds was used on the testing dataset (20% of the data).

Due to the imbalanced nature of the data, the use of the traditional accuracy metric is not considered the best way to evaluate the performance due to the bias towards the majority class [18] [34]. Therefore, metrics that are better adapted to imbalanced data were utilized. Evaluation metrics proposed for this degree of skewness include: geometric mean (G-mean) which takes into consideration both the positive and negative accuracy of each class (true positive rate and true negative rate), ROC-AUC and F1-measure scores [18] [34]. Also, balanced accuracy which is the arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate) was used [35]. Other known metrics like accuracy, precision, recall and specificity were also used as secondary comparison metrics. The main metrics are represented by the following equations:

$$True\ Positive\ Rate(TPR) = \frac{TP}{TP + FN} \tag{2}$$

$$True\ Negative\ Rate(TNR) = \frac{TN}{TN + FP} \tag{3}$$

$$G - mean = \sqrt{TPR \times TNR} \tag{4}$$

$$Balanced\ Accuracy = \frac{TPR + TNR}{2} \tag{5}$$

IV. Results and Discussion

In order to find the best-performing model, brute-force search was utilized to iterate across all possible variations. Each iteration represents a pipeline that includes a balancing technique and a machine-learning algorithm. The different variations' results can be divided into four categories:

- Results of pure models without any data balancing.
- Results of models without data balancing but with applying cost sensitive learning.
- Results of models after using different data balancing methods.
- Results of models after applying algorithmic modification.

For comparison, the G-mean score was the main evaluation metric used. If more than one model had equal scores, balanced accuracy and ROC-AUC scores were considered then F1-measure and sensitivity scores.

When the pure models were used without any balancing techniques, Naïve Bayes gave the best G-mean score of 0.63. The second performing model was Decision Tree with a G-mean score of 0.52. However, the other models performed poorly due to the effect of the data imbalance problem with scores as low as 0.14 as shown in table II.

TABLE II
Results without applying any balancing methods

Model	G-mean	Bal. Acc.	ROC-AUC	F1-score	Sens.	Spec.	Prec.	Acc.
NB	**0.63**	**0.63**	**0.63**	**0.41**	**0.60**	0.67	0.31	0.66
DT	0.52	0.58	0.58	0.33	0.33	0.83	0.33	0.73
RF	0.30	0.54	0.54	0.16	0.09	0.99	**0.61**	**0.80**
ANN	0.19	0.51	0.51	0.07	0.03	0.99	0.54	**0.80**
SVM	0.14	0.51	0.51	0.04	0.02	**1.00**	**0.61**	**0.80**

As the class weights were introduced to the model building and training the results changed as expected and SVM was the best performer with a G-mean score of 0.66 followed by Random Forest with a score of 0.64 as shown in table III.

TABLE III
Results after applying cost sensitive learning

Model	G-mean	Bal. Acc.	ROC-AUC	F1-score	Sens.	Spec.	Prec.	Acc.
SVM	**0.66**	**0.68**	**0.68**	**0.45**	**0.83**	0.53	0.31	0.59
RF	0.64	0.65	0.65	0.44	0.54	0.77	**0.37**	0.72
DT	0.52	0.58	0.58	0.33	0.33	**0.83**	0.33	**0.73**

Upon using different balancing techniques, the results obtained improved again yielding the best performance for all the models used. Model performance varied depending on the balancing methods used. When applying different under-sampling techniques, Random Forest with Random Under-sampling had the highest G-mean, balanced accuracy and ROC-AUC scores with scores 0.68, 0.69 and 0.69 respectively as shown in table IV.

On the other hand, when using over-sampling techniques, ANN with Random Over-sampling had the highest G-mean, balanced accuracy and ROC-AUC scores of 0.67, 0.68 and 0.68 respectively, as shown in table V.

TABLE IV
RESULTS AFTER APPLYING UNDER-SAMPLING METHODS

Bal. Technique	G-mean	Bal. Acc.	ROC-AUC	F1-score	Sens.	Spec.	Prec.	Acc.
Random Forest								
RUS	**0.68**	**0.69**	**0.69**	**0.46**	0.82	0.56	0.32	0.61
ALLKNN	0.67	**0.69**	**0.69**	**0.46**	0.83	0.55	0.32	0.60
TomekLinks	0.66	0.66	0.66	0.45	0.60	0.73	0.36	0.70
NearMiss	0.54	0.56	0.56	0.36	0.74	0.39	0.24	0.46
SVM								
RUS	0.66	0.68	0.68	0.45	0.82	0.54	0.31	0.59
TomekLinks	0.66	0.68	0.68	0.45	0.84	0.52	0.31	0.59
ALLKNN	0.64	0.68	0.68	0.44	**0.89**	0.46	0.30	0.55
NearMiss	0.57	0.58	0.58	0.36	0.68	0.48	0.25	0.52
ANN								
ALLKNN	0.67	0.67	0.67	0.45	0.64	0.69	0.34	0.68
RUS	0.66	0.63	0.68	0.45	0.82	0.54	0.31	0.60
NearMiss	0.55	0.57	0.57	0.35	0.69	0.44	0.24	0.49
TomekLinks	0.28	0.53	0.53	0.14	0.08	**0.98**	**0.47**	**0.80**

TABLE V
RESULTS AFTER APPLYING OVER-SAMPLING METHODS

Bal. Technique	G-mean	Bal. Acc.	ROC-AUC	F1-score	Sens.	Spec.	Prec.	Acc.
ANN								
ROS	**0.67**	**0.68**	**0.68**	**0.45**	0.81	0.55	0.32	0.61
ADASYN	0.64	0.64	0.64	0.42	0.60	0.68	0.32	0.66
SMOTE	0.63	0.63	0.63	0.41	0.56	0.71	0.33	0.68
SVM								
ROS	0.66	**0.68**	**0.68**	**0.45**	0.83	0.53	0.31	0.59
SMOTE	0.66	**0.68**	**0.68**	**0.45**	0.82	0.54	0.31	0.59
ADASYN	0.65	**0.68**	**0.68**	**0.45**	**0.85**	0.50	0.30	0.57
Random Forest								
ROS	0.66	0.66	0.66	**0.45**	0.59	0.73	0.36	0.70
ADASYN	0.54	0.61	0.61	0.37	0.33	**0.89**	**0.42**	**0.77**
SMOTE	0.54	0.61	0.61	0.37	0.32	**0.89**	**0.42**	**0.77**

The last group of balancing techniques represented the combination of under and over-sampling methods. SVM with SMOTEENN and SMOTETomek performed almost the same with equal G-mean, balanced accuracy, ROC-AUC and F1 scores but SVM with SMOTEENN had a higher sensitivity score of 0.83 as shown in table VI.

TABLE VI
RESULTS AFTER APPLYING COMBINATION OF UNDER-SAMPLING AND OVER-SAMPLING METHODS

Bal. Technique	G-mean	Bal. Acc.	ROC-AUC	F1-score	Sens.	Spec.	Prec.	Acc.
SVM								
SMOTEENN	**0.66**	**0.68**	**0.68**	**0.45**	**0.85**	0.50	0.30	0.57
SMOTETomek	**0.66**	**0.68**	**0.68**	**0.45**	0.83	0.54	0.31	0.59
Random Forest								
SMOTEENN	**0.66**	0.67	0.67	**0.45**	0.60	0.73	0.36	0.71
SMOTETomek	0.54	0.61	0.61	0.37	0.33	**0.88**	**0.42**	**0.77**

The best performing model overall was Random Forest with Random Under-sampling. It had the highest G-mean score of 0.68 followed by Random Forest with AllKNN with a G-mean score of 0.67. ANN also had close scores when used with Random Over-sampling and AllKNN with equal G-mean scores of 0.67 with slight difference in balanced accuracy and ROC-AUC scores.

On the other hand, the most consistent model was SVM as it performed equally when applying cost sensitive learning and with different balancing techniques like Random Under-sampling, Random Over-sampling, TomekLinks, SMOTE, SMOTEENN and SOMTETomek which represent the different data sampling techniques of under-sampling, over-sampling and hybrid methods. The G-mean, balanced accuracy and ROC-AUC scores yielded were as follows throughout the previously mentioned model variations: 0.66, 0.68 and 0.68 respectively.

In order to cover all the balancing techniques' categories, some algorithmic modifications were applied in the form of ensemble methods to the best performing models. Balanced Random Forest was used, while modified ensemble methods like EasyEnsemble, BalancedBagging (with both SVM and ANN) and RUSBoost (with SVM) were applied. Balanced Random Forest was the best performing model with a G-mean score of 0.68, while EasyEnsemble was the best ensemble method scoring equal G-mean score of 0.67 as the others but higher balanced accuracy and ROC-AUC scores of 0.68 and 0.68 respectively. BalancedBagging (with ANN) and RUSBoost with SVM scored equal G-mean, balanced accuracy and ROC-AUC scores of 0.67, 0.67 and 0.67 respectively. It can be noticed that the RUSBoost ensemble slightly improved the performance of the SVM model.

So to conclude the results, it was found that the best performing model was Random Forest with Random Under-sampling along with balanced Random Forest with equal G-mean scores. Table VII shows the algorithmic modification performance scores.

TABLE VII
RESULTS AFTER APPLYING ALGORITHMIC MODIFICATION

Model	G-mean	Bal. Acc.	ROC-AUC	F1-score	Sens.	Spec.	Prec.	Acc.
Balanced RF	**0.68**	**0.69**	**0.69**	**0.46**	**0.83**	0.55	0.32	0.61
EasyEnsemble	0.67	0.68	0.68	0.45	0.81	0.55	0.31	0.60
RUSBoost (SVM)	0.67	0.67	0.67	0.45	0.73	0.61	0.32	0.64
Bal.Bagg. (ANN)	0.67	0.67	0.67	0.45	0.69	**0.65**	**0.34**	**0.66**
Bal.Bagg. (SVM)	0.66	0.68	0.68	0.45	**0.83**	0.53	0.31	0.59

V. CONCLUSION

Due to the limited resources available for health care systems, the need for proper management of medical appointments is increasing. Medical appointment no-show is an obvious and recurrent problem that has major social and economic consequences. This study proposed a framework based on comparing various machine learning algorithms that can predict medical appointment no-show in an effective and accurate way. The best performing models were reported and can be used as a part of a decision making process that lower the no-show rates and increase patient satisfaction. Various balancing techniques were adopted and used in model training to overcome the data imbalance problem. The main evaluation metrics used took into consideration the positive and negative accuracies. This resulted in a better balanced evaluation of the models without bias towards the majority class which increases the dependability of the models selected. The selected best performing models were, Random Forest classifier used with Random Under-sampling and Balanced Random Forest classifier. Also, the Support Vector Machine classifier yielded the most consistent results when used with different balancing techniques which indicates the robustness of that model.

REFERENCES

[1] Z. Pei, Y. Yuan, T. Yu, and N. Li, "Dynamic allocation of medical resources during the outbreak of epidemics," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2021.

[2] N. L. Lacy, A. Paulman, M. D. Reuter, and B. Lovejoy, "Why we don't come: Patient perceptions on no-shows," *The Annals of Family Medicine*, vol. 2, no. 6, pp. 541–545, 2004. [Online]. Available: https://www.annfammed.org/content/2/6/541

[3] P. Kheirkhah, Q. Feng, L. M. Travis, S. Tavakoli-Tabasi, and A. Sharafkhaneh, "Prevalence, predictors and economic consequences of no-shows," *BMC HEALTH SERVICES RESEARCH*, vol. 16, JAN 14 2016.

[4] D. Marbouh, I. Khaleel, K. Al Shanqiti, M. Al Tamimi, M. C. E. Simsekler, S. Ellahham, D. Alibazoglu, and H. Alibazoglu, "Evaluating the Impact of Patient No-Shows on Service Quality," *RISK MANAGEMENT AND HEALTHCARE POLICY*, vol. 13, pp. 509–517, 2020.

[5] T. Jiang, J. L. Gradus, and A. J. Rosellini, "Supervised machine learning: A brief primer," *Behavior Therapy*, vol. 51, no. 5, pp. 675–687, 2020.

[6] A. Dal Pozzolo, O. Caelen, and G. Bontempi, "When is undersampling effective in unbalanced classification tasks?" 09 2015.

[7] Y.-g. Kim, Y. Kwon, and M. Paik, "Valid oversampling schemes to handle imbalance," *Pattern Recognition Letters*, vol. 125, 07 2019.

[8] S. Sai and S. Sivanandam, *Introduction to Data Mining Principles*, 01 2006, vol. 29.

[9] D. B. Ferro, S. Brailsford, C. Bravo, and H. Smith, "Improving healthcare access management by predicting patient no-show behaviour," *Decision Support Systems*, vol. 138, p. 113398, 2020.

[10] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[11] I. Mohammadi, H. Wu, A. Turkcan, T. Toscos, and B. N. Doebbeling, "Data analytics and modeling for appointment no-show in community health centers," *Journal of primary care & community health*, vol. 9, p. 2150132718811692, 2018.

[12] L. R. Chong, K. T. Tsai, L. L. Lee, S. G. Foo, and P. C. Chang, "Artificial intelligence predictive analytics in the management of outpatient mri appointment no-shows," *American Journal of Roentgenology*, vol. 215, no. 5, pp. 1155–1162, 2020.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system." New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[14] M. Nasir, N. Summerfield, A. Dag, and A. Oztekin, "A service analytic approach to studying patient no-shows," *Service Business*, vol. 14, no. 2, pp. 287–313, 2020.

[15] S. Barua, M. M. Islam, X. Yao, and K. Murase, "Mwmote–majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014.

[16] "No-show appointments dataset, kaggle.com," 2017. [Online]. Available: https://www.kaggle.com/joniarroba/noshowappointments/version/5

[17] S. Ramirez-Gallego, B. Krawczyk, S. Garcia, M. Wozniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *NEUROCOMPUTING*, vol. 239, pp. 39–57, MAY 24 2017.

[18] V. Lopez, A. Fernandez, S. Garcia, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *INFORMATION SCIENCES*, vol. 250, pp. 113–141, NOV 20 2013.

[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[20] B. Zadrozny and C. Elkan, "Learning and making decisions when costs and probabilities are both unknown," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 204–213.

[21] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 155–164.

[22] "An experiment with the edited nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 6, pp. 448–452, 1976.

[23] A. More, "Survey of resampling techniques for improving classification performance in unbalanced datasets," *arXiv preprint arXiv:1608.06048*, 2016.

[24] D. Devi, B. Purkayastha *et al.*, "Redundancy-driven modified tomek-link based undersampling: A solution to class imbalance," *Pattern Recognition Letters*, vol. 93, pp. 3–12, 2017.

[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[26] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328.

[27] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[28] C. Chen, A. Liaw, L. Breiman *et al.*, "Using random forest to learn imbalanced data," *University of California, Berkeley*, vol. 110, no. 1-12, p. 24, 2004.

[29] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2008.

[30] S. Hido, H. Kashima, and Y. Takahashi, "Roughly balanced bagging for imbalanced data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 2, no. 5-6, pp. 412–426, 2009.

[31] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2009.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

[34] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert systems with applications*, vol. 73, pp. 220–239, 2017.

[35] D. R. Velez, B. C. White, A. A. Motsinger, W. S. Bush, M. D. Ritchie, S. M. Williams, and J. H. Moore, "A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction," *Genetic Epidemiology: the Official Publication of the International Genetic Epidemiology Society*, vol. 31, no. 4, pp. 306–315, 2007.