Predicting No-showMedical Appointments Using Machine Learning

Sara Alshaya[1], Andrew McCarren[2], and Amal Al-Rasheed[3]

[1] Dublin City University, Dublin, Ireland
sara.alshaya4@mail.dcu.ie
[2] Insight Center for Data Analytics, Dublin City University, Dublin, Ireland
andrew.mccarren@dcu.ie
[3] Princess Nourah University, Riyadh, Kingdom of Saudi Arabia
aaalrasheed@pnu.edu.sa

Abstract. Health care centers face many issues due to the limited availability of resources, such as funds, equipment, beds, physicians, and nurses. Appointment absences lead to a waste of hospital resources as well as endangering patient health. This fact makes unattended medi- cal appointments both socially expensive and economically costly. This research aimed to build a predictive model to identify whether an appointment would be a no-show or not in order to reduce its consequences. This paper proposes a multi-stage framework to build an accurate predictor that also tackles the imbalanced property that the data exhibits. The first stage includes dimensionality reduction to compress the data into its most important components. The second stage deals with the imbalanced nature of the data. Different machine learning algorithms were used to build the classifiers in the third stage. Various evaluation metrics are also discussed and an evaluation scheme that fits the problem at hand is described. The work presented in this paper will help decision makers at health care centers to implement effective strategies to reduce the number of no-shows.

Keywords: Machine learning · Deep learning · No-show · Data imbalance · Dimensionality reduction

1 Introduction

Hospitals suffer from a number of different problems that affect their services in many ways. The total budget reserved for health care in the U.S. for the 2013 fiscal year was 3.8 trillion dollars, this represented 23.3% of the total gross domestic product (GDP) [1]. The limited resources available for health care in terms of funds, equipment, beds, physicians, and nurses could lead to various consequences depending on how they are allocated [5]. In addition, the early discharge of patients to admit other patients with more critical conditions [20] is common due to a lack of beds. Inefficient human resources management can

also lead to an insufficient number of available staff members to handle emergencies and disasters [30]. One of the important problems that face health care centers is when patients miss their scheduled medical appointments without cancellation, i.e., a no-show. The reservation of an appointment involves the allocation of health care providers' time, medical equipment, room, etc. Therefore, an increased ratio of no-show appointments could cause a severe waste of already scarce resources. This in turn could potentially endanger the lives of many who need timely interventions. This research aims to build a classifier to predict whether a scheduled appointment will be attended or not. This in turn could help hospital and clinic administrators to define effective strategies to mitigate the no-show problem. Overbooking could be an effective strategy which involves booking extra appointments on the days of high predicted no- shows [15,23]. Also, sending SMS reminders to patients who are likely to miss their appointments [3] could be an option. Reservation fees at the time of book- ing have also been found to be an effective deterrent [2]. Due to the nature of the data, a sub-problem arises, namely data imbalance. Most of the available datasets for this problem are imbalanced, as the percentage of missed appoint- ments is naturally lower than the percentage of those attended. Data imbalance in the training and validation sets could cause the resulting models to be biased towards the majority class (i.e., those who do not miss their appointments). Therefore, the approach proposed in this paper addresses this issue [19,22,26].

2 Related Work

2.1 Machine Learning Methods for Missed Appointments

A relevant work [16] used stepwise logistic regression on a dataset obtained from Kaggle[1], which included information about medical appointments in Brazil. The researchers divided the dataset into four independent populations, depending on age group, above-18 or under-18 (adults, children), and whether they visited the clinic once or many times (non-recurrent, recurrent). Stepwise logistic regression was used to train a classifier for each population. Area under the receiver operator characteristic curve (AUC ROC) and prediction accuracy were reported to evaluate the models. Table 1 shows the results obtained on the test sets for the four predictors. Another work by [27] also utilized stepwise logistic regres- sion. They introduced an insightful variable calculated for each patient using an empirical Markov model based on up to 10 previous appointments. Variables were selected based on a likelihood ratio according to the following criteria: (1) the p-value to enter was set at 0.05, and (2) the p-value for removal was set at 0.10. The variables that were found significant in all 24 models were: the nat- ural log of appointment age, multiple appointments per day, and the empirical Markov model value based on past attendance history. The probability that a patient will miss his/her appointments decreased as he/she got older and that effect was present in all 24 models. The area under the ROC curve was used to

[1] https://www.kaggle.com/joniarroba/noshowappointments.

assess the model's performance. The average test accuracy was 0.762. While the average test AUC ROC score was 0.713.

Table 1. Results summary of the method proposed by [16]

Population	ROC	Prediction accuracy
No-recurrent children	0.7564	79.05%
Recurrent children	0.6893	76.35%
Non-recurrent adults	0.7503	81.88%
Recurrent adults	0.7030	79.54%

A method to calculate the threshold of stepwise logistic regression has been proposed by [14] by minimizing the misclassification count. The authors assumed a higher cost for a show misclassified as a no-show than for a no-show misclassified as a show. They designed the cost function as given in (1). They assume that c_{show} is greater than $c_{no-show}$. Given these assumptions, they investigated two values $\frac{c_{show}}{c_{no-show}}$, 2 and 3, to determine their impact on minimizing show errors at the expense of additional no-show errors.

$$c_{show}[\sum E_{show}/N_{show}] + c_{no-show}[\sum E_{no-show}/N_{no-show}] \quad (1)$$

Using the cost function given by (1) for the training, they optimized the probability threshold. Given a cost ratio of 2, the error is minimized at the threshold of 0.86. For a cost ratio of 3, the threshold is 0.74. Applying a threshold of 0.74 gave a better accuracy on the training set as illustrated in Table 2. The model test accuracy was 86.1%. The overall error rate was 13.9%, which consisted of 3.9% show errors and 87.2% no-show errors.

Table 2. Error percentages on training set for the method proposed by [14]

Threshold	Show error	No-show error
0.86	11.3%	68.1 %
0.74	1.8%	91.9 %

A different study by [21] considered three different machine learning algorithms and compared their performance. In addition to the variables that were already represented in the data, the authors derived three more variables: (1) lead time, which is the time difference (in days) between the date of visit and the reservation date, (2) prior no-show rate, which is the portion of no-shows for a given patient prior to the last appointment, and (3) days since the last appointment, which is the number of days between the date of the last visit

and the date of appointment. The authors reported that smoking was one of the most significant factors related to missing medical appointments. They found that lower income and unemployment were associated with more missed medi- cal appointments. The results also showed that patients without insurance for medical services were at risk for not adhering to their appointments and consequently, their care plans. The three machine learning algorithms included in the study were, stepwise logistic regression, feed-forward neural net, and naïve Bayes. A multilayer perceptron structure was used as a neural net with a hidden layer of 25 nodes. A smoothing value of 0.1 provided the best performance for the na¨ıve Bayes classifier. Table 3 shows the results on both the training and test sets using all the prediction models as reported by the authors. As determined by the experiments, na¨ıve Bayes had the better performance.

Table 3. Summary of the results by [21] on Both Training and Testing Sets

Model	Training set		Test set	
	AUC	Accuracy (%)	AUC	Accuracy (%)
Logistic regression	0.91	80%	0.81	73%
Neural net	0.77	79%	0.66	71%
Naive Bayes	**0.96**	**92%**	**0.86**	**82%**

2.2 Methods Used for the Data Imbalance Problem

Different methods can be utilized to tackle a data imbalance problem. Some can be performed at the data level, while others can be performed at the algorithm level. A hybridization of both is also possible. Methods for addressing this prob- lem can be categorized into three major groups [11,19]: (1) data sampling, which includes either undersampling the majority class (eliminating some observations) or oversampling the minority class (replicating some observations), (2) algorith- mic modifications, which modify the learning algorithms using techniques that account for the imbalanced nature of the data, such as a balanced random for- est, and (3) cost-sensitive learning in which a higher misclassification cost for the samples from the minority class is assumed.

In the method proposed by [12], the authors tackled data imbalance in a medical diagnosis dataset by introducing a distribution sensitive oversampling approach. In the proposed method, the minority samples were divided into noise samples, unstable samples, boundary samples, and stable samples according to their location in the distribution. Depending on a minority sample's distance from other surrounding minority samples, different replication methods were applied. This was performed to ensure that newly created samples had the same characteristics as the original ones. In the replication process, minority noise and unstable samples were excluded. For each minority sample not characterized as

noise or unstable, the accumulative distance between it and its k neighbors was calculated. If the distance was less than or equal to a defined threshold, then a few samples were generated using this sample. If the cumulative distance was greater than the threshold, then as many samples as possible were created using it. Testing on real medical diagnosis data showed that compared to existing sampling algorithms, the classification learning algorithm was more accurate when using the proposed method, especially in terms of the precision and recall rate of minority classes.

A hybridization of undersampling and algorithmic modifications was proposed by [18]. The authors proposed two methods, EasyEnsemble and BalanceCascade. The EasyEnsemble method includes subsetting the data at random to ensure that the number of majority and minority samples are equal. Then, a number of sub-classifiers are trained using these subsets. The final decision is the result of combining all the sub-classifiers after each is trained using the AdaBoost [24] algorithm. In the second method, BalanceCascade, after every classifier is trained the majority samples that were classified correctly are eliminated from the training set and are then fed to the next sub-classifier, so that every classifier uses a balanced dataset. In BalanceCascade, the final classifier is different from the EasyEnsemble classifier. While EasyEnsemble's final prediction is created by forming an ensemble classifier that combines all sub-classifier predictions, BalanceCascade predicts a positive value if and only if all sub-classifiers predict a positive value. The two methods were tested using datasets suffering from a high imbalance ratio, referred to as "hard" datasets, and "easy" datasets with lower imbalance ratios. The results showed that for easy tasks, most of the class imbalance learning methods had lower AUC ROC scores than Ada. On the other hand, for hard tasks, class imbalance learning methods generally had higher AUC ROC scores than Ada, including SMOTE, Chan [7], Cascade, and Easy. The authors reported that for tasks on which ordinary methods could have high AUC scores, class-imbalance learning was unhelpful. However, Easy and Cascade reduced training time (3.50 and 5.50 for Easy and Cascade vs. 19.83 and 18.63 for Ada and Asym [28]), while their average AUC ROCs were similar to that of Ada and Asym. Therefore, class-imbalance learning was particularly helpful for hard tasks. In this paper, different machine learning algorithms were studied and compared, such as random forest, support vector classifier, and stochastic gradi- ent descent. To tackle data imbalance, the effect of using an ensemble classifier was studied. The ensemble classifier used was balanced random forest, which performed competitively.

3 Proposed Framework

In order to achieve the objective of this research, represented in designing and training a high-performing predictive model for no-show appointments, the framework used is introduced. Figure 1 illustrates the general framework used to build the proposed solution. At each phase, different techniques that were applied in order to generate various combinations of techniques. In the following

section, the purpose of each phase is highlighted along with the summarization of all techniques used in each phase.

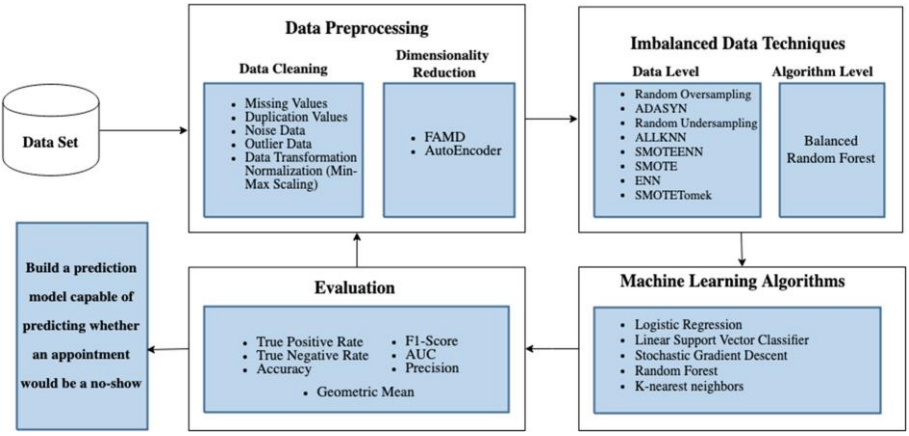

Fig. 1. The general framework followed to perform this study.

3.1 Data Collection and Preprocessing

The dataset used for this study was obtained from the Kaggle (See footnote 1) medical appointment no-show dataset. This dataset contained information about scheduled medical appointments in Brazil. This dataset is available in different versions, and the one used here was the version released in May 2016. The dataset contained 14 variables. These variables were used to derive seven more that were used in initially building the predictive model. These variables were: lead days, on same day, day of week, patient's visits count, patient's no-show rate, no-show last appointment, and appointments count at same day. All derived variables are described in Table 4. Since the weather is believed to affect appointment status, a weather dataset was studied and merged with the original dataset. The no-show appointments dataset provider indicated that all the hospital neighborhoods in the data were located in the city of Vitoria. Therefore, a dataset was obtained from the weather website[2] for the months (April, May, and June) included in the appointment dataset.

A check for missing values was performed with the appointment dataset, and none were found. Also, no duplicate records were observed. However, some problematic values were observed when analyzing the attributes to check for potential noise or outliers. For example, one appointment was scheduled for a patient whose age was recorded as–1. Since there are no negative ages, this case was considered as noise and was dropped. Also, there were seven appointments for patients older than 100 years old. There were five appointment records with

[2] https://www.tutiempo.net.

Table 4. Derived variables description.

Lead days	The number of days between the reservation day and the actual appointment day
On same day	An indication as to whether the appointment takes place on the same day it was reserved, i.e. lead days = 0
Day of week	It is represented as integer numbers (0 for Monday, 1 for Tuesday etc.)
Patient's visits count	The number of appointments the patient has in the entire dataset
Patient's no-show rate	The number of no-show appointments divided by the patient's total number of past appointments
No-show last appointment	An indication to whether the patient's last appointment was a no-show or not
Appointments count at same day	This indicates the number of appointments the patient has on the same day of the appointment under consideration

a patient age of 115 and there were two unique appointment records for patients of age 102. Since these cases were extremely rare, they were considered outliers and were removed from the analysis.

For the weather dataset, a check for missing values was performed, and total precipitation of rain and/or the melted snow indicator had four missing values. Since there was no known relationship between these missing values, they were considered to be missing completely at random. The mean of precipitation of rain and/or melted snow over the respective month was used to impute the missing values.

To enable the different algorithms and techniques in the framework to use the data, the categorical variables were encoded into numerical form. For this purpose, one-hot encoding was performed by creating a separate variable for each category of a given variable.

In datasets, where variables have disparate distribution characteristics, feature scaling is usually recommended [25]. Such characteristics can slow the learning rate, thus preventing convergence. In this research, a min-max scaling approach was performed as the variables were not normally distributed. Min-max scaling is performed by subtracting the minimum value of each variable from its respective variable's values and dividing the result by the difference between the maximum and the minimum.

3.2 Dimensionality Reduction

In this study, two methods for dimensionality reduction were applied. Since the dataset contained both categorical and numerical variables, a factor analysis

of mixed data (FAMD) [4] was used, also a deep-learning-based method that performs dimensionality reduction in an unsupervised manner, namely AutoEncoder (AE) [29], was applied. AE was implemented as a fully-connected three-layered neural network with one hidden layer and a sigmoid activation func- tion for all nodes. The input data was split in a 80:20 ratio of training to test data. The autoencoder was trained using AdaDelta optimization to minimize the binary cross-entropy loss function. For both methods, the number of resulting components was set to 10 which represents 10% of all features after the one-hot encoding.

3.3 Data Balancing

Data balancing techniques were performed to avoid bias in the learned models. A number of balancing techniques at both the data and algorithm level were explored in this research. The techniques utilized in this paper included: over-sampling balancing (random oversampling [10], adaptive synthetic (ADASYN) [10], and SMOTE [8]), undersampling balancing (random undersampling [10], AllKNN, edited nearest neighbors [13]), and hybrid techniques (SMOTEENN, SMOTETomek).

3.4 Machine Learning Algorithms

The logistic regression [6], random forest, k-nearest neighbors, support vector classifier [9], and stochastic gradient descent algorithms were implemented and evaluated. Logistic regression is a commonly used algorithm for binary classification problems similar to the one at hand. In addition, random forest is used to build a strong classifier using an ensemble of weaker ones, therefore, it yields high scores for many problems. The support vector classifier algorithm is a kernel-based method that deals with non-linearity by transforming the data points into a higher dimension space where there is a separating hyperplane. Due to its simplicity, the k-nearest-neighbors algorithm was also included in the experiments. Since most real-life datasets are linearly inseparable, the support vector classifier was included in the study. The stochastic gradient descent was also studied to speed up the training of the support vector machine. For the experiments, the number of trees in the random forest (RF) was set to 100. The number of features considered in data splitting was identified as the square root of the total number of features. The criterion selected for measuring the quality of data splits was set to the Gini index. To find the weights of the logistic regres- sion that corresponded to the least error, liblinear was used as an optimization method, which is indicated as suitable for small datasets. The support vector classifier (SVC) with a linear kernel was built, the loss (i.e., the error function) was set to hinge. The maximum number of iterations was set to 15 000, since it did not converge using the default value of 100, and more than 15 000 was computationally expensive in terms of training time. In order to speed up the training of the linear support vector classifier and to be able to set the maximum

number of iterations to a higher value, the stochastic gradient descent (SGD) method with a hinge loss function was used to fit a support vector classifier with a maximum number of iterations set to 50 000. The number of neighbors for k-nearest neighbors was specified to be five, and the algorithm used to specify the nearest neighbors was KDTree. The implementation of the balanced random forest (BRF) [18] was performed as a technique for tackling the data imbalance issue on the algorithm level. Being a type of random forest, it was a collection of decision trees, and was similarly set to 100 trees. The criterion used to measure the data splits quality was also set to the Gini index.

4 Experimental Settings

In order to organize the experiments and to facilitate the training and cross-validation of each setting, the suggested framework was implemented as a pipeline. Feature scaling was included as the first step of the pipeline. The sec- ond step was the dimensionality reduction technique utilized (FAMD (n = 10), AE, or no reduction). The data sampling balancing technique was then included, where a technique was chosen from those described earlier, or no balancing was performed. Finally, the machine learning algorithm used for training the model was executed. It was chosen from the algorithms defined for this research. If the learning algorithm was a balanced algorithm, this meant that no data sampling step was needed.

4.1 Evaluation

For each experimental setting, a 10-fold cross-validation was performed by dividing the dataset into ten splits, each one was used once for testing while the remaining nine were used for training at each step. The process was performed 10 times and performance metrics were calculated for each. Finally, the aver- age of every evaluation metric was reported. Accuracy, training time, F1-score, precision, and the AUC ROC obtained on the test set were used as evaluation metrics. True positive and true negative rates were implemented by utilizing the true positive, true negative, false positive, and false negative results from the confusion matrix. Additionally, the geometric mean (G-mean) was used since it is recommended for problems with imbalanced data [17]. G-mean can be calcu- lated in terms of both (TPR) and (TNR) as in (2).

$$G - mean = \sqrt{TPR \times TNR} \tag{2}$$

5 Experimental Results

Table 5 details the best results obtained by the different machine learning algorithms (i.e., logistic regression, random forest, linear SVC, stochastic gradient

descent, and KNN). Each machine learning algorithm was studied by combining it with the different dimensionality reduction and data balancing techniques included in this research. The best results were determined using G-mean as the primary evaluation metric; if there was more than one setting that had the top G-mean score, the F1-score was considered; and if there was still a tie, then the AUC ROC was used.

Five different settings of logistic regression obtained the best results according to the evaluation scheme described. All of these five settings used AE with 10 components as a dimensionality reduction method in combination with ADASYN, SMOTE, random undersampling, SMOTEENN, and SMORETomek.

The random forest algorithm scored the best when used in combination with FAMD with 10 components and SMOTEENN.

The support vector classifier obtained the best results on five different settings, four of them used AE and one of the balancing techniques: ADASYN, random oversampling, random undersampling, or SMOTEENN. The fifth used FAMD with AllKNN as a balancing technique.

Six different settings yielded the best results in the case of SGD, four of these settings used AE and the remaining two used no dimensionality reduction. For the settings that used AE, either RandomOverSampler, SMOTE, SMOTEENN, or SMOTETomek were used.

Since the training of the k-nearest-neighbors algorithm when applied on all features without dimensionality reduction was very time-consuming, it was only used with dimensionality reduction. This emphasizes the importance of dimensionality reduction in some cases when using all features is computationally expensive. K-nearest neighbors performed the best when used with ENN as a balancing technique and FAMD with 10 components as a dimensionality reduction method.

Lastly, the best performance obtained by balanced random forest was when combined with AE for dimensionality reduction.

The results show that according to the evaluation scheme, SGD and SVM had the top performance in terms of G-mean, F1-score, and AUC ROC. Although logistic regression had the same G-mean and AUC ROC scores, it was slightly outperformed in terms of F1-score by both SGD and SVM. Best performances were mostly obtained using AE for dimensionality reduction.

6 Contribution

This paper contributes in solving the problem of no-show medical appointments. It proposes a framework for building a prediction model using different machine learning algorithms, various balancing techniques, and dimensionality reduction methods. It tackles the problem of data imbalance to avoid the bias in the trained models. It also introduced AE and FAMD as possible options for dimensionality reduction in the no-show prediction. The effectiveness of using dimensionality reduction to enhance the performance in general and reduce the time for training time-consuming models has been empirically demonstrated. A number of derived

attributes were also calculated to increase the expressive power of the data. Since weather is believed to affect the no-show, a weather dataset was collected and merged with the original data to enhance the performance.

Table 5. Best experimental results as obtained by all included algorithms combined with all dimensionality reduction and balancing techniques.

ML algorithm	Balancing	Dimensionalityreduction	AUCROC	Acc.	PR	F1-score	TPR	TNR	G-mean	Fittingtime (sec.)
Logistic regression	ADASYN	AE (n=10)	**0.68**	0.54	0.61	0.43	0.84	0.47	**0.62**	8.65
	SMOTE	AE (n = 10)	**0.68**	0.55	0.60	0.43	0.81	0.49	**0.62**	9.55
	RandomUnderSampler	AE (n = 10)	**0.68**	0.56	0.60	0.43	0.81	0.49	**0.62**	9.25
	SMOTEENN	AE (n = 10)	**0.68**	0.53	0.61	0.43	0.88	0.44	**0.62**	14.21
	SMOTETomek	AE (n = 10)	**0.68**	0.55	0.60	0.43	0.82	0.49	**0.62**	14.99
Random forest	SMOTEENN	FAMD(n=10)	0.65	0.62	0.57	0.36	0.57	0.63	0.57	29.72
Support vector classifier	ADASYN	AE (n = 10)	**0.68**	0.52	0.62	**0.44**	0.91	0.42	**0.62**	10.48
	RandomOverSampler	AE (n = 10)	**0.68**	0.52	0.62	**0.44**	0.91	0.42	**0.62**	9.99
	RandomUnderSampler	AE (n = 10)	**0.68**	0.52	0.62	**0.44**	0.91	0.42	**0.62**	5.29
	SMOTEENN	AE (n = 10)	**0.68**	0.52	0.62	**0.44**	0.91	0.42	**0.62**	11.37
	AllKNN	FAMD(n=10)	**0.68**	0.52	0.62	**0.44**	0.92	0.41	**0.62**	11.81
Stochastic gradient descent	RandomOverSampler	AE (n = 10)	**0.68**	0.52	0.62	**0.44**	0.91	0.42	**0.62**	13.57
	SMOTE	AE (n = 10)	**0.68**	0.52	0.62	**0.44**	0.91	0.42	**0.62**	14.83
	SMOTEENN	AE (n = 10)	**0.68**	0.52	0.62	**0.44**	0.91	0.42	**0.62**	20.37
	SMOTETomek	AE (n = 10)	**0.68**	0.52	0.62	**0.44**	0.91	0.42	**0.62**	20.58
	RandomOverSampler	None	**0.68**	0.52	0.62	**0.44**	0.92	0.42	**0.62**	2.05
	SMOTEENN	None	**0.68**	0.52	0.62	**0.44**	0.92	0.42	**0.62**	648.85
K-nearest neighbors	ENN	FAMD(n=10)	0.67	0.58	0.60	0.40	0.72	0.55	0.60	7.94
Balanced random forest		AE (n = 10)	0.58	0.52	0.55	0.36	0.65	0.49	0.56	17.05

7 Conclusion

In this work, a framework for building a high-performing balanced predictor for no-show medical appointments has been proposed. Logistic regression, ran- dom forest, stochastic gradient descent, k-nearest neighbors, and linear support vector classifier have been explored as possible machine learning algorithms to apply. To avoid the consequences of data imbalance, various data balancing techniques on both data and algorithmic levels have been tested. Two differ- ent dimensionality reduction techniques, AE and FAMD, as they both fit data of a mixed nature (i.e., that includes a mixture of categorical and numerical variables) have been

implemented. Geometric mean (G-mean) was used as the primary evaluation metric, while F1-score and AUC ROC were considered when G-mean scores were equal for different settings. The best performance accord- ing to the described evaluation scheme was obtained by both SGD and SVM in combination with different dimensionality reduction and balancing techniques. The results also showed that the best scores were mostly obtained using AE for dimensionality reduction which emphasizes the usefulness of using novel unsu- pervised dimensionality reduction based on deep learning.

References

1. United states office of management and budget: fiscal year 2013 budget of the U.S. government. https://www.govinfo.gov/content/pkg/BUDGET-2013-BUD/pdf/BUDGET-2013-BUD.pdf#page=214
2. Aggarwal, A., Davies, J., Sullivan, R.: "nudge" and the epidemic of missed appointments: can behavioural policies provide a solution for missed appointments in the health service? J. Health Organ. Manag. **30**(4), 558–564 (2016)
3. Arora, S., et al.: Improving attendance at post-emergency department follow-up via automated text message appointment reminders: a randomized controlled trial. Acad. Emerg. Med. **22**(1), 31–37 (2015)
4. Bécue-Bertaut, M., Pagès, J.: Multiple factor analysis and clustering of a mixture of quantitative, categorical and frequency data. Comput. Stat. Data Anal. **52**(6), 3255–3268 (2008)
5. Belciug, S., Gorunescu, F.: Improving hospital bed occupancy and resource utilization through queuing modeling and evolutionary computation. J. Biomed. Inform. **53**, 261–269 (2015)
6. Branco, P., Torgo, L., Ribeiro, R.P.: A survey of predictive modeling on imbalanced domains. ACM Comput. Surv. (CSUR) **49**(2), 31 (2016)
7. Chan, P.K., Stolfo, S.J.: Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In: KDD 1998, pp. 164–168 (1998)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
9. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
10. Estabrooks, A., Jo, T., Japkowicz, N.: A multiple resampling method for learning from imbalanced data sets. Comput. Intell. **20**(1), 18–36 (2004)
11. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learn- ing from class-imbalanced data: review of methods and applications. Expert Syst. Appl. **73**, 220–239 (2017)
12. Han, W., Huang, Z., Li, S., Jia, Y.: Distribution-sensitive unbalanced data oversampling method for medical diagnosis. J. Med. Syst. **43**(2), 39 (2019)
13. He, H., Ma, Y.: Imbalanced lEarning: Foundations, Algorithms, Andapplications. Wiley, Hoboken (2013)
14. Huang, Y., Hanauer, D.A.: Patient no-show predictive model development using multiple data sources for an effective overbooking approach. Appl. Clin. Inform. **5**(03), 836–860 (2014)
15. Huang, Y., Zuniga, P.: Dynamic overbooking scheduling system to improve patient access. J. Oper. Res. Soc. **63**(6), 810–820 (2012)
16. Kheirkhah, P., Feng, Q., Travis, L.M., Tavakoli-Tabasi, S., Sharafkhaneh, A.: Prevalence, predictors and economic consequences of no-shows. BMC Health Serv. Res. **16**(1), 13 (2015)
17. Kubat, M., Matwin, S., et al.: Addressing the curse of imbalanced training sets: one-sided selection. In: ICML, Nashville, USA, vol. 97, pp. 179–186 (1997)
18. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **39**(2), 539–550 (2008)
19. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf. Sci. **250**, 113–141 (2013)

20. Mallor, F., Azcárate, C., Barado, J.: Control problems and management policies in health systems: application to intensive care units. Flexible Serv. Manuf. J. **28**(1–2), 62–89 (2016)
21. Mohammadi, I., Wu, H., Turkcan, A., Toscos, T., Doebbeling, B.N.: Data analytics and modeling for appointment no-show in community health centers. J. Primary Care Commun. Health **9**, 2150132718811692 (2018)
22. Nanni, L., Fantozzi, C., Lazzarini, N.: Coupling different methods for overcoming the class imbalance problem. Neurocomputing **158**, 48–61 (2015)
23. Nuti, L.A., et al.: No-shows to primary care appointments: subsequent acute care utilization among diabetic patients. BMC Health Serv. Res. **12**(1), 304 (2012)
24. Schapire, R.E.: A brief introduction to boosting. In: IJCAI, vol. 99, pp. 1401–1406 (1999)
25. Singh, B.K., Verma, K., Thoke, A.: Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification. Int. J. Comput. Appl. **116**(19) (2015)
26. Sun, Y., Wong, A.K., Kamel, M.S.: Classification of imbalanced data: a review. Int. J. Pattern Recognit. Artif. Intell. **23**(04), 687–719 (2009)
27. Vargas, D.L., et al.: Modeling patient no-show history and predicting future outpatient appointment behavior in the veterans health administration. Mil. Med. **182**(5/6), E1708 (2017)
28. Viola, P., Jones, M.: Fast and robust classification using asymmetric adaboost and a detector cascade. In: Advances in Neural Information Processing Systems, pp. 1311–1318 (2002)
29. Wang, Y., Yao, H., Zhao, S.: Auto-encoder based dimensionality reduction. Neurocomputing **184**, 232–242 (2016)
30. Xiang, Y., Zhuang, J.: A medical resource allocation model for serving emergency victims with deteriorating health conditions. Ann. Oper. Res. **236**(1), 177–196 (2016)