INDEPENDENT JOURNAL OF MANAGEMENT & PRODUCTION (IJM&P)
http://www.ijmp.jor.br v. 12, n. 8, November-December 2021
ISSN: 2236-269X
DOI: 10.14807/ijmp.v12i8.1481

PREDICTION OF INDICATORS THROUGH MACHINE LEARNING AND ANOMALY DETECTION: A CASE STUDY IN THE SUPPLEMENTARY HEALTH SYSTEM IN BRAZIL

Mirele Marques Borges
Universidade Federal do Rio Grande do Sul, Brazil
E-mail: mirelem.borges@gmail.com

Cláudio José Müller
Universidade Federal do Rio Grande do Sul, Brazil
E-mail: cmuller@producao.ufrgs.br

Submission: 11/2/2020
Revision: 12/16/2020
Accept: 12/1/2020

ABSTRACT

The research aimed to investigate the stages of a Machine Learning model process creation in order to predict the indicator over the number of medical appointments per day done in the area of supplementary health in the region of Porto Alegre / RS - Brazil and to propose a metric for anomalies detection. Literature review and applied case study was used as a methodology in this paper, besides was used the statistical software called R, in order to prepare the data and create the model. The stages of the case study was: database extraction, division of the base in training and testing, creation of functions and feature engineering, variables selection and correlation analysis, choice of the algorithms with cross-validation and tuning, training of models, application of the models in the test data, selection of the best model and proposal of the metric for anomalies detection. At the end of these stages, it was possible to select the best model in terms of MAE (Mean Absolute Error), the Random Forest, which was the algorithm with better performance when compared to Linear Regression and Neural Network. It also makes possible to identified nine anomaly points and thirty-eight warning points using the standard deviation metric. It was concluded, through the proposed methodology and the results obtained, that the steps of feature engineering and variables selection were essential for the creation and selection of the model, in addition, the proposed metric achieved the objective of generates alerts in the indicator, showing cases with possible problems or opportunities.

Keywords: Machine Learning, Indicators, Anomaly Detection, Feature Engineering e Supplementary Health System.

1. INTRODUCTION

Every day computers from all over the world are generating and storing data of the most varied types. According to James et al. (2013), with the emergence of Big Data, modeling and understanding complex databases through statistical models has become a subject in evidence in different areas of science. For Burrel (2016) and Lee (2019), Machine Learning algorithms are good tools when the goals is to make predictions in large databases, because, by combining statistical techniques and artificial intelligence, Machine Learning algorithms are able to solve problems, such as: fraud detection , prediction of indicators, prediction of behaviors and even early diagnosis of diseases.

In the Brazilian health sector, in addition to the public health system, there is also the private system called supplementary health. According to De Araujo et al. (2015), ANS (National Supplementary Health Agency) regulates more than 1500 health plan operators existing in Brazil today. Many of these operators are in an unstable financial situation, due to the difficulty in forecasting assistance costs. For this reason, these companies have been implementing technology to detect unnecessary exams, costly procedures without justification and medical fraud, thus guaranteeing a better service. Being able to predict in advance a cost behavior, number of medical appointment, number of patients, etc. improves the strategic and financial sector of health plan operators that subsidize the cost of assistance. With an accurate prediction of the indicators, a fairer value for the population can be guaranteed.

Therefore, this article aims to demonstrate the process of creating a Machine Learning model to predict the number of medical appointment in the area of supplementary health in the region of Porto Alegre / RS, in addition to proposing metrics for detecting anomalies for this indicator. The necessary steps to create a predictive model will be presented, such as: collecting and preparing the database, selecting the variables, choosing the algorithm, selecting the parameters, training the model and evaluating the results. All analyzes and codes presented in this article were developed using the statistical software R, which is an open-source programming language that makes it possible to share the knowledge developed in this paper with the whole R community.

2. MACHINE LEARNING

Marsland (2014) and Bishop (2006) define the term Machine Learning, as a set of techniques that aims to learn from historical data, that is, using computational strength to better predict patterns, behaviors, or even perform the classification and creation of groups. The numerous algorithm techniques, according to Marsland (2014) and James et al. (2013), can be classified as: supervised learning and unsupervised learning. Supervised learning algorithms are used for classification or regression problems, when the response variable is also known a priori. Still according to James et al. (2013), the unsupervised learning algorithms are a little more challenging because there is no response variable to be predicted, being normally used for models that aim to identify relationships between the variables or observations.

According to Marsland (2014), the process of creating a supervised Machine Learning model for predicting continuous data (regression problem) must follow some steps: (i) collecting and preparing the data, (ii) selecting the variables, (iii) choice of algorithm, (iv) selection of hyperparameters, (v) training of the model and (vi) evaluation of results.

The collection and preparation of data can be done by selecting a group of variables potentially important for the proposed objective. According to Marsland (2014), this set can be tested in order to choose the best set of variables present in the original base. It is at this stage that the division of the base in training and testing is also carried out. Being the training data used for all the steps related to the discovery of knowledge, and the test data having its use restricted only to the stage of validation of the results.

According to Garcia et al. (2007), it is in the stage of preparing the database that descriptive analyzes are carried out in order to know, clean, group, transform and enrich the data, the latter being known technically as Feature Engineering. According to Garla et al. (2012), Feature Engineering is the process of creating new variables from those already existing in a database, either by combining two or more variables, or by creating a new variable extracted from an existing one.

As Marsland (2014), the selection of variables is an important step from being discharged time to analyze all the variables. At this stage, are selected the most useful variables to explain the problem. According to James et al. (2013), there are several types of variable selection methods that allow a better interpretability of the model and reduces the risk of overfitting, such as: Subset Selection and Stepwise Selection. The Subset seletion method is used for any addition or removal procedure variables in a pre-existing set, or select exhaustively the subset of variables which maximize the desired result. As the number of combinations depends on the number of variables in base, it becomes costly to obtain metrics for all possible subsets of variables. To solve this problem, it is common to use the Stepwise Selection Forward or Stepwise Selection Backward Method, which are Subset Selection techniques that respectively add and remove variables sequentially.

According to Burrel (2016), the choice of the Machine Learning algorithm must consider the computational capacity available and the type of problem to be solved. In addition to these items, it is very common to perform a combination of models or to compare models in order to identify the best result. Also according to Burrel (2016), the most popular examples of Machine Learning are Neural Network, Decision Tree, Logistic Regression, in addition to Linear Regression and Random Forest.

According to Marsland (2014) and Kraska et al. (2013), many algorithms require tuning hyperparameters, which can be selected manually or through tests in order to identify the most appropriate hyperparameter. This selection is commonly made in an exhaustive way, that is, an interval is tested for each hyperparameter and all possible combinations of these intervals are performed. In order to guarantee the robustness of the results generated by such algorithms, the Cross-validation technique can be used.

Refaeilzadeh et al. (2009) defines Cross-validation as a statistical method used to validate Machine Learning algorithms, by crossing the training base with a validation base, so that all data sets are validated. The most common method is the k-fold, which divides k sub-samples of the training base, if the number chosen for k is equal to 10, the training base will be divided into 10 sub-samples that will be crossed and validated with each other.

According to Marsland (2014) and Vinyals et al. (2019), after the steps of data preparation, selection of variables, selection of algorithms and selection of parameters, it is possible to perform the training stage of the models through some computational resource. In this step, the model will receive inputs and generate outputs, that is, it will receive explanatory variables and estimate the response variable.

In the results assessment stage, the models are compared with each other to determine which technique has better performance for solving the target problem (Rodriguez-Galiano et al., 2015; Sundays, 2012). Here the performance metrics obtained from the application of the trained models in the test base are generated. The metrics commonly used for the evaluation of regression models are: Rsquared, RMSE and MAE. Rsquared indicates the correlation between

ISSN: 2236-269X
DOI: 10.14807/ijmp.v12i8.1481

observed result values and the values predicted by the model. In this case, the higher the Rsquared the better the model. The RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) are two metrics used to measure the prediction error of each model. In this case, the metric is evaluated considering that the lower the RMSE and the MAE, the better the model.

3. METHODOLOGICAL PROCEDURES

Seeking to analyze the proposed subject, the method used in this research consists of an applied case study. The data was extracted from January 2018 to July 2019 in order to predict the number of medical appointments per day in the health sector in the region of Porto Alegre / RS - Brazil and propose metrics to identify potential anomalies in this indicator, according to the steps described in the Figure 1.

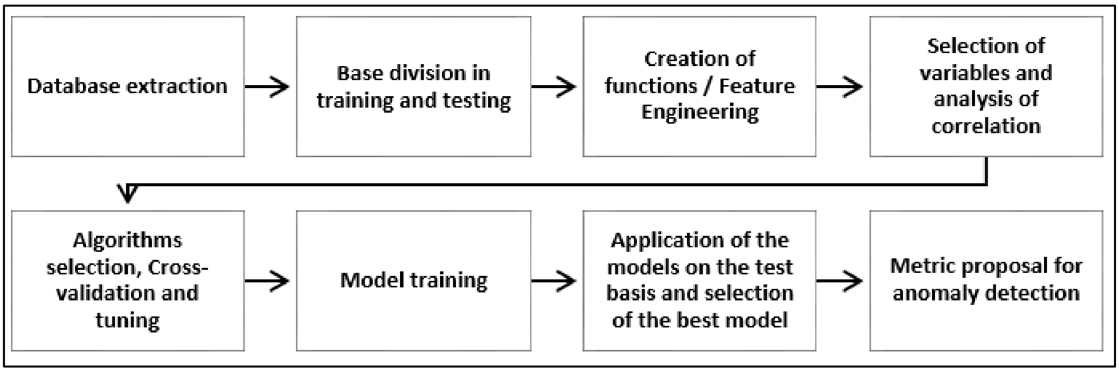
Database extraction
Base division in training and testing
Creation of functions / Feature Engineering
Selection of variables and analysis of correlation
Algorithms selection, Cross-validation and tuning
Model training
Application of the models on the test basis and selection of the best model
Metric proposal for anomaly detection

Figure 1: Case study steps

As shown in Figure 1, this research begins with the data extraction, described in item 3.1. and proceeds with the division of the base in training and testing, also detailed in item 3.1.; continuing with the creation of functions and feature engineering, described in item 3.2.; moving on to the variable selection and analysis of the correlation, also detailed in item 3.2.; after this stage, the algorithms selection, cross-validation and tuning, described in item 3.3.; and then the models were trained and the models were applied to the test base to select the best model, also described in item 3.3.; and finally, the proposed metric for anomaly detection was elaborated, according to item 3.4.

The software used to prepare the database and create the Machine Learning model whole process was the software R. The databases were saved in a repository using the extension .csv.

3.1. Base extraction and division

The object of this study consists in the creation of a Machine Learning model capable of predicting the response variable: number of medical appointment per day in the field of supplementary health in the region of Porto Alegre / RS, and thus creating anomaly detection metrics.

A database was used with 563 observations provided by a supplementary health company that operates in the region of Porto Alegre / RS, considering all dates with registered medical appointments that are characterized as emergency or elective referring to the period of January 2018 to July 2019, any identifying information was disregarded in order to ensure data anonymity. This set was selected according to the objective of the study, which is the prediction of the indicator number of medical appointment per day.

The division of the original database in training and testing aims to reserve a portion of the observations to simulate the real conditions for the implementation of the trained models. The training data was composed for the period from January 1, 2018 to December 31, 2018, representing 63.6% of the original base. The test data was created from the remaining information, that is, the period from January 1, 2019 to July 31, 2019, representing the remaining 36.4%.

3.2. Creation of functions and feature engineering

Feature Engineering aimed to create new variables from the date variable. In order to make the process of creating variables reproducible, two functions written in R language were created: “var_date” and “var_lag_diff”, which are exposed in Appendix A and Appendix B.

The "var_date" function returns a data frame with the addition of six new variables, which are: "bus_day", "week", "is_bus", "dist_holiday" and "week_month". The variable “bus_day” is a categorical variable to classify dates as being the first working day of the month, second working day, for example, the date 7/1/2018 was a Sunday, so the first working day of the month will be 7/2/2018; “Week” is a categorical variable with seven categories representing the names of the seven days of the week (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday); “Is_bus” which is a binary variable, that is, 1 for dates corresponding to business days and 0 for dates corresponding to non-business days; “Dist_holiday” is a continuous variable that measures the distance in days to the nearest holiday, with zero being the date that represents a holiday; “Week_month” is a categorical variable that indicates which week of the month a specific date belongs to. The base of holidays used in the function was

ISSN: 2236-269X
DOI: 10.14807/ijmp.v12i8.1481

obtained from the “bizdays” package (v1.0.6) function “holidaysANBIMA” available on CRAN for software R.

The "var_lag_diff" function returns a data frame with the addition of thirteen new variables, which are: "lag1", "lag2", "lag3", "lag4", "lag5", "lag6", "lag7", "lag14 ",“ Lag30 ",“ diff_lag7lag14 ",“ diff_lag1lag2 ",“ diff_lag1lag30 ”and“ diff_lag1lag7 ”. The “lagX” variables represent the value of the response variable with X days gaps, for example, “lag2” is the value of the response variable two days ago, and “lag30” is the value of the response variable 30 days ago. The “diff_lagXlagY” variables represent the difference between the “lagX” variable and the “lagY” variable.

In order to identify linear relationships between the lag variables and the response variable, Pearson's linear correlation calculation was performed two by two, as shown in Figure 2. All variables whose Pearson correlation coefficient module was greater or equal to 0.5 were considered satisfactory. After selecting variables by analyzing the Pearson correlation, it was still necessary to verify whether this composition is in fact the best possible, for this reason the Stepwise Backward technique was applied, using the lowest MAE (Mean Absolute Error) value as the selection metric.

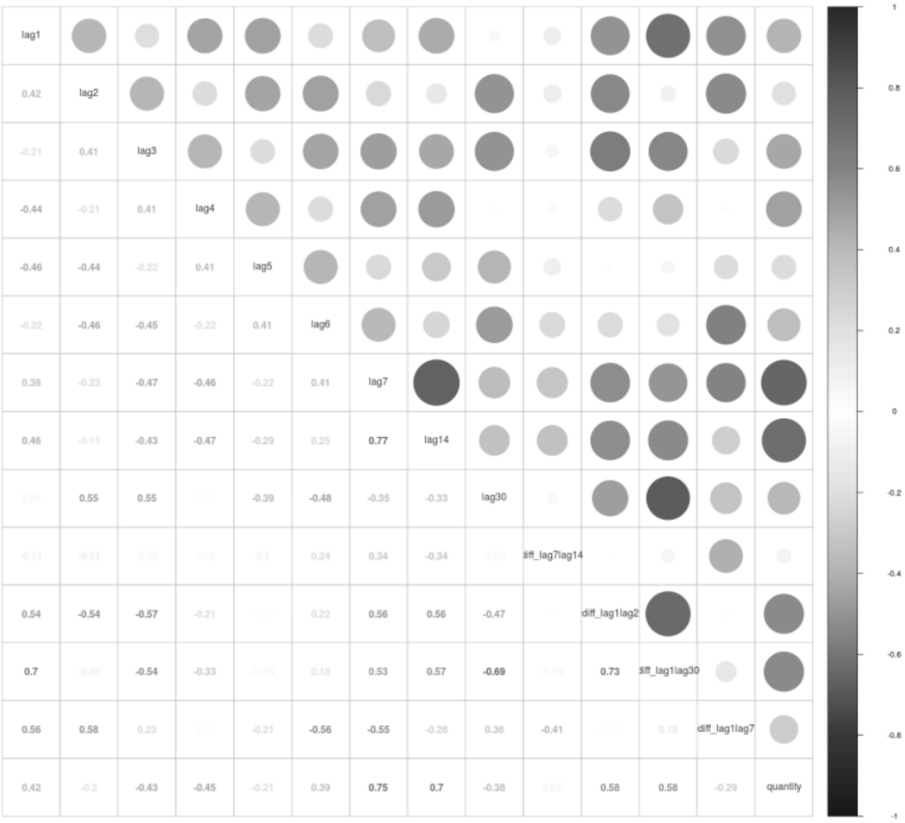

Figure 2: Pearson correlation between variables

3.3. Algorithms selection, cross-validation and tuning

The Random Forest, Linear Regression and Neural Network algorithms were selected due to the nature of the response variable, which, because it is continuous, limits the choice of algorithms to the subgroup of techniques defined in the set of supervised learning regression problems, these being the least complex algorithms for implementation and widely used in the literature to solve similar problems. As a cross-validation method, 5-fold was used, and the tuning grids for the Random Forest algorithm were 500 and 1000 for the ntree parameter, which is the number of trees, and 2, 4, 8 and 10 for the parameter mtry representing the number of branches. For the Neural Network algorithm, the tuning grids used were size and decay, size is the number of units in hidden layer and decay is the regularization parameter used to avoid overfitting.

The Random Forest, Linear Regression and Neural Network algorithms were trained on the training data so that they could later be applied on the test data to select the best model according to the MAE. In total, 1 Linear Regression model, 10 Random Forest models were trained, one for each hyperparameter combination, and 50 Neural Network models, also considering the hyperparameter combination. At the end of the training stage, the Random Forest, Linear Regression and Neural Network model was selected, whose hyperparameters represented a better MAE, to be applied to the test base.

In order to evaluate and select the best algorithm, among Random Forest, Linear Regression and Neural Network, the selected models were applied to the test data. The methodologies were compared according to their MAE values and an algorithm was chosen that obtained superior performance.

3.4. Metric proposal for the detection of anomalies

After the implementation of the higher performance algorithm, it was possible to compare the real value with the predicted value. Therefore, a metric to detect anomalies was proposed, based on the occasion when the real value is one or two standard deviations higher than the predicted value, indicating anomalous behavior, serving as a warning of possible fraud or irregular behavior.

The definition of alert and anomaly was based on the understanding of the distribution of the difference between the observed and predicted values of the training base, as shown in Figure 3. The calculation of the standard deviation as a metric was used to measure the dispersion of the curve, and the definition of the warning and anomaly points. The standard deviation of the differences between observed and predicted was equal to 244.55.

When analyzing the density of the differences, it was noted that 93% of the observations are less than one standard deviation, that is, it is expected that only 7% of the cases have a difference between observed and predicted greater than 244.55 medical appointments, represented to the right of the yellow line in Figure 3. Likewise, it is noted that 98.5% of the data are less than twice the standard deviation, that is, 1.5% of the observations have a difference between observed and predicted greater than 489.10 medical appointments, represented to the right of the red line in Figure 3.

Thus, it was defined that the detection of suspicious behavior will be through two situations: points above one deviation will be considered "Alerts" and will serve as a warning about possible irregularities, and points above two deviations will be considered "Anomalies", and should have immediate attention.

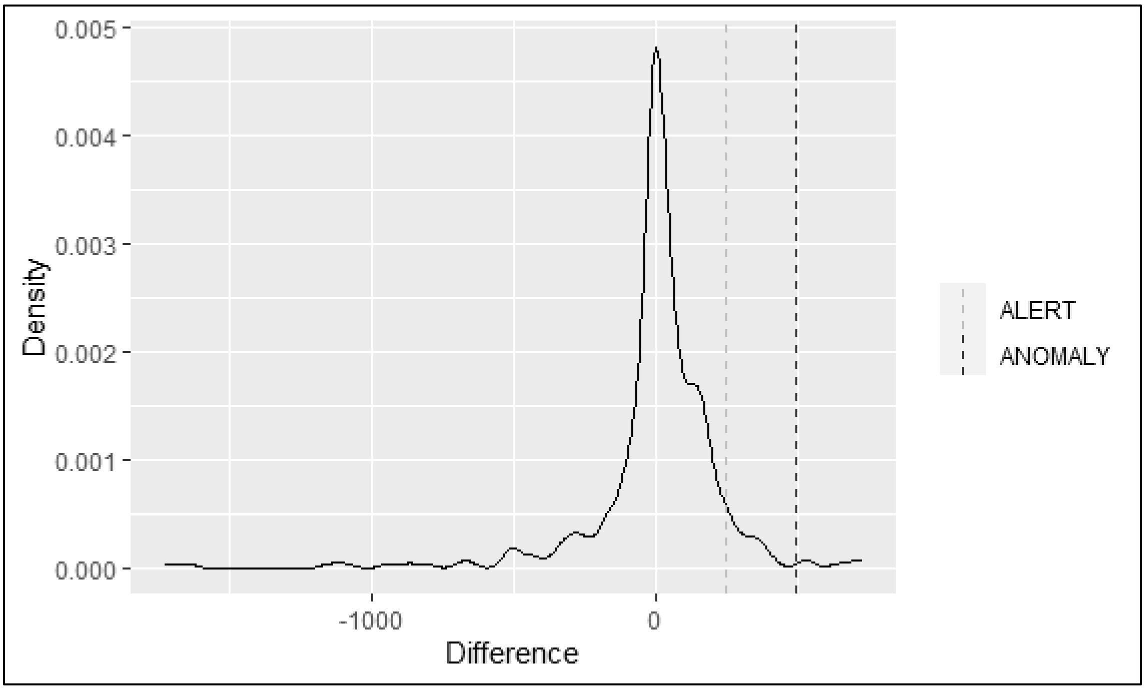

Figure 3: Distribution of the difference between the observed value and the predicted value of the training data

4. RESULTS AND DISCUSSION

From the application of the presented methodology, it was possible to create a Machine Learning model, with sufficient accuracy to predict the number of medical appointment per day in the area of supplementary health in the region of Porto Alegre / RS. According to the methodology, the applied steps were: database extraction, division of the base in training and testing, creation of functions and feature engineering, variable selection and analysis of correlation, choice of algorithms with cross-validation and tuning, training of models, application of the models on the test data, selection of the best model and, finally, the metric for anomaly detection was proposed.

The initial database contained two variables, namely the number of medical appointment grouped by date of completion. After dividing the base into training and testing, the “var_date” function was applied to create new variables. Table 1 shows a sample of the database after the creation of the new variables with the application of the “var_date” function.

Table 1: Variables after applying the “var_date” function

Variable	Format	Example
date_register	Date, format	"2018-01-01" "2018-01-02" "2018-01-03" "2018-01-04" . . .
quantity	int	1 1168 2104 2380 1478 68 3 2506 2825 2765 . . .
bus_day	num	0 1 2 3 4 5 6 7 8 9 . . .
week	chr	"Monday Tuesday Wednesday Thursday Friday Saturday" . . .
is_bus	num	0 1 1 1 1 1 0 1 1 1 . . .
dist_holiday	num	0 1 2 3 4 5 6 7 8 9 . . .
week_month	int	6 1 1 1 1 1 1 2 2 2 . . .

A second function was created in order to further increase the number of explanatory variables and thereby also generate a greater understanding of the response variable. Table 2 shows a sample of the database after the creation of the new variables with the application of the “var_lag_diff” function.

Table 2: Variables after applying the “var_lag_diff” function

Variable	Format	Example
date_register	Date, format:	"2018-01-31" "2018-02-01" "2018-02-02" "2018-02-03" . . .
quantity	int	2539 2215 123 25 2152 2478 2359 2375 1180 30 . . .
bus_day	num	30 0 1 2 4 5 6 7 8 9 . . .
week	chr	"Monday Tuesday Wednesday Thursday Friday Saturday" . . .
is_bus	num	1 1 1 1 1 1 1 1 1 1 . . .
dist_holiday	num	12 11 10 9 7 6 5 4 3 2 . . .
week_month	int	5 5 1 1 2 2 2 2 2 2 . . .
lag1	int	NA 2539 2215 123 25 2152 2478 2359 2375 1180 . . .
lag2	int	NA NA 2539 2215 123 25 2152 2478 2359 2375 . . .
lag3	int	NA NA NA 2539 2215 123 25 2152 2478 2359 . . .
lag4	int	NA NA NA NA 2539 2215 123 25 2152 2478 . . .
lag5	int	NA NA NA NA NA 2539 2215 123 25 2152 . . .
lag6	int	NA NA NA NA NA 2539 2215 123 25 2152 . . .
lag7	int	NA NA NA NA NA NA NA 2539 2215 123 . . .
lag14	int	NA NA NA NA NA NA NA NA NA NA . . .
lag30	int	NA NA NA NA NA NA NA NA NA NA . . .
diff_lag1lag2	int	NA NA -324 -2092 -98 2127 326 -119 16 -1195 . . .
diff_lag1lag7	int	NA NA NA NA NA NA NA -180 160 1057 . . .
diff_lag7lag14	int	NA NA NA NA NA NA NA NA NA NA . . .
diff_lag1lag30	int	NA NA NA NA NA NA NA NA NA NA . . .

Making use of variables "date_register" and "quantity," it was possible to create 18 new variables, shown in table 2. These variables were subjected to selection techniques, the selection being performed primarily by the Pearson correlation coefficient. As shown in Figure

2, from the criterion, correlation module greater than or equal to 0.5, the selected variables were: “lag7”, “lag14”, “diff_lag1lag2”, “diff_lag1lag30”. It is worth noting that the variables “lag7” and “lag14” have a high correlation with each other (0.77), as well as the variables “diff_lag1lag2” and “diff_lag1lag30” (0.73). In order to avoid possible multicollinearity problems, only one “lag” and “diff” variable was considered, the “lag7”and “diff_lag1lag2”. Both presented the bigger correlation in relation to the response variable, 0.75 and 0.58 respectively.

From this stage, the explanatory variables "bus_day", "week", "is_bus", "dist_holiday", "week_month", "lag7", "diff_lag1lag2", compose the equation to be used to predict the indicator of the number of medical appointment per day. However, to improve the variable selection, the Stepwise Backward technique was applied, using the lowest MAE as the selection metric. The set of variables that obtained the lowest MAE value (237.79) using the Stepwise Backward technique were: “week”, “is_bus”, “week_month” and “diff_lag1lag2”. Therefore, in Equation 1, the final equation to be applied to all algorithms is presented.

$$quantity = week + is_{bus} + week_{month} + diff_{lag1lag2} \qquad [1]$$

With the predictor selection phase completed, training began on the two algorithms presented in the methodology, which were compared according to the MAE. In the training phase, the Random Forest algorithm with hyperparameters mtry equal to 6 and ntree equal to 1000 showed a subtle gain in relation to the other models, as can be seen in Table 3. However, the Neural Network algorithm performed much lower when compared to the Linear Regression and Random Forest algorithms, for this reason the Neural Network algorithm was not applied to the test data.

Table 3: MAE comparison of models applied to the training base

Algorithm	Ntree	mtry	size	decay	RMSE	R²	MAE	RMSE SD	R² SD	MAE SD
Linear Regression	NA	NA	NA	NA	414.75	0.89	242.45	71.83	0.04	16.78
Random Forest	500	2	NA	NA	487.63	0.88	360.54	41.67	0.03	31.85
Random Forest	500	4	NA	NA	415.52	0.89	235.36	52.32	0.03	24.55
Random Forest	500	6	NA	NA	418.14	0.89	228.06	57.47	0.03	22.46
Random Forest	500	8	NA	NA	421.76	0.89	228.31	59.25	0.03	21.02
Random Forest	500	10	NA	NA	433.98	0.88	232.74	61.55	0.03	21.84
Random Forest	1000	2	NA	NA	486.21	0.88	359.16	41.72	0.03	31.71
Random Forest	1000	4	NA	NA	416.39	0.89	235.41	53.21	0.03	25.63
Random Forest	**1000**	**6**	**NA**	**NA**	**416.41**	**0.89**	**227.19**	**57.15**	**0.03**	**21.00**
Random Forest	1000	8	NA	NA	422.39	0.89	228.24	59.49	0.03	21.08
Random Forest	1000	10	NA	NA	434.56	0.88	232.44	62.22	0.03	22.89
Neural Network	NA	NA	1	0.1	2219.32	0.23	1819.36	18.98	0.16	34.60
Neural Network	NA	NA	1	0.2	2219.32	0.13	1819.36	18.98	0.06	34.60
Neural Network	NA	NA	1	0.3	2219.32	0.17	1819.36	18.98	0.02	34.60

ISSN: 2236-269X
DOI: 10.14807/ijmp.v12i8.1481

Neural Network	NA	NA	1	0.4	2219.32	0.16	1819.36	18.98	0.08	34.60
Neural Network	NA	NA	1	0.5	2219.32	0.13	1819.36	18.98	0.06	34.60
Neural Network	NA	NA	2	0.1	2219.32	0.10	1819.36	18.98	0.08	34.60
Neural Network	NA	NA	2	0.2	2219.32	0.13	1819.36	18.98	0.06	34.60
Neural Network	NA	NA	2	0.3	2219.32	0.12	1819.36	18.98	0.09	34.60
Neural Network	NA	NA	2	0.4	2219.32	0.14	1819.36	18.98	0.06	34.60
Neural Network	NA	NA	2	0.5	2219.32	0.14	1819.36	18.98	0.08	34.60
Neural Network	NA	NA	3	0.1	2219.32	0.27	1819.36	18.98	0.25	34.60
Neural Network	NA	NA	3	0.2	2219.32	0.15	1819.36	18.98	0.05	34.60
Neural Network	NA	NA	3	0.3	2219.32	0.18	1819.36	18.98	0.12	34.60
Neural Network	NA	NA	3	0.4	2219.32	0.22	1819.36	18.98	0.20	34.60
Neural Network	NA	NA	3	0.5	2219.32	0.16	1819.36	18.98	0.08	34.60
Neural Network	NA	NA	4	0.1	2219.32	0.13	1819.36	18.98	0.06	34.60
Neural Network	NA	NA	4	0.2	2219.32	0.17	1819.36	18.98	0.17	34.60
Neural Network	NA	NA	4	0.3	2219.32	0.32	1819.36	18.98	0.30	34.60
Neural Network	NA	NA	4	0.4	2219.32	0.10	1819.36	18.98	0.07	34.60
Neural Network	NA	NA	4	0.5	2219.32	0.27	1819.36	18.98	0.23	34.60
Neural Network	NA	NA	5	0.1	2219.32	0.12	1819.36	18.98	0.09	34.60
Neural Network	NA	NA	5	0.2	2219.32	0.18	1819.36	18.98	0.12	34.60
Neural Network	NA	NA	5	0.3	2219.32	0.07	1819.36	18.98	0.07	34.60
Neural Network	NA	NA	5	0.4	2219.32	0.11	1819.36	18.98	0.07	34.60
Neural Network	NA	NA	5	0.5	2219.32	0.16	1819.36	18.98	0.11	34.60
Neural Network	NA	NA	6	0.1	2219.32	0.14	1819.36	18.98	0.04	34.60
Neural Network	NA	NA	6	0.2	2219.32	0.17	1819.36	18.98	0.24	34.60
Neural Network	NA	NA	6	0.3	2219.32	0.16	1819.36	18.98	0.09	34.60
Neural Network	NA	NA	6	0.4	2219.32	0.23	1819.36	18.98	0.20	34.60
Neural Network	NA	NA	6	0.5	2219.32	0.18	1819.36	18.98	0.16	34.60
Neural Network	NA	NA	7	0.1	2219.32	0.10	1819.36	18.98	0.08	34.60
Neural Network	NA	NA	7	0.2	2219.32	0.13	1819.36	18.98	0.09	34.60
Neural Network	NA	NA	7	0.3	2219.32	0.11	1819.36	18.98	0.08	34.60
Neural Network	NA	NA	7	0.4	2219.32	0.33	1819.36	18.98	0.26	34.60
Neural Network	NA	NA	7	0.5	2219.32	0.21	1819.36	18.98	0.10	34.60
Neural Network	NA	NA	8	0.1	2219.32	0.25	1819.36	18.98	0.22	34.60
Neural Network	NA	NA	8	0.2	2219.32	0.14	1819.36	18.98	0.07	34.60
Neural Network	NA	NA	8	0.3	2219.32	0.09	1819.36	18.98	0.08	34.60
Neural Network	NA	NA	8	0.4	2219.32	0.26	1819.36	18.98	0.21	34.60
Neural Network	NA	NA	8	0.5	2219.32	0.08	1819.36	18.98	0.06	34.60
Neural Network	NA	NA	9	0.1	2219.32	0.10	1819.36	18.98	0.08	34.60
Neural Network	NA	NA	9	0.2	2219.32	0.12	1819.36	18.98	0.11	34.60
Neural Network	NA	NA	9	0.3	2219.32	0.12	1819.36	18.98	0.11	34.60
Neural Network	NA	NA	9	0.4	2219.32	0.09	1819.36	18.98	0.09	34.60
Neural Network	NA	NA	9	0.5	2219.32	0.29	1819.36	18.98	0.27	34.60
Neural Network	NA	NA	10	0.1	2219.32	0.13	1819.36	18.98	0.07	34.60
Neural Network	NA	NA	10	0.2	2219.32	0.11	1819.36	18.98	0.05	34.60
Neural Network	NA	NA	10	0.3	2219.32	0.20	1819.36	18.98	0.17	34.60
Neural Network	NA	NA	10	0.4	2219.32	0.16	1819.36	18.98	0.17	34.60
Neural Network	NA	NA	10	0.5	2219.32	0.08	1819.36	18.98	0.08	34.60

After defining the best combination of hyperparameters for the Random Forest algorithm, this, together with the Linear Regression, was applied to the test data, in order to identify which algorithm would present a better performance when simulated real implementation conditions. The Random Forest model performed better than the Linear Regression model when applied to the test base, as shown in Table 4.

Table 4: MAE comparison of the models applied to the test base

Algorithm	*ntree*	*mtry*	MAE
Random Forest	**1000**	**6**	**182.89**
Linear Regression	NA	NA	216.44

It can also be seen from Figures 3 and 4, which compare the real values (in blue) with the predicted values (in red), that the selected model was able to predict the number of medical appointment in a satisfactory way both in the training data and on the test data.

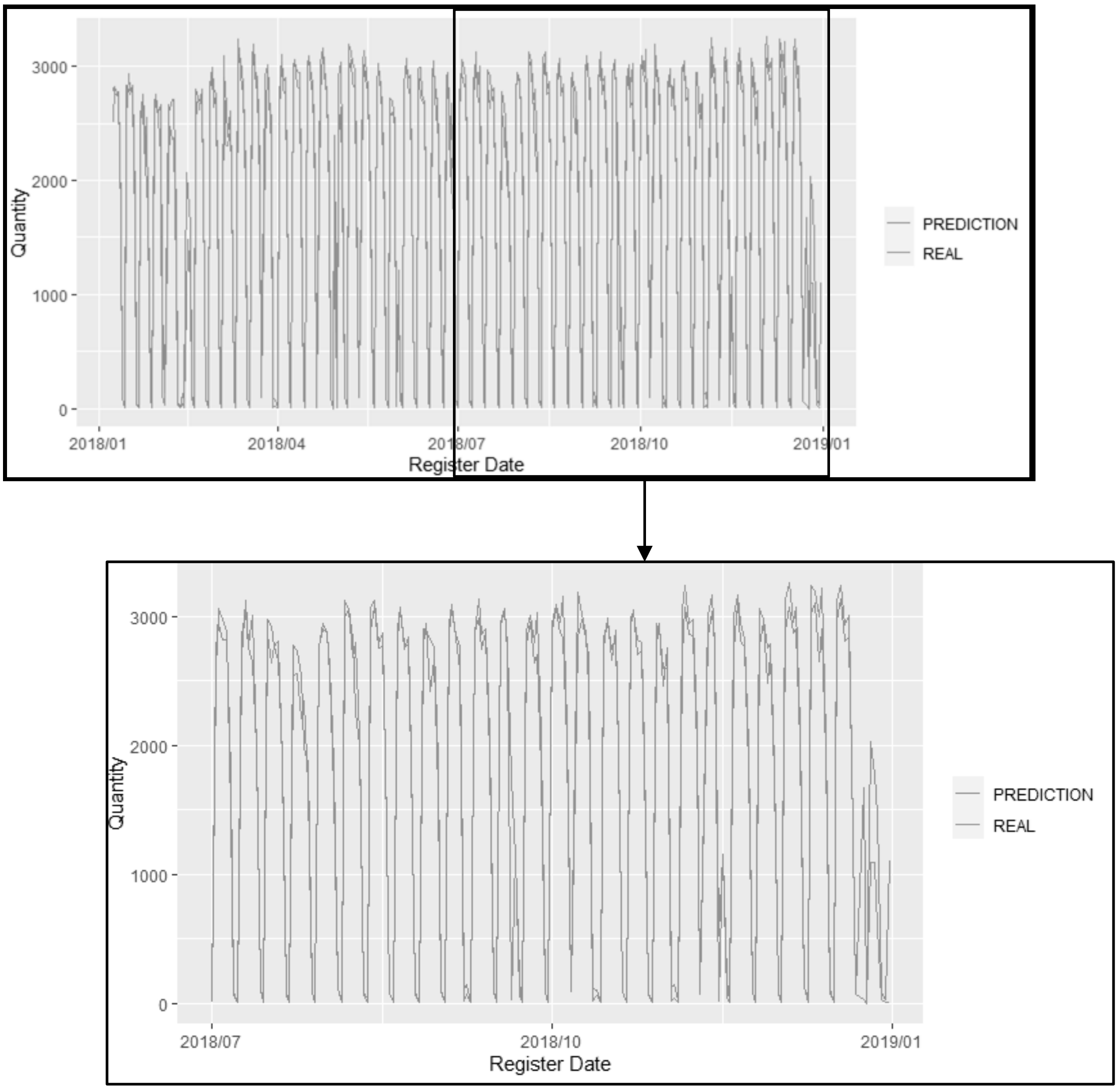

Figure 4: Comparison of the prediction of the selected model with the real value in the training base

After the prediction of the variable number of medical appointment, the anomalies and alerts identification metric that is defined by the rule was applied: If the real value is greater than the predicted value plus a standard deviation, the occurrence will be defined as an alert, if the real value is greater than the predicted value plus two standard deviations, the occurrence will be defined as anomalous.

According to Figures 5 and 6, nine anomaly situations can be identified in the test data, on March 25, 2019, March 27, 2019, March 29, 2019, April 26, 2019, May 2, 2019 , May 9, 2019, May 24, 2019, June 24, 2019 and July 29, 2019, in addition to thirty-eight alert points during the period from January to July. The prior identification of these points, as well as the investigation of the reason for such behaviors, can bring gains both in the identification of frauds, as well as in the opportunity to improve the provision of the service through a better distribution of resources.

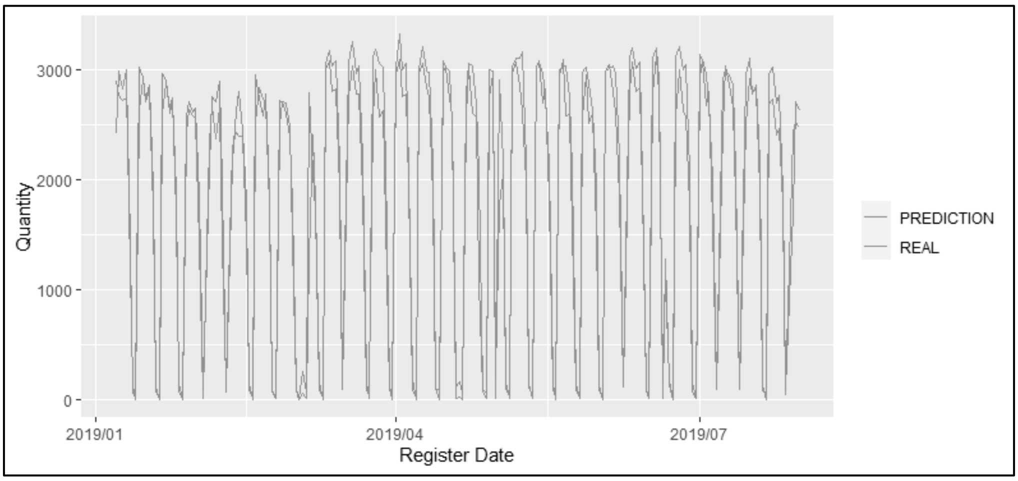

Figure 5: Comparison of the prediction of the selected model with the real value in the test base

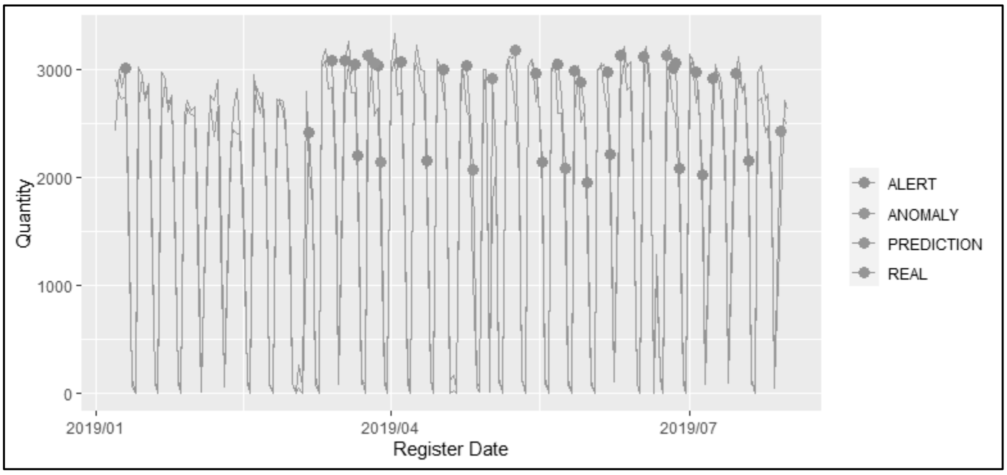

Figure 6: Identification of anomalous occurrence through standard deviation

5. CONCLUSION

The study provided a broad theoretical and applied understanding of important steps for the creation of a Machine Learning model (collecting and preparing the data, selecting the variables, choice of algorithm, selection of hyper parameters, training of the model and evaluation of results).

The creation of Feature Engineering was an essential step to understanding the behavior of the data, and combined with the development of auxiliary functions made it possible to reproduce in a more efficient and fast way this very important part of this Machine Learning process. Another essential point for the creation of a high performance model was the step of selecting the variables, because using the techniques of analysis of the correlations followed by the Stepwise Backward selection, it was possible to identify the predictors with the greatest impact on the response variable preventing overfitting.

With the functions created, the implementation, maintenance, or replication of the methodology as a whole becomes more simplified, which is easily replicable for other similar studies whose goals are the prediction of a continuous variable using as features any kind of time frames (daily, monthly, annually, etc.).

The indicator of the number of medical appointments per day is an important indicator for the area of supplementary health because this kind of KPI is strongly correlated to the operational costs and being able to predict it improves the decision making. With the Random Forest model selected and the proposed metric for detecting anomalies through the standard deviation, it was possible not only to predict the KPI results with almost 90% of accuracy, but also to compare the prediction with the real value producing alerts of anomalous occurrences. This can enable the investigation of possible problems, or new demands, in order to improve health services.

REFERENCES

Araújo, F. H. D., Santana, A. M., & Santos Neto, P. A. (2015). ma Abordagem Influenciada por Pré-processamento para Aprendizagem do Processo de Regulação Médica. **Journal of Health Informatics**, 7(1).

Bishop, C. M. (2006). Pattern recognition and machine learning. **Springer**.

Burrell, J. (2016). How the machine ‘thinks’: Understanding opacity in machine learning algorithms**. Big Data & Society**, 3(1).

Domingos, P. M. (2012). A few useful things to know about machine learning. **Commun. acm**, 5(10), 78-87.

García, E. et al. (2007). Drawbacks and solutions of applying association rule mining in learning management systems. **Proceedings of the International Workshop on Applying Data Mining in e-Learning, 1**3-22.

Garla, V. N., & Brandt, C. (2012). Ontology-guided feature engineering for clinical text classification. **Journal of biomedical informatics**, 45(5), 992-998.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). **An introduction to statistical learning**, 112, 18. New York, Springer.

Kraska, Tim et al. (2013). MLbase: A Distributed Machine-learning System. **CIDR**. 2.1.

Lee, P. P. Y. et al. (2019). **Interactive interfaces for machine learning model evaluations**. U.S. Patent n. 10,452,992.

Marsland, Stephen. (2014). Machine learning: an algorithmic perspective. **Chapman and Hall/CRC**.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning. **MIT press**.

Murdoch, W. J. et al. (2019). Definitions, methods, and applications in interpretable machine learning. **Proceedings of the National Academy of Sciences**, 116(44), 22071-22080.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. **Encyclopedia of database systems**, 532-538.

Rodriguez-Galiano, V. et al. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. **Ore Geology Reviews**, 71, 804-818.

Vinyals, O., Dean, J. A., & Hinton, G. E. (2019). **Training distilled machine learning models**. U.S. Patent n. 10,289,962.

APPENDIX A - 'VAR_DATE' FUNCTION (R CODE)

```
# Required packges
require(dplyr)
require(bizdays)
require(purrr)
require(lubridate)

#--- FUNCTION ---#
var_date <- function(db, col_name_date, format_date = "%d/%m/%Y", feriados = bizdays::holidaysANBIMA)
{

  # auxiliar function for calculate the distance to a holiday
  dist_holiday<-function(x, y = feriados)n
  {
    # distance between two dates
    menor_dist <- purrr::map2(x, y, difftime)
    db_min <- min(abs(unlist(menor_dist)))
    return(db_min)
  }

  db <- db %>%
    mutate(
      dt_ok = as.Date(get(col_name_date), format = format_date),
      first_day = ymd(format(dt_ok, "%Y-%m-01")),
      week = weekdays(dt_ok),
      is_bus = case_when(
        week == "sábado" | week == "domingo" | dt_ok %in% feriados ~ 0,
        TRUE ~ 1
      ),
      dist_holiday = unlist(purrr::map(.x = dt_ok, .f = dist_holiday)),
      week_month = stringi::stri_datetime_fields(get(col_name_date), tz = 'Etc/GMT-3')$WeekOfMonth
    )

  # creating the business day of the month
  temp <- db %>%
    filter(is_bus == 1) %>%
    mutate(
      var_temp = 1
    ) %>%
    group_by(first_day) %>%
    mutate(
      bus_day = cumsum(var_temp)) %>%
    ungroup() %>%
    select(dt_ok, bus_day)

  db <- db %>%
    left_join(temp, by = c("dt_ok" = "dt_ok")) %>%
    mutate(bus_day = if_else(is.na(bus_day),0,bus_day))


  return(db)

}
```

APPENDIX B - 'VAR_LAG_DIFF' FUNCTION (CODE IN R)

```
# Required packges
require(dplyr)
require(lubridate)
require(corrplot)

#--- FUNCTION ---#
var_lag_diff <- function(db, col_name_date, target_var, reference_value = 0.2)
{
  db <- db %>%
    arrange(get(col_name_date)) %>%
    mutate(
      lag1 = lag(get(target_var), 1),
      lag2 = lag(get(target_var), 2),
      lag3 = lag(get(target_var), 3),
      lag4 = lag(get(target_var), 4),
      lag5 = lag(get(target_var), 5),
      lag6 = lag(get(target_var), 6),
      lag7 = lag(get(target_var), 7),
      lag14 = lag(get(target_var), 14),
      lag30 = lag(get(target_var), 30),
      diff_lag7lag14 = lag7 - lag14,
      diff_lag1lag2 = lag1 - lag2,
      diff_lag1lag30 = lag1 - lag30,
      diff_lag1lag7 = lag1 - lag7
      )
  aux <- db %>%
    filter(!is.na(lag30)) %>%
    select(
      lag1,
      lag2,
      lag3,
      lag4,
      lag5,
      lag6,
      lag7,
      lag14,
      lag30,
      diff_lag7lag14,
      diff_lag1lag2,
      diff_lag1lag30,
      diff_lag1lag7,
      target_var)

  # Function return list
  list_return <- as.list(NULL)

  # return 1
  ## plot das correlacoes dos lags e diffs e a variavel target
  correl <- cor(aux)
  list_return[[1]] <- correl

  # return 2
  ## data frame with lag or diff variables that were bigger then predetermined amount

  correl <- as.data.frame(correl)
  nome_linha <- row.names(correl)

  correl <- correl %>%
    mutate(linha = nome_linha) %>%
    filter(abs(get(target_var)) >= reference_value)

  selection <- c(correl$linha, 'linha')
  tirar <- colnames(correl)[!colnames(correl) %in% selection]

  list_return[[2]] <- db

  db <- db %>%
    select(-tirar)

  list_return[[3]] <- db

  return(list_return)
}
```