Unattended Medical Appointments: Analysis and Prediction

Bowen Sebastiano[1], Dal Tio Leonardo[2], Fratello Mattia[1], Govi David[1], Merola Paolo[1]

Abstract - Every country with a functioning public health system reached the same conclusion: it is very expensive to maintain one. It is therefore highly desirable to avoid inefficiencies and waste. An effective method of limiting them is to minimize the impact of patients that set medical appointments but do not attend. A missed appointment creates Quality of Service (QoS) issues regarding the entire public-health service: QoS wise, the presence of unpredicted gaps in the daily schedule of a medical center generates longer than necessary waiting lists; financially speaking, work time of trained medical operators (particularly, of a Doctor) is of a worth no public health system can afford to lose. To face this issue, an increasing number of public institutions are releasing open datasets, in order to allow Data Scientists to perform their magic.
The following report focuses on the process of manipulating a dataset of medical appointments and predicting no-shows. The dataset used in this project was made available by the city of Vitoria (Brazil) on the Kaggle on-line platform. Six different classification algorithms have been tested for this purpose, with the best two used for actual classification.

[1]University of Milan-Bicocca, MSc - Data Science
[2]University of Milan-Bicocca, MSc - Theory and Technologies of Communication

CONTENTS

I. Introduction

One of the biggest problems that Public Health-Care systems have is long waiting times for medical appointments, even for the most important and pressing ones. Public Administrations, in order to limit said problem, have decided to penalize patient no-shows with a fine. An example of this can be found in the Italian city of Asolo (Veneto) and its USL 8 ("Local Health Unit no.8"); here, the Local Health Unit decreed that patients that did not attend appointments, not only would have to pay the unattended medical performance, but they would also be flagged by the Italian Tax Agency for fines, and further tax burdens, as well as being black-listed as "Negligent".
"A pensar mal se fa pec, ma se indovina sempre" (by thinking bad [about someone] you sin, but you always guess right). This old venetian adage, although making a comeback nowadays, is not always an accurate method of predicting no-shows. As a matter of fact not everybody is forgetful or negligent, but could well be influenced by other unknown factors that might impede attendance and result in a no-show. This hypothesis will be studied and the possible features processed and analyzed with Machine Learning techniques. Our aim was to predict, with an acceptable accuracy, the Class attribute "appointment no-shows". For this specific project we mostly used Knime, an open-source software, that allowed us to develop a complete Machine Learning Project Workflow and display it visually (see Appendix, Fig. 9).

II. Dataset and Descriptive Statistics

A. Dataset Overlook

The main dataset, downloaded from the Kaggle platform, is made of 14 attributes for a total of about 110,527 occurrences. Here are the attributes and their characteristics:

- *PatientId: numeric-nominal*
  Univocally identifies the patient[1]
- *AppointmentID: numeric-nominal*
  Univocally identifies the appointment
- *Gender: string-binary*
  Gender of the patient
- *ScheduledDay: dateTime-interval*
  Day in which the appointment was taken
- *AppointmentDay: dateTime-interval*
  Day the appointment is set for
- *Age: numeric-interval*
  Age of the patient
- *Neighborhood: string-nominal*
  Neighborhood of Vitoria where the hospital is located
- *Scholarship: numeric-binary*
  Indicates whether the Patient has a Scholarship;[2]
- *Hypertension: numeric-binary*
  A significant medical condition
- *Diabetes: numeric-binary*
  A significant medical condition
- *Alcoholism: numeric-binary*
  A significant medical condition
- *Handicap: numeric-ordinal*
  A significant medical condition
- *SMS-received: numeric-binary*
  Whether the patient received a reminder for the appointment
- *No-show: string-binary*
  Whether the patient showed up to the appointment or not

[1]We later find out that multiple appointments per patient is possible.
[2]Indicates whether the family of the Patient receives a subsidy.

Although we have quite some information to start with, it is sometimes useful to increase the amount of information by integrating content from other sources (increased *Variability*). For this reason, the main dataset has been merged with a second one containing the data collected by the meteorological station in the Vitoria Airport, later stored on Weather Underground. It gives us a possible piece of information that we hypothesized might have had an influence on our patients behavior. The two datasets have been joined by date. The weather dataset presented itself as follows:

- *Timestamp: dateTime-interval*
  Date and Time of the meteorological record
- *Rain: numeric-binary*
  Presence or absence of specific meteorological event
- *Mist: numeric-binary*
  Presence or absence of specific meteorological event
- *Storm: numeric-binary*
  Presence or absence of specific meteorological event

B. Preliminary Descriptive Statistics

Before proceeding with Feature Engineering and ML Algorithm implementation, we needed to first analyze the datasets with Descriptive Statistics in order to better understand the composition and tendencies our data had. Our first results are shown in Fig. 1.

Row ID	Min	Max	Mean	Std. d...	Skewness	Kurtosis	No. miss...	No. NaNs
Age	-1	115	37.089	23.11	0.122	-0.952	0	0
Scholarship	0	1	0.098	0.298	2.699	5.286	0	0
Hipertension	0	1	0.197	0.398	1.522	0.316	0	0
Diabetes	0	1	0.072	0.258	3.316	8.993	0	0
Alcoholism	0	1	0.03	0.172	5.471	27.928	0	0
Handcap	0	4	0.022	0.162	8.274	82.556	0	0
SMS_received	0	1	0.321	0.467	0.767	-1.412	0	0

Fig. 1. Descpritive Statistics before Preprocessing (Appendix 1)

Immediately we noticed two anomalies in our data: firstly, the *Age* feature presented negative values (either data entry error or simply illogical); secondly, the *Handicap* feature shows values greater than 1. In the former case, only 6 occurrences had negative *Age*, so they were removed. In the latter, we made the assumption that a number greater than 1 indicated more than one handicap. We made the call to simplify our model by transforming the Feature from an Integer type, that might have indicated Intensity/Quantity of Handicaps, to a dichotomic/binary type that signaled the presence or not of a Handicap. The rest of the Features were confirmed to be dichotomic/binary, indicating the presence or absence of each feature. We proceeded then to the computation

of the *Spearman Rho* indexes that show collinearity between attributes; this is important because some of the ML algorithms we will later use assume independence between attributes to correctly work. The results are reported in Fig. 2.

Row ID	Gender	Age	Schola...	Hipert...	Diabe...	Alcoh...	Handc...	SMS_r...	No-sh...
Gender	1	-0.107	-0.114	-0.056	-0.033	0.106	0.022	-0.046	-0.004
Age	-0.107	1	-0.09	0.503	0.293	0.102	0.079	0.015	-0.061
Scholarship	-0.114	-0.09	1	-0.02	-0.025	0.035	-0.009	0.001	0.029
Hipertension	-0.056	0.503	-0.02	1	0.433	0.088	0.085	-0.006	-0.036
Diabetes	-0.033	0.293	-0.025	0.433	1	0.018	0.059	-0.015	-0.015
Alcoholism	0.106	0.102	0.035	0.088	0.018	1	0.004	-0.026	-0
Handcap	0.022	0.079	-0.009	0.085	0.059	0.004	1	-0.025	-0.007
SMS_received	-0.046	0.015	0.001	-0.006	-0.015	-0.026	-0.025	1	0.126
No-show	-0.004	-0.061	0.029	-0.036	-0.015	-0	-0.007	0.126	1

Fig. 2. Correlation between attributes before preprocessing (Appendix 2

We noticed moderate correlations in the *Age-Hypertension*, *Age-Diabetes* e *Hypertension-Diabetes* attribute couples and a very weak correlation between *No_show* and *SMS_received*.

III. Preprocessing

A. Data Cleaning

Before We could proceed with proper Machine Learning, we needed to clean and format the dataset in order to properly feed our algorithms.

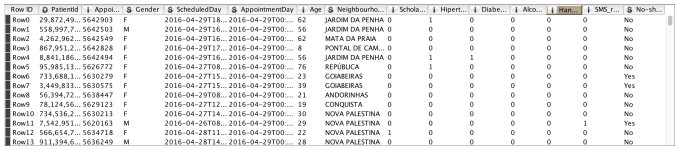

Fig. 3. First look of the main dataset before preprocessing (Appendix 3)

ScheduledDay and *AppointmentDay* were strings. We coded a Java *regex* node in order to cut part of the date (hour) that we felt was not that important.[3] After removing the hours from the time-stamp, we converted these two features into date format. *Neighborhood* has been removed from our model due to a lack of domain information that was making it look useless and *No-show* binarized.

Row ID	Date	Rain	Mist	Storm
Row0	2016-01-01	1	0	0
Row1	2016-01-02	1	0	0
Row2	2016-01-03	1	0	0
Row3	2016-01-04	1	0	0
Row4	2016-01-05	1	0	0
Row5	2016-01-06	0	0	0
Row6	2016-01-07	0	0	0

Fig. 4. Weather data after Preprocessing

Weather related features needed binarization in order to be more easily processable. The feature *Clean* was useless, being equivalent to a lack of weather state in the other features, and was removed for optimization purposes. We now had a much more interpretable series of meteorological data points.

[3]Being present only for *ScheduledDay*, and not for *AppointmentDay*, hour information came to be irrelevant.

B. New Feature Definition

Once all attributes were cleaned, we extracted a bit more information by creating a brand new feature: *Wait_Time*. This feature represent the difference between *AppointmentDay* and *ScheduledDay*. The value is expressed in days (Integer) and gives us a direct idea of the time a patient had to wait from the scheduling day to the moment the appointment actually took place.

C. Aggregation and Item Shuffling

During the preliminary Descriptive Statistics phase, we realized that there were occurrences with unique *AppointmentID* and recurring *PatientId*. This probably meant that there were different patients that had multiple appointments. We hypothesized that the tendency to show up at an appointment might depend on the person, and that there might be some recidivism happening. We than decided to proceed with two instances of our dataset, a "normal" one and an aggregated one. In the former, no further changes were applied, whereas in the latter we aggregated occurrences by *PatientId*; the result is a single occurrence for every patient having as attribute values the average values of his/her appointment. For binary attributes, the variable becomes continuous and signifies a proportion (range 0 to 1). This aggregated dataset instance might give us an idea of whether individual characteristics might influence No-shows.
Last but not least, we shuffled the dataset in order to mix the occurrences and avoid having them in any structured order; this might have happened during the data-mining/collection process, resulting, indeed, in consequences on our Learner Algorithms.

D. Descriptive Statistics after Preprocessing

Since our data structure changed, we proceeded with the same statistical analysis that we already performed in the Exploratory phase, as reported in Chapter II.

Row ID	Min	Max	Mean	Std. d...	Skewn...	Kurtosis	No. m...	No. N...
Age	0	115	37.089	23.11	0.122	-0.952	0	0
Scholarship	0	1	0.098	0.298	2.699	5.285	0	0
Hipertension	0	1	0.197	0.398	1.522	0.315	0	0
Diabetes	0	1	0.072	0.258	3.315	8.992	0	0
Alcoholism	0	1	0.03	0.172	5.47	27.926	0	0
Handcap	0	1	0.002	0.042	23.503	550.409	0	0
SMS_received	0	1	0.321	0.467	0.767	-1.412	0	0
Wait_Time	0	179	10.184	15.255	2.666	11.797	0	0
Rain	0	1	0.154	0.361	1.92	1.686	0	0
Mist	0	1	0.041	0.198	4.64	19.528	0	0
Storm	0	1	0.079	0.27	3.117	7.713	0	0

Fig. 5. Descriptive Statistics after preprocessing (Appendix 5)

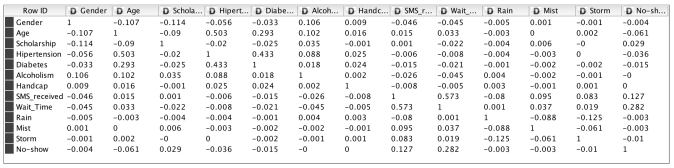

Fig. 6. Correlation between attributes after preprocessing (Appendix 6)

After preprocessing we have the same levels of correlation that we found earlier, except for stronger correlation between

Wait_Time and *SMS_received* and a new, moderately strong correlation between *Wait_time* and *No_show*.

From this preliminary analysis, we have a good clue regarding the explanatory attributes: *Wait-time* and *Sms-received* features have a good chance of explaining the class attribute.

IV. Data Analysis & Machine Learning

A. Evaluated Algorithms

Our first step was deciding which Machine Learning Algorithms were the most appropriate given our data and our objective; features are mostly categorical (of which a large portion is dichotomic), and the class attribute is categorical (dichotomic).

Another factor to be considered was whether to predict No-shows by appointment occurrence or, after aggregation, by patient occurrence; it would have been logical to go directly with the aggregation by *PatientId*, but in order avoid biases on our behalf, we proceeded with both cases. The algorithms we chose were:

- *Decision Tree J48*[4]
- *Random Forest*
- *Naive Bayes*
- *Support Vector Machine with log-loss*
- *Multi-Layer Perceptron with 3 Layers*
- *Bayesian Logistic Regression*

B. Aggregated Dataset Model Performance

1) Validity Measures: In order to objectively assess the validity of each algorithm used, we computed for each of them *Accuracy*, *Recall*, *Precision* and *F-measure*. These valued have been obtained with a holdout with stratified sampling, with 67% of data being used as *Training Set* and the rest as *Test Set*. The metrics are found in Table I.

	Recall	Precision	F-measure	Accuracy
Decision Tree J48	0.097	0.391	0.155	0.768
Random Forest	0.231	0.366	0.283	0.743
Naive Bayes	0	0.111	0	0.78
Support Vector Machine	0.015	0.345	0.028	0.777
Multi-Layer Perceptron	0	NaN	NaN	0.78
Bayesian Logistic Regression	0.625	0.329	0.431	0.637

TABLE I
Performances for Positive Class Predictions and Computed Overall Accuracy - Aggregated Dataset

2) Algorithm Choice: After computing our algorithm performance indicators, we ordered them by accuracy rating. From the most accurate to the least one, they were the *MLP*, *Naive Bayes*, *Support Vector Machine*, *Decision Tree*, *Random Forest* and *Bayesian Logistic Regression*. We would have been tempted to experiment with *MLP* and *Naive Bayes*, if it were not for *Null* or *NaN*[5] *F-measures*. Our *Naive Bayes* algorithm failed because of correlations between the features we found in both our Descriptive Statistics phases, whereas the *Multi-Layer Perceptron* classified all *No-show* occurrences as "No"[6], making the calculation of precision impossible. We decided these two algorithms were not suitable for our purpose. The next algorithms in line were *Support Vector Machine*, *Decision Tree J48*, *Random Forest* and *Bayesian Logistic Regression*. We notice from the performance metrics, though, that both *SVM* and *Decision Tree J48* have low *Recall* values, and therefore a low *F-measure*. This means that the models are not that big of an improvement compared to casual classification. We decided then to proceed with *Bayesian Logistic Regression* and *Random Forest*. *Random Forest* has a low *F-measure* (still better thought than *Decision Tree J48* and *SVM*) but good *Accuracy*. On the flip side, *Bayesian Logistic Regression* has lower *Accuracy* but a good *F-measure*. Even though the metrics were not promising, these last two algorithms have the most potential. We then proceeded to optimize these two algorithms to attempt better classification metrics. In this particular case, *Recall* is important because high values mean a lower percentage of *False Negatives* (which means higher prediction of No-shows). In a Public Health System, No-shows bear a higher cost than falsely labeled no-shows (*False Positives*) because of personnel costs. No-show countermeasures, such as text message reminders, typically bear a significantly lower cost.

3) Chosen Algorithm Analysis: Having chosen the two best candidates for our Classification Model, we focused on optimizing the performance of our two algorithms on two fronts: *feature selection* and *subset selection*. We selected features with the *Forward Wrapper* method with *Cohen's Kappa* as guidance: the higher *Cohen's Kappa* is, the better the classification model is with the attribute subset. In order to avoid both over and under-fitting [7] we proceeded with 10-fold *Cross-Validation* with stratified sampling. Performance of the algorithms was computed by averaging *Recall*, *Precision*, *Accuracy* and *F-measure* for all 10 subset instances. Performance results are found in Table II

	Recall	Precision	F-measure	Accuracy
Random Forest	0,096	0,3744	0,1527	0,7657
Bayesian Regression	0,9555	0,2875	0,4418	0,4688

TABLE II
Random Forest and Bayesian Regression averaged indicators - Aggregated Dataset

Another measure of algorithm performance is the *ROC curve* that gives us a visual cue; the guiding principle is that a larger area between the *random* classification method line and the model *ROC curve* is an indicator of good model performance; a *ROC curve* at or below the *random* line is a good indicator of useless/unusable models. Fig. 7 shows the *ROC curve* for both *Random Forest* and *Bayesian Regression* on the aggregated dataset.

[4]This decision tree algorithm splits using Cross-Entropy instead of Gini index

[5]Not a Number

[6]0 in terms of data values

[7]Over-fitting is creating a model which is very performing on a specific set of data but is not usable or does not have the same performances with other data in the same domain. Under-fitting is the same concept but with under-performing models

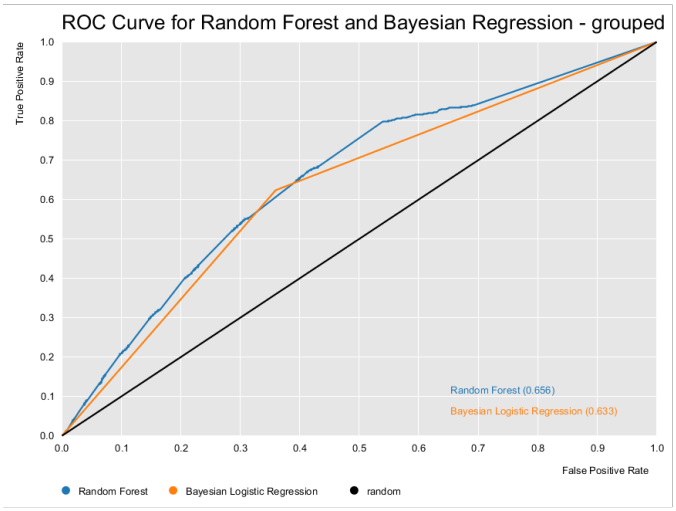

Fig. 7. ROC Curves for aggregated dataset

4) Preliminary Results with Aggregated Dataset: we noticed that our *Random Forest* implementation reached a respectable average *Accuracy* of 76.57%, but a poor *F-measure* because of low *Recall*; *Random Forest*, used on this aggregated dataset, correctly classifies only 9.6% of Positive *No-shows*; a very small part indeed considering our problem and cost minimization effort. Conversely, *Bayesian Regression* reaches a very low *Accuracy* but stronger *F-measure* thanks to a very strong *Recall* of 95.55%, but at the expense of mediocre *Precision*. *Bayesian Regression*, in this case, overestimates *No-shows*, which *might be beneficial in our case* were it not for worse than random classifier performance of 46.88% *Accuracy*, meaning bad overall performance. The performance of both *Random Forest* and *Bayesian Regression* on the aggregated dataset are, therefore, not satisfactory models to solve our problem.

C. Non-Aggregated Dataset Model Performance

Our approach to the Non-Aggregated Dataset is the same we used for the Aggregated one, in a sort of *blind* approach or *A/B Testing*[8]. As stated at the beginning of the paper, this is important in order to avoid as much biases as possible that could negatively influence our problem comprehension and solving.

1) Validity Measures: As we did with the first Dataset, we computed *Accuracy*, *Recall*, *Precision* and *F-measure* performance metrics. We proceeded with the same *Stratified Sampling Holdout*, with 67% data used as *Training Set*, the remaining as *Test Set*. Also in this case we created the *ROC curves* to assess qualitatively the models performance. The performance indicators are found in Table III.

2) Algorithm Choice: The algorithms in descending order of *Accuracy* are MLP, Support Vector Machine, Decision Tree, Naive Bayes, Random Forest e Bayesian Logistic Regression. Again, MLP classifies all values as "No", so *Precision* is

[8]A method of testing solutions simultaneously, typically used in Digital Marketing or UX Design

	Recall	Precision	F-measure	Accuracy
Decision Tree J48	0.036	0.362	0.066	0.792
Random Forest	0.212	0.347	0.263	0.76
Naive Bayes	0.066	0.368	0.113	0.788
Support Vector Machine	0.011	0.333	0.022	0.796
Multi-Layer Perceptron	0	NaN	NaN	0.798
Bayesian Logistic Regression	0.593	0.323	0.418	0.667

TABLE III
PERFORMANCES FOR POSITIVE CLASS PREDICTIONS AND COMPUTED OVERALL ACCURACY

insufficient for our purposes. Again the only algorithms with acceptable *Recall*, and consequently *F-measure*, are *Bayesian Logistic Regression* and *Random Forest*; between the two, the latter has a low *F-measure* but good overall *Accuracy*, whereas the former has lower *Accuracy* and good *F-measure*

3) Chosen Algorithm Analysis: We proceeded with the same optimization we did with the aggregated dataset, by selecting features, by using *Cohen's Kappa* as measure of attribute subset performance, and by making sure we were not Over or Under-fitting with a 10-fold Cross-Validation with stratified sampling. Average performance indicators are found in Table IV.

	Recall	Precision	F-measure	Accuracy
Random Forest	0,1762	0,3715	0,2391	0,7736
Bayesian Regression	0,6305	0,3205	0,425	0,6554

TABLE IV
RANDOM FOREST AND BAYESIAN LOGISTIC REGRESSION AVERAGED INDICATORS - NON-AGGREGATED DATASET

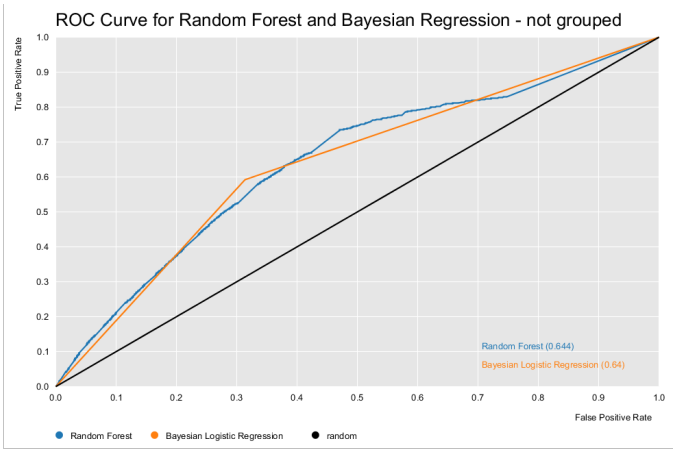

Fig. 8. ROC curves for non-aggregated dataset

4) Preliminary Results with Non-Aggregated Dataset: *Random Forest* has a satisfactory *Accuracy* whilst low *F-measure*, although greater than what happened with the Aggregated Dataset. We still reached the conclusion that its *Recall* is too low, so we cannot use it to solve our problem. *Bayesian Logistic Regression* has lower *Accuracy* than *Random Forest*, but still acceptable and coupled with a much higher *F-measure*. *Recall* is doubled compared to *Precision*, which means Positive *No-show* overestimation. As stated before, *False Negatives* are costlier than *False Positives*, therefore

Bayesian Logistic Regression, even with the lowest overall *Accuracy* between the two, is the only acceptable one for our Machine Learning problem with the Non-Aggregated dataset.

V. Conclusion and Suggestions

After applying the above mentioned algorithms, to both the original, Non-Aggregated Dataset and the Aggregated one, we found out two important clues. Firstly, the fittest model for our main problem, that is predicting No-Shows in order to minimize costs in the Public Health System, is the Bayesian Logistic Regression. After *Forward Wrapper Feature Selection*, and a 10-fold Cross-Validation, the *Bayesian Logistic Regression* Classifier has an *F-measure* of .441 and .955 *Recall* with the Aggregated Dataset and .425. *F-measure* and .63 *Recall*. Considering our cost minimization problem, we need the highest possible *Recall* whilst maintaining acceptable or good *Accuracy*, and no other Model reaches these levels. Secondly, aggregating by *PatientId* is not viable because of an overall Accuracy of .468, statistically worse than casual classification. We conclude that our best model with the data at hand is the *Bayesian Logistic Regression* applied to the Non-Aggregated Dataset. We cannot, thought, recommend our Model as a finished, viable tool for predicting No-Shows in a consistent manner without caution; in order to better minimize waste we would need to be able to better forecast. Generating new features such as *Wait_Time*, as we did, was rather useful, as it proved to be an important piece of information to feed our model. However, it still remain rather improbable to implement a better Model, even with Ensemble Learning[9], without more complete data, with more occurrences and features.
We therefore suggest the Brazilian Public Health Service to:

- *Increase number of Features in their dataset*
- *Re-Run the models and Re-Assess which Classifier is better*
- *Attempt Ensemble Learning*

References

[1] Kaggle. 2017. Medical Appointment No Shows. [ONLINE]
Available at: https://www.kaggle.com/joniarroba/noshowappointments

[2] Weather Underground. 2016. Vitoria Aeroporto, Brazil. [ONLINE]
Available at: https://goo.gl/mdxRx7

[3] Il Sole 24 Ore. 2012. Se salti la visita medica prenotata scatta la segnalazione a Equitalia. [ONLINE]
Available at: http://www.ilsole24ore.com/art/notizie/2012-01-06/salti-visita-medica-prenotata-105246.shtml?uuid=AaWlXIbE

[9] A technique in which different Machine Learning Algorithms are chained together to better predict

APPENDIX

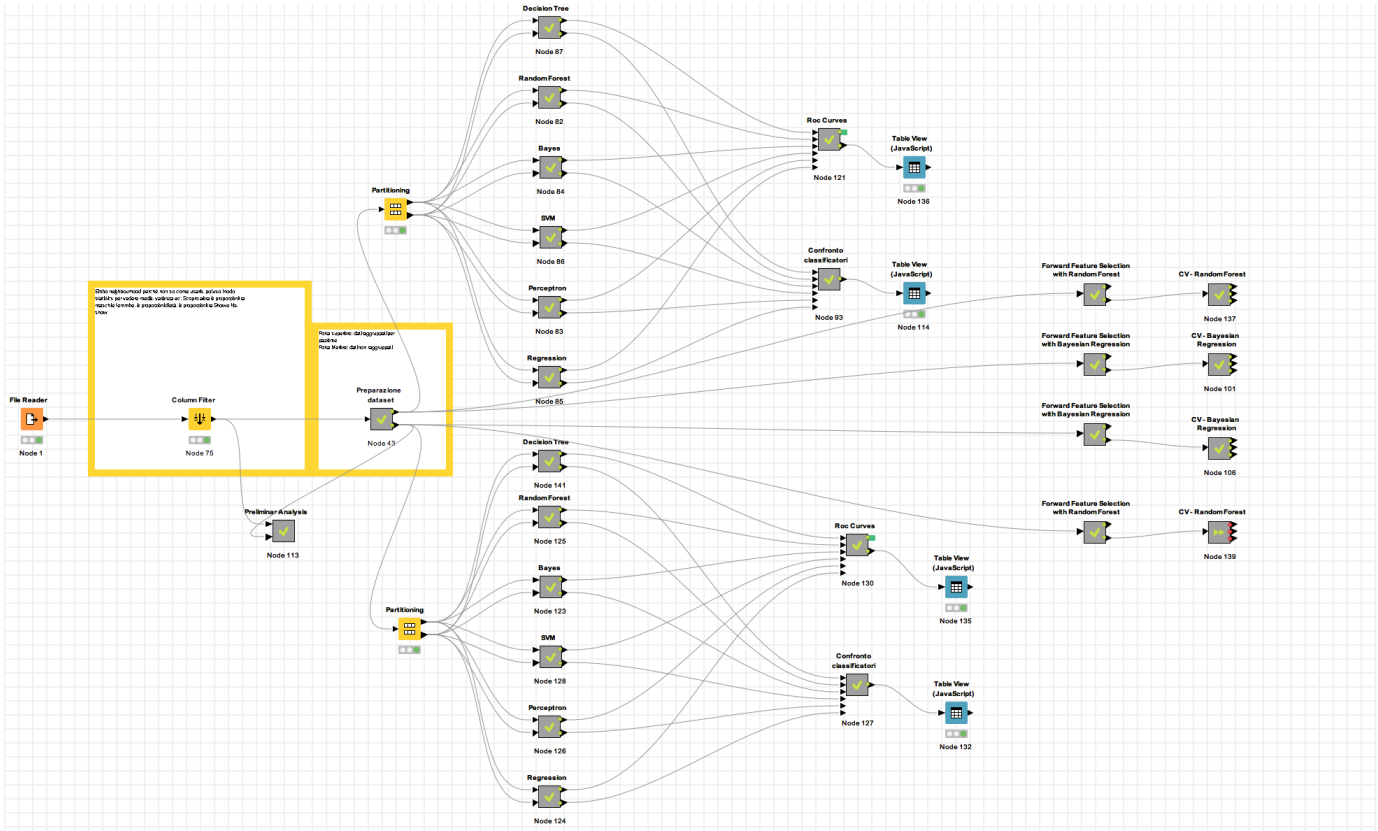

Fig. 9. The Knime Workflow

Row ID	Min	Max	Mean	Std. d...	Skewness	Kurtosis	No. miss...	No. NaNs
Age	-1	115	37.089	23.11	0.122	-0.952	0	0
Scholarship	0	1	0.098	0.298	2.699	5.286	0	0
Hipertension	0	1	0.197	0.398	1.522	0.316	0	0
Diabetes	0	1	0.072	0.258	3.316	8.993	0	0
Alcoholism	0	1	0.03	0.172	5.471	27.928	0	0
Handcap	0	4	0.022	0.162	8.274	82.556	0	0
SMS_received	0	1	0.321	0.467	0.767	-1.412	0	0

1

Row ID	Gender	Age	Schola...	Hipert...	Diabe...	Alcoh...	Handc...	SMS_r...	No-sh...
Gender	1	-0.107	-0.114	-0.056	-0.033	0.106	0.022	-0.046	-0.004
Age	-0.107	1	-0.09	0.503	0.293	0.102	0.079	0.015	-0.061
Scholarship	-0.114	-0.09	1	-0.02	-0.025	0.035	-0.009	0.001	0.029
Hipertension	-0.056	0.503	-0.02	1	0.433	0.088	0.085	-0.006	-0.036
Diabetes	-0.033	0.293	-0.025	0.433	1	0.018	0.059	-0.015	-0.015
Alcoholism	0.106	0.102	0.035	0.088	0.018	1	0.004	-0.026	-0
Handcap	0.022	0.079	-0.009	0.085	0.059	0.004	1	-0.025	-0.007
SMS_received	-0.046	0.015	0.001	-0.006	-0.015	-0.026	-0.025	1	0.126
No-show	-0.004	-0.061	0.029	-0.036	-0.015	-0	-0.007	0.126	1

2

Row ID	PatientId	Appoi...	Gender	ScheduledDay	AppointmentDay	Age	Neighbourho...	Schola...	Hipert...	Diabe...	Alco...	Han...	SMS_r...	No-sh...
Row0	29,872,49...	5642903	F	2016-04-29T18...	2016-04-29T00:...	62	JARDIM DA PENHA	0	1	0	0	0	0	No
Row1	558,997,7...	5642503	M	2016-04-29T16...	2016-04-29T00:...	56	JARDIM DA PENHA	0	0	0	0	0	0	No
Row2	4,262,962...	5642549	F	2016-04-29T16...	2016-04-29T00:...	62	MATA DA PRAIA	0	0	0	0	0	0	No
Row3	867,951,2...	5642828	F	2016-04-29T17...	2016-04-29T00:...	8	PONTAL DE CAM...	0	0	0	0	0	0	No
Row4	8,841,186...	5642494	F	2016-04-29T16...	2016-04-29T00:...	56	JARDIM DA PENHA	0	1	1	0	0	0	No
Row5	95,985,13...	5626772	F	2016-04-27T08...	2016-04-29T00:...	76	REPÚBLICA	0	1	0	0	0	0	No
Row6	733,688,1...	5630279	F	2016-04-27T15...	2016-04-29T00:...	23	GOIABEIRAS	0	0	0	0	0	0	Yes
Row7	3,449,833...	5630575	F	2016-04-27T15...	2016-04-29T00:...	39	GOIABEIRAS	0	0	0	0	0	0	Yes
Row8	56,394,72...	5638447	F	2016-04-29T08...	2016-04-29T00:...	21	ANDORINHAS	0	0	0	0	0	0	No
Row9	78,124,56...	5629123	F	2016-04-27T12...	2016-04-29T00:...	19	CONQUISTA	0	0	0	0	0	0	No
Row10	734,536,2...	5630213	F	2016-04-27T14...	2016-04-29T00:...	30	NOVA PALESTINA	0	0	0	0	0	0	No
Row11	7,542,951...	5620163	M	2016-04-26T08...	2016-04-29T00:...	29	NOVA PALESTINA	0	0	0	0	0	1	Yes
Row12	566,654,7...	5634718	F	2016-04-28T11...	2016-04-29T00:...	22	NOVA PALESTINA	1	0	0	0	0	0	No
Row13	911,394,6...	5636249	M	2016-04-28T14...	2016-04-29T00:...	28	NOVA PALESTINA	0	0	0	0	0	0	No

3

Row ID	Min	Max	Mean	Std. d...	Skewn...	Kurtosis	No. m...	No. N...
Age	0	115	37.089	23.11	0.122	-0.952	0	0
Scholarship	0	1	0.098	0.298	2.699	5.285	0	0
Hipertension	0	1	0.197	0.398	1.522	0.315	0	0
Diabetes	0	1	0.072	0.258	3.315	8.992	0	0
Alcoholism	0	1	0.03	0.172	5.47	27.926	0	0
Handcap	0	1	0.002	0.042	23.503	550.409	0	0
SMS_received	0	1	0.321	0.467	0.767	-1.412	0	0
Wait_Time	0	179	10.184	15.255	2.666	11.797	0	0
Rain	0	1	0.154	0.361	1.92	1.686	0	0
Mist	0	1	0.041	0.198	4.64	19.528	0	0
Storm	0	1	0.079	0.27	3.117	7.713	0	0

5

Row ID	Gender	Age	Schola...	Hipert...	Diabe...	Alcoh...	Handc...	SMS_r...	Wait_...	Rain	Mist	Storm	No-sh...
Gender	1	-0.107	-0.114	-0.056	-0.033	0.106	0.009	-0.046	-0.045	-0.005	0.001	-0.001	-0.004
Age	-0.107	1	-0.09	0.503	0.293	0.102	0.016	0.015	0.033	-0.003	0	0.002	-0.061
Scholarship	-0.114	-0.09	1	-0.02	-0.025	0.035	-0.001	0.001	-0.022	-0.004	0.006	-0	0.029
Hipertension	-0.056	0.503	-0.02	1	0.433	0.088	0.025	-0.006	-0.008	-0.004	-0.003	0	-0.036
Diabetes	-0.033	0.293	-0.025	0.433	1	0.018	0.024	-0.015	-0.021	-0.001	-0.002	-0.002	-0.015
Alcoholism	0.106	0.102	0.035	0.088	0.018	1	0.002	-0.026	-0.045	0.004	-0.002	-0.001	-0
Handcap	0.009	0.016	-0.001	0.025	0.024	0.002	1	-0.008	-0.005	0.003	-0.001	0.001	0
SMS_received	-0.046	0.015	0.001	-0.006	-0.015	-0.026	-0.008	1	0.573	-0.08	0.095	0.083	0.127
Wait_Time	-0.045	0.033	-0.022	-0.008	-0.021	-0.045	-0.005	0.573	1	0.001	0.037	0.019	0.282
Rain	-0.005	-0.003	-0.004	-0.004	-0.001	0.004	0.003	-0.08	0.001	1	-0.088	-0.125	-0.003
Mist	0.001	0	0.006	-0.003	-0.002	-0.002	-0.001	0.095	0.037	-0.088	1	-0.061	-0.003
Storm	-0.001	0.002	-0	0	-0.002	-0.001	0.001	0.083	0.019	-0.125	-0.061	1	-0.01
No-show	-0.004	-0.061	0.029	-0.036	-0.015	-0	0	0.127	0.282	-0.003	-0.003	-0.01	1

6