Predicting No-Shows to Medical Appointments

Kathleen Zhu and Alice Li

I. INTRODUCTION

Due to social reforms that occurred throughout the 1970s in Brazil, the Brazilian National Health Care System provides free healthcare to all citizens. Despite this, the country is plagued by socioeconomic inequalities and as a result, healthcare varies tremendously by class. Furthermore, wastes run rampant in a system that is already grossly underfunded. Our project hopes to address one source of waste and inefficiency in the Brazilian healthcare system - medical appointment no-shows - by creating a model that will predict whether or not a particular patient with a specific appointment time will be a no-show. In doing so, we hope to motivate health care providers to take measures to follow-up with patients who are potential no-shows not only to reduce administrative wastes, but also to minimize the number of individuals who miss their care.

II. DATA ANALYSIS

A. Data Overview

1) Kaggle Data: Our primary dataset is obtained from Kaggle, posted by Joni Hoppen, who obtained it from the Municipality of Vitória. The raw data contains 15 features and 299806 observations. Each observation corresponds to a different appointment within the public healthcare system in the city of Vitória, Espírito Santo, Brazil between 2014 and 2015.

The 15 features associated with each observation in the dataset. These features include the age and gender of the patient, information about the time and date the appointment was made, information about the time and date of the appointment itself, and some health conditions of the patient. The response variable is status, a binary indicator variable for whether or not the patient was a no-show.

2) Weather Data: We supplement this dataset with historical weather data from 2014 to 2015 that we scraped from *www.wunderground.com*. For each day in the years 2014 and 2015 we took the mean temperature of that day as well as the inches of precipitation. We merged this data with our Kaggle Data, so we can observe the temperature and amount of rain on the date of the appointment for each observation.

Variable Name	**Description**
NoShow	Boolean: 1 if patient is a no-show, -1 o/w
RegDate	Datetime: date of registration for the appointment
ApptDate	Datetime: date of the appointment
ApptMonth	Integer: month of the appointment
WaitTime	Integer: number of days between the date of registration and the date of appointment
DayOfTheWeek	String: day of the week of the appointment
Age	Integer: age of the patient
Gender	Boolean: 1 if patient is female, -1 if male
Diabetes	Boolean: 1 if patient has diabetes, -1 o/w
Alcoholism	Boolean: 1 if patient is an alcoholic, -1 o/w
Hypertension	Boolean: 1 if patient has hypertension, -1 o/w
Handicap	Boolean: 1 if patient has a disability, -1 o/w
Smokes	Boolean: 1 if patient smokes, -1 o/w
Scholarship	Boolean: 1 if patient is part of the financial aid program Bolsa Família, -1 o/w
Tuberculosis	Boolean: 1 if patient has tuberculosis, -1 o/w
Sms_Reminder	Boolean: 1 if an SMS reminder was sent to patient, -1 o/w
Temp	Integer: Temperature in Fahrenheit of Vitória
Precip	Integer: Precipitation in inches of Vitória

B. Data Cleaning

Overall the data seems fairly clean, there are no missing values, clear meaningful categories. We address the following issues:

1) Data Merging: We begin by combining our two datasets. We merge the weather data onto the Medical Dataset; we use *ApptDate* as the key.

2) Duplicate Observations: We removed **343** duplicate rows from our dataframe. These are likely errors, because *AppointmentRegistration* is tracked up to the minute, so the chances of two individuals having identical features and registering for an appointment at the same time is not very likely.

3) Valid Entries: We cleaned data that took impossible values:

- There were several negative ages, so we restricted *Age* to nonnegative values. We drop 6 rows with negative ages.
- For *Handicap* and *SMS_Reminder*, binary indicator variables, there were several instances where a positive indicator took a value greater than 1, (such as 2 or 4) but we restrict these values to 1.

- For ease of interpretation, we converted *WaitTime*, the number of days between a patient making the appointment to the actual date of the appointment, from a negative value to a positive one.

4) Feature Engineering:

- We convert all binary variables to one-hot encoding. For instance, for the outcome variable *NoShow* we changed the values (No-show or Show-Up) to 1 and -1, respectively.
- We create variables *RegDate*, *ApptDate*, *ApptMonth*, to track more detailed features of the dates.

C. Descriptive Statistics

NoShow shows that among the appointments within our dataset, approximately 30% were a no-show. The average wait time is nearly 14 weeks. Among patients who made an appointment, 57% received an SMS reminder regarding their appointment. The average age of the patients within this dataset was 38 years old. Of the patients in our data, 67% were female. Approximately 10% receive government stipends.

D. Data Visualization

We begin by investigating the relationships between No-Shows and the various binary indicator variables in the data. A correlation allows us to observe the pairwise associations at a glance.

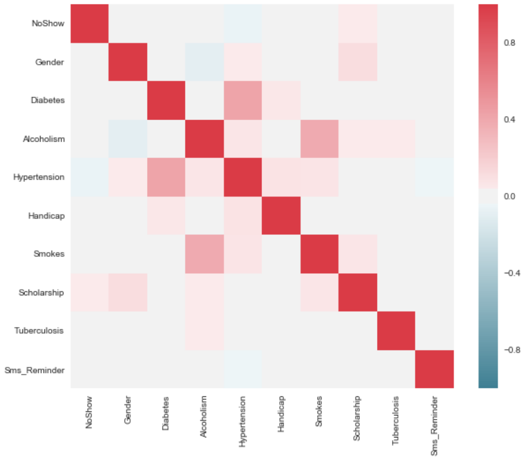

Fig. 1. No-Show Proportion by Weekday

We observe that variables like *Smokes* and *Alcoholism* are correlated, but there seems to be a lack of correlation between health conditions and showing up to appointments. There does, however seem to be a slight positive correlation between *NoShow* and *Scholarship*, and a slight negative correlation between *NoShow* and *Hypertension*.

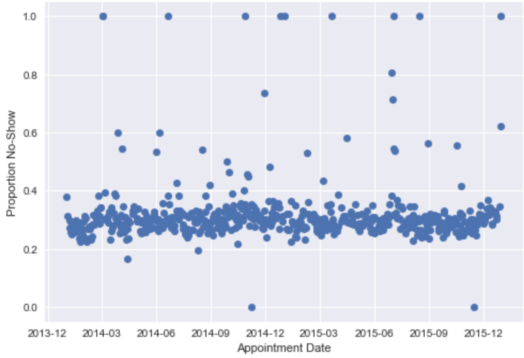

Fig. 2. No-Show Proportion by Appointment Date

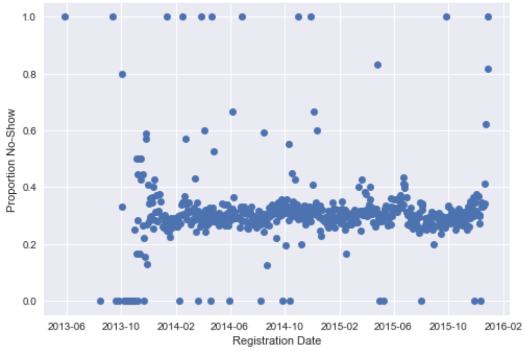

Fig. 3. No-Show Proportion by Registration Date

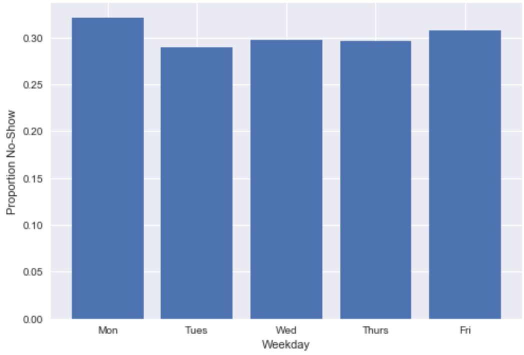

Fig. 4. No-Show Proportion by Weekday

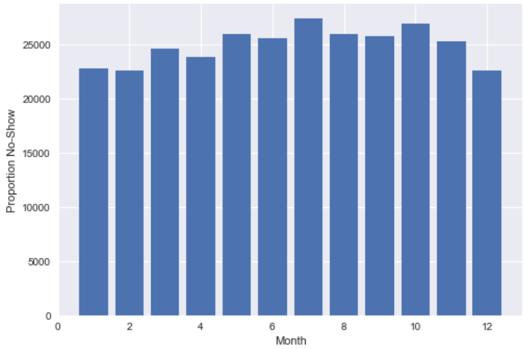

Fig. 5. No-Show Proportion by Month

We begin by investigating the relationship between time and No-Shows. From Figures 1 and 2 we observe that there doesn't seem to be a trend over time.

Next we look at bar plots that show the proportion of No-Shows by the day of the week (very few appointments for Sunday and Saturday) and month. From Figures 4 and 5, there seems to be slight differences by weekday and by month.

Then we examine the relationship between *Age* and proportion of No-Show. In Figure 6 we see that older patients, those above 60, tend to have less No-Shows than younger patients.

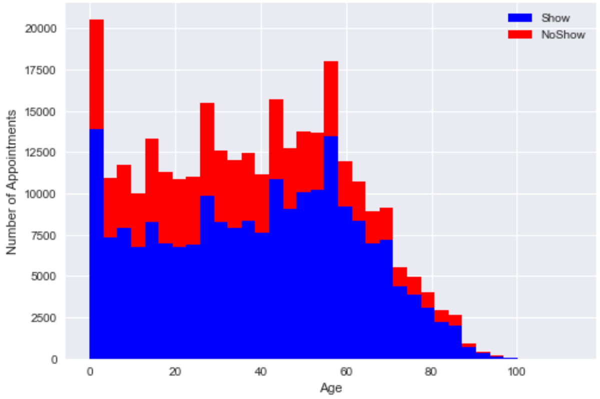

Fig. 6. Distribution of Appointments by Age and No-Show

In the scatterplot below we look at the proportion of No-Show against wait times, which seems to have a slightly positive association. Therefore, *Age* and *WaitTime* seem to be a promising predictors.

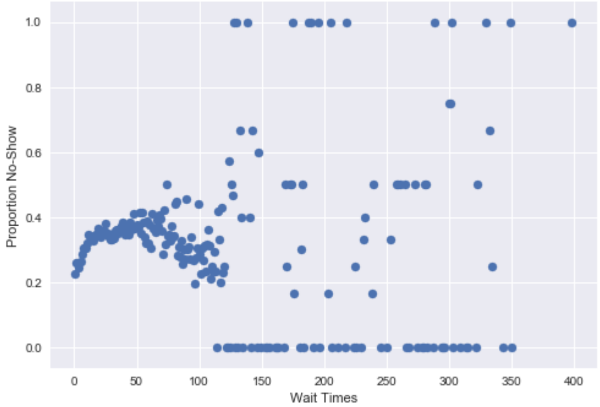

Fig. 7. No-Show Proportion by Wait Times

Finally, we examine the relationship between the proportion of No-Show patients and the weather conditions on the day of the appointment. From Figure 8, it seems that there might be a seasonality effect on the likelihood of No-Shows based on the zigzag of the data. But overall, it does not seem to be a strong predictor.

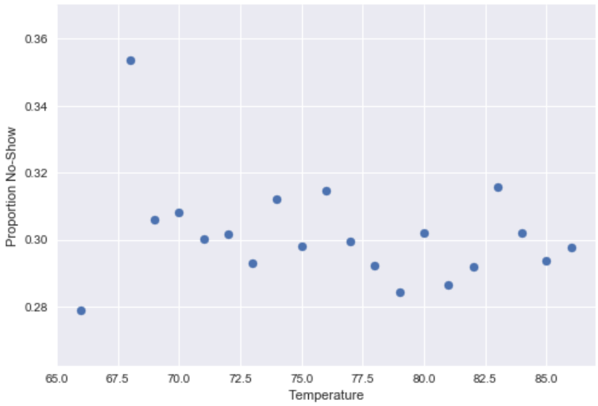

Fig. 8. No-Show Proportion by Temperature

Lastly, we examine the proportion of No-Show patients by Precipitation. In Figure 9 we see a moderately strong linear relationship between inches of Precipitation and the proportion of patients that are No-Shows. This suggests that *Precip* might be a good predictor of *NoShow*.

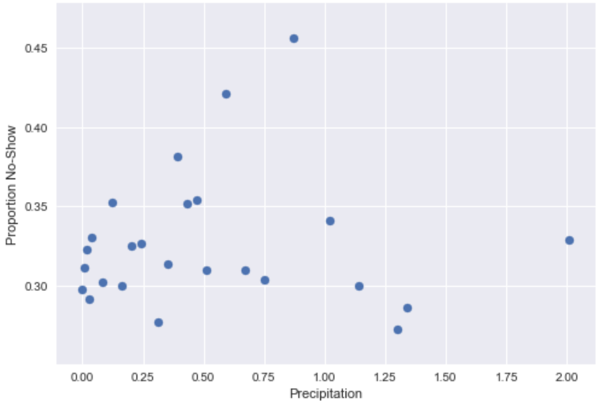

Fig. 9. No-Show Proportion by Precipitation

Therefore, based on the descriptive statistics above, the features we speculate to have some degree of predictive power are: *Scholarship*, *Hypertension*, *DayOfTheWeek*, *ApptMonth*, *Age*, *WaitTime*, and *Precip*. We begin exploring models trained on these variables.

III. MODEL SELECTION: LOSS FUNCTIONS

Our goal is to solve a classification problem; we want to determine whether or not a patient will be a no-show for their appointment. There are several loss functions used for classification problems including: 0-1 loss, quadratic loss, hinge loss, and logistic loss. There are advantages and disadvantages to each based on their properties of convexity and differentiability. We will not investigate 0-1 loss because it is neither continuous nor convex and is difficult to optimize. However we do implement the other loss functions and discuss our methodology below. We split our data into training and test sets, 80% and 20% respectively. In each of our loss functions, we will set parameter λ to 1.

A. Baseline: Quadratic Loss

Quadratic loss is not the ideal function for a classification problem,because it is sensitive to outliers and penalizes heavily for larger prediction values. However it is continuous and differentiable and therefore, easy to optimize. These properties allow for computational methods such as QR factorization and Singular Value Decompositions. Therefore, we choose to use it as our baseline model in our classification problem.

Then we trained a linear regression model on the training set, without regularization. The objective function of the linear model is as follows:

$$minimize \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

Ordinary linear regression usually outputs a value on the reals, so for classification purposes, we set a threshold of 0 for the predicted value of the model so:

$$\hat{y} = sign(w^T x_i)$$

B. Quadratic Loss with l_1 regularizer

Next, we selected regularizers to add to our loss functions. Regularization prevents overfitting, stabilizes our weight estimates so that the solution is less sensitive to small changes in the data, and produces unique solutions. Specifically, we chose the quadratic (l_2) and l_1 regularizers. The l_1 regularizer tends to produce sparse solutions, which is apt in this problem because many variables were added through one-hot encoding. We will apply these two regularizers to both the logistic and hinge loss models. The objective function of the quadratic loss with l_1 regularizer is as follows:

$$minimize \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda ||w||_1$$

C. Quadratic Loss with l_2 regularizer

We implement Quadratic Loss yet again, but with a l_2 regularizer this time.

D. Hinge Loss with l_1 regularizer

A good loss function for classification is the Hinge loss. Hinge loss is continuous, convex, and differentiable except at one point, and produces easy calculations. Also, it does not over-punish for ”overly correct” predictions.

The objective function of the hinge loss with l_1 regularization is as follows:

$$minimize \sum_{i=1}^{n} l_{hinge}(x_i, y_i; w) + \lambda ||w||_1$$

where

$$l_{hinge} = (1 - y w^T x)_+$$

E. Hinge Loss with l_2 regularizer

Next we implemented Hinge Loss regression with a quadratic regularizer. The quadratic regularizer penalizes large parameter values that destabilize estimates and shrinks coefficients to 0. The objective function of the hinge loss with l_2 regularization is as follows:

$$minimize \sum_{i=1}^{n} l_{hinge}(x_i, y_i; w) + \lambda ||w||^2$$

F. Logistic Loss with l_1 regularizer

The next type of model we decided to implement was a logistic loss. The logistic loss function converges to hinge loss for large values and is continuous and differentiable. Most importantly though, the predicted values from the logistic loss model have a probabilistic interpretation. This is due to the sigmoid loss function that ”squeezes” the predicted values, and restricts them to the interval [0,1]. Therefore we can examine both our classification predictions and the likelihood of each prediction. The objective function of the logistic loss with l_1 regularization is as follows:

$$minimize \sum_{i=1}^{n} l_{logistic}(x_i, y_i; w) + \lambda ||w||_1$$

where

$$l_{logistic} = log(1 + exp(y w^T x))_+$$

G. Logistic Loss with l_2 regularizer

We also implement logistic loss with a quadratic regularizer. The objective function of the logistic loss with l_2 regularization is as follows:

$$minimize \sum_{i=1}^{n} l_{logistic}(x_i, y_i; w) + \lambda ||w||^2$$

IV. Proximal Gradient Method

For each problem, we will be using the proximal gradient method on the training set to solve for the parameters. The proximal gradient method can be tweaked for each of the different loss functions and regularizers. The general algorithm is as follows:

1. Pick step size sequence $\alpha_t = 1$ and $w_0 \in R^d$

2. Repeat $w_{t+1} = prox_{\alpha_t r}(w^t - \alpha_t \nabla(\ell_{lossfunction}(w^t)))$

The proximal operator, which varies with each regularizer, is defined as:

$$prox_r(z) = argmin_w(r(w) + \frac{1}{2}||w - z||^2)$$

In the following sections, we will define the proximal operators for each regularizer and the gradients/subgradients for each loss function to elucidate the computations behind the packages we have used to obtain our parameters.

A. Proximal Operators

1) l_1 Regularizer: The proximal operator of the l_1 regularizer is as follows:

$$prox_r(z) = argmin_w(\lambda||w||_1 + \frac{1}{2}||w - z||^2)$$

$$= ((s_\lambda(z))_i$$

$$\text{where } (s_\lambda(z))_i = \begin{cases} z - \lambda \text{ for } z > \lambda \\ 0 \text{ for } |z| < \lambda \\ z + \lambda \text{ for } z < -\lambda \end{cases}$$

2) l_2 Regularizer: The proximal operator of the l_2 regularizer is as follows:

$$prox_r(z) = argmin_w(\lambda||w||^2 + \frac{1}{2}||w - z||^2)$$

$$= \frac{z}{2\lambda + 1}$$

B. Gradients and Subgradients

1) Quadratic Loss: The gradient of the quadratic loss is as follows:

$$\nabla\ell_{quadratic} = -X^T(yXw^t)$$

2) Hinge Loss: The subgradient of the hinge loss is as follows:

$$\text{where } \partial\ell_{hinge} = \begin{cases} \{-yx\}, \text{ for } 1 - yw^Tx > 0 \\ \{0\}, \text{ for } 1 - yw^Tx < 0 \\ \{\alpha yx, \alpha \in [0, 1]\}, \text{ otherwise} \end{cases}$$

3) Logistic Loss: The gradient of the logistic loss is as follows:

$$\nabla\ell_{quadratic} = \frac{-yexp(-yw^Tx)x}{1 + exp(-yw^Tx)}$$

V. Model Selection: Trees

A. Decision Tree

For our next model, we take an intuitive approach to classification and implement a decision tree. A decision tree is constructed from a binary splits made at each node, where each split minimizes the total squared error of the predictions made. Each node accounts for a decision, for instance *How old is a patient?* could be a question at a node, and the partitions could be sectioned by *Age* $\leq$ *20*, *20* $<$ *Age* $<$ *60* , or *Age* $\geq$ *60*. Based on which condition an observation belonged to, it would be partitioned off to a different section of data, and classified with the rest of its partition. The goal is to have each branch be as homogeneous as possible, in order to maximize the information gain from the branch. In other words, we want to minimize entropy.

Therefore, we build a decision tree model with an entropy criterion, a maximum depth of 5, and minimum nodes of size 2.

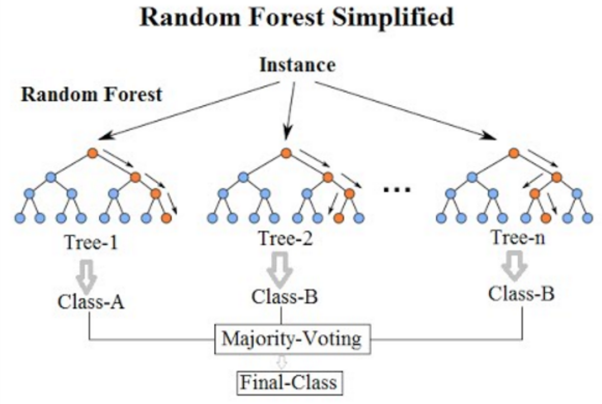

Fig. 10. Random Forest Schematic[1]

B. Random Forest

Individual decision trees are not very stable because there could be many advantageous splits for a given section of data. Therefore, an improvement upon an individual decision tree is relying on an ensemble of decision trees, appropriately named a random forest. In a random forest model, the prediction of any observation is run through all multiple decision trees and the final prediction is made based off majority voting. This is shown in Figure 10 above.

VI. RESULTS

TABLE I
MISCLASSIFICATION RATES BY MODEL

Model	Misclassification Rate
Quadratic Loss with no regularizer	0.30308
Quadratic Loss with l_1 regularizer	0.30308
Quadratic Loss with l_2 regularizer	**0.30265**
Hinge Loss with l_1 regularizer	**0.30265**
Hinge Loss with l_2 regularizer	0.30590
Logistic Loss with l_1 regularizer	0.30306
Logistic Loss with l_2 regularizer	0.30300
Decision Tree	0.30273
Random Forest	0.35041

A. Error Metric

Because what we cared most about was whether or not a patient could be predicted as no-show, we measured the number of misclassifications of each model on the test data, to determine its respective accuracy. This is given by the expression $\sum_{i=1}^{n} y_i \neq sign(w^T x_i)$.

B. Model Comparison

The quadratic loss was our baseline model for the different loss functions. Among the the quadratic loss models, the misclassification rate for the quadratic loss with l_2 regularizer was smallest with a value of 0.30265. It is also the model that performed best among all the models. Hinge loss with l_1 regularizer produced the same misclassification rate. Slightly higher is the decision tree model with a misclassification rate of 0.30273. The random forest model performed worst with a misclassification rate of 0.35041. However, overall there is not a significant difference between misclassification rates across models.

C. Sparse w

Important to note is that with all the loss functions, an l_1 regularizer produced extremely sparse results. For the quadratic loss with l_1 regularizer, only the coefficient for *Age* produced a non-zero value. In addition, this value of -0.002094 is so small as to make it even negligible. For the hinge loss with l_1 regularizer, only *WaitTime, Age,* and *ApptMonth* produced non-zero values. The values for the coefficients were -0.000460, -0.015429, and -0.018822 respectively. In this model, *WaitTime* is nearly negligible. Lastly, for the logistic loss with l_1 regularizer, only *Age* produced a non-zero value of -0.016671.

D. Small w

In addition, the optimization problems with no regularizer and a l_2 regularizer are telling of a larger issue. Even though they did not produce sparse results like the models with the l_1 regularizer did, the optimizations produced coefficients that were rather small. The one with the largest magnitude is *Age* for the hinge loss with l_2 regularization. Its value was -0.029586.

E. Decision Tree and Random Forest

Curiously, our Random Forest model has the higher misclassification rate than our Decision Tree model. It actually has the highest misclassification rate over all the models. We speculate that this is due to the fact that the Random Forest discourages over fitting. However, in a dataset with too much noise and very little signal to fit, the averaging effect of such an ensemble model causes a decline in performance.

VII. CONCLUSION

We are confident in our approach to cleaning the data and investigating the different models.

In our preliminary report, we had noticed that our initial logistic regression only had an accuracy of around 70%. As a result, we had hoped to supplement the original patient data with data on the weather of the day to create a richer, more predictive dataset.

Despite this, however, there was not much signal to capture from the features we inspected. The sparsity of the parameters of the loss functions with l_1 regularization and the small magnitudes of non-zero parameters for ones with no regularization and with l_2 regularization suggested that many of the features we examined such as age and gender, the set of health conditions we studied, and daily weather were not as predictive of a no-show as we had hoped for despite some initial promising analyses.

After extensive modeling, our best models did no better than if we had just assumed that nobody was a No-Show. This is because only about 30% of patients were actually No-Shows. So if our model indiscriminately predicted that everyone would show up for their appointment, our misclassification rate would be approximately the same. That is to say, our models did not successfully outperform the trivial model. Ultimately we cannot answer our original question: *What features are good predictors of a patient missing a scheduled medical appointment?* Therefore, we would not recommend our system of models.

VIII. FUTURE IMPROVEMENTS

In thinking about improvements to be made to the analysis of a no-show, a fruitful point of improvement would be to collect data on more indicative features of a no-show. In our preliminary analyses, the feature *Scholarship* was promising. However, it is a boolean value and may not represent the socioeconomic diversity of patients within the data set.

The dataset we have collected has information on patients from all neighborhoods within the city of Vitória. To study patient No-Shows across a range of socioeconomic statuses, it will be necessary to cull out different neighborhoods within the city of Vitória as it is unlikely that public health care systems have a record of features representing a patient's socioeconomic status. With the data on the neighborhood that a patient lives in, one could then infer average household incomes, poverty levels, employment rates, and average education levels as possible attributes.

Another promising avenue is the historical data of patient No-Shows. We imagine an improved dataset that contains a unique PatientID feature that tracks the number of times a patient has missed appointments in the past. Oftentimes the best predictor of future action is past actions, so we believe that the addition of this feature will improve predictability.

TABLE II
POTENTIAL FEATURES

PatientID	...	No. Missed Appts	Neighborhood
AAAA1234	...	0	Fradinhos
CFRZ6773	...	1	Bela Vista

Though we cannot offer a substantially predictive model, we can offer a recommendation to Vitória to keep better patient records. As these more informative records make for better data, which will make for improved models.

REFERENCES

[1] Random Forest Template for TIBCO Spotfire , Wiki Page — TIBCO Community, community.tibco.com/wiki/random-forest-template-tibco-spotfirer-wiki-page.