Machine Learning Engineer Nanodegree

Capstone Project

Edgar Valente
April 14th, 2019

I. Definition

Project Overview

In the city of Vitória (state of Espírito Santo, Brazil) there are numerous public health facilities. In these, the number of patients that don't attend to their medical appointments has reached nearly 30% from 2014 to 2015. This pattern is similar all over the country, so it isn't a local issue. This indice represents nearly R$20 million of public cash being wasted by appointments that result in no-shows, taking into account operational costs, SMS sent, confirmation call and employee's time. That's at least R$1 million monthly waste.

Problem Statement

This problem will be addressed mainly as a classification problem, where I'll try to predict if a patient will show up or not on his appointment based on demographic data (age and gender) and appointment characteristics, such as day of the week which it is scheduled and number of days from registration to scheduled appointment.

As viewed in this article, this problem has been addressed before in the San Carlos Clinical Hospital, in Madrid, having demographic data, patient history and classes for medical appointments. Unfortunately, the dataset used in this case isn't as complete, but it should still be possible to predict no-shows with a certain accuracy.

Additionally, a segmentation method through unsupervised learning will be applied in order to find the characteristics of people who show up and people who do not. Care will be taken so that the groups have high variance regarding the target label and each group will be mainly composed of either people who do or do not show up. Having said characteristics might lead to a possible recommendation of any feature that is mutable at the time of appointment.

Metrics

The prediction model shall be evaluated through f1-score, so that it takes into consideration both precision and recall, assuring that an imbalanced distribution of the target label won't have a bias on accuracy. f0.5-score will also be taken into consideration, prioritizing precision, so that it minimizes unnecessary actions that would be taken in order to assure that the patient would show up.

For the attempt in recommendation, something similar to the elbow method will be applied, so that the number of clusters with highest variance among the target label will be chosen. The difference being that the elbow method takes into account explained variance of each cluster, so that adding one more cluster is unnecessary, and I wanna choose a number that maximizes segmentation of the target label.

To measure the efficacy of the recommendation, a test set will be used to predict the target label both before and after the recommendation, and the difference of ratio between predicted show ups and no shows will be used to evaluate the recommendation.

II. Analysis

Data Exploration

The dataset used in this project has 300,000 observations and 13 variables (300000, 13). There are two numerical variables (age and days_to_appointment), two labeled categorical (gender and week_day), and other nine binary, containing 1s and 0s representing true and false, respectively.

It is important to notice that the target label (no_show) is slightly unevenly distributed, having around 70% for show-ups (1) and 30% for no-shows (0). The two numerical variables have been checked for skewness, which might need further transformations. Of these, "*days_to_appointment*" seems to be skewed to the left, as shown in the graph below, and will be dealt with when clustering with power transformation and possibly removing outliers through IQR (Interquartile Range).

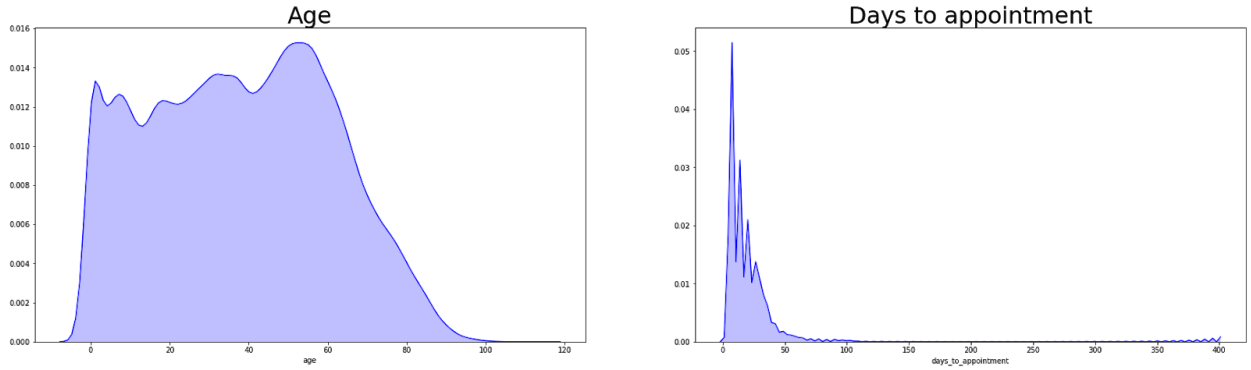

Fig. 1: distribution of age (to the left) and days_to_appointment (to the right).

The variables are more thoroughly described as follows:

- **age**: integer, age of person to make the appointment;
- **gender**: categorical, gender of person to make the appointment, allowing either male or female;
- **week_day**: categorical, the day of the week of the scheduled appointment;
- **days_to_appointment**: integer, days from appointment registration to scheduled appointment;
- **diabetes**: categorical binary, whether the person has diabetes;
- **alcoholism**: categorical binary, whether the person is an alcoholic;
- **hypertension**: categorical binary, whether the person has hypertension;
- **handicap**: categorical binary, whether the person has a handicap;
- **smokes**: categorical binary, whether the person is a smoker;
- **monetary_help**: categorical binary, whether the person receives monetary help from the government, called 'Bolsa família';
- **tuberculosis**: categorical binary, whether the person has tuberculosis;
- **sms_reminder**: categorical binary, whether the person has received an SMS reminder for the appointment one day before it;
- **no_show**: categorical target label, indicating if the person showed up or not on the appointment

Below is a sample of the dataset with its most promising features:

age	gender	week_day	days_to_app	no_show
5	f	thursday	1	0
69	f	friday	4	0
16	f	tuesday	5	1
8	m	monday	14	1
7	m	thursday	15	0

Table 1: sample of dataset with only four explanatory features and target label “no_show”.

Exploratory Visualization

Among the numerical variables (age and days_to_appointment), there are no obvious correlation between days to appointment and show up status. Age, on the other hand, has a clear distinction, where older people seem to attend to their appointments more frequently.

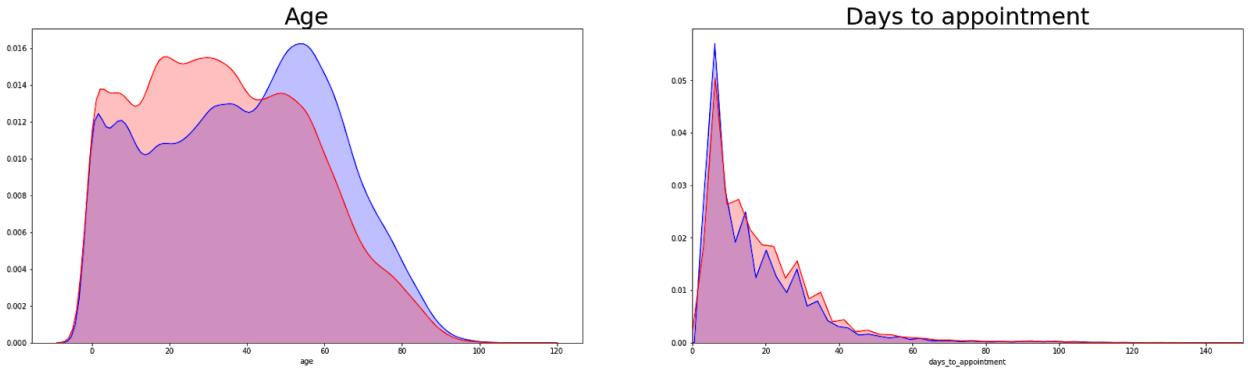

Fig. 2: distributions of age and days to appointment over the week. Blue represents show ups and red no shows.

Extending that visualization to days of the week, in order to find if there’s a specific day where people seem more likely to attend, the result is quite similar, with the exception of saturday, where there doesn’t seem to be much difference between the age of attendees.

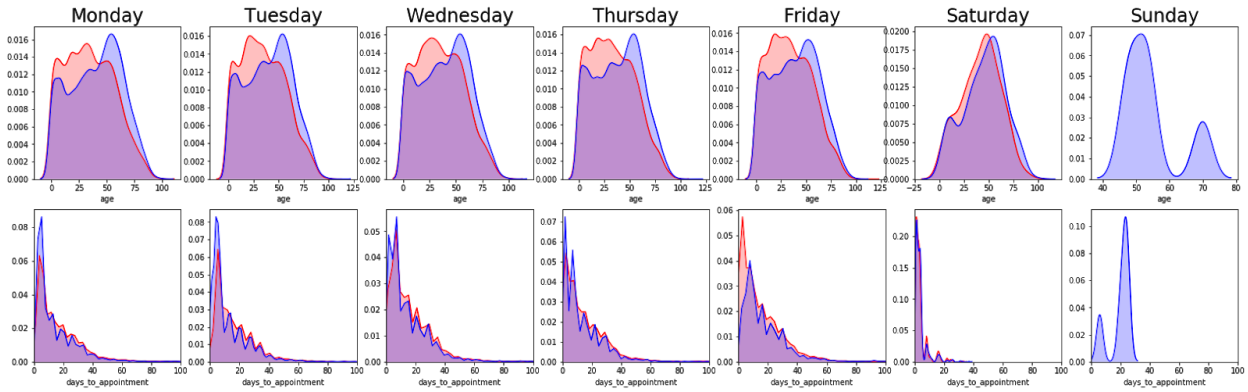

Fig. 3: distributions of age (above) and days to appointment (below) per show up per day of the week, where blue represents show up and red no show.

Comparing both numerical variables and the attendance status, we find an interesting pattern, where people tend to make later appointments on specific periods of their life, specifically between ages 0-10 (infants) and 50-80 (older people), probably meaning that these periods are where they care the most about periodic appointments. Also, there seems to be a more frequent amount of show ups at higher scheduling, which could indicate that age and days_to_appointment, together, might be a good indicator if a person is gonna attend.

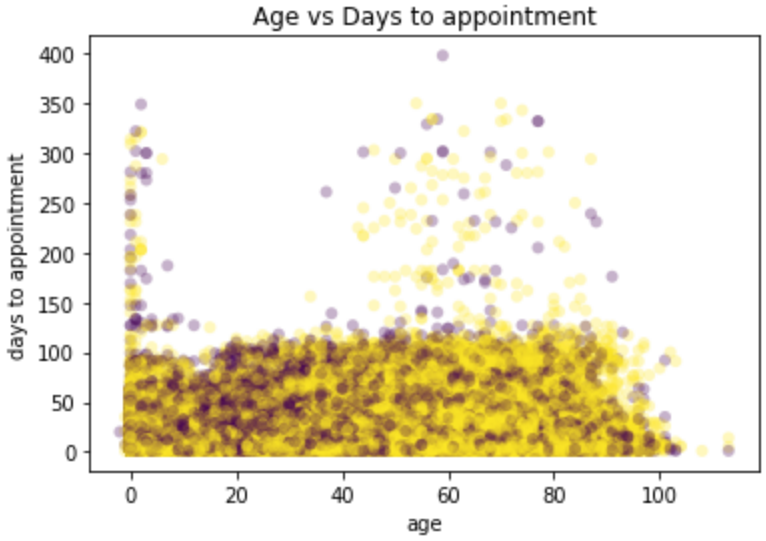

Fig. 4: Age vs days to appointment, where purple dots are no shows and yellow are show ups.

Algorithms and Techniques

The classifier for prediction will be Gradient Boosting, specifically from the *xgboost* python package, which has been used in many *Kaggle* competitions for classification. It is an ensemble of trees that takes into consideration each previously built tree's error to create the next one, and its results are among the best of its category.
The following parameters are gonna be checked for tuning the model:

- max_depth: maximum depth of each generated tree. Low value will avoid overfitting;
- learning_rate: rate at which the model updates each prediction based on the last tree. Low value will avoid overfitting;

For model selection, GridSearchCV from the module *scikit-learn* will be used, checking permutations of previously mentioned parameters while performing cross validation. Also, within the *xgboost* package, the parameter *early_stopping_rounds* will be deployed, where it stops training if the evaluation metric - in our case, f1-score - hasn't improved for the last *n* trees' predictions. This method allows us to avoid overfitting while selecting the model with best metric.

For the unsupervised learning problem, Gaussian Mixture Models from *scikit-learn* will be used, since it allows clusters with varied shapes, contrary to the hyperspherical clusters formed by other algorithms like KMeans. The inputs will be appropriately preprocessed for its application. The clusters centers are gonna be extracted from the model in order to attempt recommendation.

Benchmark

A Logistic Regression algorithm will be applied to have a base metric so that we can use it for benchmarking our final selected model. Also, GridSearchCV already cross-validates models with its hyperparameters permutations.
The results from previously mentioned article are gonna be used as a basis for comparison, since the problems are quite similar, even though the data varies slightly, having its final prediction error of around 28% (equivalent accuracy of 72%).

III. Methodology

Data Preprocessing

A few transformations had to be made in order to apply both models:

- One hot encoding (supervised and unsupervised);
- Outlier removal (supervised and unsupervised);
- Feature selection (supervised and unsupervised);
- Power transformation (unsupervised);
- Normalization (unsupervised);

Both categorical variables - *week_day* and *gender* - have been one hot encoded. For outlier removal, the Interquartile Range method was applied on *days_to_appointment* after power transformation specifically for this purpose. Since it is skewed to the left, only higher values (to the right) were removed using this method, taking out 96 exaggerated samples from the dataset, which all had *days_to_appointment* values higher than 224. More samples could have been removed, but it is also intended to have clusters formed by these individuals.

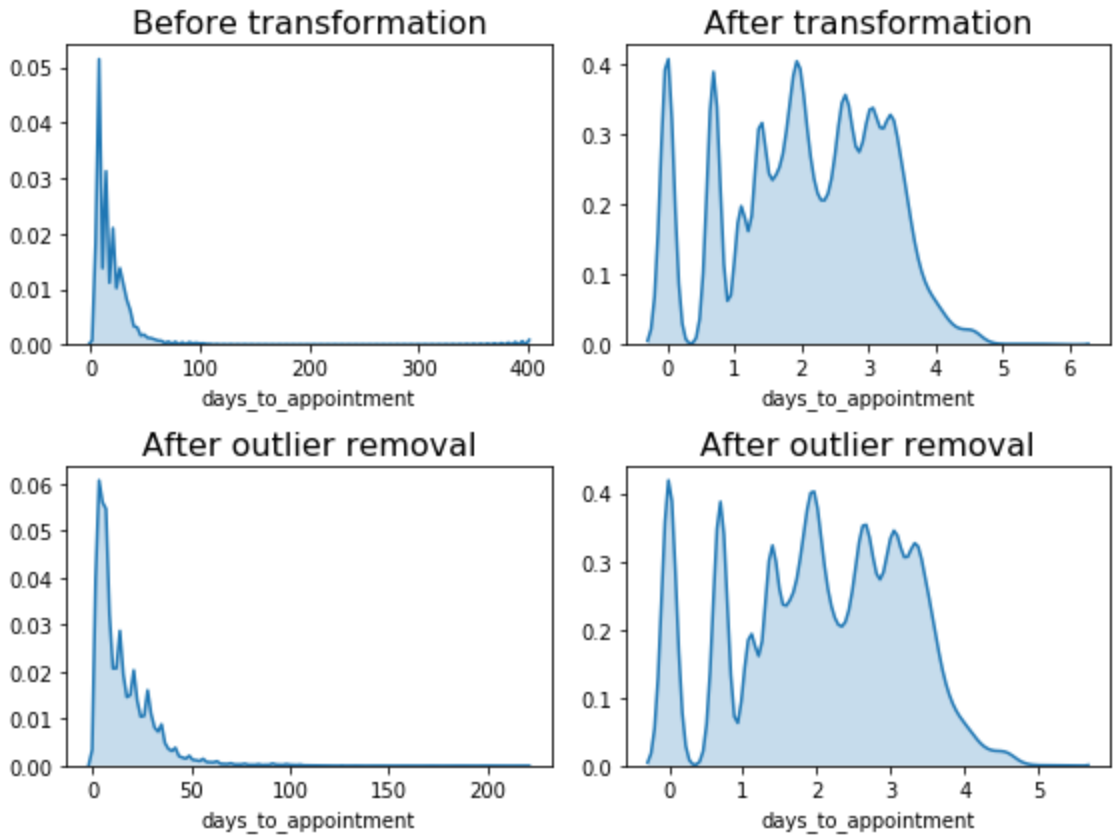

Fig. 5: distribution of days_to_appointment before and after log transformation and after outlier removal.

In order to more accurately segment clusters, feature selection had to be applied, so that each group is more likely to describe its target label.
Two methods were used for this purpose:

- Information Gain from xgboost model. It takes into consideration how much information is obtained from the data after tree splits at one specific feature.
- Mutual Information from sklearn. It segments each feature and compares its contained information with the segmented target label. The higher this metric, the more likely it is for both features to have their segments clustered together.

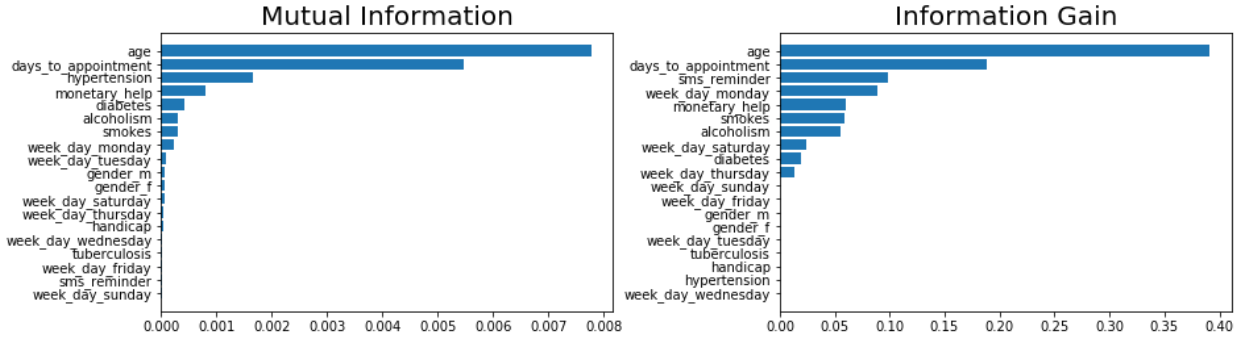

Fig. 6: feature importances calculated with Mutual Information (left) and Information Gain (right).

Since it is desired to apply both supervised and unsupervised learning on the same data, only features with high values on both metrics were chosen, those being **age**, **days_to_appointment** and **monetary_help**. Also, the categorical variables *gender* and *week_day* are not being used after feature selection, so there is no need for further one hot encoding.

Tree ensembles like the one used for classification don't need normalization, since they are not based on sample distances. Gaussian Mixture Model, on the other hand, considers euclidean distance for clustering, hence the need for normalization on both numerical variables, being applied with *scikit-learn*'s MinMaxScaler and transformed to values in the range between 0 and 1. Power transformation (log, in this case), however, is only needed on the variable *days_to_appointment*, previous to normalization, to adjust its skewness into a normal distribution.

Implementation

The preprocessing step resulted in two sets of data, all maintaining the original proportion of target label, which will be used separately for supervised and unsupervised learning:

- Train and test data split into classifier features and target label (supervised);
- Train and test data with all features, with log transformation and normalization (unsupervised);

That said, three models have been made:

- Logistic Regression model for benchmarking;
- Extreme Gradient Boosting for main prediction model;
- Gaussian Mixture Model for clustering and further recommendation attempt;

For supervised learning the first set of train and test data was used (in notebook *"supervised"*), where the features **age**, **days_to_appointment** and **monetary_help** were used, with **no_show** being the target label, and the positive label 1 means no show while 0 means show up.

The Logistic Regression model has been made using solver *lbfgs*, which is an optimization algorithm that uses low amount of computer memory. It wasn't expected to be a good model, but for benchmarking instead. Its accuracy score measured 0.7, seeming a good result, but accuracy isn't the desired metric. Its f1-score was 0.005,

which is terrible for predicting no shows. If we consider 0 (show-up) as being the positive label, this value is 0.82, but this isn't the objective in this project. This is because it predicted 99% of the test set as showing up, while there were actually only 70% of them.

This problem has been addressed when training the second model - extreme gradient boosting (XGBoost). It was cross-validated using GridSearchCV with 3 fold cross-validation and experimenting with hyperparameters *max_depth* (3, 5, and 7) and *learning_rate* (0.3, 0.5 and 0.7) to select the best ones and using f1-score for model selection, building each model 50 trees.

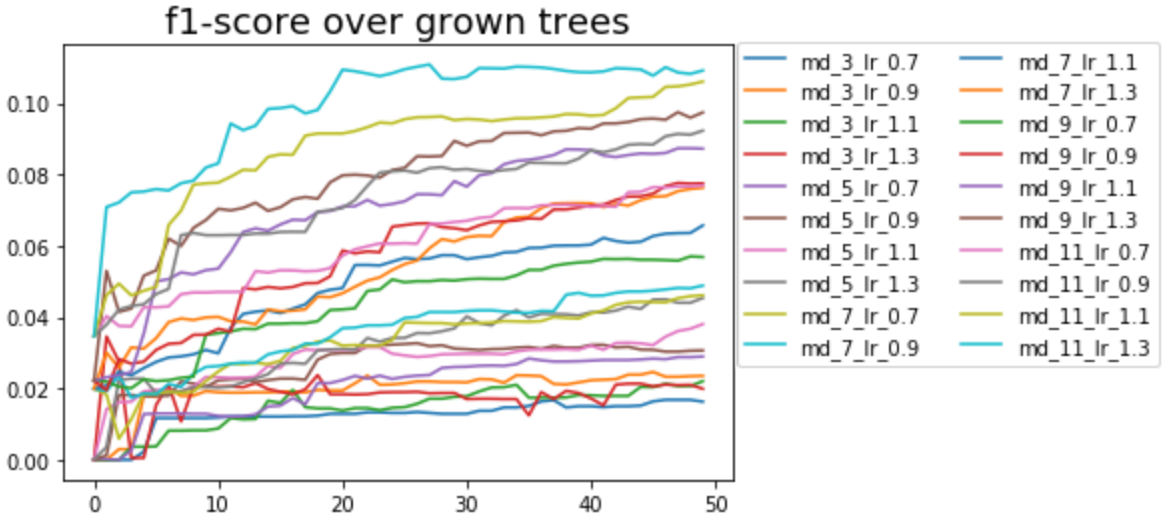

Fig. 7: f1-score measured over each model along each grown tree.

Since the result seemed to be considerably better with higher max_depth and learning_rate, I decided to manually iterate over each of these argument with even higher values to verify how the f1-score is raised along with each argument, which ended up being the highest possible (max_depth 11 and learning_rate 1.3).

Within the best model, the *xgboost* package also allows an evaluation function to select the best model trained while growing the trees, selecting the metric with lowest value. Since high f1-score is desired, the custom function supplied to the algorithm is the inverse of this metric. The model was then applied with the best found parameters, which were learning_rate 1.1 (chosen over 1.3 since it seems steadier) and max_depth 11, building 2000 maximum trees and early stopping if the evaluation metric *inverse f1-score* hasn't improved over the last 100 built trees.

The unsupervised learning approach is applied on the second set of train data (in notebook *“unsupervised”*, so that the same individuals would be used. This dataset is, however, log transformed and normalized, as mentioned in the “Data Preprocessing” section.
The Gaussian Mixture model was used, and the number of clusters desired was selected based on the standard deviation of the its target label centroid values. One model with n clusters was created for each iteration from n=2 to n=24. The standard deviation values, however, were all close to 0.5, which was unexpected. That means every cluster has a centroid “no_show” dimension of either 0.0 or 1.0. So, even though the values vary from 0.47 to 0.49, I decided to arbitrarily choose a value of n where the standard deviation was 0.49 and, in this case, n=11.

I intend to describe the recommendation functions here.

Refinement

I intend to attempt to get a better prediction model by creating a custom objective function.

IV. Results

Model Evaluation and Validation

Best results so far are of f1-score near 0.12. The recommendation doesn’t seem plausible with such a huge error in prediction model.

Justification

V. Conclusion

Free-Form Visualization

Reflection

Improvement

There is the need for more data in this case, and yet there isn't certainty of better results. Somewhere along the lines of the mentioned article, probably, since they had good results, but their data was much more complex. There is a need for the data infrastructure of hereby project's hospital to get better.