Machine Learning Engineer Nanodegree

Capstone Proposal

Edgar Valente
April 1, 2019

Domain Background and Problem Statement

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.

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.

Datasets and Inputs

The used dataset is available on Kaggle as version 3. I'll rewrite their names during the analysis for further clarity, with their updates registered in the dictionary between brackets ('[]'). It contains 300,000 observations and 15 variables (300000, 15). The label

in question is the variable 'Status', which will be renamed as 'show_up' and encoded to 1s and 0s, same as 'Gender' and 'DayOfTheWeek', so that they can be modeled. The train and test sets will be made considering the distribution of the target label, which is around 70% for show-up (1) and 30% for no-show (0).

The variables in question are:

- **Age**: integer, age of person to make the appointment [age]
- **Gender**: categorical, gender of person to make the appointment, allowing either male or female [gender]
- **AppointmentRegistration**: integer, the day the person asked for an appointment [app_registration]
- **AppointmentData**: date, the day of the scheduled appointment [app_date]
- **DayOfTheWeek**: categorical, the day of the week of the scheduled appointment [week_day]
- **Status**: categorical label, indicating if the person showed up (show-up) or not (no-show) on the appointment [show-up]
- **Diabetes**: categorical binary, whether the person has diabetes [diabetes]
- **Alcoolism**: categorical binary, whether the person is an alcoholic [alcoholism]
- **HiperTension**: categorical binary, whether the person has hypertension [hypertension]
- **Handcap**: categorical binary, whether the person has a handicap [handicap]
- **Smokes**: categorical binary, whether the person is a smoker [smokes]
- **Scholarship**: categorical binary, whether the person recieves monetary help from the government, called 'Bolsa família' [monetary_help]
- **Tuberculosis**: categorical binary, whether the person has tuberculosis [tuberculosis]
- **SMS_Reminder**: categorical binary, whether the person has recieved an SMS reminder for the appointment one day before it [sms_reminder]
- **AwaitingTime**: integer, days from appointment registration to scheduled appointment [days_to_appointment]

Solution Statement

The idea is to identify patterns between people that miss appointments and the ones that show-up and, possibly, influence those patterns. Since at the registration of the appointment it is possible to choose the date, I'm hoping to be able to recommend different dates for different people to optimize show-ups. Even if the recommendation is

not possible, I intend to develop a prediction model to at least identify possible no-shows, since it would be possible to act on them with reminders, if applicable.

Benchmark Model

Since there is no way to make an AB Test during the modeling, I intend to create up to two prediction models, depending on the evaluation metrics results: one focusing on f1-score and another focusing on f0.5-score, detailed on the Evaluation Metrics section. I intend to use Logistic Regression (so I have a basis for benchmarking) and a Gradient Boosting ensemble algorithm. Then I intend to create a clustering model - possibly a Gaussian Mixture Model - to check if there are patterns between no-shows and show-up. If there are patterns, I intend to recommend a scheduled day for different test individuals and check on the prediction model if the individual will show-up. I'll be able to tell if the recommendation is better than random if amount of predicted show-ups is higher after recommendation than before, taking into consideration the prediction model's auc or f1-score for significance.

Evaluation Metrics

For the main prediction model, I'm going to use confusion matrix with f1 and fbeta scores. I'm going to prioritize precision over recall (using beta = 0.5), since I want the predicted values to be true positives, so I can try and recommend only on highly assertive no-shows. For validating the recommendation, I'm gonna use a test set and predict it both before and after the recommendation, hoping that the amount of show-ups after recommendation is better. For the clustering model, I'm gonna use silhouette score and also measure variance between the mean of the labels (show-up) inside clusters, so that I can get the clusters with high segmentation in this variable.

Project Design

The input data will be cleaned for clarification. Preprocessing for categorical variables (week_day and gender) will be applied so that I can apply the algorithms. There are no missing values in the dataset, but outliers will be analyzed and removed, if necessary, using the interquartile range (IQR) method. They might be necessary for the unsupervised learning analysis, so that I have clusters specifically for outliers, if any.

The intended supervised learning model will probably be xgboost, and grid search will be used for cross validation and hyperparameter tuning.

I'll have either two or three models:

- Prediction model to identify no-shows;
- Prediction model for benchmark;
- Clustering model to identify patterns between show-ups and no-shows and try to recommend a variable that increases likelihood of show-up;

The models will be created using python's scikit-learn library, using its Pipelines where applicable.
The dataset, both original and edited will be found in a 'data/' directory, while any necessary Jupyter Notebook in a 'notebooks' directory. Models will be saved with pickle in the 'data/' directory as well.