CSC 265 Final Project

Andrew Dettor

Spring 2021

05/03/2021

Abstract

Medical Appointment No Shows is a dataset from Kaggle.com with data from 110,527 medical appointments. The goal of this project is to predict whether the patient will show up to the appointment or not (2-class classification). The predictors include gender, date of scheduling the appointment, date of the appointment, age, neighborhood, and various conditions the patient may have, like hypertension, diabetes, or alcoholism. Additional features were engineered from the date-related features, such as day of week or week of year. K-Means clustering was used to split the dataset into 10 clusters, and the cluster for each observation was used as an additional feature in modelling. Modelling was performed using gradient boosted random forests, with 79.73% test accuracy. The top 10 (out of 103) most important features were extracted and used to fit the model again with 79.65% test accuracy. The models' accuracy and precision scores were calculated and analyzed.

Introduction

A patient missing a medical appointment is costly; medical staff has their time wasted and the patient has to reschedule. It would be useful to be able to predict whether or not a patient will miss an appointment. The goal of the Medical Appointment No Shows dataset on Kaggle.com is exactly that; given information about the patient, how likely are they to not show up to the appointment?

The dataset contains the following predictors with 110,527 examples:

- Patientid
- AppointmentID
- Gender
- ScheduledDay (in ISO8601 time format)
- AppointmentDay (in ISO8601 time format)
- Age
- Neighborhood around the medical practice
- Scholarship (government assistance for medical costs)
- Hypertension
- Diabetes

- Alcoholism
- Handicap
- SMS_received (whether the patient was reminded via text)
- No-show (the response variable)

Important Notes about features:

- PatiendId and AppointmentID were dropped
- Age is the only continuous variable; **all the rest are discrete/categorical**.
- Features were engineered from the ISO8601 time format variable using the lubridate R package. Various aspects about the time were extracted, such as hour of day, month of year, day of year, day of month, etc.
- There were 81 unique neighborhoods, which were one-hot-encoded.
- After data cleaning and feature engineering, there were 103 predictor variables, up from the original 14.
- There were no missing values

Methodology

Supervised Learning Method

For modelling this dataset, gradient boosted random forests were used. To understand this modelling technique, it is important to talk about decision trees. A decision tree is a way to model data by splitting up the feature space to separate the training examples. This is done through “splits” on features. Splits are decided based on the value of a given feature that best separates the data. Various metrics are used for deciding where to split, such as accuracy, or purity based metrics like gini index or entropy. The more “pure” the split, the better it divides the dataset. The first split in a tree is the value of the feature that gives the best value of the metric, among all values of all features. Further splits are the second/third/fourth etc best feature/value combination. Random forests are an extension of the decision tree; they use multiple trees with a few tricks to reduce variance and decorrelate the trees. Variance is reduced through bootstrapping the dataset, and the trees are decorrelated by looking at a random subset of the features for each split in each tree (usually only the square root of the total number of features). Gradient boosting is an extension of random forests. It uses the mistakes of the previous trees to make improvements in future trees. It uses the idea of a gradient to do this, akin to gradients used in optimization. A train/test split of 75%/25% was used.

Unsupervised Learning Method

K-means clustering is an unsupervised learning method used to create an additional feature in this dataset for the supervised method mentioned above. The idea was to combine supervised and unsupervised methods to extract more information from the data. K-means clustering starts by initializing K “centroids” at random locations. Centroids are the name for the centers of each cluster; the closest centroid to a data point is that point’s cluster. The clusters are iteratively moved such that the variation within clusters is as small as possible. K-Means always finds a

local optimum for cluster centroid locations, and every data point is guaranteed to belong to a cluster. It is important to note this method does not utilize the ground truth labels for each data point; it groups data points without that knowledge. However, the data must be scaled for the clustering method to work effectively.

Feature Selection Method

The feature selection method used is the relative importance of features in gradient boosted random forests. This is an embedded method because GBRF's model and perform feature selection simultaneously. The top 10 most important features were found and used to fit the model again, to attempt to reduce variance by having lower dimensionality. The relative importance of each feature is calculated by finding how much each predictor decreases impurity among all the splits among all the trees in the boosted model[1].

Clarifying the Results

The methods used to make more sense of the results are Precision and class imbalance. Precision in this context is the proportion of people who were predicted to not show up, who actually didn't show up. This metric's importance is context specific, but in this context it is important because one wouldn't want to plan for a patient to not show up then have them show up. Class imbalance here is the balance of values in the response variable. Understanding this is important because it gives a reference for what a good accuracy or precision value is. If the classes are not split 50/50, the base expected accuracy can be very different from 50%.

Data Analysis/Summaries

Supervised Learning Method

Gradient Boosted Random Forests were fit on the dataset and grid-search was performed to find optimal hyperparameters. One model was using all features after feature engineering (103 total), and the other used the top 10 features from the first model (feature selection method results discussed later). The grid of hyperparameters consisted of different values for number of trees, and shrinkage value, which details how much the next tree in the boosting process impacts the prediction result. For both models 0.3 was the best shrinkage and 125 was the best number of trees. Unsurprisingly, the model with all the features took significantly longer to fit than the one with 10. The model with all features had a training set accuracy of 79.65% and precision of 41.69%, and a test set accuracy of 79.73% and precision of 39.42%. The test set model seems to have traded off higher accuracy for lower precision. The model with 10 features performed almost the exact same, with a training set accuracy of 79.56% and precision of 38.83%, and a test set accuracy of 79.65% and precision of 37.69%. This one had a similar tradeoff, but one of a smaller magnitude. Note: a training/test split of 75%/25% was used.

GBM Models Performance

	Test (all features)	Test (best 10 features)	Train (all features)	Train (best 10 features)
Accuracy	79.73	79.65	79.65	79.56
Recall	1.48	1.76	1.68	1.86
Precision	39.42	37.69	41.69	38.83
F1	2.84	3.37	3.22	3.54

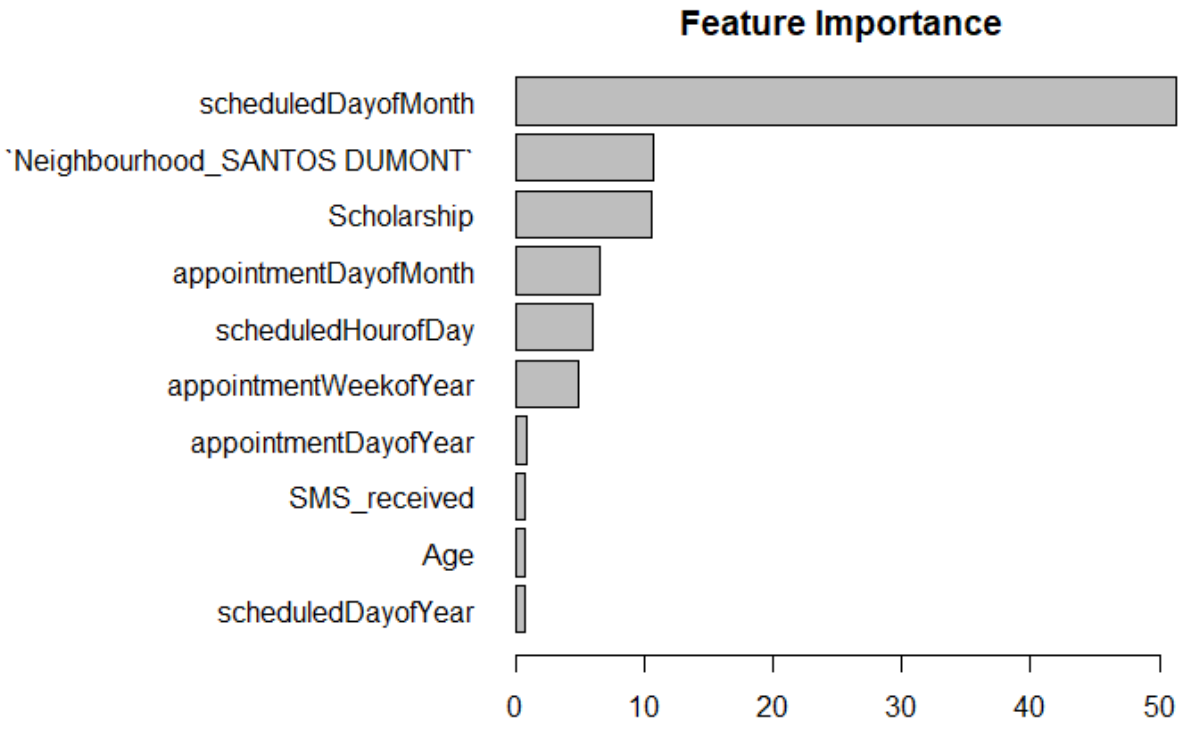
Unsupervised Learning Method

K-means clustering was fit on a scaled version of the dataset. The goal was to combine supervised and unsupervised methods for more predictive power. Here, the train and test sets were both used to create a “Cluster” feature for the supervised learning method. The way I did it does not have target leakage because no class labels were used. 10 clusters were chosen, even though there are only two groups in the response variable, because there might be subgroups within those two groups. The distribution of observations among clusters was about as expected. There were two groups with the vast majority of observations, then a few observations were sprinkled throughout the rest of the clusters (see picture).

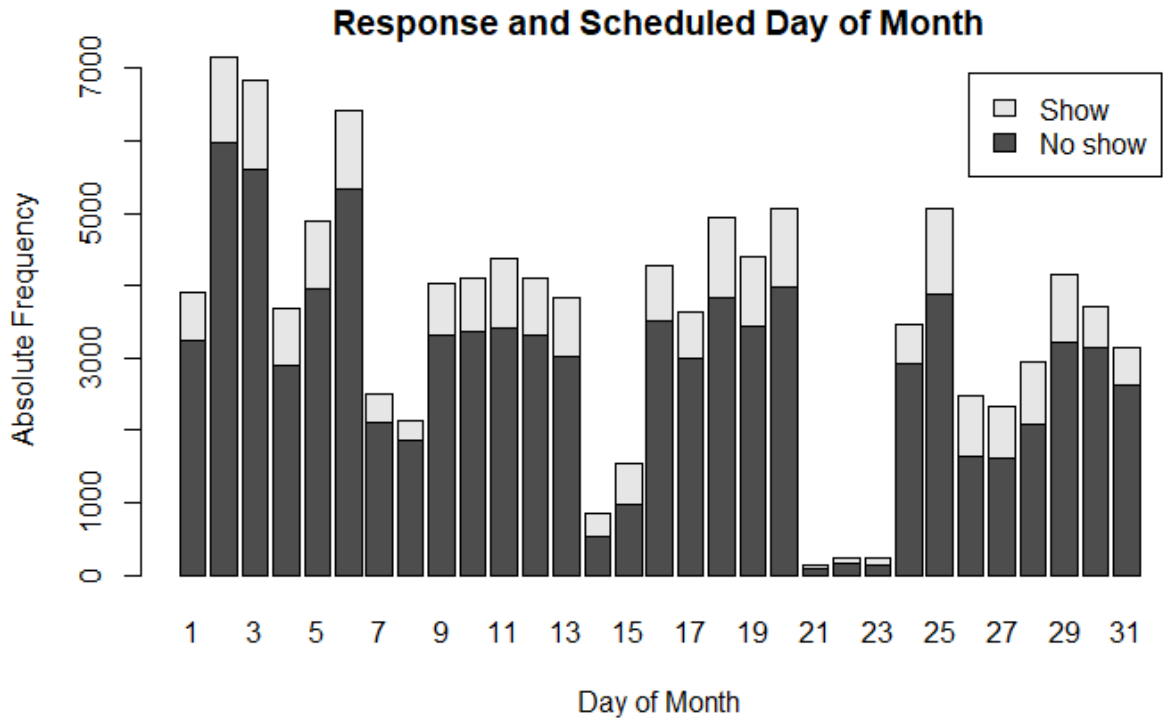
The way most of the data points split into 2 classes shows that unsupervised methods can work to classify sometimes.

Feature Selection Method

Feature importance was calculated during the fitting process each gradient boosted random forest. Below are the top 10 most important features in the model with all features.



There is a lot to notice here. First, the clusters found didn't seem to be that important to the model. Second, the day of the month that the appointment was scheduled was by far the most important in determining whether or not a patient would show up. My guess is that most of the missed appointments were scheduled on a certain day of the month. Other than that it seems very odd. Maybe there's something wrong with the dataset? Moving on, the neighborhood with the most appointment misses is Santos Dumont, which is probably far away from the medical practice, or it's just difficult to transit from that place. Surprisingly, those with government assistance (Scholarship) were also unlikely to show up to the appointment. Furthermore, some weeks or days of the year might be holidays or extra busy so it might be more common for people to miss appointments on those days.

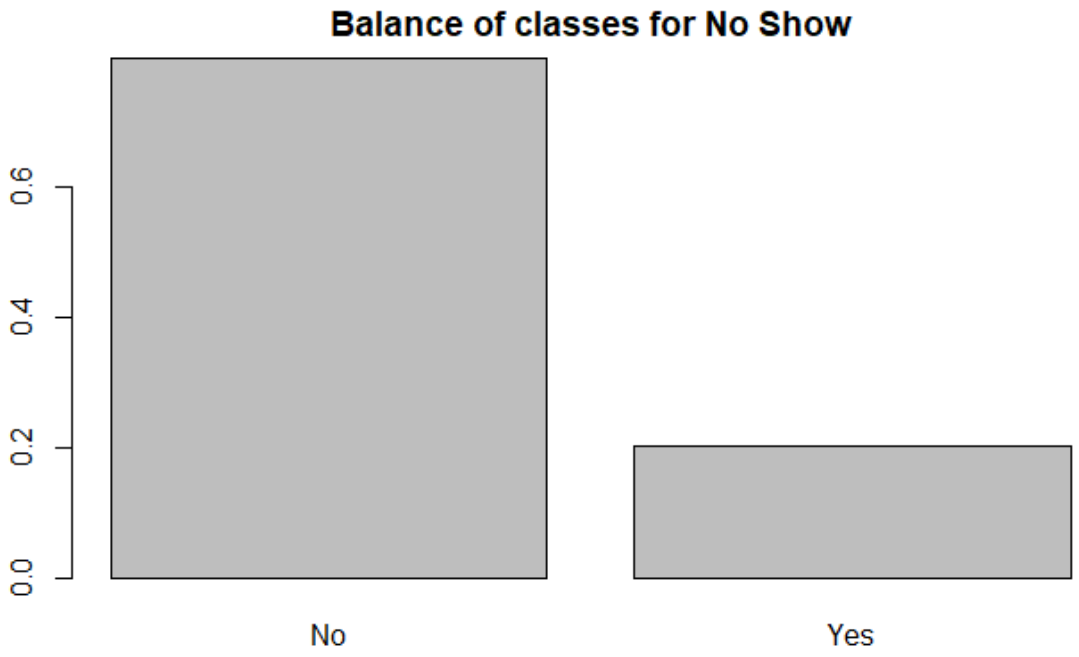


Judging by this graph, it seems that most people miss appointments that are scheduled on the first few days of the month. What's strange is that this isn't referring to the date of the appointment, it's the day the appointment is scheduled. I was not expecting the date of scheduling the appointment to have more of an impact than the day of the appointment.

Clarifying the Results

As one can see by the picture above, all the models had a precision around 40%. This means when the model predicted the patient would not show up, 60% of the time they actually did show up. This obviously would not be good to use in practice, because it's horrible customer service to assume the patient will not show up and give someone else their spot, just to have the original patient show up.

Additionally, class imbalance plays a big role in these models' performance.



As one can see, the majority of observations have “No” as the response. This means the accuracy of these models is inflated. The high amount of “No”’s in the data is probably why the models have such low precision; oftentimes the model predicts “No” when it should predict “Yes” just because “No” is simply much more common.

Discussion and Critics

One issue I had was with scaling the data for clustering. Do I still scale categorical variables? Even ones that only contain 0’s and 1’s? I was not sure but I did it anyway. It seemed to cluster mainly into 2 classes and those clusters had the same proportions as the actual response variable, so I assumed it went well. However, the supervised learning method did not make much use of the clusters, which was odd. There might be an issue with the dataset because there’s no way “scheduled day of month” has a larger effect than almost everything else combined. If I were to move forward or use this model in the real world, I would optimize for precision, for reasons previously mentioned. I also had issues decoding the ISO8601 time features but once I figured it out it was fine. Furthermore, I fear the results from this dataset are dataset-specific. The only neighborhoods in the dataset are very specific ones in Brazil, so it’s not very generalizable.

Conclusion

Clearly, predicting whether patients will show up to an appointment or not is possible, but the dataset used should have an even balance of classes, or the models should be optimized for precision. The unsupervised clustering algorithm of K-means showed there are two classes in the data, which aligns well with there being two classes in the response variable. The

unsupervised method of gradient boosted random forests gave good accuracy of about 80%, but it doesn't look as good when combined with 40% precision. The most important predictors for if someone will show up or not are the day of the month the appointment is scheduled, if they are in a hot-spot neighborhood in terms of people missing appointments, and if they received government assistance.

Reference

1. https://www.displayr.com/how-is-variable-importance-calculated-for-a-random-forest/