Predicting Patients’ Show or No-show for Medical Appointments

Abstract

No one likes to wait! Especially when they wat to see a doctor. It would be great if we can solve this issue with a model that predicts whether a patient is going to show up or not. Hence, giving a chance to other patients and reducing the long waiting hours. For that, I decided to work on appointments dataset found on Kaggle to build a prediction model to help with this issue.

Design

This project aims to predict patient’s show/no-show status. Therefore, helping care providers to manage their schedules efficiently, reduce wait time for walk-ins and increase their chances to see a doctor. After obtaining the dataset, I did some data exploration and cleaning preparing it for modeling. Then built two models in order to get optimistic results to solve the issue.

Data

The dataset contains more than 100K appointment records with 14 features. Features include details about appointments like schedule date, appointment date and no-show status. Patient’s features include, age, neighborhood, health condition and gender. Most of them were examined thoroughly to choose from for modeling.

Algorithm

Feature Engineering

- Converting categorical features to binary dummy variables [gender & no-show]
- Calculating the distance between schedule day & appointment day [didn’t add any value, so it was dropped later]
- After trying different combinations, features concerning the patients themselves were selected, 7 in total

Model

Two models were tested rigorously with cross validation to choose bet performance model. Between Logistic Regression and Random Forest, the second was performing slightly better in testing but way better in testing. As a result, I opted for Random Forest.

Logistic Regression:

- Accuracy 0.624
- Balanced Accuracy 0.500

Random Forest:

- Accuracy 0.626
- Balanced Accuracy 0.522

Model Evaluation and Selection

The data was imbalanced, so I used hyper-method of oversample and undersample to tackle this issue. 50K records were used. It was divided 80/20 between training and testing. Random Forest model has slightly better performance but not that great, as shown below

- Accuracy 0.62
- F1 0.45
- Recall 0.52
- Precision 0.58

The correlation was weak to begin with; different feature-selection techniques were applied including PCA, but did not make any difference to the results.

Tools

- Numpy and Pandas for data manipulation
- Scikit-learn for modeling
- Matplotlib and Seaborn for plotting