Medical Appointment No Shows Prediction

https://github.com/f18bc/data1030project

Bochen Fu

Data Science Initiative

1 Introduction

For health care facilities, receiving a patient's appointment request, then completing the appointment process, but the patient not showing up can be a frustrating issue and a potential cost. Therefore, it becomes beneficial if facilities can medical no-shows in advance.

This report analyzes a dataset of medical appointment no-shows in Vitoria, capital of the state of Espírito Santo, Brazil. In 2013, Vitoria's government started collecting scheduling data in all municipal health care facilities, which includes medical appointment no-shows anonymously, using unique patient ids. This report uses its second version on Kaggle, with the help of Joni Hoppen and Aquarela Advanced Analytics under a CC BY-NC-SA 4.0 license, which contains 110,527 data in 2016.

Each row corresponds to a patient's medical appointment, and there are 14 attributes. There are two variables name typos to be fixed: Hipertension (hypertension), and Handcap (handicap), which is the level of handicap.

Here, we focus on medical no-shows, and use No-show as the target variable. The goal is to predict if a patient will show up for an appointment, and it is a classification problem.

There is no missing data. For individual variables, PatientId and AppointmentID are unique identifiers. Male, Scholarship, Hipertension, Diabetes, Alcoholism, Handcap, SMS_received, and No-show are categorical attributes. Scholarship means whether a patient was receiving benefits of the 'Bolsa Família' social welfare program from the Brazilian Government. Neighbourhood indicates where an appointment took place in Vitoria. Age is a contiguous variable, ranging from -1 to 115. ScheduledDay and AppointmentDay represent the date and time of schedules and appointments. For privacy, the times of AppointmentDay were all set to 00:00. ScheduledDay ranges from November 2015 to May 2016, and AppointmentDay ranges from April 2016 to May 2016. AppointmentDay does not contain a national holiday or Sunday.

Regarding previous work, Helmonds 2018 used several machine learning algorithms to the best model for predicting medical no-shows. It applied Logistic regression, Decision tree, Bagged decision tree, Random Forest, Gaussian Naive Bayes, and Neural network. The result showed that only Logistic regression did not offer meaningful results based on F1 scores and AUC. Moreover, another research (PANDIRI 2019) on Kaggle aimed to predict which features affect medical no-shows. It used Decision tree, Random forest, and GridSearchCV, and concluded that Gender, Age, Neighbourhood, Scholarship, and Hypertension are top features that affect medical shows and no-shows.

2 EDA

First, we study the target variable, No-show.

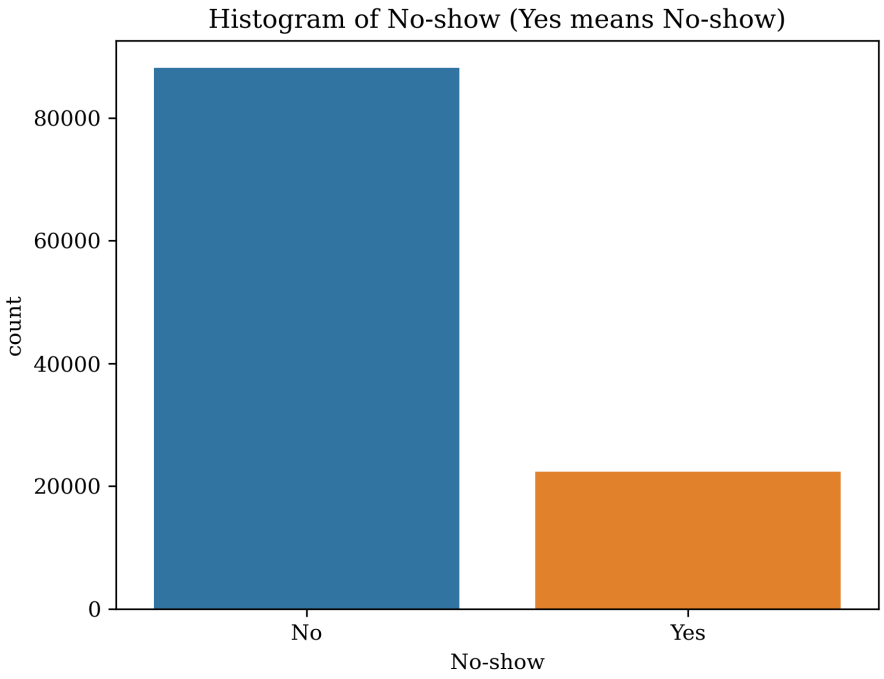

Figure 1: Histogram of No-show (Yes means No-show)

The above plot shows about 20.2% of the patients did not show up (Yes for No-show) for appointments, while 79.8% of the patients showed up (No for No-show), which means people tend to show up.

Moreover, we study three variables that are important and could affect medical no-show.

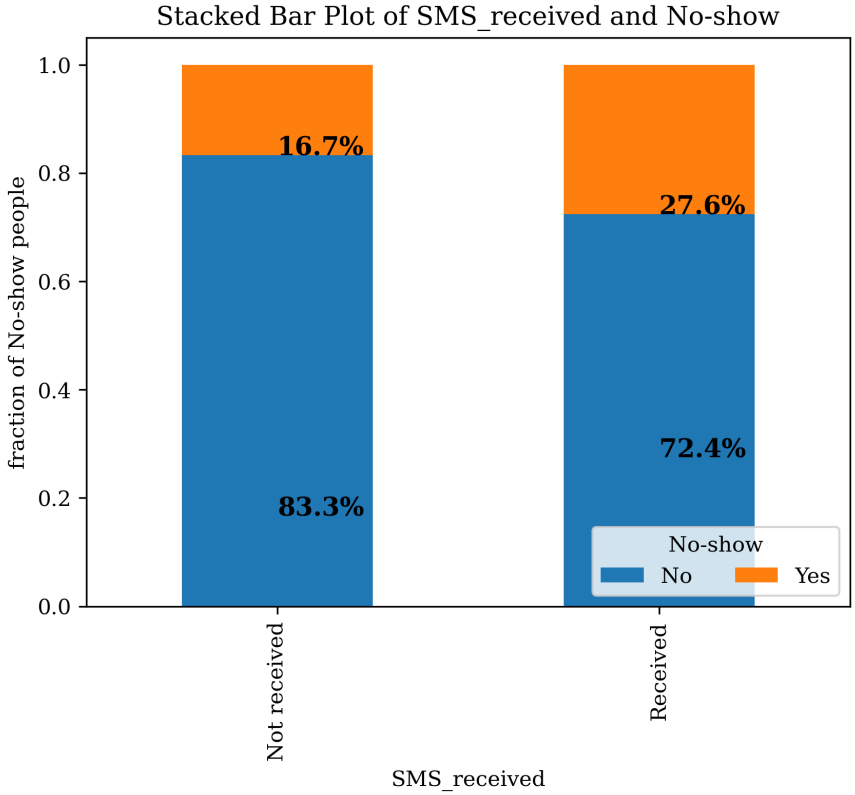

Figure 2: Stacked Bar Plot of SMS_received and No-show

We plot SMS_received and No-show. We can conclude that 16.7% of patients not receiving SMS did not show up,

and 27.6% of patients (10.9% higher) receiving SMS showed up. It means that receiving SMS may increase medical no-shows, which is unexpected.

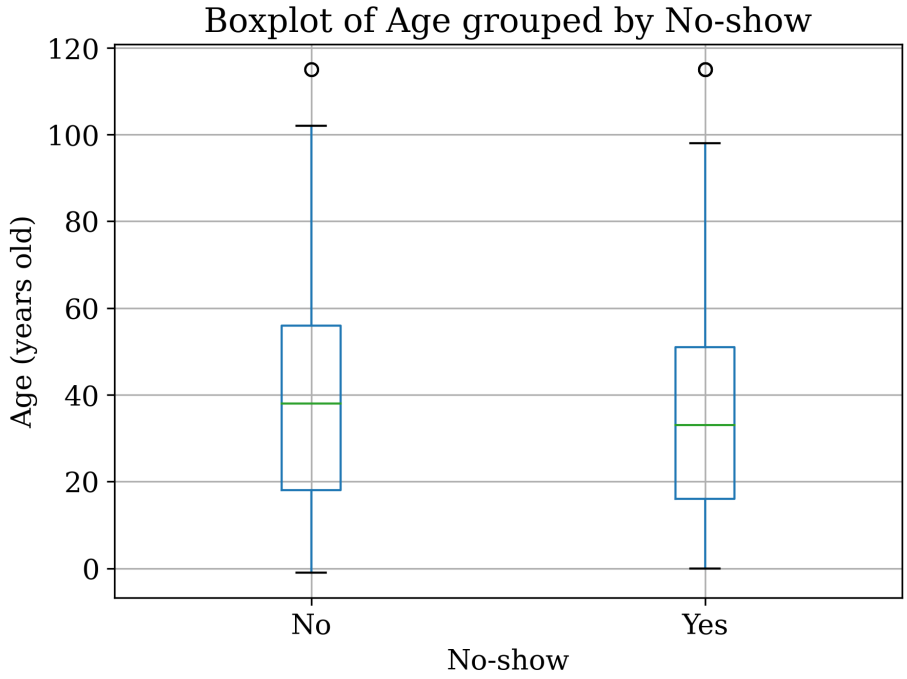

Figure 3: Boxplot of Age grouped by No-show

Figure 4: Bar Plot of Age

Also, we plot Age and no-shows. The first plot shows that the average age of patients who did not show up is lower than those who showed up. Also, it indicates an erroneous value of Age being negative (-1). The second plot shows a lot of observations in Age 0 and 1. For those Age groups, other reasons, namely their patient's are possible, since usually patients of Age 0 or 1 do not decide on medical show-up. Also, there are 5 people with Age of 115 here, we consider it naturally normal.

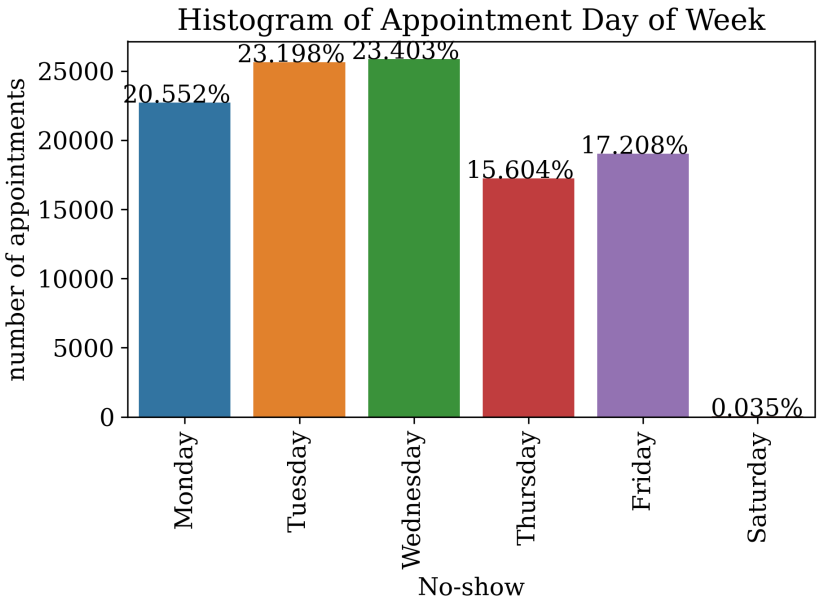

Figure 5: Histogram of Appointment Day of Week

Then, we check the day of week. The above plot shows Wednesday and Tuesday are the top 2 weekdays. Saturday has the least percentage (0.035%). The reason is that oftentimes, health-care facilities do not accept appointments on weekends.

3 Method

3.1 Splitting and Data Preprocessing

The dataset is iid. Here, since about 79% patients are unique, we do not consider a group structure. Also, there is no time series, since we want to predict if a patient shows up or not.

Also, we use the basic split, since the data is iid and big enough ignore the randomness. We first choose a train data size with 60% of original data, split the others in half, then make the validation and test data size both 20%.

Regarding changes for variables, we change AppointmentDay to AppointmentDayofWeek, to easily see how the day of week affects the prediction. For Neighbourhood, we label each as a unique value. Also, for ScheduledDay, we delete the feature because here to focus more on AppointmentDay, since ScheduledDay barely affects it. For Age, we have one patient of -1 age. Since it makes no sense that a person has negative age, and one observation is -1, we delete the row. Moreover, we delete identifiers PatientId and AppointmentID.

Regarding encoders, we use OneHotEncoder for categorical features: Gender, Scholarship, Hypertension, Diabetes, Alcoholism, Handicap, SMS_received and Neighbourhood. Neighbourhood has 81 unique values, and others are 0 or 1. Since Handicap and AppointmentDayofWeek have orders, we use OrdinalEncoder. Handicap has logical order, from no handicapped to serious handicapped, and AppointmentDayofWeek has chronological order. For Age, we use MinMaxScaler, since it is not normally distributed, and we know its max and min based on experiences.

After preprocessing, we have 110526 data and 11 columns (10 features).

3.2 ML Pipeline

This report splits the data using 6-2-2 method as mentioned above and preprocesses the data, then calculates the test score. Here, it uses accuracy score as its evaluation metric since it is a classification problem, and we want to measure the ratio of the sum of true positives (TP) and true negatives (TN) out of all the predictions made, for better medical no-shows prediction to reduce the financial cost. It repeats this method 5 times for 5 different random states, and it returns the 5 best models and their 5 test scores for each algorithm, and collects 5 test sets. For non-deterministic ML methods, we calculate the standard deviation of test scores, which demonstrates how much uncertainty in calculations. XGBoost uses early-stopping (round 50).

It utilizes 6 different machine learning models: Lasso (L1), Ridge (L2), Elastic Net, Random Forest, KNN, and XGBoost. The tuned parameters and values are listed as follows.

Algorithms	Tuned Parameters	Tuned Values
Lasso (L1)	C	10 numbers spaced evenly on a log scale from -5 to 5 with base 10
Ridge (L2)	C	10 numbers spaced evenly on a log scale from -5 to 5 with base 10
Elastic Net	l1 ratio	0.1, 0.3, 0.5, 0.7, 0.9
Random Forest	max_depth max_features	1, 3, 10, 30, 100 0.25, 0.5, 0.75, 1.0
KNN	n_neighbors weights	1,2,5,10,50 uniform, distance
XGBoost	max_depth	1, 5, 10, 15, 20

4 Result

The baseline score is 0.79807, which is predicting all to show-up.

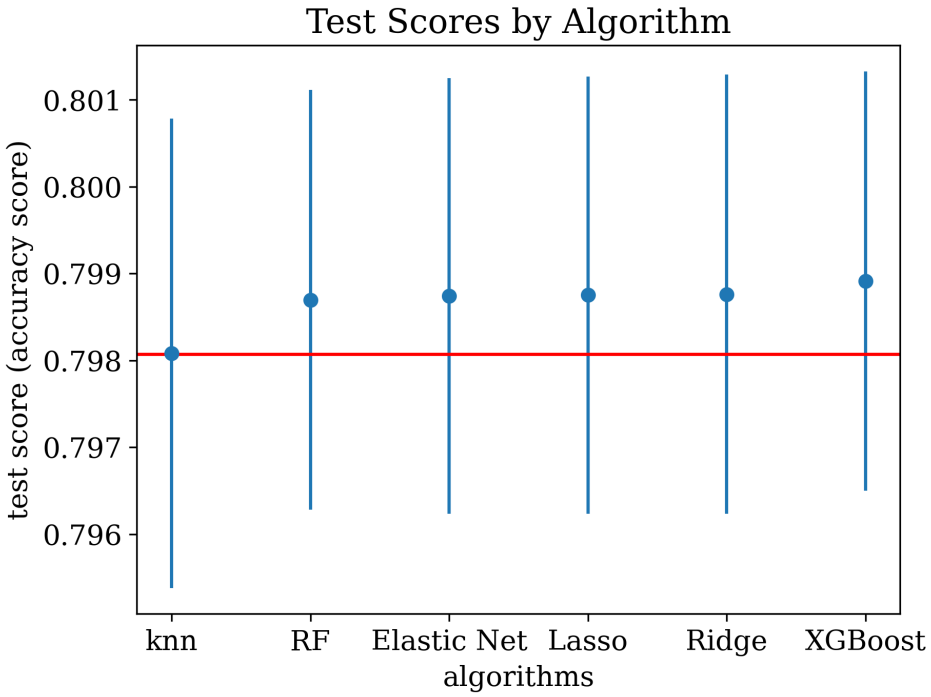

Figure 6: Test Scores by Algorithm

From the above plot, we can conclude that XGBoost has the best mean test score (0.79891), Random forest is the second worst (0.79870) KNN has the worst one(0.79808), which is slightly better than the baseline score. Elastic NetLasso, Lasso, and Ridge have smiliar mean accuracy scores of 0.79874, 7.79875, and 0.79876. Also, XGBoost has the smallest standard deviation (0.00241), Random forest has the second smallest one(0.00242), and KNN has the largest one (0.00270). Elastic Net, Lasso, and Ridge, have similar standard deviations, which are 0.00251, 0.00252, and 0.00253. For standard deviations above the baseline, KNN is 0.005, Random Forest is 0.261, Elastic Next is 0.269, Lasso is 0.272, Ridge is 0.274, and XGBoost is 0.351.

Overall, XGBoost is most predictive, with the best test score and standard deviation.

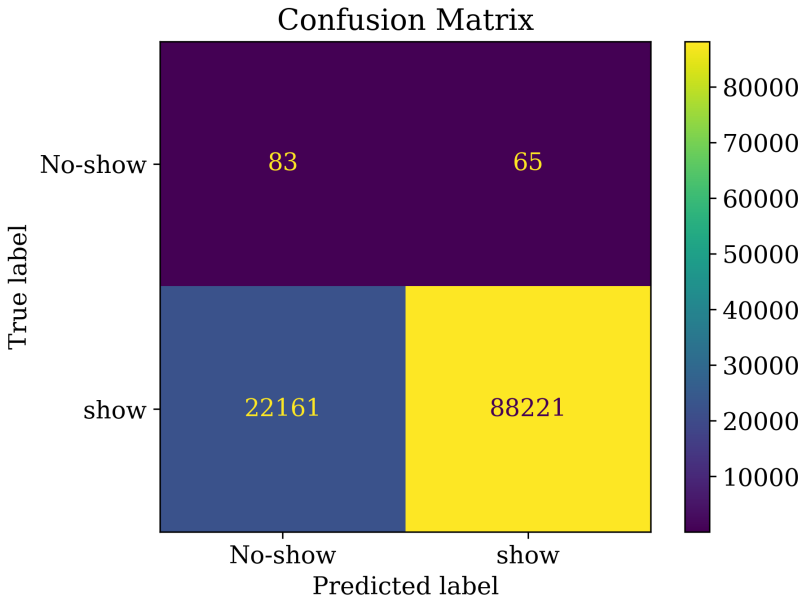

Figure 7: Confusion Matrix

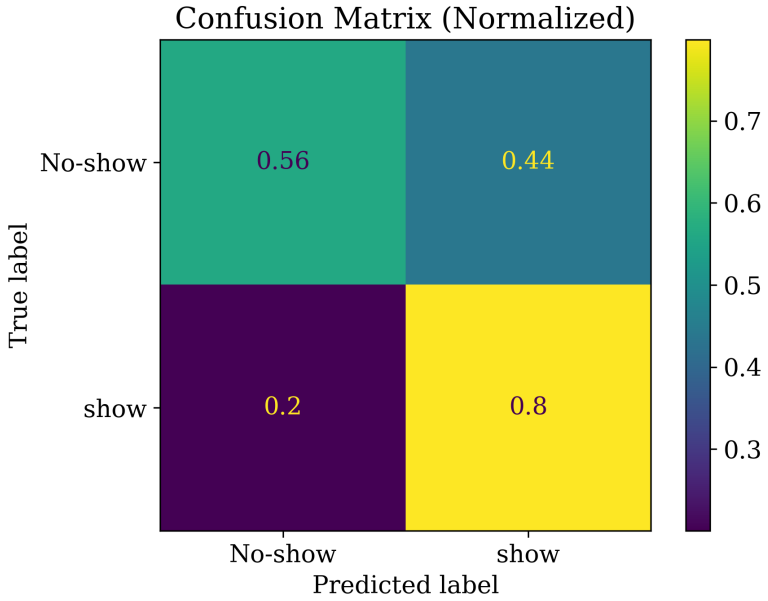

Figure 8: Confusion Matrix (Normalized)

The confusion matrix of 5 XGBoost test sets shows the TP is 83, FP is 65, FN is 22161, and TN is 88221, and the normalized one shows the TP is 0.56, FP is 0.44, FN is 0.2, and TN is 0.8. Here, true positive (TP) means when the model correctly predicts the positive class (medical no-show), and true negative (TN) means the model correctly predicts the negative class (medical show). Therefore, we can conclude that it performs well in predicting medical show-ups, but not very well on medical no-shows.

Moreover, the f1 score is 0.00741, the f0.5 score is 0.01817, and the f2 score is 0.00466. Here, all f scores are low due to low TP and FP compared to TN and FN. Therefore, we stick with accuracy score to maximize both TP and TN since the data is not imbalanced.

Next, we choose one XGBoost model(max_depth = 10, n_estimators = 200, learning_rate = 0.01, colsample_bytree = 0.9, subsample = 0.66). It performs best on train and test data. It has an accuracy score of 0.79988 on train set, and 0.80146 on test score. Also, we study its global and local importances.

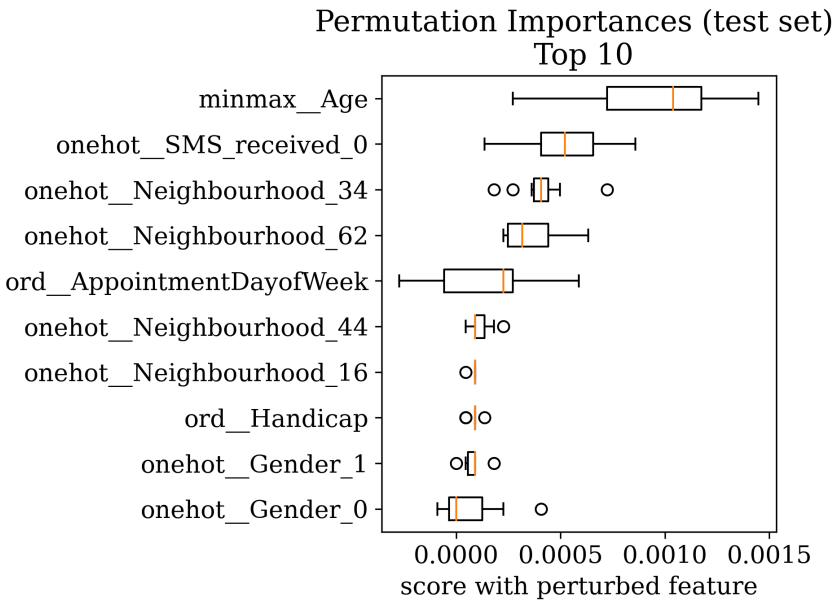

Figure 9: Permutation Importances (test set) Top 10

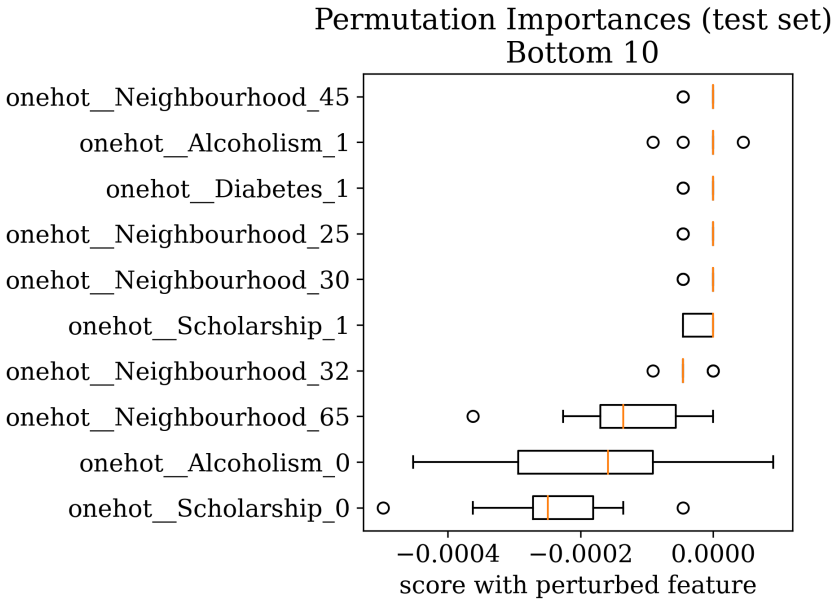

Figure 10: Permutation Importances (test set) Bottom 10

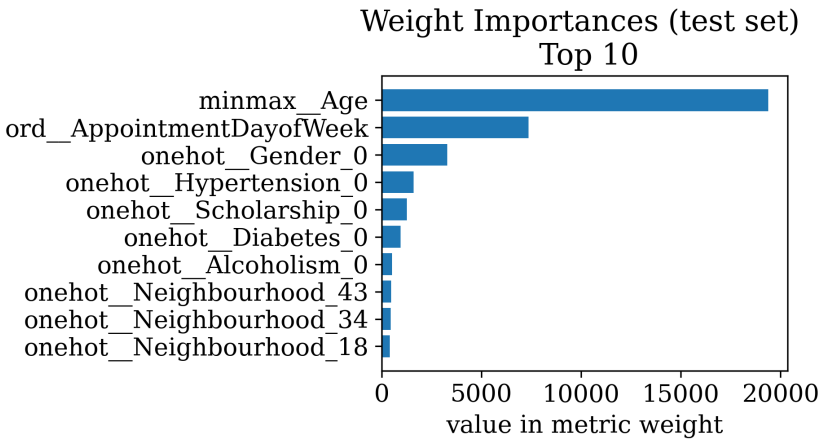

Figure 11: Weight Importances (test set) Top 10

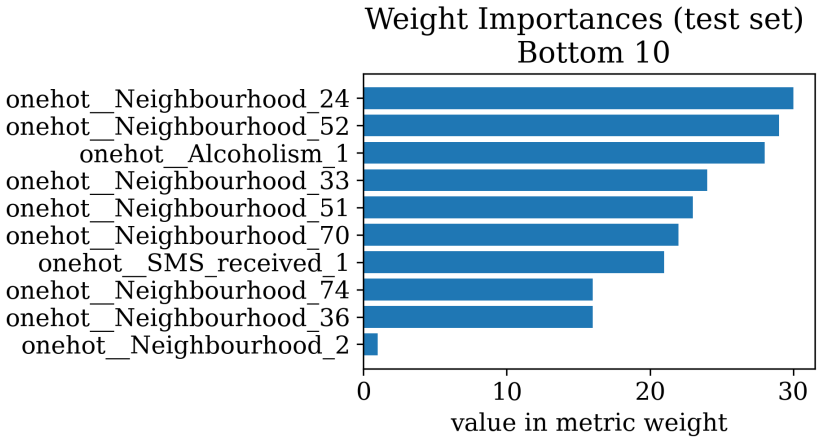

Figure 12: Weight Importances (test set) Bottom 10

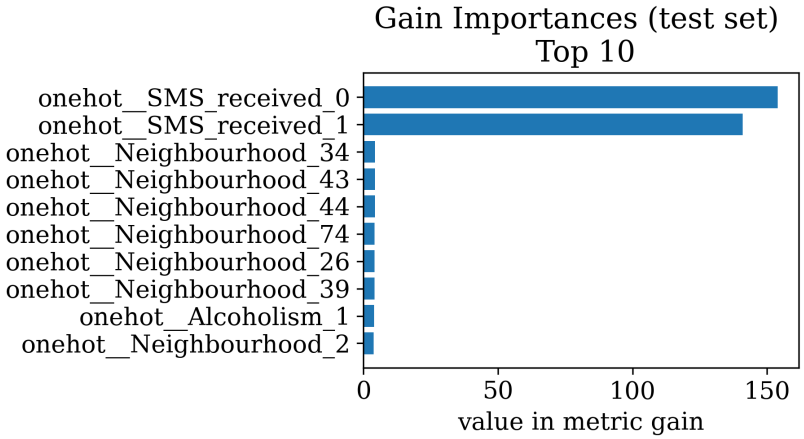

Figure 13: Gain Importances (test set) Top 10

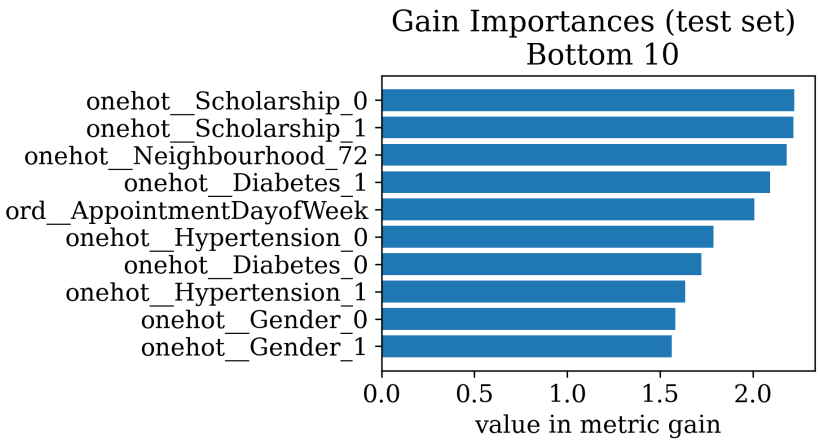

Figure 14: Gain Importances (test set) Bottom 10

SHAP Global Importances (test set) Top 10

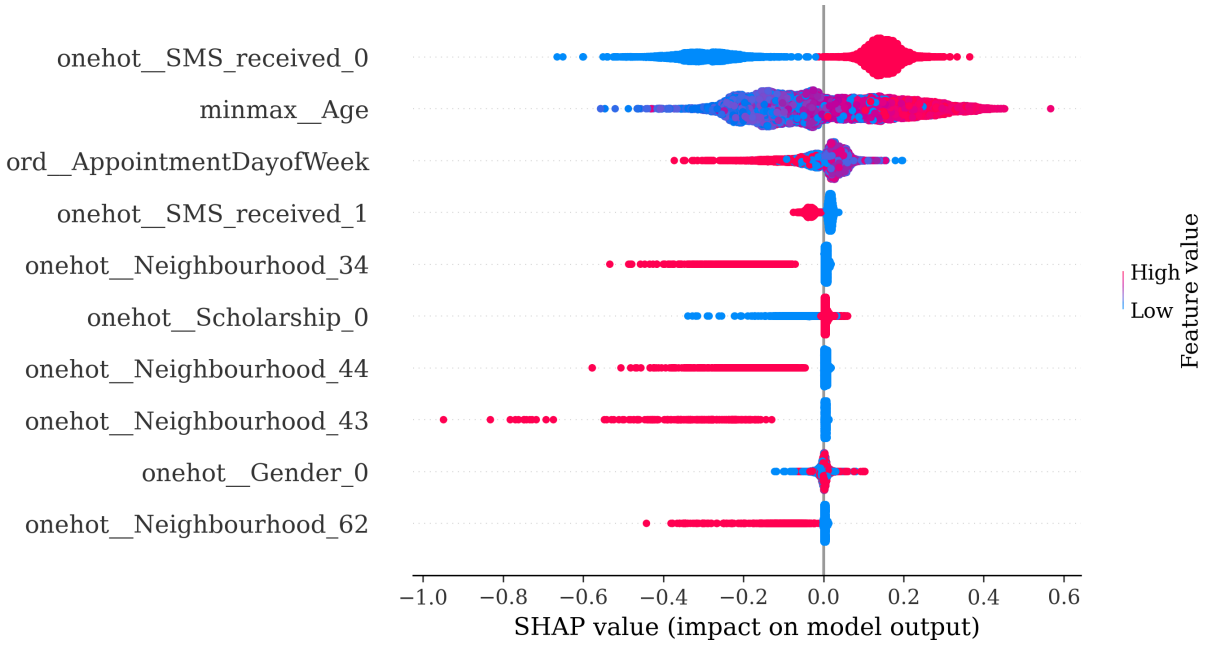

Figure 15: SHAP Global Importances (test set) Top 10

SHAP Global Importances (test set) Bottom 10

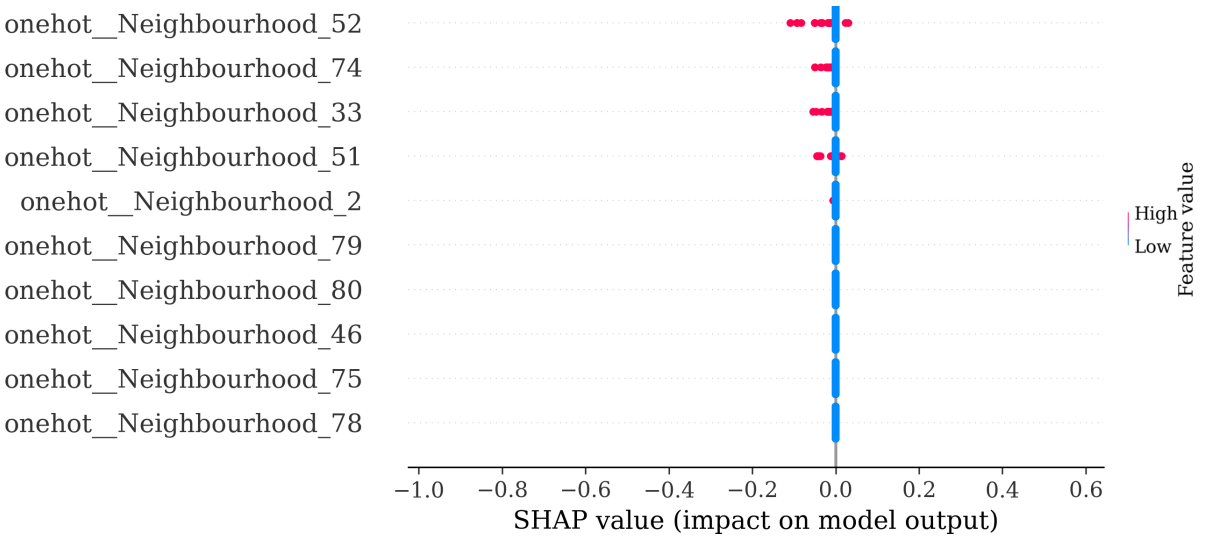

Figure 16: SHAP Global Importances (test set) Bottom 10

We calculate permutation, Weight, Gain, and SHAP as global importance. Here, we need to consider both Wight and Gain. Neighbourhood and Age have more values than other categorical features, which means they should perform better in Weight. However, for the rest categorical features, they have close small number of possible values, which makes Weight a good metric in this report. Moreover, Gain interprets the relative importance of each feature. Here, some features frequently appearing on top 5 are onehot_SMS_received_0, minmax_Age, ord_AppointmentDayOfWeek, and the least frequent important one is onehot_Scholarship_1.

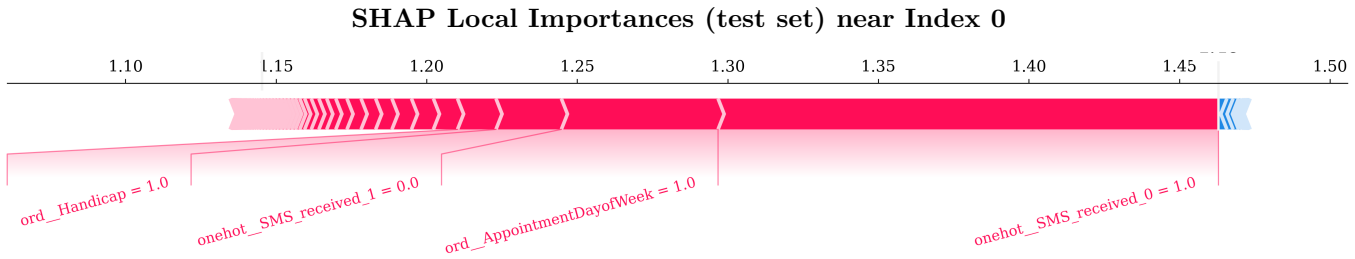

Figure 17: SHAP Local Importances (test set) near Index 0

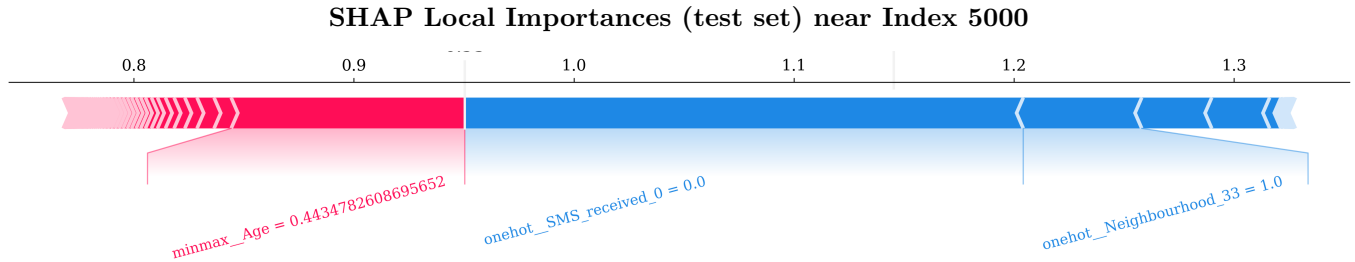

Figure 18: SHAP Local Importances (test set) near Index 5000

SHAP Local Importances (test set) near Index 15000

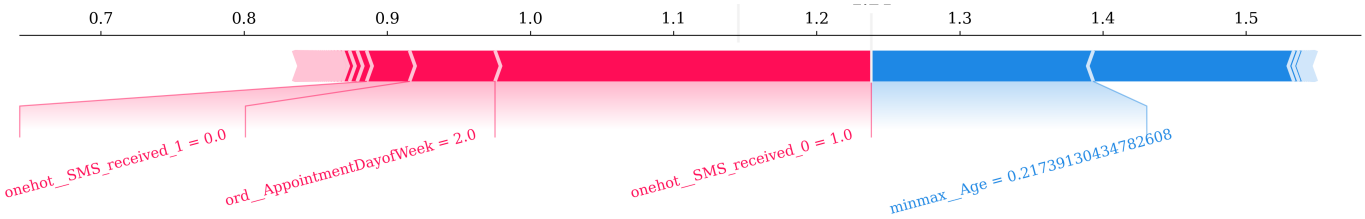

Figure 19: SHAP Local Importances (test set) near Index 15000

For local importance, we choose indices 0, 5000, and 15000, and we found a similar result: onehot_SMS_received_0, minmax_Age, ord_AppointmentDayOfWeek are the most important.

The result is important since it shows us the way to predict medical no-shows. For example, we can send more SMS notifications to make patients show up. Also, it is interesting because it shows age and AppointmentDayOfWeek can be important factors to consider. Especially for AppointmentDayOfWeek, as mentioned above, there is no holiday or Sunday, and only a few observations on Saturday. It is worthwhile to study further how AppointmentDayOfWeek affects medical no-shows.

5 Outlook

This report does not use sophisticated Feature Engineering, ignores some correlation, and extra data can be used for better model performance and interpretability.

For Feature Engineering, we can further categorize Neighborhood by finding some similar aspects. Since the global importance shows some Neighborhoods are important, it is worth examining how Neighborhoods affect medical no-shows.

Also, Diabetes and Hypertension could have a correlation. Research Midha et al. 2015 shows that the risk of developing hypertension is 1.5-2.0 times higher in diabetic patients in contrast to nondiabetic patients. Therefore, operations such as deleting one feature may change the model's performance.

Moreover, extra data could be beneficial, such as whether a patient shows up for an appointment at other medical facilities and weather information on appointment's days, given the reasons behind a person having a medical no-show are sometimes complicated.

References

Helmonds, Joep (June 2018). "Predicting no-shows in Brazilian primary care". LIACS. URL: https://theses.liacs.nl/pdf/2017-2018-HelmondsJoep.pdf.

Hoppen, Joni (2016). "Medical Appointment No Shows, version 5". URL: https://www.kaggle.com/datasets/joniarroba/noshowappointments.

Midha, Tanu et al. (2015). "Correlation between hypertension and hyperglycemia among young adults in India". In: *World Journal of Clinical Cases: WJCC* 3.2, p. 171.

PANDIRI, SAMRAT (May 2019). *Predict show/noshow - eda+visualization+model*. URL: https://www.kaggle.com/code/samratp/predict-show-noshow-eda-visualization-model.