Appointment No-Shows Report

Dovid Burns

3/13/18

Springboard Capstone One

1. Define the Problem

Why do patients miss their appointments? This simple question underpins one of the most disruptive challenges facing medical offices around the globe. Data analytics lends powerful tools to be able to predict the likelihood of a patient no-show. This information could be adapted by individual practices and hospitals around the globe, allowing for local healthcare networks to make significant staffing adjustments. An example of improved staffing practices includes overbooking certain time blocks or days with larger numbers of patients who have higher probabilities of failing to present for their appointments. This would allow the healthcare provider to schedule fewer staff members on days or time blocks with patients who are likely to arrive for their appointments, thereby potentially averting significant financial losses to healthcare institutions due to a problem as simple as patient no-shows.

2. Identify Client

This study uses data from over 110,000 public healthcare appointments in Vitória, Espírito Santo, Brazil (acquired from kaggle.com datasets), and attempts to indicate the top factors predicting patient no-show. Our potential clients are healthcare providers in Brazil who are directly negatively affected by patients not showing up to their appointment.

3. Describe Data Set

Each row of data contains information from a single appointment. There are exactly 14 columns and 110,527 rows that were in the public record. The first column is PatientId, which is recorded as datatype float and is not unique as there are some patients with multiple appointments. For this analysis, PatientId will be ignored as we will examine each appointment separately. The second column is AppointmentID. Information in this column is recorded as numbers of data type int64. This is a unique identifier for each row and this is what will be used as our index for machine learning. The third column is Gender, which is recorded as datatype string containing either 'M' or 'F'. The fourth column is ScheduledDay, which is imported as a datetime and is both the date and time when the patient scheduled the future appointment. The fifth column is AppointmentDay, which is imported as a datetime and is simply the calendar date of the scheduled appointment. The data includes appointments ranging from 2016-04-29 through 2016-06-08. Unfortunately, the time of the scheduled appointment was not included in the data set. The sixth column is Age, which is recorded as data type int64 and contains the age (in years) of the patient at the time of the appointment. The age range is 0 to 115. The seventh column is Neighbourhood, which is recorded as data type string, and refers to the neighborhood where the appointment took place. There are eighty-one distinct neighborhoods in the data set. The eighth column is Scholarship, which is recorded as datatype int64 of either 0 or 1, where a 1 indicates that the patient's family qualified for a government scholarship based on financial need. The ninth column is Hipertension, which is recorded as data type int64 of either 0 or 1, where a 1 represents the patient having hypertension. The tenth column is Diabetes, which is recorded as datatype int64 of either 0 or 1, where a 1 represents the patient having diabetes. The eleventh column is Alcoholism, which is recorded as datatype int64 of either 0 or 1, where a 1 represents the patient having alcoholism. The twelfth column is Handicap, which is recorded as datatype int64 of either 0, 1, 2 or 3, where the value represents the number of handicaps that the patient has. The thirteenth column is SMS_received, which is recorded as datatype int64 of either 0 or 1, where a 1 represents the patient having received a text message reminder about their upcoming appointment. The fourteenth and final column is No-show, which is data type string of either Yes or No, where a Yes affirms that the patient failed to arrive for their appointment.

The data set that we are working with is relatively clean. We ran the pandas command .info() which showed no NaN's in the dataframe. Looking at each row we did find some problems with individual data points. In the age column we found one age of -1. This is definitely an error, thus we have removed it from the data set.

We used the appointment date column to find out the day of the week that the appointments took place using .dt.weekday_name in order to assess whether there exists a correlation between the day of the week and appointment no-shows. We found that most of the appointments were scheduled to occur Monday through Friday, with only a few exceptions. There were 39 appointments scheduled to take place on Saturday, and, since the next least-common scheduling day had over 17,000 appointments in our dataset, we have decided to exclude the Saturday appointments as outliers.

We examined the time that passed between when the client scheduled the appointment and when the appointment took place. We called this new variable wait. Strangely, there were some wait time spans with negative values, meaning the appointment was scheduled after the appointment was due to have taken place. Since this is impossible, and there are only five appointments in which this occurred, we have removed these from the data set. It is interesting to note that all five of these appointments were no-shows.

Upon examining the Neighborhood data, we discovered that some neighborhoods had very few appointments. These regions were treated as outliers from the dataset. We will only keep neighborhoods with greater than

50 appointments so this excludes appointments from ILHA DO BOI, ILHA DO FRADE, AEROPORTO, ILHAS OCEÂNICAS DE TRINDADE, and PARQUE INDUSTRIAL. These neighborhoods together comprise less than sixty appointments, so we simply removed them from our dataset without a significant loss of data.

Our data set now has two continuous variables, age and wait. To statistically analyze and model with them, manual binning was required. To create the bin ranges, we looked at trends of no-show rate as age and wait time increased. Finding natural breaks in the no-show rate gave the following six bins for age in years: 0-3, 4-7, 8-27, 28-40, 41-60, and greater than 60. The amount of appointments in each age bin are as follow: 8943 patients were 0-3 years of age, 5735 patients were 4-7 years of age, 26566 patients were 8-27 years of age, 19398 patients were 28-40 years of age, 19732 patients were 41-60 years of age, and 30052 patients were 61+ years of age. A similar process for wait time gave the following 5 groups of wait times in days: 0, 1, 2-4, 5-9 and greater than ten days. The number of appointments in each wait bin are as follow: 38536 patients had same day appointments, 5206 patients had a wait of 1 day, 14723 patients had a wait of 2-4 days, 16145 patients had a wait of 5-9 days, and 35816 patients had a wait of 10 or more days.

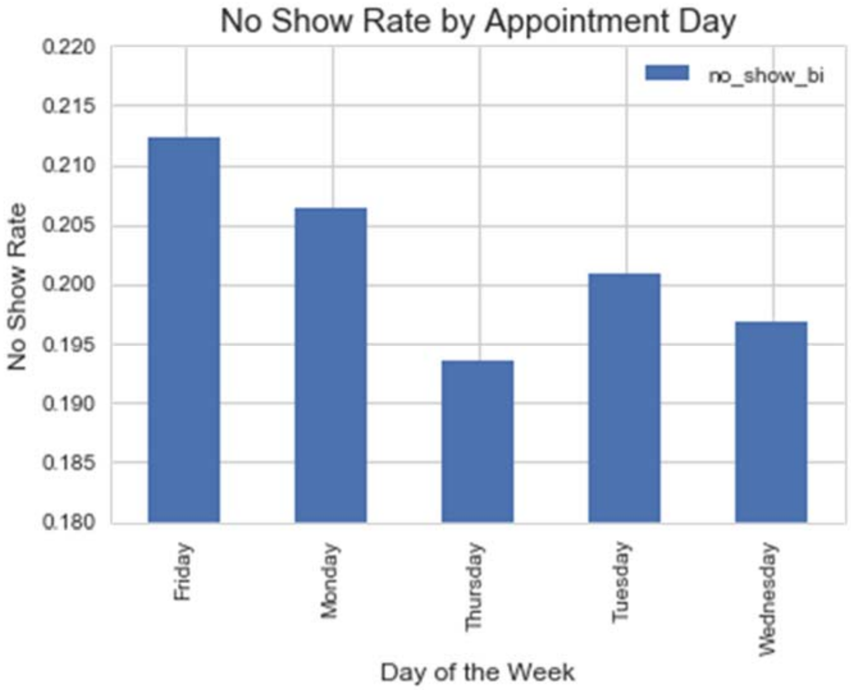
4. Other Potential Data

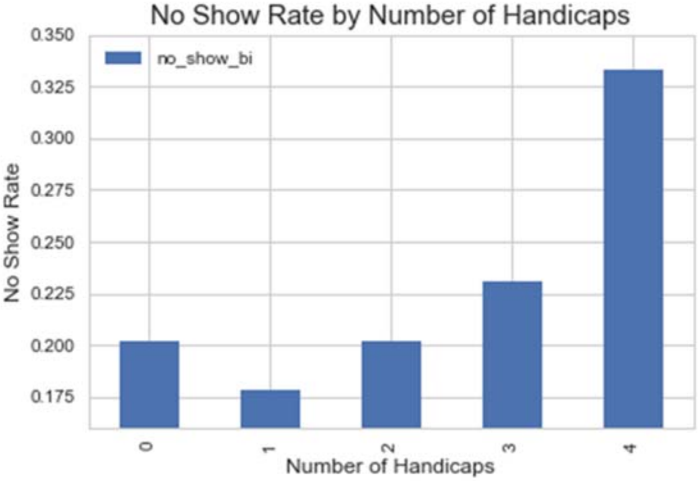
While our data set does contain valuable data, there is further information that would most likely make our analysis stronger. It would be helpful to know the type of appointment, e.g. routine physical, sick-visit with primary doctor, specialist visit or follow-up. Additionally, knowing the time of the appointment could also be a useful factor in predicting likelihood of no-show. Further health background data, such as patient pregnancy, AIDS, cancer, etc. would potentially play a role in patient no-show as well. Knowing the type of insurance that the patient has could also be a predictor in no-show.

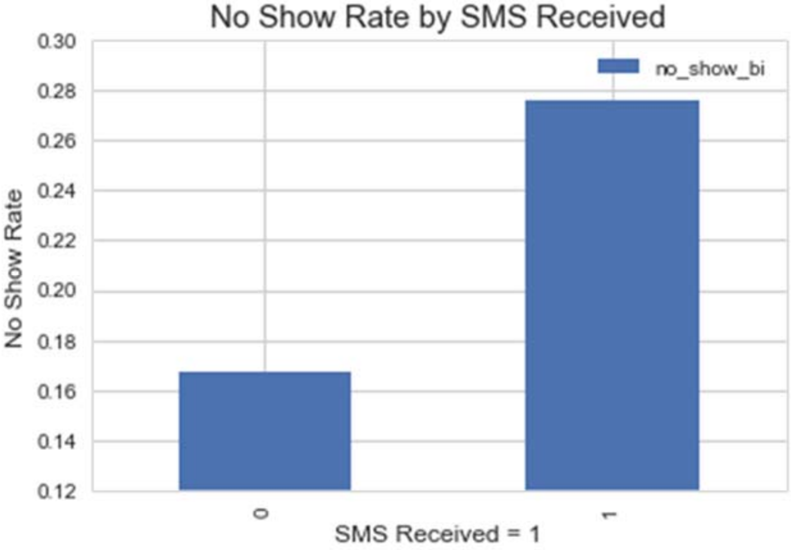
Further, knowing the approval ratings of the hospitals and/or doctors might be helpful in predicting no-show. Lastly, having a longer time frame of appointment data to work on would most certainly increase the predicting power.

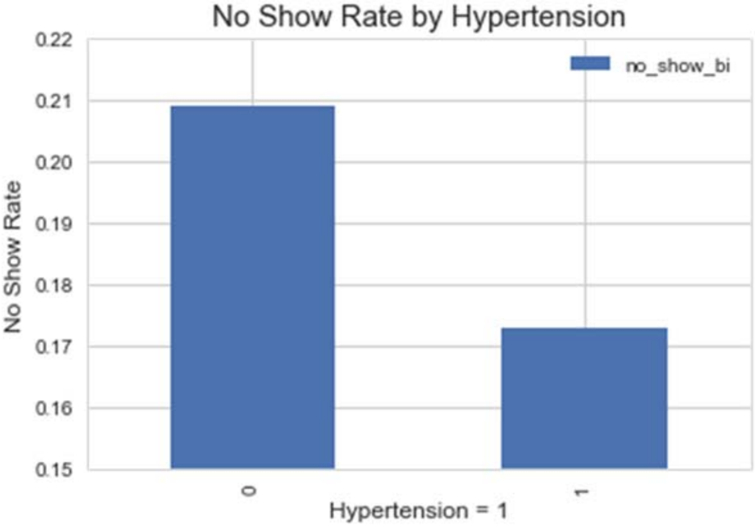
5. Initial Findings

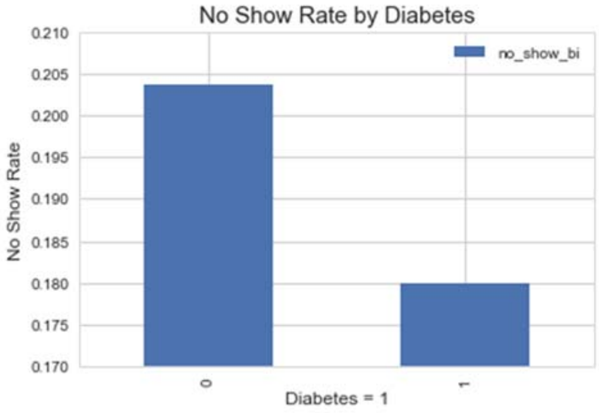
Examination of the data story telling showed that the leading indicators of missing an appointment are appointment day, patient handicap, hypertension, diabetes, scholarship receipt, text message reminder, age binned, time between appointment scheduled to appointment day, and neighborhood. Our initial survey indicated that there did not seem to be a correlation between missing an appointment and alcoholism or gender. To further verify these claims, we ran t-tests for independence and chi-squared tests between these variables. To check for correlation between the independent variables above, we created a correlation heat map of all the numerical variables. Below are plots showing the for select variables.

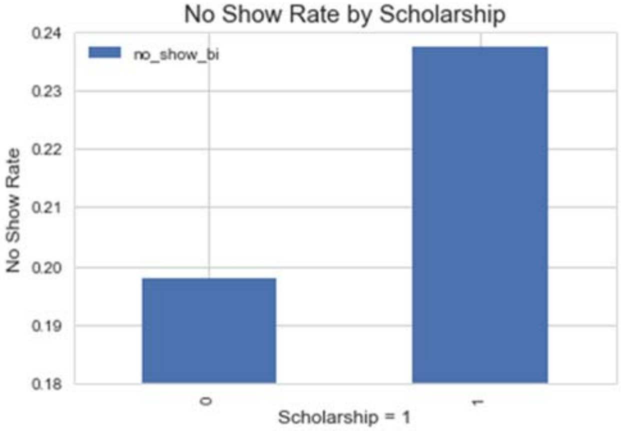


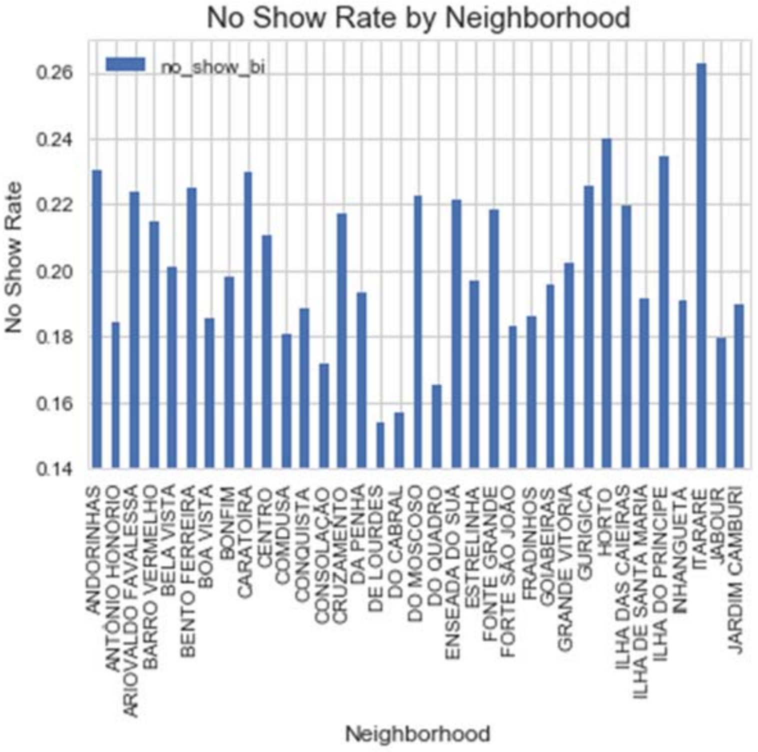

No Show Rate by Number of Handicaps
no_show_bi
No Show Rate
0.350
0.325
0.300
0.275
0.250
0.225
0.200
0.175
0
1
2
3
4
Number of Handicaps


No Show Rate by SMS Received
no_show_bi
No Show Rate
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0.16
0.14
0.12
0
1
SMS Received = 1


No Show Rate by Hypertension
no_show_bi
No Show Rate
0.22
0.21
0.20
0.19
0.18
0.17
0.16
0.15
0
1
Hypertension = 1


No Show Rate by Diabetes
no_show_bi
No Show Rate
0.210
0.205
0.200
0.195
0.190
0.185
0.180
0.175
0.170
0
1
Diabetes = 1


No Show Rate by Scholarship
no_show_bi
No Show Rate
0.24
0.23
0.22
0.21
0.20
0.19
0.18
0
1
Scholarship = 1


No Show Rate by Neighborhood
no_show_bi
No Show Rate
0.26
0.24
0.22
0.20
0.18
0.16
0.14
ANDORINHAS
ANTÔNIO HONÓRIO
ARIOVALDO FAVALESSA
BARRO VERMELHO
BELA VISTA
BENTO FERREIRA
BOA VISTA
BONFIM
CARATOÍRA
CENTRO
COMDUSA
CONQUISTA
CONSOLAÇÃO
CRUZAMENTO
DA PENHA
DE LOURDES
DO CABRAL
DO MOSCOSO
DO QUADRO
ENSEADA DO SUÁ
ESTRELINHA
FONTE GRANDE
FORTE SÃO JOÃO
FRADINHOS
GOIABEIRAS
GRANDE VITÓRIA
GURIGICA
HORTO
ILHA DAS CAIEIRAS
ILHA DE SANTA MARIA
ILHA DO PRÍNCIPE
INHANGUETÁ
ITARARÉ
JABOUR
JARDIM CAMBURI
Neighborhood

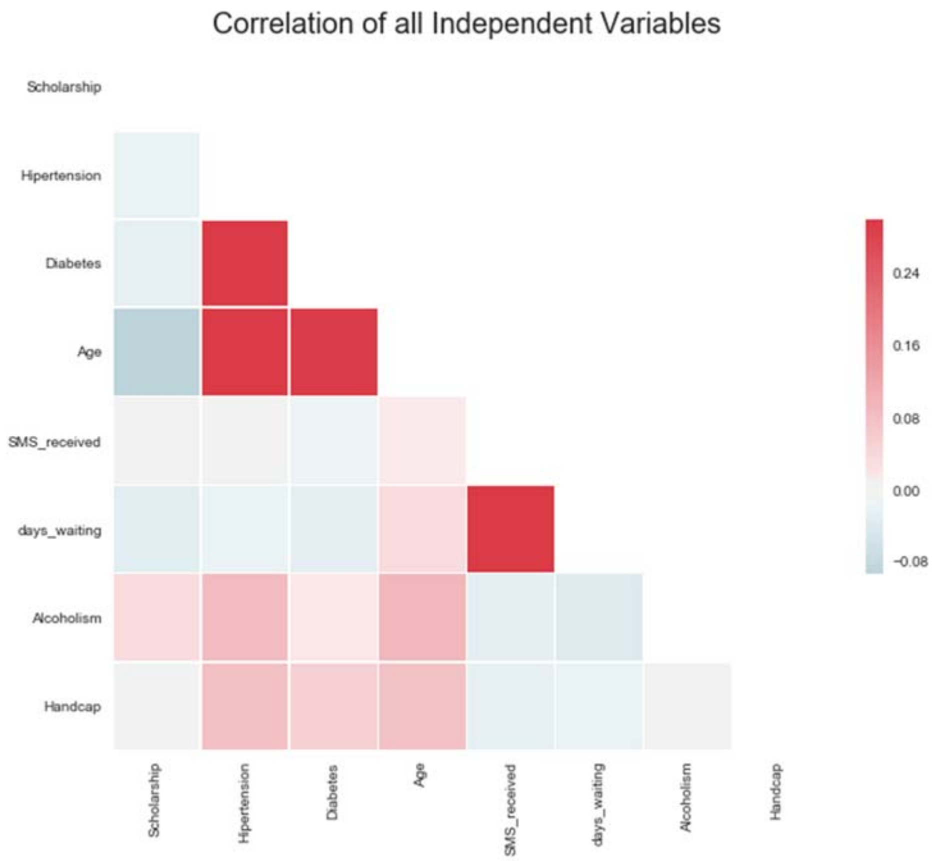
Using a chi-squared contingency test built into python, we found the following p-values for correlation with no-show rate:

Feature	Correlation with no-show: p-value
appointment day	1.8e-5
handicap	0.11
hypertension	2.4e-32
diabetes	4.9e-7
scholarship	3.2e-22
received a text reminder	0.0
age binned	6e-174
time between appointment scheduled to appointment day binned	0.0
neighborhood	1.3e-60
gender	0.19
alcoholism	0.98

Given the previous analysis of mean values, the p-values that we expected to refute any correlation was p-value for gender = 0.19 and alcoholism = 0.98. We have only accepted variables as statistically significant in predicting no-shows that have a p-value of less than 0.01. Therefore, appointment day, hypertension, diabetes, scholarship, received a text reminder, age binned, time between appointment scheduled to appointment day binned, and neighborhood are all good predictors of a patient not showing up for an appointment. The difference in no-show rates between patients with and without these features are as follows:

Feature	Percent Difference No Show Rate
appointment day	<= 1.9%
hypertension	3.6%
diabetes	2.4%
scholarship	3.9%
received a text reminder	10.9%
age binned	<= 9.8%
time between appointment scheduled to appointment day binned	<= 27.8%
neighborhood	<= 14%

The strongest indicators based on smallest p-value and largest difference in no-show rates are received a text, age binned, time between appointments scheduled to appointment day binned and neighborhood.



The heat map above depicts some correlation between several independent variables. We see a positive correlation between diabetes and hypertension. Likewise, we also note a positive correlation between age and hypertension, age and diabetes and a slight negative correlation between age and scholarship. Lastly, the time that elapses between scheduling and the appointment date is correlated with receiving a reminder text. As the highest correlation found between these variables is only 0.24, we can confidently use all the important variables without a correlation bias.

6. MACHINE LEARNING

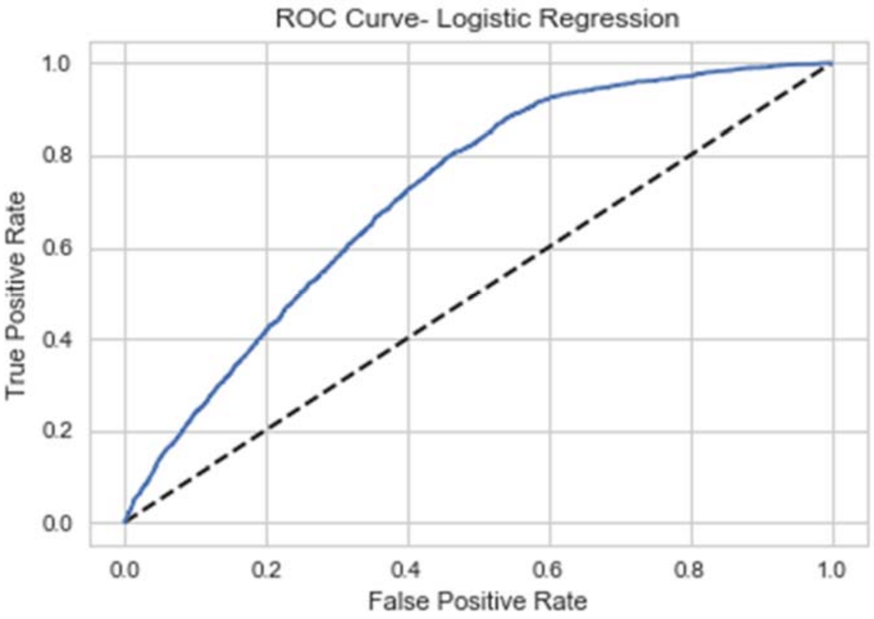
a) Data ML Pre-Processing

To run machine learning algorithms on this data, the information needs to be translated into a format that can be interpreted by the algorithms. Each column that we are using must be in either a binary form of 0 or 1. While some columns were already in this form—e.g. SMS_received—the rest needed to be converted from a categorical variable to binary values with the pandas get_dummies function. We dropped the first dummy column to prevent variable correlation in the columns. Next, we split the data and targets into two groups, a training group and a testing group, using train_test_split. At this point we ran machine learning algorithms on the dataset, but we found that we were not getting reliable predictions. To increase the accuracy of the algorithms, we split the data again and then ran SMOTE—Synthetic Minority Over-Sampling Technique—on the second training segment. For K-Nearest Neighbors and Random Forest we used grid search to tune the hyperparameters. To determine the effectiveness of the models in no-show prediction, we ran f1-tests. We used a custom threshold to maximize f1-scores for no-show prediction. Below are the scores for different configurations of these machine learning algorithms.

b) Logistic Regression CV Classifier

We first implemented LogisticREgressionCV from sklearn's linear model using only the top features of our dataset. These top features determined previously were SMS_received, Age_Binned, Wait_Binned and Neighborhood. Our initial model predicted that all patients would arrive for their appointments. Using 10-fold cross validation and f1 scoring gave a few predictions of not showing up and a f1-score of 0.01. This is because the predict function uses a threshold of 0.5 for predictions, but our data has a natural no-show rate of approximately 0.2. We used predict_proba() to implement different thresholds and found a threshold of greater than 0.21 gave the highest f1-score of 0.446. Additionally, we had an AUC (Area Under the Receiver Operating Characteristic curve) of 0.725. To further improve our model's performance, we split the data again and ran the SMOTE algorithm from imblearn's over sampling package to balance the number of patients who showed up and did not show up to their appointment. Here we found the same max f1-score of 0.446 but now with a threshold of 0.5. Additionally, after applying SMOTE we the AUC decreased slightly to 0.724.

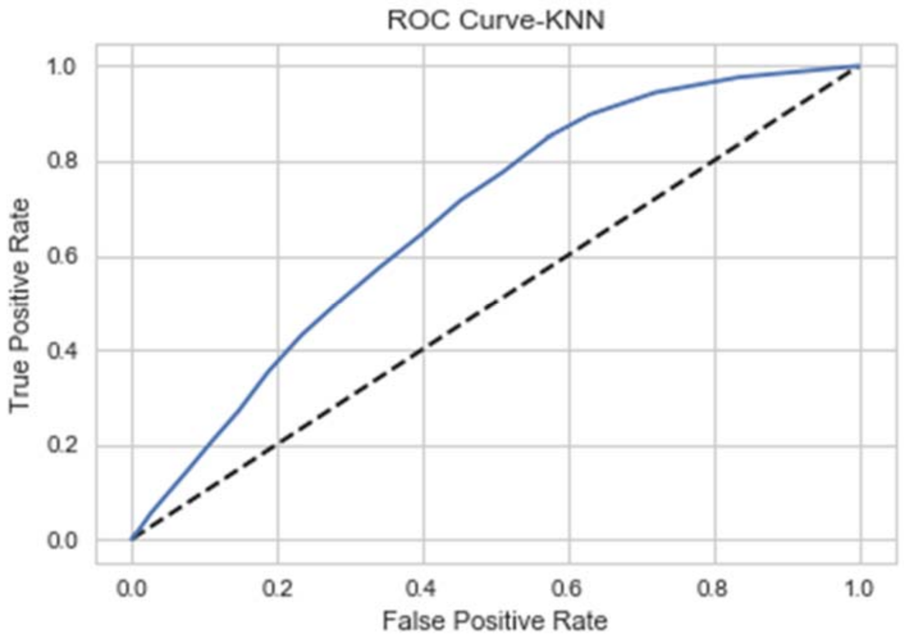
We then included the following additional features of statistical significance: scholarship, hypertension, diabetes and appointment day. We ran SMOTE algorithm as again on this larger dataset. With the balanced larger dataset, we found a max f1-score of 0.439 with a threshold of greater than 0.51. The AUC was 0.714.



c) K-Nearest Neighbors Classifier

The next algorithm we utilized was the K-Neighbors Classifier. We first tried this with an 80%-train, 20%-test split. We used only the top features of the data. This resulted in an AUC of 0.65 and a f1-score of 0.245. A grid search found that 18 neighbors seemed to be ideal. When we implemented a threshold, we could get a maximum f1-score of 0.424 with a threshold of 0.22. We applied SMOTE on just the top variables which resulted in AUC of 0.69. The balanced data had a f1-score with no threshold of 0.347. When we applied a threshold of 0.22 we found a max f1-score of 0.431.

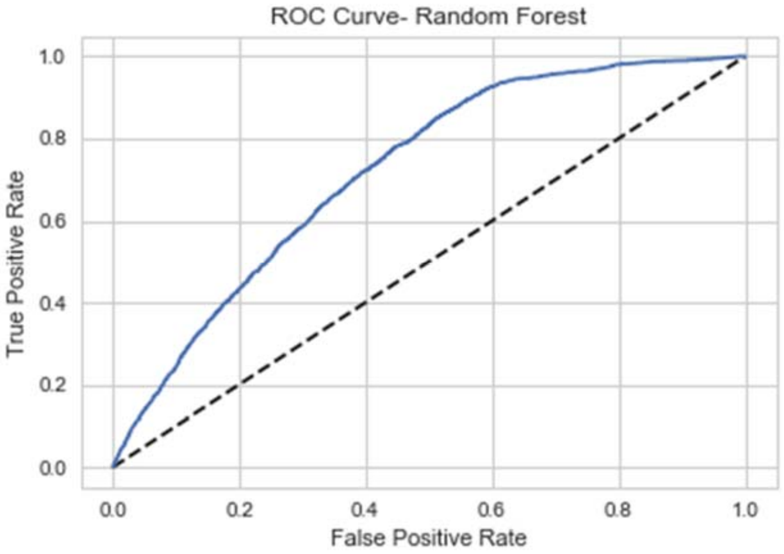
We then included all the variables and rebalanced the data using SMOTE. We could get a maximum f1-score of 0.414 with a threshold of 0.22 and 18 neighbors. The data generated an AUC of 0.68.



d) Random Forest Classifier

We first ran the RandomForestClassifier() on our top categories and found a small f1-score of 0.11, as almost everyone was classified as having successfully arrived for their appointments. The AUC was 0.704. Running a grid search cv for hyperparameter tuning advised using max_depth= 30, min_samples_leaf= 1, n_estimators= 9. This change increased the AUC to 0.716; however, without balancing the data, the f1-score dropped to 0.05. Applying a threshold of over 0.22 to the predic_proba function increased the f1-score to 0.434.

We rebalanced the data and included the remaining variables. We tuned the hyperparameters with grid search and a manual method which resulted in an AUC of 0.72 and a maximum f1-score of 0.44 with a threshold of greater than 0.43. The best hyperparameters we found were a maximum depth of 35 and min_samples_leaf of 12 using 26 estimators.



7. Conclusion

We found the best predictions using logistic regression cv, followed by random forest, then, followed by k-nearest neighbors. However, these analyses were all very similar in their overall effectiveness. Using these algorithms, our client can now predict which appointments are more likely to be missed. Healthcare groups and physicians' offices will be able use this information to schedule more patients or to bring in fewer practitioners on days when more patients are predicted to fail to arrive for their appointments. Additionally, medical offices can use this algorithm to implement different effective reminder methods for patients who are predicted to no-show to increase the likelihood that these clients do not actually miss their appointments. Another recommendation to medical office staff would be to call likely no-show patients early on the date of their appointment to find out if the patients anticipate arriving to their appointments. The appointments freed by clients who cancel in this manner could then be offered to a back-up waiting list of patients who would like to take last-minute appointments. Alternatively, if there are days where the algorithm predicts very unlikely no-shows for the scheduled appointments, more doctors could be brought in to help with back up.