Prognostication of Patient No-Show at Health Clinics using Contrasting Classifiers

Post Graduate Program in Data Science Engineering
Location: Bangalore
Batch: PGPDSE-Jan'22

Submitted by:
Anikesh Aich
Ashwin Venkatramanan S
Deepika Pandita
Mahipal Bhagat
Yogeshwara S M

Mentored by:

Mr. Ashish Sharma

ACKNOWLEDGEMENT

We take this wonderful opportunity to express our most sincere gratitude to all those who helped and supported us towards the completion of this capstone project successfully. We are beholden to our project mentor Mr. Ashish Sharma for his expert guidance and assistance throughout the course of this project.

Our most sincere thanks to Mrs. Vidya, our academic counsellor, Great learning for their time, consistent support and guidance towards completion of the project.

We would like to thank Ms. Gowthami, our Program manager, Great Learning, Mr. Manvendra Singh, Teaching assistant, Great Learning and Mr. Bhushan, Teaching assistant, Great Learning who have always been pushing, motivating us and clarifying any doubts upon progress through our project. We also would like thank Great Learning and Great Lakes institute of management for the opportunity presented to us in doing this capstone project.

Our dearest thanks to our families and friends for their moral, untiring support which took us to successfully completing our objectives.

Anikesh Aich
Ashwin Venkatramanan S
Deepika Pandita
Mahipal Bhagat
Yogeshwara S M

DECLARATION

We hereby declare that this submission entitled '**Prognostication of Patient No Show at Health Clinics using Contrasting Classifiers'** is our own work and that to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any degree or diploma of the university or any other institute of higher learning.

The information derived from the literature has been duly acknowledged in list of references that are provided.

Date: 15th July, 2022

Place: Great Learning, Bangalore

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The Healthcare industry has seen an exponential increase in demand and cost pressures in the past few years. Despite huge efforts, there are underlying issues that contribute to inevitable higher costs and poor utilization of resources.

One of the major issues faced by healthcare providers is the Patient No-show at Hospitals and Clinics. No-Show occurs when a patient is unable to attend a scheduled appointment without any prior intimation to the healthcare provider. This leads to under-utilized time slots and resources which can affect the business in a negative manner. According to a study in the United States. It is observed that 30% of patients miss their scheduled appointments leading to a loss of a whopping $150 billion dollars.

There is still limited research that suggests a comprehensive impact of no-shows along with the underlying reasons for no-show cases. Through this case study we aim to analyze the different factors that contribute to the no-shows in a particular city of Brazil.

Industry Review

Healthcare organisations face the issue of no-shows that ranges from 3-80% across the world. The chances of these patients who have a history of no-shows needing treatment with chronic care and services of emergency eventually increases considerably that leads to high medical costs and wasteful use of hospital resources. In resource intensive areas like Radiology and Surgeries, this kind of underutilization can lead to steep financial burdens on the healthcare providers. Further, they can inevitably affect the patient's health with a delay in treatment. This can also lead to poor customer satisfaction and negative reviews.

Several studies from the past have shown us these missed medical appointments are directly correlated with mortality. We also came across studies that have shown that patients with lower socio-economic status, prior no-shows and insurance services tend to more frequently miss their appointments. Other studies have thrown light on factors that influence no-show such as Age, Gender, Quality of service, Number of earlier appointments. Another case study showed this phenomenon common among male patients and patients belonging to lower income families.

Understanding the data can help us to enhance the future prediction of any impending no-shows so that appropriate measures can be taken. This can be creating a Digital Booking system along with providing alerts to the patients.

According to a study conducted in a South American rural clinic, the no-show reasons include factors like Transportation, long waiting times, bad weather and longer distance to the clinics

Current Practices

Several intervention techniques have been incorporated into these patient appointments that range from simple telephonic reminders or short message services (SMS) that reminds the patient of his appointment to advanced methods that uses Machine Learning and Artificial

Intelligence that could potentially decrease the frequency of no-showers. A study showed that by increasing the extent of communication in this chain, the rate of no-show decreased from 49% to 18% that sustained this no-show rate for approximately 2 years. Other alternatives include scheduling appointments for patients that show prior no-shows on a less-busier day, providing shuttle services to patients to and from the healthcare institution., home visits etc.

Literature Survey

Bigby et al. in 1983, found that for patients with prior predicted probability of no-shows greater than 20%, a 66.67% cost-savings was generated. This suggested that pinpointing the right predictors for generating a predictive classifier is fundamental. His study used Jrip and Hoeffding trees as these models could handle large datasets in considerable time and costs. Also, their ability to adapt to changes in features leading to no-shows were significant and thereby, these algorithms have the ability to adjust to any new concept that developed over time.

Other studies showed the positive correlation between waiting period or lead time which is the time-span between the appointment scheduled day and the actual appointment day and the patient no-show. This could potentially be a very important predictor for any predictor model in the healthcare industry. One more integral predictor that was identified from existing reports was if the patient is subsidized or self-paying that signifies his/her financial situation. Further, prior history of no-shows is an important feature as learned from literature.

Many models have been built in the past to predict the no-show of patients without cancellation that used different predictor variables, some of them have been listed in the table below.

No.	Author	Algorithm Used	Model Results
1.	Goffman et al, 2017	Logistic Regression	No shows reduced from 35% to 12.16%
2.	Ding et al, 2018	Regularized Logistic Regression	• Proven the value of fitting local level models. • Highlighted the importance of developing “personalized” risk scores.
3.	Devasahay, Karpagam and Ma, 2017	Logistic Regression, Decision Tree, SVM	Decision tree was the best fit model.
4.	Lee et al, 2017	Decision Tree, Logistic Regression, Gradient Boosting, Random Forest,	XGBoost was selected with AUC of 0.832, Precision of 0.785

		Elastic-Net, XGBoost	
5.	Gromish et al, 2010	Logistic Regression-Backward Elimination	Model was able predict patients who will miss more than 20% of appointments.
6.	Lenzi, Ben et al, 2019	Stepwise-Naïve, mixed effect logistic regression	The best model developed has AUC 80.9%
7.	Elvira et al, 2018	Gradient Boosting, General Linear model deep learning	Gradient Boosting algorithm with 74% AUC was selected.
8.	Alaedinni, Yang et al, 2015	Hybrid probabilistic model based on linear regression and empirical Bayesian	Built a model which predicts the real pattern correctly with small variance.

Table 1: Previous literature on similar datasets

Negative Impact of No-Shows

No-show impacts both the healthcare providers as well as the patient. Some of the effects include:

- Longer wait times
- Delay in Treatments
- Higher risk of getting admitted to Emergency Room
- Increase in Medical and Emergency expenses
- Customer dissatisfaction
- Loss for the hospitals and labs

Dataset Information

Name: Medical Appointment No-shows

The Dataset analysis was taken from OpenML which consisted of appointment details from Public Health Centers at a city in Brazil. The data collected for appointments was spread over a 6 weeks' time frame.

The dataset consists of 110527 observations with 14 variables.

PROBLEM STATEMENT

From the current dataset the following problem statements have been framed to achieve a good predictor model.

- To predict if an appointment made by a patient would be a no-show using various predictive classifier models, thereby helping the organization to improve their working efficiency.
- To explore the dataset to identify all the features that can influence patients' show-up.
- To identify if an SMS, a patient targeted intervention technique, impacts the frequency of patients showing up for their respective appointments.
- To analyze if the variables identified from the chosen dataset is enough to make a comprehensive predictor.

VARIABLE CATEGORIZATION WITH DESCRIPTION

The dataset consists of 14 variables. Out of these 13 are independent and 1 is the target variable. The variables are a mixture of numerical as well as categorical data type.

Independent Variables:

There are thirteen (13) variables involved in the study as mentioned below.

Variable	Datatype	Description
PatientId	Float	Identification allotted to each patient.
AppointmentID	Integer	Identification of each appointment.
Gender	Object	Female (0) or Male (1).
ScheduledDay	Object	The day of the actual appointment, when the patient has to visit the doctor.
Hypertension	Object	Whether the person is suffering from Hypertension or not? YES = 1 & NO = 0

Scholarship	Integer	Whether the patient receives Health Aid from Government. YES = 1 & NO = 0
Handicap	Integer	Any physical disability? YES = 1 & NO = 0
SMS_received	Integer	Is a message sent to the patient? YES = 1 & NO = 0
AppointmentDay	Object	The day someone called or registered the appointment, this is before appointment of course.
Age	Integer	Age of the patient
Diabetes	Object	Whether the person is suffering from Diabetes or not? YES = 1 & NO = 0
Alcoholism	Integer	Whether the person consumes Alcohol, or not? YES = 1 & NO = 0
Neighborhood	Object	The neighborhood or reside place of the patient.

Table 2: Variable Description

Target Variable:

The target variable of the Dataset is No-show.

We have to predict whether the patient is going to show up at the clinic or not.

In the given dataset 20% of the patients gave a no-show for their scheduled appointments and 80% showed up at the clinic. Thus, we observe there is a significant presence of class imbalance.

DATA PRE-PROCESSING

Data preprocessing is one of the crucial steps before we start to do any analysis and inference. It is the process of preparing the raw data and making it suitable so that it can work properly on a Machine learning model. The data is mostly flawed at times and we have to clean it and put it in a formatted manner. For this purpose, we use Data pre-processing.

The real-world data has lot of missing values and incorrect values which can render the data unusable to training purposes. The cleaner the data, the better the Machine Learning model works. This further leads to improved efficiency and accuracy which is very important in any business domain.

The dataset consists of 110527 observations with 14 variables.

Datatype Verification

The datatypes of the variables are enumerated as seen from table that describes every variable. Here we observed that Alcoholism, SMS-received, Handicap are given as Integer but they are categorical variables. We need to convert them to categorical variables so that the model can be trained properly.

Missing value treatment

The next step of data pre-processing is to handle the missing data in the datasets. A dataset with missing values may prove a hindrance while model training. So, it is necessary to handle missing values present in the dataset.
Thus, we can conclude that we have missing values in several columns with Neighborhood being the highest at 17.68% of the values missing.

Check for Outliers

The data has several outliers in different categorical and numerical columns. For making the base model, we don’t do outlier treatment and proceed with all the rows that are available from the dataset.

EXPLORATORY DATA ANALYSIS

Univariate Analysis

1) For Numerical Variables

Age

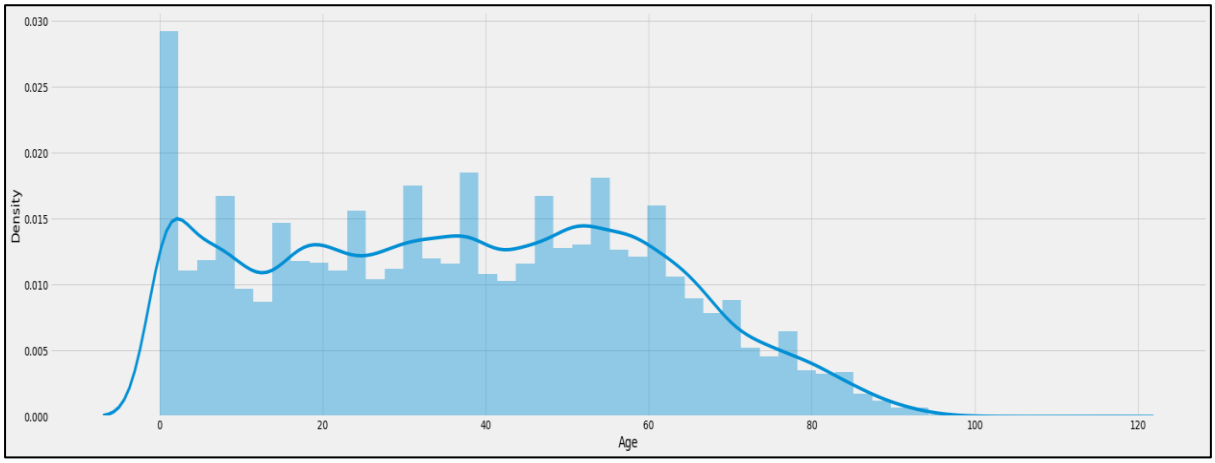


Figure 1:Univariate Analysis of Age using Distplot

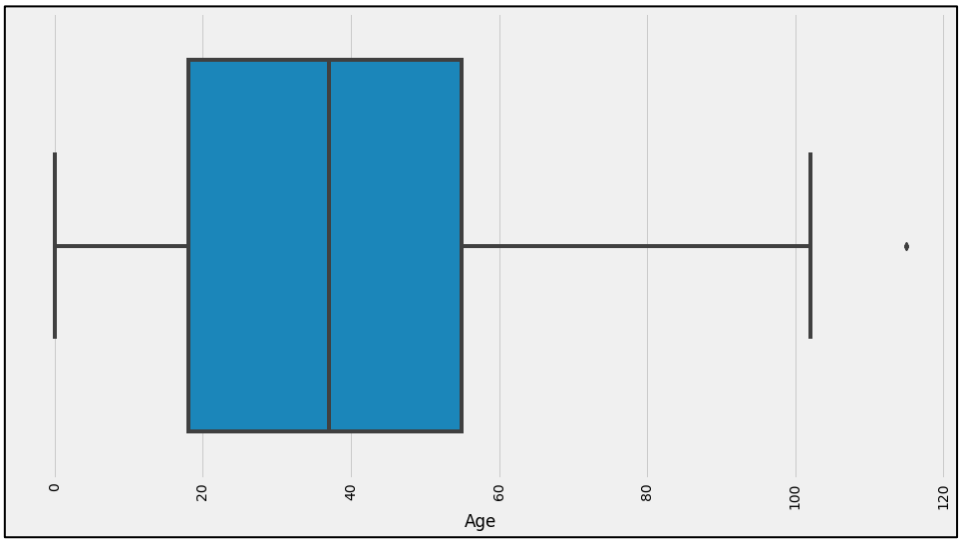


Figure 2:Checking Outliers using Boxplot

Kurtosis: -0.952267394656098
Skewness: 0.12165801789597985
1) According to the above distribution, age is right skewed.
2) It is Platykurtic and has flatter peak with thinner tails.
3) Inter-Quartile Range (IQR) lies between 18-55 years of age.
4) From the box plot, we infer that there are outliers present.

2) For Categorical variables

Scholarship

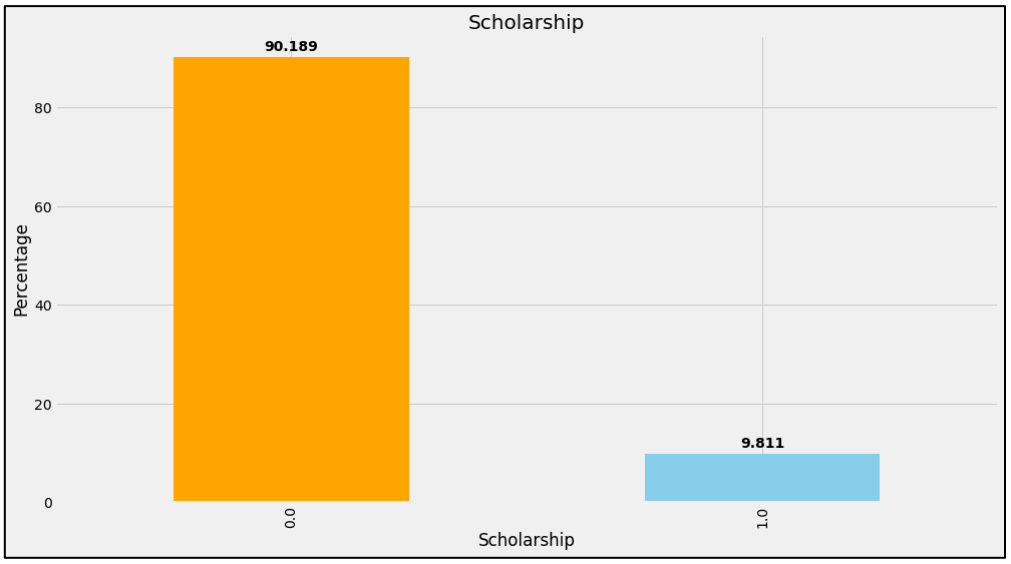


Figure 3: Univariate Analysis of Scholarship

The Scholarship refers to the enrolment in the Bolsa program and according to the above bar-plot, only 9.8% are enrolled whereas 90% are not part of this program.
This shows majority of the patients are not receiving any form of aid from the government.

Hypertension

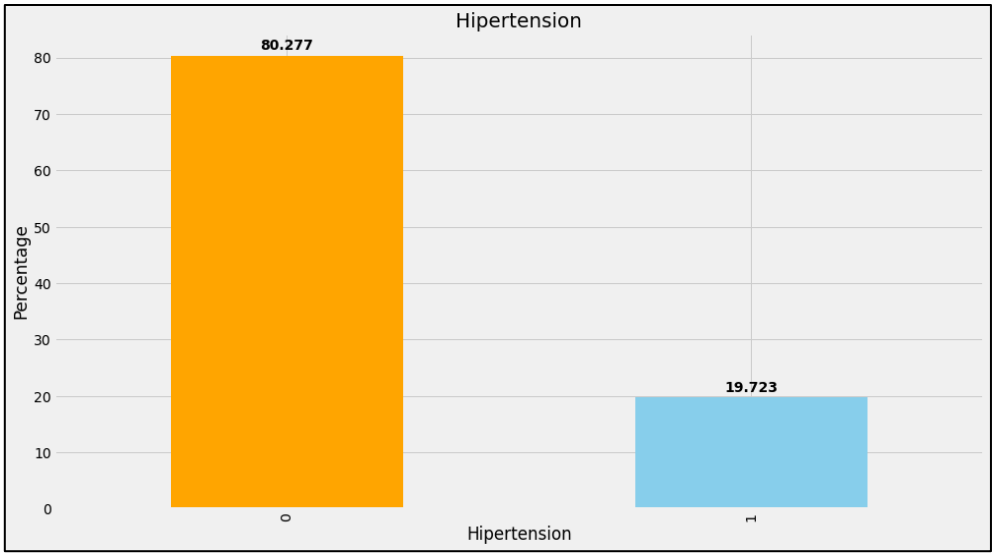


Figure 4: Univariate Analysis of Hypertension

According to the Bar-plot, around 80% of the patients suffer from Hypertension and 20% are the ones that have no Hypertension.

Diabetes

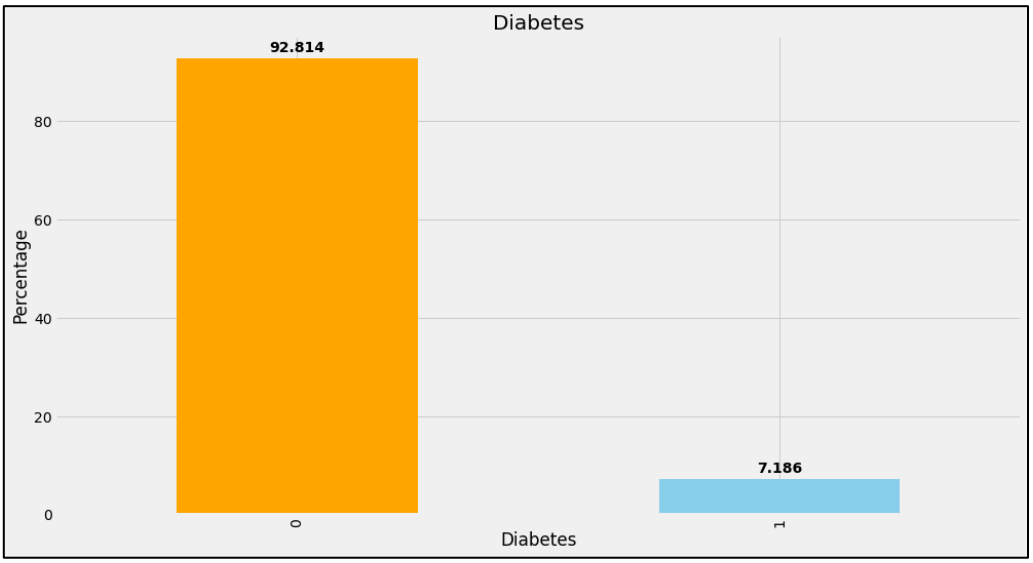


Figure 5: Univariate Analysis of Diabetes

Only 7 % are Diabetes patients in the dataset and 92.8% have no history of Diabetes.

Alcoholism

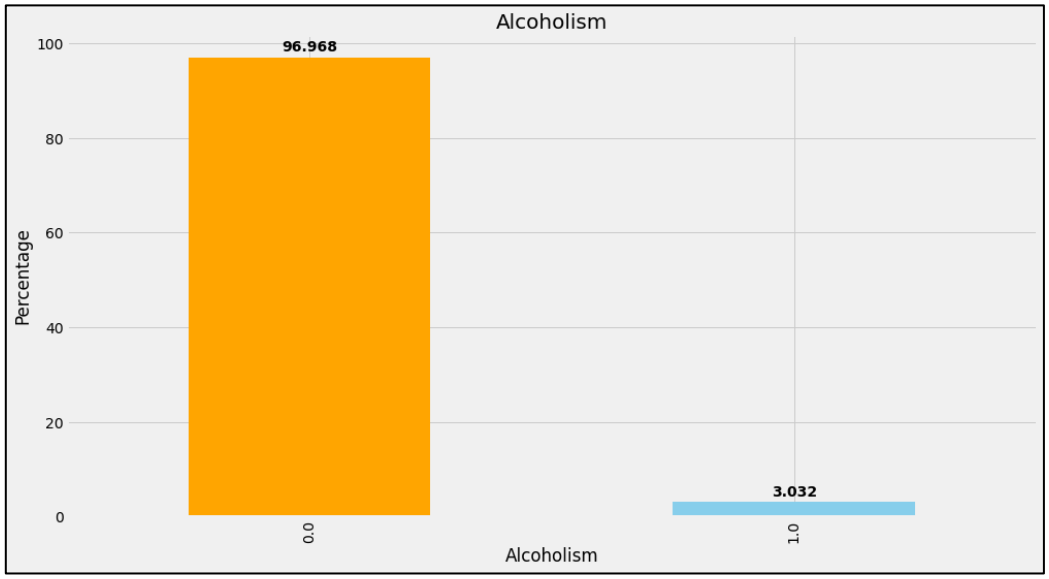


Figure 6: Univariate Analysis of Alcoholism

In the given dataset 96.9% are not Alcohol consumers and only 3% consume alcohol.

Handicap

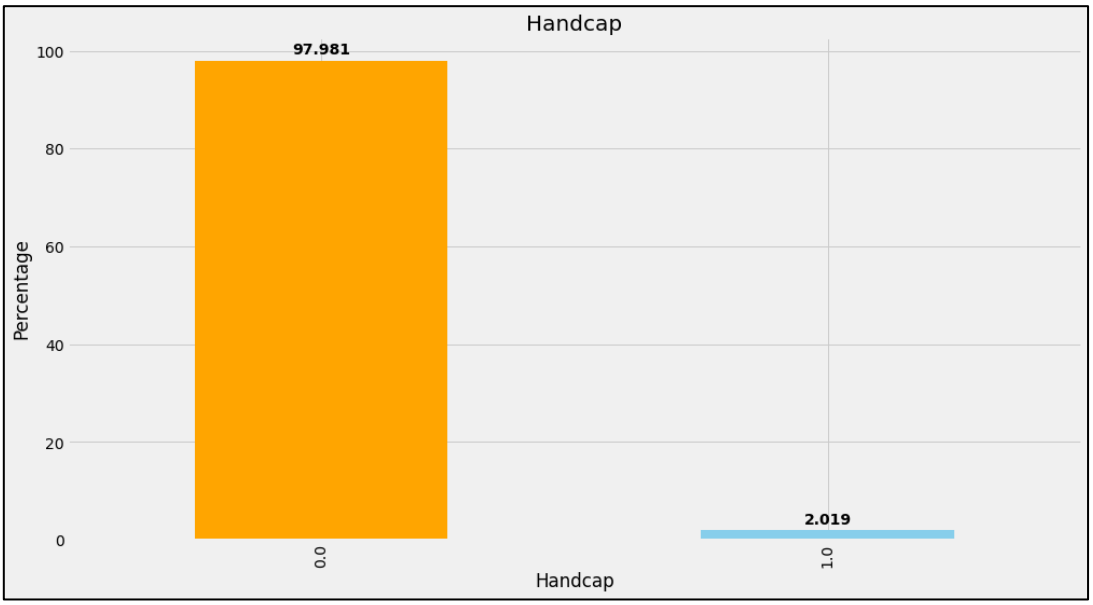


Figure 7: Univariate Analysis of Handicap

In the variable, Handicap only 2% of the patients have disability and 97.9% are disability free individuals.

SMS received

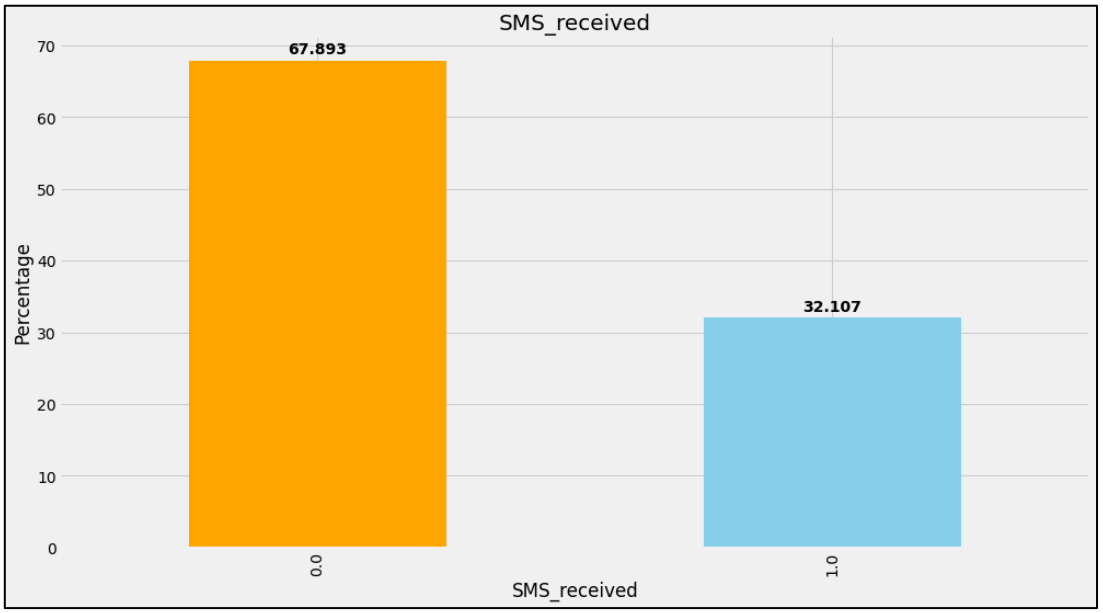


Figure 8: Univariate Analysis of SMS Received

Only 32% of the patients received a SMS notification whereas 67.8 % received no SMS notification.

No-show

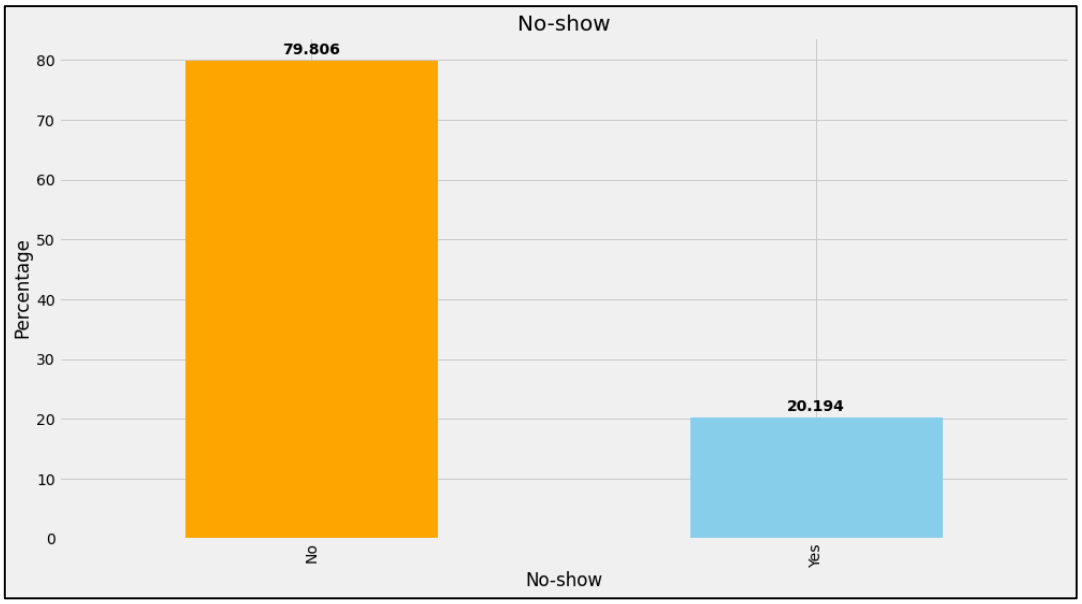


Figure 9: Univariate Analysis of No-show

In the given bar-plot, we can clearly see that a huge majority, around 79.8% didn't showed up to the clinic for the appointment whereas 20.1% actually made it to the appointment.
There is a considerable class imbalance in this particular variable.

Neighborhood

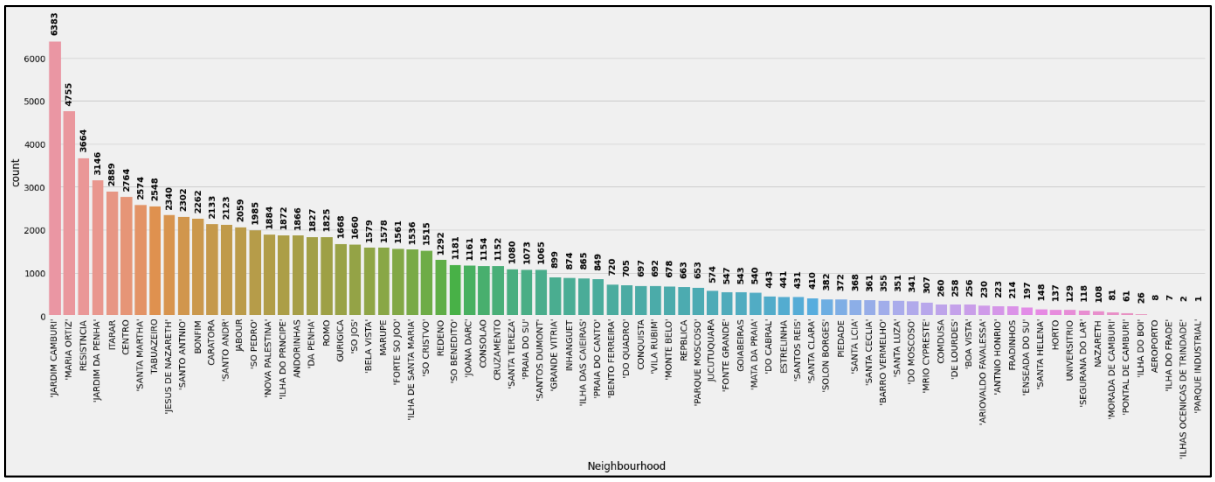


Figure 10: Univariate Analysis of Neighbourhood

According to the Neighborhood bar plot we have 81 neighborhoods with the top 3 being the following with their percentages.
The neighborhood with the highest number of visitations is “Jardim-Camburi” (6.98%) followed by “Maria -Ortiz”(5.25%) and “Resistencia”(4%).
Due to presence of large number of neighborhoods, we have bucketized them into fewer groups.

Bi-variate Analysis

1) No-show vs sms_received

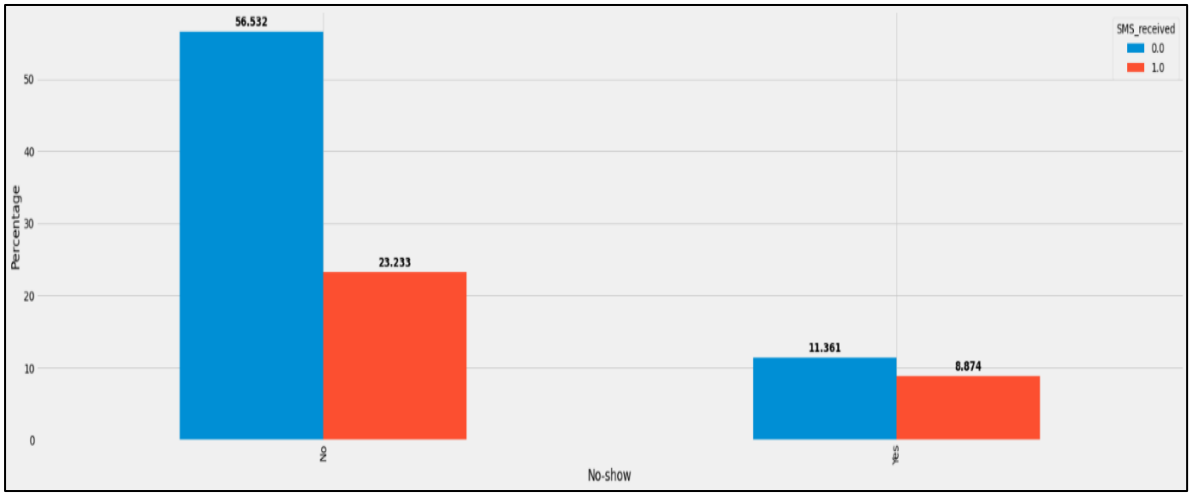


Figure 11: Bi-variate Analysis of No-show vs SMS Received

From the above plot we see that out of 80% of the total patients who came for appointment, 56.53% of patients did not receive any prior SMS and 23.23% received prior SMS.
On the other hand, out of the total 20% of the patients who did not show up for their appointments 11.36% did not receive any prior SMS and 8.87% did receive prior SMS.

2) No-show vs Alcoholism

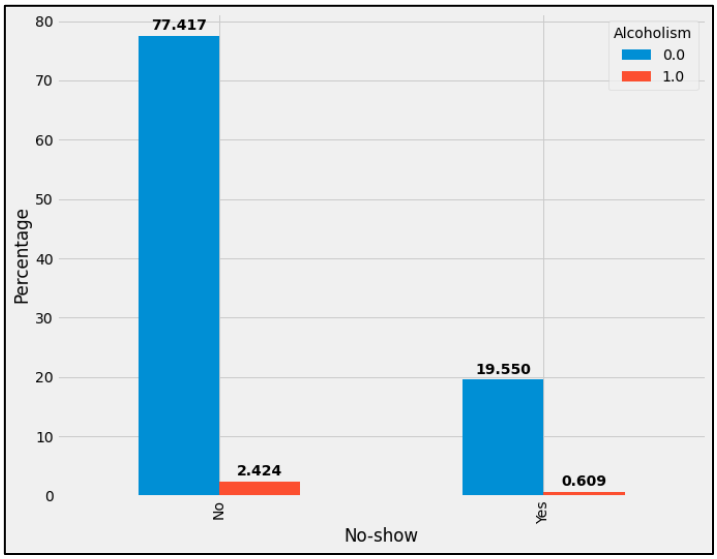


Figure 12: Bi-variate Analysis of No-show vs Alcoholism

From the above plot we see that out of 80% of the total patients who came for appointment, 77.42% of patients does not consume alcohol whereas 2.42% of the patients consumes alcohol.
On the other hand, out of the total 20% of the patients who did not show up for their appointments, 19.55% of patients does not consume alcohol whereas 0.61% of the patients consumes alcohol.

3) No-show vs Diabetes

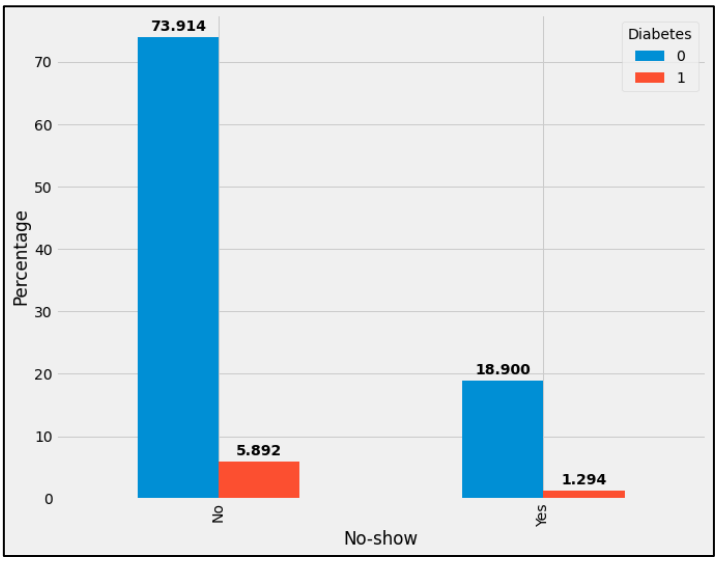


Figure 13: Bi-variate Analysis of No-show vs Diabetes

From the above metrics and plot we see that out of 80% of the total patients who came for appointment, 73.91% of patients does not suffer from diabetes whereas 5.89% of the patients suffers from diabetes.
On the other hand, out of the total 20% of the patients who did not show up for their appointments 18.89% of patients does not suffer from diabetes whereas 1.29% of the patients suffers from diabetes.

4) No-show vs Hypertension

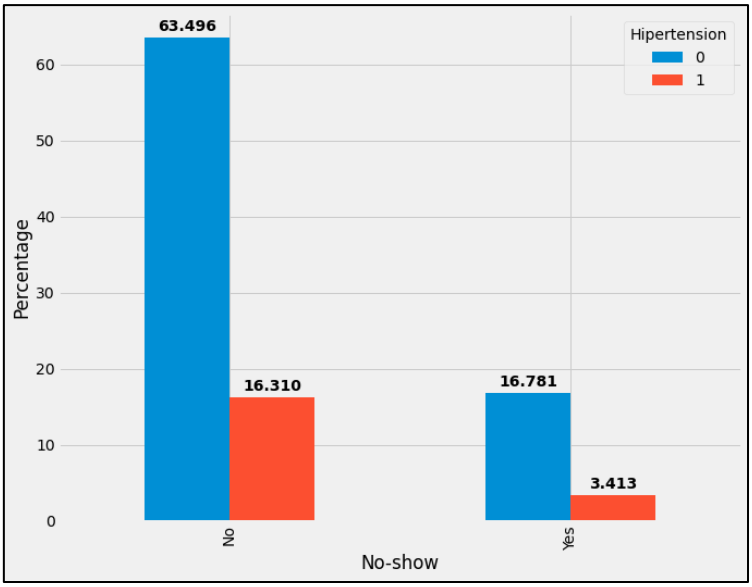


Figure 14: Bi-variate Analysis of No-show vs Hypertension

From the above metrics and plot we see that out of 80% of the total patients who came for appointment, 63.49% of patients does not suffer from Hypertension whereas 16.31% of the patients suffers from Hypertension. On the other hand, out of the total 20% of the patients who did not show up for their appointments, 16.78% of patients does not suffer from Hypertension whereas 3.41% of the patients suffers from Hypertension.

5) No-show vs Scholarship

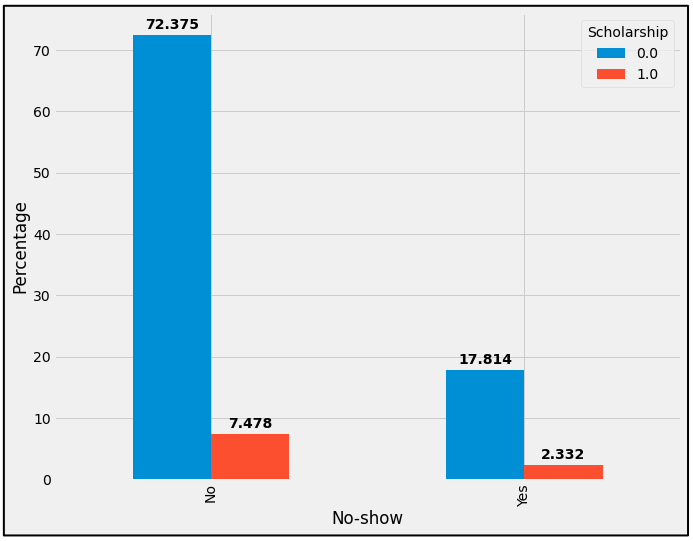


Figure 15: Bi-variate Analysis of No-show vs Scholarship

From the above metrics and plot we see that out of 80% of the total patients who came for appointment, 72.37% of patients does not have scholarship whereas 7.48% of the patients have scholarship.

On the other hand, out of the total 20% of the patients who did not show up for their appointments 17.81% of patients does not have scholarship whereas 2.33% of the patients have scholarship.

6) No-show vs Age

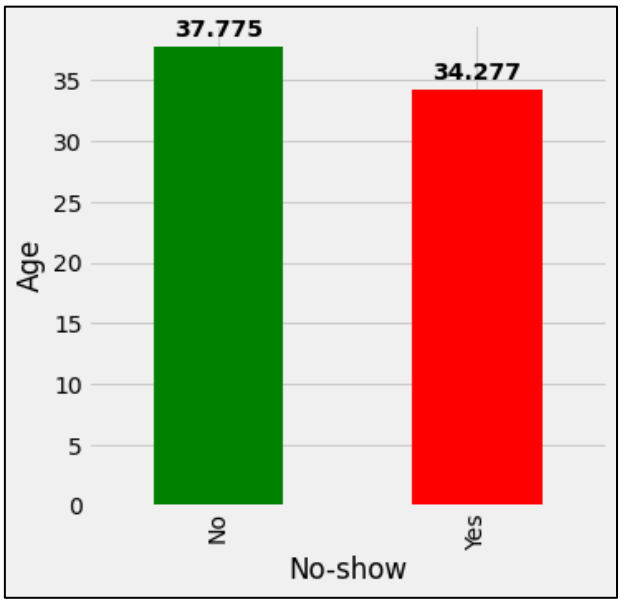


Figure 16: Bi-variate Analysis of No-show vs Age

From the above metrics and plot we see that, for the patients who showed up for the appointment their average age is 37 years, whereas for the patients who did not show up for the appointments their average age is around 34 years.

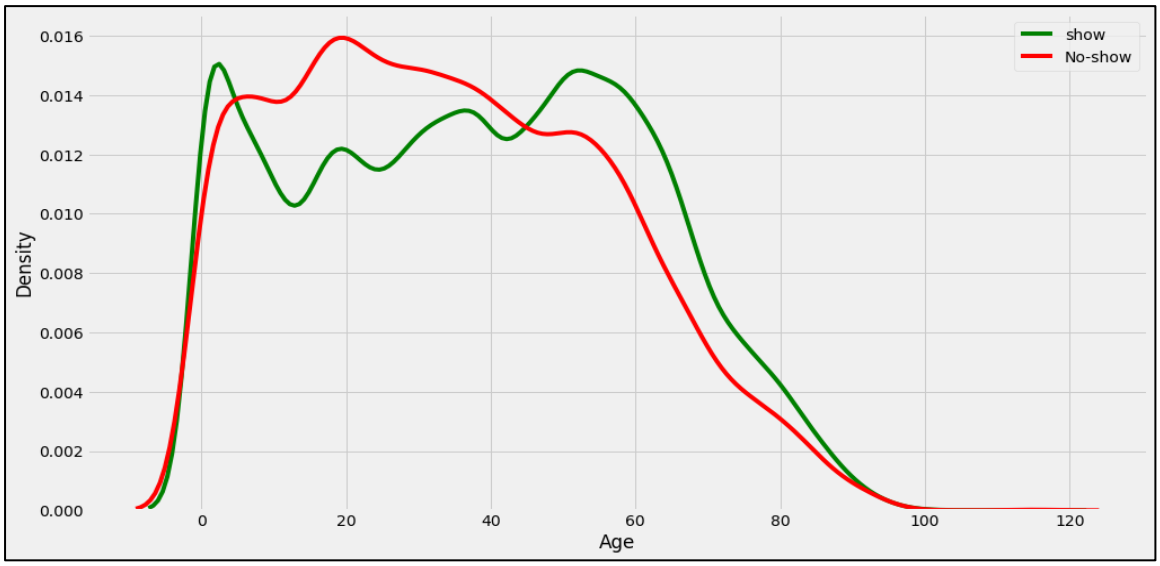


Figure 17:Distribution of Age using KDE plot

As we see in the KDE plot, the distribution of Age is nearly normal for the people who have not missed their appointment schedule, while there is a bit of positive skewness comparatively in the Age among those who have missed the appointment schedule.

7) No_show vs Handicap

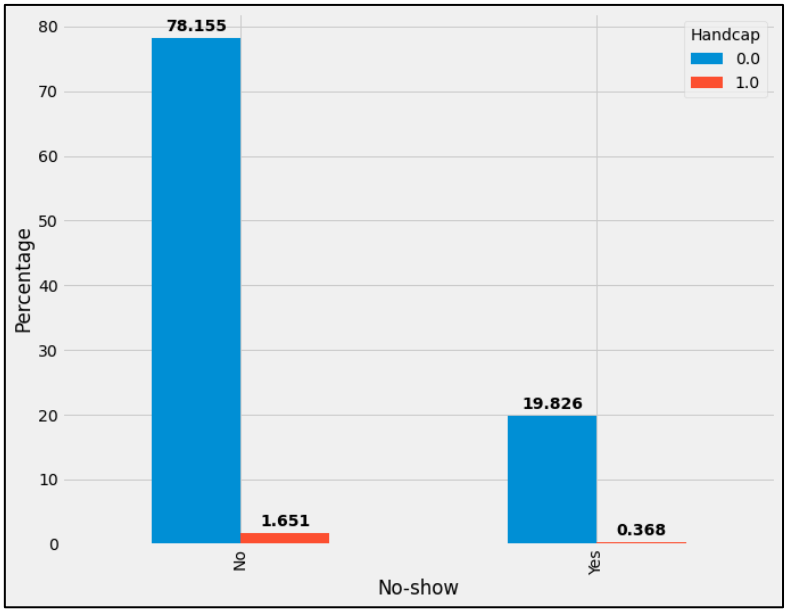


Figure 18:Bi-variate Analysis of No-show vs Handicap

From the above metrics and plot we see that out of 80% of the total patients who came for appointment, 78.15% of patients does not have any disability whereas 1.65% of the patients have some kind of disability.
On the other hand, out of the total 20% of the patients who did not show up for their appointments 19.82% of patients does not have any disability whereas 0.37% of the patients have some kind of disability.

8) No_show vs Gender

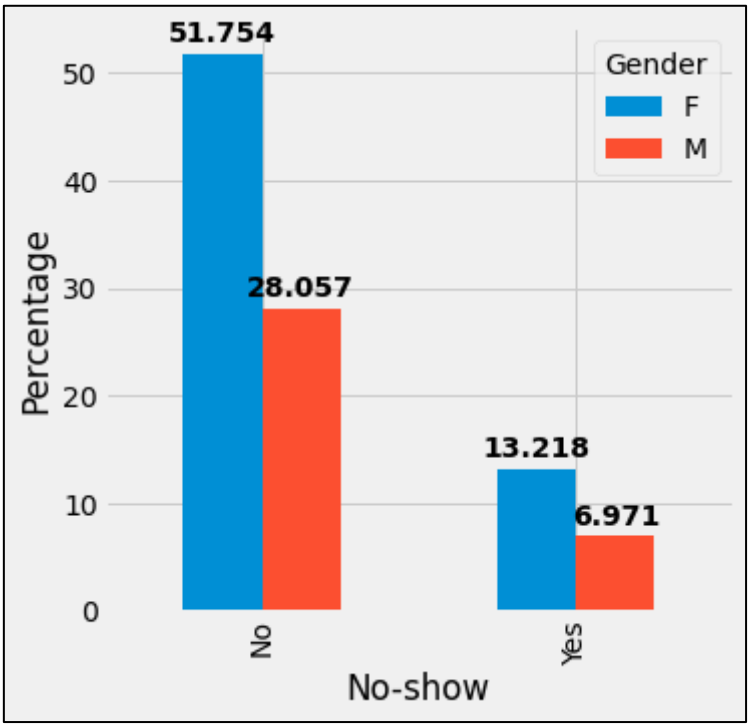


Figure 19::Bi-variate Analysis of No-show vs Handicap

From the above metrics and plot we see that out of 80% of the total patients who came for appointment, 51.75% of patients are female whereas 28.05% of the patients are male.
On the other hand, out of the total 20% of the patients who did not show up for their appointments 13.21% of patients are female whereas 6.97% of the patients are male.

Multi-variate Analysis

1) **Gender vs Age based on No-show**

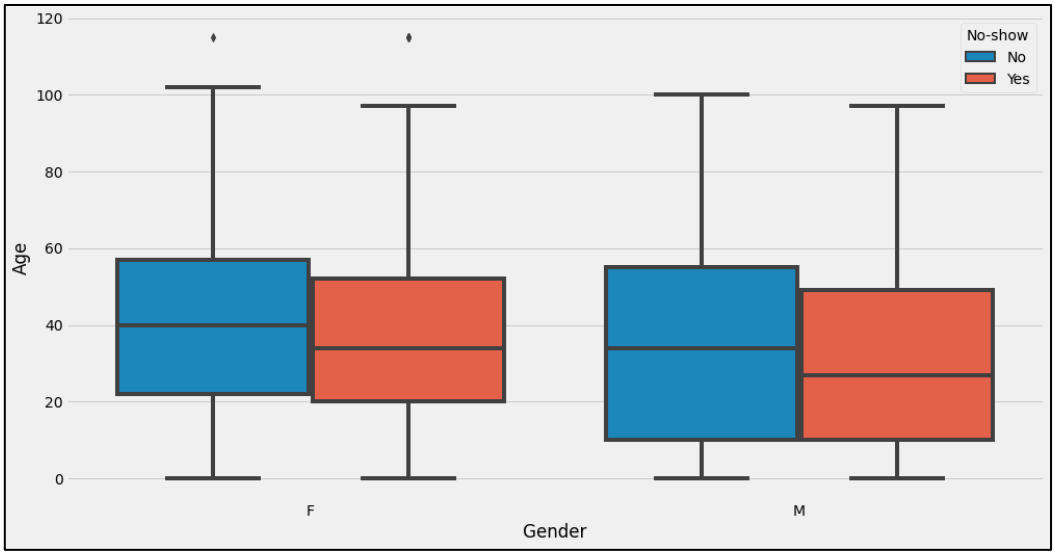


Figure 20: Multi-variate Analysis of Age vs Gender

We see that when gender is female the mean value of age is higher for the patients who showed up for the appointments than those who did not show up. It is the same for male patients. Therefore, we can infer that most of the patients who did not show up for the appointment belongs to the younger age group.

FEATURE ENGINEERING

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in the Machine Learning model.
It helps to represent an underlying problem to predictive models in a better way, which as a result, improve the accuracy of the model for unseen data. The predictive model contains predictor variables and an outcome variable, and while the feature engineering process selects the most useful predictor variables for the model.
In the project, we have identified a few features from the given data that can be extracted to gain more insights about the Target variable.

1) **Neighbourhood**

The neighborhood variable consists of 81 different values and they have been grouped together based on the state that they reside it.

- Group1: No-show percentage less than 1%
- Group2: No-show between percentage between 1%-2%

- Group3: No-show between percentage between 2%-3%
- Group4: No-show between percentage between 3%-4%
- Group5: No-show between percentage between 4%-5%
- Group6: No-show between percentage between 5%-6%
- Group7: No-show between percentage between 6%-7%
- Missing: A new category that includes all the missing values in the neighbour feature.

Since the Missing Value percentage is greater than 17% instead of imputing the missing values, we created a missing value category. Also imputing the missing neighborhood values with the mode of this category wouldn't work.

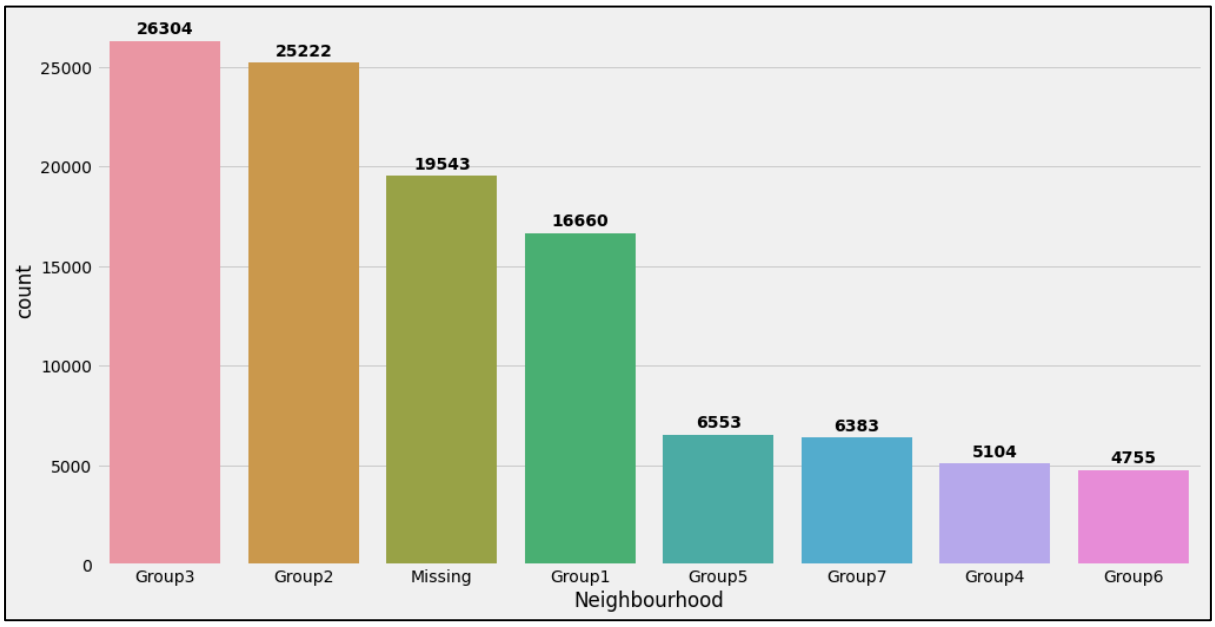


Figure 21: Univariate Analysis of the value counts of Neighbourhood

A univariate analysis of the value counts of the neighborhood indicates that the highest patients come from group 3 in neighborhood that contains all the localities that show 2-3% of no-shows.

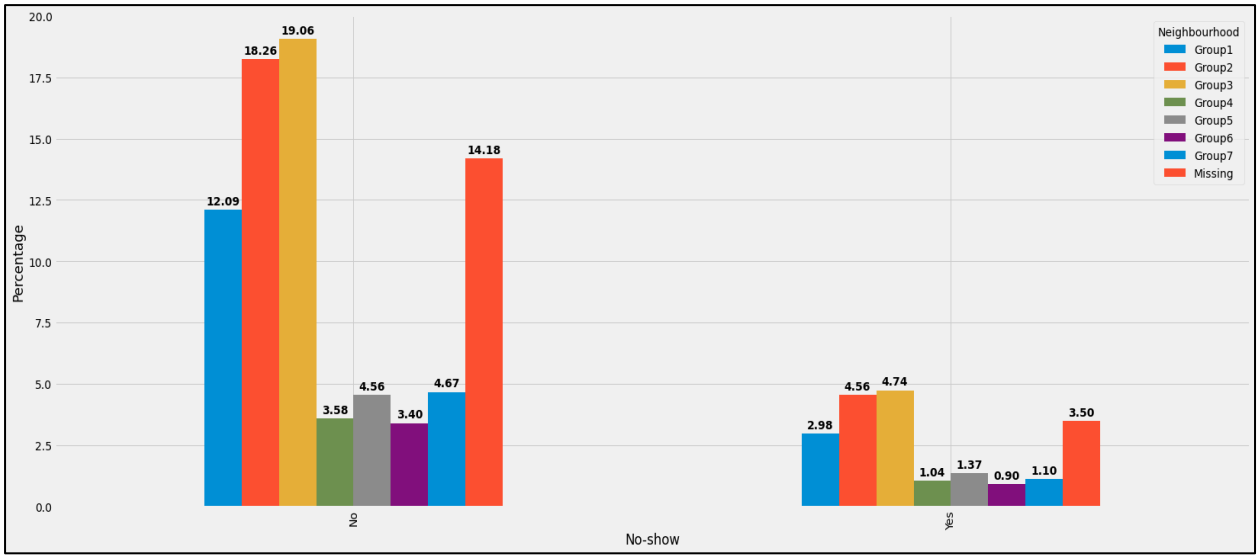


Figure 22: Bi-Variate analysis of No-show vs Neighbourhood

A bivariate analysis of no-shows and neighborhood indicates that group 3 from neighborhood dominates the frequency in both the groups of shows and no-shows since the count of group 3 is highest in making appointments.

2) Waiting Period

A new column by the name 'WaitingPeriod' is created to understand the number of days between Appointment Day and Scheduled Day. This definitely helps us to understand if there is an impact of waiting period on No_show.
This is achieved by simple subtraction of ScheduledDay from the AppointmentDay variable.

Bivariate analysis of No-show vs Waitingperiod

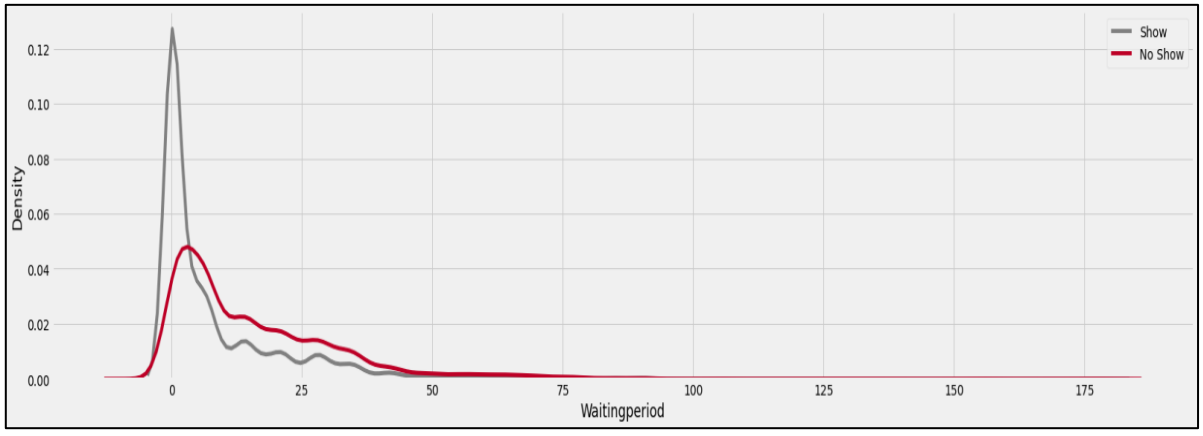


Figure 23: Bi-variate Analysis of No-show vs Waiting period

We infer from this density plot that as the waiting period between the scheduled day and appointment day increases, the chances of the patient not showing up increases as well.

3) Prior Appointment

A new feature is created to check if a patient has made multiple appointments since the PatientId for a patient is a unique ID.

Bivariate analysis of No-show vs prior_appointments

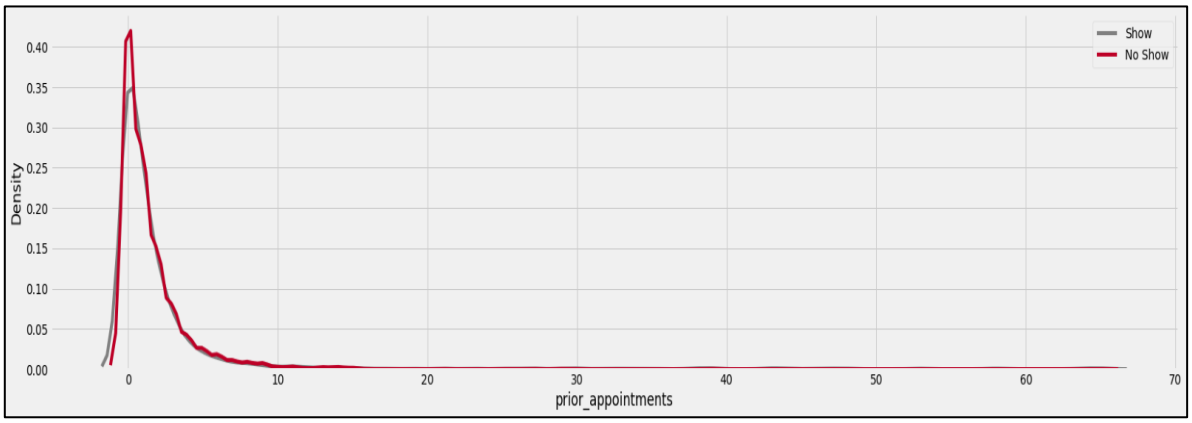


Figure 24: Bi-variate Analysis of No-show vs Prior appointments

We see from this distplot that the frequency of patients with no history of appointments tending to not show-up for their first appointment is greater than the regular patients with history of prior appointments.
But as the number of prior appointments increases the frequency of patients showing up for their appointments is not significantly different.

4) Prior No-shows

A new feature was created to check for the number of prior no-shows a patient has a history of.

Bivariate analysis No-show vs prior_no_shows

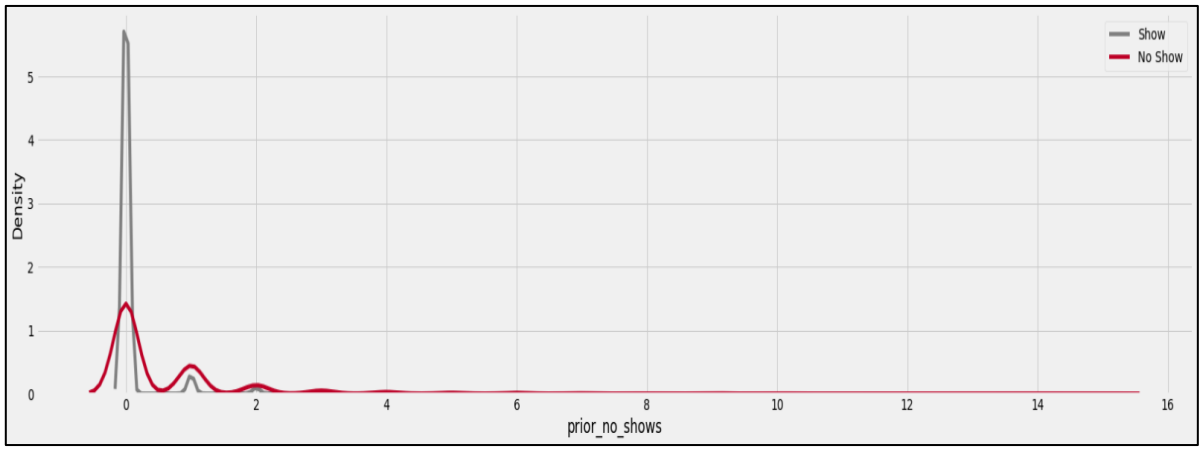


Figure 25:Bivariate analysis No-show vs prior_no_shows

We infer that, when patients show no history of no-shows, the frequency of them showing up for their current appointment is better than the ones that have a history of no-shows prior to this.
When there is history of no-shows for the patient, the chances of them not showing up to this current appointment is more as we can see from the plot.

5) Regular patient

A new feature regular_patient is created to check if the patient has made any prior appointments.

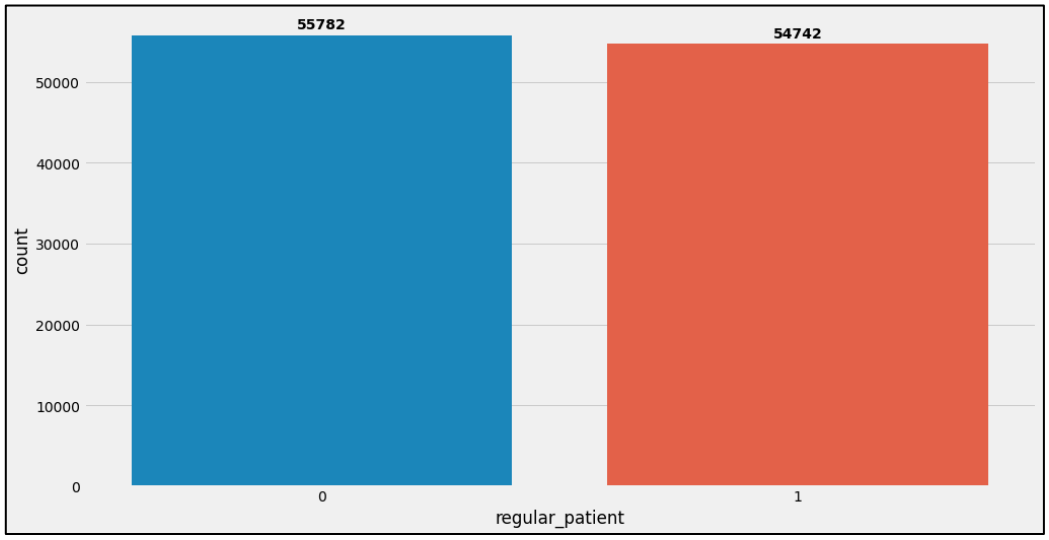


Figure 26: Checking of Prior appointments

We observe that off all the records the dataset has, patients with prior appointments are 54782 in number and patients that are making their first appointments are 55782 in number.

Multi-variate Analysis of feature engineered variables

1) Gender vs Waiting Period based on No-show

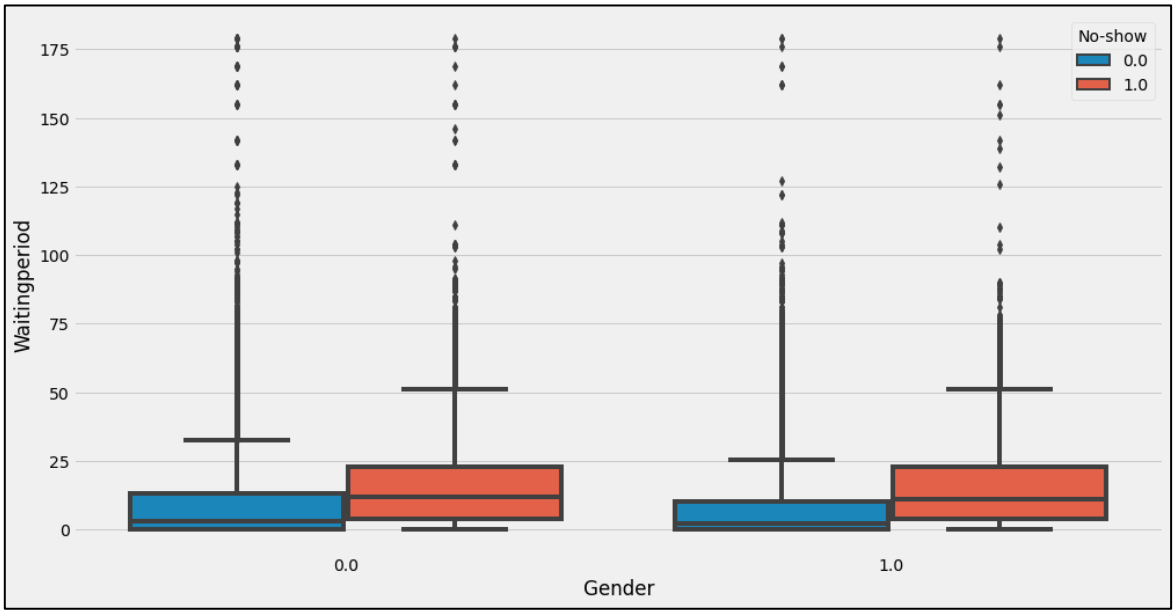


Figure 27: multi-variate analysis of Gender vs Waiting Period

From the above boxplot we see that for the female patients who show up for the appointments the waiting period is very low, on the other hand for male patients who show up for the appointments the waiting period is even lower where the median value is close to zero.
For patients who don't show up for the appointments the waiting period is almost the same for both female and male.

2) Neighbourhood vs Waiting Period based on No-show

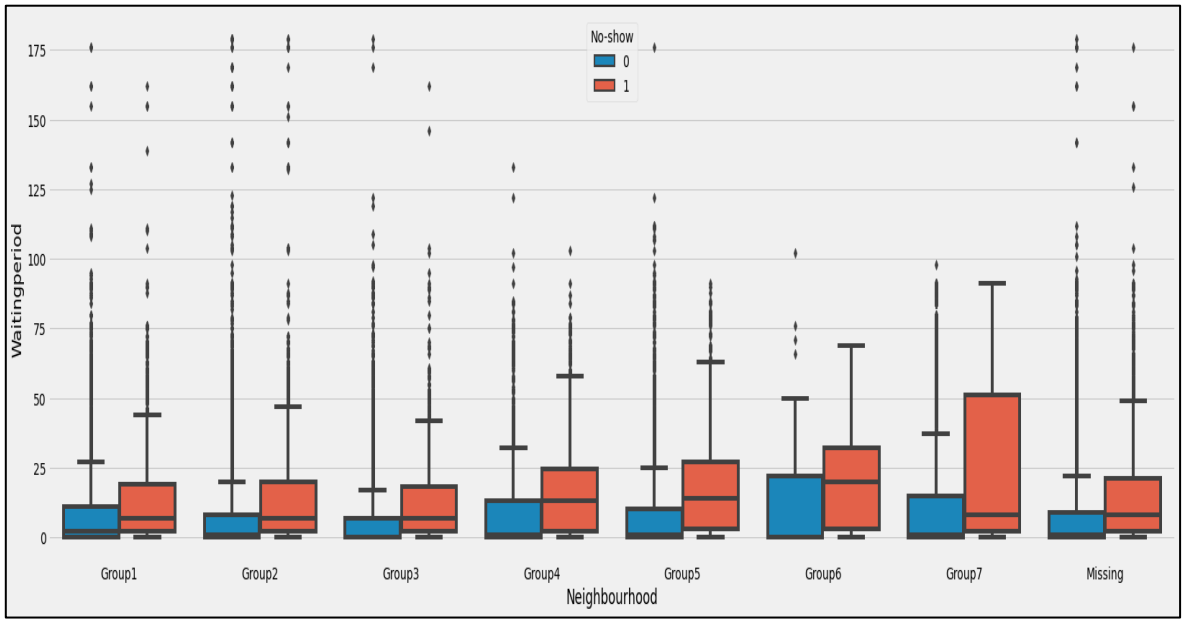


Figure 28: multi-variate analysis of Neighbourhood vs No-show

From the above plot we see that for the patients who showed up for the appointment, Neighbourhood_group2 have the lowest waiting period compared to other Neighbourhood and Neighbourhood_group6 has the highest waiting period. For the patients who did not show up for the appointment Neighbourhood_group3 has the lowest waiting period whereas Neighbourhood_group7 has the highest waiting period compared to other Neighbourhoods.

3) Correlation Heatmap

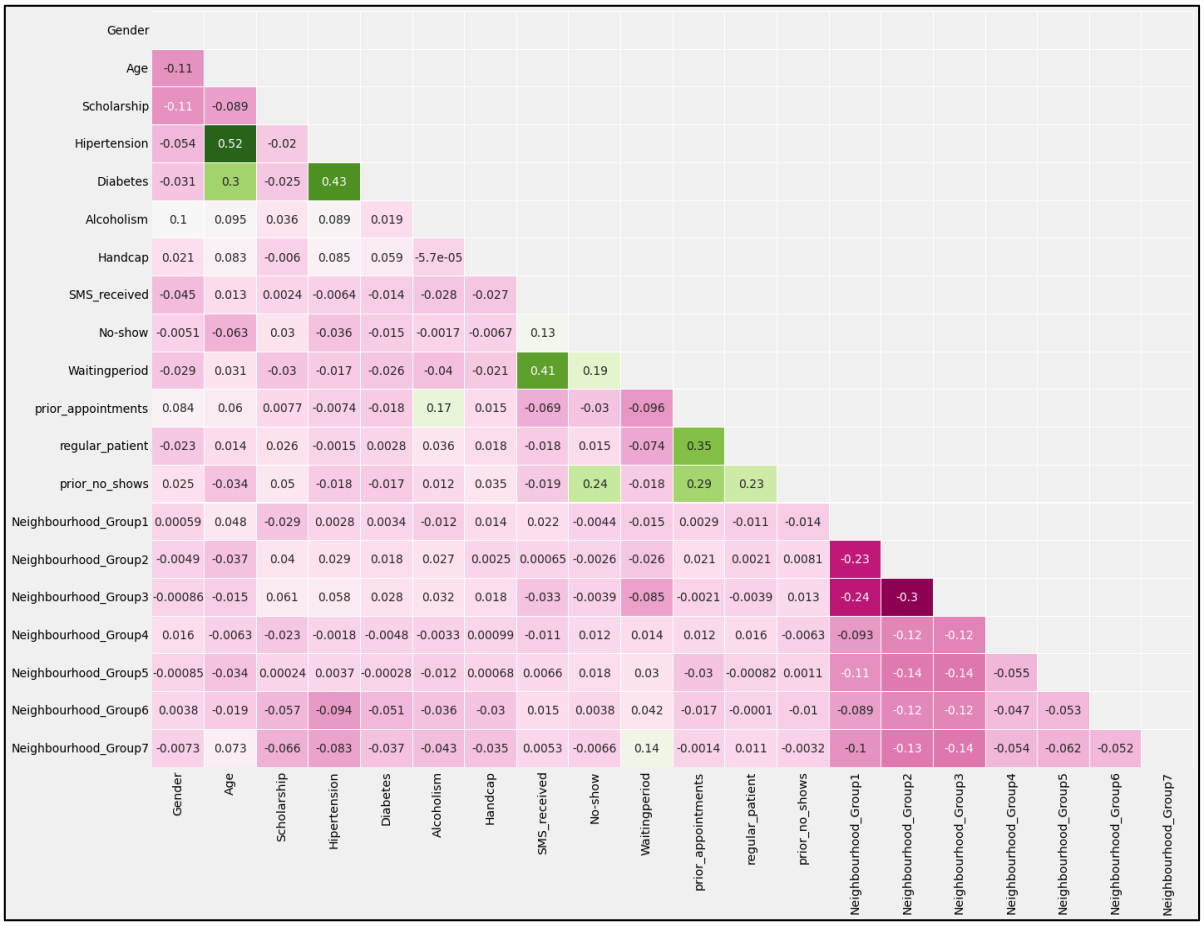


Figure 29: Correlation Heatmap

We infer from the above correlation heatmap that there is a chance of having multi-collinearity between variables showing relatively high correlation and, in our case, we observe that the variables age and hypertension show a correlation of 0.52 which may indicate the multicollinearity among them.

STATISTICAL TEST

1. Chi-Square Test Results:

The following results show us the relationship between Categorical Variables with respect to the Target Variable

Features	**P-Values**
Gender	0.090886
Scholarship	0.000000
Hypertension	0.000000
Diabetes	0.000000
Alcoholism	0.669593
Handicap	0.247162
SMS_received	0.000000
prior_appointments	0.000000
regular_patient	0.000001
prior_no_shows	0.000000
Neighbourhood_Group1	0.149715
Neighbourhood_Group2	0.393616
Neighbourhood_Group3	0.203474
Neighbourhood_Group4	0.000048
Neighbourhood_Group5	0.000000
Neighbourhood_Group6	0.219031
Neighbourhood_Group7	0.030222

Table 3: Chi-Square Test Result

H0: There is no relationship between the categorical and target variable.

H1: There is a relationship between the categorical and target variable.

Since statistical tests are highly sensitive to small changes in the data points, we proceeded with a level of significance equal to 0.20 and based on the hypotheses framed we consider only those variables that have p-value less than 0.20. Here we see that Alcoholism, Handicap, Neighbourhood_Group1,Neighbourhood_Group2,Neighbourhood_Group3, Neighbourhood_Group6 have no relationship with the target variable. So, although individually these variables have no relationship with the target, we have to see how these variables help in classification altogether.

2. ANOVA Tests:

The following results show us the relationship between Numerical Variables with respect to the Target Variable:

.

Features	P-Values
Age	0.000000
Waitingperiod	0.000000

Table 4: ANOVA Test Result

Here we see that Age and Waitingperiod are having a difference in group means based on the target variable. Therefore, we can conclude that Age and Waitingperiod are having a relationship with the target variable.

Also, we tried to test for collinearity for each independent variable using chi-square and we saw that there is relationship between these independent variables but we cannot determine the degree of relationship using statistical chi square test. So, we went ahead with the base model building.

BASE MODEL-1

We have selected logistic model as a base model so that we can observe how each individual variable is helping the model in classifying the target variable.

```
Optimization terminated successfully.
         Current function value: 0.493230
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:                No-show   No. Observations:               110524
Model:                          Logit   Df Residuals:                   110508
Method:                           MLE   Df Model:                           15
Date:                Fri, 10 Jun 2022   Pseudo R-squ.:                 0.01957
Time:                        07:22:36   Log-Likelihood:                -54514.
converged:                       True   LL-Null:                       -55602.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                         -1.4196      0.024    -58.467      0.000      -1.467      -1.372
Gender                        -0.0202      0.016     -1.223      0.221      -0.052       0.012
Age                           -0.0057      0.000    -14.647      0.000      -0.006      -0.005
Scholarship                    0.1964      0.025      7.815      0.000       0.147       0.246
Hipertension                  -0.1022      0.024     -4.185      0.000      -0.150      -0.054
Diabetes                       0.0738      0.034      2.166      0.030       0.007       0.141
Alcoholism                     0.1274      0.046      2.742      0.006       0.036       0.218
Handcap                        0.0203      0.058      0.347      0.728      -0.094       0.134
SMS_received                   0.6459      0.016     41.561      0.000       0.615       0.676
Neighbourhood_Group1           0.0025      0.027      0.092      0.927      -0.050       0.055
Neighbourhood_Group2           0.0041      0.024      0.168      0.867      -0.043       0.051
Neighbourhood_Group3           0.0223      0.024      0.931      0.352      -0.025       0.069
Neighbourhood_Group4           0.1848      0.038      4.805      0.000       0.109       0.260
Neighbourhood_Group5           0.1775      0.035      5.103      0.000       0.109       0.246
Neighbourhood_Group6           0.0473      0.041      1.167      0.243      -0.032       0.127
Neighbourhood_Group7          -0.0023      0.037     -0.061      0.951      -0.075       0.071
==============================================================================================
```

Figure 30: Base Model-1

This is a base model that has been created from the variables available from our dataset dropping unnecessary variables. It doesn't include any variables that have been feature engineered. Just the Neighbourhood variable from the raw dataset has been bucketed based on the proportion of No-show of the target variable.

From the above model summary, we see that Gender, Handicap, Neighbourhood_Group1, Neighbourhood_Group2,Neighbourhood_Group3,Neighbourhood_Group6, Neighbourhood_Group7 are statistically insignificant. The Pseudo R-squ (McFadden's R-squ) value is 0.02 indicating that it is not a good model. Since Pseudo R-squ does not prove to be a good evaluation metric for the model, we use other evaluation metrics to evaluate this model as shown below.

Train Report

```
Confusion Matrix-Train
 [[61709     0]
 [15657     0]]

Classification Report -Train
              precision    recall  f1-score   support

           0       0.80      1.00      0.89     61709
           1       0.00      0.00      0.00     15657

    accuracy                           0.80     77366
   macro avg       0.40      0.50      0.44     77366
weighted avg       0.64      0.80      0.71     77366

Accuracy-Train
 0.7976242793992193

ROC_AUC-Train
 0.5952337553832857
```

Figure 31: Train Report for Base model-1

Test Report

```
Confusion Matrix-Test
 [[26496     0]
 [ 6662     0]]

Classification Report -Test
              precision    recall  f1-score   support

           0       0.80      1.00      0.89     26496
           1       0.00      0.00      0.00      6662

    accuracy                           0.80     33158
   macro avg       0.40      0.50      0.44     33158
weighted avg       0.64      0.80      0.71     33158

Accuracy-Test
 0.7990831775137222

ROC_AUC-Test
 0.5957148972804513
```

Figure 32: Test Report for Base Model-1

Here we tried different evaluation metrics and we see from the confusion matrix that the model is not at all able to predict the class of interest, i.e., patients who did not show up for the appointments. Accuracy score is 0.79, weighted average f1-score is 0.71 wherein f1-score for the class of interest is 0. Also, the AUC-ROC score is 0.59 which is very close to null model score of 0.5. From these evaluation metric scores we can conclude that this model is not a good model for prediction and we need to either improve the model or try out different model and see whether they are providing better predictions.

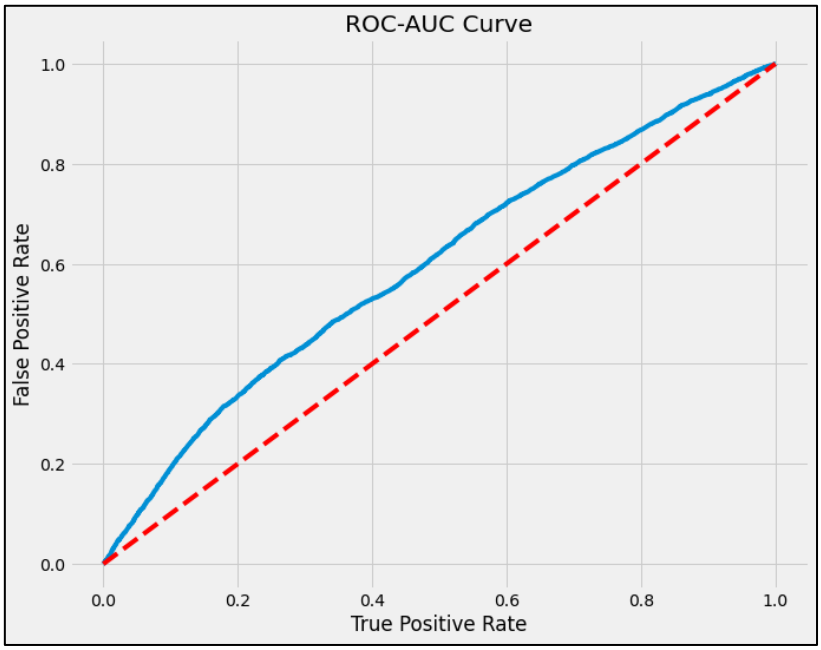


Figure 33: ROC-AUC Curve for Base Model-1

In the above ROC-AUC plot we see that the model ROC-AUC curve is very close to the null model curve indicating that the base logistic model is not at all a good model and we need to improve the model for better prediction first using the feature engineered variables.

BASE MODEL-2

```
Optimization terminated successfully.
         Current function value: 0.447096
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:                No-show   No. Observations:               110524
Model:                          Logit   Df Residuals:                   110504
Method:                           MLE   Df Model:                           19
Date:                Fri, 10 Jun 2022   Pseudo R-squ.:                  0.1113
Time:                        07:30:01   Log-Likelihood:                -49415.
converged:                       True   LL-Null:                       -55602.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                         -1.6903      0.027    -62.294      0.000      -1.743      -1.637
Gender                        -0.0071      0.017     -0.403      0.687      -0.041       0.027
Age                           -0.0048      0.000    -11.746      0.000      -0.006      -0.004
Scholarship                    0.1231      0.027      4.537      0.000       0.070       0.176
Hipertension                  -0.0991      0.026     -3.847      0.000      -0.150      -0.049
Diabetes                       0.1105      0.036      3.082      0.002       0.040       0.181
Alcoholism                     0.2668      0.050      5.333      0.000       0.169       0.365
Handcap                       -0.1027      0.065     -1.590      0.112      -0.229       0.024
SMS_received                   0.4298      0.017     24.672      0.000       0.396       0.464
Waitingperiod                  0.0210      0.001     40.382      0.000       0.020       0.022
prior_appointments            -0.2534      0.007    -35.373      0.000      -0.267      -0.239
regular_patient                0.2043      0.021      9.944      0.000       0.164       0.245
prior_no_shows                 1.3334      0.019     70.637      0.000       1.296       1.370
Neighbourhood_Group1           0.0562      0.028      1.993      0.046       0.001       0.112
Neighbourhood_Group2           0.0313      0.026      1.221      0.222      -0.019       0.081
Neighbourhood_Group3           0.0716      0.025      2.818      0.005       0.022       0.121
Neighbourhood_Group4           0.1949      0.041      4.798      0.000       0.115       0.274
Neighbourhood_Group5           0.0840      0.037      2.269      0.023       0.011       0.157
Neighbourhood_Group6           0.0316      0.043      0.742      0.458      -0.052       0.115
Neighbourhood_Group7          -0.2535      0.041     -6.222      0.000      -0.333      -0.174
==============================================================================================
```

Figure 34: Base Model-2

This is an improved base model that has been created from the variables available from our dataset dropping unnecessary variables and including grouped Neighbourhood and feature engineered variables.

From the above model summary, we see that Gender, Handicap, Neighbourhood_Group2, Neighbourhood_Group6 are statistically insignificant. The Pseudo R-square (McFadden's R-square) value is 0.11 indicating that it is not a good model but it is better than the base model. Since Pseudo R2 does not a good evaluation metric for the model we use other evaluation metrics to evaluate this model as shown below.

Train Report-2

```
Confusion Matrix-Train
 [[60739   970]
 [13295  2362]]

Classification Report -Train
              precision    recall  f1-score   support

           0       0.82      0.98      0.89     61709
           1       0.71      0.15      0.25     15657

    accuracy                           0.82     77366
   macro avg       0.76      0.57      0.57     77366
weighted avg       0.80      0.82      0.76     77366

Accuracy-Train
 0.8156166791613887

ROC_AUC-Train
 0.7345480458678262
```

Figure 35: Train Report for Base Model-2

Test Report-2

```
Confusion Matrix-Test
 [[26080   416]
 [ 5636  1026]]

Classification Report -Test
              precision    recall  f1-score   support

           0       0.82      0.98      0.90     26496
           1       0.71      0.15      0.25      6662

    accuracy                           0.82     33158
   macro avg       0.77      0.57      0.57     33158
weighted avg       0.80      0.82      0.77     33158

Accuracy-Test
 0.8174799445081127

ROC_AUC-Test
 0.7374856919771375
```

Figure 36: Test Report for Base Model-2

Here we tried different evaluation metrics and we see from the confusion matrix that the model although better than the base model1 is only able to predict about 15% of the class of interest, i.e., patients who did not show up for the appointments. Accuracy score is 0.82, weighted average f1-score is 0.77 wherein f1-score for the class of interest, i.e., No-show is 0. Also, the AUC-ROC score is 0.75 which is quite far from the null model score of 0.5. From these evaluation metric scores we can conclude that this model although has improved a lot after including feature engineered variables, it is only an average model and cannot be used for final prediction and we need to either improve the model further or try out different model and see whether they are providing better predictions.

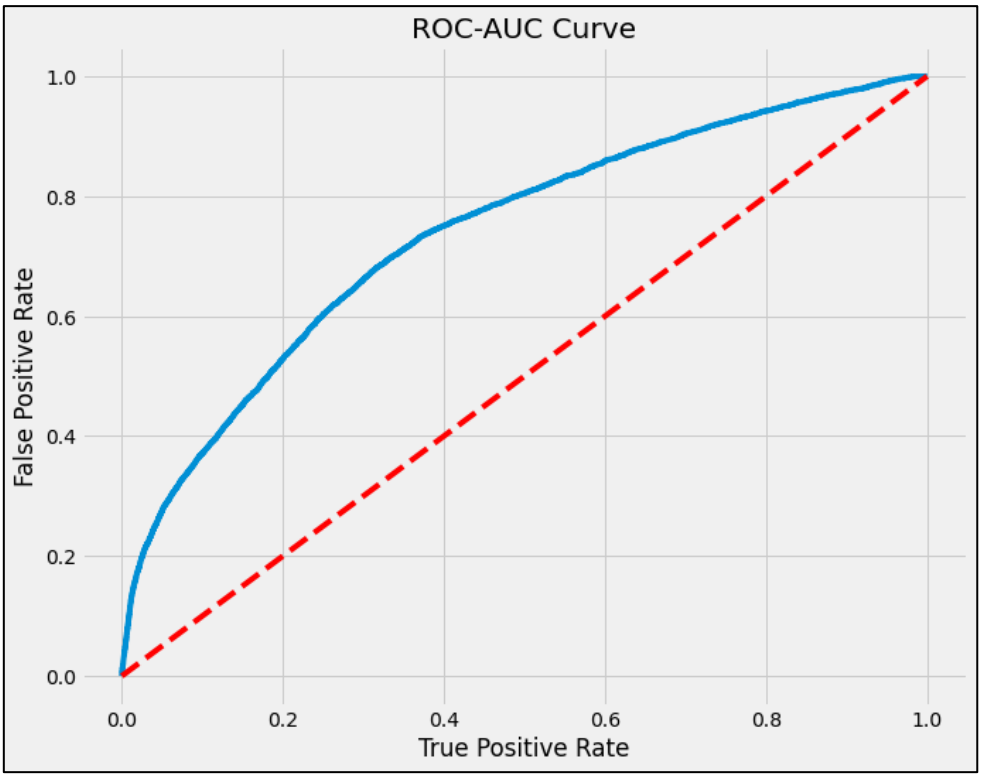


Figure 37: ROC-AUC Curve for Base Model-2

In the above ROC-AUC plot we see that the model ROC-AUC curve is quite far from the null model curve indicating that this logistic model with feature engineered variables is just an average model that's not good enough to be used as the final predictor and we need to improve the model or try out different models to achieve better predictions.

BASE MODEL IMPROVEMENT

Outlier detection and Treatment

As seen in *Figure 2* and *Figure 28*, we detected outliers in the two numerical variables in our dataset. The two variables are age and waiting period and the outliers in these features were treated. Upon removal of these outliers by the interquartile range (IQR) method where any

datapoints lying beyond the upper and lower whiskers in the data distribution we observed that close to 30,000 out of the 110,527 data entries were removed that may remove useful information for the model building. Therefore, it was decided to cap these outliers with the values of the upper and whisker where any outlier value beyond the upper whisker is replaced with the upper whisker value which is (quartile3+1.5*IQR) and any value beyond the lower whisker is replace with the value of the lower whisker which is (quartile1-1.5*IQR). Hence the outliers were treated successfully.

Scaling the variables

The numeric variables were scaled using the StandardScaler() function. The Standard scaler converts the numerical variables with different ranges to a standard Z-score range from +3 to -3. Therefore, the shape of the data remains the same and both the numerical variables remain in the same range.

After outlier treatment and scaling the numerical variables, we moved onto building a logistic model with the feature engineered variables that was optimized by Youden's index. Youden's index is the max(tpr-fpr) value whose threshold we use to obtain the predicted values. We built a classification report and calculated the F2-score and ROC_AUC score as seen in the figures below.

	TPR	FPR	Threshold	Difference
0	0.705194	0.333522	0.172738	0.371672
1	0.705194	0.333560	0.172735	0.371634
2	0.705494	0.333899	0.172636	0.371594
3	0.705194	0.333635	0.172732	0.371558
4	0.705644	0.334088	0.172573	0.371556

```
              precision    recall  f1-score   support

           0       0.90      0.67      0.77     26496
           1       0.35      0.71      0.47      6662

    accuracy                           0.67     33158
   macro avg       0.62      0.69      0.62     33158
weighted avg       0.79      0.67      0.71     33158


F2 score:  0.584465681150764

AUC ROC Score:  0.7421838091238142
```

Figure 38:Classification Report and ROC_AUC Score using Youden's Index Threshold for Base Logistic Regression

For maximum difference between the tpr and fpr for this model, the threshold we see from the figure above is 0.17. Using this threshold, we calculate the predicted values of the test set and generated the classification report where we observed a recall of 0.71 for the minority class. The calculated F2 score was 0.58 and a AUC_ROC score of 0.74. These metrics are the chosen ones to evaluate the model. Also, these metrics might improve upon treating the class imbalance in our dataset and building more suitable non-linear models. We moved onto treating the class imbalance with various techniques.

Treating Class-Imbalance

We used four different class imbalance techniques to bring balance in distribution of the classes to build further models with an aim to improve the metrics of the predictor.

1) SMOTE

SMOTE (synthetic minority oversampling technique) is universally used as the most common method to solve any class imbalance. The minority class is randomly replicated to bring balance in class distribution by linear interpolation of this class in our target variable. These synthetic training points are obtained by randomly selecting one or more k-nearest neighbors for each example in the minority class which in our case is '1'.Upon using this technique on the training set of the data, we built a model and optimized the same with a threshold corresponding to its Youden's index and compared the generated classification reports and calculated metrics.

	TPR	FPR	Threshold	Difference
0	0.661513	0.367188	0.397937	0.294326
1	0.661513	0.367225	0.397864	0.294288
2	0.661363	0.367150	0.397977	0.294213
3	0.661363	0.367188	0.397948	0.294175
4	0.661213	0.367112	0.398038	0.294101

```
              precision    recall  f1-score   support

           0       0.88      0.63      0.74     26496
           1       0.31      0.66      0.42      6662

    accuracy                           0.64     33158
   macro avg       0.60      0.65      0.58     33158
weighted avg       0.77      0.64      0.67     33158


F2 score:  0.5402854060415848

AUC ROC Score:  0.6945145880875672
```

Figure 39:Classification Report and ROC_AUC Score using Youden's Index Threshold for Base Logistic Regression coupled with SMOTE

With a threshold of 0.40 we obtain the classification report by calculating the predicted values of the test data. The recall for the minority class was observed to be 0.66 and the F2-score and AUC_ROC score for this model after SMOTE analysis was 0.54 and 0.69 respectively.

2) ENN

Edited Nearest Neighbor (ENN) method works by identifying the K-nearest neighbor of each entry initially and checks whether the majority class from these observation's k-nearest neighbor is the same as the observation's class or not. The observation and its knn is removed if this majority class of the observation's Knn and the observation's class are different. We used a default value for k=3 in ENN. Therefore, it makes this technique an under sampler. Upon using this technique on the training set of the data, we built a model and optimized the same with a threshold corresponding to its Youden's index and compared the generated classification reports and calculated metrics.

	TPR	FPR	Threshold	Difference
0	0.566046	0.235205	0.643710	0.330841
1	0.564245	0.233469	0.646441	0.330776
2	0.564845	0.234073	0.645271	0.330772
3	0.565896	0.235130	0.643733	0.330766
4	0.564395	0.233658	0.645855	0.330737

```
              precision    recall  f1-score   support

           0       0.88      0.76      0.82     26496
           1       0.38      0.57      0.45      6662

    accuracy                           0.72     33158
   macro avg       0.63      0.67      0.63     33158
weighted avg       0.78      0.72      0.74     33158


F2 score:  0.5143387268411143

AUC ROC Score:  0.7252969770188771
```

Figure 40:Classification Report and ROC_AUC Score using Youden's Index Threshold for Base Logistic Regression coupled with ENN

With a threshold of 0.64 we obtain the classification report by calculating the predicted values of the test data. The recall for the minority class was observed to be 0.57 and the F2-score and AUC_ROC score for this model after SMOTE analysis was 0.52 and 0.72 respectively.

3) ADASYN

Adaptive Synthetic (ADASYN) first finds the impurity of the neighborhood for each of the minority observations by taking the ratio of majority observations in the neighborhood and k. The impurity for each point in the minority class is figured by calculating a impurity ratio, and higher this impurity ratio, more synthetic points are generated for that particular entry. Thus, the name adaptive and hence this technique is a over sampler.

Upon using this technique on the training set of the data, we built a model and optimized the same with a threshold corresponding to its Youden's index and compared the generated classification reports and calculated metrics.

	TPR	FPR	Threshold	Difference
0	0.659562	0.367829	0.401950	0.291733
1	0.659712	0.368018	0.401821	0.291694
2	0.659862	0.368207	0.401675	0.291655
3	0.660012	0.368357	0.401607	0.291655
4	0.659412	0.367791	0.401999	0.291620

	precision	recall	f1-score	support
0	0.88	0.63	0.74	26496
1	0.31	0.66	0.42	6662
accuracy			0.64	33158
macro avg	0.60	0.65	0.58	33158
weighted avg	0.77	0.64	0.67	33158

F2 score: 0.5386388153378444

AUC ROC Score: 0.6892656182924062

Figure 41:Classification Report and ROC_AUC Score using Youden's Index Threshold for Base Logistic Regression coupled with ADASYN

With a threshold of 0.40 we obtain the classification report by calculating the predicted values of the test data. The recall for the minority class was observed to be 0.66 and the F2-score and AUC_ROC score for this model after SMOTE analysis was 0.54 and 0.68 respectively.

4) SMOTE + ENN

This method combines an over sampler with a under sampler, therefore, SMOTE generates synthetic copies for minority class and the ENN method deletes few observations from both these majority and minority classes that are identified as having different class between the observation's class and Knn majority class.

Upon using this technique on the training set of the data, we built a model and optimized the same with a threshold corresponding to its Youden's index and compared the generated classification reports and calculated metrics.

	TPR	FPR	Threshold	Difference
0	0.715551	0.408665	0.263538	0.306885
1	0.715551	0.408741	0.263515	0.306810
2	0.715101	0.408326	0.263847	0.306775
3	0.715251	0.408514	0.263660	0.306736
4	0.714950	0.408250	0.263957	0.306700

	precision	recall	f1-score	support
0	0.89	0.59	0.71	26496
1	0.31	0.72	0.43	6662
accuracy			0.62	33158
macro avg	0.60	0.65	0.57	33158
weighted avg	0.77	0.62	0.65	33158

F2 score: 0.5642354946381648

AUC ROC Score: 0.703906947385815

Figure 42:Classification Report and ROC_AUC Score using Youden's Index Threshold for Base Logistic Regression coupled with SMOTE+ENN

With a threshold of 0.26 we obtain the classification report by calculating the predicted values of the test data. The recall for the minority class was observed to be 0.72 and the F2-score and AUC_ROC score for this model after SMOTE analysis was 0.56 and 0.70 respectively.

We observe that none of these class imbalance treatment methods seem to have a significant effect on the metrics of the logistic regression model. Therefore, we moved onto building other linear, non-linear and ensemble models that could have improved results probably because most of the variables in our dataset are categorical. Therefore, an improved model is expected with a non-linear model construction.

LINEAR, NON-LINEAR AND ENSEMBLE MODELS

As seen in the logistic models after improving them using Youden's index and applying class imbalance Techniques we see that the ROC AUC score improved to 0.74 and F2 score improved to 0.58. Next, we tried to improve the model based on other linear and non-linear models such as:

- Logistic Regression Model
- Decision Tree Model
- Random Forest Model
- Gaussian Naïve Bayes Model
- Bernoulli Naïve Bayes Model
- AdaBoost Model
- Gradient Boosting Model
- CatBoost Model
- LightGBM Model
- XGBoost Model
- KNN Model

1) Logistic Regression

Logistic regression is the process of modelling the probability for a discrete outcome provided an input variable. Usually, logistic regression models provide a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can

give results where there are more than two possible discrete results. Logistic regression is a useful analysis method for classification scenarios, where we are trying to check if a new sample fits best into a particular category

2) Decision Trees

Decision tree is one of the most powerful and popular tools for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node symbolises a test on an attribute, each branch stands for an outcome of the test, and each leaf node (terminal node) holds a class label.

A tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The construction of a decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory analysis. Decision trees can handle high-dimensional and huge volume data. In general, decision tree classifier has good accuracy.

3) Adaboost

AdaBoost also called Adaptive Boosting is a technique in Machine Learning employed as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level which is Decision trees with only 1 split. These trees are also called Decision Stumps.

This algorithm builds a model and gives equal weights to all the data points. It then assigns higher weights to points that are wrongly classified. All the points which have higher weights are given more importance in the following model. It will keep training models until and unless a lower error is received.

4) KNN

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning techniques. K-NN algorithm assumes the similarity between the new data and available data and put the new data into the category which is most similar to the available categories. The algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears, then it can be easily classified into a proper category by using K- NN algorithm. K-NN is a non-parametric algorithm, i.e., it does not make any assumptions on given data.

5) Random Forest

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from every tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and reduces the problem of overfitting.

6) LightGBM

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- It has faster training speed and higher efficiency.
- It takes up lower memory usage.
- It provides better accuracy.
- It has the support of parallel, distributed, and GPU learning.
- It is capable of handling large-scale data.

7) Catboost

CatBoost is a recently open-sourced machine learning algorithm from Yandex. It can easily be combined with deep learning frameworks like Google's TensorFlow and Apple's Core ML. It can work with diverse data types to help solve diverse range of problems that businesses face today. Also, it provides the best-in-class accuracy.

It has two most important advantages:

- It gives state-of-the-art results without extensive data training as compared to other machine learning methods,
- It provides powerful out-of-the-box support for the more descriptive data formats.

8) Xgboost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) to solve many data science problems in a fast and accurate way.

9) Gradient Boost

Gradient boosting is a method known for its speed and accuracy, most probably with large and complex datasets. This algorithm starts by building a decision stump and then assigning equal weights to every data point. Then it increases the weights for all the points which are misclassified and lowers the weight for those that are correctly classified. A new decision stump is made for these weighted data points. The objective behind this is to improve the predictions made by the first stump.

10) Bernoulli Naive Bayes

Bernoulli Naive Bayes is a type of Naive Bayes. Naive Bayes is a classification algorithm of Machine Learning based on Bayes theorem which gives the likelihood of occurrence of the event. Naive Bayes classifier is a probabilistic classifier which given an input predicts the probability of the input being classified for all the classes. It is also called conditional probability.

Two crucial assumptions made for Naive Bayes Classifier are as:

- The attributes are independent of each other and does not affect each other's performance and this is the reason it is called 'naive'.
- The features are given equal importance. All features are necessary to predict outcome and are given equal importance.

11) Gaussian Naive-Bayes

Naive Bayes can be applied to real-valued attributes, mostly by assuming a Gaussian distribution. This type of naive Bayes is called Gaussian Naive Bayes. Other functions can be used to estimate the distribution of the data, but the Gaussian (or Normal distribution) is the easiest to work with because you only need to find the mean and the standard deviation from the training data.

Model building

For each of the above models we have used the same 4 class imbalance techniques, tuned them using Randomize dSearchCV() function and evaluated them based on their F2 score. Based on the best F2 score, a model was picked and its CV 5-fold mean and standard deviation was calculated that is used to calculate the Variance and Bias error of each model respectively.

Upon doing so, we built 55 models and calculated few of the scoring metrics for each of these models that have been listed in *table 5* below.

Model Name	Accuracy train	Precision train	Recall train	F1 train	F1 Avg Weighted train	F2 train	ROC AUC train	Accuracy test	Precision test	Recall test	F1 test	F1 Avg Weighted test	F2 test	ROC AUC test
RandomForestClassifier SMOTE	0.6822	0.6247	0.9123	0.7416	0.6644	0.8354	0.7831	0.5382	0.2892	0.8831	0.4357	0.5741	0.626	0.747
RandomForestClassifier ENN	0.7977	0.7251	0.9668	0.8287	0.7913	0.9064	0.9082	0.5382	0.2892	0.8831	0.4357	0.5741	0.626	0.7546
XGBClassifier SMOTE	0.6789	0.6244	0.8982	0.7367	0.6627	0.8258	0.6789	0.5405	0.2876	0.8635	0.4315	0.5775	0.6166	0.6612
RandomForestClassifier ADASYN	0.6819	0.6379	0.8765	0.7384	0.6681	0.8155	0.7508	0.5514	0.2905	0.8472	0.4327	0.5893	0.6125	0.7353
GradientBoostingClassifier ADASYN	0.6424	0.6085	0.8458	0.7078	0.6255	0.7846	0.6462	0.5123	0.2728	0.8493	0.4129	0.5486	0.597	0.6461
LGBMClassifier	0.5743	0.29	0.7649	0.4205	0.6145	0.5762	0.6438	0.575	0.2916	0.773	0.4235	0.615	0.5811	0.6467
BernoulliNB ADASYN	0.647	0.6596	0.6421	0.6507	0.647	0.6455	0.6809	0.6642	0.3398	0.703	0.4581	0.6964	0.5792	0.7194
XGBClassifier ADASYN	0.5923	0.5715	0.8148	0.6718	0.5694	0.7509	0.5894	0.4598	0.2514	0.8471	0.3877	0.4906	0.5747	0.6068
LogisticRegression ADASYN	0.6467	0.6708	0.6091	0.6385	0.6463	0.6205	0.6965	0.685	0.3531	0.6725	0.463	0.7137	0.5695	0.7386
AdaBoostClassifier	0.2019	0.2019	1	0.336	0.0679	0.5585	0.6193	0.2019	0.2019	1	0.336	0.0679	0.5585	0.6136
LGBMClassifier ADASYN	0.5122	0.5122	1	0.6774	0.347	0.84	0.5	0.2019	0.2019	1	0.336	0.0679	0.5585	0.5
GaussianNB ADASYN	0.5802	0.5634	0.8009	0.6615	0.5571	0.7386	0.6603	0.4457	0.2427	0.8227	0.3748	0.4765	0.5566	0.6939
XGBClassifier ENN	0.7945	0.8029	0.7875	0.7951	0.7945	0.7905	0.7955	0.6693	0.3371	0.6595	0.4461	0.7	0.5536	0.6658
LogisticRegression ENN	0.7982	0.8382	0.7453	0.789	0.7977	0.7622	0.8795	0.706	0.3671	0.6296	0.4638	0.7301	0.5509	0.7438
GaussianNB ENN	0.7652	0.7777	0.751	0.7641	0.7652	0.7562	0.8205	0.6581	0.327	0.6552	0.4363	0.6904	0.5457	0.6957
GaussianNB SMOTE	0.6494	0.6412	0.6785	0.6593	0.6491	0.6707	0.6915	0.631	0.3094	0.6711	0.4235	0.667	0.5439	0.682
BernoulliNB ENN	0.7809	0.8037	0.7506	0.7763	0.7808	0.7607	0.8404	0.6851	0.3458	0.6274	0.4458	0.7125	0.5395	0.7099
LogisticRegression SMOTE	0.6748	0.7033	0.6048	0.6503	0.6732	0.6222	0.7411	0.718	0.3764	0.6036	0.4636	0.739	0.5386	0.7407
BernoulliNB SMOTE	0.6666	0.6839	0.6194	0.6501	0.6658	0.6313	0.7154	0.6937	0.3514	0.6105	0.446	0.7192	0.532	0.7111
KNeighborsClassifier ENN	0.9763	0.9749	0.9784	0.9766	0.9763	0.9777	0.9981	0.6769	0.3372	0.6211	0.4371	0.7055	0.5316	0.6887
GradientBoostingClassifier SMOTE	0.5192	0.5108	0.9055	0.6532	0.4348	0.7843	0.5192	0.2861	0.2052	0.8825	0.333	0.2525	0.5316	0.5088
GradientBoostingClassifier	0.2762	0.2034	0.8863	0.3309	0.2358	0.5303	0.504	0.2752	0.2026	0.8823	0.3296	0.235	0.5281	0.5019
KNeighborsClassifier ADASYN	0.8344	0.8063	0.8906	0.8464	0.8337	0.8724	0.9213	0.6722	0.328	0.5942	0.4227	0.7007	0.5112	0.6849
CatBoostClassifier ENN	0.9655	0.9751	0.9563	0.9656	0.9655	0.96	0.9954	0.7598	0.4227	0.5179	0.4655	0.7684	0.4956	0.7617
KNeighborsClassifier SMOTE	0.8466	0.8231	0.883	0.852	0.8464	0.8703	0.9314	0.6969	0.344	0.5521	0.4239	0.7195	0.4925	0.6873
GradientBoostingClassifier ENN	0.5799	0.575	0.6525	0.6113	0.5776	0.6354	0.5786	0.4985	0.2328	0.6462	0.3423	0.5438	0.4768	0.5537
GaussianNB SMOTE + ENN	0.7137	0.6146	0.6018	0.6081	0.713	0.6043	0.7434	0.7199	0.3574	0.4854	0.4117	0.7345	0.4529	0.6723

Model Name	Accuracy train	Precision train	Recall train	F1 train	F1 Avg Weighted train	F2 train	ROC AUC train	Accuracy test	Precision test	Recall test	F1 test	F1 Avg Weighted test	F2 test	ROC AUC test
AdaBoostClassifier SMOTE + ENN	0.2464	0.261	0.5685	0.3578	0.1878	0.4601	0.3139	0.1908	0.1611	0.7146	0.2629	0.1354	0.4235	0.3864
BernoulliNB SMOTE + ENN	0.7392	0.6773	0.5607	0.6135	0.7331	0.5807	0.7795	0.757	0.4029	0.422	0.4123	0.7591	0.4181	0.6972
XGBClassifier	0.7544	0.3989	0.426	0.412	0.7574	0.4203	0.6317	0.7539	0.3969	0.4207	0.4085	0.7566	0.4157	0.6296
LogisticRegression SMOTE + ENN	0.7506	0.7291	0.5163	0.6046	0.7391	0.5483	0.8126	0.7877	0.4691	0.3898	0.4258	0.7801	0.4034	0.7397
KNeighborsClassifier SMOTE + ENN	0.9233	0.8582	0.9491	0.9014	0.924	0.9294	0.984	0.7614	0.4067	0.3964	0.4015	0.7602	0.3984	0.674
CatBoostClassifier SMOTE	0.8964	0.9609	0.8265	0.8886	0.8959	0.8503	0.9554	0.8127	0.5711	0.2917	0.3861	0.7878	0.3233	0.7517
DecisionTreeClassifier SMOTE	0.6219	0.839	0.3018	0.4439	0.5788	0.3461	0.6481	0.8092	0.5534	0.2854	0.3766	0.7842	0.316	0.6336
RandomForestClassifier SMOTE + ENN	0.7536	0.8135	0.4315	0.5639	0.7306	0.4762	0.8294	0.8092	0.5534	0.2854	0.3766	0.7842	0.316	0.7444
DecisionTreeClassifier SMOTE + ENN	0.7536	0.8135	0.4315	0.5639	0.7306	0.4762	0.7184	0.8092	0.5534	0.2854	0.3766	0.7842	0.316	0.6338
DecisionTreeClassifier	0.8117	0.5644	0.2965	0.3888	0.7878	0.3276	0.6436	0.8092	0.5534	0.2854	0.3766	0.7842	0.316	0.636
DecisionTreeClassifier ENN	0.6971	0.9356	0.4315	0.5906	0.6741	0.4836	0.736	0.8092	0.5534	0.2854	0.3766	0.7842	0.316	0.6337
DecisionTreeClassifier ADASYN	0.5682	0.7937	0.2122	0.3348	0.5034	0.2486	0.6017	0.8092	0.5534	0.2854	0.3766	0.7842	0.316	0.6331
CatBoostClassifier ADASYN	0.8959	0.9644	0.8273	0.8906	0.8956	0.8515	0.9562	0.8139	0.5804	0.2836	0.381	0.7876	0.3159	0.7527
LGBMClassifier SMOTE	0.4466	0.4416	0.4032	0.4215	0.4456	0.4103	0.4466	0.473	0.1666	0.4023	0.2357	0.5247	0.3136	0.4466
GaussianNB	0.7911	0.4724	0.2926	0.3614	0.7714	0.3167	0.6978	0.7895	0.4658	0.2893	0.3569	0.7697	0.313	0.6965
KNeighborsClassifier	0.8502	0.7373	0.4014	0.5198	0.8322	0.4416	0.8701	0.8036	0.5265	0.2749	0.3612	0.7784	0.304	0.6872
CatBoostClassifier	0.8594	0.8391	0.3755	0.5189	0.8371	0.4222	0.8626	0.8169	0.6089	0.2605	0.3649	0.7864	0.2941	0.7557
GradientBoostingClassifier SMOTE + ENN	0.4881	0.3458	0.4333	0.3846	0.4964	0.4124	0.4767	0.4885	0.1591	0.3578	0.2203	0.5389	0.2863	0.4397
CatBoostClassifier SMOTE + ENN	0.9428	0.9712	0.8708	0.9183	0.942	0.8892	0.9834	0.8228	0.6775	0.2337	0.3475	0.7864	0.269	0.7606
RandomForestClassifier	0.825	0.6918	0.2408	0.3572	0.7894	0.2769	0.7465	0.8231	0.6823	0.2319	0.3462	0.7863	0.2672	0.7449
LGBMClassifier SMOTE + ENN	0.7393	0.8728	0.3441	0.4936	0.7023	0.3915	0.6574	0.8194	0.6523	0.2267	0.3365	0.7826	0.2607	0.5981
BernoulliNB	0.8195	0.6858	0.1957	0.3046	0.7768	0.2284	0.7137	0.8172	0.6719	0.1856	0.2909	0.7731	0.217	0.7128
XGBClassifier SMOTE + ENN	0.6323	0.5042	0.2417	0.3268	0.5919	0.2698	0.5514	0.728	0.2675	0.1995	0.2286	0.7124	0.2102	0.5307
LogisticRegression	0.8199	0.758	0.1588	0.2626	0.7692	0.1886	0.7409	0.8183	0.7502	0.1502	0.2503	0.7661	0.1788	0.7398
AdaBoostClassifier ENN	0.2194	0.1219	0.0873	0.1018	0.2044	0.0926	0.2276	0.4516	0.085	0.1756	0.1145	0.5042	0.1447	0.3485
AdaBoostClassifier SMOTE	0.3255	0.2149	0.1315	0.1632	0.2991	0.1426	0.3381	0.4516	0.085	0.1756	0.1145	0.5042	0.1447	0.3485
AdaBoostClassifier ADASYN	0.3228	0.2286	0.1356	0.1702	0.296	0.1476	0.342	0.4516	0.085	0.1756	0.1145	0.5042	0.1447	0.3485
LGBMClassifier ENN	0.502	0.9933	0.0165	0.0325	0.3446	0.0206	0.7272	0.8004	0.7801	0.0164	0.0322	0.7158	0.0204	0.6371

Table 5: Scoring metrics of all the constructed models

Accuracy: Accuracy is the sum of True Positives and True Negatives divided by the total number of entries. Unless there is a class balance accuracy won't be accurate.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Precision is the total number of True Positives out of all the positives predicted by the model

$$Precision = \frac{TP}{TP + FP}$$

Recall: Recall is the total number of True Positives out of the actual positives predicted by the model. Recall is also called sensitivity or True Positive Rate.

$$Recall = \frac{TP}{TP + FN}$$

F1 Score: F1 Score sums up the predicted performance of the constructed model by taking into account two competing metrics – precision and recall. Therefore, F1 score is the Harmonic mean of Precision and recall of the model. By default, F1 Score is calculated for '1' Class Label.

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Weighted F1 Score: This metric is calculated by calculating the average of all per-class F1 scores while taking into account each class's actual occurrence in the dataset that is called support.

F2 Score: This F-Beta Measure signifies the effect of increasing the importance of recall and lowering the importance of precision. By maximizing Recall measures, we actually minimize False Negatives, which means this F-Beta measure focuses more on minimizing False Negatives.

$$F2\ Score = \frac{\left((1 + 2^2) * Precision * Recall\right)}{(2^2 * Precision + Recall)}$$

ROC-AUC: It stands for 'Area Under the ROC curve'. This provides the aggregate measure of performance across all possible classification thresholds. Minimum ROC-AUC score for a model can't be lesser than 0.5.

CV 5-Fold - Mean: Cross Validation (CV) is a method in statistics used to evaluate and compare machine learning algorithms by dividing the data into n segments by taking a scoring method. We have segmented the data into five for cross validation of each model by using the scoring method as 'F2 Score', the mean F2 Score of each of the models in cross validation was

calculated. This metric can be used to calculate the Variance Error by subtracting the metric from 1.

CV 5-Fold SD: The Standard Deviation (SD) of these F2 Scores for each of the segment in the cross validation of each model is calculated. This metric is otherwise called Bias Error of the model.

In the Formulae above the terms TP, TN, FP, FN mean True Positives, True Negatives, False Positives, False Negatives respectively, where:

- **True Positive (TP)**: The model Predicting True while the actual value is also true.
- **True Negative (TN)**: The predicting false when the actual value is also false.
- **False Positive (FP)**: The model predicting True when the actual value is False. This is also known as Type I Error.
- **False Negative (FN)**: The model predicting false when the actual value is true. This is also known as Type II Error.

Table 5 as seen above depicts a scorecard of all 55 models that we constructed with all the metrics that are mentioned above. Also, this scorecard was sorted based of the 'F2 Score' from high to low.

We observed two models viz., RandomForestClassifier+Smote and RandomForestClassifier+ENN with the same highest F2 Score of 0 .626. Also, we noticed that the second model overfits the training data relative to the first model by comparing the F2 Scores of the trained and test sets of these two models as seen in the figure below.

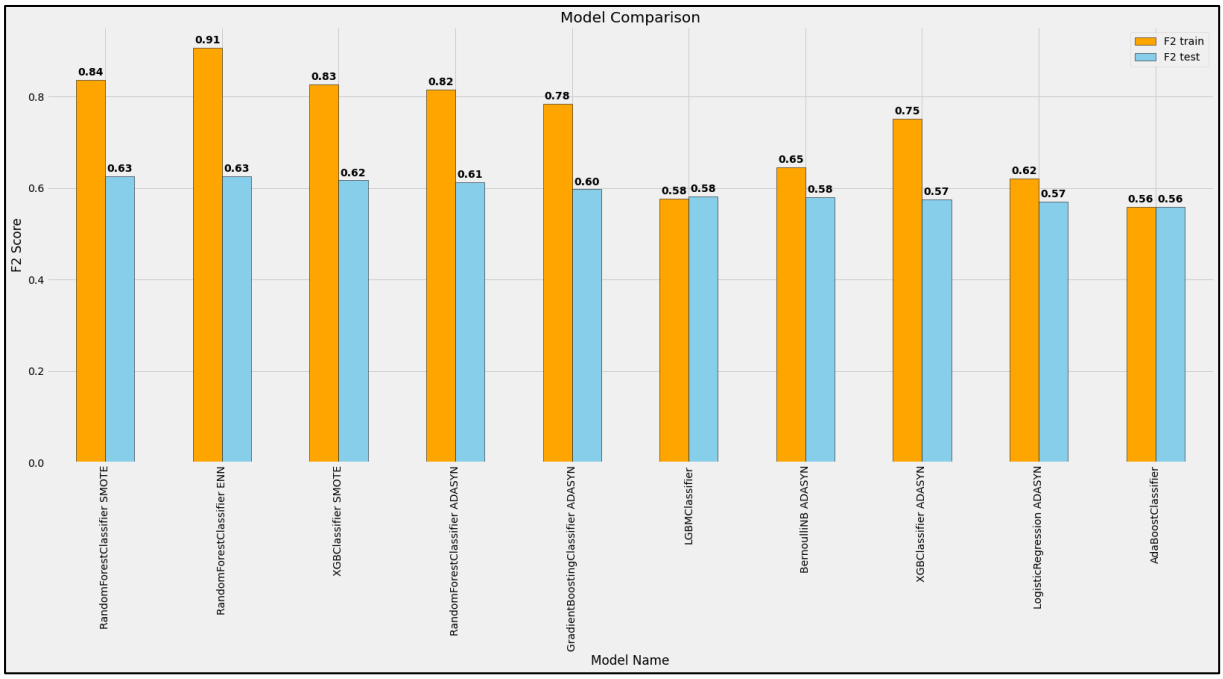


Figure 43: Boxplots comparing the train and test F2 scores for the top 10 models

Therefore, we pick the RandomForestClassifier+Smote as the best model for our given dataset. We cross-validated this particular model to calculated CV 5-fold scores that can be used to calculate the variance error and bias error of this model. The scoring metric to identify the scores was 'F2 score' since that's the chief metric of evaluation of the models constructed. We obtained a CV 5-fold mean score of 0.6252 and CV 5-fold standard deviation of 0.0033 for this RandomForestClassifier coupled with SMOTE technique.
Therefore, the Variance Error and the Bias Error for this particular model is 1 – 0.6252 = 0.3748 and 0.0033 respectively.

This model was further boosted to potentially decrease the variance error and thereby increase the performance of the model further.

Improving the model efficiency

Several boosting techniques were used to boost this Random-forest model to increase or enhance this model's performance. Upon boosting it was observed that none of the boosting classifiers significantly enhanced model performance or decreased the variance error of the model. Being a bagging model inherently, the random-forest model didn't show any significant change upon boosting. The training data overfit even more significantly and therefore, we pick the best model to be the RandomForestClassifier+SMOTE with tuned hyperparameters to be the best predictor model for this dataset with a F2 test metric of 0.626 and an ROC-AUC score of 0.747.

MODEL EXPLANATION

We understand the Random Forest classifier coupled with the SMOTE technique of oversampling to be the best fitting model.

One popular machine learning algorithm that is part of the supervised learning is the Random Forest classifier that can be used for both classification and regression problems. This particular model is a concept of ensemble learning that combines multiple classifiers to solve the problem and enhance model performance. Multiple Decision Tree classifiers on various subsets of the chosen dataset are generated and the Forest model takes the average of these trees to improve the predictive accuracy of the dataset. Therefore, this model takes prediction from every decision tree instead of relying on one and based on the voting majority, the final output is predicted. The number of trees and subsequently the number of subsets of the dataset is passed on to the classifier function and greater the number of trees, higher is the accuracy and prevents the problem of overfitting.

This makes the random forest model inherently a bagging model.

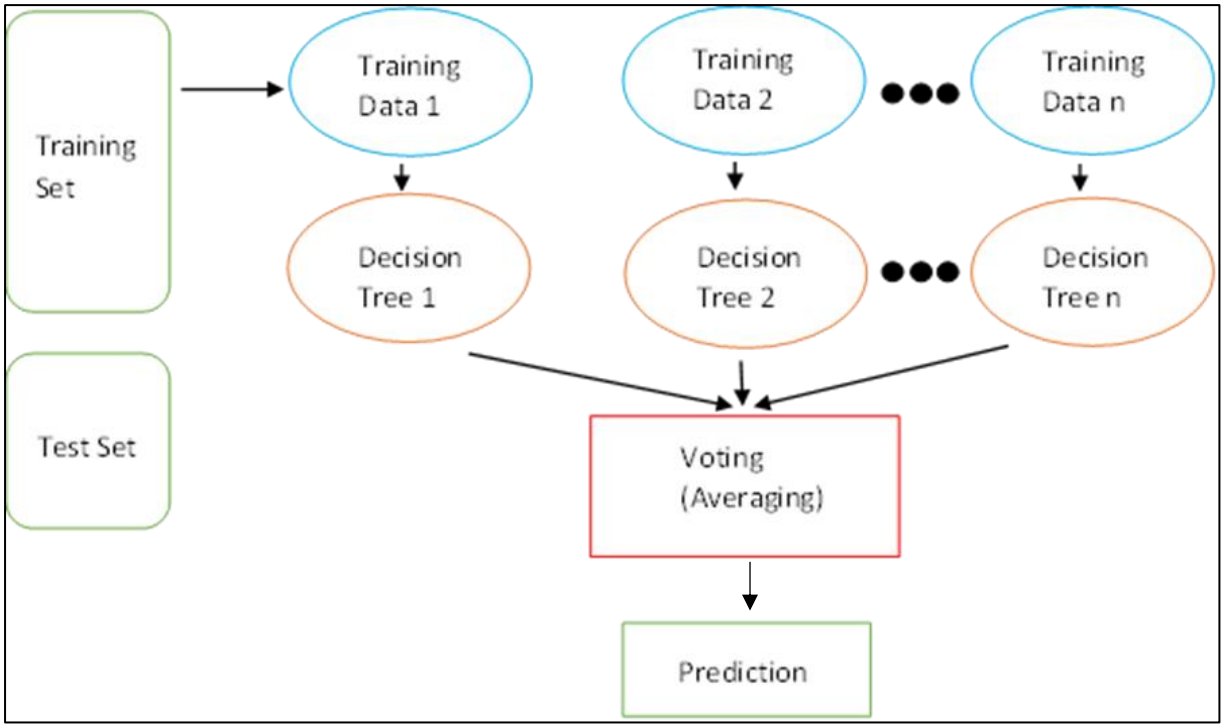


Figure 44:Working of the Random Forest algorithm

As seen in *Figure 44*, the random forest model combines the output of multiple trees to predict the class of the dataset, and in doing so, some decision trees may predict the correct while the other may potentially not. This is where voting comes into play where together with all the trees the correct output is predicted. For this we have two assumptions for this classifier:

Assumptions of Random Forest Classifier

- The prediction made by each tree must have very low correlations.
- For the classifier to predict accurate results rather than a guessed result, there should be some actual values in the feature variable of the dataset.

The random forest model takes lesser time relatively compared to other machine learning algorithms. The predicted output is with high accuracy even for large datasets that runs very efficiently with this model. Also, the model predicts with fairly high accuracy when a large amount of data is missing.

Advantages of Random Forest Model

- It reduces overfitting in decision trees and helps to improve the accuracy.
- It is flexible to both classification and regression problems.
- It works with both categorical and continuous values.
- It automates missing values present in the data.
- Normalizing of data is not required as it uses a rule-based approach.
- The model offers relative feature importance that allows us to select the most contributing features of the predictor model.

Disadvantages of Random Forest Model

- It requires much computational power as well as resources as it builds numerous trees to combine their outputs.
- It Also requires much time for training as it combines a lot of decision trees to determine the class.
- Due to ensemble of decision trees, it also suffers interpretability and fails to determine the significance of each variable.
- Although it can be used for regression problems, it is not advisable to solve regression tasks using this particular model.

MODEL UNDERSTANDING AND BUSINESS INTERPRETATION

As we understand, precision is the accuracy of all the positive predictions and recall is the ratio of instances being positive to the ones that are actually detected as positives and they are both metrics of performance for classification algorithms.

In the case of our model, the target variable is no-show which is a negative trait. The 0s indicate a patient showing up and the 1s which are usually the positive traits is actually the negative trait in our dataset. With respect to our dataset, we define recall of the minority class to be the percentage of data belonging to no-show which the model correctly predicts as belonging to that no-show class. Precision therefore, is the measure which gives the correct prediction of no-shows out of all predicted no-shows or accurate prediction of no-shows per false prediction of no-show.

Ideally, we expect both the values of recall and precision to be high for a prediction model but unfortunately, this doesn't happen in real scenarios. The precision-recall trade-off can be significant in models where recall is more important than precision or vice versa.

From a hospital or clinic's perspective, the organization wouldn't want the model to predict a show, when in actuality the patient doesn't show-up on the scheduled date. This would mean that the hospital should look to minimize the false negatives which consequently would maximize recall of the model to avoid loss of money and resources. The other way where the model predicts a no-show when in actuality the patient shows up on his scheduled date that's given by false positives isn't as significant as the false negatives from the business perspective and hence, we give more weightage and significance to recall over precision in our model.

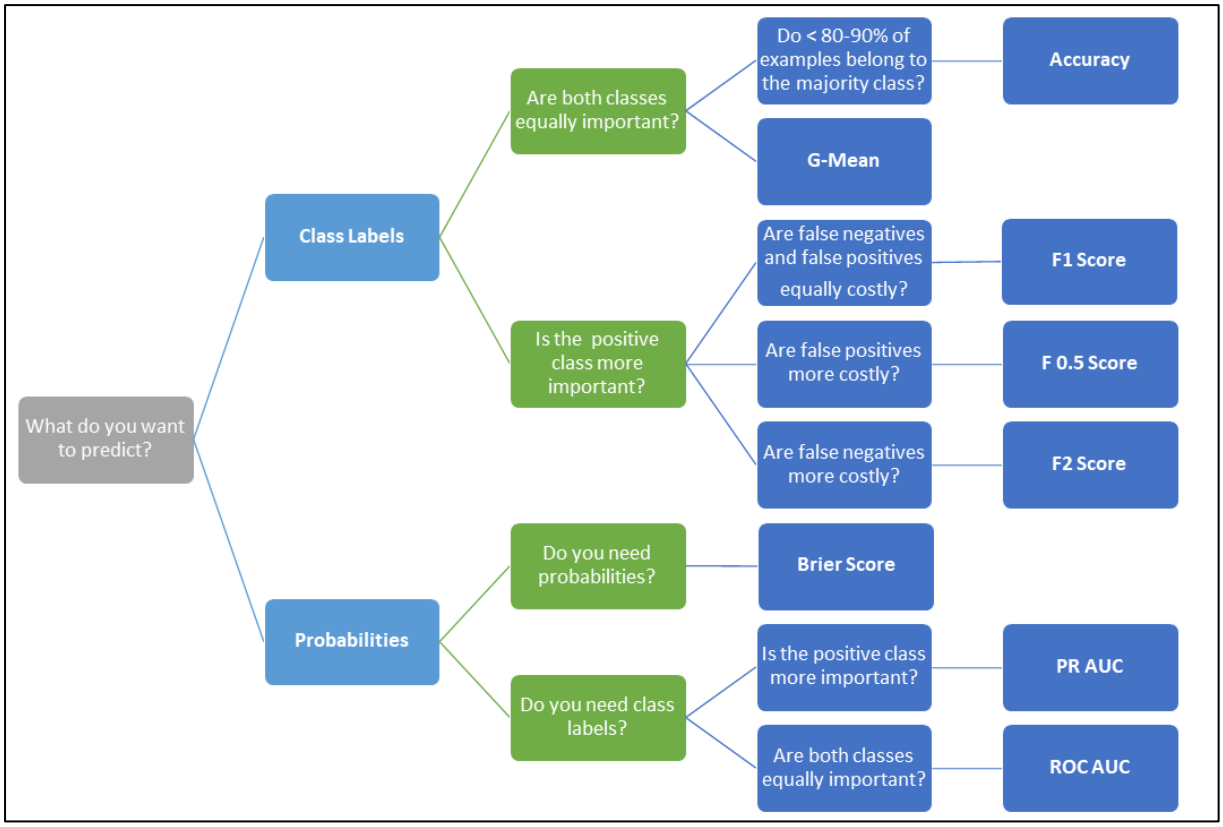


Figure 45:Criteria to select model metrics to evaluate model performance

Based on literature, as seen in *Figure 45*, when false negatives prove to be costlier for the business organisation, we evaluate the model based on a F-Beta metric called the F2 score.

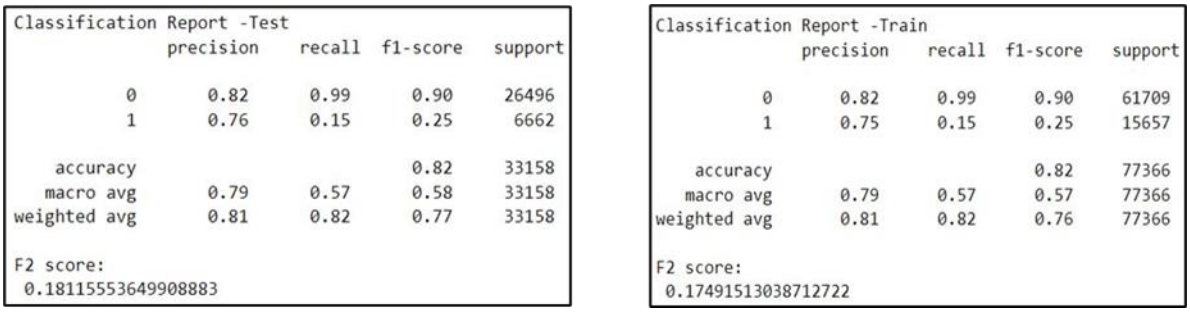
The F-Beta measure is a mere generalisation of the F-measure where a configuration parameter called beta is added where the default value of this beta measure is. For a default value of 1, the F1 measure of the model is calculated where the model is evaluated taking both the precision and recall metrics into account using a single score. A smaller beta measure of 0.5 is taken when the model has to be given more weight to precision and less to recall, whereas the larger beta value of 2 is given when more weightage is given to recall over precision.

- F-Beta = ((1 + beta^2) * Precision * Recall) / (beta^2 * Precision + Recall)

Therefore, considering our model building we take the best model to be the one with the highest F2 score = 0.626 of Random-forest model coupled with SMOTE.

Comparison with benchmark

Benchmark model is the Base model-2 where a logistic regression model was built that included the feature engineered variables. The model metrics can be observed from the figure given below for both the train and test set.



Classification Report -Test

	precision	recall	f1-score	support
0	0.82	0.99	0.90	26496
1	0.76	0.15	0.25	6662
accuracy			0.82	33158
macro avg	0.79	0.57	0.58	33158
weighted avg	0.81	0.82	0.77	33158

F2 score:
0.18115553649908883

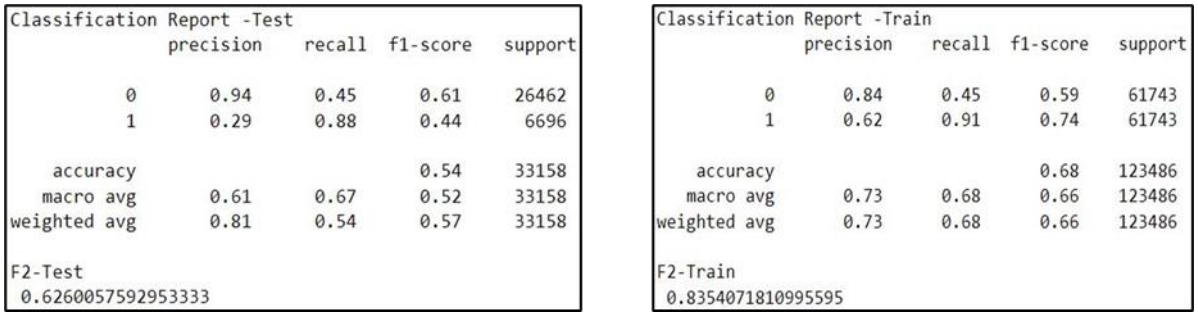
Classification Report -Train

	precision	recall	f1-score	support
0	0.82	0.99	0.90	61709
1	0.75	0.15	0.25	15657
accuracy			0.82	77366
macro avg	0.79	0.57	0.57	77366
weighted avg	0.81	0.82	0.76	77366

F2 score:
0.17491513038712722

Figure 46:Model metrics of the benchmark model

We noticed that the recall score of the minority class which is 1 is 0.15 and the F2 score of the test set is 0.1811 for this base model. We compared it with the metrics of the tuned random forest model coupled with SMOTE analysis that are given in the figure below.



Classification Report -Test

	precision	recall	f1-score	support
0	0.94	0.45	0.61	26462
1	0.29	0.88	0.44	6696
accuracy			0.54	33158
macro avg	0.61	0.67	0.52	33158
weighted avg	0.81	0.54	0.57	33158

F2-Test
0.6260057592953333

Classification Report -Train

	precision	recall	f1-score	support
0	0.84	0.45	0.59	61743
1	0.62	0.91	0.74	61743
accuracy			0.68	123486
macro avg	0.73	0.68	0.66	123486
weighted avg	0.73	0.68	0.66	123486

F2-Train
0.8354071810995595

Figure 47: Model metrics of the best model for our dataset

We noticed that the recall scores have greatly improved compared to the base model. The recall measure we observe of the minority class of the test set is 0.88 and the F2 score of the same test set is 0.626 which lets us conclude that the hyperparameter tuned random forest model has greatly improved from the base model and would work with much higher performance than the latter.

FEATURE IMPORTANCE

The feature importance outlines which features are relevant to the model as well as business. It can help with better understanding of the solved problem and sometimes lead to model improvements by employing the feature selection.
In our case, we have used both Gini Importance (Built in Feature Importance) and Feature Importance computed using SHAP Values.

Gini Importance: In case of Gini Importance, the features for internal nodes are selected using some criterion, which for classification can be 'Gini impurity' or 'information gain', and for regression is 'variance reduction'. We next measure how each feature reduces the impurity of the split and the feature with highest reduction is selected for internal node. For each feature we collect how on average it reduces the impurity. The average over all trees in the forest is the measure of the feature importance. The biggest advantage of this method is the speed of computation - all needed values are computed during the Radom Forest training. The drawback

of the method is its tendency to select numerical and categorical features with high cardinality as important.

SHAP: SHAP stands for Shapley Additive Explanation. Shapley values calculate the importance of a feature by comparing what a model predicts with and without the feature. However, since the order in which a model sees features can affect its predictions, this is done in every possible order, so that the features are fairly compared. SHAP thereby removes the disadvantage of the Gini Importance.

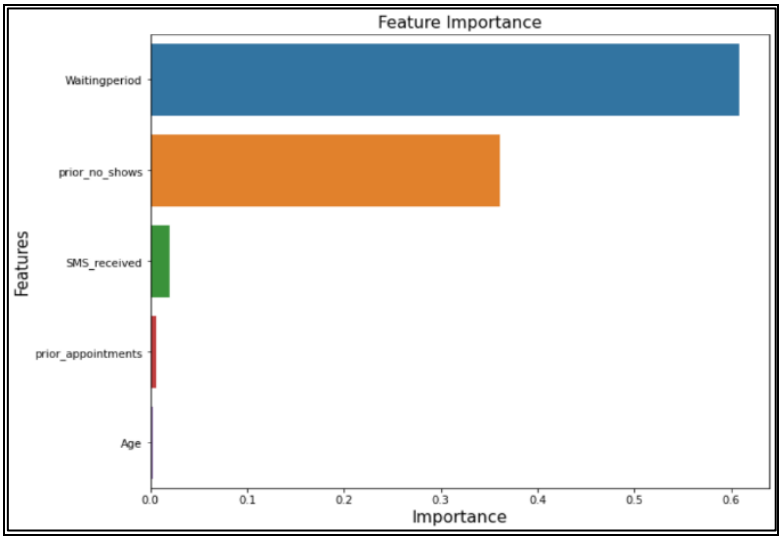


Figure 48:Barplot signifying the feature importance of our model

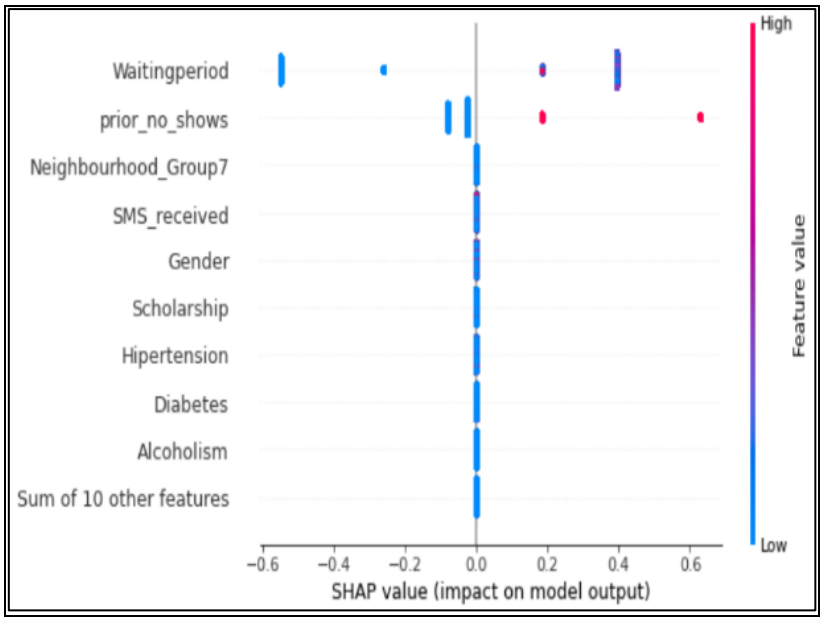


Figure 49: Beeswarm plot signifying feature importance

In the above figures we see that 'Waiting period' is the most important Feature in our model and from the SHAP beeswarm plot we can say that higher the waiting period for a patient before appointment, the likelihood of the patient not showing up will be higher and lower the waiting period the likelihood of patient showing up will be higher. Also, for the second important feature 'prior no shows', the higher the number of prior no shows, higher is the likelihood of the patient not showing up for the appointment. From the Bar plot we can say that sms_received is slightly more important feature in the model prediction compared to the remaining features but from the beeswarm plot we see that its importance is negligible.

BUSINESS SOLUTION

Understanding feature importance is mandatory to provide essential business solutions to cater our problem statements:

1) Patients who have a longer waiting period (that is scheduled their appointments early) most likely forgot about the appointment. A possible solution would be to send them a reminder through phone calls, since SMSs do not seem to be much effective.
2) The reason for patients not showing up for the appointments might also be that the patients with a history of no shows have become more habitual and taken it for granted that they will get appointment on another date if they require. So, if the health Centers are unable to convince these patients to show up, double-booking their slot could be a possible solution.
3) Patients should be given a time frame to cancel their appointment without any cancellation fee. For Example, A patients can be allowed to cancel their appointment 48 hours before the scheduled appointment date. This will allow to save Hospital resources.
4) Also, if the option to cancel their appointment is given, these empty slots can be given to the patients who needs immediate healthcare services.
5) Another possible solution is to charge an appointment/Booking fee which will be refundable in case they cancel the appointment before 48 hours of scheduled appointment date and it will be included in the consultation fee, if they show up for the appointment. But it will not be refunded if they do not show up for the appointment.

LIMITATIONS, CHALLENGES AND SCOPE

LIMITATIONS

1) The dataset originates from Public Health Centers around a city in Brazil which consists of appointment details of around 1 lakh patients with scheduled doctor appointments. The model might have been more robust if the data would have belonged to different regions of the country.

2) The duration of the data collected is for a span of just 6 weeks, which is too small as it does not give us the freedom to analyze how appointment varies across all seasons in a year.
3) Additional number of features might have allowed for better prediction of the model - like Doctor Fees, how far the patients' home is from the health center, etc.

CHALLENGES

1) We faced challenges on robust model tuning and while calculating the cross-validation score on most of the models. Due to computational limitations, we are limited to using Randomized Search as hyper parameter tuning techniques instead of using Grid Search, HyperOpt, etc. Additionally, we have just computed Train and Test score (despite knowing its limitations) instead of cross-validation as it took more than 12 hours to complete even half of the execution.
2) Overfitting / underfitting in the models could have been drastically reduced if we were able to do proper model tuning, which again brings us back to existing computational limitations.

SCOPE

1) We can try performing hyper-parameter tuning with different and more advanced tuning techniques.
2) Computing the variance-error and bias-error to check for both the performance and stability in model prediction, and then perform bagging / boosting depending upon the scores for more improvement.
3) Train the model again once the newer data is available to check whether it is producing more robust results.
4) We can try some other imbalanced technique to get more balanced data in order to achieve better recall and precision.
5) Utilizing some robust data sampling techniques to obtain a smaller sample of the data which truly represents the population data so as to reduce the computational complexity.
6) Exploring some web-based IDE for python like Google Collab for model training and tuning with faster lead time.
7) At the end we can deploy the final built model using streamlit and create an app / web interface while giving it to the client as business solution as it will be an easier tool for them to predict the newer data points as and when they have it.

REFERENCES

1. Alaeddini, A., Yang, K., Reddy, C., & Yu, S. (2011). A probabilistic model for predicting the probability of no-show in hospital appointments. *Health Care Management Science*, *14*(2), 146–157. https://doi.org/10.1007/S10729-011-9148-9

2. AlMuhaideb, S., Alswailem, O., Alsubaie, N., Ferwana, I., & Alnajem, A. (2019). Prediction of hospital no-show appointments through artificial intelligence algorithms. *Annals of Saudi Medicine*, *39*(6), 373. https://doi.org/10.5144/0256-4947.2019.373

3. Liu, D., Shin, W. Y., Sprecher, E., Conroy, K., Santiago, O., Wachtel, G., & Santillana, M. (2022). Machine learning approaches to predicting no-shows in pediatric medical appointment. *Npj Digital Medicine 2022 5:1*, *5*(1), 1–11. https://doi.org/10.1038/s41746-022-00594-w

4. McQueenie, R., Ellis, D. A., McConnachie, A., Wilson, P., & Williamson, A. E. (2019). Morbidity, mortality and missed appointments in healthcare: a national retrospective data linkage study. *BMC Medicine*, *17*(1), 2. https://doi.org/10.1186/S12916-018-1234-0)

5. Mohammadi, I., Wu, H., Turkcan, A., Toscos, T., & Doebbeling, B. N. (2018). Data Analytics and Modeling for Appointment No-show in Community Health Centers. *Journal of Primary Care & Community Health*, *9*. https://doi.org/10.1177/2150132718811692

6. Sun, Y., Wong, A. K. C., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(4), 687–719. https://doi.org/10.1142/S0218001409007326

7. *Tour of Evaluation Metrics for Imbalanced Classification*. (n.d.). Retrieved July 14, 2022, from https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

8. *A Gentle Introduction to the Fbeta-Measure for Machine Learning*. (n.d.). Retrieved July 14, 2022, from https://machinelearningmastery.com/fbeta-measure-for-machine-learning/