Aston University
BIRMINGHAM UK

AM41UD-T3 – Course Work

Submitted by: Syed Muhammad Wali Rizwan
Student ID: 200218625
Email: 200218625@aston.ac.uk
Submitted to: Dr Yordan Raykov

Medical No-Show Appointments

Contents

Abstract

Now a day's healthcare sector all over the world face a big problem with patients who booked their appointment and did not show up on the following day. As a result of this the efficiency and productivity of healthcare system compromises which leads towards higher costs and diminish an overall productivity of medical staff. This became the challenge for healthcare sector and this problem needs to be solved in order to enhance the productivity of healthcare system. So as a Data Analyst our objective is to design and implement such predictive models which lower an impact of patient no show up problem to the hospital on the day of their appointment. These predictive models help medical staff to overcome the problem of overbooking by patient and could be beneficial in managing their daily patients.

In order to demonstrate our analysis, we use dataset from kaggle website and this belongs to one of the hospital in Brazil. The dataset used in this analysis was collected in May 2016 which contains 110527 entries following with 13 attributes and they are Patient-ID, Appointment-ID, Gender, Schedule-day, Appointment-day, Age, Neighborhood, Scholarship, Hypertension, Diabetes, Alcoholism, Handicap, SMS-received and No-show. This dataset is analyzed via Feature engineering and machine learning algorithms using python language on jupyter notebook. In addition, we showed the evaluating performance of machine learning models and their accuracy and also picked the best predictive model for this classification problem. Our goal for this is explore the factors that affects missing appointments and can be alert a day before via SMS to such patients so that they can't miss their appointment.

Introduction

Since all regulations and services of healthcare system are gone up against with patients not appearing on their appointments which brings frustration and irritation with medical staff with inefficiency of organization and social cost. This also irritates the following patients who wait for their turn. Healthcare staff tries their best to minimize this problem and increase their efficiency towards work. So in this analysis we point out some problems by analyzing the data and give their solution so that by apply these techniques in real life, lots of time and effort could be safe and it also improves the efficiency in healthcare sector. Airline sector is the first to identify this problem because of that they have delays in their schedule and empty seats in the plane which cause money. One study showed that a hospital with 15k no-show annually cost around one million dollars with an average 62 appointment a day. The prediction of whether the patient shows up or not is really hard to predict but with the help of machine learning algorithms we try our best to solve this issue and tries to provide a permanent solution to this obstacle.

Dataset Description

This dataset is taken from Kaggle website and it has the record of about 110k medical appointments. https://www.kaggle.com/joniarroba/noshowappointments. Our analysis mainly focusses on the question of whether or not the patient comes on their appointment. This dataset has some feature parameter which has patient's personal information like their age, gender and medical condition. which we take as an attributes for our data exploration and visualization, based on them we perform our analysis, explore some features to see what are the causes involve in no show process and perform some machine learning algorithms to predict the main causes which impact the possibility that a person would miss their appointment.

Information about given features in dataset which are as follow,

Patient-ID: This column has the patient Id, there is a chance of duplication in this column because some patients might book more than one appointment which leads to the problem of overbooking and patient will see the doctor and goes and his duplicated appointment cause delay for others.

Appointment-ID: This column contains an appointment Id of patients which is given by them so this field should be unique.

Gender: This column has patients gender as male or female.

Scheduled-Day: This column has a Date and time for patient appointment choose by them.

Appointment-Day: This column contains an information about the date and time for their appointment which they called to booked.
Age: This column has the patient's age.

Neighborhood: This column contains the location of neighboring hospitals.

Scholarship: This column contains the information that whether or not the patient is enrolled in welfare program of the country.

Hipertension: This column has an information about the patient's health condition related to hypertension.

Diabetes: This column has an information about the patient's health condition related to diabetes.

Alcoholism: This column has an information of patient about the habit of consumption of alcohol.

Handcap: This column refers that the patient needs any assistant or not.

SMS_received: It indicates that the patient has received a reminder text message for their appointment or not.

No-Show: It is most important column in our dataset as it shows whether or not the patient showed up on the given day. “No” if the patient shows up and “Yes” if they did not show up to their appointment.

Our goal is analysis this dataset gives conclusions with the help of our data analysis skills.

Data Understanding

First we load our dataset in jupyter notebook to see its structure, this data is given as a comma separated file (CSV). After that we clean our dataset by checking if there are any null values in the given columns and rows, this is step and important while dealing with any kind of data, then we checked the type of our given columns as which column contains integer value, which has float and which has strings or object in it.

Dataset

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	...	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	...	1	0	0	0	0	No
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	...	0	0	0	0	0	No
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	...	0	0	0	0	0	No
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	...	0	0	0	0	0	No
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	...	1	1	0	0	0	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110522	2.572134e+12	5651768	F	2016-05-03T09:15:35Z	2016-06-07T00:00:00Z	56	...	0	0	0	0	1	No
110523	3.596266e+12	5650093	F	2016-05-03T07:27:33Z	2016-06-07T00:00:00Z	51	...	0	0	0	0	1	No
110524	1.557663e+13	5630692	F	2016-04-27T16:03:52Z	2016-06-07T00:00:00Z	21	...	0	0	0	0	1	No
110525	9.213493e+13	5630323	F	2016-04-27T15:09:23Z	2016-06-07T00:00:00Z	38	...	0	0	0	0	1	No
110526	3.775115e+14	5629448	F	2016-04-27T13:30:56Z	2016-06-07T00:00:00Z	54	...	0	0	0	0	1	No

10527 rows × 14 columns

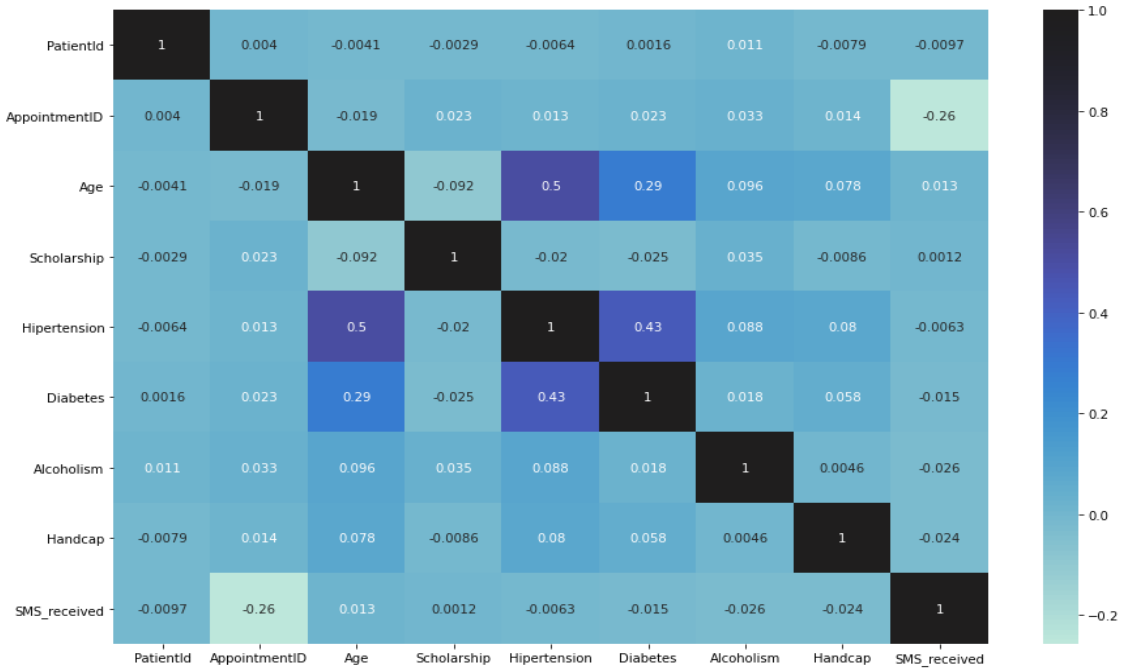
Data types

```
health_care.info()
```

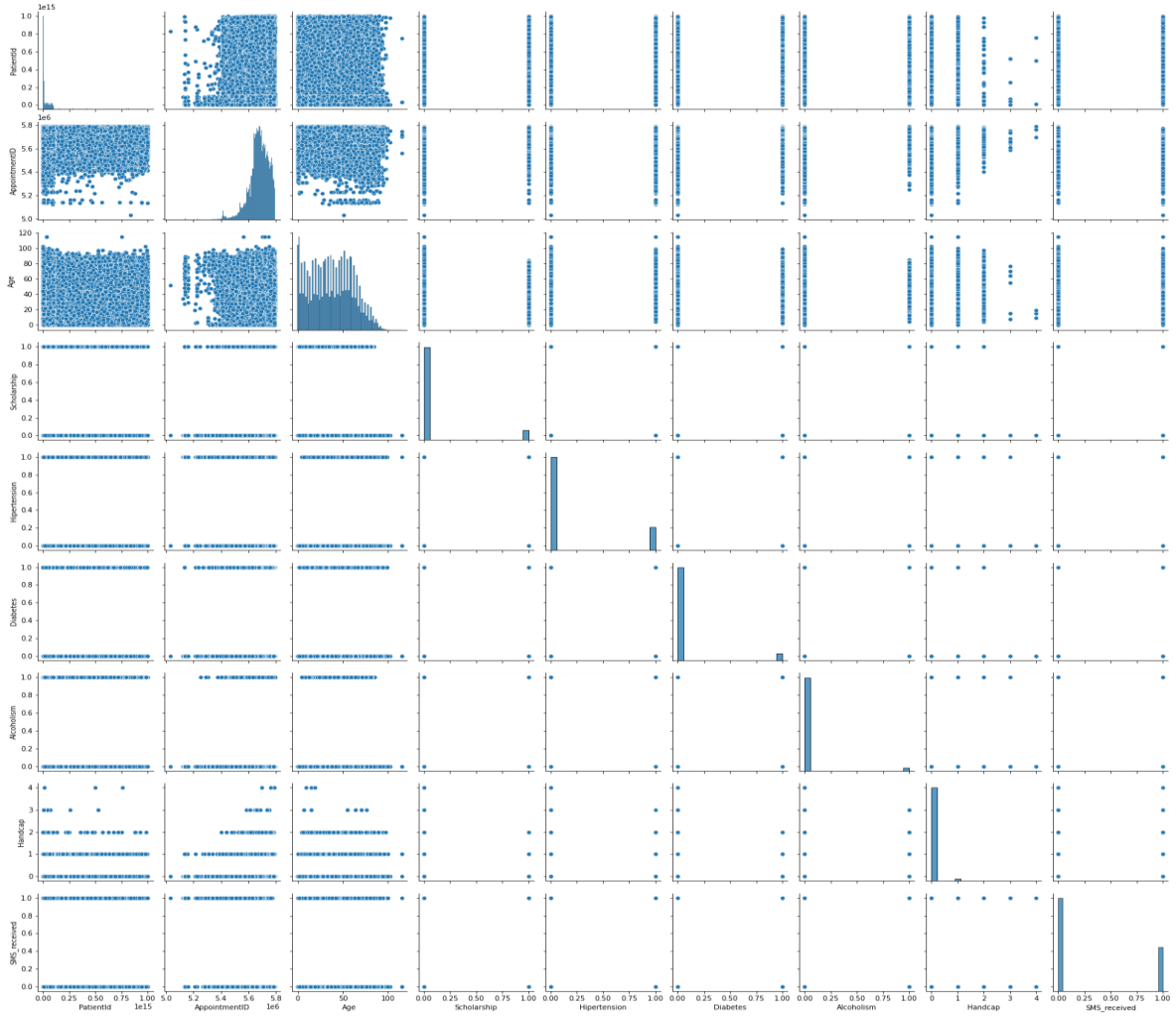
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   PatientId       110527 non-null  float64
 1   AppointmentID   110527 non-null  int64  
 2   Gender          110527 non-null  object 
 3   ScheduledDay    110527 non-null  object 
 4   AppointmentDay  110527 non-null  object 
 5   Age             110527 non-null  int64  
 6   Neighbourhood   110527 non-null  object 
 7   Scholarship     110527 non-null  int64  
 8   Hipertension    110527 non-null  int64  
 9   Diabetes        110527 non-null  int64  
 10  Alcoholism      110527 non-null  int64  
 11  Handcap         110527 non-null  int64  
 12  SMS_received    110527 non-null  int64  
 13  No-show         110527 non-null  object 
dtypes: float64(1), int64(8), object(5)
memory usage: 11.8+ MB
```

Our dataset contains 14 columns and it has 110527 rows, by running a .info() function we identify the data format and types for each column, this gives us insights and help us to plan our strategy and to shape our analysis. At first we need to plot our dataframe and to do this we use python's seaborn library functions known as heat-map and pairplot() this will help us to obtain an understanding for the data types, following we will also point out the data types in each column separately by using pandas library function .unique().

Heatmap



Pair-plot



Above heat-map shows us that there is some correlation between Age and Hypertension with less extent between Hypertension and Diabetes.
By pair-plot we notice that our dataset has two types of data such as categorical data and quantitative data. Doing further more analysis we would be able to associate the exact types for each columns.
Gender, Diabetes, Alcoholism, Handicap, SMS-Received and No-Show are the Categorical Data in our Data frame while Age, Patient-ID, Appointment-ID, Appointment-Day and Scheduled-Day are Quantitative data in our Data frame.

Data Processing

We check our dataset to look for missing values and duplication by running .isnull() and .duplicated() function. We find out that there are no null values in our data set. However, we find out that they are repeated values in patient-ID column and they are 48228. Looking at the cell

above, we observe that the patient-ID contains more values then an appointment, seeing at the output of the value_counts(),

```
P = sum(health_care.PatientId.duplicated())
print('patient-Id','----------',P)
health_care.PatientId.value_counts().head(5)
```

```
patient-Id ---------- 48228
39]: 8.221459e+14    88
     9.963767e+10    84
     2.688613e+13    70
     3.353478e+13    65
     2.584244e+11    62
     Name: PatientId, dtype: int64
```

We notice the 5 patients that booked the highest number of appointments. Now our analysis will mainly depend on the appointments counts rather than patients count but we would reference the number of patients in order to identify any correlation between specific patterns.

Incorrect Data Validation

Now we are checking our data set to look for any incorrect entry or any negative value in it, we find out that in appointment-day column the time portion is set to be zero because of that appointment is not going to be included in this analysis and looking at the Age column we notice that there is one record with (-1) as per review of the documentation of dataset on Kaggle we find out that negative value is for the person who is not born yet while other suggest that this might be mistake so temporarily we will exclude this record from analysis.

In this section, we explore each column separately to identify the presence of any data error or mistake, so that they can be eliminated or fixed, as per our analysis. Patient-Id have more than one appointment so this column is useful in the analysis, Appointment-Id column will also be considered, Gender is categorical column will be included, Scheduled-Day column will only use to calculate the waiting duration between appointments, Age is a quantitative column which we will include, Neighborhood column will limit our analysis to top 20 neighbors with highest number of appointments.

```
AppointmentID -------------- 0
PatientId ------------------ 0
Gender --------------------- ['F' 'M']
ScheduledDay --------------- 103549
AppointmentDay ------------- 27
Age ------------------------ [-1]
Neighborhood --------------- 81
Scholarship ---------------- [0 1]
Hipertension --------------- [1 0]
Diabetes ------------------- [0 1]
Alcoholism ----------------- [0 1]
Handcap -------------------- [0 1 2 3 4]
SMS-Received --------------- [0 1]
No-show -------------------- ['No' 'Yes']
```

Data Preparation

In order to present data as an informative way, we will create data ranges for some unique variables like Age and Waiting duration, while appoint data groups with larger frequencies for certain variable like Neighborhood and determine functions that help us to summarize and visualize our data. All the process will be done in this section and we prepare our data for further analysis.

First we will rename our columns as PatientID, AppointmentID, Gender, ScheduledDay, AppointmentDay, Age, Neighborhood, Scholarship, Hypertension, Diabetes, Alcoholism, Handicap, SmsReceived and Status, by doing this our columns makes our scene as it will allow other reader to quickly identify the column by looking at it. Then we convert Patient-ID as an integer type as it is easy to work with integer values, convert Appointment-Day and Scheduled-Day from string to Date time format while we create two more columns for appointment booking date and appointment day. As the appointment time were set to 00:00 in all appointments therefore we won't include appointment time in our analysis.

	PatientID	AppointmentID	Gender	Age	Neighborhood	...	Status	AppointmentDay	DayofWeek	ScheduledDay	WaitingDuration
0	29872499824296	5642903	F	62	JARDIM DA PENHA	...	No	2016-04-29	4	2016-04-29	0 days
1	558997776694438	5642503	M	56	JARDIM DA PENHA	...	No	2016-04-29	4	2016-04-29	0 days
2	4262962299951	5642549	F	62	MATA DA PRAIA	...	No	2016-04-29	4	2016-04-29	0 days
3	867951213174	5642828	F	8	PONTAL DE CAMBURI	...	No	2016-04-29	4	2016-04-29	0 days
4	8841186448183	5642494	F	56	JARDIM DA PENHA	...	No	2016-04-29	4	2016-04-29	0 days
...	...	...	...	...	...	...	...	...	...	...	...
110522	2572134369293	5651768	F	56	MARIA ORTIZ	...	No	2016-06-07	1	2016-05-03	35 days
110523	3596266328735	5650093	F	51	MARIA ORTIZ	...	No	2016-06-07	1	2016-05-03	35 days
110524	15576631729893	5630692	F	21	MARIA ORTIZ	...	No	2016-06-07	1	2016-04-27	41 days
110525	92134931435557	5630323	F	38	MARIA ORTIZ	...	No	2016-06-07	1	2016-04-27	41 days
110526	377511518121127	5629448	F	54	MARIA ORTIZ	...	No	2016-06-07	1	2016-04-27	41 days

110527 rows × 16 columns

There are five appointments where the appointment was recorded after the actual appointment. We also examine the no-show column and observe that none of the patients who booked their appointment were show up. Now after this process we exclude these five records with the negative waiting duration thus the new dataset would be of 110522 records. Eliminate the row with negative Age value while update the values on other columns. All this data processing is done to better visualize our dataset as for better understanding the data we have to visualize it as this makes the data to understand easily.

Give a new column that include Age groups in which we grouped patient ages in 6 categories from 0 to 119 in years with an interval of 20 years [0-19, 20-39, 40-59,60-79, 80-99 and 100-119], our analysis for this columns are based on these groups and we study these groups to identify that what age group has a higher chance to miss their appointment.

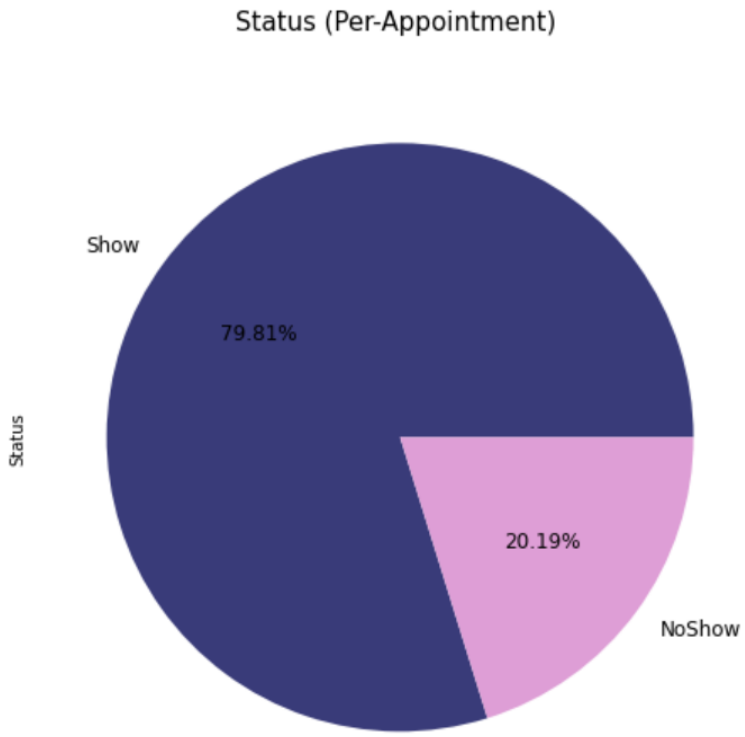
We also define a new column for to include waiting duration ranging from 0-179 with an interval of 30 days, this help to better understand the cause of no show on appointment day. Neighborhood column is quite large as we see before, thus we limit our analysis to the top 20 neighborhood with largest number of appointments.

We also define a new column for to include waiting duration ranging from 0-179 with an interval of 30 days, this help to better understand the cause of no show on appointment day. Neighborhood column is quite large as we see before, thus we limit our analysis to the top 20 neighborhood with largest number of appointments.

For neighborhood column we create a new data frame in which we limit patients to those who are based in one of the top 20 neighborhood with following 3 steps, first we create a list for 20 neighborhoods with highest appointments then in step 2 we create a data frame based on that list and finally we have applied filtration on status columns to include the no show records only. All these technique helps us to better visualize and analyze our data and these are the most important procedures of analyzing any sort of data given data to extract most information from it and to develop better understanding of analysis.

Data Exploration

Show vs No-Show

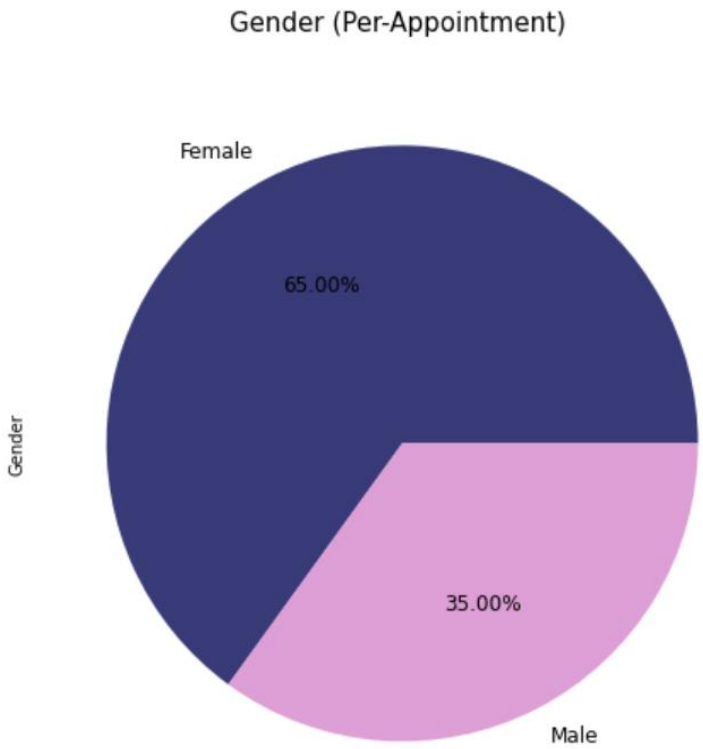


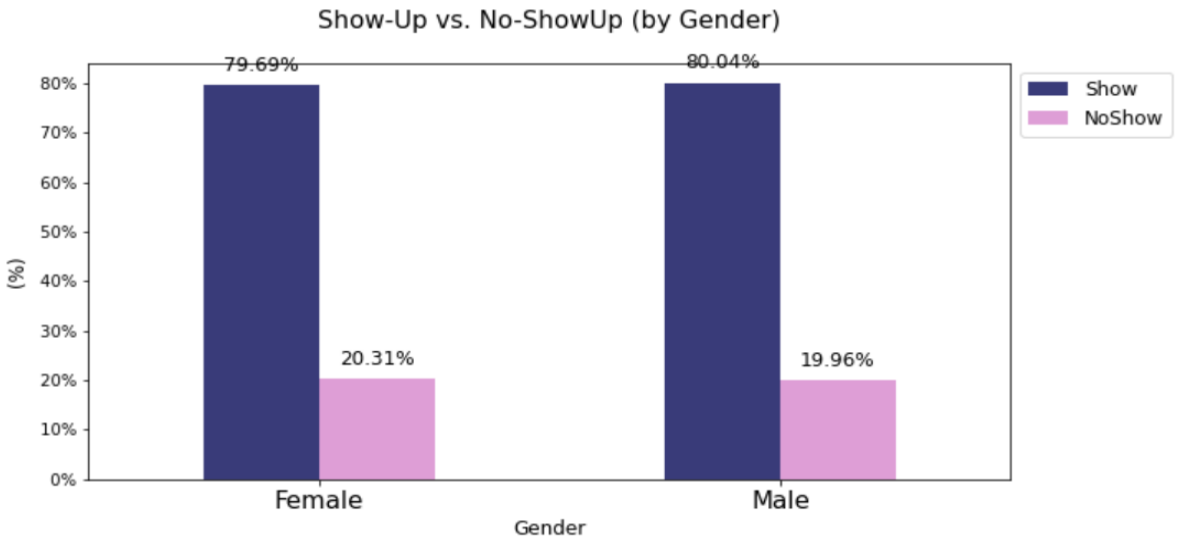
Above chart showed us the overall show-up rate of patients is 79.8% to the hospital on the appointment day, we calculated this chart on the basis of Appointment-Id

Data Features Visualization

On this section we calculate the category proportion with in each variable as per Appointment-Id and present the proportion in percentage on pie chart, then we calculate the show-up rate vs no-show-up rate within each variable and plot the results in bar chart. Our visualization is totally depending on Appointment-Id and not on Patient-Id as we find out that there are duplications in Patient-Id, so we limit our visualization to Appointment-Id for better results and analysis.

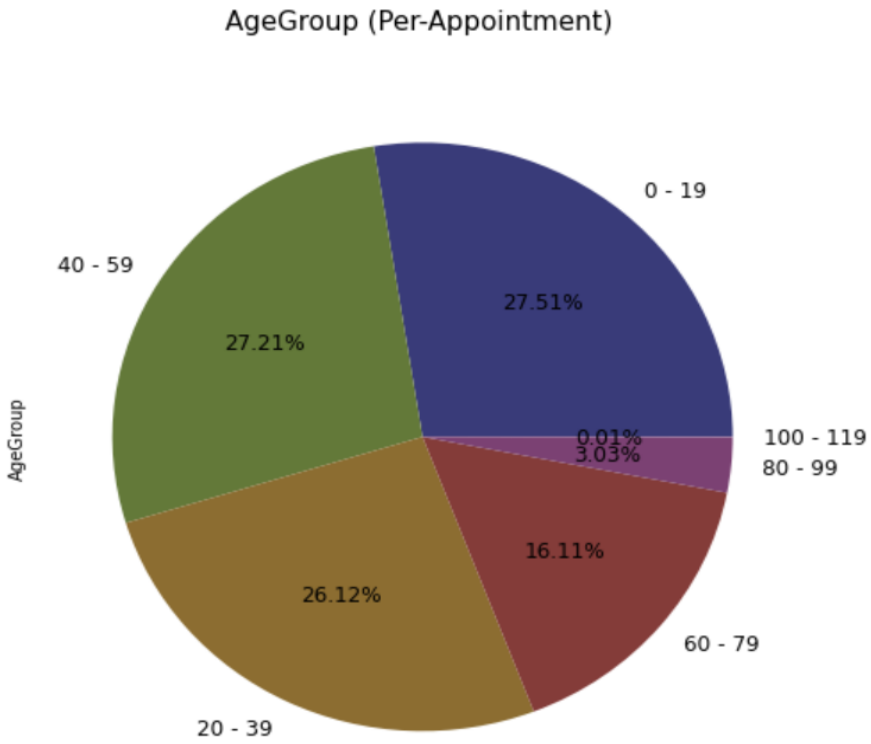
Gender

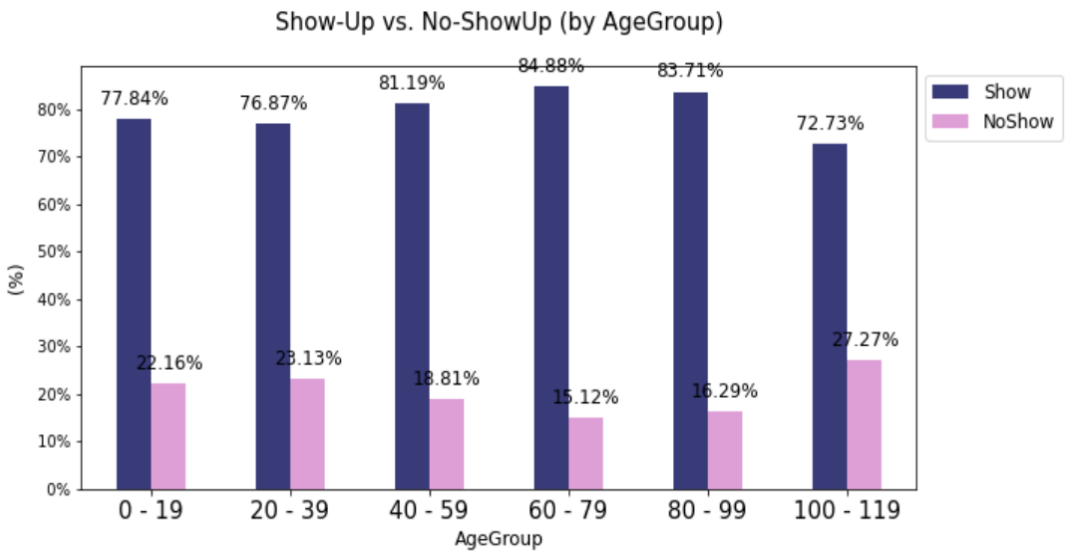




Looking at the above graphs we found out that the proportion of appointments where patients are females is greater than that of male patients while show-up rate of male patients are slightly greater than that of female patients.

Age Group

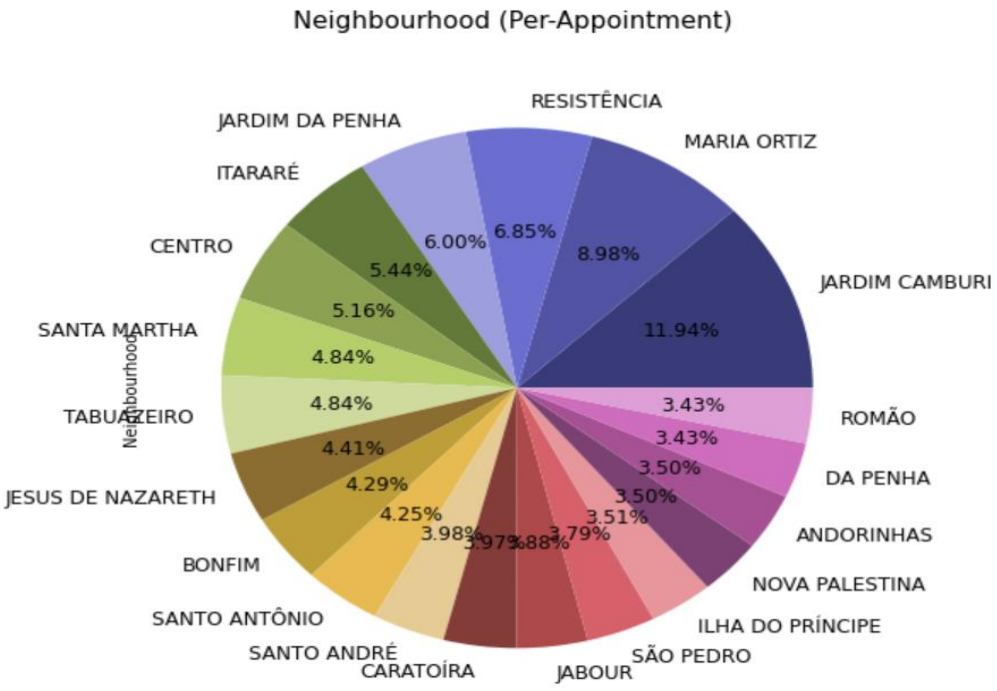


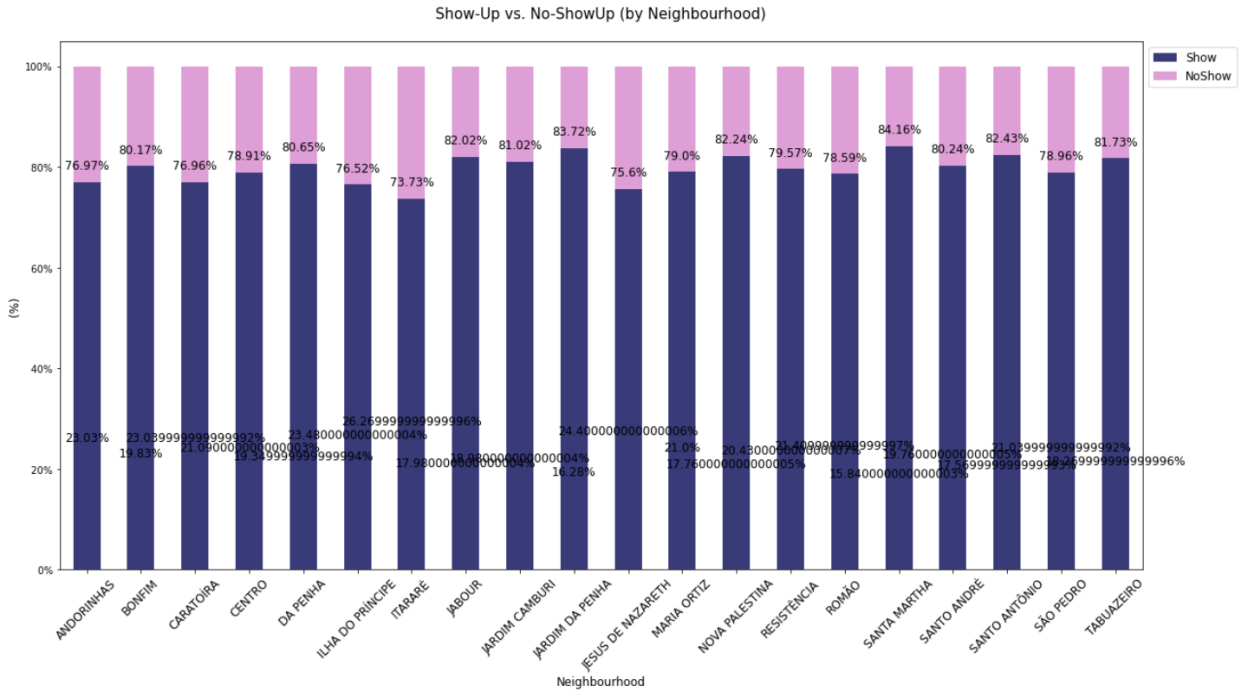


Looking at the above graphs, we found out that the proportion of appointments where patients between (0-19) group is higher among the other age groups while for (100-119) age group is the smallest with 0.01%. The show-up rate of age group (60-79) is the highest with 84.88% and it is lowest for (100-119) age group.

Neighborhood

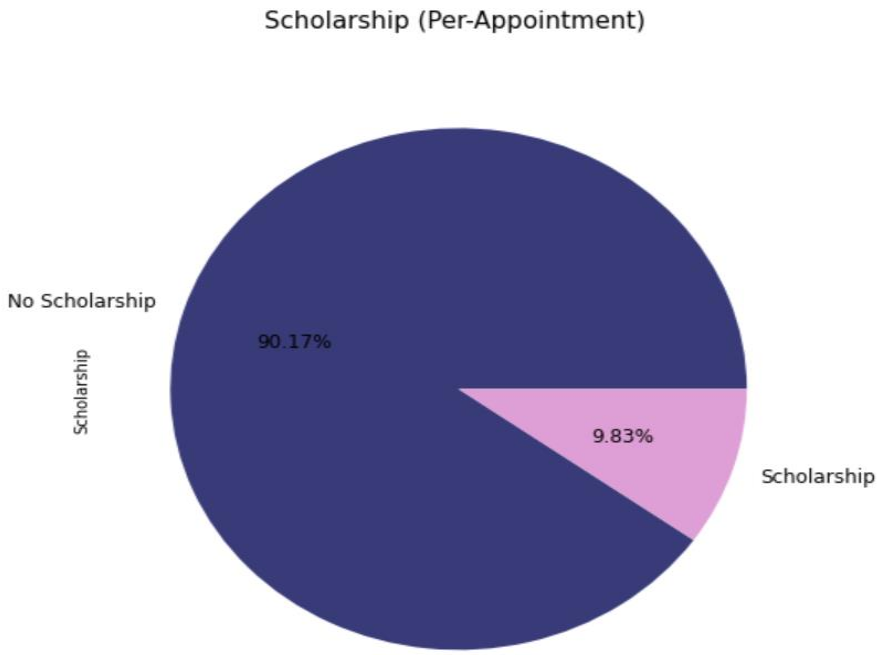
As we mentioned before that we limited the number of neighborhood in our analysis to 20 instead of 81 in order to include the top 20 neighborhood with highest appointment numbers.

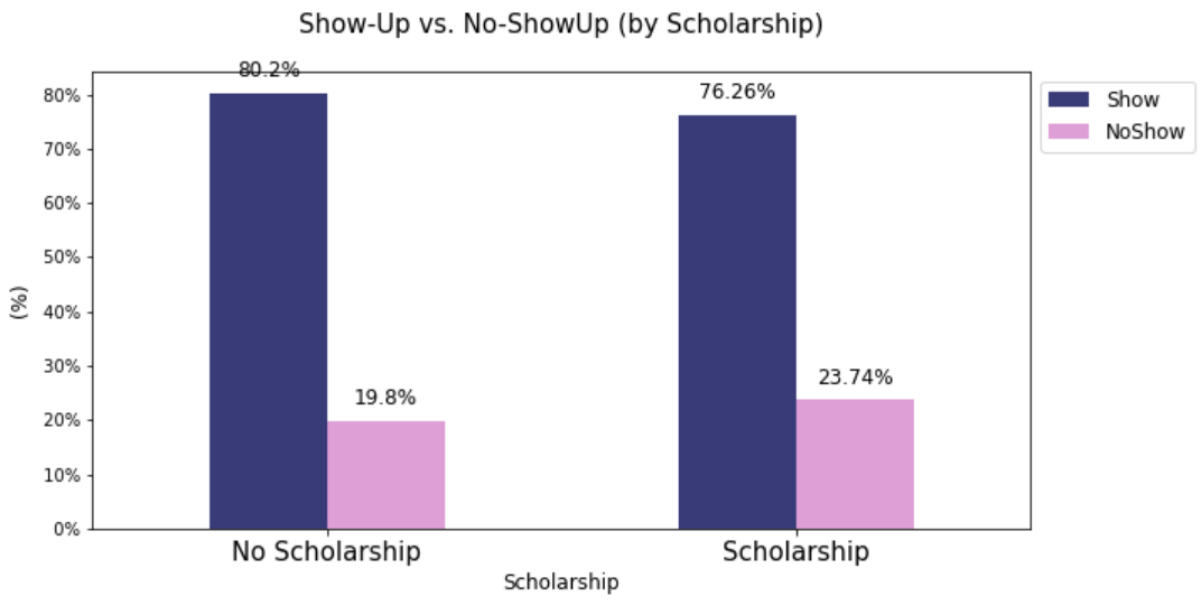




Looking at the above graphs, we found out that the proportion of appointments where patients are lived near Jardim Camburi is higher, among other neighborhoods with 11.94%, where as for DA Penha and Romao is the lowest with 0.01%. The show-up rate of Santa Martha is the highest among other hospitals with 84.16% and it is lowest for Itarare with 73.73%.

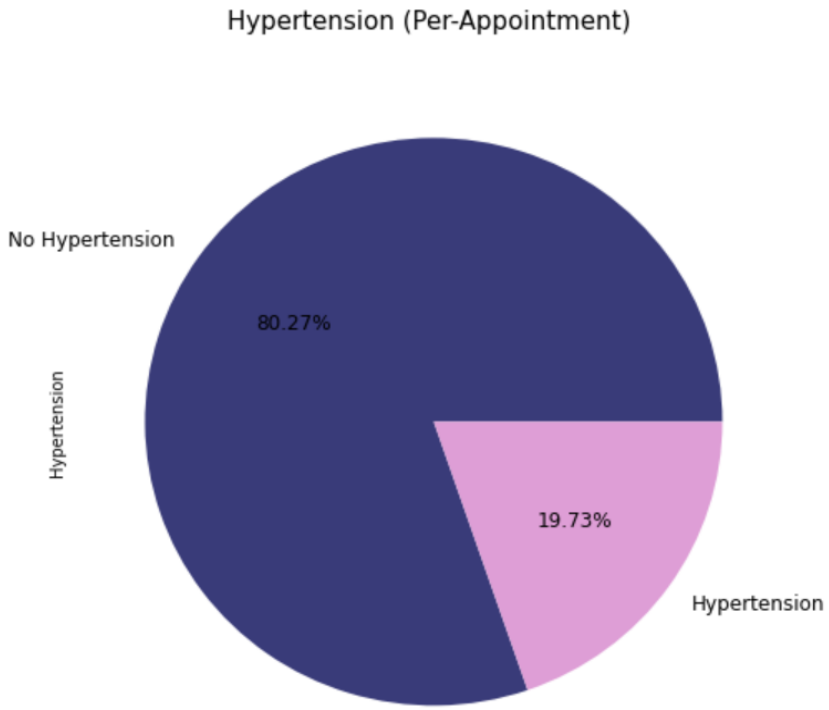
Scholarship

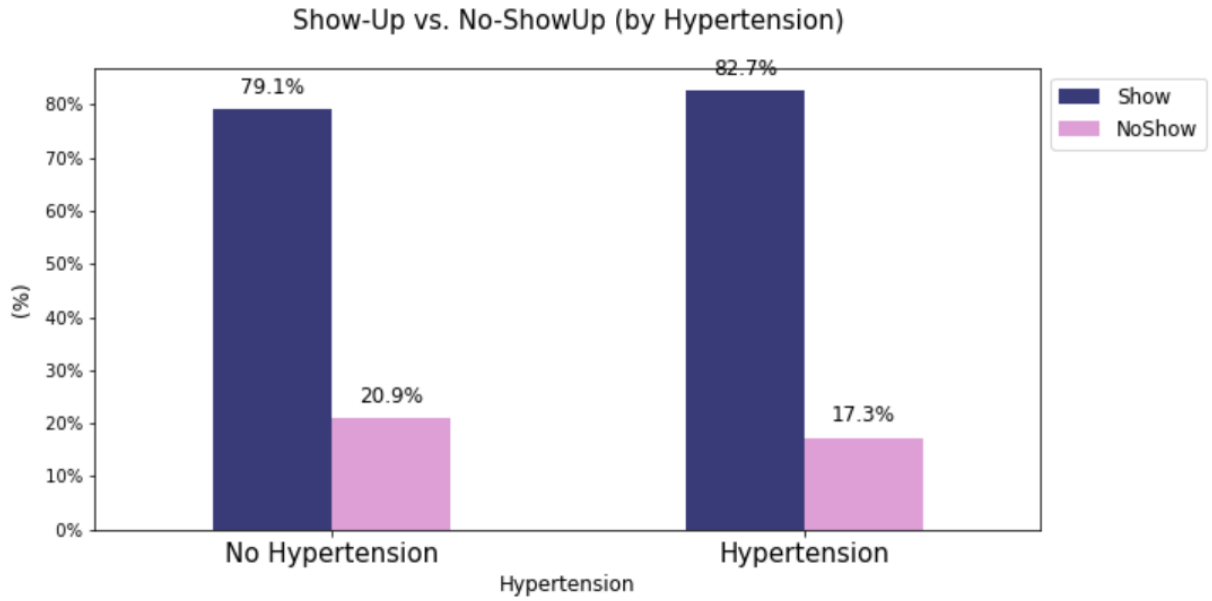




Looking at the above graphs, we found out that the proportion of appointments where patients do not have a scholarship is the largest while the show-up rate of patients who do not have a scholarship is the greater with 80.2% and the patients who have scholarship is 76.26%

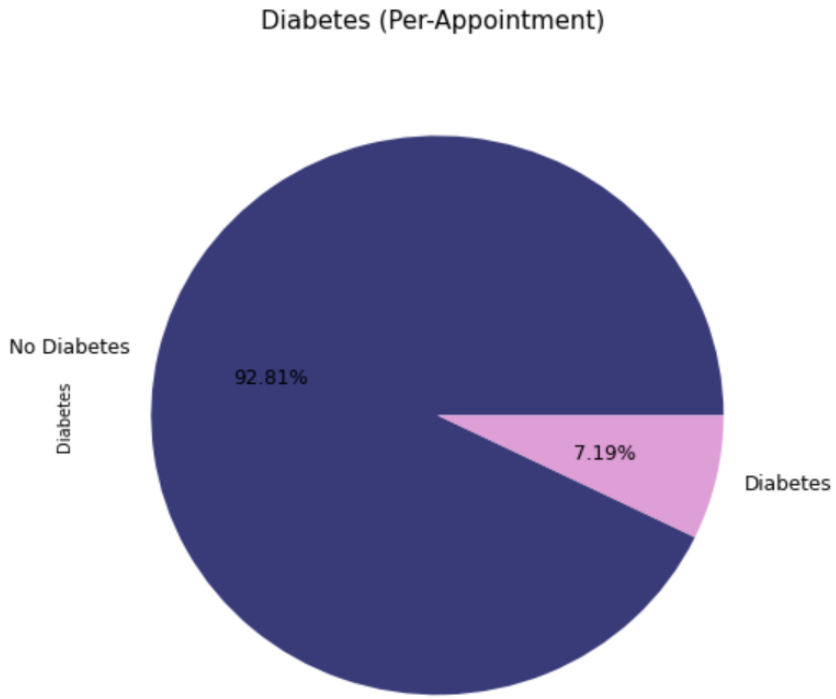
Hypertension

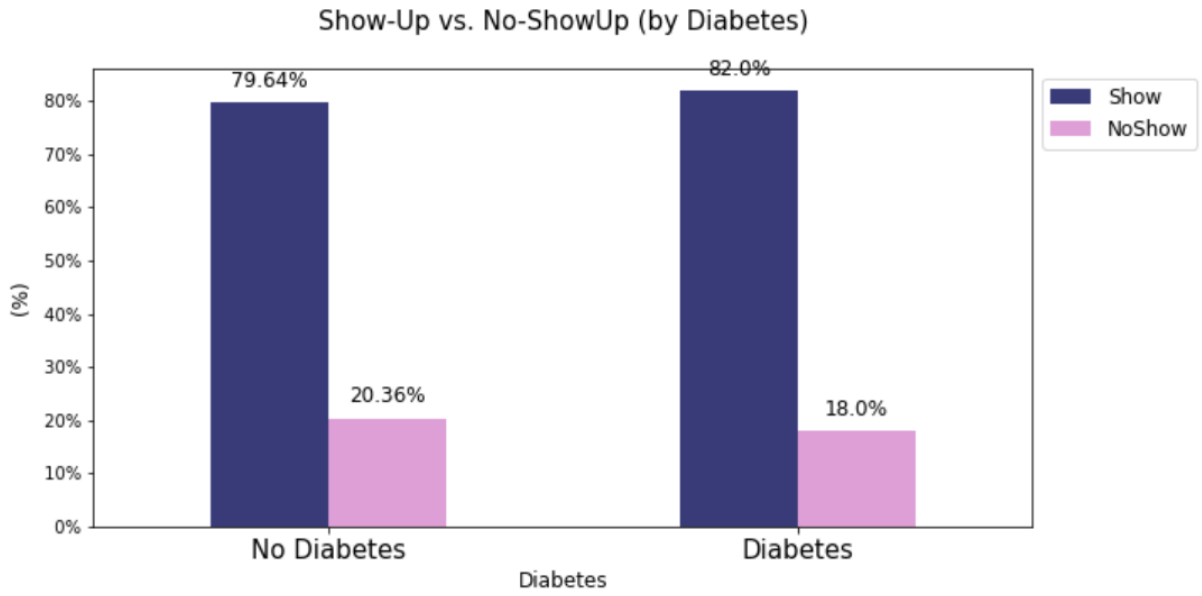




Looking at the above graphs, we found out that the proportion of appointments where patients do not have hypertension is greater while the show-up rate of patients who have hypertension is the higher with 82.7% and the patients who do not have hypertension is 79.1%

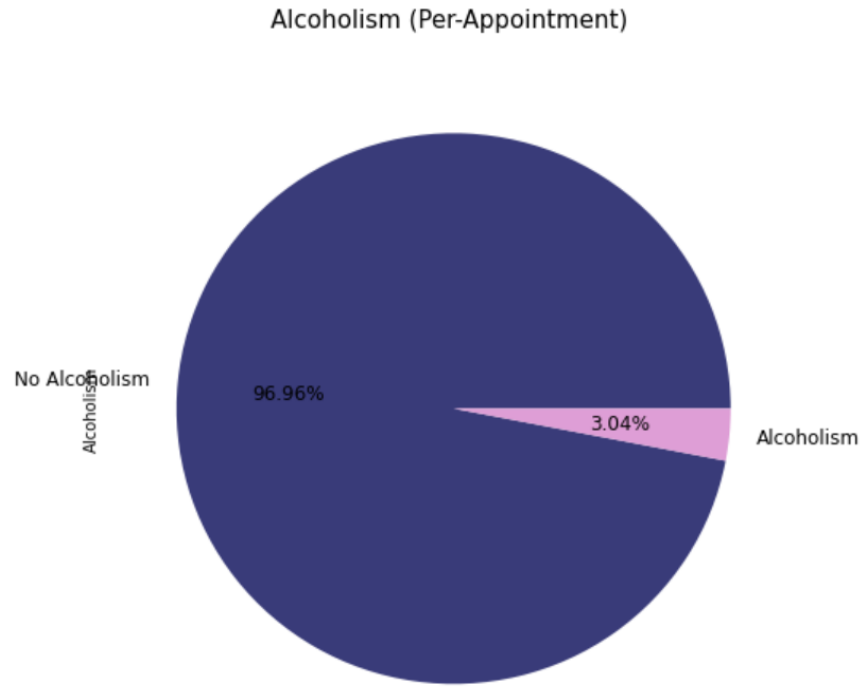
Diabetes

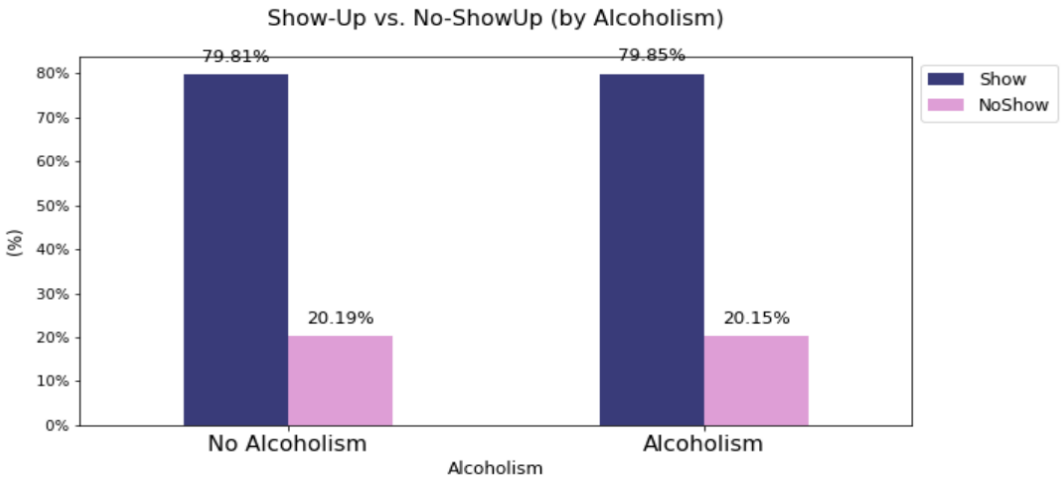




Looking at the above graphs, we found out that the proportion of appointments where patients do not have diabetes is higher with 92.81% while the show-up rate of patients who have diabetes is largest with 82% and the patients who do not have diabetes is 79.64%

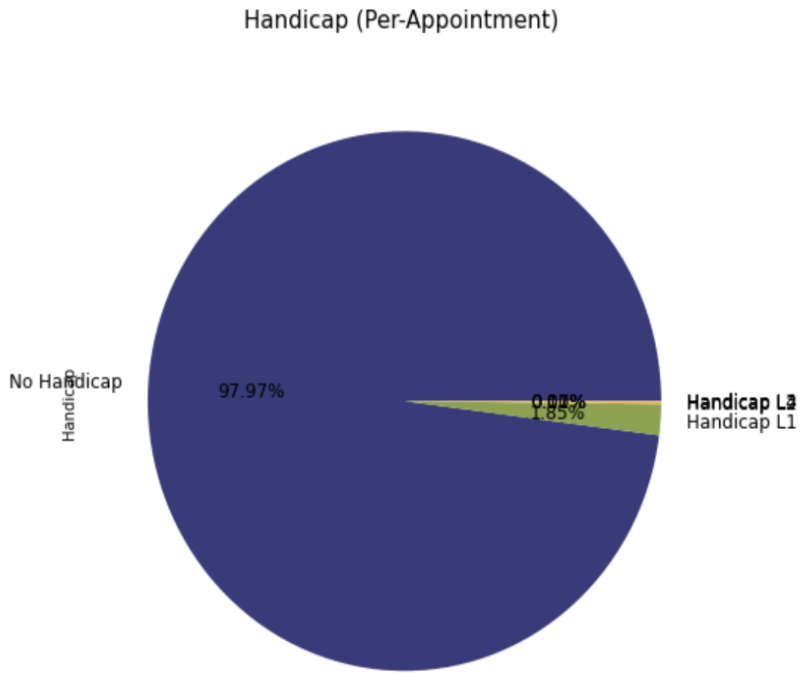
Alcoholism

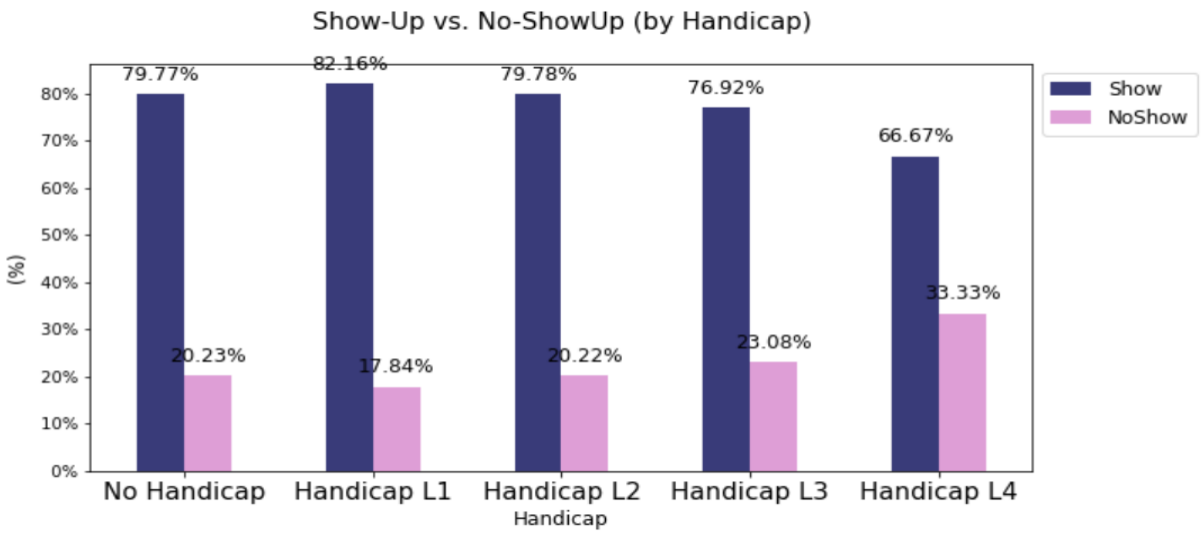




Looking at the above graphs, we found out that the proportion of appointments where patients who do not drink are the largest with 96.96% while the patients who experience Alcoholism (3.04% of the total population) have the highest show-up rate of 79.85% (higher than the average show-up rate), where patients who don't experience Alcohol (96.96 % of the total population) have a show-up rate of 79.81% (equal to the average show-up rate).

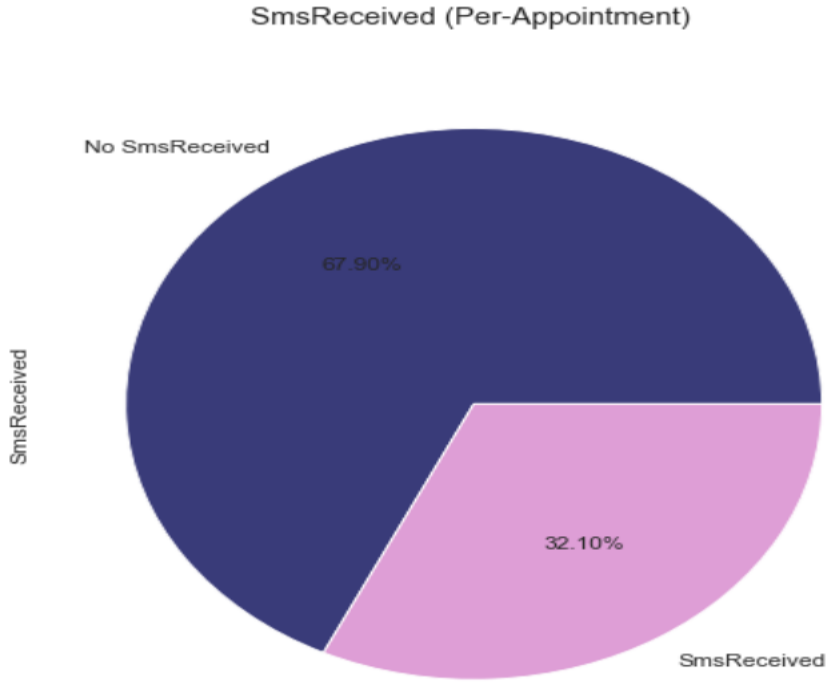
Handicap

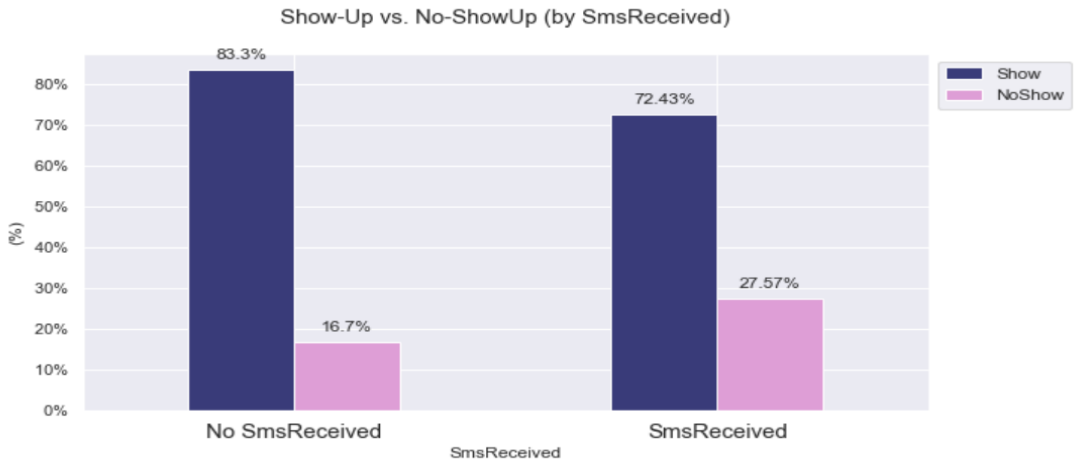




Looking at the above graphs, we found out that the proportion of appointments where patients who do not need special attention are largest while the patients who experience Handicap Level 1 (1.85% of the total population) have the highest show-up rate of 82.16% (higher than the average show-up rate), where patients who experience Handicap Level 4 (0.0027 % of the total population) have a show-up rate of 66.67% (less than the average show-up rate).

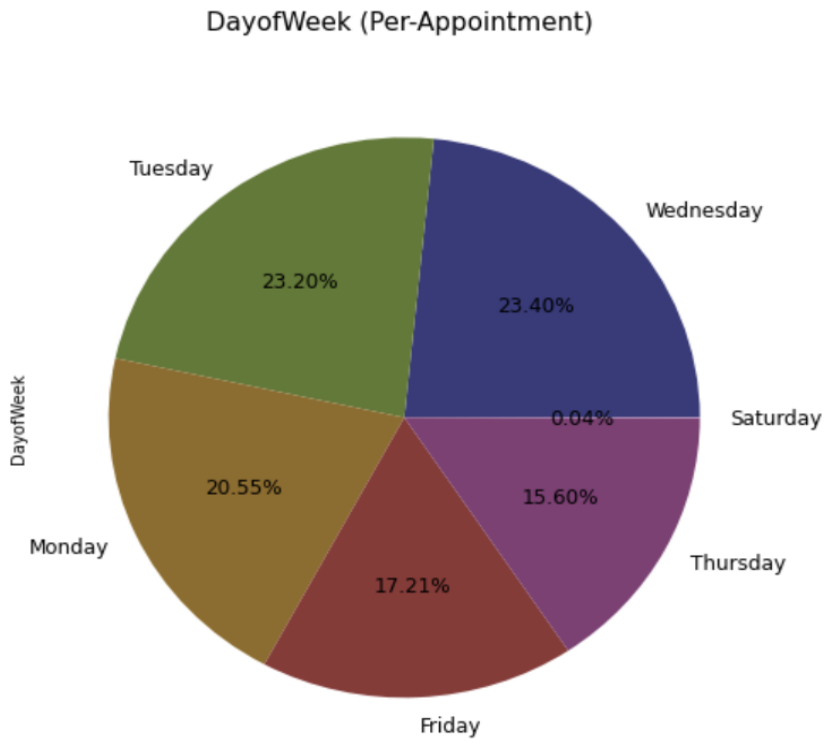
SMS-Received

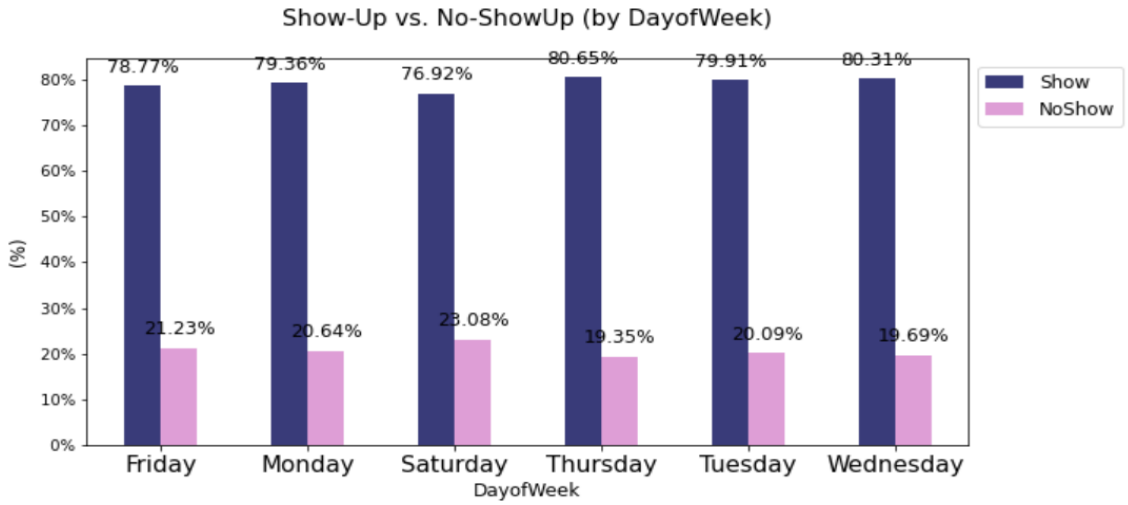




Looking at the above graphs, we found out that the proportion of appointments where patients did not receive SMS reminder is higher with 67.90% while the show-up rate of patients who do not receive a remainder is higher with 83.3% and for the patients who received a remainder is 72.43%.

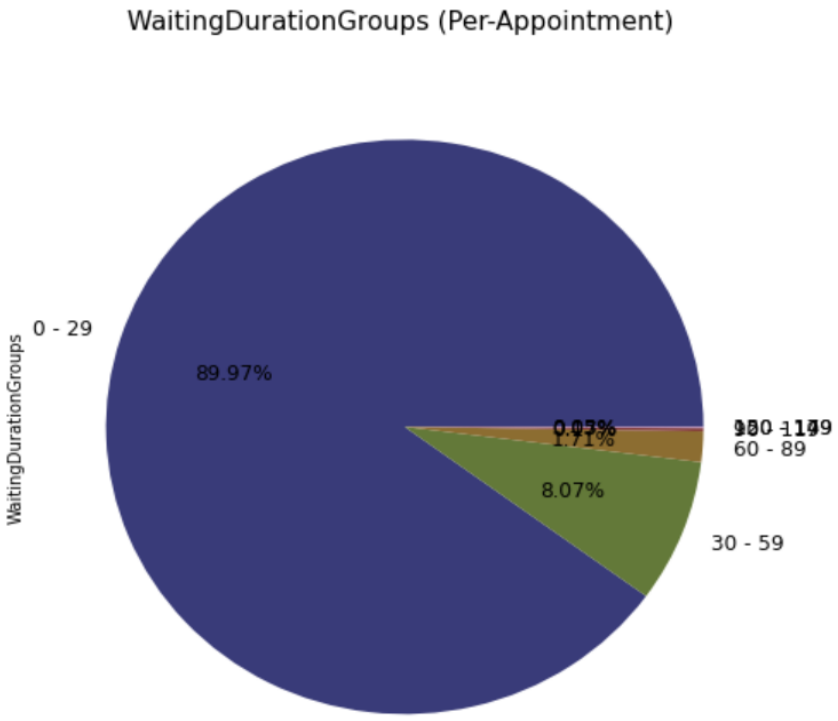
Days of Week

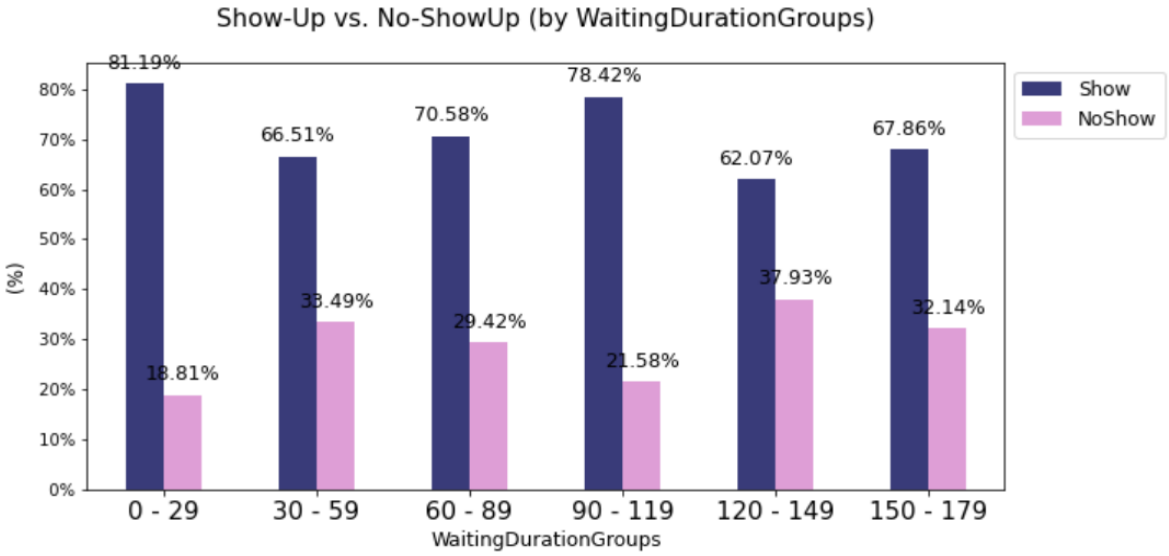




Now by looking at the above graphs, we found out that the proportion of appointments where patients when scheduled on a Wednesday is 23.40% which is the highest, where the lowest proportion of appointments is where patients scheduled on a Saturday, and that is 0.04% while the show-up rate of patients who have their appointments on Thursday (15.60% of the total population) have the highest show-up rate of 80.65% (higher than the average show-up rate), where patients who have their appointments on Saturday (0.04% of the total population) have the lowest show-up rate of 76.92% (less than the average show-up rate).

Waiting Duration



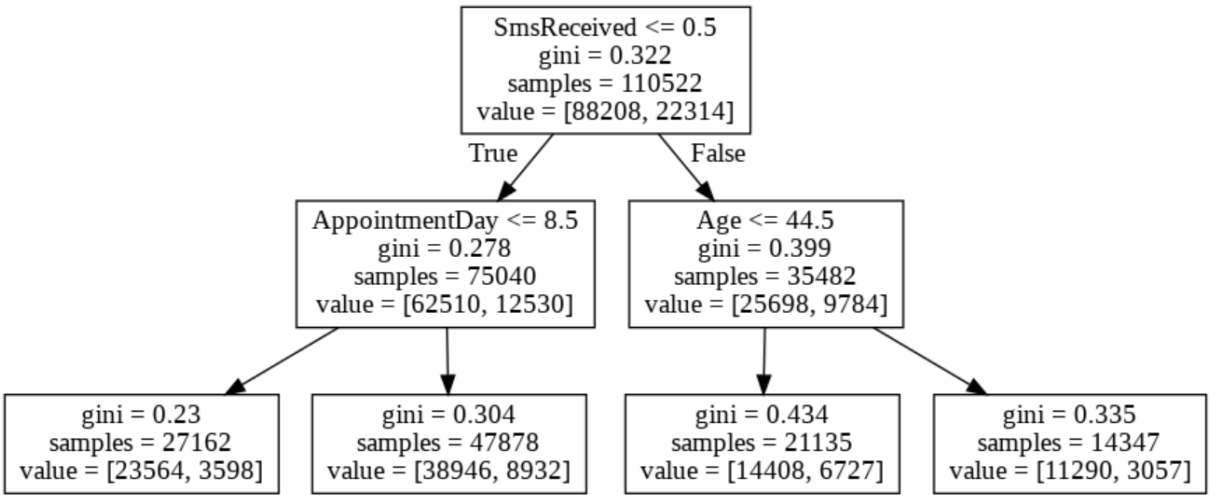


Looking at the above graphs, we found out that the proportion of appointments where patients have a waiting duration between (0 - 29) days is 89.97%, where the proportion of appointments where patients have a waiting duration between (120 - 149) days is 0.03% while the show-up rate of patients whose waiting duration falls in the category (0 - 29 days) (89.97% of the total population) have the highest show-up rate of 81.19% (higher than the average show-up rate), where patients whose waiting duration falls in the category (120 - 149 days) (0.03% of the total population) have a show-up rate of 62.07% (less than the average show-up rate).

Data Modeling

Decision Tree

We make a Decision Tree based on the columns. A decision tree is a diagram that helps us to better understand the course of action or show a statistical probability.



Each branch of the decision tree represents a possible decision, outcome, or reaction. The end branches on the tree represent the end results of the decision The decision tree classifies the data points into the same category, although our dataset isn't skewed. Hence moving onto the next model in our analysis.

Logistic Regression

Logistic Regression is one of the primary algorithms that employ a sigmoid function and considered to best for binary classification. It is the most frequently used machine learning algorithm. The Logistic Regression model was developed in the study as the benchmark model for comparison. Logistic regression assumes no error in the output variable (y), it considered removing outliers and possibly misclassified instances from training data. The coefficients in logistic regression using a process called maximum-likelihood estimation.

```
"-----Logistic_Regression-----"

Accuracy: 0.7984437675302349
------------------------------------------------------------
confusion matrix:
[[26367   207]
 [ 6476   107]]
------------------------------------------------------------
classification report:

              precision    recall  f1-score   support

           0       0.80      0.99      0.89     26574
           1       0.34      0.02      0.03      6583

    accuracy                           0.80     33157
   macro avg       0.57      0.50      0.46     33157
weighted avg       0.71      0.80      0.72     33157

------------------------------------------------------------
Tuned Model Parameters: {'penalty': 'none', 'tol': 0.0001}
```

F1 score is a function of precision and recall which is needed when we want to check the balance between precision and recall, accuracy can contribute to a large number of true negative which is most business scenario but we don't focus much on false negative and positive as this cost business a cost so we check the F1 score of our model to seek the balance between prescision and recall of big data sets. In our case it shows 89% for logistic regression.

Random Forest

Random forest is a machine learning algorithm that grows multiple decision trees during training phase. The random forest then chooses the decision of the majority of the trees as the final decision. It is a type of ensemble learning method, where a group of weak models combine to form a

powerful model. Random forest uses multiple trees which reduces the risk of overfitting. It is an ensemble learning method that predicts based on the majority vote and runs on large datasets. It can predict in the presence of missing data and maintain accuracy when a large proportion of data is missing.

```
"-----RandomForest_Classifier-----"

Accuracy: 0.7812829870012366
------------------------------------------------------------
confusion matrix:
[[24920  1654]
 [ 5598   985]]
------------------------------------------------------------
classification report:

              precision    recall  f1-score   support

           0       0.82      0.94      0.87     26574
           1       0.37      0.15      0.21      6583

    accuracy                           0.78     33157
   macro avg       0.59      0.54      0.54     33157
weighted avg       0.73      0.78      0.74     33157

------------------------------------------------------------
Tuned Model Parameters: {'min_samples_split': 9, 'n_estimators': 100}
```

F1 score for random forest is 87%

K-Nearest Neighbor

K-Nearest Neighbor is a density estimation method which can be used for classification purposes through the use of Bayes' theorem. It is a simple algorithm that can stores all the available data and classifies the new data or case based on a similarity measure. It is mostly used to classify data points based on how their neighbors are classified. This means that k-nearest neighbor falls in the category of supervised learning. It has parameters that refers to the number of nearest neighbors to be included in the process at the testing stage and consider in attaching a label to the new data point. Bayes' theorem required modelling the class conditional densities for each class separately and then joining them with their corresponding priors to come up with new models for the posterior probabilities. These posterior probabilities then used to make classification decisions.

```
"-----KNeighbors_Classifier-----"

Accuracy: 0.7885212775582833
------------------------------------------------------------
confusion matrix:
[[25531  1043]
 [ 5969   614]]
------------------------------------------------------------
classification report:

              precision    recall  f1-score   support

           0       0.81      0.96      0.88     26574
           1       0.37      0.09      0.15      6583

    accuracy                           0.79     33157
   macro avg       0.59      0.53      0.51     33157
weighted avg       0.72      0.79      0.73     33157

------------------------------------------------------------
Tuned Model Parameters: {'n_neighbors': 6}
```

F1 score for K-nearest Neighbor is 88%.

XgBoost

One of the most popular machine learning algorithm now a day's is XGBoost, whether you want to perform regression or classification, it is well known to given best result solution than any other machine learning algorithm. Xgboost is faster than other classifiers because it can harness the power of multi core computers and GPU's across networks making it feasible to train big datasets. Xgboost has parameters for regularization, missing values, cross validation, scikit learn etc. It belongs to boosting algorithm family and uses gradient boosting framework, which is a sequential technique that combines set of week learners and predict accuracy out of it, the model outcomes are based on weights of previous model in order to give accuracy it lower the weight for predicted value and weighted higher for classified value.

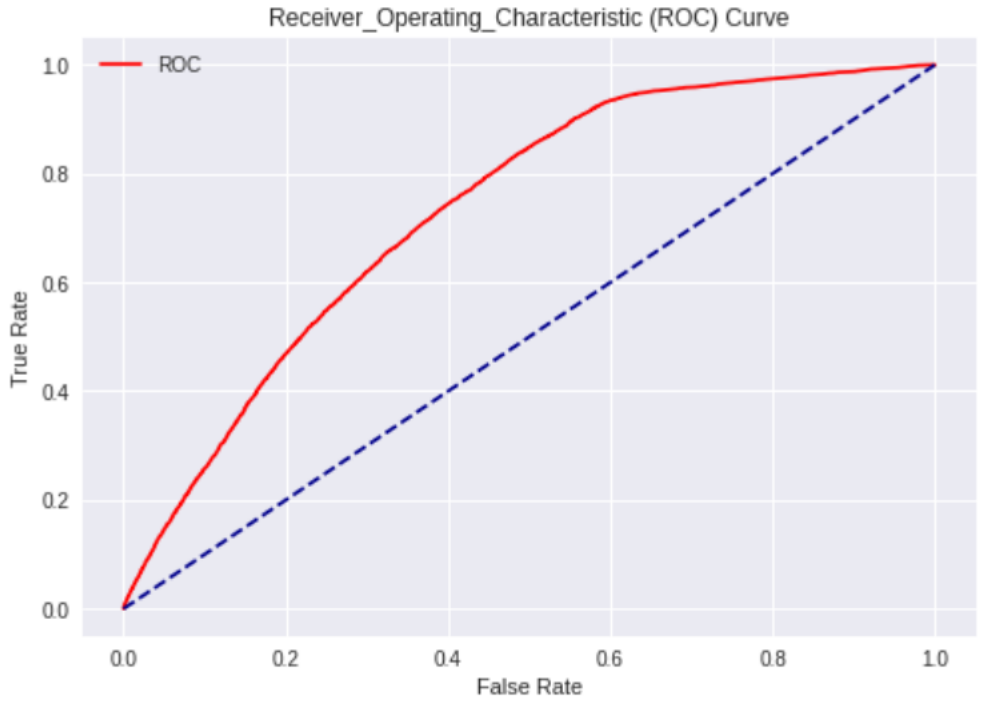
```
Accuracy -0.79639372 params {'max_depth': 7, 'gamma': '0.036', 'reg_alpha': '0.310', 'learning_rate': '0.089', 'colsample_bytree': '0.993'}
Accuracy -0.79692367 params {'max_depth': 6, 'gamma': '0.148', 'reg_alpha': '0.182', 'learning_rate': '0.188', 'colsample_bytree': '0.411'}
Accuracy -0.79676856 params {'max_depth': 3, 'gamma': '0.319', 'reg_alpha': '0.184', 'learning_rate': '0.195', 'colsample_bytree': '0.734'}
Accuracy -0.79666516 params {'max_depth': 6, 'gamma': '0.477', 'reg_alpha': '0.241', 'learning_rate': '0.058', 'colsample_bytree': '0.394'}
Accuracy -0.79666516 params {'max_depth': 2, 'gamma': '0.146', 'reg_alpha': '0.037', 'learning_rate': '0.070', 'colsample_bytree': '0.419'}
Accuracy -0.79613520 params {'max_depth': 7, 'gamma': '0.126', 'reg_alpha': '0.355', 'learning_rate': '0.095', 'colsample_bytree': '0.966'}
Accuracy -0.79672979 params {'max_depth': 3, 'gamma': '0.240', 'reg_alpha': '0.178', 'learning_rate': '0.125', 'colsample_bytree': '0.992'}
Accuracy -0.79692367 params {'max_depth': 6, 'gamma': '0.127', 'reg_alpha': '0.011', 'learning_rate': '0.113', 'colsample_bytree': '0.656'}
Accuracy -0.79538551 params {'max_depth': 8, 'gamma': '0.317', 'reg_alpha': '0.025', 'learning_rate': '0.173', 'colsample_bytree': '0.725'}
Accuracy -0.79685904 params {'max_depth': 5, 'gamma': '0.413', 'reg_alpha': '0.038', 'learning_rate': '0.103', 'colsample_bytree': '0.530'}
Accuracy -0.79676856 params {'max_depth': 5, 'gamma': '0.312', 'reg_alpha': '0.022', 'learning_rate': '0.075', 'colsample_bytree': '0.880'}
Accuracy -0.79672979 params {'max_depth': 5, 'gamma': '0.460', 'reg_alpha': '0.083', 'learning_rate': '0.094', 'colsample_bytree': '0.412'}
Accuracy -0.79610935 params {'max_depth': 6, 'gamma': '0.134', 'reg_alpha': '0.200', 'learning_rate': '0.138', 'colsample_bytree': '0.861'}
Accuracy -0.79680734 params {'max_depth': 5, 'gamma': '0.347', 'reg_alpha': '0.159', 'learning_rate': '0.138', 'colsample_bytree': '0.490'}
Accuracy -0.79634202 params {'max_depth': 6, 'gamma': '0.181', 'reg_alpha': '0.285', 'learning_rate': '0.181', 'colsample_bytree': '0.604'}
Accuracy -0.79674271 params {'max_depth': 6, 'gamma': '0.232', 'reg_alpha': '0.319', 'learning_rate': '0.109', 'colsample_bytree': '0.873'}
Accuracy -0.79665223 params {'max_depth': 6, 'gamma': '0.109', 'reg_alpha': '0.025', 'learning_rate': '0.182', 'colsample_bytree': '0.724'}
Accuracy -0.79679442 params {'max_depth': 3, 'gamma': '0.124', 'reg_alpha': '0.388', 'learning_rate': '0.089', 'colsample_bytree': '0.993'}
Accuracy -0.79700123 params {'max_depth': 5, 'gamma': '0.071', 'reg_alpha': '0.044', 'learning_rate': '0.101', 'colsample_bytree': '0.834'}
Accuracy -0.79657468 params {'max_depth': 4, 'gamma': '0.004', 'reg_alpha': '0.134', 'learning_rate': '0.117', 'colsample_bytree': '0.930'}
Accuracy -0.79682027 params {'max_depth': 4, 'gamma': '0.050', 'reg_alpha': '0.104', 'learning_rate': '0.157', 'colsample_bytree': '0.302'}
Accuracy -0.79543721 params {'max_depth': 7, 'gamma': '0.044', 'reg_alpha': '0.235', 'learning_rate': '0.155', 'colsample_bytree': '0.785'}
Accuracy -0.79674271 params {'max_depth': 4, 'gamma': '0.192', 'reg_alpha': '0.079', 'learning_rate': '0.130', 'colsample_bytree': '0.307'}
Accuracy -0.79574743 params {'max_depth': 7, 'gamma': '0.084', 'reg_alpha': '0.129', 'learning_rate': '0.200', 'colsample_bytree': '0.624'}
Accuracy -0.79670394 params {'max_depth': 4, 'gamma': '0.002', 'reg_alpha': '0.235', 'learning_rate': '0.058', 'colsample_bytree': '0.539'}
Accuracy -0.79497189 params {'max_depth': 8, 'gamma': '0.182', 'reg_alpha': '0.067', 'learning_rate': '0.158', 'colsample_bytree': '0.815'}
Accuracy -0.79666516 params {'max_depth': 5, 'gamma': '0.080', 'reg_alpha': '0.271', 'learning_rate': '0.078', 'colsample_bytree': '0.350'}
Accuracy -0.79529503 params {'max_depth': 8, 'gamma': '0.215', 'reg_alpha': '0.108', 'learning_rate': '0.149', 'colsample_bytree': '0.678'}
Accuracy -0.79579913 params {'max_depth': 7, 'gamma': '0.277', 'reg_alpha': '0.158', 'learning_rate': '0.169', 'colsample_bytree': '0.573'}
Accuracy -0.79688490 params {'max_depth': 7, 'gamma': '0.017', 'reg_alpha': '0.059', 'learning_rate': '0.051', 'colsample_bytree': '0.466'}
Accuracy -0.79665223 params {'max_depth': 2, 'gamma': '0.080', 'reg_alpha': '0.328', 'learning_rate': '0.085', 'colsample_bytree': '0.783'}
Accuracy -0.79680734 params {'max_depth': 4, 'gamma': '0.278', 'reg_alpha': '0.204', 'learning_rate': '0.102', 'colsample_bytree': '0.907'}
Accuracy -0.79680734 params {'max_depth': 6, 'gamma': '0.162', 'reg_alpha': '0.260', 'learning_rate': '0.119', 'colsample_bytree': '0.829'}
Accuracy -0.79663931 params {'max_depth': 5, 'gamma': '0.062', 'reg_alpha': '0.354', 'learning_rate': '0.192', 'colsample_bytree': '0.335'}

Accuracy -0.79665223 params {'max_depth': 2, 'gamma': '0.080', 'reg_alpha': '0.328', 'learning_rate': '0.085', 'colsample_bytree': '0.783'}
Accuracy -0.79680734 params {'max_depth': 4, 'gamma': '0.278', 'reg_alpha': '0.204', 'learning_rate': '0.102', 'colsample_bytree': '0.907'}
Accuracy -0.79680734 params {'max_depth': 6, 'gamma': '0.162', 'reg_alpha': '0.260', 'learning_rate': '0.119', 'colsample_bytree': '0.829'}
Accuracy -0.79663931 params {'max_depth': 5, 'gamma': '0.062', 'reg_alpha': '0.354', 'learning_rate': '0.192', 'colsample_bytree': '0.335'}
Accuracy -0.79678149 params {'max_depth': 3, 'gamma': '0.370', 'reg_alpha': '0.172', 'learning_rate': '0.133', 'colsample_bytree': '0.693'}
Accuracy -0.79665223 params {'max_depth': 6, 'gamma': '0.100', 'reg_alpha': '0.294', 'learning_rate': '0.066', 'colsample_bytree': '0.370'}
Accuracy -0.79679442 params {'max_depth': 3, 'gamma': '0.266', 'reg_alpha': '0.126', 'learning_rate': '0.191', 'colsample_bytree': '0.461'}
Accuracy -0.79568280 params {'max_depth': 7, 'gamma': '0.209', 'reg_alpha': '0.097', 'learning_rate': '0.143', 'colsample_bytree': '0.934'}
Accuracy -0.79663931 params {'max_depth': 2, 'gamma': '0.155', 'reg_alpha': '0.229', 'learning_rate': '0.165', 'colsample_bytree': '0.781'}
Accuracy -0.79687197 params {'max_depth': 5, 'gamma': '0.040', 'reg_alpha': '0.040', 'learning_rate': '0.102', 'colsample_bytree': '0.630'}
Accuracy -0.79713049 params {'max_depth': 6, 'gamma': '0.022', 'reg_alpha': '0.396', 'learning_rate': '0.071', 'colsample_bytree': '0.585'}
Accuracy -0.79672979 params {'max_depth': 4, 'gamma': '0.029', 'reg_alpha': '0.398', 'learning_rate': '0.065', 'colsample_bytree': '0.565'}
Accuracy -0.79683319 params {'max_depth': 8, 'gamma': '0.420', 'reg_alpha': '0.369', 'learning_rate': '0.051', 'colsample_bytree': '0.729'}
Accuracy -0.79688490 params {'max_depth': 5, 'gamma': '0.494', 'reg_alpha': '0.334', 'learning_rate': '0.080', 'colsample_bytree': '0.501'}
Accuracy -0.79702708 params {'max_depth': 6, 'gamma': '0.003', 'reg_alpha': '0.306', 'learning_rate': '0.072', 'colsample_bytree': '0.669'}
Accuracy -0.79710463 params {'max_depth': 6, 'gamma': '0.017', 'reg_alpha': '0.376', 'learning_rate': '0.072', 'colsample_bytree': '0.591'}
Accuracy -0.79666516 params {'max_depth': 7, 'gamma': '0.018', 'reg_alpha': '0.374', 'learning_rate': '0.060', 'colsample_bytree': '0.440'}
Accuracy -0.79654883 params {'max_depth': 8, 'gamma': '0.107', 'reg_alpha': '0.348', 'learning_rate': '0.093', 'colsample_bytree': '0.583'}
Accuracy -0.79715634 params {'max_depth': 6, 'gamma': '0.390', 'reg_alpha': '0.371', 'learning_rate': '0.108', 'colsample_bytree': '0.513'}
Accuracy -0.79658760 params {'max_depth': 7, 'gamma': '0.449', 'reg_alpha': '0.396', 'learning_rate': '0.111', 'colsample_bytree': '0.391'}
100%|██████████| 50/50 [13:11<00:00, 15.84s/it, best loss: -0.7971563368448265]
CPU times: user 25min 49s, sys: 3.57 s, total: 25min 52s
Wall time: 13min 12s
```

```
CPU times: user 39.9 s, sys: 72.6 ms, total: 40 s
Wall time: 20.4 s
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.5130650431588437,
              gamma=0.3900504221927368, learning_rate=0.1080254516610683,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=None,
              n_estimators=600, n_jobs=-1, nthread=None,
              objective='binary:logistic', random_state=0,
              reg_alpha=0.37086848638891284, reg_lambda=0.9875697933695895,
              scale_pos_weight=1, seed=None, silent=None, subsample=1,
              verbosity=1)
```

```
print('Accuracy_Score','----------', accuracy_score)
print('Roc_Auc_Score','----------', roc_auc_score)
```

```
Accuracy_Score ---------- 0.801
Roc_Auc_Score ---------- 0.732
```



Apart for every score we get above, accuracy score of Xgboost is 80.1%, which best for model with roc score of 73.2%. Receiver operating characteristic curve or Roc curve describe the trade-off between true positive and true negative rate for predicted model using different probabilities.

We picked Xgboost for best model fitting as its precision rate is 80.1% which better among above models. It can predict easily with unseen data and give better result for patients to show up or not, which is help for medical to predict that a person is coming or not then they can send SMS reminder to such patients for their appointments that can save time for medical staff as well as other patients.

Conclusion

For this business problem we analyze the every given possible feature that become the reason for patients in missing their appointments. The overall show up rate is 79.81% as per given data, we also observe that the show up rate of male patients is slightly higher then female patients, for an age group the show up for group (60-79) is much higher among others and many other factors are deeply discussing in this analysis. Most of the calculation perform in this analysis are based on

Appointment-Id and not on Patient-Id because of duplicity, since the time dimension for appointment time were set to 00:00 were not able to address that. Given Data is not consistent so we eliminate six entries from original data set. For the visualization part as most of the columns are categorical data, the visualization charts were mainly bar charts and pie charts in this analysis, Histograms and scatter plots were excluded from this analysis because of the above reason.

For predictive model we select Xgboost for best training the model and predicted the higher accuracy score above all, if our model is used for unseen data it can predict the best result for patients who miss their appointment. As medical staff knows that who can miss their appointment or not, they can take action a day before by sending a remainder text to that particular patient, this save a lot of cost and energy to healthcare sector. Another factor of overbooking can be solved with help of this classifier as it predicts what factors cause the overbooking issue, which can be resolve. Finally, one thing under consideration is that after creating as many as predictive models, we can't predict 100% because at the end of the day we, humans are unpredictable and lots of factors may cause a patient for not showing up for example an emergency issue but we try our best to give a general solution that can save cost and help the healthcare system to become better.

References

https://www.kaggle.com/joniarroba/noshowappointments

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00384-9

https://marcellovictorino.github.io/project/Project-1-Appointments-No-Show/#

https://www.datascience-pm.com/crisp-dm-2/

https://www.datacamp.com/community/tutorials/xgboost-in-python

https://medium.com/healthfdn-data-analytics/predicting-missed-hospital-appointments-using-machine-learning-what-are-the-risks-a388348109d

https://blog.dataiku.com/health-care-predictive-analytics

https://towardsdatascience.com/using-randomforest-to-predict-medical-appointment-no-shows-b33575e3ff42

https://assets.researchsquare.com/files/rs-33216/v2/01fa20bd-b358-4da7-9370-0eb2f4b51ce0.pdf?c=1600373399