CSCI322: Data Analysis (Spring 2021)

# Investigate a Dataset: Analysis and Classification

Supervised by: Dr Mustafa El-Attar & Eng. Aly Mohamed
Date: 2/7/2021

Team Members:
1) Mahmoud Wessam
2) Mazen Mobtasem
3) Salma Hossam Zakzouk

# Agenda

1) Project Description

2) Dataset Used

3) Cleaning stage: Assessing data & data wrangling

4) Analyzing & Visualizing stage

- ➢General exploration of dataset
- ➢Exploratory data analysis

5)Creating ML models for data prediction

- ➢Logistic Regression Model
- ➢KNN Model
- ➢Decision Tree Model
- ➢Random Forest Classifier Model

# Project Description

We are going to clean, analyze, & visualize data from a dataset (No-show appointments) and get insights about the data. The target of the project is to know what factors are important for us to know in order to predict if a patient will show up for their scheduled appointment? This question will be answered after finishing the 3 above steps using python code in Jupyter Notebook: Cleaning, analyzing & visualizing. Then we are going to develop ML models for data prediction such as:

- Logistic Regression Model
- KNN Model
- Decision Tree Model
- Random Forest Classifier Model

# Dataset used

This dataset collects information from 100k medical appointments in Brazil and is focused on the question of whether or not patients show up for their appointment. A number of characteristics about the patient are included in each row.

● ‘ScheduledDay’ tells us on what day the patient set up their appointment.

● ‘Neighborhood’ indicates the location of the hospital.

● ‘Scholarship’ indicates whether or not the patient is enrolled in Brasilian welfare program Bolsa Família.

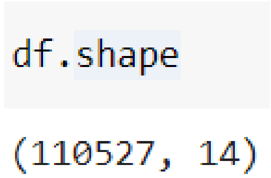
● Be careful about the encoding of the last column: it says ‘No’ if the patient showed up to their appointment, and ‘Yes’ if they did not show up.

The link of the dataset:

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

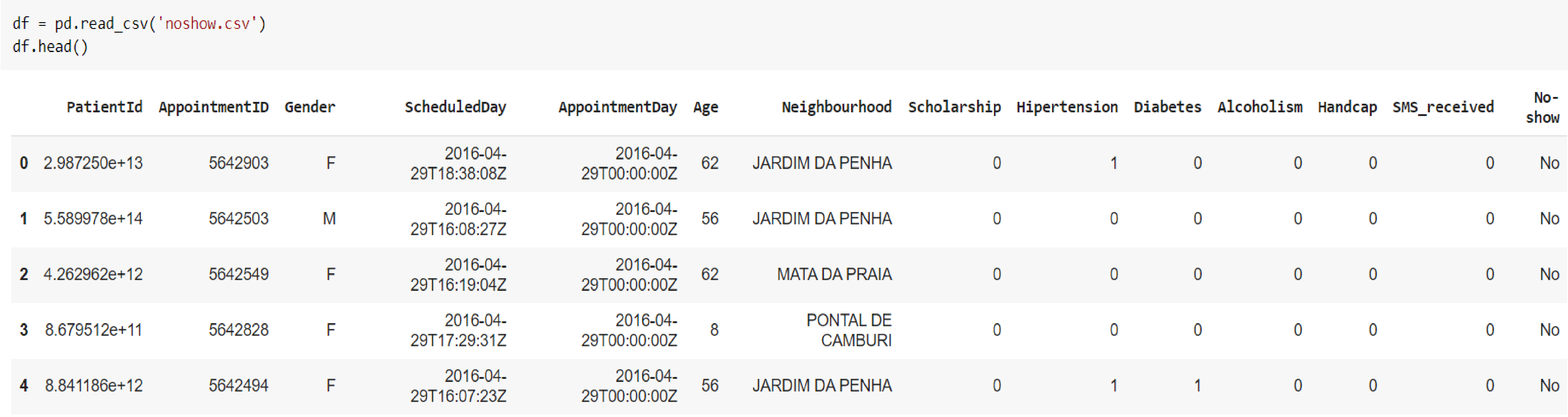
# Cleaning stage: Assessing data & data wrangling

First, we will have a view of the data in the dataset we are working on, to explore the columns and the size of the dataset which is (110527, 14), that is the data contains 14 columns and 110527 record.



```
df.shape
```

```
(110527, 14)
```

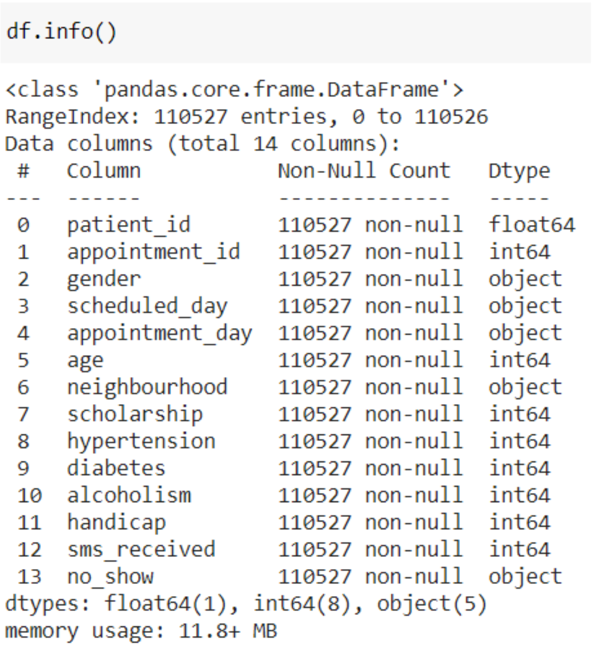


```
df = pd.read_csv('noshow.csv')
df.head()
```

| | PatientId | AppointmentID | Gender | ScheduledDay | AppointmentDay | Age | Neighbourhood | Scholarship | 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 | JARDIM DA PENHA | 0 | 1 | 0 | 0 | 0 | 0 | No |
| 1 | 5.589978e+14 | 5642503 | M | 2016-04-29T16:08:27Z | 2016-04-29T00:00:00Z | 56 | JARDIM DA PENHA | 0 | 0 | 0 | 0 | 0 | 0 | No |
| 2 | 4.262962e+12 | 5642549 | F | 2016-04-29T16:19:04Z | 2016-04-29T00:00:00Z | 62 | MATA DA PRAIA | 0 | 0 | 0 | 0 | 0 | 0 | No |
| 3 | 8.679512e+11 | 5642828 | F | 2016-04-29T17:29:31Z | 2016-04-29T00:00:00Z | 8 | PONTAL DE CAMBURI | 0 | 0 | 0 | 0 | 0 | 0 | No |
| 4 | 8.841186e+12 | 5642494 | F | 2016-04-29T16:07:23Z | 2016-04-29T00:00:00Z | 56 | JARDIM DA PENHA | 0 | 1 | 1 | 0 | 0 | 0 | No |

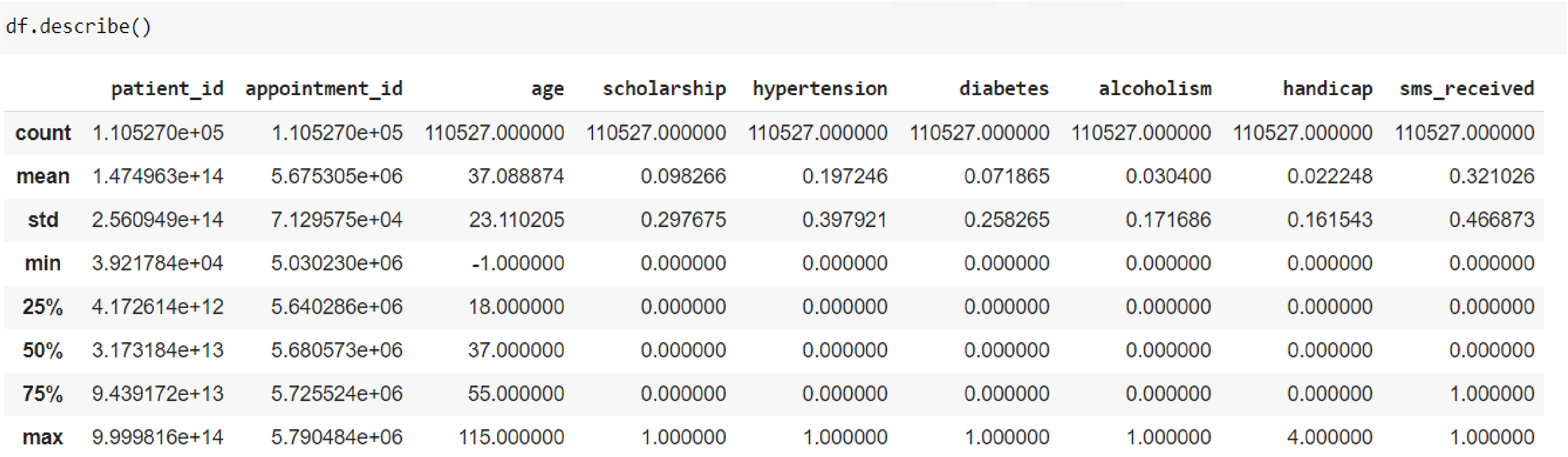
# Cleaning stage: Assessing data & data wrangling

Afterwards, we have looked at the datatypes of the columns and general numerical description about the data.



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   patient_id      110527 non-null  float64
 1   appointment_id  110527 non-null  int64  
 2   gender          110527 non-null  object 
 3   scheduled_day   110527 non-null  object 
 4   appointment_day 110527 non-null  object 
 5   age             110527 non-null  int64  
 6   neighbourhood   110527 non-null  object 
 7   scholarship     110527 non-null  int64  
 8   hypertension    110527 non-null  int64  
 9   diabetes        110527 non-null  int64  
 10  alcoholism      110527 non-null  int64  
 11  handicap        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
```



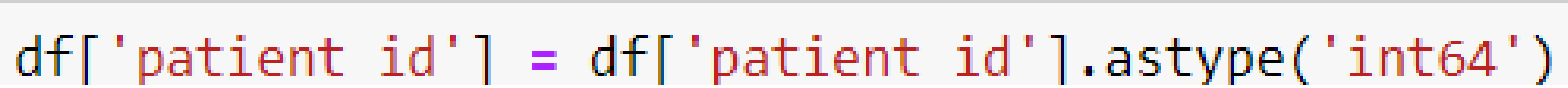
```
df.describe()
```

|  | patient_id | appointment_id | age | scholarship | hypertension | diabetes | alcoholism | handicap | sms_received |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 1.105270e+05 | 1.105270e+05 | 110527.000000 | 110527.000000 | 110527.000000 | 110527.000000 | 110527.000000 | 110527.000000 | 110527.000000 |
| **mean** | 1.474963e+14 | 5.675305e+06 | 37.088874 | 0.098266 | 0.197246 | 0.071865 | 0.030400 | 0.022248 | 0.321026 |
| **std** | 2.560949e+14 | 7.129575e+04 | 23.110205 | 0.297675 | 0.397921 | 0.258265 | 0.171686 | 0.161543 | 0.466873 |
| **min** | 3.921784e+04 | 5.030230e+06 | -1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 4.172614e+12 | 5.640286e+06 | 18.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 3.173184e+13 | 5.680573e+06 | 37.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **75%** | 9.439172e+13 | 5.725524e+06 | 55.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| **max** | 9.999816e+14 | 5.790484e+06 | 115.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 4.000000 | 1.000000 |

# Cleaning stage: Assessing data & data wrangling

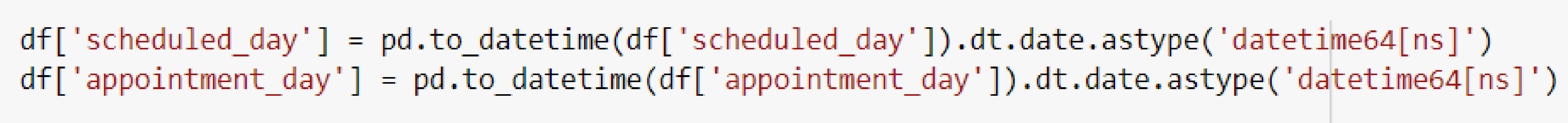
We have noticed that some cleaning has to be done in the cleaning process that is:

1) The patient_id data type is float as shown before, so it must be changed to be int



```
df['patient_id'] = df['patient_id'].astype('int64')
```

2) The scheduled_day and appointment_day columns type should be changed to datetime

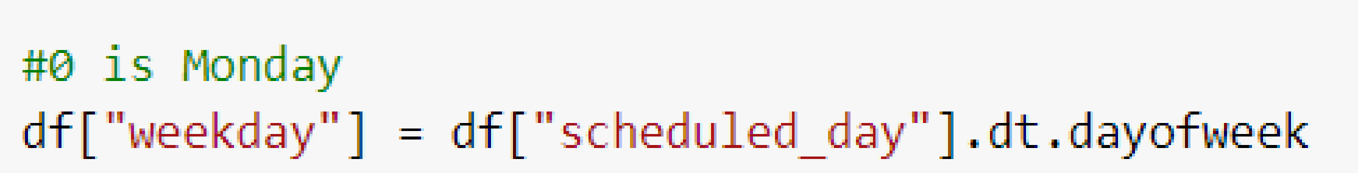


```
df['scheduled_day'] = pd.to_datetime(df['scheduled_day']).dt.date.astype('datetime64[ns]')
df['appointment_day'] = pd.to_datetime(df['appointment_day']).dt.date.astype('datetime64[ns]')
```

# Cleaning stage: Assessing data & data wrangling

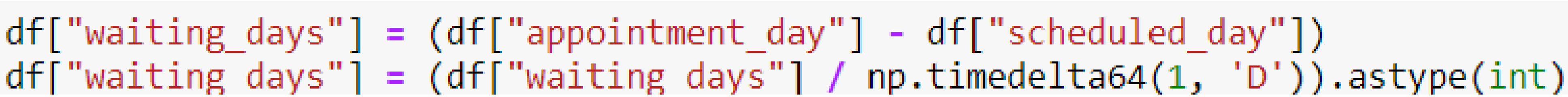
We have noticed that some cleaning has to be done in the cleaning process that is:

3) Adding Weekday column where 0 represents Monday



```
#0 is Monday
df["weekday"] = df["scheduled_day"].dt.dayofweek
```

4) Adding waiting_days column which is the difference between appointment and scheduled day and changing waiting_days type to int for further analysis



```
df["waiting_days"] = (df["appointment_day"] - df["scheduled_day"])
df["waiting_days"] = (df["waiting_days"] / np.timedelta64(1, 'D')).astype(int)
```

# Analyzing & Visualizing stage: General exploration of dataset

To get to know the dataset better, we will explore the dataset in general. That is, we will ask some questions for the analyzing and visualizing stage. Finally, we will get insights of the data and will be more able to answer the main question we are doing the analysis for which is: What factors are important for us to know in order to predict if a patient will show up for their scheduled appointment?

# General exploration: Asking questions for analyzing & visualizing stage

1) What is the ratio between males and females?

2) What is the ratio between show and no-show?

3) What is the most frequent scheduled date?

4) What is the most frequent appointment date?

5) What is the relation between scheduled date and appointment date?

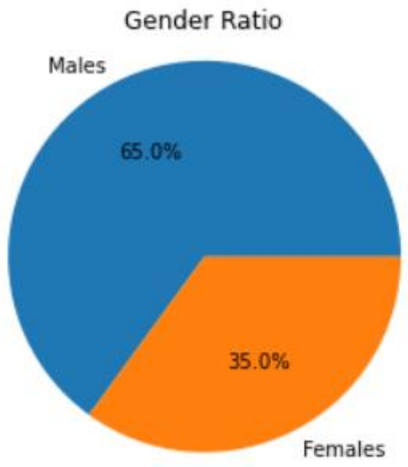
6) How many patients for each age?

7) What is the count of patients in each neighbourhood?

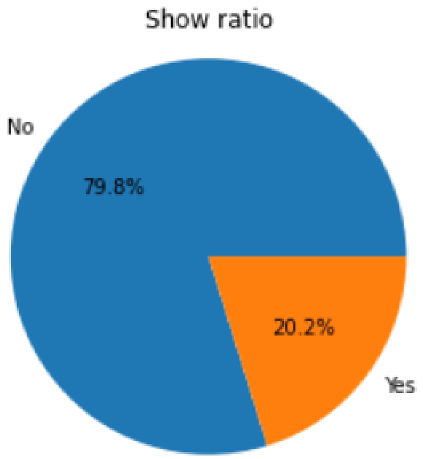
8) What is the number of patients by number of awaiting days?

# General exploration: Analyzing & Visualizing data

1) What is the ratio between males and females?

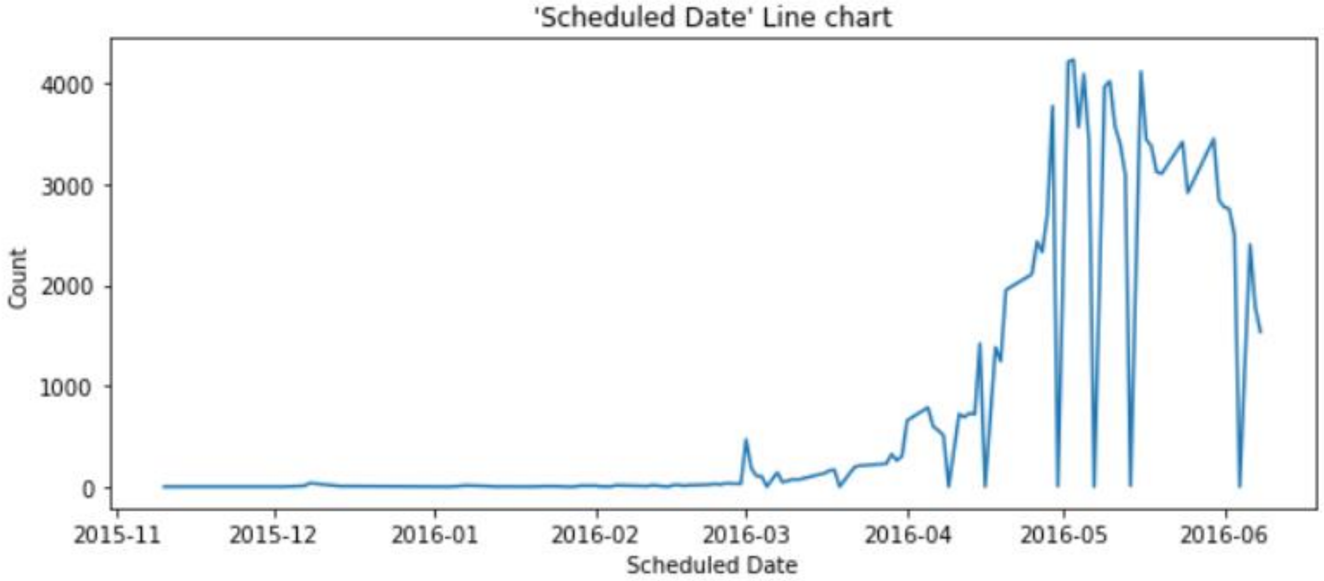


2) What is the ratio between show and no-show?



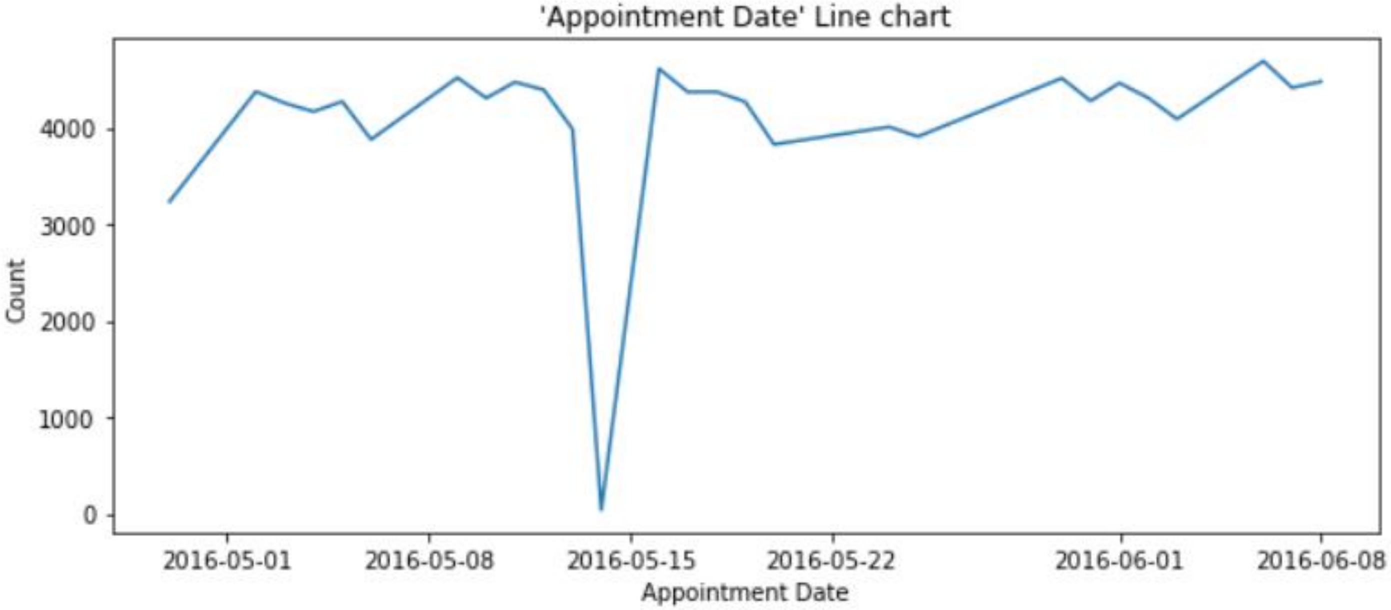
# General exploration: Analyzing & Visualizing data

3) What is the most frequent scheduled day?



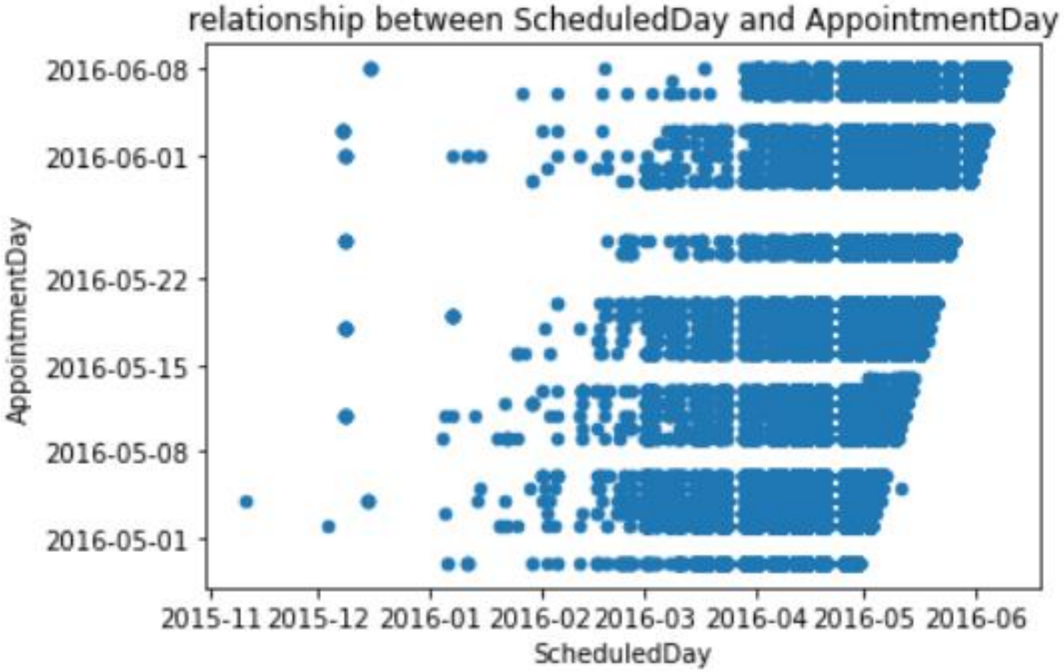
# General exploration: Analyzing & Visualizing data

4) What is the most frequent appointment day?



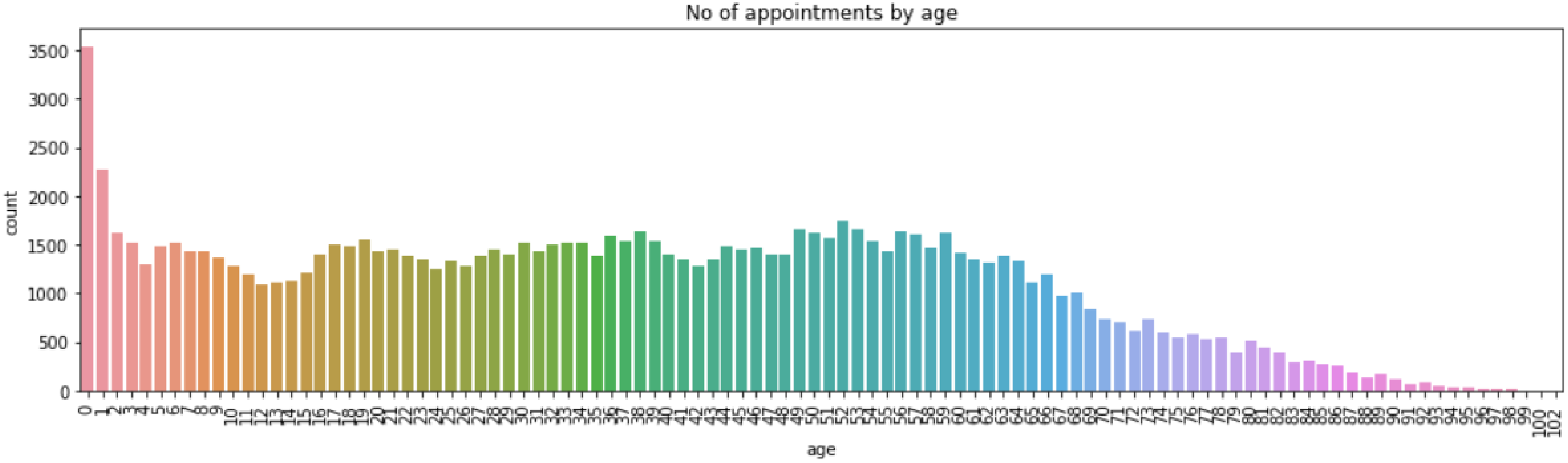
# General exploration: Analyzing & Visualizing data

5) What is the relation between scheduled day and appointment day?



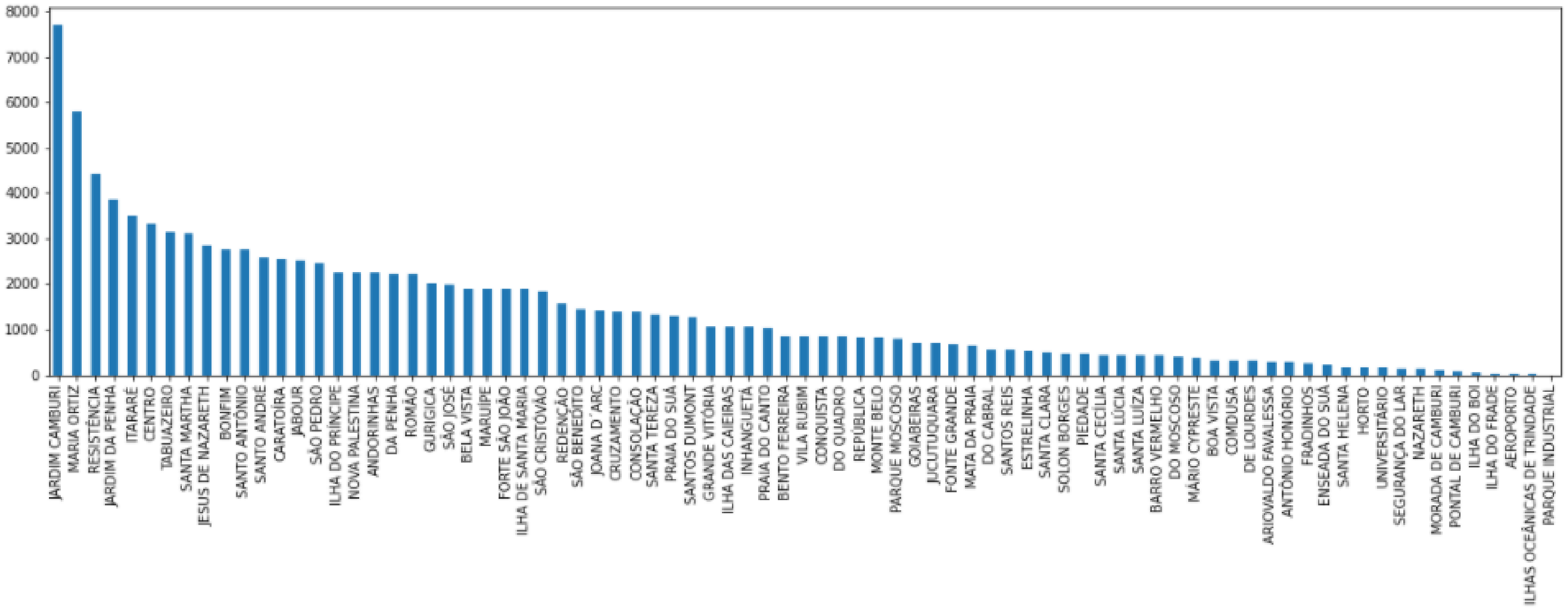
# General exploration: Analyzing & Visualizing data

6) How many patients for each age?



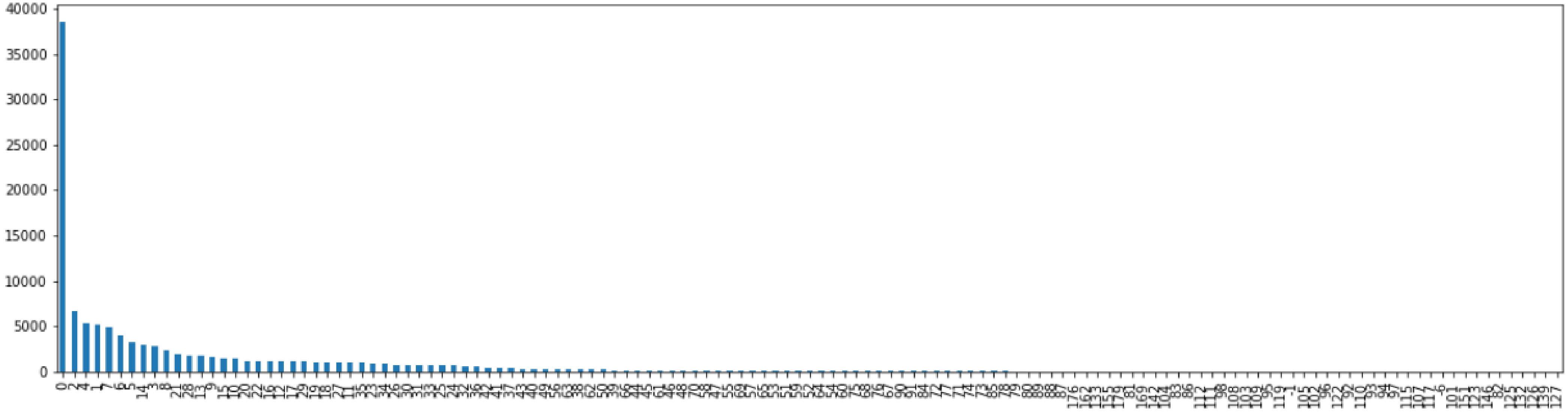
# General exploration: Analyzing & Visualizing data

7) What is the count of patients in each neighbourhood?



# General exploration: Analyzing & Visualizing data

8) What is the number of patients by number of awaiting days?



# General exploration: Insights of data

1) Males represent 65% of the dataset and females represent 35% only.

2) Only 20.2% of patients showed, while 79.8% didn’t show.

3) The most scheduled date is mainly between months 5 and 6 of 2016.

4) The appointment dates mainly have the same number of patients yet, there is a huge drop near 15/5/2016.

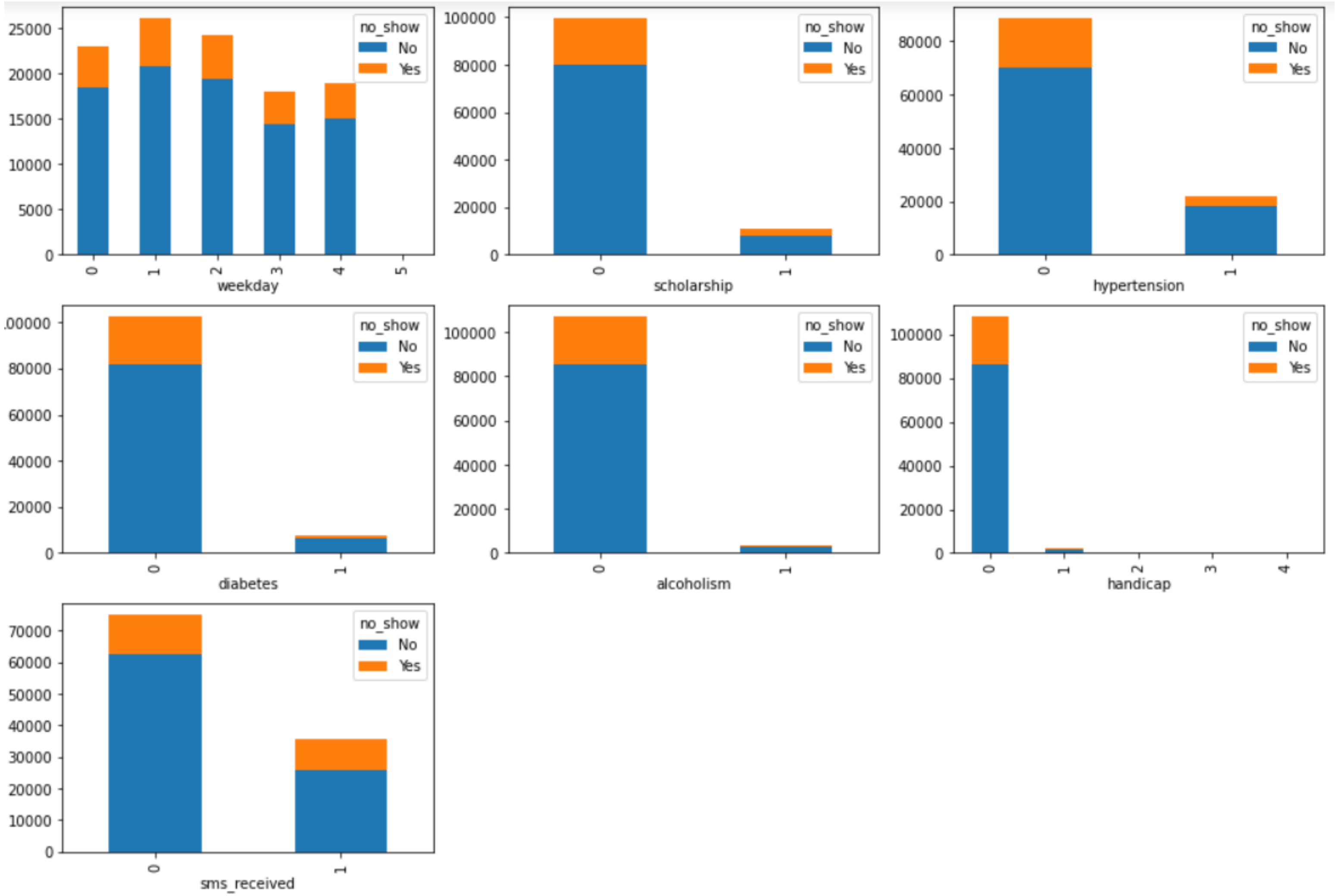
5) The majority of patients are of really young age.

6) Most of the patients are from Jardim Camburi.

7) The highest number of patients is when the awaiting days is 0.

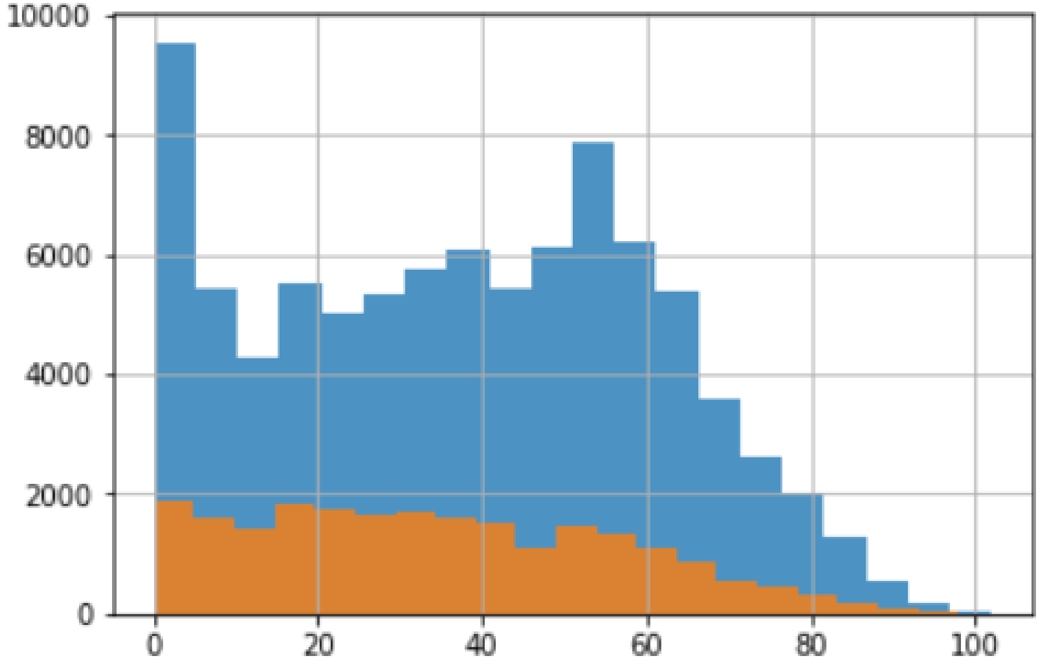
# General exploration: The main question

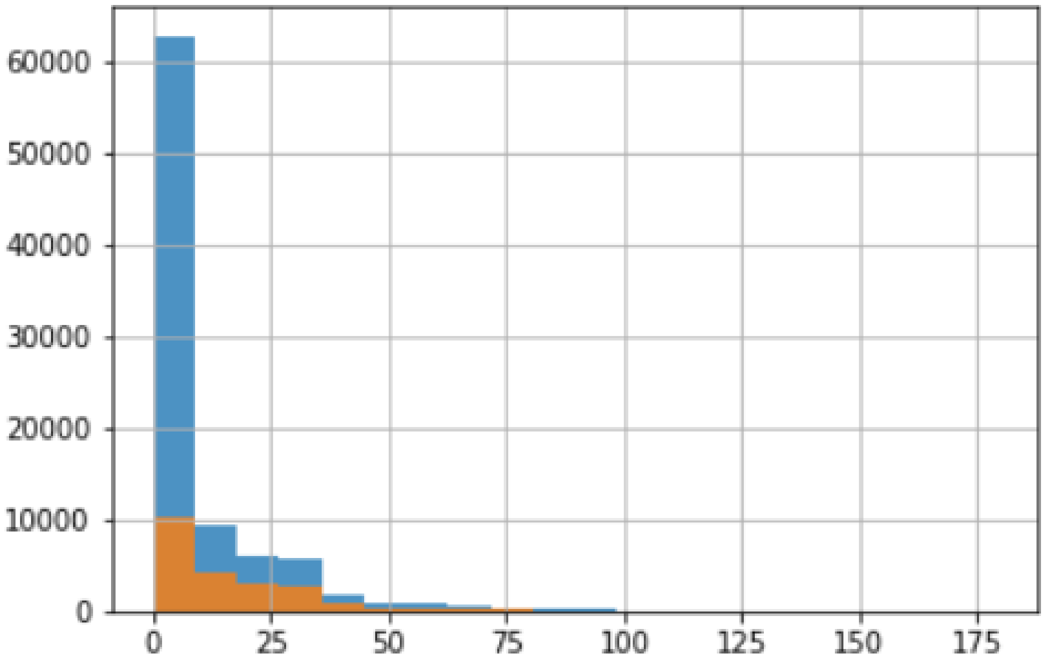
The main goal of this analysis is to find why patients miss their appointment, what factors are important for us to know in order to predict if a patient will show up for their scheduled appointment? The question will be answered by investigating categorical & numerical variables to find if there is any correlation between them and no-show.



# Exploratory data analysis :Analyzing & Visualizing stage (Categorical variables)

# Exploratory data analysis :Analyzing & Visualizing stage (Numerical variables)





# Exploratory data analysis : Insights of data

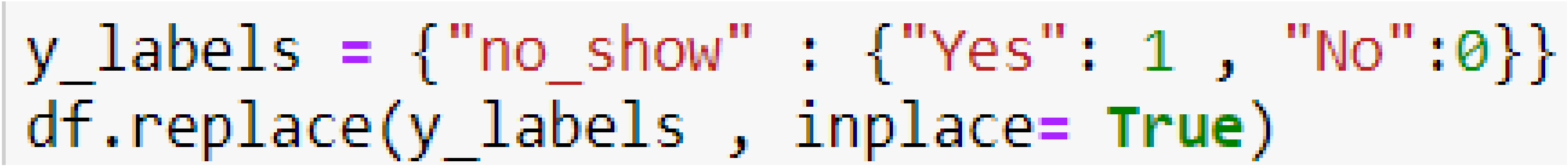
1) For all categorical variables the distributions of show / no-show for different categories look very similar. There is no clear indication of any of these variables having bigger than others impact on show / no-show characteristics.

2) For numerical variables, such as age, kids and patients in their 60s, 70s, and 80s are more likely to show to their appointments

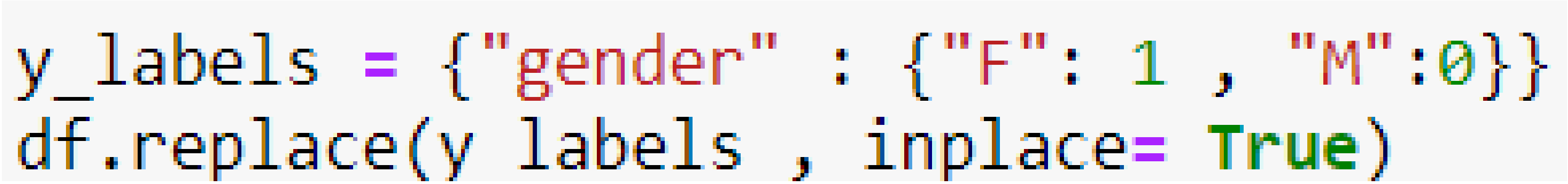
3) For numerical variable, such as waiting days (the most affecting variable in our dataset), shows that when the waiting day is shorter, the patient is more likely to show up for their appointments.

# Creating ML models for data prediction

1) Convert categorical data to numerical data for the models



```
y_labels = {"no_show" : {"Yes": 1 , "No":0}}
df.replace(y_labels , inplace= True)
```



```
y_labels = {"gender" : {"F": 1 , "M":0}}
df.replace(y labels , inplace= True)
```

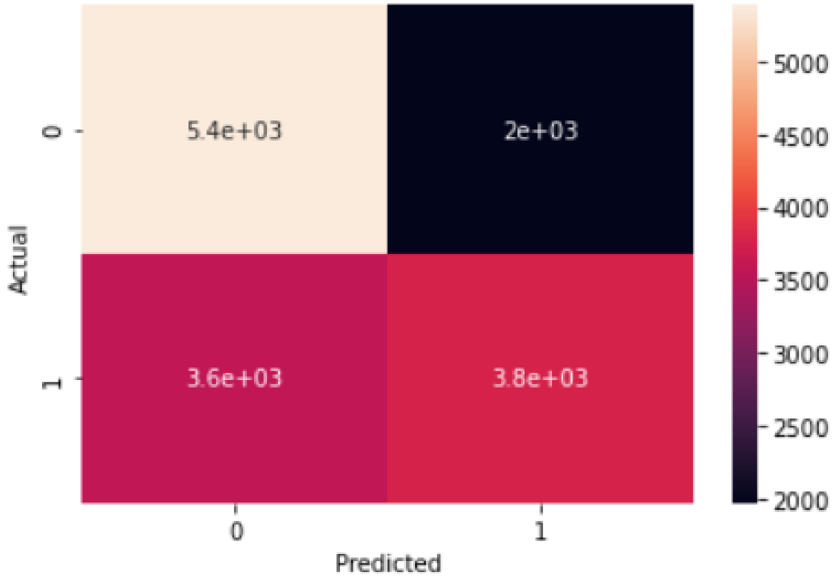
2) Split the data into actual data and test data

- We have used age and waiting_days columns only as they are the most affecting factors according to our analysis

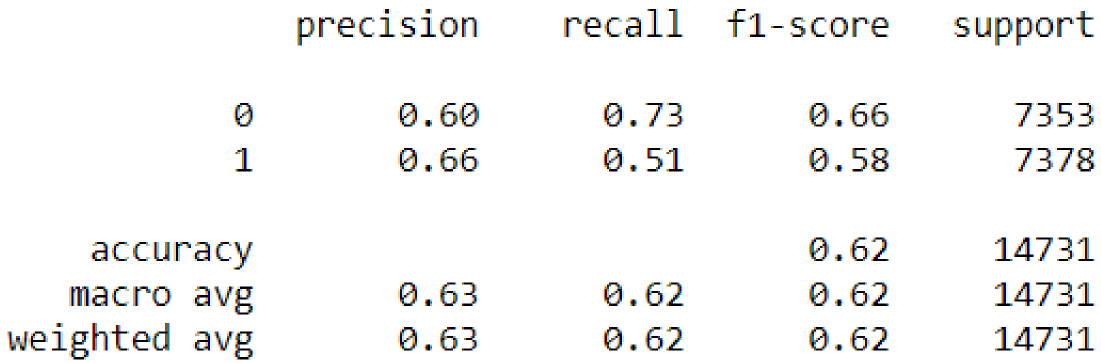
# A- Logistic Regression Model

- Accuracy of the model: 0.62 with cross-validation

- Confusion Matrix



Classification Report: Avg. accuracy = 0.6209



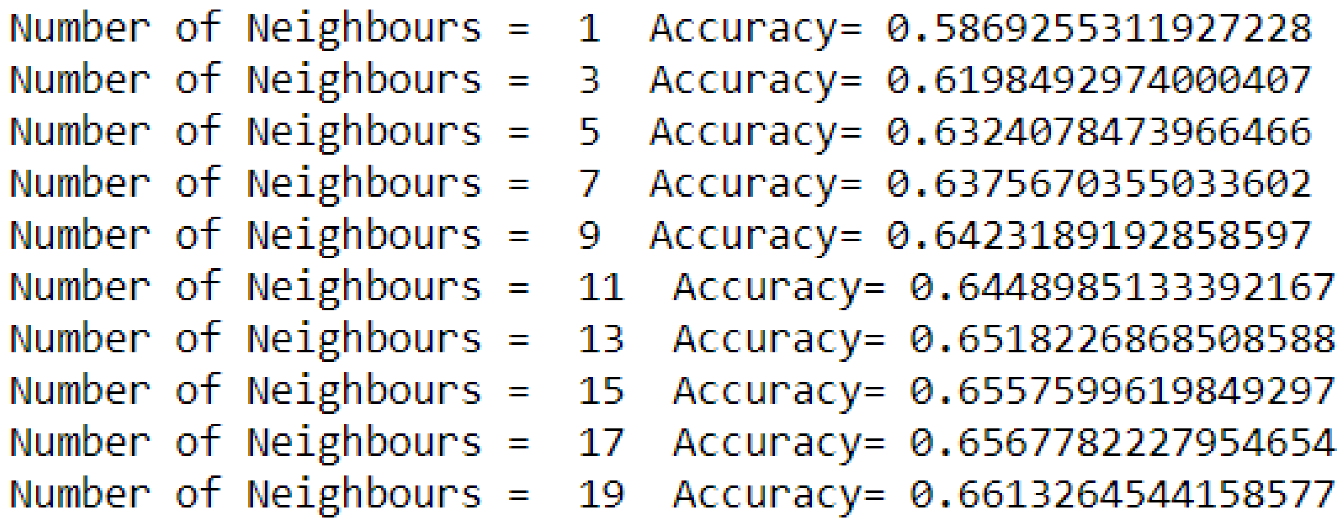
```
              precision    recall  f1-score   support

           0       0.60      0.73      0.66      7353
           1       0.66      0.51      0.58      7378

    accuracy                           0.62     14731
   macro avg       0.63      0.62      0.62     14731
weighted avg       0.63      0.62      0.62     14731
```

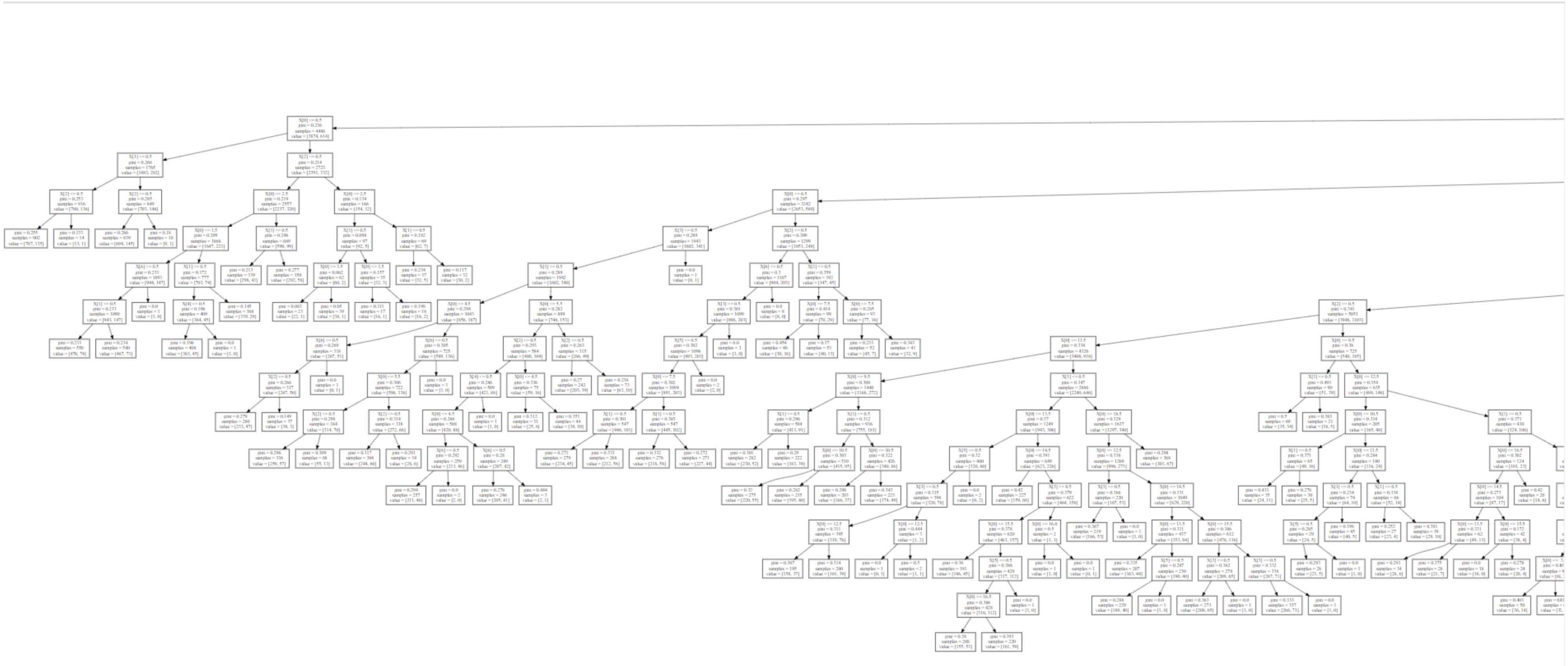
# B- KNN Model

KNN had the accuracy of 66.13% at 19 Neighbors



```
Number of Neighbours =  1  Accuracy= 0.5869255311927228
Number of Neighbours =  3  Accuracy= 0.6198492974000407
Number of Neighbours =  5  Accuracy= 0.6324078473966466
Number of Neighbours =  7  Accuracy= 0.6375670355033602
Number of Neighbours =  9  Accuracy= 0.6423189192858597
Number of Neighbours =  11  Accuracy= 0.6448985133392167
Number of Neighbours =  13  Accuracy= 0.6518226868508588
Number of Neighbours =  15  Accuracy= 0.6557599619849297
Number of Neighbours =  17  Accuracy= 0.6567782227954654
Number of Neighbours =  19  Accuracy= 0.6613264544158577
```

# C- Decision Tree Model

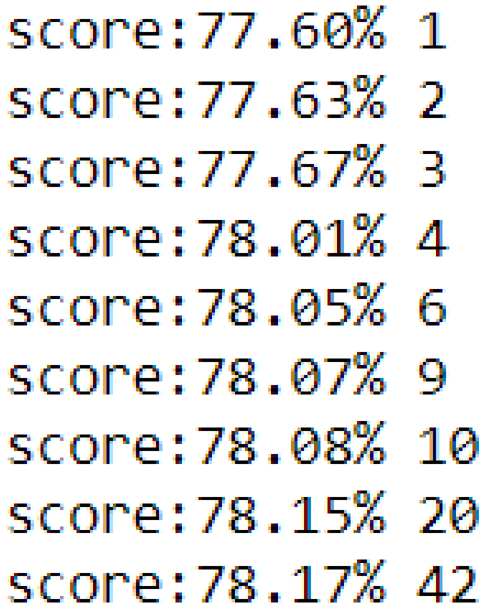


# C- Decision Tree Model

- The shown tree is only part of the actual tree, the actual tree will be attached with the code.
- The Decision Tree Classifier got to an accuracy of 78.37%

# D- Random Forest Classifier Model

- Score: 78.07%



```
score:77.60% 1
score:77.63% 2
score:77.67% 3
score:78.01% 4
score:78.05% 6
score:78.07% 9
score:78.08% 10
score:78.15% 20
score:78.17% 42
```

# Conclusion

The dataset was cleaned where a few problems like unifying names, removing wrong data, adding new features based on existing data were managed to enable analysis on the data. We also investigated some of the variables and prepared observations while comparing them to each other. Finally, we implemented some machine learning models to further investigate such as logistic regression, KNN, decision trees and lastly random forests. We can conclude that Random Forest Classifier is the best model to use on working on this dataset from out models to use as it gives a higher accuracy than other models and does not have overfitting.

# Thank you.