investigate-a-dataset-template

September 13, 2020

1 Project:Medical Appointment No Shows analysis project

1.1 Table of Contents

Introduction
Data Wrangling
outlier values
Exploratory Data Analysis
What factors are important for us to know in order to predict if a patient will show up for their scheduled appointment?
Can gender affect visit ?
Does age affect the visit?
what about the neighborhood ?
what about the Scholarship?
what about the Hypertension ?
what about the Diabetes ?
what about the Alcoholism ?
what about the Alcoholism ?
hat about the SMSReceived ?
Conclusions

## Introduction Hey in this notebook we will analysis data of medical Appointment No Shows I will do the best in this data and will show what i learning until now in this nanodegree this data contain 14 columns it is type datatime, float , int and object first we will clean, wrangling and Exploratory data Let's do this one step at a time

```
In [29]: #  import libraries
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
         %matplotlib inline
```

## Data Wrangling

1.1.1 General Properties

```
In [30]: # Load your data and print out a few lines. Perform operations to inspect data
         #   types and look for instances of missing or possibly errant data.
```

```
        df = pd.read_csv('noshowappointments-kagglev2-may-2016.csv')
        df.head()
```

```
Out[30]:       PatientId  AppointmentID Gender          ScheduledDay  \
         0  2.987250e+13        5642903      F  2016-04-29T18:38:08Z
         1  5.589978e+14        5642503      M  2016-04-29T16:08:27Z
         2  4.262962e+12        5642549      F  2016-04-29T16:19:04Z
         3  8.679512e+11        5642828      F  2016-04-29T17:29:31Z
         4  8.841186e+12        5642494      F  2016-04-29T16:07:23Z

                  AppointmentDay  Age      Neighbourhood  Scholarship  Hipertension  \
         0  2016-04-29T00:00:00Z   62    JARDIM DA PENHA            0             1
         1  2016-04-29T00:00:00Z   56    JARDIM DA PENHA            0             0
         2  2016-04-29T00:00:00Z   62      MATA DA PRAIA            0             0
         3  2016-04-29T00:00:00Z    8  PONTAL DE CAMBURI            0             0
         4  2016-04-29T00:00:00Z   56    JARDIM DA PENHA            0             1

            Diabetes  Alcoholism  Handcap  SMS_received No-show
         0         0           0        0             0      No
         1         0           0        0             0      No
         2         0           0        0             0      No
         3         0           0        0             0      No
         4         1           0        0             0      No
```

```
In [31]: #exploration data shape
         df.shape
```

```
Out[31]: (110527, 14)
```

```
In [32]: #the basic information about the data
         df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
PatientId         110527 non-null float64
AppointmentID     110527 non-null int64
Gender            110527 non-null object
ScheduledDay      110527 non-null object
AppointmentDay    110527 non-null object
Age               110527 non-null int64
Neighbourhood     110527 non-null object
Scholarship       110527 non-null int64
Hipertension      110527 non-null int64
Diabetes          110527 non-null int64
Alcoholism        110527 non-null int64
Handcap           110527 non-null int64
SMS_received      110527 non-null int64
No-show           110527 non-null object
```

```
dtypes: float64(1), int64(8), object(5)
memory usage: 11.8+ MB
```

```
In [33]: #Find out the null values in the data
         df.isnull().sum()
```

```
Out[33]: PatientId         0
         AppointmentID     0
         Gender            0
         ScheduledDay      0
         AppointmentDay    0
         Age               0
         Neighbourhood     0
         Scholarship       0
         Hipertension      0
         Diabetes          0
         Alcoholism        0
         Handcap           0
         SMS_received      0
         No-show           0
         dtype: int64
```

```
In [34]: #Knowing the duplicate values in the data
         df.duplicated().any()
```

```
Out[34]: False
```

```
In [35]: # Describing data
         df.describe()
```

```
Out[35]:           PatientId  AppointmentID            Age    Scholarship  \
         count  1.105270e+05   1.105270e+05  110527.000000  110527.000000
         mean   1.474963e+14   5.675305e+06      37.088874       0.098266
         std    2.560949e+14   7.129575e+04      23.110205       0.297675
         min    3.921784e+04   5.030230e+06      -1.000000       0.000000
         25%    4.172614e+12   5.640286e+06      18.000000       0.000000
         50%    3.173184e+13   5.680573e+06      37.000000       0.000000
         75%    9.439172e+13   5.725524e+06      55.000000       0.000000
         max    9.999816e+14   5.790484e+06     115.000000       1.000000

          Hipertension       Diabetes     Alcoholism        Handcap  \
  count  110527.000000  110527.000000  110527.000000  110527.000000
  mean        0.197246       0.071865       0.030400       0.022248
  std         0.397921       0.258265       0.171686       0.161543
  min         0.000000       0.000000       0.000000       0.000000
  25%         0.000000       0.000000       0.000000       0.000000
  50%         0.000000       0.000000       0.000000       0.000000
  75%         0.000000       0.000000       0.000000       0.000000
```

```
max          1.000000       1.000000       1.000000       4.000000

        SMS_received
count  110527.000000
mean        0.321026
std         0.466873
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         1.000000
```

1.1.2 Data Cleaning

If you notice you will find that both columns ScheduledDay and AppointmentDay initialize as objects We'll update to datetime

```
In [36]: ## Converting the date information in string to datetime type:
         df['ScheduledDay'] = pd.to_datetime(df.ScheduledDay)
         df['AppointmentDay'] = pd.to_datetime(df.AppointmentDay)
```

```
In [37]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
PatientId         110527 non-null float64
AppointmentID     110527 non-null int64
Gender            110527 non-null object
ScheduledDay      110527 non-null datetime64[ns, UTC]
AppointmentDay    110527 non-null datetime64[ns, UTC]
Age               110527 non-null int64
Neighbourhood     110527 non-null object
Scholarship       110527 non-null int64
Hipertension      110527 non-null int64
Diabetes          110527 non-null int64
Alcoholism        110527 non-null int64
Handcap           110527 non-null int64
SMS_received      110527 non-null int64
No-show           110527 non-null object
dtypes: datetime64[ns, UTC](2), float64(1), int64(8), object(3)
memory usage: 11.8+ MB
```

2 Let's look for the outlier values in that data

```
In [38]: #Let's start with a age
         #if you looke will see that thir outlir her
         (df.Age<=0).any()
```

```
Out[38]: True

In [39]: #We will only take the booleans for age
         df= df[(df.Age >= 0) & (df.Age <= 100)]

In [40]: ((df.Age <0) & (df.Age > 100) ).any()

Out[40]: False

In [41]: print(sorted(df.Age.unique()))

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 2
```

Let's make sure of the gender

```
In [42]: df.Gender.unique()
         #ok gender is good

Out[42]: array(['F', 'M'], dtype=object)
```

3 Let's check all of the columns

```
In [43]: print(df.Hipertension.unique() )
         print(df.Diabetes.unique()  )
         print(df.Alcoholism.unique() )
         print(df.Handcap.unique()  )

         #ok evrything is good

[1 0]
[0 1]
[0 1]
[0 1 2 3 4]


In [44]: df.head()

Out[44]:       PatientId  AppointmentID Gender              ScheduledDay  \
         0  2.987250e+13        5642903      F 2016-04-29 18:38:08+00:00
         1  5.589978e+14        5642503      M 2016-04-29 16:08:27+00:00
         2  4.262962e+12        5642549      F 2016-04-29 16:19:04+00:00
         3  8.679512e+11        5642828      F 2016-04-29 17:29:31+00:00
         4  8.841186e+12        5642494      F 2016-04-29 16:07:23+00:00

                       AppointmentDay  Age      Neighbourhood  Scholarship  \
         0 2016-04-29 00:00:00+00:00   62    JARDIM DA PENHA            0
         1 2016-04-29 00:00:00+00:00   56    JARDIM DA PENHA            0
         2 2016-04-29 00:00:00+00:00   62      MATA DA PRAIA            0
```

```
3 2016-04-29 00:00:00+00:00    8  PONTAL DE CAMBURI            0
4 2016-04-29 00:00:00+00:00   56    JARDIM DA PENHA            0

   Hipertension  Diabetes  Alcoholism  Handcap  SMS_received No-show
0             1         0           0        0             0      No
1             0         0           0        0             0      No
2             0         0           0        0             0      No
3             0         0           0        0             0      No
4             1         1           0        0             0      No
```

```
In [45]: df.shape
```

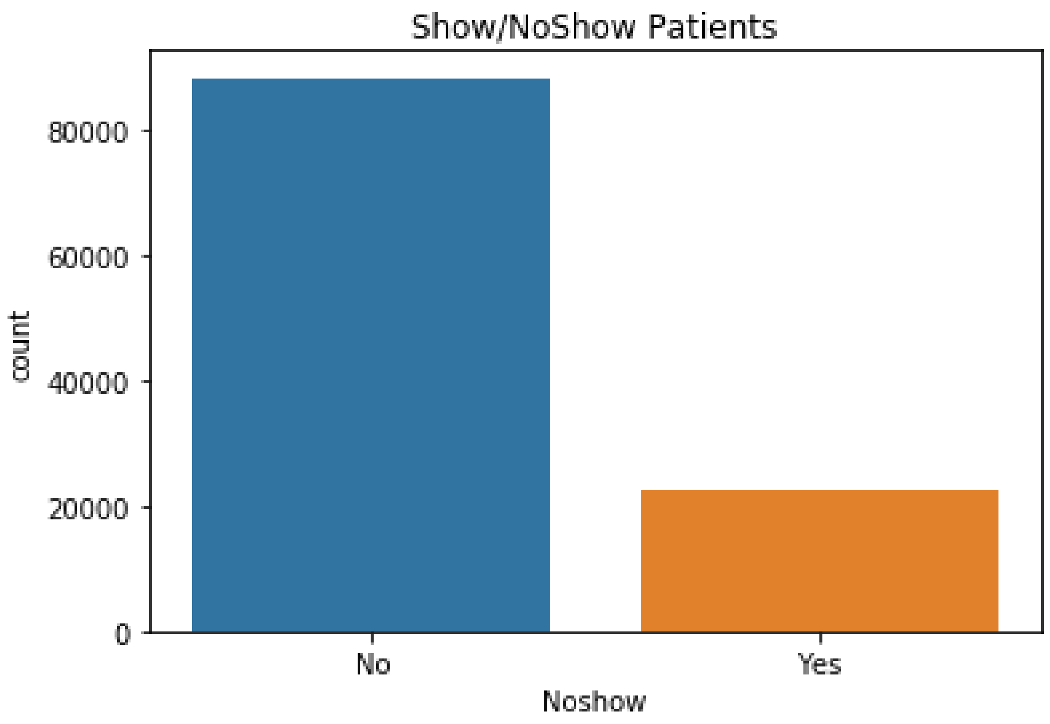
```
Out[45]: (110519, 14)
```

## Exploratory Data Analysis ### Research Question 1 (What factors are important for us to know in order to predict if a patient will show up for their scheduled appointment?)

```
In [46]: #First, we will change the name of No-show to Noshow. In order not to hinder progress
         df.rename(columns = {'No-show':'Noshow',}, inplace = True)
         df.head()
```

```
Out[46]:       PatientId  AppointmentID Gender              ScheduledDay  \
         0  2.987250e+13        5642903      F 2016-04-29 18:38:08+00:00
         1  5.589978e+14        5642503      M 2016-04-29 16:08:27+00:00
         2  4.262962e+12        5642549      F 2016-04-29 16:19:04+00:00
         3  8.679512e+11        5642828      F 2016-04-29 17:29:31+00:00
         4  8.841186e+12        5642494      F 2016-04-29 16:07:23+00:00

                      AppointmentDay  Age      Neighbourhood  Scholarship  \
         0 2016-04-29 00:00:00+00:00   62    JARDIM DA PENHA            0
         1 2016-04-29 00:00:00+00:00   56    JARDIM DA PENHA            0
         2 2016-04-29 00:00:00+00:00   62      MATA DA PRAIA            0
         3 2016-04-29 00:00:00+00:00    8  PONTAL DE CAMBURI            0
         4 2016-04-29 00:00:00+00:00   56    JARDIM DA PENHA            0

            Hipertension  Diabetes  Alcoholism  Handcap  SMS_received Noshow
         0             1         0           0        0             0     No
         1             0         0           0        0             0     No
         2             0         0           0        0             0     No
         3             0         0           0        0             0     No
         4             1         1           0        0             0     No
```

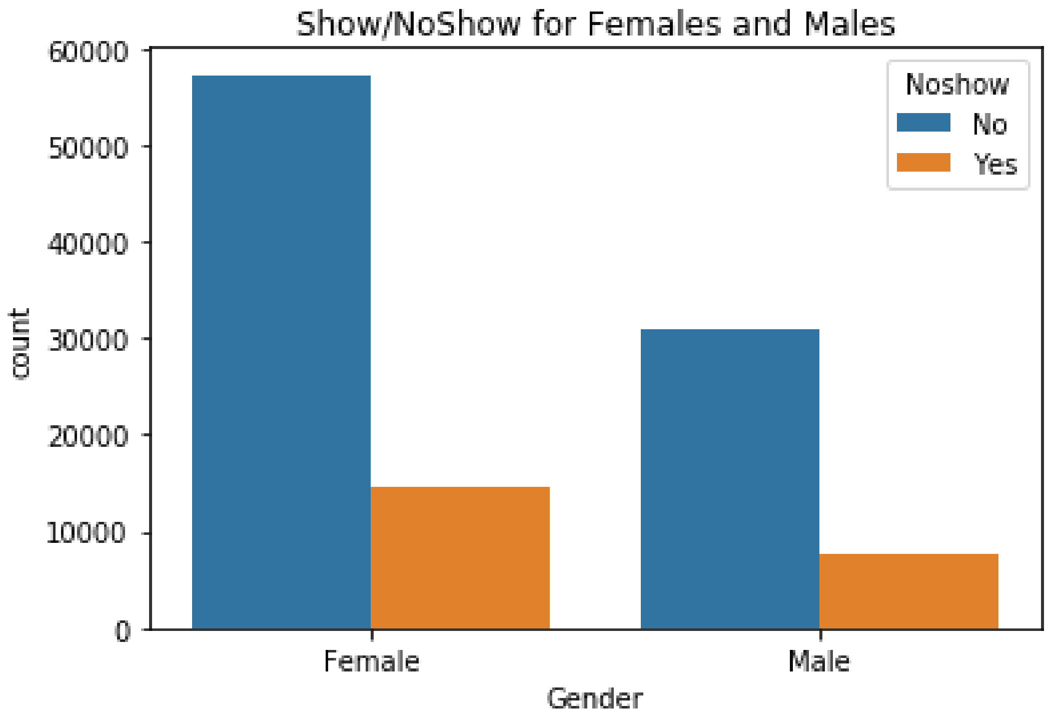
```
In [47]: #Below we can see that out of 110519 patients around 88,000 of them have turned up and

         ax = sns.countplot(x=df.Noshow, data=df)
         ax.set_title("Show/NoShow Patients")
         plt.show()
```



4 Question 2 Can gender affect visit ?

We can see that of the 88,000 patients that appeared, about 57,000 were female and 31,000 were male. Of the 22,500 patients who did not come for a visit, about 15,000 were females and 7,500 were males The ratio of females to males who attended appears to be the same as that which did not come to visit, and therefore gender does not affect

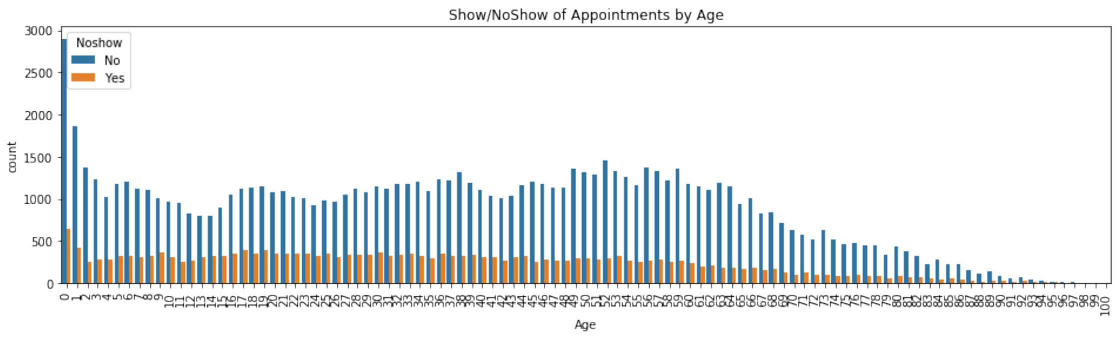
```
In [48]: ax = sns.countplot(x=df.Gender, hue=df.Noshow, data=df)
         ax.set_title("Show/NoShow for Females and Males")
         x_ticks_labels=['Female', 'Male']
         ax.set_xticklabels(x_ticks_labels)
         plt.show()
```



4.0.1 Research Question 3 (Does age affect the visit?)

From the above visualization, it appears that the ratio of Show to NoShow is nearly the same for all ages except for "Age 0" and "Age 1". We will get better clarity on the ratio of Show to NoShow for all ages. so age does not affect the commitment to visit much

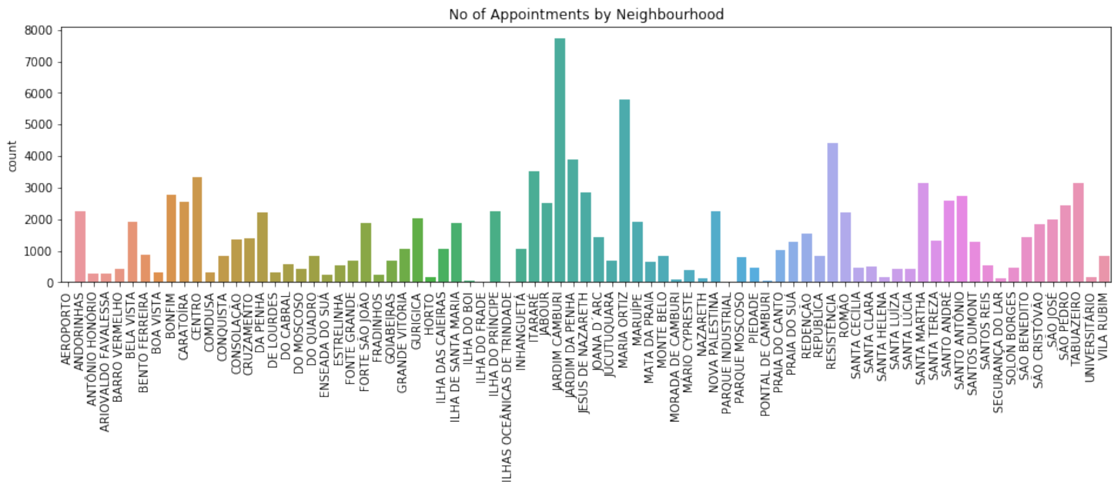
```
In [49]: plt.figure(figsize=(16,4))
         plt.xticks(rotation=90)
         ax = sns.countplot(x=df.Age, hue=df.Noshow)
         ax.set_title("Show/NoShow of Appointments by Age")
         plt.show()
```



5 Question 4 what about the neighborhood ?

From visualization we can see that the number of patients for few Neighbourhood's is very high.

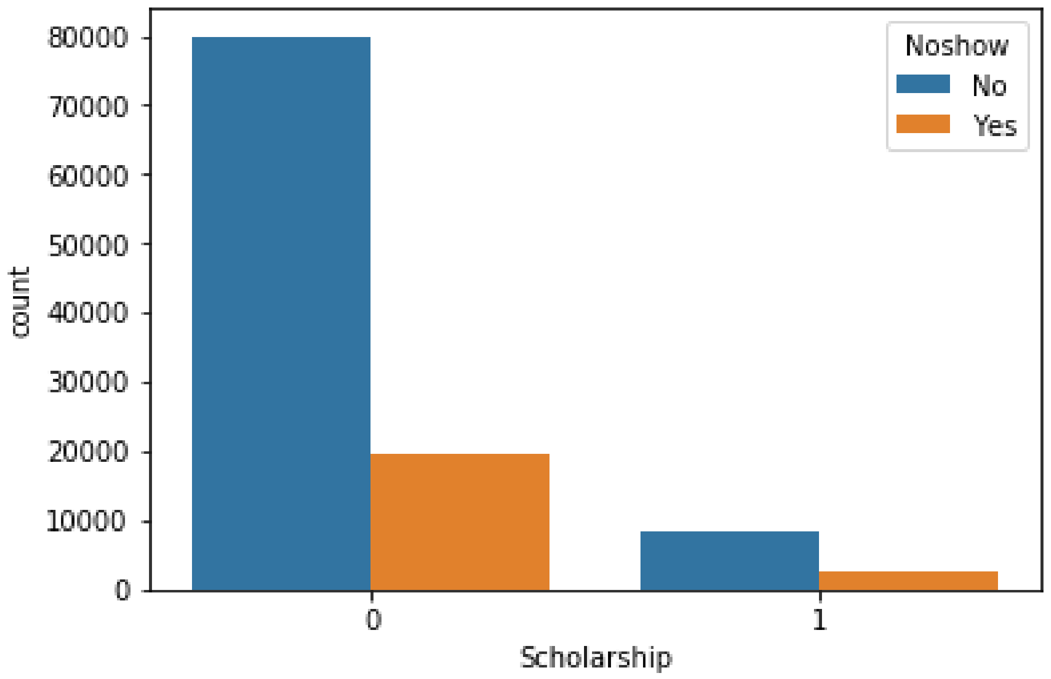
```
In [50]: #Below we will see the patients count for each Neighbourhood.

         plt.figure(figsize=(16,4))
         plt.xticks(rotation=90)
         ax = sns.countplot(x=np.sort(df.Neighbourhood))
         ax.set_title("No of Appointments by Neighbourhood")
         plt.show()
```



6 Question 5 what about the Scholarship?

From visualization we can see that there are around 100,000 patients without Scholarship and out of them around 80% have come for the visit. Out of the 10,500 patients with Scholarship around 75% of them have come for the visit. So, Scholarship feature could help us in determining if a patient will turn up for the visit after an appointment.

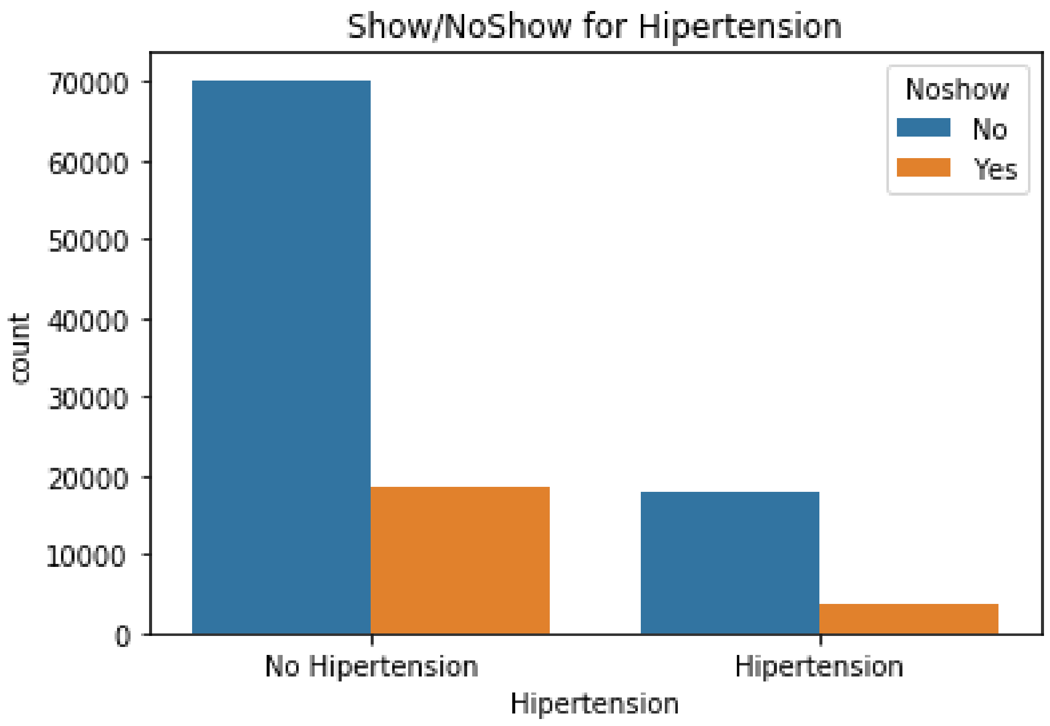
```
In [51]: x = sns.countplot(x=df.Scholarship, hue=df.Noshow, data=df)
         ax.set_title("Show/NoShow for Scholarship")
         x_ticks_labels=['No Scholarship', 'Scholarship']
         ax.set_xticklabels(x_ticks_labels)
         plt.show()
```



7 Question 6 what about the Hypertension ?

By visualizing, we can see that there are about 88,000 patients suffering from high blood pressure and about 78% of them attended the visit. Of 22,500 patients with high blood pressure, about 85% came to visit. Therefore, the high blood pressure feature can help us determine whether a patient will show up on a post-appointment visit.

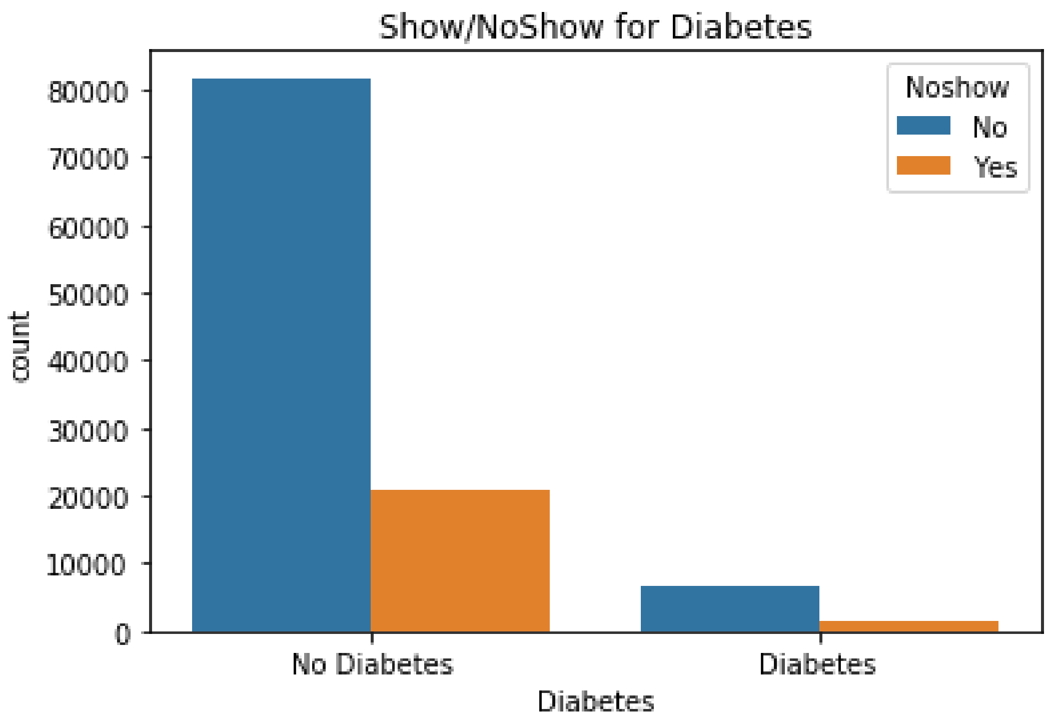
```
In [52]: ax = sns.countplot(x=df.Hipertension, hue=df.Noshow, data=df)
         ax.set_title("Show/NoShow for Hipertension")
         x_ticks_labels=['No Hipertension', 'Hipertension']
         ax.set_xticklabels(x_ticks_labels)
         plt.show()
```



8 Question 7 what about the Diabetes ?

From visualization we can see that there are about 102,000 diabetics and about 80% of them attended the visit. Of the 8,500 diabetic patients, about 83% came to visit. Therefore, the diabetes feature can help us determine whether a patient will attend the post-appointment visit.

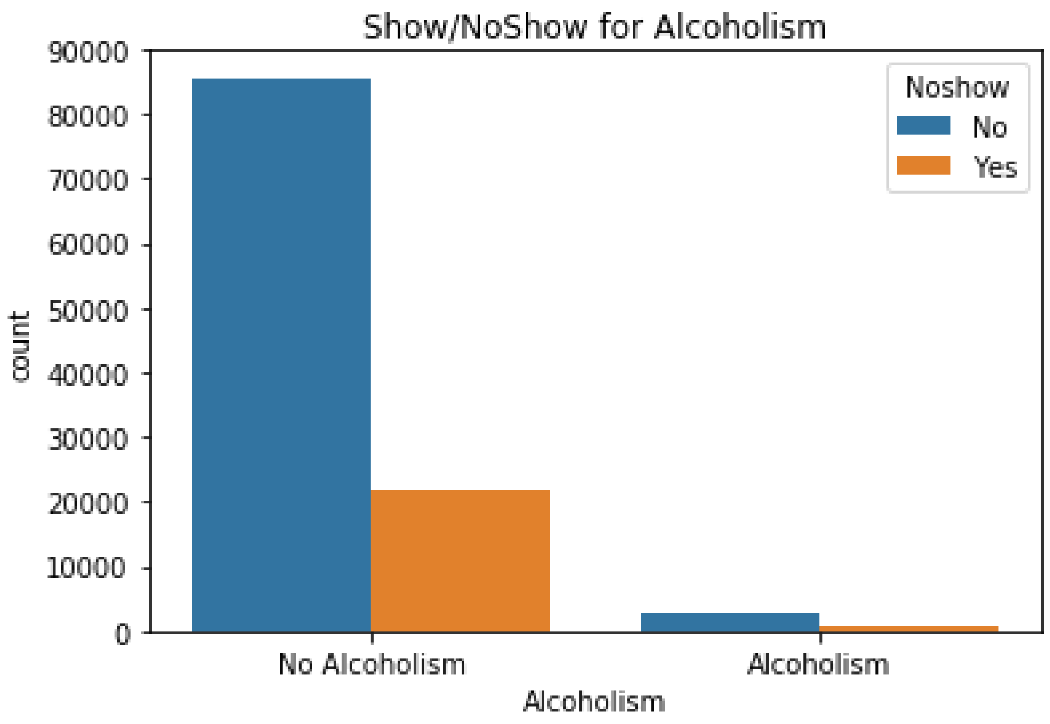
```
In [53]: ax = sns.countplot(x=df.Diabetes, hue=df.Noshow, data=df)
         ax.set_title("Show/NoShow for Diabetes")
         x_ticks_labels=['No Diabetes', 'Diabetes']
         ax.set_xticklabels(x_ticks_labels)
         plt.show()
```



9 Question 8 what about the Alcoholism ?

By visualizing, we can see that there are about 105,000 patients who do not suffer from alcoholism and about 80% of them attended the visit. Of the 5,500 patients with alcohol addiction, about 80% attended the visit. Since the rate of visits for non-alcoholic patients is the same, this may not help us determine whether or not the patient is coming for a visit.

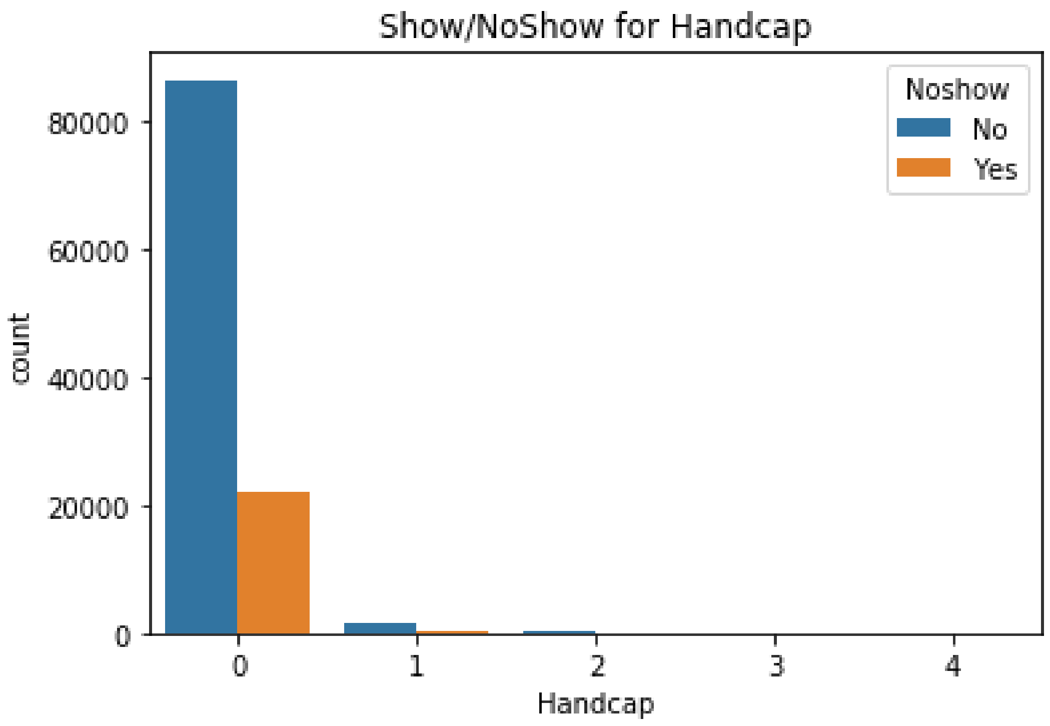
```
In [54]: ax = sns.countplot(x=df.Alcoholism, hue=df.Noshow, data=df)
         ax.set_title("Show/NoShow for Alcoholism")
         x_ticks_labels=['No Alcoholism', 'Alcoholism']
         ax.set_xticklabels(x_ticks_labels)
         plt.show()
```



10 Question 9 what about the Alcoholism ?

Through visualization, we can see that there are about 110,000 unobstructed patients and about 80% of them have come for a visit. Since we see a clear distinction between different levels of disability, this feature will help us determine if a patient will come for a visit after making an appointment.

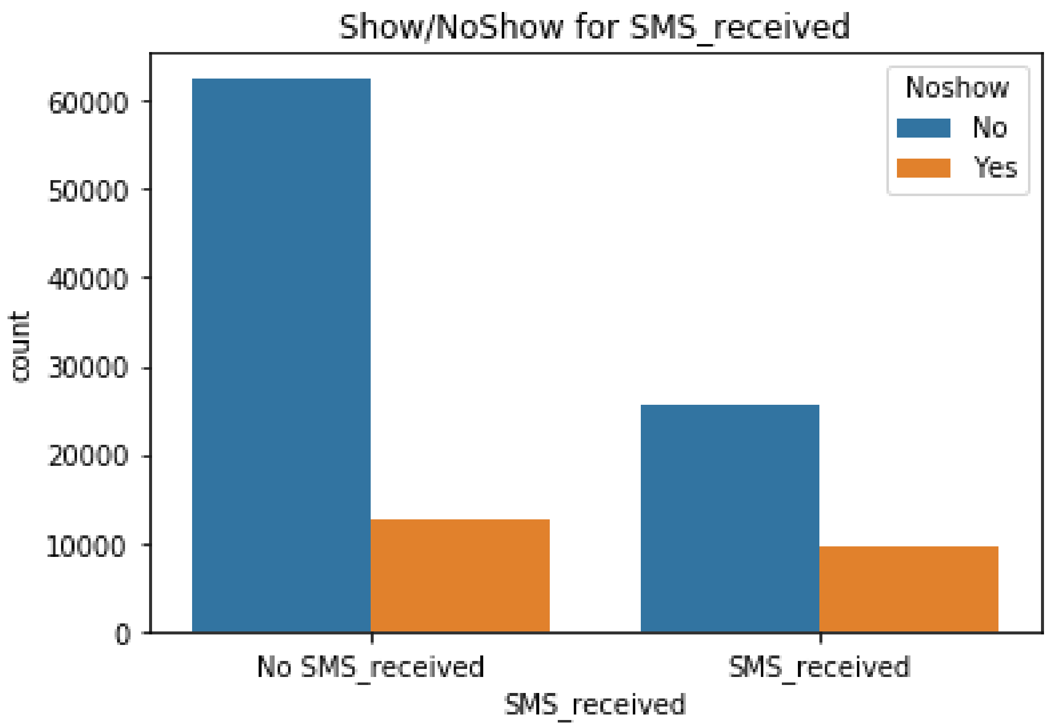
```
In [55]: ax = sns.countplot(x=df.Handcap, hue=df.Noshow, data=df)
         ax.set_title("Show/NoShow for Handcap")
         plt.show()
```



11 Question 10 what about the SMSReceived ?

Through visualization, we can see that there are about 75,000 patients who did not receive text messages, and about 84% of them attended the visit. Of the 35,500 patients who received text messages, about 72% attended the visit. This feature will help us determine if a patient will come for a visit after scheduling an appointment.

```
In [56]: ax = sns.countplot(x=df.SMS_received, hue=df.Noshow, data=df)
         ax.set_title("Show/NoShow for SMS_received")
         x_ticks_labels=['No SMS_received', 'SMS_received']
         ax.set_xticklabels(x_ticks_labels)
         plt.show()
```



## conclusions Finally, we performed a comprehensive analysis of the data and familiarized us with the factors that affect the visit and the factors that do not, in order to exclude it. We applied everything we had learned in the class and used most of the functions explained By analyzing and tracking the results, we find that gender and age are the most important factors. As we saw earlier, females and young adults appear to be more likely to appoint than males and older adults. Neighborhood and high blood pressure come after sex and age, as there are some neighborhoods where diseases are common and high blood pressure patients tend to appear if they have it or not. So we need to research more factors to help the patient remember appointments and appear. But there are still some unsatisfactory results, such as the way we deal with data that expresses time. Do we use it as it is or do we divide it into a number of columns Does the day, month, or year affect whether the patient comes to the visit, or not? Does the name of the neighborhood represent a difference or is it just a coincidence that there are a large number of patients from the same neighborhood, perhaps they are relatives of the doctor, for example is the data provided by the dataset sufficient to answer this questions? Also, are their columns such as (PatientId and AppointmentID) will affect on the visit All of these are questions that need more research and study because they represent an obstacle We will learn it by training more

```
In [ ]:
```

```
In [ ]:
```