Data Scientist Test

By Julian Munoz - due date: June 30th - 4pm

In this dataset (https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008#) you have 3 different outputs:

1. No readmission;

2. A readmission in less than 30 days (this situation is not good, because maybe your treatment was not appropriate);

3. A readmission in more than 30 days (this one is not so good as well the last one, however, the reason could be the state of the patient.

Your task is either to classify a patient-hospital outcome or to cluster them aiming at finding patterns that give a distinct insight.

To do so, we suggest you create a notebook, like Jupyter (if you use python) or a Rmarkdown report (in case you use R) and make it available for us, i.e. github.

Hint to success in your quest: Develop and stay clear of the data science process you'll perform over the dataset and highlight important aspects you might consider affordable to discuss over.

You have up to a day before the technical interview to share your results of this test.

Good luck.

DataScientistTest development contents Using Exploratory Data Analysis (EDA) process:

Following the EDA process I carried out the following steps:

1. Check dataset provided

a. Load dataset.

b. Check dataset for "errors" or "inaccuracies":

i. Missing values

ii. Outline Atypical values

iii. Correlation

2. Clean data, process, establish a hypothesis

present results / script to results generation

Report to communicate results

Conclusions / Decision making

1. Checking dataset a. Loading the dataset

In [4]:

```
import pandas #activating pandas library
```

In [5]:

```
pandas.read_csv('diabetic_data.csv') #reading dataset from csv file
```

Out[5]:

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital
0	2278392	8222157	Caucasian	Female	[0-10)	?	6	25	1	1

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital
1	149190	55629189	Caucasian	Female	[10-20)	?	1	1	7	3
2	64410	86047875	AfricanAmerican	Female	[20-30)	?	1	1	7	2
3	500364	82442376	Caucasian	Male	[30-40)	?	1	1	7	2
4	16680	42519267	Caucasian	Male	[40-50)	?	1	1	7	1
...	...	...	...	...	...	...	...	...	...	...
101761	443847548	100162476	AfricanAmerican	Male	[70-80)	?	1	3	7	3
101762	443847782	74694222	AfricanAmerican	Female	[80-90)	?	1	4	5	5
101763	443854148	41088789	Caucasian	Male	[70-80)	?	1	1	7	1
101764	443857166	31693671	Caucasian	Female	[80-90)	?	2	3	7	10
101765	443867222	175429310	Caucasian	Male	[70-80)	?	1	1	7	6

101766 rows × 50 columns

In [6]:

```
#using pandas profiling library to check data
import pandas_profiling
import pandas as pd
```

1. Loading dataset b. Checking dataset for errors or inaccuracies

In [7]:

```
#reading the dataset
data=pd.read_csv("diabetic_data.csv")
```

In [8]:

```
data.head() #checking data
```

Out[8]:

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	...	ci
0	2278392	8222157	Caucasian	Female	[0-10)	?	6	25	1	1	...	
1	149190	55629189	Caucasian	Female	[10-20)	?	1	1	7	3	...	
2	64410	86047875	AfricanAmerican	Female	[20-30)	?	1	1	7	2	...	
3	500364	82442376	Caucasian	Male	[30-40)	?	1	1	7	2	...	
4	16680	42519267	Caucasian	Male	[40-50)	?	1	1	7	1	...	

5 rows × 50 columns

In [9]:

```
data.tail() #checking data
```

Out[9]:

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital
101761	443847548	100162476	AfricanAmerican	Male	[70-80)	?	1	3	7	3
101762	443847782	74694222	AfricanAmerican	Female	[80-90)	?	1	4	5	5
101763	443854148	41088789	Caucasian	Male	[70-80)	?	1	1	7	1

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital
101764	443857166	31693671	Caucasian	Female	[80-90)	?	2	3	7	10
101765	443867222	175429310	Caucasian	Male	[70-80)	?	1	1	7	6

5 rows × 50 columns

In [10]:

```
data.sample(100) #sampling records
```

Out[10]:

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital
39310	122351166	43372989	Caucasian	Male	[50-60)	?	1	1	7	7
88594	284966970	58160520	AfricanAmerican	Male	[90-100)	?	1	6	7	3
90025	292107282	111514617	Caucasian	Male	[80-90)	?	2	22	7	4
82181	255761652	90694521	Caucasian	Female	[60-70)	?	3	1	1	1
77636	236096514	112108122	Caucasian	Male	[50-60)	?	3	1	7	4
...	...	...	...	...	...	...	...	...	...	...
50689	152002716	35440182	AfricanAmerican	Female	[30-40)	?	1	1	7	3
83466	261485016	1978605	Caucasian	Female	[50-60)	?	2	6	7	6
96691	379016084	113825277	Hispanic	Male	[70-80)	?	3	1	1	9
47704	146475366	2040903	Caucasian	Female	[90-100)	?	1	6	7	4
19736	70083552	77613561	Caucasian	Male	[70-80)	?	3	1	1	2

100 rows × 50 columns

In [11]:

```
#using pandas profiling report to know a little more about the data that I am going to analyze
pandas_profiling.ProfileReport(data)
```

Overview

Dataset statistics

Number of variables	50
Number of observations	101766
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	38.8 MiB
Average record size in memory	400.0 B

Variable types

Numeric	13
Categorical	34
Boolean	3

Warnings

`examide` has constant value "False"	Constant
`citoglipton` has constant value "False"	Constant
`medical_specialty` has a high cardinality: 73 distinct values	High cardinality
`diag_1` has a high cardinality: 717 distinct values	High cardinality
`diag_2` has a high cardinality: 749 distinct values	High cardinality
`diag_3` has a high cardinality: 790 distinct values	High cardinality
`encounter_id` is highly correlated with `patient_nbr`	High correlation

Out[11]:

In [50]:
```
#Values of readmitted column
data['readmitted'].value_counts()
```

Out[50]:
```
NO     54864
>30    35545
<30    11357
Name: readmitted, dtype: int64
```

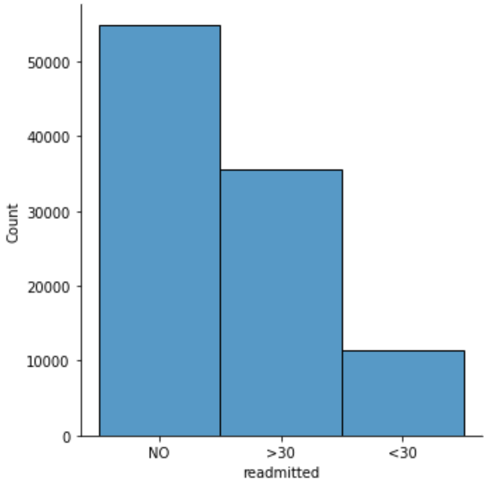
In [51]:
```
#describe column
data['readmitted'].describe()
```

Out[51]:
```
count     101766
unique         3
top           NO
freq       54864
Name: readmitted, dtype: object
```

In [52]:
```
import seaborn as sns
```

In [54]:
```
#distribution of target column
sns.displot(data['readmitted'])
```

Out[54]:
```
<seaborn.axisgrid.FacetGrid at 0x15415ae92b0>
```



In [55]:

```
#get descriptive statistics for all the numerical columns in the dataset
data.describe()
```

Out[55]:

	encounter_id	patient_nbr	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	num_lab_procedures	num_procedures
count	1.017660e+05	1.017660e+05	101766.000000	101766.000000	101766.000000	101766.000000	101766.000000	101766.000000
mean	1.652016e+08	5.433040e+07	2.024006	3.715642	5.754437	4.395987	43.095641	1.339730
std	1.026403e+08	3.869636e+07	1.445403	5.280166	4.064081	2.985108	19.674362	1.705807
min	1.252200e+04	1.350000e+02	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
25%	8.496119e+07	2.341322e+07	1.000000	1.000000	1.000000	2.000000	31.000000	0.000000
50%	1.523890e+08	4.550514e+07	1.000000	1.000000	7.000000	4.000000	44.000000	1.000000
75%	2.302709e+08	8.754595e+07	3.000000	4.000000	7.000000	6.000000	57.000000	2.000000
max	4.438672e+08	1.895026e+08	8.000000	28.000000	25.000000	14.000000	132.000000	6.000000

2. Clean data, process, establish a hypothesis

After performing some tests on the data, it must be determined if the readmissions of patients correspond to their origin from another hospital or if the readmission is due to other factors such as the result of the variables of insulin, max_glu_serum, A1Cresult or another diagnosis.

My hypothesize:The readmission of patients is directly associated with the decision to perform the A1c test on the patient, not with their origin from another hospital.

In [56]:

```
#importing Data Analytic Baseline library
import pandas as pd
import dabl
```

In [59]:

```
#performing some data cleaning to determine patient origin
```

In [57]:

```
data_clean = dabl.clean(data, target_col='Outcome', verbose=1)
data_clean.head()
```

```
Detected feature types:
continuous       2
dirty_float      3
low_card_int    11
categorical     17
date             0
free_string      0
useless         17
dtype: int64
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
c:\users\jmunoz\appdata\local\programs\python\python39\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, t
olerance)
   3080             try:
-> 3081                 return self._engine.get_loc(casted_key)
   3082             except KeyError as err:
```

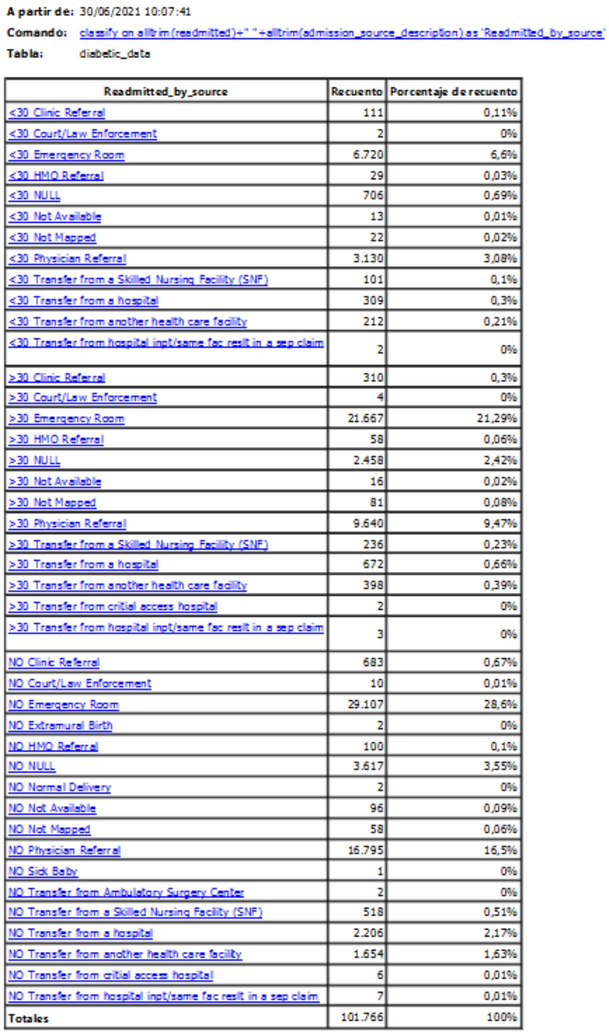
```
pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'Outcome'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
<ipython-input-57-b6898c0ad34b> in <module>
----> 1 data_clean = dabl.clean(data, target_col='Outcome', verbose=1)
      2 data_clean.head()

c:\users\jmunoz\appdata\local\programs\python\python39\lib\site-packages\dabl\preprocessing.py in clean(X, type_hints, return_types,
target_col, verbose)
    463
    464         # discard dirty float targets that cant be converted to float
--> 465         if target_col is not None and types_p['dirty_float'][target_col]:
    466             warn("Discarding dirty_float targets that cannot be converted "
    467                  "to float.", UserWarning)

c:\users\jmunoz\appdata\local\programs\python\python39\lib\site-packages\pandas\core\series.py in __getitem__(self, key)
    851
    852         elif key_is_scalar:
--> 853             return self._get_value(key)
    854
    855         if is_hashable(key):

c:\users\jmunoz\appdata\local\programs\python\python39\lib\site-packages\pandas\core\series.py in _get_value(self, label, takeable)
    959
    960         # Similar to Index.get_value, but we do not fall back to positional
--> 961         loc = self.index.get_loc(label)
    962         return self.index._get_values_for_loc(self, loc, label)
    963

c:\users\jmunoz\appdata\local\programs\python\python39\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, t
olerance)
   3081                 return self._engine.get_loc(casted_key)
   3082             except KeyError as err:
-> 3083                 raise KeyError(key) from err
   3084
   3085         if tolerance is not None:

KeyError: 'Outcome'
```

In [40]:

```python
import sweetviz
import pandas as pd
```

Due to some issues related to python libraries, errors were presented with the analysis, I had to perform them with ACL

First check... crosstab readmission by admission source



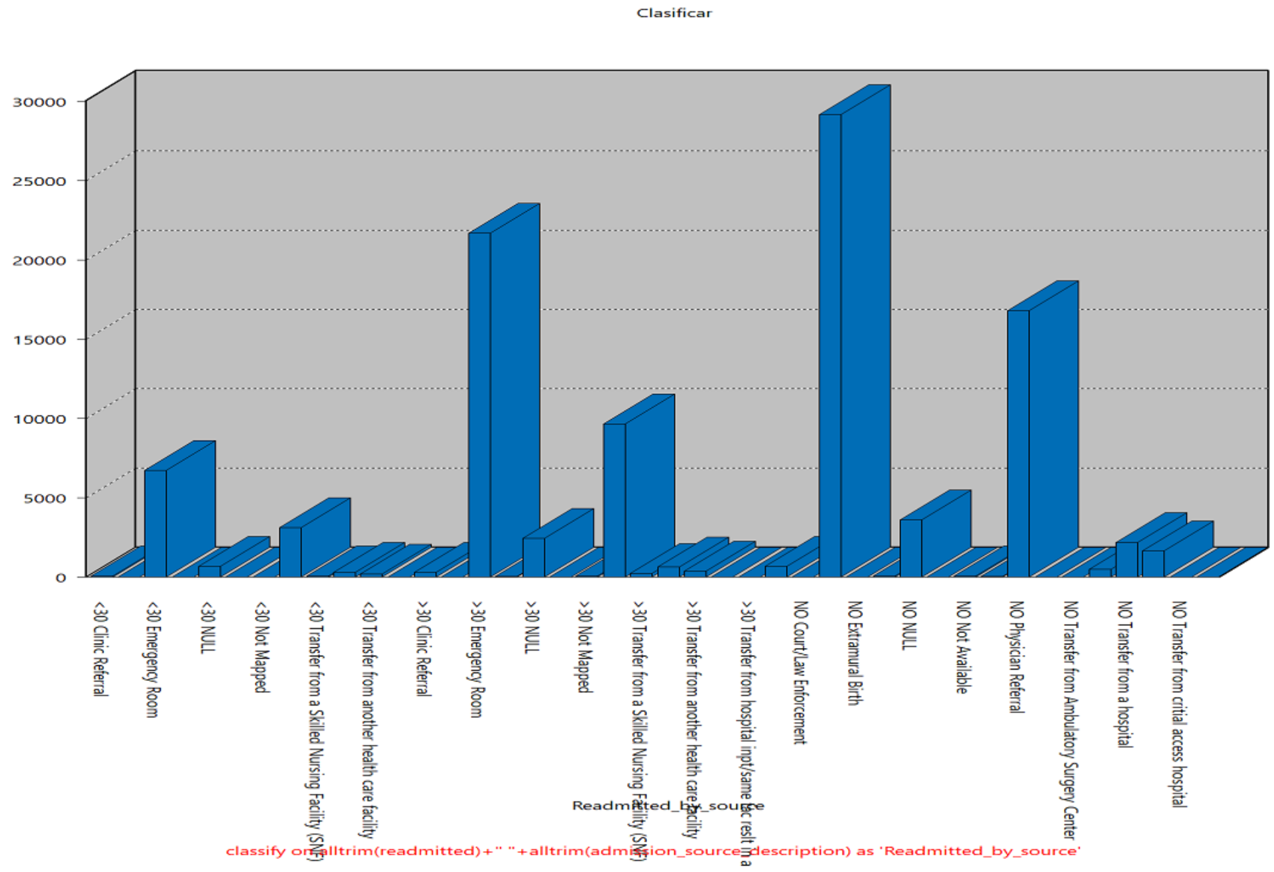
A partir de: 30/06/2021 10:07:41

Comando: classify on alltrim(readmitted)+" "+alltrim(admission_source_description) as 'Readmitted_by_source'

Tabla: diabetic_data

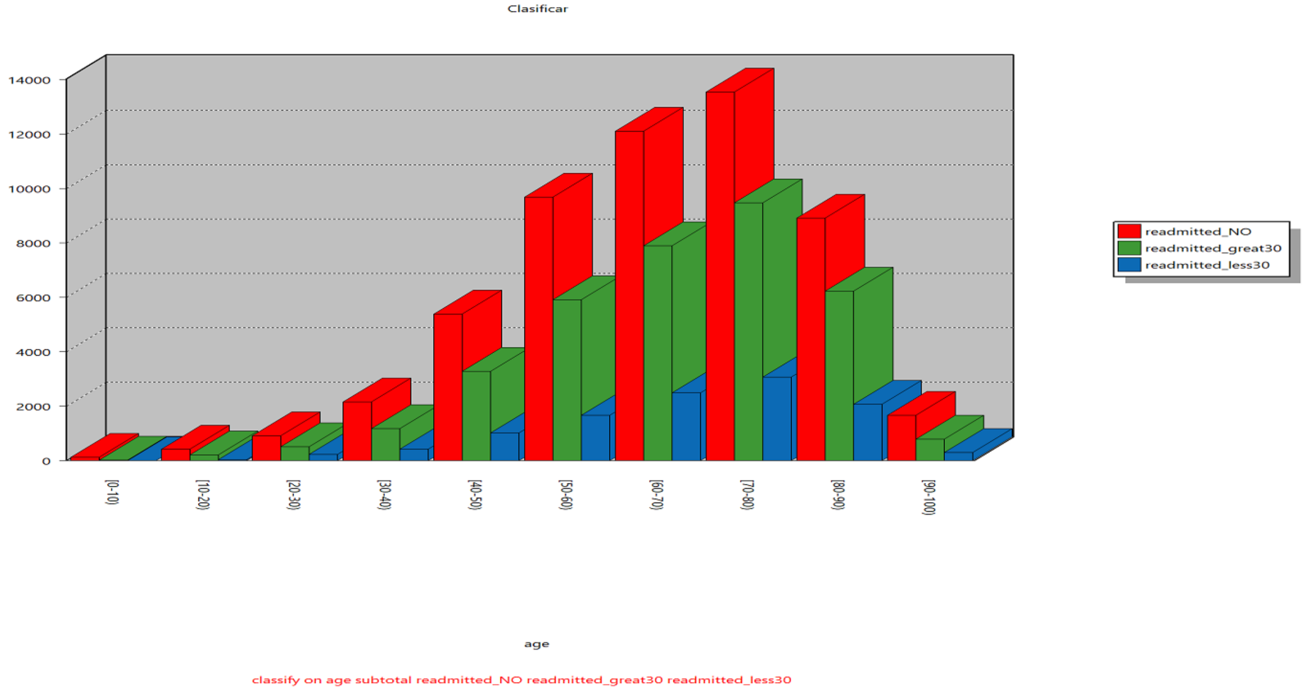
Readmitted_by_source	Recuento	Porcentaje de recuento
<30 Clinic Referral	111	0,11%
<30 Court/Law Enforcement	2	0%
<30 Emergency Room	6.720	6,6%
<30 HMO Referral	29	0,03%
<30 NULL	706	0,69%
<30 Not Available	13	0,01%
<30 Not Mapped	22	0,02%
<30 Physician Referral	3.130	3,08%
<30 Transfer from a Skilled Nursing Facility (SNF)	101	0,1%
<30 Transfer from a hospital	309	0,3%
<30 Transfer from another health care facility	212	0,21%
<30 Transfer from hospital inpt/same fac reslt in a sep claim	2	0%
>30 Clinic Referral	310	0,3%
>30 Court/Law Enforcement	4	0%
>30 Emergency Room	21.667	21,29%
>30 HMO Referral	58	0,06%
>30 NULL	2.458	2,42%
>30 Not Available	16	0,02%
>30 Not Mapped	81	0,08%
>30 Physician Referral	9.640	9,47%
>30 Transfer from a Skilled Nursing Facility (SNF)	236	0,23%
>30 Transfer from a hospital	672	0,66%
>30 Transfer from another health care facility	398	0,39%
>30 Transfer from critial access hospital	2	0%
>30 Transfer from hospital inpt/same fac reslt in a sep claim	3	0%
NO Clinic Referral	683	0,67%
NO Court/Law Enforcement	10	0,01%
NO Emergency Room	29.107	28,6%
NO Extramural Birth	2	0%
NO HMO Referral	100	0,1%
NO NULL	3.617	3,55%
NO Normal Delivery	2	0%
NO Not Available	96	0,09%
NO Not Mapped	58	0,06%
NO Physician Referral	16.795	16,5%
NO Sick Baby	1	0%
NO Transfer from Ambulatory Surgery Center	2	0%
NO Transfer from a Skilled Nursing Facility (SNF)	518	0,51%
NO Transfer from a hospital	2.206	2,17%
NO Transfer from another health care facility	1.654	1,63%
NO Transfer from critial access hospital	6	0,01%
NO Transfer from hospital inpt/same fac reslt in a sep claim	7	0,01%
Totales	101.766	100%

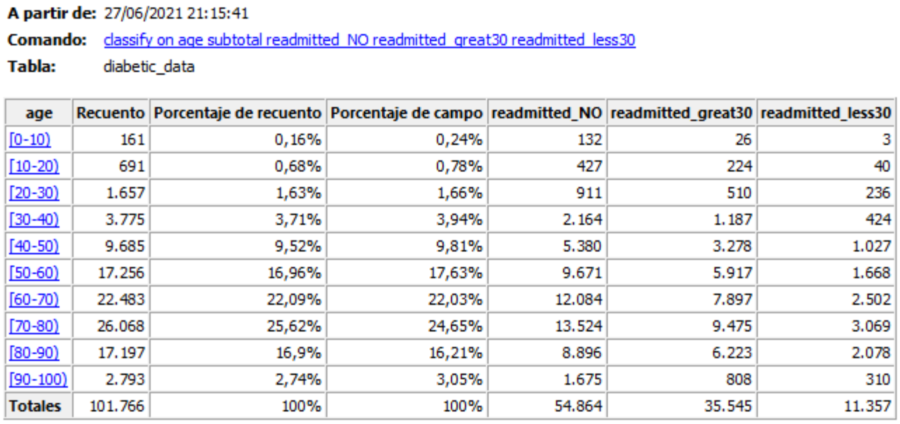
graph of readmission by admission source



Evidence to verify hypothesize: Many patients readmitted from Emergency Room and Skilled Nursing facility (SNF)

Check patients by age





A partir de: 27/06/2021 21:15:41

Comando: classify on age subtotal readmitted_NO readmitted_great30 readmitted_less30

Tabla: diabetic_data

age	Recuento	Porcentaje de recuento	Porcentaje de campo	readmitted_NO	readmitted_great30	readmitted_less30
[0-10)	161	0,16%	0,24%	132	26	3
[10-20)	691	0,68%	0,78%	427	224	40
[20-30)	1.657	1,63%	1,66%	911	510	236
[30-40)	3.775	3,71%	3,94%	2.164	1.187	424
[40-50)	9.685	9,52%	9,81%	5.380	3.278	1.027
[50-60)	17.256	16,96%	17,63%	9.671	5.917	1.668
[60-70)	22.483	22,09%	22,03%	12.084	7.897	2.502
[70-80)	26.068	25,62%	24,65%	13.524	9.475	3.069
[80-90)	17.197	16,9%	16,21%	8.896	6.223	2.078
[90-100)	2.793	2,74%	3,05%	1.675	808	310
Totales	101.766	100%	100%	54.864	35.545	11.357

In [ ]: