Solucion_Laboratorio6

March 17, 2021

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

[2]:
```
os.chdir("D:\Social Data Consulting\Python for Data Science\data")
```

[3]:
```
fileCsv="titanic.csv"
df_titanic=pd.read_csv(fileCsv,sep=',')
df_titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1310 entries, 0 to 1309
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   pclass     1309 non-null   float64
 1   survived   1309 non-null   float64
 2   name       1309 non-null   object 
 3   sex        1309 non-null   object 
 4   age        1046 non-null   float64
 5   sibsp      1309 non-null   float64
 6   parch      1309 non-null   float64
 7   ticket     1309 non-null   object 
 8   fare       1308 non-null   float64
 9   cabin      295 non-null    object 
 10  embarked   1307 non-null   object 
 11  boat       486 non-null    object 
 12  body       121 non-null    float64
 13  home.dest  745 non-null    object 
dtypes: float64(7), object(7)
memory usage: 143.4+ KB
```

[4]:
```
df_titanic.head()
```

[4]:
```
   pclass  survived                                             name     sex  \
0     1.0       1.0                    Allen, Miss. Elisabeth Walton  female
1     1.0       1.0                   Allison, Master. Hudson Trevor    male
2     1.0       0.0                     Allison, Miss. Helen Loraine  female
3     1.0       0.0             Allison, Mr. Hudson Joshua Creighton    male
4     1.0       0.0  Allison, Mrs. Hudson J C (Bessie Waldo Daniels)  female

        age  sibsp  parch  ticket      fare    cabin embarked boat   body  \
0  29.0000    0.0    0.0   24160  211.3375       B5        S    2    NaN
1   0.9167    1.0    2.0  113781  151.5500  C22 C26        S   11    NaN
2   2.0000    1.0    2.0  113781  151.5500  C22 C26        S  NaN    NaN
3  30.0000    1.0    2.0  113781  151.5500  C22 C26        S  NaN  135.0
4  25.0000    1.0    2.0  113781  151.5500  C22 C26        S  NaN    NaN

                        home.dest
0                    St Louis, MO
1  Montreal, PQ / Chesterville, ON
2  Montreal, PQ / Chesterville, ON
3  Montreal, PQ / Chesterville, ON
4  Montreal, PQ / Chesterville, ON
```

EVALUACION Y LIMPIEZA DE LA DATA

1.Solo considerar las variables para la limpieza de datos(class,sex,age,sibsp,parch,fare,embarked)

[5]:
```
variablesEliminar=["survived","name","ticket","cabin","boat","body","home.dest"]
```

[6]:
```
df_titanic.drop(variablesEliminar,axis=1, inplace=True)
```

Nos quedamos con las variables necesarias para la limpieza de la data

[7]:
```
df_titanic.head()
```

[7]:
```
   pclass     sex      age  sibsp  parch      fare embarked
0     1.0  female  29.0000    0.0    0.0  211.3375        S
1     1.0    male   0.9167    1.0    2.0  151.5500        S
2     1.0  female   2.0000    1.0    2.0  151.5500        S
3     1.0    male  30.0000    1.0    2.0  151.5500        S
4     1.0  female  25.0000    1.0    2.0  151.5500        S
```

[8]:
```
#Cantidad de data NaN en cada columna
df_titanic.isnull().sum()
```

[8]:
```
pclass        1
sex           1
age         264
sibsp         1
parch         1
```

```
fare          2
embarked      3
dtype: int64
```

[9]:
```
#porcentaje de columnas con valores iniciales
(df_titanic.isnull().sum()/ len(df_titanic))*100
```

[9]:
```
pclass       0.076336
sex          0.076336
age         20.152672
sibsp        0.076336
parch        0.076336
fare         0.152672
embarked     0.229008
dtype: float64
```

2.Eliminar filas que tengan menos de 3 valores reales

[10]:
```
df_titanic.dropna(thresh=2, inplace=True)
df_titanic
```

[10]:
```
      pclass     sex      age  sibsp  parch      fare embarked
0        1.0  female  29.0000    0.0    0.0  211.3375        S
1        1.0    male   0.9167    1.0    2.0  151.5500        S
2        1.0  female   2.0000    1.0    2.0  151.5500        S
3        1.0    male  30.0000    1.0    2.0  151.5500        S
4        1.0  female  25.0000    1.0    2.0  151.5500        S
...      ...     ...      ...    ...    ...       ...      ...
1304     3.0  female  14.5000    1.0    0.0   14.4542        C
1305     3.0  female      NaN    1.0    0.0   14.4542        C
1306     3.0    male  26.5000    0.0    0.0    7.2250        C
1307     3.0    male  27.0000    0.0    0.0    7.2250        C
1308     3.0    male  29.0000    0.0    0.0    7.8750        S

[1309 rows x 7 columns]
```

[11]:
```
#porcentaje de columnas con valores perdidos luego de thresh=2
(df_titanic.isnull().sum()/ len(df_titanic))*100
```

[11]:
```
pclass       0.000000
sex          0.000000
age         20.091673
sibsp        0.000000
parch        0.000000
fare         0.076394
embarked     0.152788
dtype: float64
```

3. Eliminar todas las filas con valor "nan" en la columna "age"

[12]:
```
df_titanic.dropna(subset=['age'], inplace=True)
df_titanic
```

[12]:

	pclass	sex	age	sibsp	parch	fare	embarked
0	1.0	female	29.0000	0.0	0.0	211.3375	S
1	1.0	male	0.9167	1.0	2.0	151.5500	S
2	1.0	female	2.0000	1.0	2.0	151.5500	S
3	1.0	male	30.0000	1.0	2.0	151.5500	S
4	1.0	female	25.0000	1.0	2.0	151.5500	S
...	...	...	...	...	...	...	...
1301	3.0	male	45.5000	0.0	0.0	7.2250	C
1304	3.0	female	14.5000	1.0	0.0	14.4542	C
1306	3.0	male	26.5000	0.0	0.0	7.2250	C
1307	3.0	male	27.0000	0.0	0.0	7.2250	C
1308	3.0	male	29.0000	0.0	0.0	7.8750	S

[1046 rows x 7 columns]

[13]:
```
#porcentaje de columnas con valores perdidos luego del subset['age']
(df_titanic.isnull().sum()/ len(df_titanic))*100
```

[13]:
```
pclass      0.000000
sex         0.000000
age         0.000000
sibsp       0.000000
parch       0.000000
fare        0.095602
embarked    0.191205
dtype: float64
```

4. Imputar la variable "fare" y la variable "embarked" teniendo en cuenta el tipo de variable.

Usaremos SimpleImputer

[25]:
```
from sklearn.impute import SimpleImputer
```

[26]:
```
#Para la variable Embarked usaremos most_Frequent por ser una variable object
```

[27]:
```
imp_moda = SimpleImputer(strategy="most_frequent")
```

[30]:
```
df_titanic_embarked= imp_moda.fit_transform(df_titanic[["embarked"]])
```

[32]:
```
df_titanic[["embarked"]]=df_titanic_embarked
```

[35]:
```
(df_titanic.isnull().sum()/len(df_titanic))*100
```

[35]:
```
pclass      0.000000
sex         0.000000
age         0.000000
sibsp       0.000000
parch       0.000000
fare        0.095602
embarked    0.000000
dtype: float64
```

[37]:
```
#Ahora a imputar la variable fare
df_titanic.fare
```

[37]:
```
0       211.3375
1       151.5500
2       151.5500
3       151.5500
4       151.5500
          ...
1301      7.2250
1304     14.4542
1306      7.2250
1307      7.2250
1308      7.8750
Name: fare, Length: 1046, dtype: float64
```

[38]:
```
imp_mean= SimpleImputer(strategy="mean")
```

[39]:
```
df_titanic_fare= imp_mean.fit_transform(df_titanic[["fare"]])
```

[40]:
```
df_titanic[["fare"]]=df_titanic_fare
```

[41]:
```
(df_titanic.isnull().sum()/len(df_titanic))*100
```

[41]:
```
pclass      0.0
sex         0.0
age         0.0
sibsp       0.0
parch       0.0
fare        0.0
embarked    0.0
dtype: float64
```

[ ]:
```

```