Titanic Dataset- Handling missing values

April 13, 2023

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import nbconvert
```

[2]:
```
df=sns.load_dataset('titanic')
```

[3]:
```
df.head()
```

[3]:
```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

[4]:
```
#Check missing value in dataset
df.isnull()
```

[4]:
```
     survived  pclass    sex    age  sibsp  parch   fare  embarked  class  \
0       False   False  False  False  False  False  False     False  False
1       False   False  False  False  False  False  False     False  False
2       False   False  False  False  False  False  False     False  False
3       False   False  False  False  False  False  False     False  False
4       False   False  False  False  False  False  False     False  False
..        ...     ...    ...    ...    ...    ...    ...       ...    ...
886     False   False  False  False  False  False  False     False  False
887     False   False  False  False  False  False  False     False  False
888     False   False  False   True  False  False  False     False  False
889     False   False  False  False  False  False  False     False  False
890     False   False  False  False  False  False  False     False  False
```

```
       who  adult_male   deck  embark_town  alive  alone
0    False       False   True        False  False  False
1    False       False  False        False  False  False
2    False       False   True        False  False  False
3    False       False  False        False  False  False
4    False       False   True        False  False  False
..     …           …     …           …      …      …
886  False       False   True        False  False  False
887  False       False  False        False  False  False
888  False       False   True        False  False  False
889  False       False  False        False  False  False
890  False       False   True        False  False  False

[891 rows x 15 columns]
```

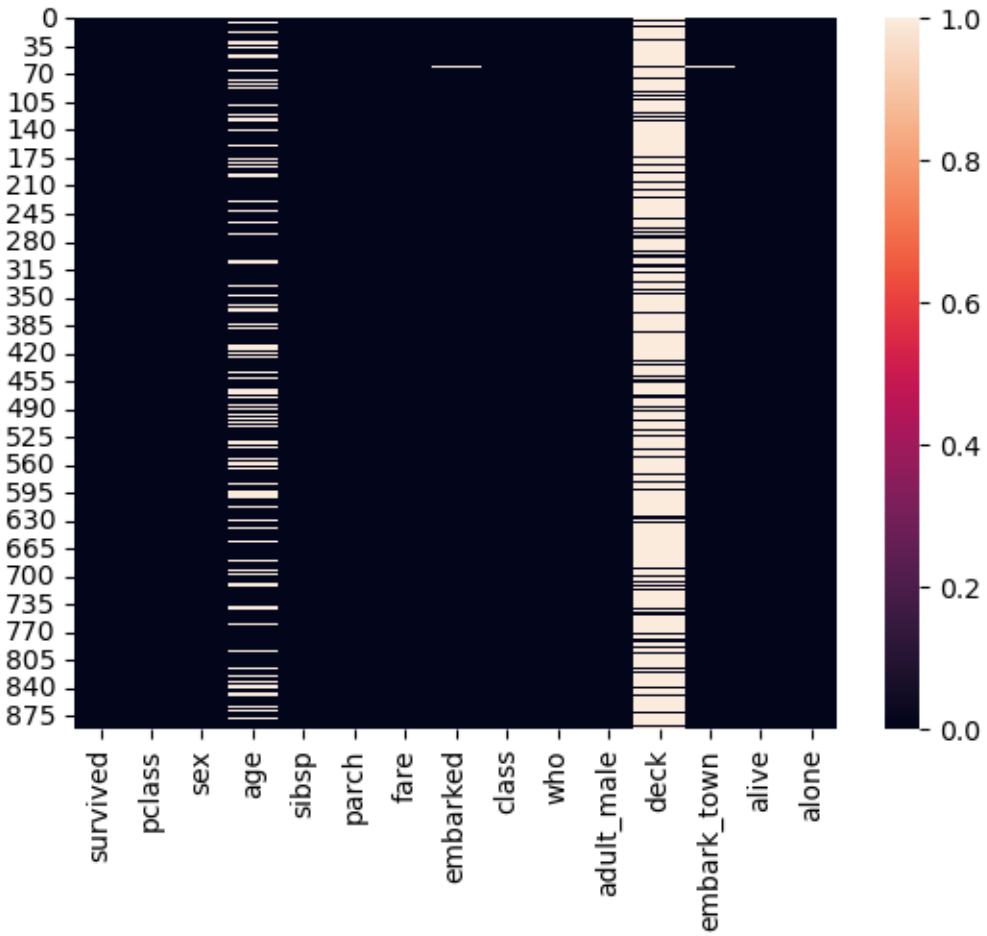
[5]:
```
df.isnull().sum()
```

[5]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

[6]:
```
# getting error here beacuse of (ValueError: could not convert string to float:␣
↪'male')
#sns.heatmap(df)
```

[7]:
```
sns.heatmap(df.isnull())
```

[7]:
```
<AxesSubplot:>
```



[8]:
```
#Handling missing values by deleting rows
df.dropna()
```

[8]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	\
1	1	1	female	38.0	1	0	71.2833	C	First	
3	1	1	female	35.0	1	0	53.1000	S	First	
6	0	1	male	54.0	0	0	51.8625	S	First	
10	1	3	female	4.0	1	1	16.7000	S	Third	
11	1	1	female	58.0	0	0	26.5500	S	First	
..	…	…	…	…	…	…	…	…	…	
871	1	1	female	47.0	1	1	52.5542	S	First	
872	0	1	male	33.0	0	0	5.0000	S	First	
879	1	1	female	56.0	0	1	83.1583	C	First	
887	1	1	female	19.0	0	0	30.0000	S	First	
889	1	1	male	26.0	0	0	30.0000	C	First	

	who	adult_male	deck	embark_town	alive	alone

```
1    woman       False    C    Cherbourg   yes  False
3    woman       False    C  Southampton   yes  False
6      man        True    E  Southampton    no   True
10   child       False    G  Southampton   yes  False
11   woman       False    C  Southampton   yes   True
..     …         …  …          …   …    …
871  woman       False    D  Southampton   yes  False
872    man        True    B  Southampton    no   True
879  woman       False    C    Cherbourg   yes  False
887  woman       False    B  Southampton   yes   True
889    man        True    C    Cherbourg   yes   True

[182 rows x 15 columns]
```

[9]:
```
##rowwise deletion
df.dropna().shape
```

[9]:
```
(182, 15)
```

[10]:
```
df.shape
```

[10]:
```
(891, 15)
```

1 Note-> Above practice is not good by deleting the rows, beacuse here we are missing lot of data

[11]:
```
## Handling missing values by deleting Column Wise
df.dropna(axis=1)
```

[11]:
```
     survived  pclass     sex  sibsp  parch     fare   class    who  \
0           0       3    male      1      0   7.2500   Third    man
1           1       1  female      1      0  71.2833   First  woman
2           1       3  female      0      0   7.9250   Third  woman
3           1       1  female      1      0  53.1000   First  woman
4           0       3    male      0      0   8.0500   Third    man
..        …     …      …    …     …      …     …    …
886         0       2    male      0      0  13.0000  Second    man
887         1       1  female      0      0  30.0000   First  woman
888         0       3  female      1      2  23.4500   Third  woman
889         1       1    male      0      0  30.0000   First    man
890         0       3    male      0      0   7.7500   Third    man

     adult_male alive  alone
0          True    no  False
1         False   yes  False
2         False   yes   True
```

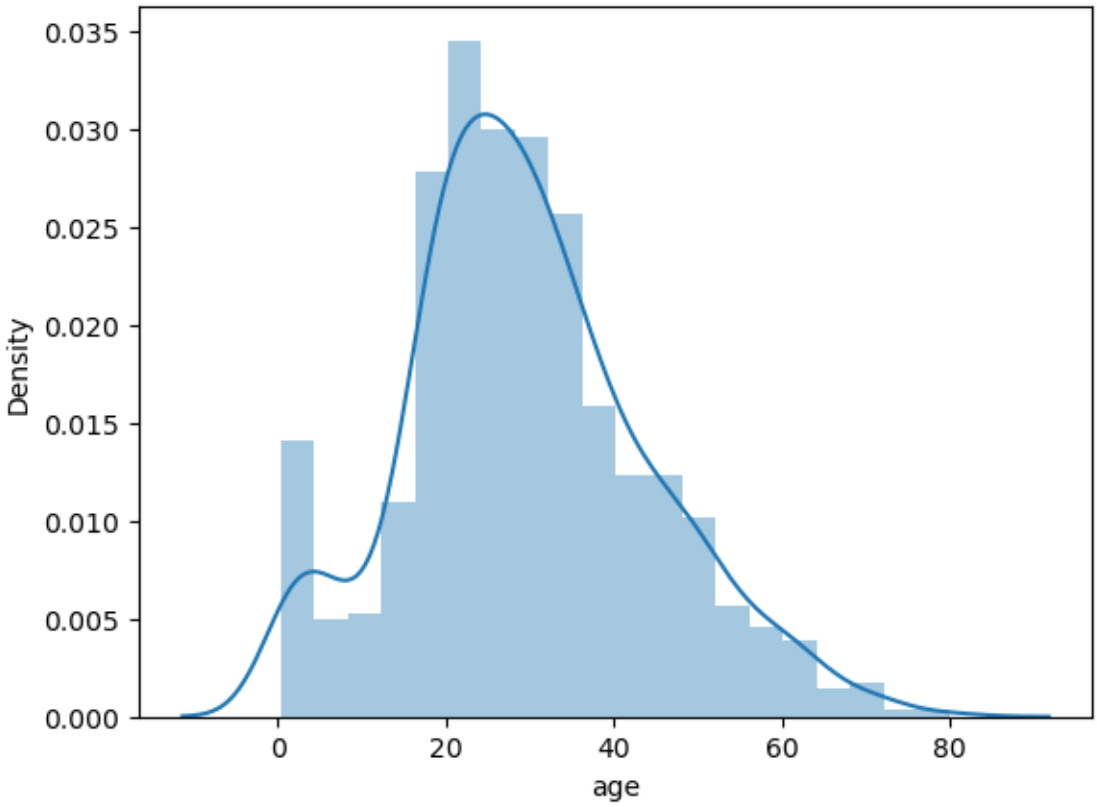
```
3          False   yes  False
4           True    no   True
..           …   …    …
886         True    no   True
887        False   yes   True
888        False    no  False
889         True   yes   True
890         True    no   True

[891 rows x 11 columns]
```

[12]:
```
## Imputation Technique
### 1-Mean value Imputation
```

[13]:
```
sns.distplot(df['age'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[13]: <AxesSubplot:xlabel='age', ylabel='Density'>



[14]:
```
df.age.isnull().sum()
```

[14]:
```
177
```

[15]:
```
df['Age_mean']=df['age'].fillna(df['age'].mean())
```

[16]:
```
df[['Age_mean','age']]
```

[16]:
```
      Age_mean   age
0    22.000000  22.0
1    38.000000  38.0
2    26.000000  26.0
3    35.000000  35.0
4    35.000000  35.0
..         …   …
886  27.000000  27.0
887  19.000000  19.0
888  29.699118   NaN
889  26.000000  26.0
890  32.000000  32.0

[891 rows x 2 columns]
```

[17]:
```
## Above techniques works when our data is normally distrubted.
```

2 Meadian Value Imputation-When data is skewed or- when we have outliers we used this technique

[18]:
```
df['Age_median']=df['age'].fillna(df['age'].median())
```

[19]:
```
df[['Age_mean','age','Age_median']]
```

[19]:
```
      Age_mean   age  Age_median
0    22.000000  22.0        22.0
1    38.000000  38.0        38.0
2    26.000000  26.0        26.0
3    35.000000  35.0        35.0
4    35.000000  35.0        35.0
..         …   …           …
886  27.000000  27.0        27.0
887  19.000000  19.0        19.0
888  29.699118   NaN        28.0
889  26.000000  26.0        26.0
890  32.000000  32.0        32.0
```

```
[891 rows x 3 columns]
```

3 3- Mode Value Imputation- Used for Categarial data

[20]:
```
df[df['embarked'].isnull()]
```

[20]:
```
     survived  pclass     sex   age  sibsp  parch  fare embarked  class  \
61          1       1  female  38.0      0      0  80.0      NaN  First
829         1       1  female  62.0      0      0  80.0      NaN  First

       who  adult_male deck embark_town alive  alone  Age_mean  Age_median
61   woman       False    B         NaN   yes   True      38.0        38.0
829  woman       False    B         NaN   yes   True      62.0        62.0
```

[21]:
```
df['embarked'].unique()
```

[21]:
```
array(['S', 'C', 'Q', nan], dtype=object)
```

[22]:
```
df['age'].notna()
```

[22]:
```
0       True
1       True
2       True
3       True
4       True
       …
886     True
887     True
888    False
889     True
890     True
Name: age, Length: 891, dtype: bool
```

[23]:
```
df[df['age'].notna()]
```

[23]:
```
     survived  pclass     sex   age  sibsp  parch     fare embarked   class  \
0           0       3    male  22.0      1      0   7.2500        S   Third
1           1       1  female  38.0      1      0  71.2833        C   First
2           1       3  female  26.0      0      0   7.9250        S   Third
3           1       1  female  35.0      1      0  53.1000        S   First
4           0       3    male  35.0      0      0   8.0500        S   Third
..        …     …      …   …    …     …    …       …      …     …
885         0       3  female  39.0      0      5  29.1250        Q   Third
886         0       2    male  27.0      0      0  13.0000        S  Second
887         1       1  female  19.0      0      0  30.0000        S   First
889         1       1    male  26.0      0      0  30.0000        C   First
```

```
890         0       3    male  32.0      0      0   7.7500        Q   Third

       who  adult_male deck  embark_town alive  alone  Age_mean  Age_median
0      man        True  NaN  Southampton    no  False      22.0        22.0
1    woman       False    C    Cherbourg   yes  False      38.0        38.0
2    woman       False  NaN  Southampton   yes   True      26.0        26.0
3    woman       False    C  Southampton   yes  False      35.0        35.0
4      man        True  NaN  Southampton    no   True      35.0        35.0
..     …          …   …          …    …    …       …         …
885  woman       False  NaN   Queenstown    no  False      39.0        39.0
886    man        True  NaN  Southampton    no   True      27.0        27.0
887  woman       False    B  Southampton   yes   True      19.0        19.0
889    man        True    C    Cherbourg   yes   True      26.0        26.0
890    man        True  NaN   Queenstown    no   True      32.0        32.0

[714 rows x 17 columns]
```

[24]:
```
df[df['age'].notna()]['embarked'].mode()
```

[24]:
```
0    S
Name: embarked, dtype: object
```

[25]:
```
df[df['age'].notna()]['embarked'].mode()[0]
```

[25]:
```
'S'
```

[26]:
```
mode= df[df['age'].notna()]['embarked'].mode()[0]
```

[27]:
```
mode
```

[27]:
```
'S'
```

[28]:
```
df['embarked_mode']=df['embarked'].fillna(mode)
```

[29]:
```
df[['embarked_mode','embarked']]
```

[29]:
```
    embarked_mode embarked
0               S        S
1               C        C
2               S        S
3               S        S
4               S        S
..            …      …
886             S        S
887             S        S
888             S        S
889             C        C
890             Q        Q
```

```
[891 rows x 2 columns]
```

[30]:
```
df['embarked_mode'].isnull().sum()
```

[30]:
```
0
```

[ ]:
```

```