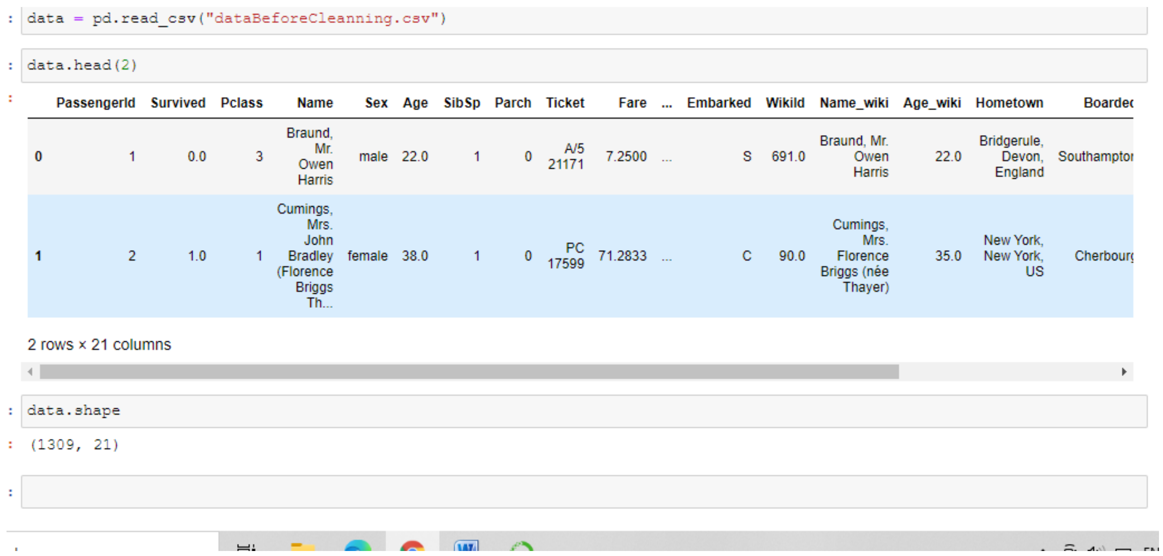
Data_Wrangle_Report

This projects is about titanic dataset which contain information about the passengers who was on it while its sank for more details look at

https://en.wikipedia.org/wiki/Passengers_of_the_RMS_Titanic

(you can gathered it from https://www.datacamp.com/community/tutorials/k-means-clustering-python).

The original data set contains 1039 rows with 21 attributes(columns)



```
: data = pd.read_csv("dataBeforeCleanning.csv")

: data.head(2)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	...	Embarked	WikiId	Name_wiki	Age_wiki	Hometown	Boardec
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	...	S	691.0	Braund, Mr. Owen Harris	22.0	Bridgerule, Devon, England	Southamptor
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	...	C	90.0	Cumings, Mrs. Florence Briggs (née Thayer)	35.0	New York, New York, US	Cherbour

2 rows × 21 columns

```
: data.shape
: (1309, 21)
```

Now, we will discuss about Cleaning :

```
In [1]: import numpy as np
        import pandas as pd
```

1st I we read our data.

```
data = pd.read_csv("dataBeforeCleanning.csv")
```

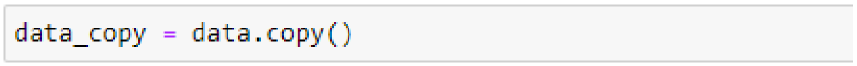
Assessment & issues:

tidy issue: the column name contain two values first value is surname and second value is full name

To solve this issue we must split name column into 2 columns

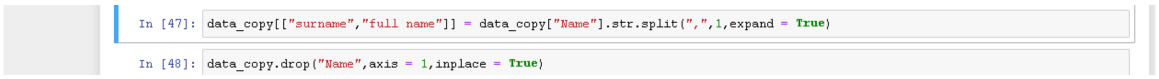
1st column is surname , 2nd column is full name

But first before starting data preprocessing operations we must take a copy of this data



```
data_copy = data.copy()
```

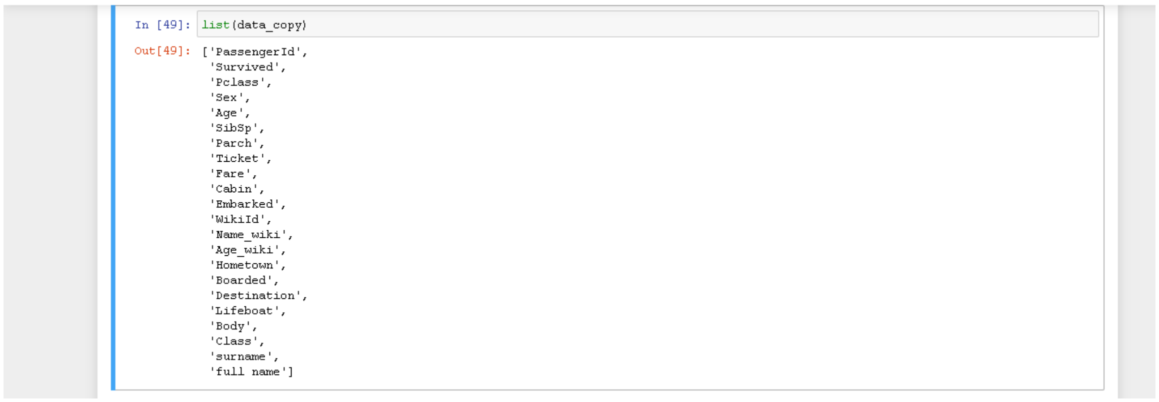
Now, we split the data column then remove it.



```
In [47]: data_copy[["surname","full name"]] = data_copy["Name"].str.split(",",1,expand = True)

In [48]: data_copy.drop("Name",axis = 1,inplace = True)
```

output after splitting this column into 2 columns:



```
In [49]: list(data_copy)

Out[49]: ['PassengerId',
          'Survived',
          'Pclass',
          'Sex',
          'Age',
          'SibSp',
          'Parch',
          'Ticket',
          'Fare',
          'Cabin',
          'Embarked',
          'WikiId',
          'Name_wiki',
          'Age_wiki',
          'Hometown',
          'Boarded',
          'Destination',
          'Lifeboat',
          'Body',
          'Class',
          'surname',
          'full name']
```

Quality issues:

Second issue: there are many duplicates and redundancy columns like:

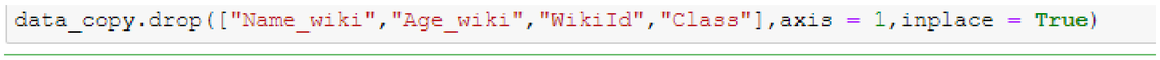
Age and Age_Wiki

Name and Name Wiki

Passengers Id and Wiki Id

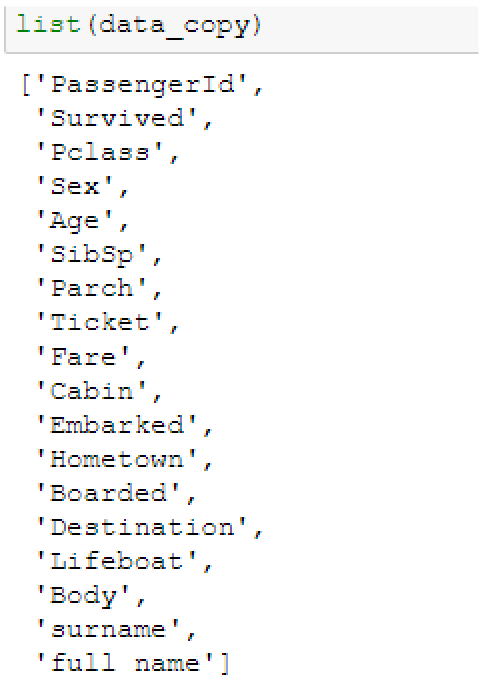
Solution:

we should drop the duplicates columns:



```
data_copy.drop(["Name_wiki","Age_wiki","WikiId","Class"],axis = 1,inplace = True)
```

the output of new lists after dropping the duplicates columns:

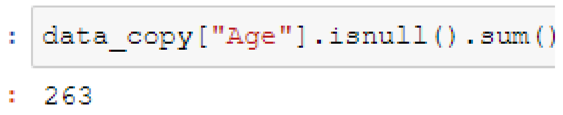


```
list(data_copy)
```

```
['PassengerId',
 'Survived',
 'Pclass',
 'Sex',
 'Age',
 'SibSp',
 'Parch',
 'Ticket',
 'Fare',
 'Cabin',
 'Embarked',
 'Hometown',
 'Boarded',
 'Destination',
 'Lifeboat',
 'Body',
 'surname',
 'full name']
```

Third Issue:

There are missing values in Age column

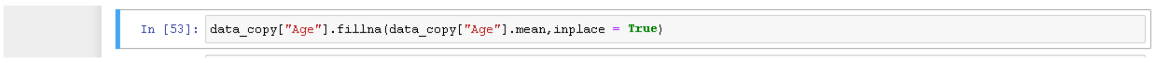


```
: data_copy["Age"].isnull().sum()

: 263
```

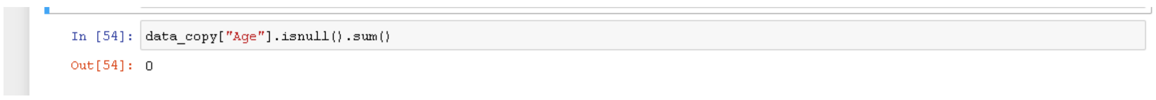
Solution:

Since the age maybe an effective parameter we should fill the missing value with the column's mean



```
In [53]: data_copy["Age"].fillna(data_copy["Age"].mean,inplace = True)
```

then after doing that we find the missing values in this column is 0

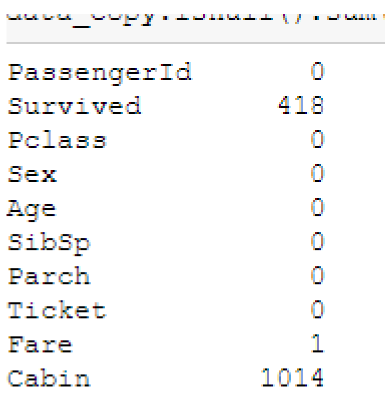


```
In [54]: data_copy["Age"].isnull().sum()

Out[54]: 0
```

Fourth Issue:

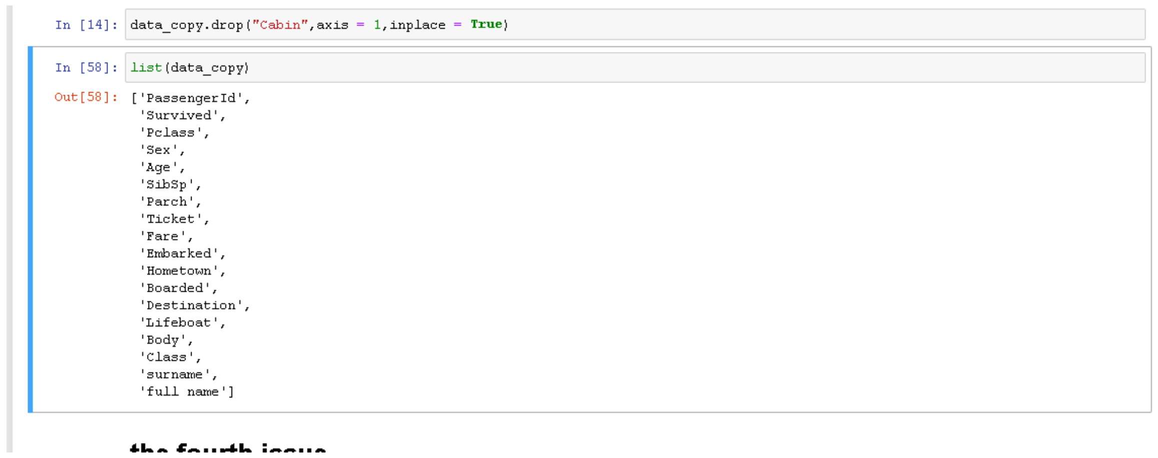
most of cabin column is missing data



```
data_copy.isnull().sum(
PassengerId        0
Survived         418
Pclass             0
Sex                0
Age                0
SibSp              0
Parch              0
Ticket             0
Fare               1
Cabin           1014
```

Solution:

Since it is not important feature we can drop cabin column , then the new list will be:



```
In [14]: data_copy.drop("Cabin",axis = 1,inplace = True)

In [58]: list(data_copy)

Out[58]: ['PassengerId',
          'Survived',
          'Pclass',
          'Sex',
          'Age',
          'SibSp',
          'Parch',
          'Ticket',
          'Fare',
          'Embarked',
          'Hometown',
          'Boarded',
          'Destination',
          'Lifeboat',
          'Body',
          'Class',
          'surname',
          'full name']
```

the fourth issue

Fifth Issue:

There are missing values in lifeboat & body columns:

Solution:

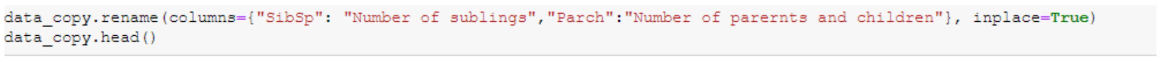
Drop lifeboat and body columns:

Sixth Issue:

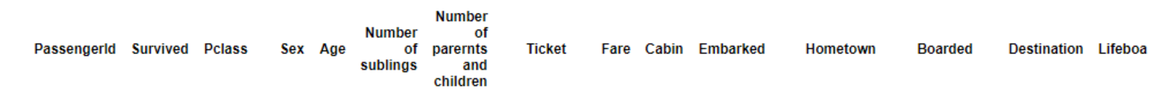
Sibsp and Parch column Names are not interpretable

Solution:

Rename SibiSip column with Number of Siblings and parch to number of parents and children



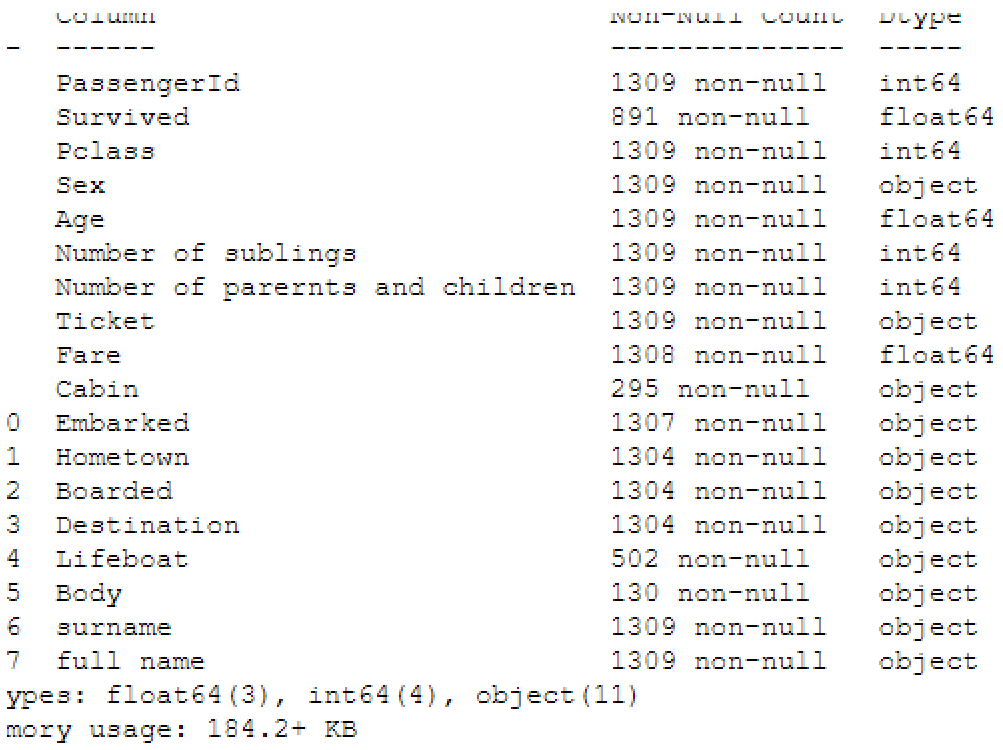
```
data_copy.rename(columns={"SibSp": "Number of sublings","Parch":"Number of parernts and children"}, inplace=True)
data_copy.head()
```



PassengerId	Survived	Pclass	Sex	Age	Number of sublings	Number of parernts and children	Ticket	Fare	Cabin	Embarked	Hometown	Boarded	Destination	Lifeboa

Seventh issue:

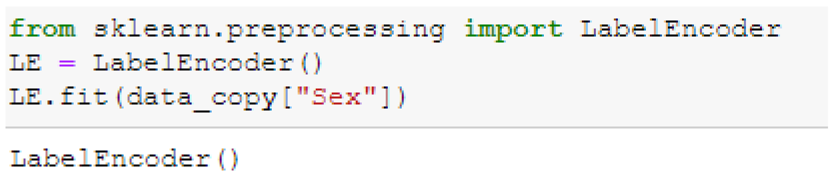
data has columns which contain string data and k mean clustering can only handling numeric data



```
   Column                           Non-Null Count  Dtype
-  ------                           --------------  -----
   PassengerId                      1309 non-null   int64
   Survived                         891 non-null    float64
   Pclass                           1309 non-null   int64
   Sex                              1309 non-null   object
   Age                              1309 non-null   float64
   Number of sublings               1309 non-null   int64
   Number of parernts and children  1309 non-null   int64
   Ticket                           1309 non-null   object
   Fare                             1308 non-null   float64
   Cabin                            295 non-null    object
0  Embarked                         1307 non-null   object
1  Hometown                         1304 non-null   object
2  Boarded                          1304 non-null   object
3  Destination                      1304 non-null   object
4  Lifeboat                         502 non-null    object
5  Body                             130 non-null    object
6  surname                          1309 non-null   object
7  full name                        1309 non-null   object
ypes: float64(3), int64(4), object(11)
mory usage: 184.2+ KB
```

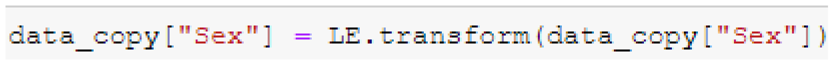
Solution:

We will convert columns that what we will use in the model into numeric using Label Encoder class from sklearn library we will need only “Sex” column so we will encode this column and drop other string columns.



```
from sklearn.preprocessing import LabelEncoder
LE = LabelEncoder()
LE.fit(data_copy["Sex"])
```

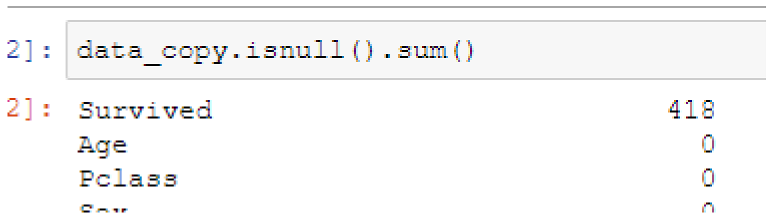
```
LabelEncoder()
```



```
data_copy["Sex"] = LE.transform(data_copy["Sex"])
```

Seventh Issue:

Survived column has many missing values

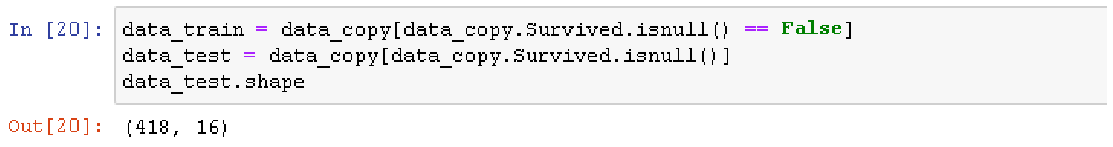


```
2]: data_copy.isnull().sum()

2]: Survived                    418
    Age                           0
    Pclass                        0
```

Solution:

Since we have done must of the data cleaning work and we choiced the columns that we will need we will separate our data into two separated files first file will contain our cleaned data without raws which contain null values and we will use this data to train our clustering model to predict the missing values in Survived column in the second file(test_data)



```
In [20]: data_train = data_copy[data_copy.Survived.isnull() == False]
         data_test = data_copy[data_copy.Survived.isnull()]
         data_test.shape

Out[20]: (418, 16)
```