Statistical Machine Learning Approaches to Liver Disease Prediction

Team ID: PNT2022TMID26382

Team Leader : Madhulika I

Team Member : Rithika AM

Team Member : Kavya A P

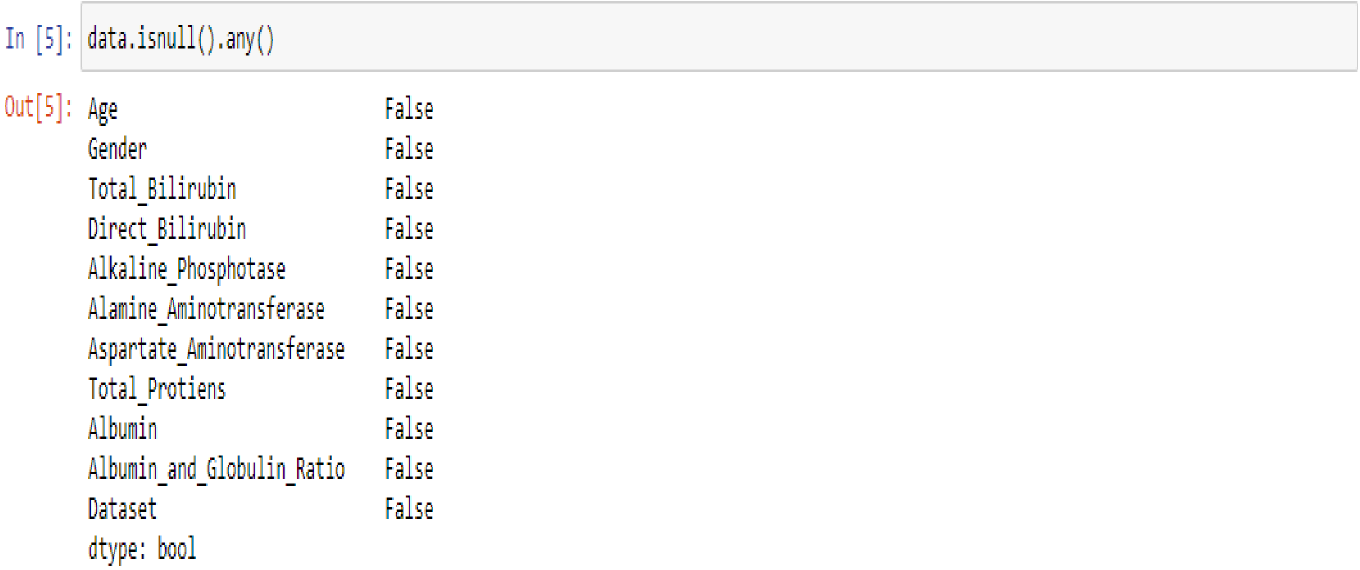
Team Member : Kirthiga S

Checking for null values and Handling Null Values:

This method commonly used to handle the null values. Here, we either delete a particular row if it has a null value for a particular feature and a particular column if it has more than 70-75% of missing values. This method is advised only when there are enough samples in the data set. Missing values can be handled by deleting the rows or columns having null values. If columns have more than half of the rows as null then the entire column can be dropped. The rows which are having one or more columns values as null can also be dropped

We will be using isnull().any() method to see which column has missing values.

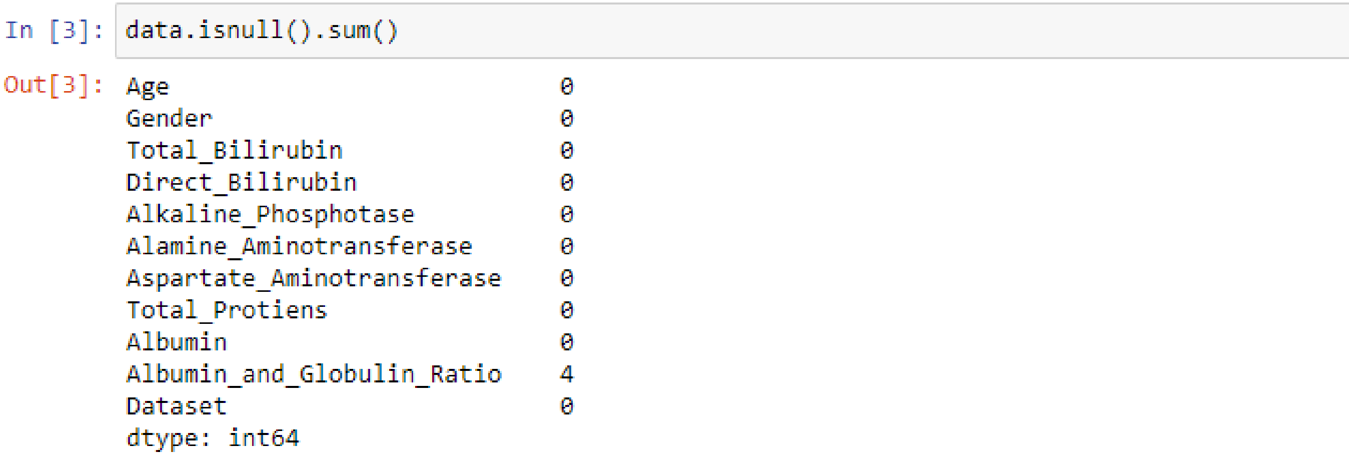
- This isnull().any() method return two values, False and True.
- False return that Column has No Null Values.
- True return that Column has Null values.



```
In [5]: data.isnull().any()
```

```
Out[5]: Age                           False
        Gender                        False
        Total_Bilirubin               False
        Direct_Bilirubin              False
        Alkaline_Phosphotase          False
        Alamine_Aminotransferase      False
        Aspartate_Aminotransferase    False
        Total_Protiens                False
        Albumin                       False
        Albumin_and_Globulin_Ratio    False
        Dataset                       False
        dtype: bool
```

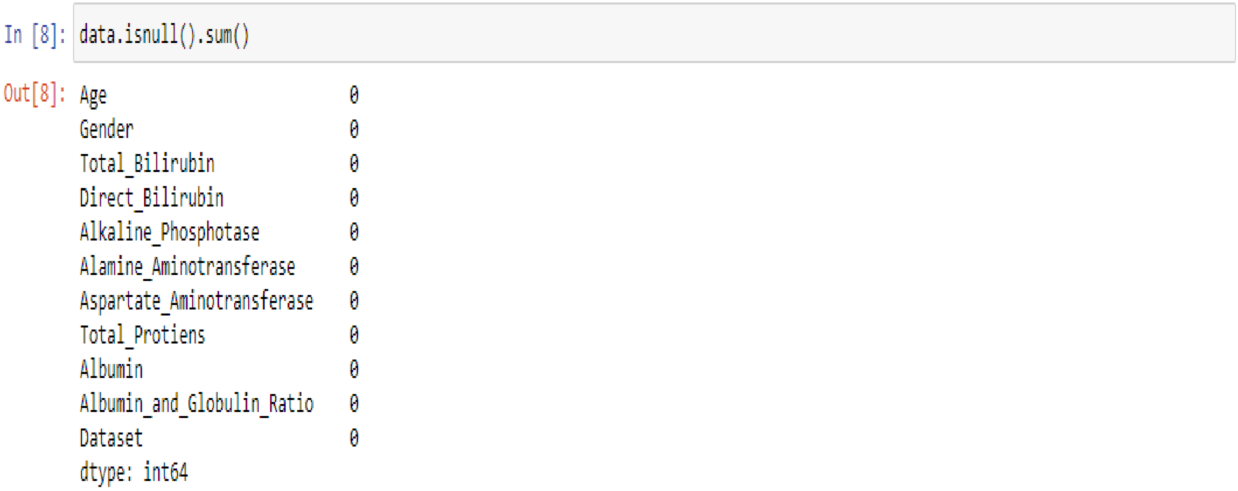
Let us check how many numbers of null records present in the Closing Value column using sum() function.



```
In [3]: data.isnull().sum()

Out[3]: Age                           0
        Gender                        0
        Total_Bilirubin               0
        Direct_Bilirubin              0
        Alkaline_Phosphotase          0
        Alamine_Aminotransferase      0
        Aspartate_Aminotransferase    0
        Total_Protiens                0
        Albumin                       0
        Albumin_and_Globulin_Ratio    4
        Dataset                       0
        dtype: int64
```

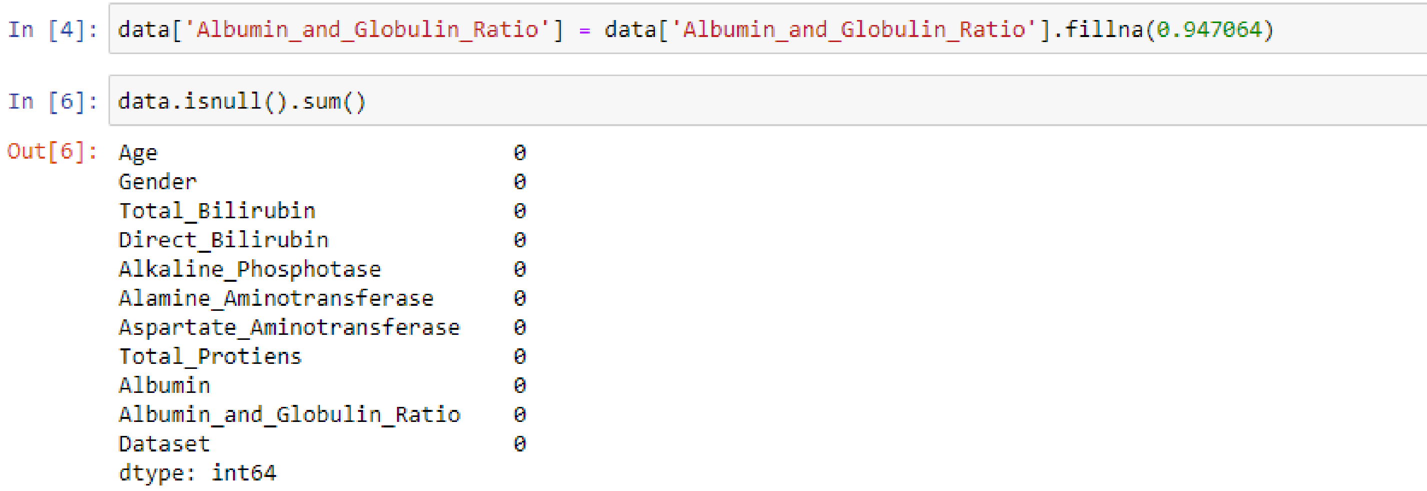
We can notice that, there are 4 null values are there in the column Albumin_and_Globulin_Ratio.



```
In [8]: data.isnull().sum()

Out[8]: Age                           0
        Gender                        0
        Total_Bilirubin               0
        Direct_Bilirubin              0
        Alkaline_Phosphotase          0
        Alamine_Aminotransferase      0
        Aspartate_Aminotransferase    0
        Total_Protiens                0
        Albumin                       0
        Albumin_and_Globulin_Ratio    0
        Dataset                       0
        dtype: int64
```

Now will handle or fill that null values with the help of fillna() method.



```
In [4]: data['Albumin_and_Globulin_Ratio'] = data['Albumin_and_Globulin_Ratio'].fillna(0.947064)

In [6]: data.isnull().sum()

Out[6]: Age                           0
        Gender                        0
        Total_Bilirubin               0
        Direct_Bilirubin              0
        Alkaline_Phosphotase          0
        Alamine_Aminotransferase      0
        Aspartate_Aminotransferase    0
        Total_Protiens                0
        Albumin                       0
        Albumin_and_Globulin_Ratio    0
        Dataset                       0
        dtype: int64
```

In real world data, there are some instances where a particular element is absent because of various reasons, such as, corrupt data, failure to load the information, or incomplete extraction. Handling the missing values is one of the greatest challenges faced by analysts, because making the right decision on how to handle it generates robust data models. Let us look at different ways of imputing the missing values.