Hands_on_Handling_missing_value_with_List_&_Pairwise_Deletion_with

September 1, 2023

0.1 Handling Missing Value in Machine Learning

0.1.1 Finding the missing values in Dataset

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

[2]:
```
# Import the dataset
df_titanic = pd.read_csv("https://raw.githubusercontent.com/atulpatelDS/
 ↪Data_Files/master/Titanic/titanic_train.csv")
```

[3]:
```
df_titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

[4]:
```
df_titanic.describe(include=['O'])  ## For categorical
```

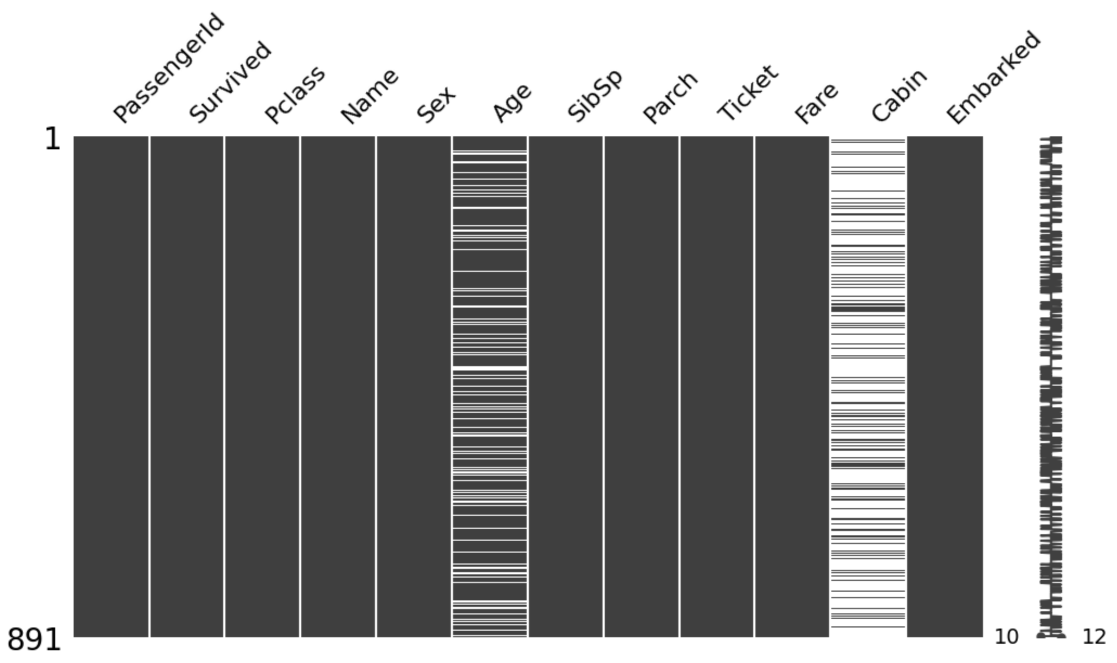
[4]:
```
                            Name   Sex  Ticket    Cabin Embarked
count                        891   891     891      204      889
unique                       891     2     681      147        3
top     Braund, Mr. Owen Harris  male  347082  B96 B98        S
freq                           1   577       7        4      644
```

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

[5]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

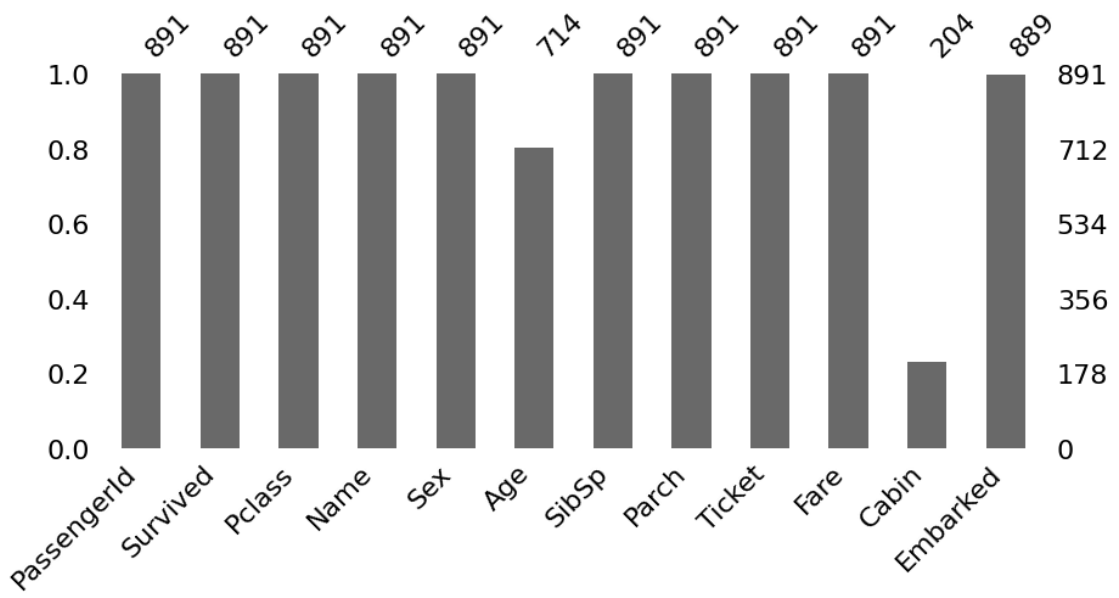
[7]:
```
# Program to visualize missing values in dataset
# Importing the libraries
import missingno as msno
# Visualize missing values as a matrix
msno.matrix(df_titanic,figsize=(12,6))
```

[7]:
```
<Axes: >
```



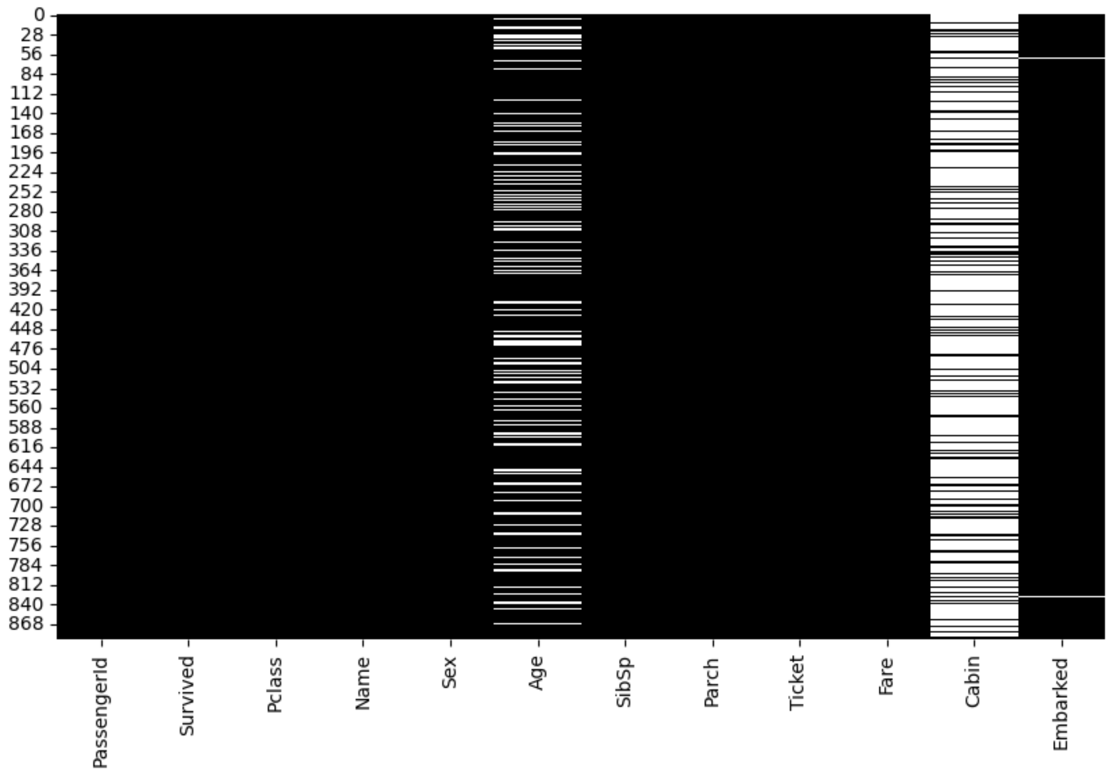
```
[8]: # Visualize the number of missing
     # values as a bar chart
     msno.bar(df_titanic,figsize=(10,4))
```

```
[8]: <Axes: >
```



[9]:
```
#sns.heatmap(df_titanic.isnull(),cmap="viridis")
plt.figure(figsize=(10,6))
sns.heatmap(
    data=df_titanic.isnull(),
    cmap=sns.color_palette(['black', 'yellow', 'orange', 'white']),cbar=False)
```

[9]:
```
<Axes: >
```



It is cleary visible that most of the null values are available in column AGE and CABIN

[10]:
```
# lets find out the percentage of missssing vale in each column
Percent_Missing_Value = df_titanic.isnull().sum()*100/len(df_titanic)
Percent_Missing_Value
```

[10]:
```
PassengerId     0.000000
Survived        0.000000
Pclass          0.000000
Name            0.000000
Sex             0.000000
Age            19.865320
SibSp           0.000000
Parch           0.000000
Ticket          0.000000
```

```
Fare             0.000000
Cabin           77.104377
Embarked         0.224467
dtype: float64
```

0.2 Method 1- Removal or Deletion of missing value

List wise Deletion of Missing Value

- Use dropna(), drop() functions

[11]:
```
# Import the dataset
df_titanic = pd.read_csv("https://raw.githubusercontent.com/atulpatelDS/
 ↪Data_Files/master/Titanic/titanic_train.csv")
```

[12]:
```
df_titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

[13]:
```
df_titanic.isnull().sum()
```

[13]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
```

```
Cabin          687
Embarked         2
dtype: int64
```

[14]:
```python
## Total no of samples where any column has NaN
df_titanic[(df_titanic["Age"].isnull()==True)|
           (df_titanic["Cabin"].isnull()==True)|
           (df_titanic["Embarked"].isnull()==True)]
```

[14]:
```
     PassengerId  Survived  Pclass                                         Name  \
0              1         0       3                      Braund, Mr. Owen Harris
2              3         1       3                       Heikkinen, Miss. Laina
4              5         0       3                     Allen, Mr. William Henry
5              6         0       3                             Moran, Mr. James
7              8         0       3               Palsson, Master. Gosta Leonard
..           …         …       …                                            …
884          885         0       3                        Sutehall, Mr. Henry Jr
885          886         0       3         Rice, Mrs. William (Margaret Norton)
886          887         0       2                         Montvila, Rev. Juozas
888          889         0       3     Johnston, Miss. Catherine Helen "Carrie"
890          891         0       3                           Dooley, Mr. Patrick

        Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
0      male  22.0      1      0         A/5 21171   7.2500   NaN        S
2    female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
4      male  35.0      0      0            373450   8.0500   NaN        S
5      male   NaN      0      0            330877   8.4583   NaN        Q
7      male   2.0      3      1            349909  21.0750   NaN        S
..      …     …      …      …               …      …   …      …
884    male  25.0      0      0   SOTON/OQ 392076   7.0500   NaN        S
885  female  39.0      0      5            382652  29.1250   NaN        Q
886    male  27.0      0      0            211536  13.0000   NaN        S
888  female   NaN      1      2        W./C. 6607  23.4500   NaN        S
890    male  32.0      0      0            370376   7.7500   NaN        Q

[708 rows x 12 columns]
```

[15]:
```python
df_titanic.shape
```

[15]:
```
(891, 12)
```

[16]:
```python
## Lets delete all the Rows where we have NaN values
df_titanic_new = df_titanic.dropna()
```

[17]:
```python
df_titanic_new.shape
```

[17]:
```
(183, 12)
```

[18]:
```
891-708
```

[18]:
```
183
```

We have total 891 samples and 708 out of 891 have the NaN value.If we delete all the rows where we have NaN value then we can only get sample size 183, which is very less value to make the model so we cannot delete all the rows.

[19]:
```
df_titanic_new[(df_titanic_new["Age"].isnull()==True)|
               (df_titanic_new["Cabin"].isnull()==True)|
               (df_titanic_new["Embarked"].isnull()==True)]
```

[19]:
```
Empty DataFrame
Columns: [PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket,
Fare, Cabin, Embarked]
Index: []
```

[20]:
```
df_titanic_new .info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 183 entries, 1 to 889
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  183 non-null    int64
 1   Survived     183 non-null    int64
 2   Pclass       183 non-null    int64
 3   Name         183 non-null    object
 4   Sex          183 non-null    object
 5   Age          183 non-null    float64
 6   SibSp        183 non-null    int64
 7   Parch        183 non-null    int64
 8   Ticket       183 non-null    object
 9   Fare         183 non-null    float64
 10  Cabin        183 non-null    object
 11  Embarked     183 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 18.6+ KB
```

[21]:
```
df_titanic[(df_titanic["Age"].isnull()==True)&
               (df_titanic["Cabin"].isnull()==True)
               ]
```

[21]:
```
    PassengerId  Survived  Pclass                            Name  \
5             6         0       3                Moran, Mr. James
17           18         1       2    Williams, Mr. Charles Eugene
19           20         1       3         Masselmani, Mrs. Fatima
26           27         0       3           Emir, Mr. Farred Chehab
```

```
28             29         1       3          O'Dwyer, Miss. Ellen "Nellie"
..            …         …      …                                     …
859           860         0       3                           Razi, Mr. Raihed
863           864         0       3        Sage, Miss. Dorothy Edith "Dolly"
868           869         0       3              van Melkebeke, Mr. Philemon
878           879         0       3                         Laleff, Mr. Kristo
888           889         0       3  Johnston, Miss. Catherine Helen "Carrie"

        Sex  Age  SibSp  Parch      Ticket     Fare Cabin Embarked
5      male  NaN      0      0      330877   8.4583   NaN        Q
17     male  NaN      0      0      244373  13.0000   NaN        S
19   female  NaN      0      0        2649   7.2250   NaN        C
26     male  NaN      0      0        2631   7.2250   NaN        C
28   female  NaN      0      0      330959   7.8792   NaN        Q
..      …   …     …     …        …       …   …       …
859    male  NaN      0      0        2629   7.2292   NaN        C
863  female  NaN      8      2    CA. 2343  69.5500   NaN        S
868    male  NaN      0      0      345777   9.5000   NaN        S
878    male  NaN      0      0      349217   7.8958   NaN        S
888  female  NaN      1      2  W./C. 6607  23.4500   NaN        S

[158 rows x 12 columns]
```

We can see there are 158 Rows where we have NaN value in both Age and Cabin in same row.We can delete these rows.

[22]:
```python
# lets drop those rows where both Age and Cabin are null
df_titanic_new1 = df_titanic.drop(df_titanic.index[(df_titanic["Age"].
 ↪isnull()==True)&
               (df_titanic["Cabin"].isnull()==True)
              ])
```

[23]:
```python
df_titanic_new1
```

[23]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …       …     …
885          886         0       3
886          887         0       2
887          888         1       1
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 …     …     …      …
885               Rice, Mrs. William (Margaret Norton)  female  39.0      0
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …       …     …      …
885      5            382652  29.1250   NaN        Q
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[733 rows x 12 columns]
```

[24]:
```
891-158
```

[24]:
```
733
```

[25]:
```
# If we want to drop all those rows where we have all NaN value in all columns␣
↪in a row.
df_titanic_new3 = df_titanic.dropna(how="all")
```

[26]:
```
df_titanic_new3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

As we don't have any row where all the value in NaN. So It did not delete any row.

[27]:
```
# Import Dataset
import pandas as pd
df_sales = pd.read_excel("https://raw.githubusercontent.com/atulpatelDS/
 ↪Data_Files/master/Feature_Engineering/Missing_Value/Sales.xlsx")
```

[28]:
```
df_sales
```

[28]:
```
         Date  Day_Temp  No_of_Customers   Sales
0  2020-10-01      30.0            100.0  3112.0
1  2020-10-02       NaN            115.0  3682.0
2  2020-10-03      31.0              NaN  2774.0
3  2020-10-04      29.0            105.0  3182.0
4  2020-10-05      33.0            104.0  1368.0
5  2020-10-07       NaN              NaN     NaN
6  2020-11-24      26.0             90.0  4232.0
7  2020-11-25       NaN             96.0     NaN
8  2020-11-26      27.0            100.0  2356.0
9  2020-11-28       NaN              NaN     NaN
10 2020-11-29      23.0             94.0  1254.0
11 2020-11-30      22.0             91.0  4232.0
```

[29]:
```
df_sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Date             12 non-null     datetime64[ns]
 1   Day_Temp         8 non-null      float64
 2   No_of_Customers  9 non-null      float64
 3   Sales            9 non-null      float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 512.0 bytes
```

[30]:
```
## Set Date as Index
df_sales.set_index("Date",inplace=True)
```

[31]:
```
df_sales
```

[31]:
```
            Day_Temp  No_of_Customers   Sales
Date
2020-10-01      30.0            100.0  3112.0
2020-10-02       NaN            115.0  3682.0
2020-10-03      31.0              NaN  2774.0
2020-10-04      29.0            105.0  3182.0
2020-10-05      33.0            104.0  1368.0
2020-10-07       NaN              NaN     NaN
2020-11-24      26.0             90.0  4232.0
2020-11-25       NaN             96.0     NaN
2020-11-26      27.0            100.0  2356.0
2020-11-28       NaN              NaN     NaN
2020-11-29      23.0             94.0  1254.0
2020-11-30      22.0             91.0  4232.0
```

[32]:
```
## Lets Delete row where atlest one NaN value
df_sales_1 = df_sales.dropna()
```

[33]:
```
df_sales_1
```

[33]:
```
            Day_Temp  No_of_Customers   Sales
Date
2020-10-01      30.0            100.0  3112.0
2020-10-04      29.0            105.0  3182.0
2020-10-05      33.0            104.0  1368.0
2020-11-24      26.0             90.0  4232.0
2020-11-26      27.0            100.0  2356.0
2020-11-29      23.0             94.0  1254.0
2020-11-30      22.0             91.0  4232.0
```

[34]:
```
## If we want to delete Row where all rows data are NaN.
df_sales_2 = df_sales.dropna(how="all")
```

[35]:
```
df_sales_2
```

[35]:
```
            Day_Temp  No_of_Customers   Sales
Date
2020-10-01      30.0            100.0  3112.0
2020-10-02       NaN            115.0  3682.0
2020-10-03      31.0              NaN  2774.0
2020-10-04      29.0            105.0  3182.0
2020-10-05      33.0            104.0  1368.0
2020-11-24      26.0             90.0  4232.0
2020-11-25       NaN             96.0     NaN
2020-11-26      27.0            100.0  2356.0
2020-11-29      23.0             94.0  1254.0
```

```
2020-11-30      22.0               91.0  4232.0
```

[36]:
```
# If we want to keep only those rows where we have atleast one valid value.
# thresh : int, optional : Require that many non-NA values.
df_sales_3 = df_sales.dropna(thresh=1)
df_sales_3
```

[36]:
```
            Day_Temp  No_of_Customers   Sales
Date
2020-10-01      30.0            100.0  3112.0
2020-10-02       NaN            115.0  3682.0
2020-10-03      31.0              NaN  2774.0
2020-10-04      29.0            105.0  3182.0
2020-10-05      33.0            104.0  1368.0
2020-11-24      26.0             90.0  4232.0
2020-11-25       NaN             96.0     NaN
2020-11-26      27.0            100.0  2356.0
2020-11-29      23.0             94.0  1254.0
2020-11-30      22.0             91.0  4232.0
```

[37]:
```
# If we want to keep only those rows where we have atleast two valid value.
# thresh : int, optional : Require that many non-NA values.
df_sales_4 = df_sales.dropna(thresh=2)
df_sales_4
```

[37]:
```
            Day_Temp  No_of_Customers   Sales
Date
2020-10-01      30.0            100.0  3112.0
2020-10-02       NaN            115.0  3682.0
2020-10-03      31.0              NaN  2774.0
2020-10-04      29.0            105.0  3182.0
2020-10-05      33.0            104.0  1368.0
2020-11-24      26.0             90.0  4232.0
2020-11-26      27.0            100.0  2356.0
2020-11-29      23.0             94.0  1254.0
2020-11-30      22.0             91.0  4232.0
```

[38]:
```
# If we want to keep only those rows where we have atleast three valid value.
# thresh : int, optional : Require that many non-NA values.
df_sales_5 = df_sales.dropna(thresh=3)
df_sales_5
```

[38]:
```
            Day_Temp  No_of_Customers   Sales
Date
2020-10-01      30.0            100.0  3112.0
2020-10-04      29.0            105.0  3182.0
2020-10-05      33.0            104.0  1368.0
```

```
2020-11-24       26.0             90.0  4232.0
2020-11-26       27.0            100.0  2356.0
2020-11-29       23.0             94.0  1254.0
2020-11-30       22.0             91.0  4232.0
```

Pair wise Deletion of Missing Value

[39]:
```
# Lets Delete the NaN value column wise.
# lets find out the percentage of misssing vale in each column
Percent_Missing_Value = df_titanic.isnull().sum()*100/len(df_titanic)
Percent_Missing_Value
```

[39]:
```
PassengerId     0.000000
Survived        0.000000
Pclass          0.000000
Name            0.000000
Sex             0.000000
Age            19.865320
SibSp           0.000000
Parch           0.000000
Ticket          0.000000
Fare            0.000000
Cabin          77.104377
Embarked        0.224467
dtype: float64
```

[40]:
```
df_titanic.head()
```

[40]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

We can clearly see that only 20% of the AGE data is missing. The proportion of the AGE missing is likely small enough for resonable replacement with some form of imputation. Now see the CABIN column , It looks like we are missing too much of that data to do something usefull with at a basic level. We will drop it.

If columns have more than half of rows as null then the entire column can be dropped.

[41]:
```
# Lets delete the Cabin column
df_titanic_new2 = df_titanic.drop("Cabin",axis=1)
```

[42]:
```
df_titanic_new2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(4)
memory usage: 76.7+ KB
```

[43]:
```
# Import Dataset
import pandas as pd
df_saless = pd.read_excel("https://raw.githubusercontent.com/atulpatelDS/
↪Data_Files/master/Feature_Engineering/Missing_Value/Saless.xlsx")
```

[44]:
```
df_saless
```

[44]:
```
        Date  Store_Type  City_Type  Day_Temp  No_of_Customers   Sales  \
0 2020-10-01           1          1      30.0            100.0  3112.0
1 2020-10-02           2          1      32.0            115.0  3682.0
2 2020-10-03           3          3      31.0              NaN  2774.0
3 2020-10-04           1          2      29.0            105.0  3182.0
4 2020-10-05           1          2      33.0            104.0  1368.0
5 2020-10-07           2          2       NaN              NaN     NaN
6 2020-11-24           2          3      26.0             90.0  4232.0
7 2020-11-25           3          3       NaN             96.0     NaN
8 2020-11-26           2          2      27.0            100.0  2356.0
9 2020-11-28           3          1       NaN              NaN     NaN
```

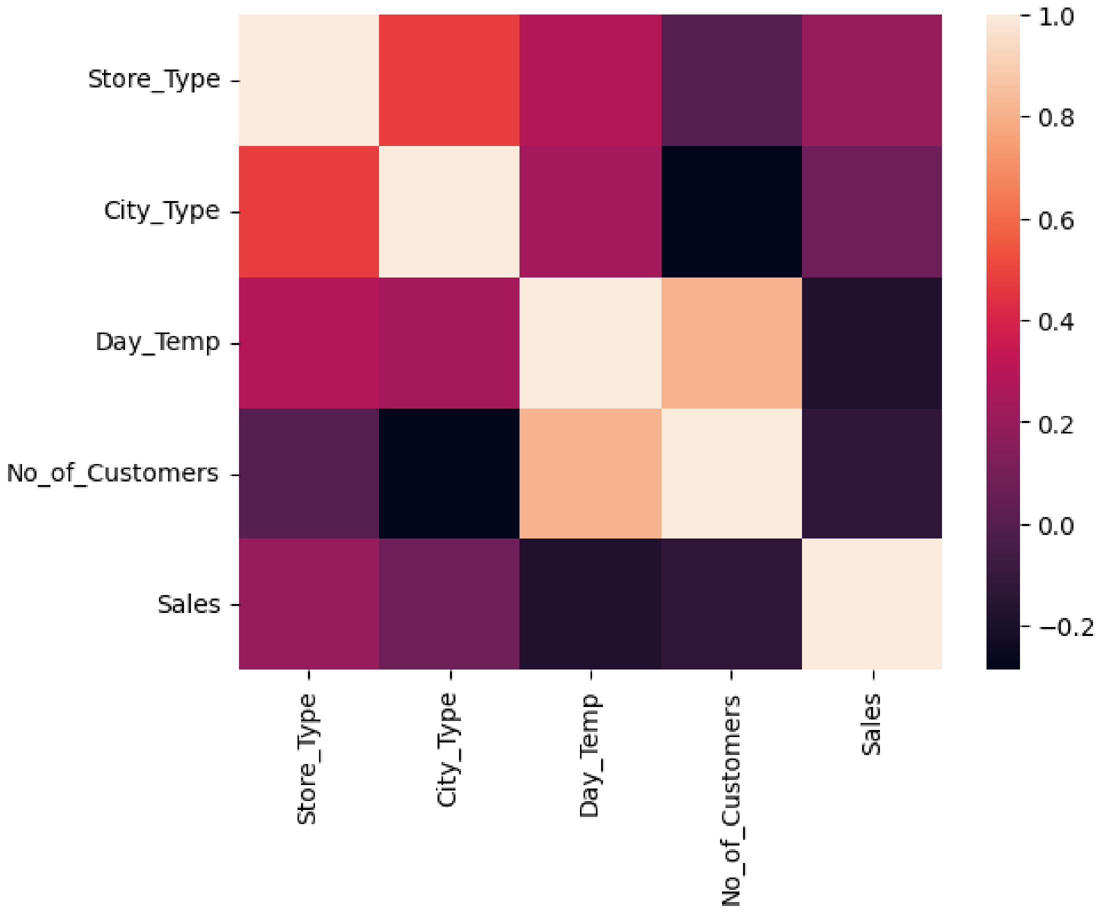
```
10 2020-11-29           1          1      23.0              94.0  1254.0
11 2020-11-30           1          1      22.0              91.0  4232.0

   Product_Quality
0                A
1                A
2                A
3              NaN
4                B
5                B
6                C
7              NaN
8                B
9                A
10               A
11               A
```

[45]:
```
sns.heatmap(df_saless.corr())
```

```
<ipython-input-45-3ad31eb899e5>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df_saless.corr())
```

[45]:
```
<Axes: >
```


Store_Type
City_Type
Day_Temp
No_of_Customers
Sales
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

```
[45]:
```