In [2]:
```
###  Handling of Inappropriate Data
```

In [3]:
```
import numpy as np
import pandas as pd
```

In [4]:
```
df = pd.read_csv("C:/Users/SW20407278/Desktop/Final AI/Hands-On/Handling_Inapprop
```

In [5]:
```
df
```

Out[5]:

| | CustomerID | Age_Group | Rating(1-5) | Hotel | FoodPreference | Bill | NoOfPax | EstimatedSala |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 20-25 | 4 | Ibis | veg | 1300 | 2 | 400 |
| **1** | 2 | 30-35 | 5 | LemonTree | Non-Veg | 2000 | 3 | 590 |
| **2** | 3 | 25-30 | 6 | RedFox | Veg | 1322 | 2 | 300 |
| **3** | 4 | 20-25 | -1 | LemonTree | Veg | 1234 | 2 | 1200 |
| **4** | 5 | 35+ | 3 | Ibis | Vegetarian | 989 | 2 | 450 |
| **5** | 6 | 35+ | 3 | Ibys | Non-Veg | 1909 | 2 | 1222 |
| **6** | 7 | 35+ | 4 | RedFox | Vegetarian | 1000 | -1 | 211 |
| **7** | 8 | 20-25 | 7 | LemonTree | Veg | 2999 | -10 | 3456 |
| **8** | 9 | 25-30 | 2 | Ibis | Non-Veg | 3456 | 3 | -999 |
| **9** | 9 | 25-30 | 2 | Ibis | Non-Veg | 3456 | 3 | -999 |
| **10** | 10 | 30-35 | 5 | RedFox | non-Veg | -6755 | 4 | 877 |

In [6]:
```
df.duplicated()
```

Out[6]:
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9      True
10    False
dtype: bool
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   CustomerID       11 non-null     int64 
 1   Age_Group        11 non-null     object
 2   Rating(1-5)      11 non-null     int64 
 3   Hotel            11 non-null     object
 4   FoodPreference   11 non-null     object
 5   Bill             11 non-null     int64 
 6   NoOfPax          11 non-null     int64 
 7   EstimatedSalary  11 non-null     int64 
 8   Age_Group.1      11 non-null     object
dtypes: int64(5), object(4)
memory usage: 920.0+ bytes
```

In [8]:
```
###   Identify the data type for each of the column
###   Check for duplicate records and remove them
###   Check for duplicate columns and remove them
```

In [10]:
```
## Dropping duplicate rows
df.drop_duplicates(inplace =  True)
```

In [11]:
```
df
```

Out[11]:

| | CustomerID | Age_Group | Rating(1-5) | Hotel | FoodPreference | Bill | NoOfPax | EstimatedSala |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 20-25 | 4 | Ibis | veg | 1300 | 2 | 400 |
| **1** | 2 | 30-35 | 5 | LemonTree | Non-Veg | 2000 | 3 | 590 |
| **2** | 3 | 25-30 | 6 | RedFox | Veg | 1322 | 2 | 300 |
| **3** | 4 | 20-25 | -1 | LemonTree | Veg | 1234 | 2 | 1200 |
| **4** | 5 | 35+ | 3 | Ibis | Vegetarian | 989 | 2 | 450 |
| **5** | 6 | 35+ | 3 | Ibys | Non-Veg | 1909 | 2 | 1222 |
| **6** | 7 | 35+ | 4 | RedFox | Vegetarian | 1000 | -1 | 211 |
| **7** | 8 | 20-25 | 7 | LemonTree | Veg | 2999 | -10 | 3456 |
| **8** | 9 | 25-30 | 2 | Ibis | Non-Veg | 3456 | 3 | -999 |
| **10** | 10 | 30-35 | 5 | RedFox | non-Veg | -6755 | 4 | 877 |

In [12]:
```
## Resetting Index

index = np.array(list(range(0, len(df))))
df.set_index(index, inplace=True)
```

In [13]:
```
df
```

Out[13]:
```
   CustomerID  Age_Group  Rating(1-      Hotel  FoodPreference   Bill  NoOfPax  EstimatedSalar
                                5)
0           1      20-25          4       Ibis             veg   1300        2            4000
1           2      30-35          5  LemonTree         Non-Veg   2000        3            5900
2           3      25-30          6     RedFox             Veg   1322        2            3000
3           4      20-25         -1  LemonTree             Veg   1234        2           12000
4           5        35+          3       Ibis      Vegetarian    989        2            4500
5           6        35+          3       Ibys         Non-Veg   1909        2           12222
6           7        35+          4     RedFox      Vegetarian   1000       -1            2112
7           8      20-25          7  LemonTree             Veg   2999      -10           34567
8           9      25-30          2       Ibis         Non-Veg   3456        3           -9999
9          10      30-35          5     RedFox         non-Veg  -6755        4            8777
```

In [14]:
```
## Dropping duplicate columns
df.drop( ['Age_Group.1'] , axis = 1, inplace=True)
```

In [15]:
```
df
```

Out[15]:
```
   CustomerID  Age_Group  Rating(1-      Hotel  FoodPreference   Bill  NoOfPax  EstimatedSalar
                                5)
0           1      20-25          4       Ibis             veg   1300        2            4000
1           2      30-35          5  LemonTree         Non-Veg   2000        3            5900
2           3      25-30          6     RedFox             Veg   1322        2            3000
3           4      20-25         -1  LemonTree             Veg   1234        2           12000
4           5        35+          3       Ibis      Vegetarian    989        2            4500
5           6        35+          3       Ibys         Non-Veg   1909        2           12222
6           7        35+          4     RedFox      Vegetarian   1000       -1            2112
7           8      20-25          7  LemonTree             Veg   2999      -10           34567
8           9      25-30          2       Ibis         Non-Veg   3456        3           -9999
9          10      30-35          5     RedFox         non-Veg  -6755        4            8777
```

In [64]:
```
## Identified CustomerID, Bill, EstimatedSalary as continuous variables
## If the column is continuous, check if the negative or positive values are allo
## If negative value is not applicable for the column, replace it with NaN.
```

In [67]:

```
df.CustomerID.loc [df.CustomerID < 0 ] = np.nan
```

```
C:\Users\Swati\AppData\Local\Temp\ipykernel_5984\3811053347.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df.CustomerID.loc [df.CustomerID < 0 ] = np.nan
```

In [69]:

```
df.Bill.loc[df.Bill < 0]  = np.nan
```

```
C:\Users\Swati\AppData\Local\Temp\ipykernel_5984\2083596671.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df.Bill.loc[df.Bill < 0]  = np.nan
```

In [70]:

```
df.EstimatedSalary.loc[df.EstimatedSalary < 0]  = np.nan
```

```
C:\Users\Swati\AppData\Local\Temp\ipykernel_5984\401954487.py:1: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df.EstimatedSalary.loc[df.EstimatedSalary < 0]  = np.nan
```

In [71]:

```
df
```

Out[71]:

| | CustomerID | Age_Group | Rating(1-5) | Hotel | FoodPreference | Bill | NoOfPax | EstimatedSala |
|---|---|---|---|---|---|---|---|---|
| **0** | 1.0 | 20-25 | 4 | Ibis | veg | 1300.0 | 2 | 40000 |
| **1** | 2.0 | 30-35 | 5 | LemonTree | Non-Veg | 2000.0 | 3 | 59000 |
| **2** | 3.0 | 25-30 | 6 | RedFox | Veg | 1322.0 | 2 | 30000 |
| **3** | 4.0 | 20-25 | -1 | LemonTree | Veg | 1234.0 | 2 | 120000 |
| **4** | 5.0 | 35+ | 3 | Ibis | Vegetarian | 989.0 | 2 | 45000 |
| **5** | 6.0 | 35+ | 3 | Ibys | Non-Veg | 1909.0 | 2 | 122220 |
| **6** | 7.0 | 35+ | 4 | RedFox | Vegetarian | 1000.0 | -1 | 21122 |
| **7** | 8.0 | 20-25 | 7 | LemonTree | Veg | 2999.0 | -10 | 345673 |
| **8** | 9.0 | 25-30 | 2 | Ibis | Non-Veg | 3456.0 | 3 | Na |
| **9** | 10.0 | 30-35 | 5 | RedFox | non-Veg | NaN | 4 | 87777 |

In [72]:

```
##
## Identified Rating,NoOfPax as discrete variables
## If the column is discrete check if the negative or positive values are allowed
## If negative value is not applicable for the column, replace it with NaN.
## Check the values for discrete column falls in specified range. IF it is going
```

In [73]:

```
df['Rating(1-5)'].loc[  (df['Rating(1-5)'] < 0)  |  (df['Rating(1-5)'] > 5) ]  =
```

```
C:\Users\Swati\AppData\Local\Temp\ipykernel_5984\4210872344.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df['Rating(1-5)'].loc[  (df['Rating(1-5)'] < 0)  |  (df['Rating(1-5)'] > 5) ]
= np.nan
```

In [74]:

```
df
```

Out[74]:

| | CustomerID | Age_Group | Rating(1-5) | Hotel | FoodPreference | Bill | NoOfPax | EstimatedSala |
|---|---|---|---|---|---|---|---|---|
| **0** | 1.0 | 20-25 | 4.0 | Ibis | veg | 1300.0 | 2 | 40000 |
| **1** | 2.0 | 30-35 | 5.0 | LemonTree | Non-Veg | 2000.0 | 3 | 59000 |
| **2** | 3.0 | 25-30 | NaN | RedFox | Veg | 1322.0 | 2 | 30000 |
| **3** | 4.0 | 20-25 | NaN | LemonTree | Veg | 1234.0 | 2 | 120000 |
| **4** | 5.0 | 35+ | 3.0 | Ibis | Vegetarian | 989.0 | 2 | 45000 |
| **5** | 6.0 | 35+ | 3.0 | Ibys | Non-Veg | 1909.0 | 2 | 122220 |
| **6** | 7.0 | 35+ | 4.0 | RedFox | Vegetarian | 1000.0 | -1 | 21122 |
| **7** | 8.0 | 20-25 | NaN | LemonTree | Veg | 2999.0 | -10 | 345673 |
| **8** | 9.0 | 25-30 | 2.0 | Ibis | Non-Veg | 3456.0 | 3 | Na |
| **9** | 10.0 | 30-35 | 5.0 | RedFox | non-Veg | NaN | 4 | 87777 |

In [75]:

```
df['NoOfPax'].loc[  (df['NoOfPax'] < 1)  |  (df['NoOfPax'] > 20) ]  = np.nan
```

```
C:\Users\Swati\AppData\Local\Temp\ipykernel_5984\1417669336.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df['NoOfPax'].loc[  (df['NoOfPax'] < 1)  |  (df['NoOfPax'] > 20) ]  = np.nan
```

In [76]:

```
df
```

Out[76]:

| | CustomerID | Age_Group | Rating(1-5) | Hotel | FoodPreference | Bill | NoOfPax | EstimatedSala |
|---|---|---|---|---|---|---|---|---|
| **0** | 1.0 | 20-25 | 4.0 | Ibis | veg | 1300.0 | 2.0 | 40000 |
| **1** | 2.0 | 30-35 | 5.0 | LemonTree | Non-Veg | 2000.0 | 3.0 | 59000 |
| **2** | 3.0 | 25-30 | NaN | RedFox | Veg | 1322.0 | 2.0 | 30000 |
| **3** | 4.0 | 20-25 | NaN | LemonTree | Veg | 1234.0 | 2.0 | 120000 |
| **4** | 5.0 | 35+ | 3.0 | Ibis | Vegetarian | 989.0 | 2.0 | 45000 |
| **5** | 6.0 | 35+ | 3.0 | Ibys | Non-Veg | 1909.0 | 2.0 | 122220 |
| **6** | 7.0 | 35+ | 4.0 | RedFox | Vegetarian | 1000.0 | NaN | 21122 |
| **7** | 8.0 | 20-25 | NaN | LemonTree | Veg | 2999.0 | NaN | 345673 |
| **8** | 9.0 | 25-30 | 2.0 | Ibis | Non-Veg | 3456.0 | 3.0 | Na |
| **9** | 10.0 | 30-35 | 5.0 | RedFox | non-Veg | NaN | 4.0 | 87777 |

In [77]:

```
## Identofied Age_Group, Hotel, FoodPreference as categorical variables
## Check for the unique categories.
## Handle the spelling mistakes and case errors
```

In [78]:

```
df.Age_Group.unique()
```

Out[78]:

```
array(['20-25', '30-35', '25-30', '35+'], dtype=object)
```

In [79]:

```
df.Hotel.unique()
```

Out[79]:

```
array(['Ibis', 'LemonTree', 'RedFox', 'Ibys'], dtype=object)
```

In [80]:

```
df.Hotel.replace(['Ibys','ibis','IbIs'],'Ibis'  , inplace=True)
```

In [81]:

```
df.FoodPreference.unique()
```

Out[81]:

```
array(['veg', 'Non-Veg', 'Veg', 'Vegetarian', 'non-Veg'], dtype=object)
```

In [82]:

```
df.FoodPreference.replace(['Vegetarian','veg'],'Veg'  , inplace=True)
df.FoodPreference.replace(['non-Veg'], 'Non-Veg', inplace=True)
```

In [83]:
```
df
```

Out[83]:

| | CustomerID | Age_Group | Rating(1-5) | Hotel | FoodPreference | Bill | NoOfPax | EstimatedSala |
|---|---|---|---|---|---|---|---|---|
| **0** | 1.0 | 20-25 | 4.0 | Ibis | Veg | 1300.0 | 2.0 | 40000 |
| **1** | 2.0 | 30-35 | 5.0 | LemonTree | Non-Veg | 2000.0 | 3.0 | 59000 |
| **2** | 3.0 | 25-30 | NaN | RedFox | Veg | 1322.0 | 2.0 | 30000 |
| **3** | 4.0 | 20-25 | NaN | LemonTree | Veg | 1234.0 | 2.0 | 120000 |
| **4** | 5.0 | 35+ | 3.0 | Ibis | Veg | 989.0 | 2.0 | 45000 |
| **5** | 6.0 | 35+ | 3.0 | Ibis | Non-Veg | 1909.0 | 2.0 | 122220 |
| **6** | 7.0 | 35+ | 4.0 | RedFox | Veg | 1000.0 | NaN | 21122 |
| **7** | 8.0 | 20-25 | NaN | LemonTree | Veg | 2999.0 | NaN | 345673 |
| **8** | 9.0 | 25-30 | 2.0 | Ibis | Non-Veg | 3456.0 | 3.0 | Na |
| **9** | 10.0 | 30-35 | 5.0 | RedFox | Non-Veg | NaN | 4.0 | 87777 |

In [ ]: