week2

May 13, 2023

1 DATA GLACER - WEEK2

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

2 1. Dataset

2.1 1.1 Cab_Data.csv

[2]:
```
df_cab = pd.read_csv('Cab_Data.csv')
print('Dataset Shape:')
print(f"\033[1m{df_cab.shape}\033[0m")
df_cab.head()
```

```
Dataset Shape:
(359392, 7)
```

[2]:

	Transaction ID	Date of Travel	Company	City	KM Travelled	Price Charged	Cost of Trip
0	10000011	42377	Pink Cab	ATLANTA GA	30.45	370.95	313.635
1	10000012	42375	Pink Cab	ATLANTA GA	28.62	358.52	334.854
2	10000013	42371	Pink Cab	ATLANTA GA	9.04	125.20	97.632
3	10000014	42376	Pink Cab	ATLANTA GA	33.17	377.40	351.602
4	10000015	42372	Pink Cab	ATLANTA GA	8.73	114.62	97.776

2.1.1 1.1.1 Field names and data types

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 359392 entries, 0 to 359391
```

```
Data columns (total 7 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Transaction ID  359392 non-null  int64
 1   Date of Travel  359392 non-null  int64
 2   Company         359392 non-null  object
 3   City            359392 non-null  object
 4   KM Travelled    359392 non-null  float64
 5   Price Charged   359392 non-null  float64
 6   Cost of Trip    359392 non-null  float64
dtypes: float64(3), int64(2), object(2)
memory usage: 19.2+ MB
```

2.1.2 1.1.2 Missing Values and Duplicate Values Check

[3]:
```
df_cab.isnull().sum()
```

[3]:
```
Transaction ID    0
Date of Travel    0
Company           0
City              0
KM Travelled      0
Price Charged     0
Cost of Trip      0
dtype: int64
```

[9]:
```
dup_count = df_cab[df_cab.duplicated()].shape[0]
print('Duplicate Rows for Cab_Data.csv: ', dup_count)
```

```
Duplicate Rows for Cab_Data.csv:  0
```

- No missing values for Cab_Data.csv
- No Duplicate values for Cab_Data.csv

2.2 1.2 Customer_ID.csv

[3]:
```
df_cus = pd.read_csv('Customer_ID.csv')
print('Dataset Shape:')
print(f"\033[1m{df_cus.shape}\033[0m")
df_cus.head()
```

```
Dataset Shape:
(49171, 4)
```

[3]:

	Customer ID	Gender	Age	Income (USD/Month)
0	29290	Male	28	10813
1	27703	Male	27	9237
2	28712	Male	53	11242
3	28020	Male	23	23327

```
4        27182   Male   33                8536
```

2.2.1 1.2.1 Field names and data types

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49171 entries, 0 to 49170
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Customer ID         49171 non-null  int64
 1   Gender              49171 non-null  object
 2   Age                 49171 non-null  int64
 3   Income (USD/Month)  49171 non-null  int64
dtypes: int64(3), object(1)
memory usage: 1.5+ MB
```

2.2.2 1.2.2 Missing Values and Duplicate Values

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

[13]:
```
Customer ID           0
Gender                0
Age                   0
Income (USD/Month)    0
dtype: int64
```

[16]:
```
dup_count = df_cus[df_cus.duplicated()].shape[0]
print('Duplicate Rows for Customer_ID.csv: ', dup_count)
```

```
Duplicate Rows for Customer_ID.csv:  0
```

- No missing values for Customer_ID.csv
- No duplicate values for Customer_ID.csv

2.3 1.3 City.csv

[4]:
```
df_city = pd.read_csv('City.csv')
print('Dataset Shape:')
print(f"\033[1m{df_city.shape}\033[0m")
df_city.head()
```

```
Dataset Shape:
(20, 3)
```

[4]:
```
             City   Population      Users
0     NEW YORK NY   8,405,837    302,149
1      CHICAGO IL   1,955,130    164,468
```

```
2  LOS ANGELES CA   1,595,037    144,132
3         MIAMI FL   1,339,155     17,675
4  SILICON VALLEY   1,177,609     27,247
```

2.3.1 1.3.1 Field names and data types

[19]:
```
df_city.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   City        20 non-null     object
 1   Population  20 non-null     object
 2   Users       20 non-null     object
dtypes: object(3)
memory usage: 608.0+ bytes
```

2.3.2 1.3.2 Missing Values and Duplicate Values

[21]:
```
df_city.isnull().sum()
```

[21]:
```
City          0
Population    0
Users         0
dtype: int64
```

[20]:
```
dup_count = df_city[df_city.duplicated()].shape[0]
print('Duplicate Rows for City.csv: ', dup_count)
```

```
Duplicate Rows for City.csv:  0
```

- No missing values for City.csv
- No duplicate values for City.csv

2.4 1.4 Transaction_ID.csv

[5]:
```
df_trans = pd.read_csv('Transaction_ID.csv')
print('Dataset Shape:')
print(f"\033[1m{df_trans.shape}\033[0m")
df_trans.head()
```

```
Dataset Shape:
(440098, 3)
```

[5]:
```
   Transaction ID  Customer ID Payment_Mode
0        10000011        29290         Card
1        10000012        27703         Card
```

```
2        10000013        28712         Cash
3        10000014        28020         Cash
4        10000015        27182         Card
```

2.4.1 1.4.1 Field names and data types

[23]:
```
df_trans.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440098 entries, 0 to 440097
Data columns (total 3 columns):
 #   Column          Non-Null Count   Dtype 
---  ------          --------------   ----- 
 0   Transaction ID  440098 non-null  int64 
 1   Customer ID     440098 non-null  int64 
 2   Payment_Mode    440098 non-null  object
dtypes: int64(2), object(1)
memory usage: 10.1+ MB
```

2.4.2 1.4.2 Missing Values and Duplicate Values

[26]:
```
df_trans.isnull().sum()
```

[26]:
```
Transaction ID    0
Customer ID       0
Payment_Mode      0
dtype: int64
```

[25]:
```
dup_count = df_trans[df_trans.duplicated()].shape[0]
print('Duplicate Rows for Transaction_ID.csv: ', dup_count)
```

```
Duplicate Rows for Transaction_ID.csv:  0
```

- No missing values for Transaction_ID.csv
- No duplicate values for Transaction_ID.csv