step_1_data_cleaning

July 6, 2025

1 Data cleaning - Mobile User Behavior

1.1 librerias y datos

[1]:
```
import pandas as pd
```

[2]:
```
original_dataset = pd.read_csv("user_behavior_dataset_original.csv")
```

1.2 Exploracion inicial

- head() »> para ver los primeros registros
- tail() »> para ver los ultimos registros
- shape »> para ver el numero de filas y columnas. O la forma del dataset
- info() »> para ver informaciones generales del dataset y sus columnas

[4]:
```
original_dataset.head(4)
```

[4]:
```
   User ID    Device Model Operating System  App Usage Time (min/day)  \
0        1  Google Pixel 5          Android                       393
1        2       OnePlus 9          Android                       268
2        3    Xiaomi Mi 11          Android                       154
3        4  Google Pixel 5          Android                       239

   Screen On Time (hours/day)  Battery Drain (mAh/day)  \
0                         6.4                     1872
1                         4.7                     1331
2                         4.0                      761
3                         4.8                     1676

   Number of Apps Installed  Data Usage (MB/day)  Age  Gender  \
0                        67                 1122   40    Male
1                        42                  944   47  Female
2                        32                  322   42    Male
3                        56                  871   20    Male

   User Behavior Class
0                    4
1                    3
```

```
2                    2
3                    3
```

[5]:
```
original_dataset.tail(4)
```

[5]:
```
     User ID        Device Model Operating System  App Usage Time (min/day)  \
696      697        Xiaomi Mi 11          Android                       316
697      698      Google Pixel 5          Android                        99
698      699  Samsung Galaxy S21          Android                        62
699      700           OnePlus 9          Android                       212

     Screen On Time (hours/day)  Battery Drain (mAh/day)  \
696                         6.8                     1965
697                         3.1                      942
698                         1.7                      431
699                         5.4                     1306

     Number of Apps Installed  Data Usage (MB/day)  Age  Gender  \
696                        68                 1201   59    Male
697                        22                  457   50  Female
698                        13                  224   44    Male
699                        49                  828   23  Female

     User Behavior Class
696                    4
697                    2
698                    1
699                    3
```

[6]:
```
original_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   User ID                     700 non-null    int64
 1   Device Model                700 non-null    object
 2   Operating System            700 non-null    object
 3   App Usage Time (min/day)    700 non-null    int64
 4   Screen On Time (hours/day)  700 non-null    float64
 5   Battery Drain (mAh/day)     700 non-null    int64
 6   Number of Apps Installed    700 non-null    int64
 7   Data Usage (MB/day)         700 non-null    int64
 8   Age                         700 non-null    int64
 9   Gender                      700 non-null    object
 10  User Behavior Class         700 non-null    int64
dtypes: float64(1), int64(7), object(3)
```

```
memory usage: 60.3+ KB
```

[7]:
```
original_dataset.shape
```

[7]:
```
(700, 11)
```

1.3 Data cleaning

1.3.1 Valores nulos

Buscar por valores nulos en el dataset

[9]:
```
original_dataset.isnull().sum()
```

[9]:
```
User ID                        0
Device Model                   0
Operating System               0
App Usage Time (min/day)       0
Screen On Time (hours/day)     0
Battery Drain (mAh/day)        0
Number of Apps Installed       0
Data Usage (MB/day)            0
Age                            0
Gender                         0
User Behavior Class            0
dtype: int64
```

1.3.2 Valores duplicados

Buscar por valores duplicados

[11]:
```
original_dataset.duplicated().sum()
```

[11]:
```
np.int64(0)
```

1.3.3 Valores de columnas categoricas

Verificar los valores de cada columna, para buscar por inconsistencias.

[12]:
```
original_dataset["Device Model"].value_counts()
```

[12]:
```
Device Model
Xiaomi Mi 11          146
iPhone 12             146
Google Pixel 5        142
OnePlus 9             133
Samsung Galaxy S21    133
Name: count, dtype: int64
```

[15]:
```
original_dataset["Gender"].value_counts()
```

[15]:
```
Gender
Male      364
Female    336
Name: count, dtype: int64
```

[16]:
```
original_dataset["Operating System"].value_counts()
```

[16]:
```
Operating System
Android    554
iOS        146
Name: count, dtype: int64
```

[17]:
```
original_dataset["User Behavior Class"].value_counts()
```

[17]:
```
User Behavior Class
2    146
3    143
4    139
5    136
1    136
Name: count, dtype: int64
```

Definir la columna “User Behavior Class”, pues el titulo no sugiere de que se trata.

Info del dataset:

- User Behavior Class: Classification of user behavior based on usage patterns (1 to 5).
- Each entry is categorized into one of five user behavior classes, ranging from light to extreme usage.

La columna ‘User Behavior Class’ contiene valores numéricos (del 1 al 5), pero estos números representan categorías de usuarios, no cantidades reales. Por eso, tiene sentido tratar esta columna como una variable categórica ordinal.

Una nueva columna se incluye para convertir los valores numericos a variables categoricas, así:

- 1: ‘Muy bajo’,
- 2: ‘Bajo’,
- 3: ‘Medio’,
- 4: ‘Alto’,
- 5: ‘Muy alto’

[22]:
```
original_dataset["Defined Behavior"] = original_dataset["User Behavior Class"].
 ↪map({
    1: 'Very low',
    2: 'Low',
    3: 'Medium',
    4: 'High',
    5: 'Very high'
}).astype('category')
```

[23]:
```
original_dataset["Defined Behavior"].value_counts()
```

[23]:
```
Defined Behavior
Low          146
Medium       143
High         139
Very high    136
Very low     136
Name: count, dtype: int64
```

1.3.4 Valores de columnas numericas

[24]:
```
original_dataset.describe().T
```

[24]:

	count	mean	std	min	25%	\
User ID	700.0	350.500000	202.216880	1.0	175.75	
App Usage Time (min/day)	700.0	271.128571	177.199484	30.0	113.25	
Screen On Time (hours/day)	700.0	5.272714	3.068584	1.0	2.50	
Battery Drain (mAh/day)	700.0	1525.158571	819.136414	302.0	722.25	
Number of Apps Installed	700.0	50.681429	26.943324	10.0	26.00	
Data Usage (MB/day)	700.0	929.742857	640.451729	102.0	373.00	
Age	700.0	38.482857	12.012916	18.0	28.00	
User Behavior Class	700.0	2.990000	1.401476	1.0	2.00	

	50%	75%	max
User ID	350.5	525.25	700.0
App Usage Time (min/day)	227.5	434.25	598.0
Screen On Time (hours/day)	4.9	7.40	12.0
Battery Drain (mAh/day)	1502.5	2229.50	2993.0
Number of Apps Installed	49.0	74.00	99.0
Data Usage (MB/day)	823.5	1341.00	2497.0
Age	38.0	49.00	59.0
User Behavior Class	3.0	4.00	5.0

1.3.5 Configuracion del nombre de las columnas

Cambiar el nombre de las columnas para evitar posibles conflictos posteriormente.

[25]:
```
# Eliminar espacios al inicio y final de cada nombre de columna

original_dataset.columns = original_dataset.columns.str.strip()
```

[26]:
```
# Convertir todos los nombres de columna a minúsculas

original_dataset.columns = original_dataset.columns.str.lower()
```

[27]:
```
# Reemplazar los espacios internos por guiones bajos

```

```
original_dataset.columns = original_dataset.columns.str.replace(" ", "_")
```

[31]:
```
original_dataset.columns
```

[31]:
```
Index(['user_id', 'device_model', 'operating_system',
       'app_usage_time_(min/day)', 'screen_on_time_(hours/day)',
       'battery_drain_(mah/day)', 'number_of_apps_installed',
       'data_usage_(mb/day)', 'age', 'gender', 'user_behavior_class',
       'defined_behavior'],
      dtype='object')
```

1.4 Exportar el archivo

[32]:
```
original_dataset.to_csv("user_behavior_dataset_clean.csv", index=False)
```