Skills Network

Hands-on Practice Lab: Importing Dataset - Laptops Pricing

Estimated time needed: **20** minutes

In this lab, you will practice the process of loading and drawing basic insights on a dataset as learnt through the module. You are being provided with a fresh dataset on 'Laptop Pricing' which will be used for all the practice labs throughout the course.

Objectives

After completing this lab you will be able to:

- Import a dataset from a CSV file to a Pandas dataframe
- Develop some basic insights about the dataset

Setup

For this lab, we will be using the following libraries:

- `skillsnetwork` for downloading the daataset
- `pandas` for managing the data.
- `numpy` for mathematical operations.

Importing Required Libraries

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

The data set to be used is available on the link below.

The functions below will download the dataset into your browser:

In [2]:
```
from pyodide.http import pyfetch

async def download(url, filename):
    response = await pyfetch(url)
    if response.status == 200:
        with open(filename, "wb") as f:
            f.write(await response.bytes())
```

In [3]:
```
file_path = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DA0101EN-Coursera/laptop_pricing_dataset_base.c
```

To obtain the dataset, utilize the download() function as defined above:

In [4]:
```
await download(file_path, "laptops.csv")
file_name="laptops.csv"
```

In [5]:
```
df = pd.read_csv(file_name)
```

> Note: This version of the lab is working on JupyterLite, which requires the dataset to be downloaded to the interface. While working on the downloaded version of this notebook on their local machines, the learners can simply **skip the steps above**, and simply use the URL directly in the `pandas.read_csv()` function. You can uncomment and run the statements in the cell below.

In [ ]:
```
#filepath = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DA0101EN-Coursera/laptop_pricing_dataset_base.c
#df = pd.read_csv(filepath, header=None)
```

Task #1:

Load the dataset to a pandas dataframe named 'df'

Print the first 5 entries of the dataset to confirm loading.

In [6]:
```
# Write your code below and press Shift+Enter to execute.
df = pd.read_csv(file_name, header=None)
print(df.head())
```

```
     0  1          2  3  4  5       6    7  8    9    10    11
0  Acer  4  IPS Panel  2  1  5   35.56  1.6  8  256   1.6   978
1  Dell  3    Full HD  1  1  3  39.624  2.0  4  256   2.2   634
2  Dell  3    Full HD  1  1  7  39.624  2.7  8  256   2.2   946
3  Dell  4  IPS Panel  2  1  5  33.782  1.6  8  128  1.22  1244
4    HP  4    Full HD  2  1  7  39.624  1.8  8  256  1.91   837
```

▸ Click here for solution

Task #2:

Add headers to the dataframe

The headers for the dataset, in sequence, are "Manufacturer", "Category", "Screen", "GPU", "OS", "CPU_core", "Screen_Size_inch", "CPU_frequency", "RAM_GB", "Storage_GB_SSD", "Weight_kg" and "Price".
Confirm insertion by printing the first 10 rows of the dataset.

```
In [7]: # Write your code below and press Shift+Enter to execute.
        df.columns = [ "Manufacturer", "Category", "Screen", "GPU", "OS", "CPU_core", "Screen_Size_inch", "CPU_frequency", "RAM_GB", "Storage_GB_SSD", "Weight_kg
        df.columns
```

```
Out[7]: Index(['Manufacturer', 'Category', 'Screen', 'GPU', 'OS', 'CPU_core',
               'Screen_Size_inch', 'CPU_frequency', 'RAM_GB', 'Storage_GB_SSD',
               'Weight_kg', 'Price'],
              dtype='object')
```

▸ Click here for solution

Task #3:

Replace '?' with 'NaN'

Replace the '?' entries in the dataset with NaN value, recevied from the Numpy package.

```
In [8]: # Write your code below and press Shift+Enter to execute.
        df1=df.replace('?',np.NaN)
        df1
```

Out[8]:

	Manufacturer	Category	Screen	GPU	OS	CPU_core	Screen_Size_inch	CPU_frequency	RAM_GB	Storage_GB_SSD	Weight_kg	Price
0	Acer	4	IPS Panel	2	1	5	35.56	1.6	8	256	1.6	978
1	Dell	3	Full HD	1	1	3	39.624	2.0	4	256	2.2	634
2	Dell	3	Full HD	1	1	7	39.624	2.7	8	256	2.2	946
3	Dell	4	IPS Panel	2	1	5	33.782	1.6	8	128	1.22	1244
4	HP	4	Full HD	2	1	7	39.624	1.8	8	256	1.91	837
...	...	...	...	...	...	...	...	...	...	...	...	...
233	Lenovo	4	IPS Panel	2	1	7	35.56	2.6	8	256	1.7	1891
234	Toshiba	3	Full HD	2	1	5	33.782	2.4	8	256	1.2	1950
235	Lenovo	4	IPS Panel	2	1	5	30.48	2.6	8	256	1.36	2236
236	Lenovo	3	Full HD	3	1	5	39.624	2.5	6	256	2.4	883
237	Toshiba	3	Full HD	2	1	5	35.56	2.3	8	256	1.95	1499

238 rows × 12 columns

▸ Click here for solution

Task #4:

Print the data types of the dataframe columns

Make a note of the data types of the different columns of the dataset.

```
In [9]: # Write your code below and press Shift+Enter to execute.
        df1.dtypes
```

```
Out[9]: Manufacturer         object
        Category              int64
        Screen               object
        GPU                   int64
        OS                    int64
        CPU_core              int64
        Screen_Size_inch     object
        CPU_frequency       float64
        RAM_GB                int64
        Storage_GB_SSD        int64
        Weight_kg            object
        Price                 int64
        dtype: object
```

▸ Click here for solution

Task #5:

Print the statistical description of the dataset, including that of 'object' data types.

In [10]:
```
# Write your code below and press Shift+Enter to execute.
df1.describe(include="all")
```

Out[10]:

	Manufacturer	Category	Screen	GPU	OS	CPU_core	Screen_Size_inch	CPU_frequency	RAM_GB	Storage_GB_SSD	Weight_kg	Price
count	238	238.000000	238	238.000000	238.000000	238.000000	234	238.000000	238.000000	238.000000	233	238.000000
unique	11	NaN	2	NaN	NaN	NaN	9	NaN	NaN	NaN	77	NaN
top	Dell	NaN	Full HD	NaN	NaN	NaN	39.624	NaN	NaN	NaN	2.2	NaN
freq	71	NaN	161	NaN	NaN	NaN	89	NaN	NaN	NaN	21	NaN
mean	NaN	3.205882	NaN	2.151261	1.058824	5.630252	NaN	2.360084	7.882353	245.781513	NaN	1462.344538
std	NaN	0.776533	NaN	0.638282	0.235790	1.241787	NaN	0.411393	2.482603	34.765316	NaN	574.607699
min	NaN	1.000000	NaN	1.000000	1.000000	3.000000	NaN	1.200000	4.000000	128.000000	NaN	527.000000
25%	NaN	3.000000	NaN	2.000000	1.000000	5.000000	NaN	2.000000	8.000000	256.000000	NaN	1066.500000
50%	NaN	3.000000	NaN	2.000000	1.000000	5.000000	NaN	2.500000	8.000000	256.000000	NaN	1333.000000
75%	NaN	4.000000	NaN	3.000000	1.000000	7.000000	NaN	2.700000	8.000000	256.000000	NaN	1777.000000
max	NaN	5.000000	NaN	3.000000	2.000000	7.000000	NaN	2.900000	16.000000	256.000000	NaN	3810.000000

▸ Click here for solution

Task #6:

Print the summary information of the dataset.

In [11]:
```
# Write your code below and press Shift+Enter to execute.
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 238 entries, 0 to 237
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Manufacturer      238 non-null    object 
 1   Category          238 non-null    int64  
 2   Screen            238 non-null    object 
 3   GPU               238 non-null    int64  
 4   OS                238 non-null    int64  
 5   CPU_core          238 non-null    int64  
 6   Screen_Size_inch  234 non-null    object 
 7   CPU_frequency     238 non-null    float64
 8   RAM_GB            238 non-null    int64  
 9   Storage_GB_SSD    238 non-null    int64  
 10  Weight_kg         233 non-null    object 
 11  Price             238 non-null    int64  
dtypes: float64(1), int64(7), object(4)
memory usage: 18.7+ KB
```

▸ Click here for solution

Congratulations! You have completed the lab

Authors

Abhishek Gagneja

Vicky Kuo