RWANDA POLYTECHNIC

MUSANZE COLLEGE

Date: 07th/11/2024

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

OPTION: INFORMATION TECHNOLOGY

PROGRAM: B-TECH

MODULE: MACHINE LEARNING

ASSIGNMENT GROUP

GROUP MEMBERS:

- Iradukunda Mugisha Enock **24RP15801**
- Ishimwe Deborah **24RP15460**

Questions:

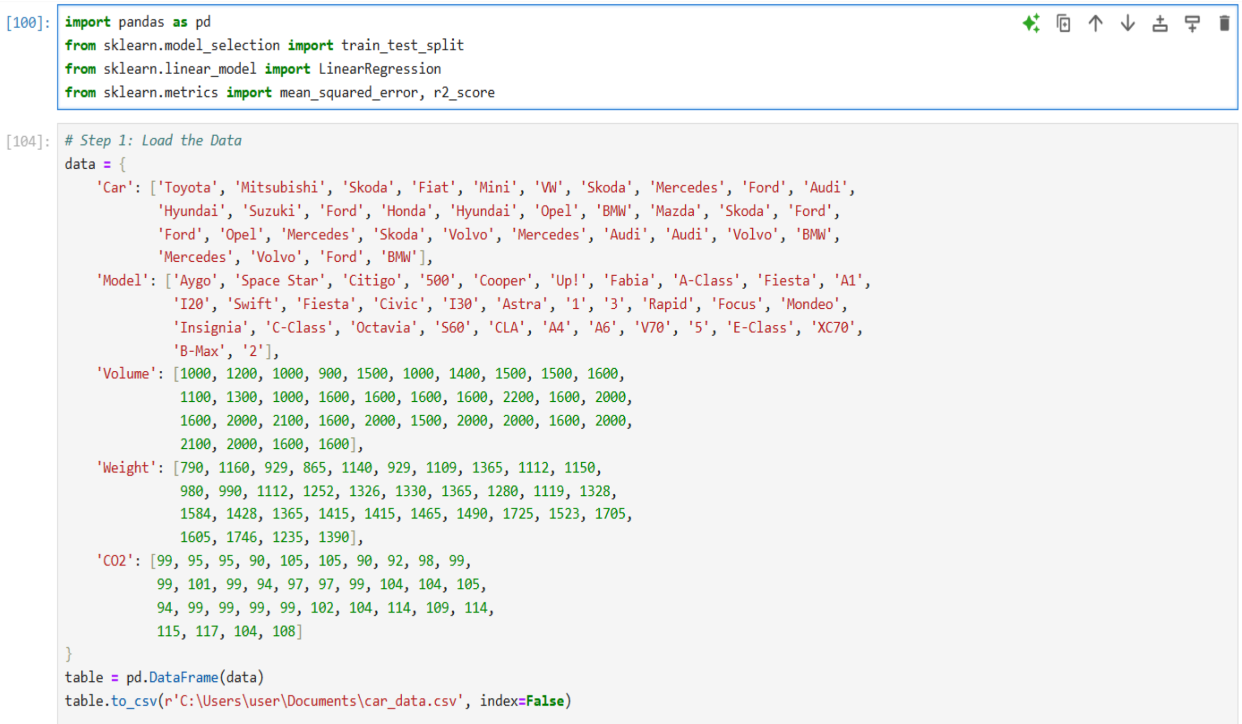
Create a list of the independent values, and then designate this list as variable X. Add the dependent values to the y variable.

Show Python code for reading dataset

Summarizing the Dataset

1. Read the basic Information about the dataset
2. Dimensions of Dataset
3. Listing all top 10 data,
4. Listing all bottom 10 data,
5. View the Statistical Summary

1. This code essentially prepares the car data in a structured format (DataFrame) and exports it to a CSV file for further use.



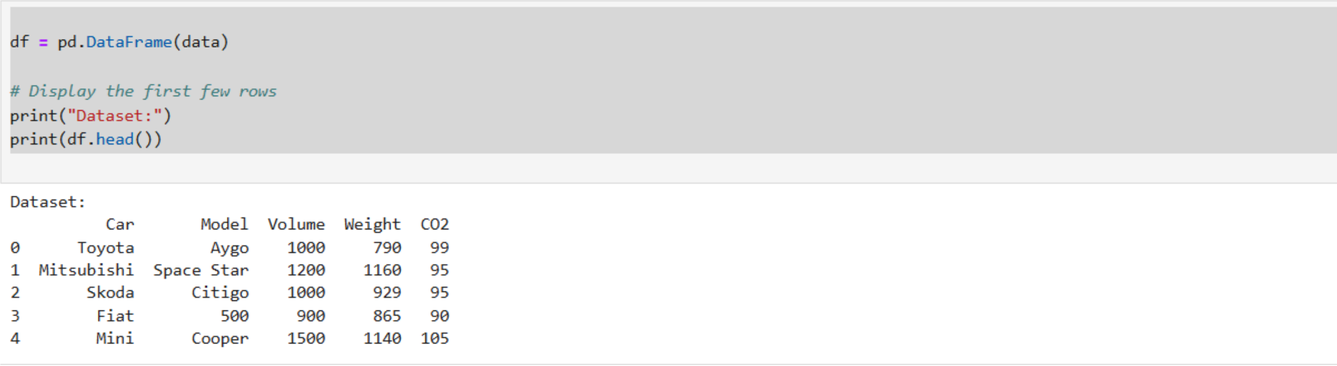
```
[100]: import pandas as pd
       from sklearn.model_selection import train_test_split
       from sklearn.linear_model import LinearRegression
       from sklearn.metrics import mean_squared_error, r2_score
```

```
[104]: # Step 1: Load the Data
       data = {
           'Car': ['Toyota', 'Mitsubishi', 'Skoda', 'Fiat', 'Mini', 'VW', 'Skoda', 'Mercedes', 'Ford', 'Audi',
                   'Hyundai', 'Suzuki', 'Ford', 'Honda', 'Hyundai', 'Opel', 'BMW', 'Mazda', 'Skoda', 'Ford',
                   'Ford', 'Opel', 'Mercedes', 'Skoda', 'Volvo', 'Mercedes', 'Audi', 'Audi', 'Volvo', 'BMW',
                   'Mercedes', 'Volvo', 'Ford', 'BMW'],
           'Model': ['Aygo', 'Space Star', 'Citigo', '500', 'Cooper', 'Up!', 'Fabia', 'A-Class', 'Fiesta', 'A1',
                     'I20', 'Swift', 'Fiesta', 'Civic', 'I30', 'Astra', '1', '3', 'Rapid', 'Focus', 'Mondeo',
                     'Insignia', 'C-Class', 'Octavia', 'S60', 'CLA', 'A4', 'A6', 'V70', '5', 'E-Class', 'XC70',
                     'B-Max', '2'],
           'Volume': [1000, 1200, 1000, 900, 1500, 1000, 1400, 1500, 1500, 1600,
                      1100, 1300, 1000, 1600, 1600, 1600, 1600, 2200, 1600, 2000,
                      1600, 2000, 2100, 1600, 2000, 1500, 2000, 2000, 1600, 2000,
                      2100, 2000, 1600, 1600],
           'Weight': [790, 1160, 929, 865, 1140, 929, 1109, 1365, 1112, 1150,
                      980, 990, 1112, 1252, 1326, 1330, 1365, 1280, 1119, 1328,
                      1584, 1428, 1365, 1415, 1415, 1465, 1490, 1725, 1523, 1705,
                      1605, 1746, 1235, 1390],
           'CO2': [99, 95, 95, 90, 105, 105, 90, 92, 98, 99,
                   99, 101, 99, 94, 97, 97, 99, 104, 104, 105,
                   94, 99, 99, 99, 99, 102, 104, 114, 109, 114,
                   115, 117, 104, 108]
       }
       table = pd.DataFrame(data)
       table.to_csv(r'C:\Users\user\Documents\car_data.csv', index=False)
```

Generated dataset :

car_data 07/11/2024 13:13 365 Suite - Spreadsh... 1 KB

2. The code creates a pandas DataFrame `df` from the `data` dictionary, which contains car information. It then prints the first five rows of the dataset using `df.head()` to provide a quick preview of the data.

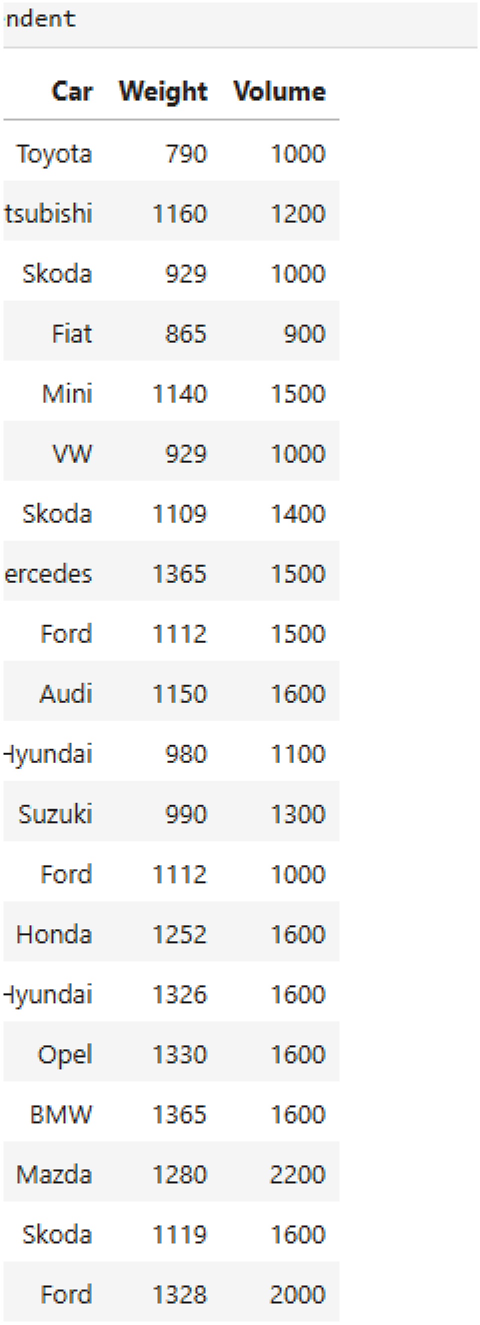


```python
df = pd.DataFrame(data)

# Display the first few rows
print("Dataset:")
print(df.head())
```

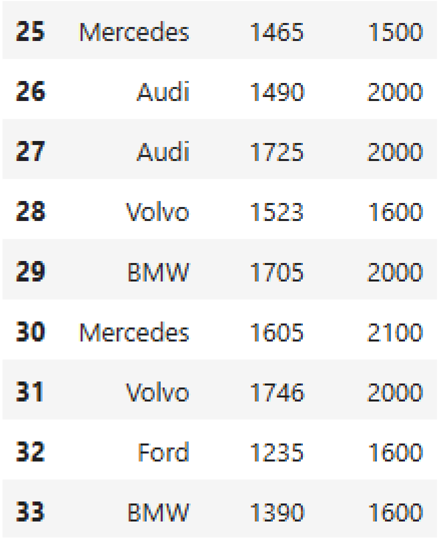
```
Dataset:
          Car       Model  Volume  Weight  CO2
0      Toyota        Aygo    1000     790   99
1  Mitsubishi  Space Star    1200    1160   95
2       Skoda      Citigo    1000     929   95
3        Fiat         500     900     865   90
4        Mini      Cooper    1500    1140  105
```

3. The term "Independent" typically refers to the set of variables in a dataset that are used to predict or explain the dependent variable.



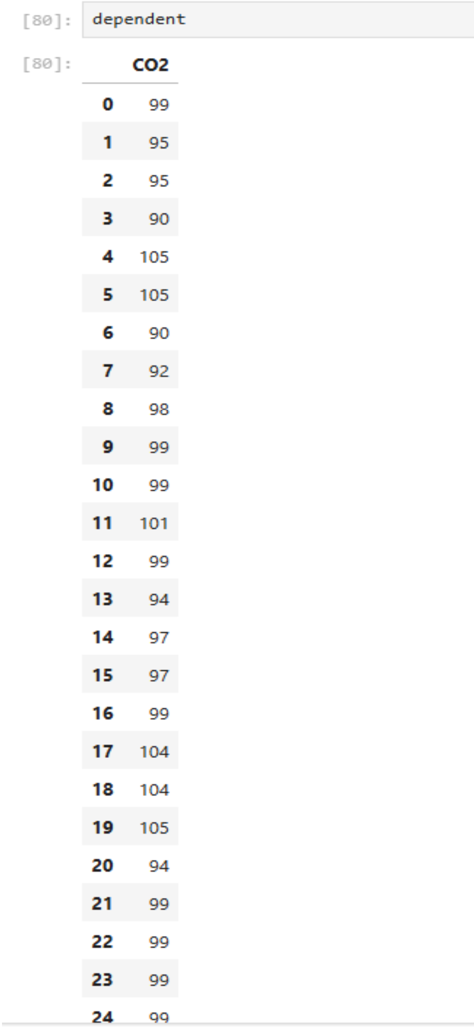
```
ndent
```

Car	Weight	Volume
Toyota	790	1000
tsubishi	1160	1200
Skoda	929	1000
Fiat	865	900
Mini	1140	1500
VW	929	1000
Skoda	1109	1400
ercedes	1365	1500
Ford	1112	1500
Audi	1150	1600
Hyundai	980	1100
Suzuki	990	1300
Ford	1112	1000
Honda	1252	1600
Hyundai	1326	1600
Opel	1330	1600
BMW	1365	1600
Mazda	1280	2200
Skoda	1119	1600
Ford	1328	2000



25	Mercedes	1465	1500
26	Audi	1490	2000
27	Audi	1725	2000
28	Volvo	1523	1600
29	BMW	1705	2000
30	Mercedes	1605	2100
31	Volvo	1746	2000
32	Ford	1235	1600
33	BMW	1390	1600

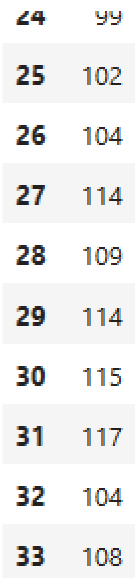
I.I Dependent display



```
[80]: dependent
```

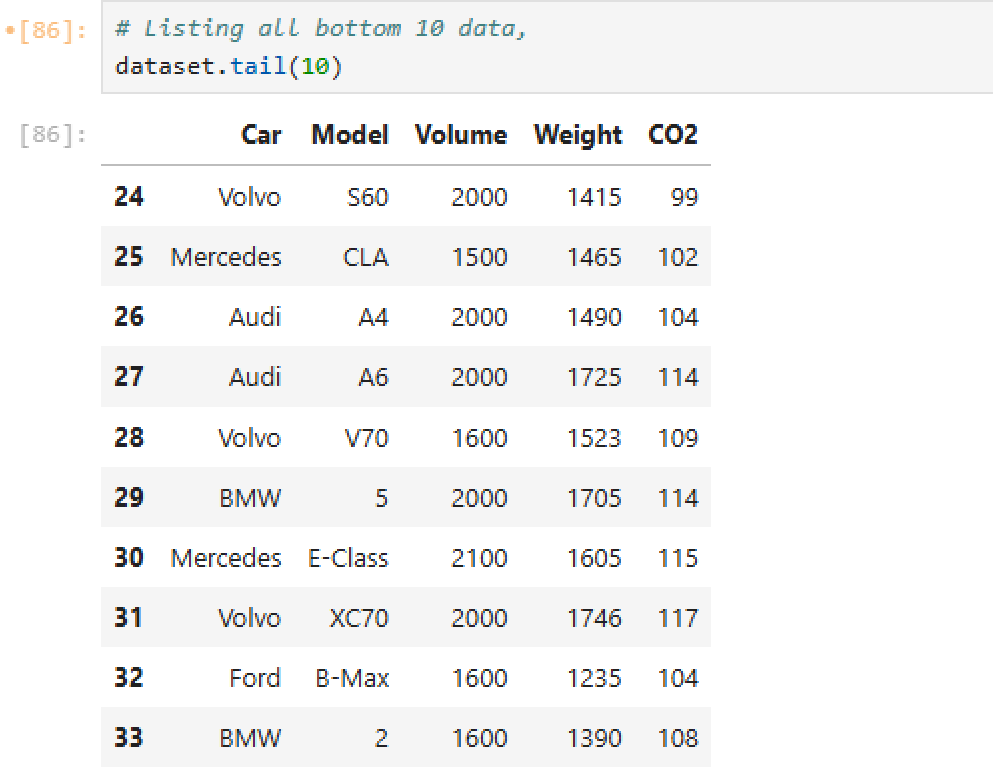
[80]:

	CO2
0	99
1	95
2	95
3	90
4	105
5	105
6	90
7	92
8	98
9	99
10	99
11	101
12	99
13	94
14	97
15	97
16	99
17	104
18	104
19	105
20	94
21	99
22	99
23	99
24	99



24	99
25	102
26	104
27	114
28	109
29	114
30	115
31	117
32	104
33	108

4. The code uses `dataset.tail(10)` to display the last 10 rows of the dataset. This function is useful for reviewing the bottom part of the data, often to check for any outliers or patterns at the end of the dataset.

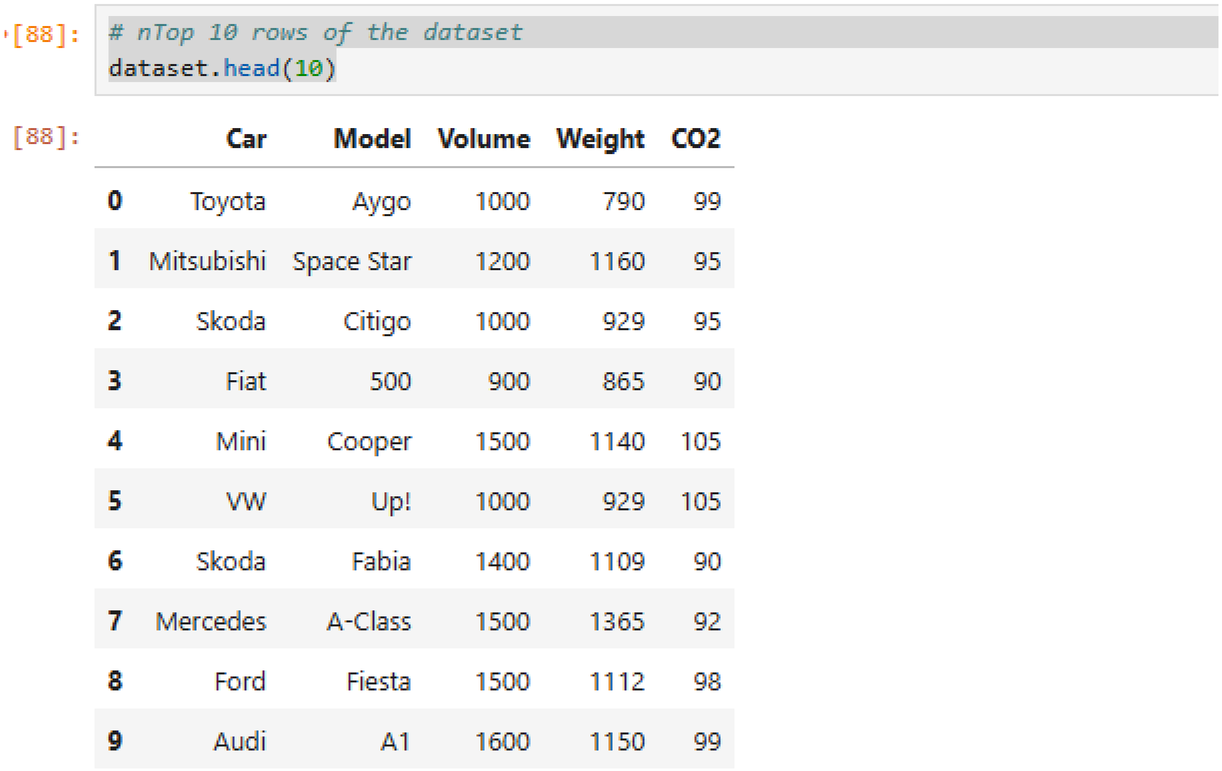


```
•[86]:  # Listing all bottom 10 data,
        dataset.tail(10)
```

[86]:

	Car	Model	Volume	Weight	CO2
24	Volvo	S60	2000	1415	99
25	Mercedes	CLA	1500	1465	102
26	Audi	A4	2000	1490	104
27	Audi	A6	2000	1725	114
28	Volvo	V70	1600	1523	109
29	BMW	5	2000	1705	114
30	Mercedes	E-Class	2100	1605	115
31	Volvo	XC70	2000	1746	117
32	Ford	B-Max	1600	1235	104
33	BMW	2	1600	1390	108

6. The command `dataset.head(10)` returns the first 10 rows of the dataset. It's commonly used to quickly preview the top rows of a DataFrame in Pandas.

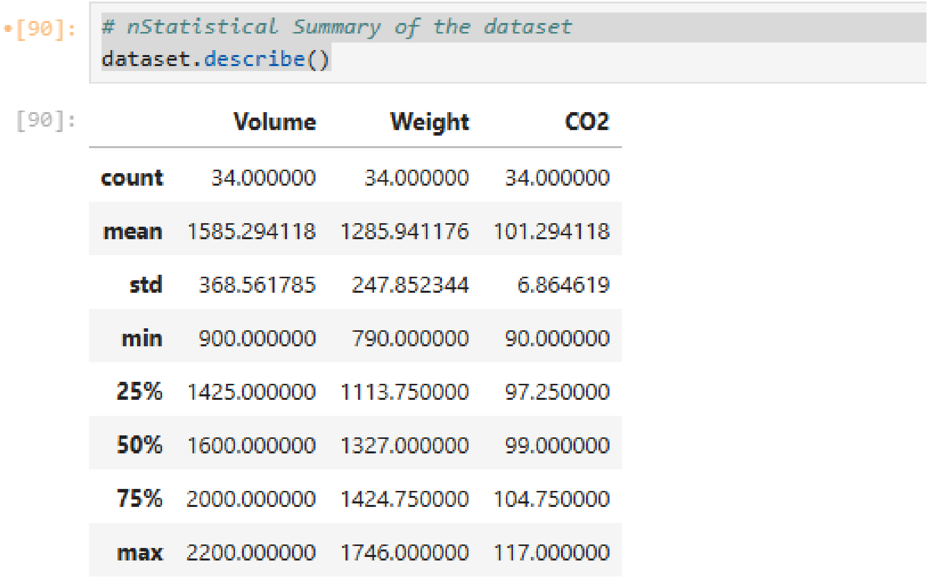


```
[88]: # nTop 10 rows of the dataset
      dataset.head(10)
```

[88]:

	Car	Model	Volume	Weight	CO2
0	Toyota	Aygo	1000	790	99
1	Mitsubishi	Space Star	1200	1160	95
2	Skoda	Citigo	1000	929	95
3	Fiat	500	900	865	90
4	Mini	Cooper	1500	1140	105
5	VW	Up!	1000	929	105
6	Skoda	Fabia	1400	1109	90
7	Mercedes	A-Class	1500	1365	92
8	Ford	Fiesta	1500	1112	98
9	Audi	A1	1600	1150	99

7. The command `dataset.describe()` provides a statistical summary of the dataset.It helps to understand the distribution and spread of the data.



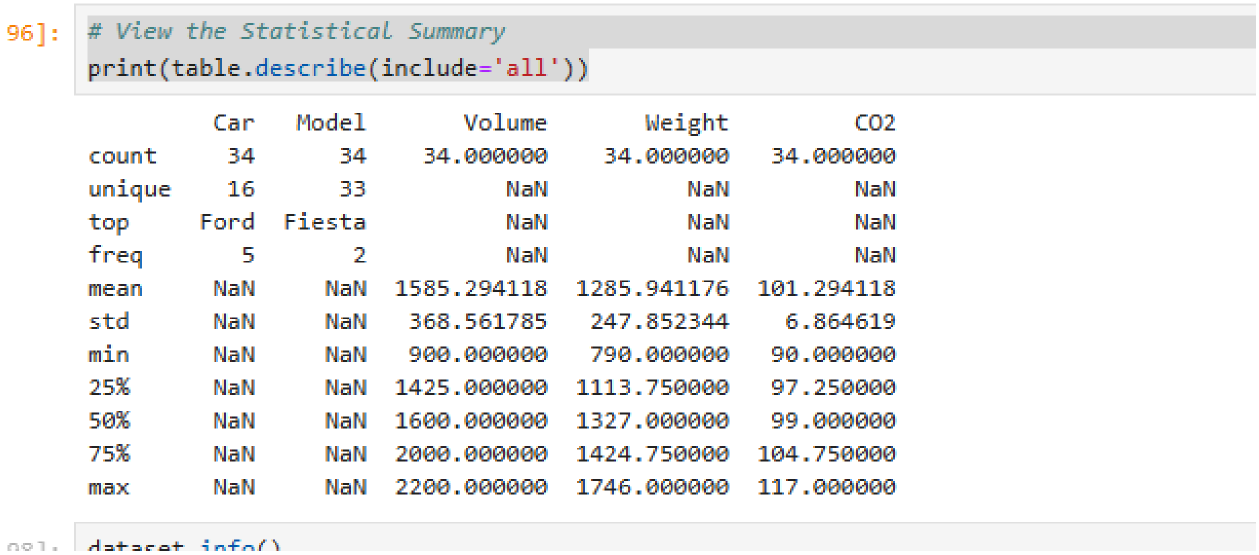
```
[90]: # nStatistical Summary of the dataset
      dataset.describe()
```

[90]:

	Volume	Weight	CO2
count	34.000000	34.000000	34.000000
mean	1585.294118	1285.941176	101.294118
std	368.561785	247.852344	6.864619
min	900.000000	790.000000	90.000000
25%	1425.000000	1113.750000	97.250000
50%	1600.000000	1327.000000	99.000000
75%	2000.000000	1424.750000	104.750000
max	2200.000000	1746.000000	117.000000

8. The command `print(table.describe(include='all'))` generates a statistical summary of the dataset, including both numerical and categorical columns. By specifying

`include='all'`, it provides insights such as count, unique values, top values, frequency, and statistical measures (mean, standard deviation, etc.) for all data types in the table.

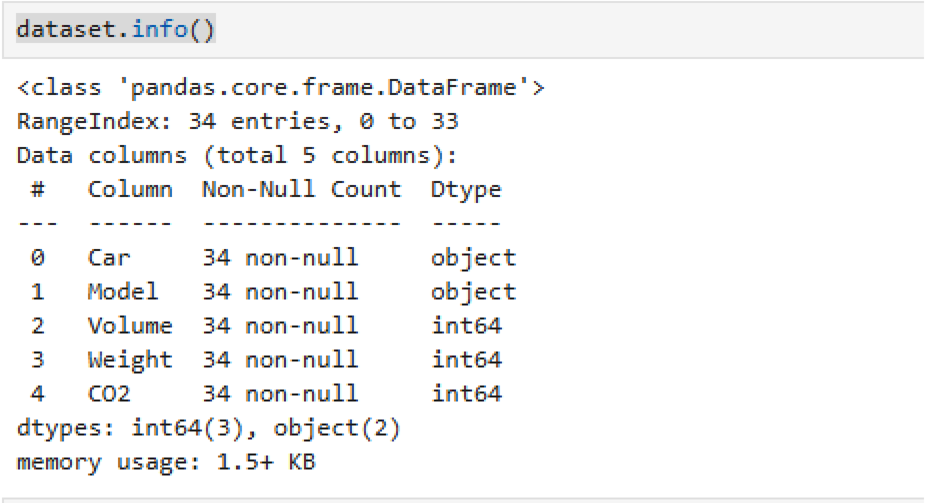


```
96]: # View the Statistical Summary
     print(table.describe(include='all'))

         Car   Model       Volume       Weight         CO2
count     34      34    34.000000    34.000000   34.000000
unique    16      33          NaN          NaN         NaN
top     Ford  Fiesta          NaN          NaN         NaN
freq       5       2          NaN          NaN         NaN
mean     NaN     NaN  1585.294118  1285.941176  101.294118
std      NaN     NaN   368.561785   247.852344    6.864619
min      NaN     NaN   900.000000   790.000000   90.000000
25%      NaN     NaN  1425.000000  1113.750000   97.250000
50%      NaN     NaN  1600.000000  1327.000000   99.000000
75%      NaN     NaN  2000.000000  1424.750000  104.750000
max      NaN     NaN  2200.000000  1746.000000  117.000000

98]: dataset.info()
```

9.`dataset.info()` displays a summary of the dataset, including the number of entries, column names, data types, and non-null counts.



```
dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Car     34 non-null     object
 1   Model   34 non-null     object
 2   Volume  34 non-null     int64 
 3   Weight  34 non-null     int64 
 4   CO2     34 non-null     int64 
dtypes: int64(3), object(2)
memory usage: 1.5+ KB
```