Content: Used Cars Prices

Here you find 78612 records about used cars: 60 Brand, 382 Model, 33 Modelyear, 1839 CarModel, 1397 AveragePrice, 893 MinimumPrice, 916 MaximumPrice, over 128 Months/Year.

Some Questions you should ask yourself about.

1. Is there any duplicates? .
2. What about the nulls?.
3. Does all columns has the a correct format in its values? if its not how should you make it better?
4. Datatypes?
5. Before starting , after seeing some info about the dataset and from the first look on the dataset , what columns you think will not be necessary in our dataset? (io: what columns you think dropping it will be better?).   feel free to wirte only their names in the next cell

Double Click here to start writing

unnecessary columns:

1. web-scraper-order

###1- import Libraries (numpy & Pandas)

```
import numpy as np
import pandas as pd
```

####2- read data

```
df= pd.read_csv('/content/used_car_prices.csv')
```

###3- print the first 8 row

```
df.head(8)

  web-scraper-order                Car Model Month/Year Average price  \
0      1680204632-1  Skoda Octavia A8 2022    2023-03   967,000 EGP
1      1680204632-2  Skoda Octavia A8 2022    2023-02   979,000 EGP
2      1680204632-3  Skoda Octavia A8 2022    2023-01   917,000 EGP
3      1680204632-4  Skoda Octavia A8 2022    2022-12   881,000 EGP
4      1680204632-5  Skoda Octavia A8 2022    2022-11   868,000 EGP
5      1680204632-6  Skoda Octavia A8 2022    2022-10   797,000 EGP
6      1680204632-7  Skoda Octavia A8 2022    2022-09   837,000 EGP
7      1680204632-8  Skoda Octavia A8 2022    2022-08   779,000 EGP

  Minimum price  Maximum price
0   926,000 EGP  1,017,000 EGP
1   931,000 EGP  1,045,000 EGP
2   893,000 EGP    950,000 EGP
3   793,000 EGP    950,000 EGP
```

```
4   789,000 EGP    950,000 EGP
5   789,000 EGP    808,000 EGP
6   770,000 EGP    874,000 EGP
7   722,000 EGP    855,000 EGP
```

###4- print the shape Like (There are 84548 rows and 22 columns)

```
print(f'There are {df.shape[0]} rows and {df.shape[1]} columns')

There are 79090 rows and 6 columns
```

###5- try seeing some information about the data

```
df .info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 79090 entries, 0 to 79089
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   web-scraper-order  79090 non-null  object
 1   Car Model          79090 non-null  object
 2   Month/Year         78612 non-null  object
 3   Average price      78612 non-null  object
 4   Minimum price      78612 non-null  object
 5   Maximum price      78612 non-null  object
dtypes: object(6)
memory usage: 3.6+ MB

# a description of your data

for i in list(df.columns):

    print("\n ************ "+i+" ************\n")
    print("\n",df[i].value_counts())
    print("\n",df[i].describe(),"\n")


 ************ web-scraper-order ************


 1680204632-1        1
1680204632-52725    1
1680204632-52732    1
1680204632-52731    1
1680204632-52730    1
                   ..
1680204632-26363    1
1680204632-26362    1
1680204632-26361    1
```

```
1680204632-26360    1
1680204632-79090    1
Name: web-scraper-order, Length: 79090, dtype: int64

 count            79090
unique           79090
top       1680204632-1
freq                 1
Name: web-scraper-order, dtype: object


 ************ Car Model ************


 Hyundai Excel 1996      128
Chevrolet Cruze 2010    128
Hyundai Verna 2007      128
Daewoo Lanos 2000       128
Chevrolet Optra 2008    128
                       ...
Ford Focus 2022           2
Hyundai I30 2008          2
DFSK Glory 330 2021       2
Audi Q3 2022              2
Ford B-Max 2014           2
Name: Car Model, Length: 1908, dtype: int64

 count                  79090
unique                  1908
top       Hyundai Excel 1996
freq                     128
Name: Car Model, dtype: object


 ************ Month/Year ************


 2022-11    1480
2022-12    1446
2022-10    1388
2023-01    1352
2022-09    1300
           ...
2012-02      84
2012-06      77
2012-08      73
2012-07      72
2012-01      71
Name: Month/Year, Length: 128, dtype: int64
```

```
 count       78612
unique        128
top       2022-11
freq         1480
Name: Month/Year, dtype: object


 ************ Average price ************


 76,000 EGP       593
78,000 EGP       590
74,000 EGP       586
79,000 EGP       570
77,000 EGP       569
                ... 
1,414,000 EGP      1
1,652,000 EGP      1
1,650,000 EGP      1
1,424,000 EGP      1
2,128,000 EGP      1
Name: Average price, Length: 1397, dtype: int64

 count          78612
unique          1397
top       76,000 EGP
freq             593
Name: Average price, dtype: object


 ************ Minimum price ************


 71,000 EGP       1345
105,000 EGP      1227
81,000 EGP       1150
114,000 EGP      1081
90,000 EGP       1059
                 ... 
1,126,000 EGP       1
1,107,000 EGP       1
1,511,000 EGP       1
974,000 EGP         1
1,292,000 EGP       1
Name: Minimum price, Length: 893, dtype: int64

 count          78612
unique           893
top       71,000 EGP
freq            1345
```

```
Name: Minimum price, dtype: object


 ************ Maximum price ************


 81,000 EGP        1260
105,000 EGP       1245
90,000 EGP        1159
95,000 EGP        1075
114,000 EGP       1023
                  ...
1,686,000 EGP        1
1,591,000 EGP        1
1,397,000 EGP        1
1,021,000 EGP        1
1,634,000 EGP        1
Name: Maximum price, Length: 916, dtype: int64

 count          78612
unique            916
top        81,000 EGP
freq             1260
Name: Maximum price, dtype: object
```

###6- show the number of duplicates here

```
df.duplicated().sum()
```

```
0
```

###7- show number of nulls

```
df.isnull().sum()
```

```
web-scraper-order      0
Car Model              0
Month/Year           478
Average price        478
Minimum price        478
Maximum price        478
dtype: int64
```

###8- updated the column names (To Start with Upper Litter)

```
df.columns = df.columns.str.title()
```

###9- Convert Average price, Minimum price, Maximum price to Numerical Values

```
df["Average Price"].fillna(df["Average Price"].mode()[0],
inplace=True)
df["Minimum Price"].fillna(df["Minimum Price"].mode()[0],
inplace=True)
df["Maximum Price"].fillna(df["Maximum Price"].mode()[0],
inplace=True)
df['Average Price'] = df['Average Price'].str.replace('EGP','')
df['Minimum Price'] = df['Minimum Price'].str.replace('EGP','')
df['Maximum Price'] = df['Maximum Price'].str.replace('EGP','')
df['Average Price'] = df['Average Price'].str.replace(',','')
df['Minimum Price'] = df['Minimum Price'].str.replace(',','')
df['Maximum Price'] = df['Maximum Price'].str.replace(',','')
df['Average Price'] = df['Average Price'].astype('int')
df['Minimum Price'] = df['Minimum Price'].astype('int')
df['Maximum Price'] = df['Maximum Price'].astype('int')
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 79090 entries, 0 to 79089
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Web-Scraper-Order  79090 non-null  object
 1   Car Model          79090 non-null  object
 2   Month/Year         78612 non-null  object
 3   Average Price      79090 non-null  int64
 4   Minimum Price      79090 non-null  int64
 5   Maximum Price      79090 non-null  int64
dtypes: int64(3), object(3)
memory usage: 3.6+ MB
```

###10- Split Month/Year Column to Year and Month

```
df[['Year','Month']]= df['Month/Year'].str.split('-',expand= True)
df.head()

  Web-Scraper-Order              Car Model Month/Year  Average
Price  \
0      1680204632-1  Skoda Octavia A8 2022    2023-03        967000

1      1680204632-2  Skoda Octavia A8 2022    2023-02        979000

2      1680204632-3  Skoda Octavia A8 2022    2023-01        917000

3      1680204632-4  Skoda Octavia A8 2022    2022-12        881000

4      1680204632-5  Skoda Octavia A8 2022    2022-11        868000


   Minimum Price  Maximum Price  Year Month
```

```
0         926000        1017000  2023    03
1         931000        1045000  2023    02
2         893000         950000  2023    01
3         793000         950000  2022    12
4         789000         950000  2022    11
```

###11- let's start with (Average price ,Minimum price ,Maximum price) , see the number of nulls in them , replace these nulls with the (mean , median and mode) of these columns and in the end check that there are not nulls in data

```
#already did it above
df.isna().sum()
```

```
Web-Scraper-Order      0
Car Model              0
Month/Year           478
Average Price          0
Minimum Price          0
Maximum Price          0
Year                 478
Month                478
dtype: int64
```