

MINI PROJECT

Exploratory Data Analysis (EDA)

A project report on exploring the dataset

on

Car sales

Submitted by

NIVETHA SREE M

Computer Science and Engineering

Adhiyamaan College Of Engineering, Hosur.

Date: 12/12/2022

Objective:

To Explore the dataset (EDA) .

Dataset: Car Sales.

Link for dataset: <https://www.kaggle.com/datasets/gagandeep16/car-sales>

REPORT:

```
[ ] # Mini Project
# EDA-Exploratory Data Analysis
# Dataset-"https://www.kaggle.com/datasets/gagandeep16/car-sales"
# Topic:Car Sales dataset
```

#11/12/2022

```
✓ 0s import pandas as pd
df = pd.read_csv("/content/Car_sales.csv")
df
```

```
#importing library
#data frame for dataset
```

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wheelbase	Width	Length	Curb_weight	Fuel_capacity	Fuel_
0	Acura	Integra	16.919	16.360	Passenger	21.50	1.8	140.0	101.2	67.3	172.4	2.639	13.2	
1	Acura	TL	39.384	19.875	Passenger	28.40	3.2	225.0	108.1	70.3	192.9	3.517	17.2	
2	Acura	CL	14.114	18.225	Passenger	NaN	3.2	225.0	106.9	70.6	192.0	3.470	17.2	
3	Acura	RL	8.588	29.725	Passenger	42.00	3.5	210.0	114.6	71.4	196.6	3.850	18.0	
4	Audi	A4	20.397	22.255	Passenger	23.99	1.8	150.0	102.6	68.2	178.0	2.998	16.4	
...
152	Volvo	V40	3.545	NaN	Passenger	24.40	1.9	160.0	100.5	67.6	176.6	3.042	15.8	
153	Volvo	S70	15.245	NaN	Passenger	27.50	2.4	168.0	104.9	69.3	185.9	3.208	17.9	
154	Volvo	V70	17.531	NaN	Passenger	28.80	2.4	168.0	104.9	69.3	186.2	3.259	17.9	
155	Volvo	C70	3.493	NaN	Passenger	45.50	2.3	236.0	104.9	71.5	185.7	3.601	18.5	
156	Volvo	S80	18.969	NaN	Passenger	36.00	2.9	201.0	109.9	72.1	189.8	3.600	21.1	

✓ 0s completed at 10:55 PM

● X



```
df.size
```

```
#size of the dataset (or) number of elements in dataframe
```

```
2512
```

```
[ ] df.shape
```

```
#shape of the dataframe(rows,columns)
```

```
(157, 16)
```

```
[ ] df.info()
```

```
#categories or titles of each column (or) the information of the dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 157 entries, 0 to 156
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Manufacturer           157 non-null    object
1   Model                  157 non-null    object
2   Sales_in_thousands     157 non-null    float64
3   __year_resale_value    121 non-null    float64
4   Vehicle_type           157 non-null    object
5   Price_in_thousands    155 non-null    float64
6   Engine_size            156 non-null    float64
7   Horsepower             156 non-null    float64
8   Wheelbase              156 non-null    float64
```

```
[ ] 11 Curb_weight        155 non-null    float64
    12 Fuel_capacity      156 non-null    float64
    13 Fuel_efficiency     154 non-null    float64
    14 Latest_launch       157 non-null    object
    15 Power_perf_factor   155 non-null    float64
dtypes: float64(12), object(4)
memory usage: 19.8+ KB
```

```
[ ] df.isnull().sum()
```

```
#finds the null
```

```
Manufacturer      0
Model              0
Sales_in_thousands  0
__year_resale_value  36
Vehicle_type       0
Price_in_thousands  2
Engine_size        1
Horsepower         1
Wheelbase          1
Width              1
Length             1
Curb_weight        2
Fuel_capacity       1
Fuel_efficiency     3
Latest_launch       0
Power_perf_factor   2
dtype: int64
```

```
[ ] df.iloc[2:17,1:6]
```

```
#slicing of data using index number
```

	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands
2	CL	14.114	18.225	Passenger	NaN

```
✓ 0s completed at 10:55 PM
```

```
[ ] df.iloc[2:17,1:6]
```

```
#slicing of data using index number
```

	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands
2	CL	14.114	18.225	Passenger	NaN
3	RL	8.588	29.725	Passenger	42.000
4	A4	20.397	22.255	Passenger	23.990
5	A6	18.780	23.555	Passenger	33.950
6	A8	1.380	39.000	Passenger	62.000
7	323i	19.747	NaN	Passenger	26.990
8	328i	9.231	28.675	Passenger	33.400
9	528i	17.527	36.125	Passenger	38.900
10	Century	91.561	12.475	Passenger	21.975
11	Regal	39.350	13.740	Passenger	25.300
12	Park Avenue	27.851	20.190	Passenger	31.965
13	LeSabre	83.257	13.360	Passenger	27.885
14	DeVille	63.729	22.525	Passenger	39.895
15	Seville	15.943	27.100	Passenger	44.475
16	Eldorado	6.536	25.725	Passenger	39.665

```
[ ] df.nunique
```

```
#unique names present in the dataframe
```

✓ 0s completed at 10:55 PM

```
df.nunique
```

```
#unique names present in the dataframe
```

```
<bound method DataFrame.nunique of
0      Acura  Integra  16.919  16.360
1      Acura    TL    39.384  19.875
2      Acura    CL    14.114  18.225
3      Acura    RL     8.588  29.725
4      Audi     A4    20.397  22.255
..      ...      ...      ...      ...
152     Volvo   V40     3.545    NaN
153     Volvo   S70    15.245    NaN
154     Volvo   V70    17.531    NaN
155     Volvo   C70     3.493    NaN
156     Volvo   S80    18.969    NaN

Vehicle_type  Price_in_thousands  Engine_size  Horsepower  Wheelbase \
0      Passenger              21.50           1.8        140.0       101.2
1      Passenger              28.40           3.2        225.0       108.1
2      Passenger              NaN           3.2        225.0       106.9
3      Passenger             42.00           3.5        210.0       114.6
4      Passenger             23.99           1.8        150.0       102.6
..      ...      ...      ...      ...      ...
152     Passenger             24.40           1.9        160.0       100.5
153     Passenger             27.50           2.4        168.0       104.9
154     Passenger             28.80           2.4        168.0       104.9
155     Passenger             45.50           2.3        236.0       104.9
156     Passenger             36.00           2.9        201.0       109.9

Width  Length  Curb_weight  Fuel_capacity  Fuel_efficiency  Latest_Launch \
0      67.3   172.4        2.639          13.2            28.0        2/2/2012
1      70.3   192.9        3.517          17.2            25.0        6/3/2011
2      70.6   192.0        3.470          17.2            26.0        1/4/2012
3      71.4   196.6        3.850          18.0            22.0        3/10/2011
4      68.2   178.0        2.998          16.4            27.0        10/8/2011
```

✓ 0s completed at 10:55 PM

```
[ ] df.groupby("Manufacturer",sort = False).size()
```

```
#sorting in Alphabetical order
```

```
Manufacturer
Acura      4
Audi       3
BMW        3
Buick      4
Cadillac   5
Chevrolet  9
Chrysler   7
Dodge     11
Ford      11
Honda      5
Hyundai    3
Infiniti   1
Jaguar     1
Jeep       3
Lexus      6
Lincoln    3
Mitsubishi 7
Mercury    6
Mercedes-B 9
Nissan      7
Oldsmobile 6
Plymouth   4
Pontiac    6
Porsche    3
Saab       2
Saturn     5
Subaru     2
Toyota     9
Volkswagen 6
Volvo      6
```

✓ 0s completed at 10:55 PM

```
[ ] df.Model.value_counts()
```

```
#count of specified(Model) from dataframe
```

```
Neon      2
Integra   1
Cutlass   1
Sentra    1
Altima    1
..
Windstar  1
Expedition 1
Ranger    1
F-Series  1
S80       1
Name: Model, Length: 156, dtype: int64
```

```
[ ] df.Fuel_capacity.value_counts()
```

```
18.5    14
17.0     9
20.0     8
19.0     8
16.0     7
13.2     6
15.9     6
15.0     5
14.5     5
17.5     5
21.1     4
17.2     4
16.4     4
18.0     4
12.1     3
16.6     3
```

✓ 0s completed at 10:55 PM

```
[ ] import numpy as np                                     #importing library
    Lowest_fuel_capacity = np.sum((df['Fuel_capacity'] >= 10)&(df['Fuel_capacity'] < 15))      #divide a column into categories
    Low_fuel_capacity = np.sum((df['Fuel_capacity'] >= 15)&(df['Fuel_capacity'] < 20))          #on the basis of fuel capacity
    Average_fuel_capacity = np.sum((df['Fuel_capacity'] >= 20)&(df['Fuel_capacity'] < 25))
    High_fuel_capacity = np.sum((df['Fuel_capacity'] >= 25)&(df['Fuel_capacity'] < 30))
    print(Lowest_fuel_capacity,Low_fuel_capacity,Average_fuel_capacity,High_fuel_capacity)
```

27 93 23 9

```
[ ] np.min(df.Fuel_capacity)                                #the least fuel holding capacity
```

10.3

```
[ ] np.max(df.Fuel_capacity)                                #the highest fuel holding capacity
```

32.0

```
[ ] #CONCLUSION OF THE EDA
    #data frame for dataset
    #size of the dataset (or) number of elements in dataframe
    #shape of the dataframe(rows,columns)
    #categories or titles of each column (or) the information of the dataframe
    #finds the null
    #slicing of data using index number
    #unique names present in the dataframe
    #sorting in Alphabetical order
    #count of specified(Model) from dataframe
    #divide a column into categories on the basis of fuel capacity
    #the least fuel holding capacity
    #the highest fuel holding capacity
```

✓ Done - completed at 10:55 PM

LINK TO DRIVE: <https://tinyurl.com/rinexpojnivetha>