

LAB 2 (H GROUP) DM ML

```
import pandas as pd

import warnings
warnings.filterwarnings('ignore')

auto = pd.read_csv('/content/Automobile.csv')
```

HEAD OF DATASET

```
auto.head()
```

	symboling	normalized_losses	make	fuel_type	aspiration	number_of_doors	body_style	drive_wheels	engine_location
0	3	168	alfa-romero	gas	std	two	convertible	rwd	front
1	3	168	alfa-romero	gas	std	two	convertible	rwd	front
2	1	168	alfa-romero	gas	std	two	hatchback	rwd	front
3	2	164	audi	gas	std	four	sedan	fwd	front
4	2	164	audi	gas	std	four	sedan	4wd	front

5 rows × 26 columns

ROWS AND COLUMNS

```
auto.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201 entries, 0 to 200
Data columns (total 26 columns):
#   Column                Non-Null Count  Dtype
---  -
0   symboling              201 non-null    int64
1   normalized_losses      201 non-null    int64
2   make                   201 non-null    object
3   fuel_type              201 non-null    object
4   aspiration              201 non-null    object
5   number_of_doors        201 non-null    object
6   body_style             201 non-null    object
7   drive_wheels           201 non-null    object
8   engine_location        201 non-null    object
9   wheel_base             201 non-null    float64
10  length                 201 non-null    float64
11  width                  201 non-null    float64
12  height                 201 non-null    float64
13  curb_weight            201 non-null    int64
14  engine_type            201 non-null    object
15  number_of_cylinders     201 non-null    object
16  engine_size            201 non-null    int64
17  fuel_system            201 non-null    object
18  bore                   201 non-null    float64
19  stroke                 201 non-null    float64
20  compression_ratio      201 non-null    float64
21  horsepower             201 non-null    int64
22  peak_rpm               201 non-null    int64
23  city_mpg               201 non-null    int64
24  highway_mpg            201 non-null    int64
25  price                  201 non-null    int64
dtypes: float64(7), int64(9), object(10)
memory usage: 41.0+ KB
```

AVERAGE PRIZE

```
auto.mean()[-1] #auto.mean() gives mean of all columns. we choose -1 index for last column which is price

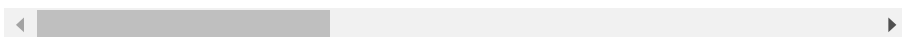
13207.129353233831
```

COSTLIEST CAR

```
value=(auto['price'] == auto['price'].max())
auto[value]
```

	symboling	normalized_losses		make	fuel_type	aspiration	number_of_doors	body
71	1	140		mercedes-benz	gas	std	two	

1 rows × 26 columns

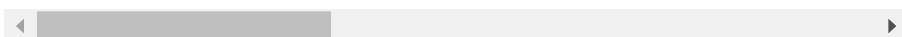


CHEAPEST CAR

```
value=(auto['price'] == auto['price'].min())
auto[value]
```

	symboling	normalized_losses		make	fuel_type	aspiration	number_of_doors	body
134	2	83		subaru	gas	std	two	hatchback

1 rows × 26 columns



number OF CARS WITH HORSEPOWER >100

```
value=(auto['horsepower']>100)
auto[value].count()
```

```
symboling          90
normalized_losses  90
make               90
fuel_type          90
aspiration         90
number_of_doors    90
body_style         90
drive_wheels       90
engine_location    90
wheel_base         90
length            90
width             90
height            90
curb_weight        90
engine_type        90
number_of_cylinders 90
engine_size        90
fuel_system        90
bore               90
stroke            90
compression_ratio  90
horsepower         90
peak_rpm           90
city_mpg           90
highway_mpg        90
price              90
dtype: int64
```

NUMBER OF HATCHBACK CARS

```
value=(auto['body_style']=='hatchback')
auto[value].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 68 entries, 2 to 186
Data columns (total 26 columns):
#   Column              Non-Null Count  Dtype
---  -
0   symboling           68 non-null    int64
1   normalized_losses   68 non-null    int64
2   make                68 non-null    object
3   fuel_type           68 non-null    object
4   aspiration           68 non-null    object
```

```

5  number_of_doors    68 non-null    object
6  body_style         68 non-null    object
7  drive_wheels       68 non-null    object
8  engine_location    68 non-null    object
9  wheel_base         68 non-null    float64
10 length            68 non-null    float64
11 width             68 non-null    float64
12 height            68 non-null    float64
13 curb_weight       68 non-null    int64
14 engine_type       68 non-null    object
15 number_of_cylinders 68 non-null    object
16 engine_size       68 non-null    int64
17 fuel_system       68 non-null    object
18 bore              68 non-null    float64
19 stroke            68 non-null    float64
20 compression_ratio 68 non-null    float64
21 horsepower        68 non-null    int64
22 peak_rpm          68 non-null    int64
23 city_mpg          68 non-null    int64
24 highway_mpg       68 non-null    int64
25 price             68 non-null    int64

```

```

dtypes: float64(7), int64(9), object(10)
memory usage: 14.3+ KB

```

3 MOST COMMON CARS

```
auto['make'].value_counts().head(3)
```

```

toyota    32
nissan     18
mazda     17
Name: make, dtype: int64

```

MAKE OF CAR PURCHASED AT 7099

```

value=(auto['price']== 7099)
auto[value]['make']

```

```

87    nissan
Name: make, dtype: object

```

CAR PRICE>40000

```
auto[auto["price"]>40000]
```

	symboling	normalized_losses	make	fuel_type	aspiration	number_of_doors	body_style	drive_wheels	engine_locati
15	0	149	bmw	gas	std	two	sedan	rwd	fr
70	0	140	mercedes-benz	gas	std	four	sedan	rwd	fr
71	1	140	mercedes-benz	gas	std	two	hardtop	rwd	fr

```
3 rows × 26 columns
```

SEDAN AND PRICE<7000

```
auto[(auto['body_style']=='sedan') & (auto['price']<7000)]
```

	symboling	normalized_losses	make	fuel_type	aspiration	number_of_doors	body_style	drive_wheels	engine_locati
19	0	81	chevrolet	gas	std	four	sedan	fwd	fr
24	1	148	dodge	gas	std	four	sedan	fwd	fr
42	0	110	isuzu	gas	std	four	sedan	rwd	fr
50	1	113	mazda	gas	std	four	sedan	fwd	fr
82	1	125	mitsubishi	gas	std	four	sedan	fwd	fr
86	1	128	nissan	gas	std	two	sedan	fwd	fr
88	1	128	nissan	gas	std	two	sedan	fwd	fr
Double-click (or enter) to edit									
152	0	91	toyota	gas	std	four	sedan	fwd	fr
10 rows × 26 columns									

[Colab paid products](#) - [Cancel contracts here](#)

