


Name: Vinayak V Thayil
Roll No: AM.EN.U4CSE21161

```
#1
import pandas as pd
df1=pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data")
df1
```



	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.60	...	130	mpfi	3.47	2.68	9.00	111	5000	21	27	13495
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	111	5000	21	27	16500
1	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47	9.0	154	5000	19	26	16500
2	2	164	audi	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.40	10.0	102	5500	24	30	13950
3	2	164	audi	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19	3.40	8.0	115	5500	18	22	17450
4	2	?	audi	gas	std	two	sedan	fwd	front	99.8	...	136	mpfi	3.19	3.40	8.5	110	5500	19	25	15250
...
199	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	...	141	mpfi	3.78	3.15	9.5	114	5400	23	28	16845
200	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	...	141	mpfi	3.78	3.15	8.7	160	5300	19	25	19045
201	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	...	173	mpfi	3.58	2.87	8.8	134	5500	18	23	21485
202	-1	95	volvo	diesel	turbo	four	sedan	rwd	front	109.1	...	145	idi	3.01	3.40	23.0	106	4800	26	27	22470
203	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	...	141	mpfi	3.78	3.15	9.5	114	5400	19	25	22625

204 rows × 26 columns

```
#2
import pandas as pd
headers = ["symboling", "normalized-losses", "make", "fuel-type", "aspiration", "num-of-doors", "body-style",
           "drive-wheels", "engine-location", "wheel-base", "length", "width", "height", "curb-weight",
           "engine-type", "num-of-cylinders", "engine-size", "fuel-system", "bore", "stroke", "compression-ratio",
           "horsepower", "peak-rpm", "city-mpg", "highway-mpg", "price"]
print(df1)
```

	symboling	normalized-losses	make	fuel-type	aspiration	\
0	3	122.0	alfa-romero	gas	std	
1	3	122.0	alfa-romero	gas	std	
2	1	122.0	alfa-romero	gas	std	
3	2	164.0	audi	gas	std	
4	2	164.0	audi	gas	std	
..	
200	-1	95.0	volvo	gas	std	
201	-1	95.0	volvo	gas	turbo	
202	-1	95.0	volvo	gas	std	
203	-1	95.0	volvo	diesel	turbo	
204	-1	95.0	volvo	gas	turbo	

	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	\
0	two	convertible	rwd	front	88.6	...	
1	two	convertible	rwd	front	88.6	...	
2	two	hatchback	rwd	front	94.5	...	
3	four	sedan	fwd	front	99.8	...	
4	four	sedan	4wd	front	99.4	...	
..	
200	four	sedan	rwd	front	109.1	...	
201	four	sedan	rwd	front	109.1	...	
202	four	sedan	rwd	front	109.1	...	
203	four	sedan	rwd	front	109.1	...	
204	four	sedan	rwd	front	109.1	...	

	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	\
0	130	mpfi	3.47	2.68	9.0	111.0	
1	130	mpfi	3.47	2.68	9.0	111.0	
2	152	mpfi	2.68	3.47	9.0	154.0	
3	109	mpfi	3.19	3.40	10.0	102.0	
4	136	mpfi	3.19	3.40	8.0	115.0	
..	
200	141	mpfi	3.78	3.15	9.5	114.0	
201	141	mpfi	3.78	3.15	8.7	160.0	
202	173	mpfi	3.58	2.87	8.8	134.0	

203	145	idi	3.01	3.40	23.0	106.0
204	141	mpfi	3.78	3.15	9.5	114.0

	peak-rpm	city-mpg	highway-mpg	price
0	5000.0	21	27	13495.0
1	5000.0	21	27	16500.0
2	5000.0	19	26	16500.0
3	5500.0	24	30	13950.0
4	5500.0	18	22	17450.0
..
200	5400.0	23	28	16845.0
201	5300.0	19	25	19045.0
202	5500.0	18	23	21485.0
203	4800.0	26	27	22470.0
204	5400.0	19	25	22625.0

[205 rows x 26 columns]

```
#3
import pandas as pd
print(df1.head())
```

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	\
0	3	?	alfa-romero	gas	std	two	
1	3	?	alfa-romero	gas	std	two	
2	1	?	alfa-romero	gas	std	two	
3	2	164	audi	gas	std	four	
4	2	164	audi	gas	std	four	

	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	\
0	convertible	rwd	front	88.6	...	130	
1	convertible	rwd	front	88.6	...	130	
2	hatchback	rwd	front	94.5	...	152	
3	sedan	fwd	front	99.8	...	109	
4	sedan	4wd	front	99.4	...	136	

	fuel-system	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	\
0	mpfi	3.47	2.68	9.0	111	5000	21	
1	mpfi	3.47	2.68	9.0	111	5000	21	
2	mpfi	2.68	3.47	9.0	154	5000	19	
3	mpfi	3.19	3.40	10.0	102	5500	24	
4	mpfi	3.19	3.40	8.0	115	5500	18	

	highway-mpg	price
0	27	13495
1	27	16500
2	26	16500
3	30	13950
4	22	17450

[5 rows x 26 columns]

```
#4
print(df1.head())
```

	symboling	normalized-losses	make	fuel-type	aspiration	\
0	3	NaN	alfa-romero	gas	std	
1	3	NaN	alfa-romero	gas	std	
2	1	NaN	alfa-romero	gas	std	
3	2	164.0	audi	gas	std	
4	2	164.0	audi	gas	std	

	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	\
0	two	convertible	rwd	front	88.6	...	
1	two	convertible	rwd	front	88.6	...	
2	two	hatchback	rwd	front	94.5	...	
3	four	sedan	fwd	front	99.8	...	
4	four	sedan	4wd	front	99.4	...	

	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	\
0	130	mpfi	3.47	2.68	9.0	111.0	
1	130	mpfi	3.47	2.68	9.0	111.0	
2	152	mpfi	2.68	3.47	9.0	154.0	
3	109	mpfi	3.19	3.40	10.0	102.0	
4	136	mpfi	3.19	3.40	8.0	115.0	

	peak-rpm	city-mpg	highway-mpg	price
0	5000.0	21	27	13495.0
1	5000.0	21	27	16500.0
2	5000.0	19	26	16500.0

3	5500.0	24	30	13950.0
4	5500.0	18	22	17450.0

[5 rows x 26 columns]

#5

```
missing_values = df1.isna()
print(missing_values)
```

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	\
0	False	True	False	False	False	False	False
1	False	True	False	False	False	False	False
2	False	True	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
..
200	False	False	False	False	False	False	False
201	False	False	False	False	False	False	False
202	False	False	False	False	False	False	False
203	False	False	False	False	False	False	False
204	False	False	False	False	False	False	False

	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	\
0	False	False	False	False	...	False	False
1	False	False	False	False	...	False	False
2	False	False	False	False	...	False	False
3	False	False	False	False	...	False	False
4	False	False	False	False	...	False	False
..
200	False	False	False	False	...	False	False
201	False	False	False	False	...	False	False
202	False	False	False	False	...	False	False
203	False	False	False	False	...	False	False
204	False	False	False	False	...	False	False

	fuel-system	bore	stroke	compression-ratio	horsepower	peak-rpm	\
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
..
200	False	False	False	False	False	False	False
201	False	False	False	False	False	False	False
202	False	False	False	False	False	False	False
203	False	False	False	False	False	False	False
204	False	False	False	False	False	False	False

	city-mpg	highway-mpg	price
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
..
200	False	False	False
201	False	False	False
202	False	False	False
203	False	False	False
204	False	False	False

[205 rows x 26 columns]

#6

```
missing_counts = df1.isna().sum()
print(missing_counts)
```

symboling	0
normalized-losses	41
make	0
fuel-type	0
aspiration	0
num-of-doors	2
body-style	0
drive-wheels	0
engine-location	0
wheel-base	0
length	0
width	0
height	0

```

curb-weight      0
engine-type      0
num-of-cylinders 0
engine-size      0
fuel-system      0
bore             4
stroke          4
compression-ratio 0
horsepower       2
peak-rpm         2
city-mpg         0
highway-mpg      0
price           4
dtype: int64

```

```

#7
columns_with_missing_values = df1.columns[df1.isna().any()].tolist()
print("Columns with at least one missing value:", columns_with_missing_values)

```

```

Columns with at least one missing value: ['num-of-doors', 'bore', 'stroke', 'horsepower', 'peak-rpm', 'price']

```

```

#8
missing_indexes = df1[df1["normalized-losses"].isna()].index
print(missing_indexes)

```

```

Int64Index([], dtype='int64')

```

```

print(df1.columns)

Index(['symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiration',
      'num-of-doors', 'body-style', 'drive-wheels', 'engine-location',
      'wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-type',
      'num-of-cylinders', 'engine-size', 'fuel-system', 'bore', 'stroke',
      'compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg',
      'highway-mpg', 'price'],
      dtype='object')

```

```

#9
mean_normalized_losses = df1["normalized-losses"].mean()
mean_stroke = df1["stroke"].mean()
mean_bore = df1["bore"].mean()
df1["normalized-losses"].fillna(mean_normalized_losses, inplace=True)
df1["stroke"].fillna

```

```

<bound method Series.fillna of 0      2.68
1      2.68
2      3.47
3      3.40
4      3.40
...
200     3.15
201     3.15
202     2.87
203     3.40
204     3.15
Name: stroke, Length: 201, dtype: float64>

```

```

#11
mode_num_of_doors = df1["num-of-doors"].mode()[0]
df1["num-of-doors"].fillna(mode_num_of_doors, inplace=True)
print(df1)

```

```

symboling  normalized-losses  make  fuel-type  aspiration  \
0           3           122.0  alfa-romero    gas         std
1           3           122.0  alfa-romero    gas         std
2           1           122.0  alfa-romero    gas         std
3           2           164.0    audi        gas         std
4           2           164.0    audi        gas         std
..         ...           ...         ...         ...         ...
200        -1           95.0   volvo        gas         std
201        -1           95.0   volvo        gas        turbo
202        -1           95.0   volvo        gas         std
203        -1           95.0   volvo    diesel        turbo
204        -1           95.0   volvo        gas        turbo

```

	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	\
0	two	convertible	rwd	front	88.6	...	
1	two	convertible	rwd	front	88.6	...	
2	two	hatchback	rwd	front	94.5	...	
3	four	sedan	fwd	front	99.8	...	
4	four	sedan	4wd	front	99.4	...	
..	
200	four	sedan	rwd	front	109.1	...	
201	four	sedan	rwd	front	109.1	...	
202	four	sedan	rwd	front	109.1	...	
203	four	sedan	rwd	front	109.1	...	
204	four	sedan	rwd	front	109.1	...	

	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	\
0	130	mpfi	3.47	2.68	9.0	111.0	
1	130	mpfi	3.47	2.68	9.0	111.0	
2	152	mpfi	2.68	3.47	9.0	154.0	
3	109	mpfi	3.19	3.40	10.0	102.0	
4	136	mpfi	3.19	3.40	8.0	115.0	
..	
200	141	mpfi	3.78	3.15	9.5	114.0	
201	141	mpfi	3.78	3.15	8.7	160.0	
202	173	mpfi	3.58	2.87	8.8	134.0	
203	145	idi	3.01	3.40	23.0	106.0	
204	141	mpfi	3.78	3.15	9.5	114.0	

	peak-rpm	city-mpg	highway-mpg	price
0	5000.0	21	27	13495.0
1	5000.0	21	27	16500.0
2	5000.0	19	26	16500.0
3	5500.0	24	30	13950.0
4	5500.0	18	22	17450.0
..
200	5400.0	23	28	16845.0
201	5300.0	19	25	19045.0
202	5500.0	18	23	21485.0
203	4800.0	26	27	22470.0
204	5400.0	19	25	22625.0

[201 rows x 26 columns]

```
#12
df1["horsepower"].fillna(method="ffill", inplace=True)
df1["peak-rpm"].fillna(method="ffill", inplace=True)
print(df1)
```

	symboling	normalized-losses	make	fuel-type	aspiration	\
0	3	122.0	alfa-romero	gas	std	
1	3	122.0	alfa-romero	gas	std	
2	1	122.0	alfa-romero	gas	std	
3	2	164.0	audi	gas	std	
4	2	164.0	audi	gas	std	
..	
200	-1	95.0	volvo	gas	std	
201	-1	95.0	volvo	gas	turbo	
202	-1	95.0	volvo	gas	std	
203	-1	95.0	volvo	diesel	turbo	
204	-1	95.0	volvo	gas	turbo	

	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	\
0	two	convertible	rwd	front	88.6	...	
1	two	convertible	rwd	front	88.6	...	
2	two	hatchback	rwd	front	94.5	...	
3	four	sedan	fwd	front	99.8	...	
4	four	sedan	4wd	front	99.4	...	
..	
200	four	sedan	rwd	front	109.1	...	
201	four	sedan	rwd	front	109.1	...	
202	four	sedan	rwd	front	109.1	...	
203	four	sedan	rwd	front	109.1	...	
204	four	sedan	rwd	front	109.1	...	

	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	\
0	130	mpfi	3.47	2.68	9.0	111.0	
1	130	mpfi	3.47	2.68	9.0	111.0	
2	152	mpfi	2.68	3.47	9.0	154.0	
3	109	mpfi	3.19	3.40	10.0	102.0	
4	136	mpfi	3.19	3.40	8.0	115.0	
..	
200	141	mpfi	3.78	3.15	9.5	114.0	
201	141	mpfi	3.78	3.15	8.7	160.0	

202	173	mpfi	3.58	2.87	8.8	134.0
203	145	idi	3.01	3.40	23.0	106.0
204	141	mpfi	3.78	3.15	9.5	114.0

	peak-rpm	city-mpg	highway-mpg	price
0	5000.0	21	27	13495.0
1	5000.0	21	27	16500.0
2	5000.0	19	26	16500.0
3	5500.0	24	30	13950.0
4	5500.0	18	22	17450.0
..
200	5400.0	23	28	16845.0
201	5300.0	19	25	19045.0
202	5500.0	18	23	21485.0
203	4800.0	26	27	22470.0
204	5400.0	19	25	22625.0

[201 rows x 26 columns]

```
#13
df1.dropna(subset=["price"], inplace=True)
print(df1)
```

	symboling	normalized-losses	make	fuel-type	aspiration	\
0	3	122.0	alfa-romero	gas	std	
1	3	122.0	alfa-romero	gas	std	
2	1	122.0	alfa-romero	gas	std	
3	2	164.0	audi	gas	std	
4	2	164.0	audi	gas	std	
..	
200	-1	95.0	volvo	gas	std	
201	-1	95.0	volvo	gas	turbo	
202	-1	95.0	volvo	gas	std	
203	-1	95.0	volvo	diesel	turbo	
204	-1	95.0	volvo	gas	turbo	

	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	\
0	two	convertible	rwd	front	88.6	...	
1	two	convertible	rwd	front	88.6	...	
2	two	hatchback	rwd	front	94.5	...	
3	four	sedan	fwd	front	99.8	...	
4	four	sedan	4wd	front	99.4	...	
..	
200	four	sedan	rwd	front	109.1	...	
201	four	sedan	rwd	front	109.1	...	
202	four	sedan	rwd	front	109.1	...	
203	four	sedan	rwd	front	109.1	...	
204	four	sedan	rwd	front	109.1	...	

	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	\
0	130	mpfi	3.47	2.68	9.0	111.0	
1	130	mpfi	3.47	2.68	9.0	111.0	
2	152	mpfi	2.68	3.47	9.0	154.0	
3	109	mpfi	3.19	3.40	10.0	102.0	
4	136	mpfi	3.19	3.40	8.0	115.0	
..	
200	141	mpfi	3.78	3.15	9.5	114.0	
201	141	mpfi	3.78	3.15	8.7	160.0	
202	173	mpfi	3.58	2.87	8.8	134.0	
203	145	idi	3.01	3.40	23.0	106.0	
204	141	mpfi	3.78	3.15	9.5	114.0	

	peak-rpm	city-mpg	highway-mpg	price
0	5000.0	21	27	13495.0
1	5000.0	21	27	16500.0
2	5000.0	19	26	16500.0
3	5500.0	24	30	13950.0
4	5500.0	18	22	17450.0
..
200	5400.0	23	28	16845.0
201	5300.0	19	25	19045.0
202	5500.0	18	23	21485.0
203	4800.0	26	27	22470.0
204	5400.0	19	25	22625.0

[201 rows x 26 columns]

```
#14
column_data_types = df1.dtypes
print(column_data_types)
```

```

symboling          int64
normalized-losses  float64
make              object
fuel-type         object
aspiration        object
num-of-doors      object
body-style        object
drive-wheels      object
engine-location   object
wheel-base       float64
length            float64
width             float64
height            float64
curb-weight       int64
engine-type       object
num-of-cylinders  object
engine-size       int64
fuel-system       object
bore              float64
stroke            float64
compression-ratio float64
horsepower        float64
peak-rpm          float64
city-mpg          int64
highway-mpg       int64
price             float64
dtype: object
```

```
#15
df1["bore"] = df1["bore"].astype(float)
df1["stroke"] = df1["stroke"].astype(float)
column_data_types = df1.dtypes
print(column_data_types)
```

```

symboling          int64
normalized-losses  float64
make              object
fuel-type         object
aspiration        object
num-of-doors      object
body-style        object
drive-wheels      object
engine-location   object
wheel-base       float64
length            float64
width             float64
height            float64
curb-weight       int64
engine-type       object
num-of-cylinders  object
engine-size       int64
fuel-system       object
bore              float64
stroke            float64
compression-ratio float64
horsepower        float64
peak-rpm          float64
city-mpg          int64
highway-mpg       int64
price             float64
dtype: object
```

```
#16
columns_to_normalize = ["length", "width", "height"]
for column in columns_to_normalize:
    df1[column] = (df1[column] - df1[column].min()) / (df1[column].max() - df1[column].min())
print(df1[columns_to_normalize])
```

```

      length  width  height
0  0.413433  0.324786  0.083333
1  0.413433  0.324786  0.083333
2  0.449254  0.444444  0.383333
3  0.529851  0.504274  0.541667
4  0.529851  0.521368  0.541667
```

```
..      ...      ...      ...
200  0.711940  0.735043  0.641667
201  0.711940  0.726496  0.641667
202  0.711940  0.735043  0.641667
203  0.711940  0.735043  0.641667
204  0.711940  0.735043  0.641667
```

```
[201 rows x 3 columns]
```