Exercises :

1. Predict price of mercedes that is 4 years old with 45000 mileage
2. Predict price of BMW that is 7 years old with 86000 mileage
3. Display the score (accuracy) of the model

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import joblib
```

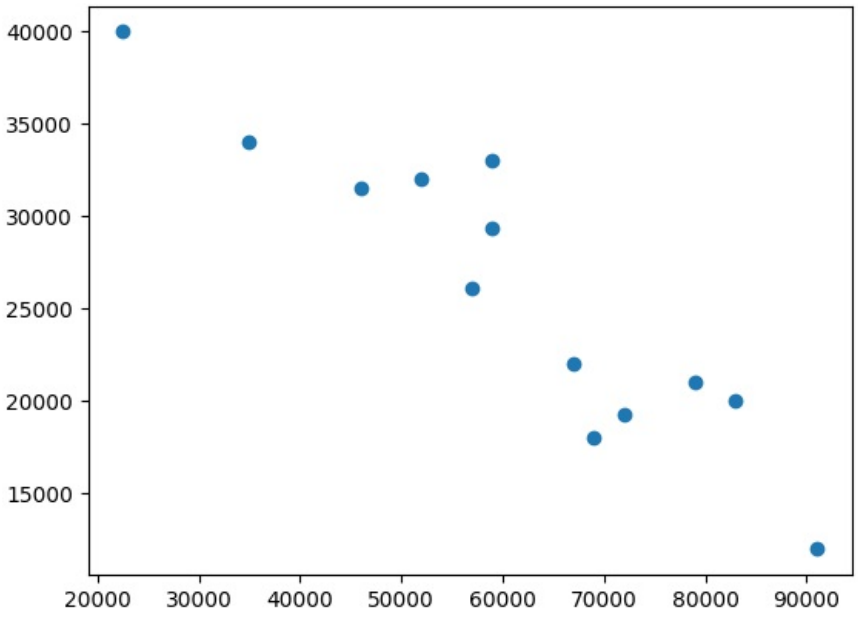
In [3]:
```
data = pd.read_csv('carprices.csv')
data
```

Out[3]:

	Car Model	Mileage	Sell Price($)	Age(yrs)
0	BMW X5	69000	18000	6
1	BMW X5	35000	34000	3
2	BMW X5	57000	26100	5
3	BMW X5	22500	40000	2
4	BMW X5	46000	31500	4
5	Audi A5	59000	29400	5
6	Audi A5	52000	32000	5
7	Audi A5	72000	19300	6
8	Audi A5	91000	12000	8
9	Mercedez Benz C class	67000	22000	6
10	Mercedez Benz C class	83000	20000	7
11	Mercedez Benz C class	79000	21000	7
12	Mercedez Benz C class	59000	33000	5

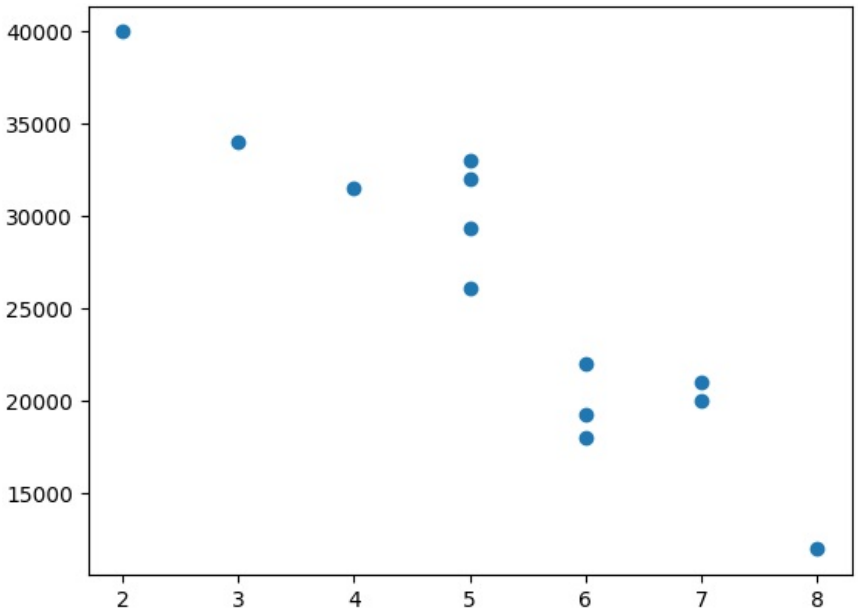
In [6]:
```
%matplotlib inline

plt.scatter(data['Mileage'], data['Sell Price($)'])
```

Out[6]:
```
<matplotlib.collections.PathCollection at 0x1f6d50698d0>
```



In [11]:
```
plt.scatter(data['Age(yrs)'], data['Sell Price($)'])
```

Out[11]:
```
<matplotlib.collections.PathCollection at 0x1f6d9e68490>
```



In [13]:
```
dummies_var = pd.get_dummies(data['Car Model']).astype(int)
dummies_var
```

Out[13]:

	Audi A5	BMW X5	Mercedez Benz C class
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	0	0	1
10	0	0	1
11	0	0	1
12	0	0	1

In [17]:
```
merged_data = pd.concat([data, dummies_var], axis=1)
merged_data
```

Out[17]:

	Car Model	Mileage	Sell Price($)	Age(yrs)	Audi A5	BMW X5	Mercedez Benz C class
0	BMW X5	69000	18000	6	0	1	0
1	BMW X5	35000	34000	3	0	1	0
2	BMW X5	57000	26100	5	0	1	0
3	BMW X5	22500	40000	2	0	1	0
4	BMW X5	46000	31500	4	0	1	0
5	Audi A5	59000	29400	5	1	0	0
6	Audi A5	52000	32000	5	1	0	0
7	Audi A5	72000	19300	6	1	0	0
8	Audi A5	91000	12000	8	1	0	0
9	Mercedez Benz C class	67000	22000	6	0	0	1
10	Mercedez Benz C class	83000	20000	7	0	0	1
11	Mercedez Benz C class	79000	21000	7	0	0	1
12	Mercedez Benz C class	59000	33000	5	0	0	1

In [20]:
```
final_data = merged_data.drop(['Car Model', 'Audi A5'], axis=1)
final_data
```

Out[20]:

	Mileage	Sell Price($)	Age(yrs)	BMW X5	Mercedez Benz C class
0	69000	18000	6	1	0
1	35000	34000	3	1	0
2	57000	26100	5	1	0
3	22500	40000	2	1	0
4	46000	31500	4	1	0
5	59000	29400	5	0	0
6	52000	32000	5	0	0
7	72000	19300	6	0	0
8	91000	12000	8	0	0
9	67000	22000	6	0	1
10	83000	20000	7	0	1
11	79000	21000	7	0	1
12	59000	33000	5	0	1

In [24]:
```
model = LinearRegression()
```

In [27]:
```
X = final_data.drop(['Sell Price($)'], axis=1)
y = final_data['Sell Price($)']
```

In [28]:
```
model.fit(X, y)
```

Out[28]:
```
▾ LinearRegression
LinearRegression()
```

In [32]:
```
model.predict([[45000, 4, 0, 1]])
```

```
C:\Users\User\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:465: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[32]:
```
array([36991.31721061])
```

In [34]:
```
model.predict([[86000, 7, 1, 0]])
```

```
C:\Users\User\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:465: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[34]:
```
array([11080.74313219])
```

In [35]:
```
model.score(X, y)
```

Out[35]:
```
0.9417050937281082
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js