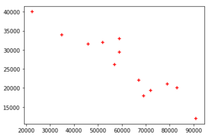
In [21]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

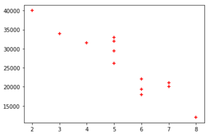
In [43]:
```
%matplotlib inline
plt.scatter(new_df['Mileage'],new_df['Sell Price($)'], color='red', marker='+')
```

Out[43]:
```
<matplotlib.collections.PathCollection at 0x16387bd3550>
```



In [44]:
```
%matplotlib inline
plt.scatter(new_df['Age(yrs)'],new_df['Sell Price($)'], color='red', marker='+')
```

Out[44]:
```
<matplotlib.collections.PathCollection at 0x1638c091430>
```



In [7]:
```
df=pd.read_csv('E:/machine learning/carprices.csv')
```

In [8]:
```
df
```

Out[8]:

	CarModel	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 [9]:
```
dummy=pd.get_dummies(df.CarModel)
```

In [10]:
```
dummy
```

Out[10]:

	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 [14]:
```
merge=pd.concat([df,dummy] ,axis='columns')
```

In [15]:
```
merge
```

Out[15]:

	CarModel	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 [18]:
```
new_df=merge.drop(['CarModel','Mercedez Benz C class'],axis='columns')
```

In [19]:
```
new_df
```

Out[19]:

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

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

In [25]:
```
x=new_df.drop(['Sell Price($)'],axis='columns')
```

In [26]:
```
x
```

Out[26]:

	Mileage	Age(yrs)	Audi A5	BMW X5
0	69000	6	0	1
1	35000	3	0	1
2	57000	5	0	1
3	22500	2	0	1
4	46000	4	0	1
5	59000	5	1	0
6	52000	5	1	0
7	72000	6	1	0
8	91000	8	1	0
9	67000	6	0	0
10	83000	7	0	0
11	79000	7	0	0
12	59000	5	0	0

In [27]:
```
y=new_df['Sell Price($)']
```

In [28]:
```
y
```

Out[28]:
```
0     18000
1     34000
2     26100
3     40000
4     31500
5     29400
6     32000
7     19300
8     12000
9     22000
10    20000
11    21000
12    33000
Name: Sell Price($), dtype: int64
```

In [29]:
```
model.fit(x,y)
```

Out[29]:
```
LinearRegression()
```

In [35]:
```
model.predict([[75000,3,1,0]]) #predicts the selling price of an audi a5 which has mileage 75000 and has run for 3 years
```

Out[35]:
```
array([24766.56726931])
```

In [34]:
```
model.predict([[50000,1,0,1]]) #predicts the selling price of a bmw x5 which has mileage 50000 and has run for 1 year
```

Out[34]:
```
array([32399.86029584])
```

In [33]:
```
model.predict([[33000,5,0,0]]) #predicts the selling price of a mercedez benz c class which has mileage 33000 and has run for 5 years
```

Out[33]:
```
array([40100.32871566])
```

In [36]:
```
model.predict([[11500,9,1,0]])
```

Out[36]:
```
array([40274.5984973])
```

In [37]:
```
model.predict([[77071,5,0,0]])
```

Out[37]:
```
array([23788.67789387])
```

In [38]:
```
model.predict([[25000,5,1,0]]) #predicts the selling price of an audi a5 which has mileage 25000 and has run for 5 years
```

Out[38]:
```
array([40607.76473351])
```

In [39]:
```
model.predict([[55016,3,0,1]]) #predict the selling price of a bmw x5 which has mileage 55016 and has run for 3 years
```

Out[39]:
```
array([27878.42061517])
```

In [40]:
```
model.score(x,y) #gives the accuracy of the model out of 1
```

Out[40]:
```
0.9417050937281082
```