Mahi Prashant Nakhate

10

In [39]:
```
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

In [40]:
```
dataset = pd.read_csv("house dataset.csv")
dataset.head(10)
```

Out[40]:

	area	price
0	8450	208500
1	9600	181500
2	11250	223500
3	9550	140000
4	14260	250000
5	14115	143000
6	10084	307000
7	10382	200000
8	6120	129900
9	7420	118000

In [47]:
```
dataset.tail()
```

Out[47]:

	area	price
1455	7917	175000
1456	13175	210000
1457	9042	266500
1458	9717	142125
1459	9937	147500

In [42]:
```
dataset.shape
```

Out[42]:
```
(1460, 2)
```

In [15]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   area    1460 non-null   int64
 1   price   1460 non-null   int64
dtypes: int64(2)
memory usage: 22.9 KB
```

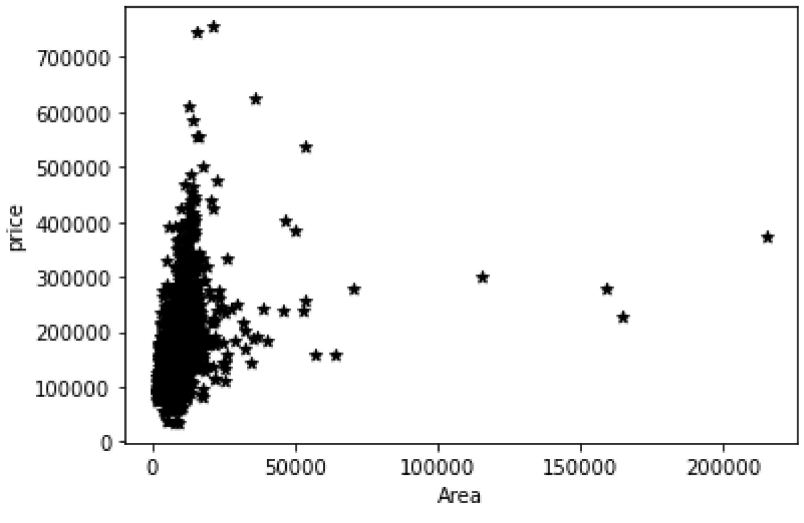
In [43]:
```
dataset.describe
```

Out[43]:
```
<bound method NDFrame.describe of        area   price
0      8450  208500
1      9600  181500
2     11250  223500
3      9550  140000
4     14260  250000
...     ...     ...
1455   7917  175000
1456  13175  210000
1457   9042  266500
1458   9717  142125
1459   9937  147500

[1460 rows x 2 columns]>
```

In [49]:
```
plt.xlabel('Area')
plt.ylabel('price')
plt.scatter(dataset.area, dataset.price, color='black', marker='*')
```

Out[49]: <matplotlib.collections.PathCollection at 0x22b055fe820>



In [50]:
```
X = dataset.drop("price",axis="columns")
X
```

Out[50]:

	area
0	8450
1	9600
2	11250
3	9550
4	14260
...	...
1455	7917
1456	13175
1457	9042
1458	9717
1459	9937

1460 rows × 1 columns

In [51]:
```
Y = dataset.price
Y
```

Out[51]:
```
0       208500
1       181500
2       223500
3       140000
4       250000
         ...
1455    175000
1456    210000
1457    266500
1458    142125
1459    147500
Name: price, Length: 1460, dtype: int64
```

In [52]:
```
model = LinearRegression()
model.fit(X,Y)
```

Out[52]: LinearRegression()

x=2564 LandAreainSqFt = [[x]] PredictedmodelResult = model.predict(LandAreainSqFt) print(PredictedmodelResult)

In [43]:
```
m=model.coef_
print(m)
```

```
[2.09997195]
```

In [42]:
```
b=model.intercept_
print(b)
```

```
158836.1518968766
```

In [44]:
```
y = m*x + b
print("the price of {0} Square feet Land is : {1}".format(x,y[0]))
```

```
the price of 2564 Square feet Land is : 164220.47998105508
```

In [46]:
```
import pandas as pd
from sklearn.linear_model import LinearRegression
```

Type *Markdown* and LaTeX: α^2

PART B-EXAM marks

In [ ]:
```
import pandas as pd
from sklearn.linear_model import LinearRegresion
```

In [54]:
```
dataset = pd.read_csv('exam data.csv')
dataset.head()
```

Out[54]:

	hours	age	internet	marks
0	6.83	15	1	78.50
1	6.56	16	0	76.74
2	NaN	17	1	78.68
3	5.67	18	0	71.82
4	8.67	19	1	84.19

In [55]:
```
print(dataset.shape)
print(dataset.head(5))
```

```
(201, 4)
   hours  age  internet  marks
0   6.83   15         1  78.50
1   6.56   16         0  76.74
2    NaN   17         1  78.68
3   5.67   18         0  71.82
4   8.67   19         1  84.19
```

In [56]:
```
X = dataset.iloc[:,:-1].values
print(X.shape)
X
```

```
(201, 3)
```

Out[56]:
```
array([[ 6.83, 15.  ,  1.  ],
       [ 6.56, 16.  ,  0.  ],
       [  nan, 17.  ,  1.  ],
       [ 5.67, 18.  ,  0.  ],
       [ 8.67, 19.  ,  1.  ],
       [ 7.55, 20.  ,  0.  ],
       [ 6.67, 15.  ,  0.  ],
       [ 8.99, 16.  ,  0.  ],
       [ 5.19, 17.  ,  1.  ],
       [ 6.75, 18.  ,  0.  ],
       [ 6.59, 19.  ,  0.  ],
       [ 8.56, 20.  ,  1.  ],
       [ 7.75, 15.  ,  0.  ],
       [ 7.9 , 16.  ,  1.  ],
       [ 8.19, 17.  ,  0.  ],
       [ 6.55, 18.  ,  1.  ],
       [ 6.36, 19.  ,  0.  ],
       [ 8.44, 20.  ,  1.  ],
```

In [64]:
```
dataset.columns[dataset.isna().any()]
```

Out[64]:
```
Index(['hours'], dtype='object')
```

In [69]:
```
dataset.hours = dataset.hours.fillna(dataset.hours.mean())
```

In [73]:
```
X = dataset.iloc[:,:-1].values
print(X.shape)
X
```

```
(201, 3)
```

Out[73]:
```
array([[ 6.83      , 15.        ,  1.        ],
       [ 6.56      , 16.        ,  0.        ],
       [ 6.98142857, 17.        ,  1.        ],
       [ 5.67      , 18.        ,  0.        ],
       [ 8.67      , 19.        ,  1.        ],
       [ 7.55      , 20.        ,  0.        ],
       [ 6.67      , 15.        ,  0.        ],
       [ 8.99      , 16.        ,  0.        ],
       [ 5.19      , 17.        ,  1.        ],
       [ 6.75      , 18.        ,  0.        ],
       [ 6.59      , 19.        ,  0.        ],
       [ 8.56      , 20.        ,  1.        ],
       [ 7.75      , 15.        ,  0.        ],
       [ 7.9       , 16.        ,  1.        ],
       [ 8.19      , 17.        ,  0.        ],
       [ 6.55      , 18.        ,  1.        ],
       [ 6.36      , 19.        ,  0.        ],
       [ 8.44      , 20.        ,  1.        ]
```

In [74]:
```
Y = dataset.iloc[:, -1].values
Y
```

Out[74]:
```
array([78.5 , 76.74, 78.68, 71.82, 84.19, 81.18, 76.99, 85.46, 70.66,
       77.82, 75.37, 83.88, 79.5 , 80.76, 83.08, 76.03, 76.04, 85.11,
       82.5 , 80.58, 82.18, 83.36, 70.67, 75.02, 70.96, 83.33, 74.75,
       75.65, 74.15, 80.17, 82.27, 76.14, 71.1 , 84.35, 83.08, 76.76,
       81.24, 78.21, 73.08, 83.23, 70.27, 86.41, 71.1 , 82.84, 82.38,
       72.96, 77.46, 70.11, 72.38, 71.41, 72.22, 77.77, 84.44, 71.45,
       82.21, 85.48, 75.03, 86.65, 70.9 , 71.7 , 73.61, 79.41, 76.19,
       80.43, 85.78, 70.06, 81.25, 81.7 , 69.27, 82.79, 71.8 , 71.79,
       74.97, 78.61, 77.59, 72.33, 72.08, 77.33, 70.05, 73.34, 84.  ,
       82.93, 76.63, 75.36, 77.29, 72.87, 73.4 , 81.74, 71.85, 84.6 ,
       79.56, 82.1 , 72.08, 79.1 , 81.01, 76.48, 75.39, 68.57, 83.64,
       82.3 , 75.18, 82.03, 82.99, 79.26, 77.55, 77.07, 72.1 , 73.25,
       74.25, 70.58, 81.08, 75.04, 76.38, 80.86, 78.42, 74.44, 70.34,
       85.04, 73.61, 75.55, 76.2 , 82.69, 76.83, 79.53, 83.57, 85.95,
       76.02, 77.65, 77.01, 74.49, 73.19, 71.86, 75.8 , 72.46, 78.39,
       83.48, 83.15, 71.22, 85.98, 83.91, 84.58, 80.31, 82.55, 75.52,
       83.82, 85.15, 82.75, 74.34, 82.02, 86.12, 71.87, 76.7 , 81.7 ,
       70.78, 78.45, 70.2 , 83.37, 75.52, 81.57, 80.72, 80.81, 79.49,
       79.17, 77.07, 82.04, 71.94, 81.6 , 70.79, 82.68, 83.08, 71.18,
```

In [75]:
```
model = LinearRegression()
model.fit(X,Y)
```

Out[75]:
```
LinearRegression()
```

In [76]:
```
a = [[10,12,0]]
predictedmodelResult=model.predict(a)
print(predictedmodelResult)
```

```
[89.45790294]
```

In [77]:
```
dataset.describe()
```

Out[77]:

	hours	age	internet	marks
count	201.000000	201.000000	201.000000	201.000000
mean	6.981429	17.467662	0.552239	77.951244
std	1.250338	1.720523	0.498505	4.919626
min	4.150000	15.000000	0.000000	68.570000
25%	5.790000	16.000000	0.000000	73.400000
50%	6.981429	17.000000	1.000000	77.770000
75%	8.070000	19.000000	1.000000	82.300000
max	8.990000	20.000000	1.000000	86.990000

In [ ]:

In [ ]: