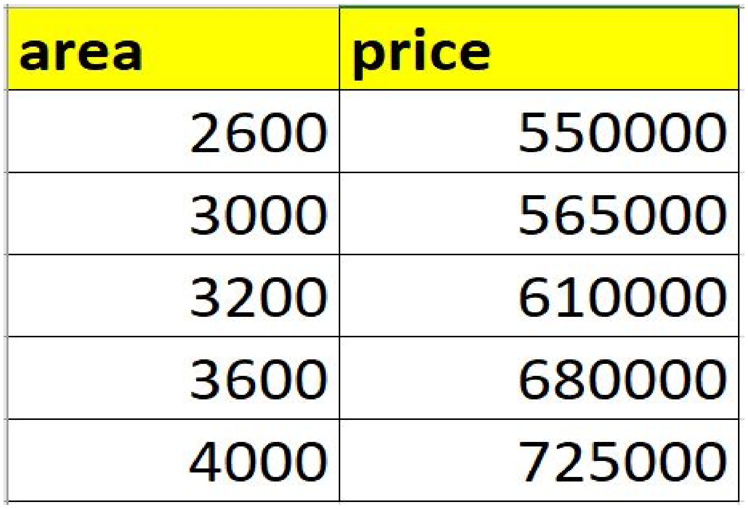
Machine Learning With Python: Linear Regression With One Variable

Sample problem of predicting home price in monroe, new jersey (USA)

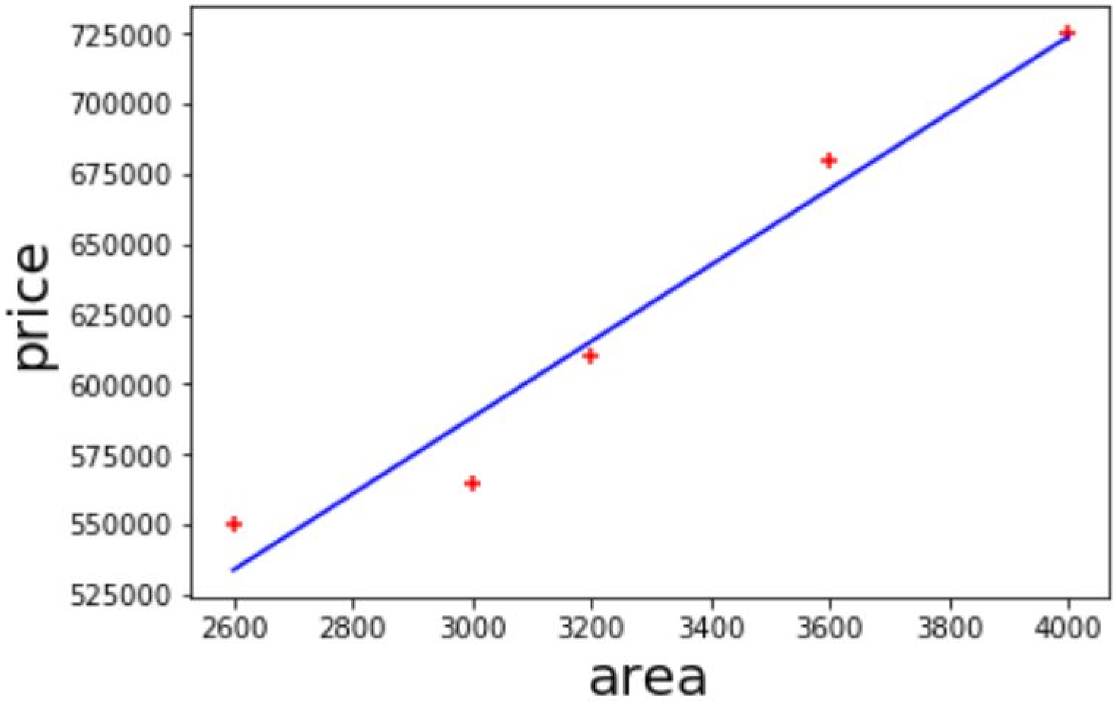
Below table represents current home prices in monroe township based on square feet area, new jersey



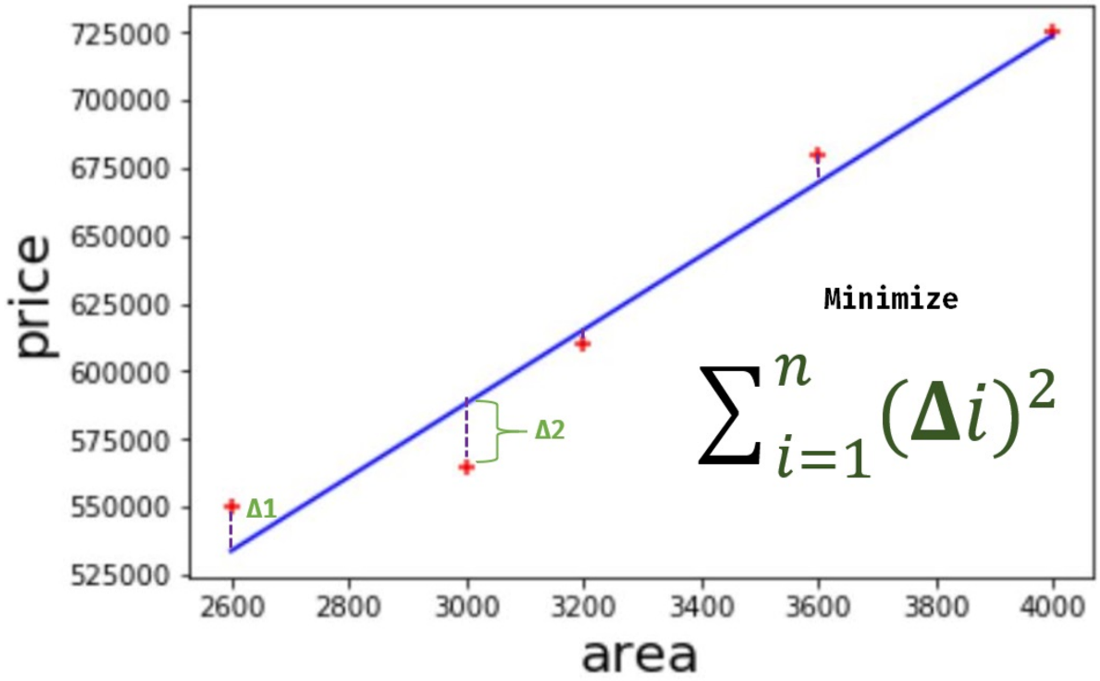
area	price
2600	550000
3000	565000
3200	610000
3600	680000
4000	725000

Problem Statement: Given above data build a machine learning model that can predict home prices based on square feet area

You can represent values in above table as a scatter plot (values are shown in red markers). After that one can draw a straight line that best fits values on chart.



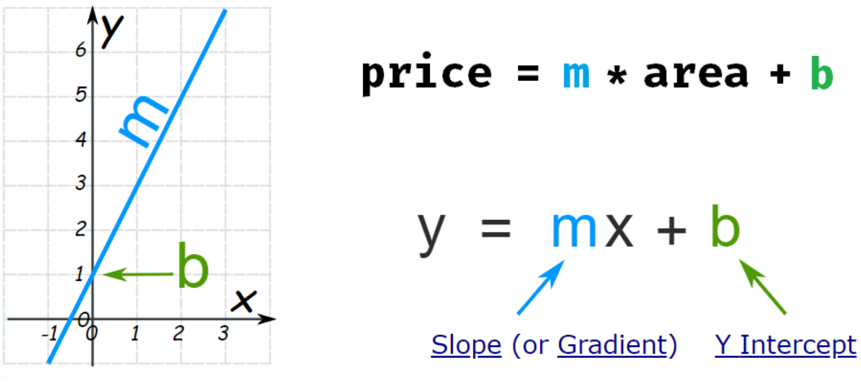
You can draw multiple lines like this but we choose the one where total sum of error is minimum



You might remember about linear equation from your high school days math class. Home prices can be presented as following equation,

home price = m * (area) + b

Generic form of same equation is,



Reference: https://www.mathsisfun.com/algebra/linear-equations.html

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

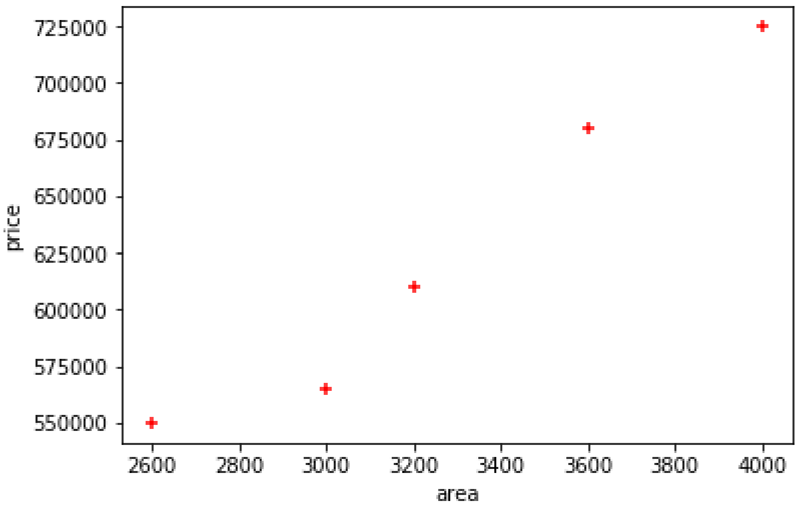
```
In [2]: df = pd.read_csv('homeprices.csv')
        df
```

Out[2]:

	area	price
0	2600	550000
1	3000	565000
2	3200	610000
3	3600	680000
4	4000	725000

In [3]:
```
%matplotlib inline
plt.xlabel('area')
plt.ylabel('price')
plt.scatter(df.area,df.price,color='red',marker='+')
```

Out[3]:
```
<matplotlib.collections.PathCollection at 0x25c8eb78d68>
```



In [5]:
```
new_df = df.drop('price',axis='columns')
new_df
```

Out[5]:

	area
0	2600
1	3000
2	3200
3	3600
4	4000

In [8]:

```
price = df.price
price
```

Out[8]:

```
0    550000
1    565000
2    610000
3    680000
4    725000
Name: price, dtype: int64
```

In [9]:

```
# Create linear regression object
reg = linear_model.LinearRegression()
reg.fit(new_df,price)
```

Out[9]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

(1) Predict price of a home with area = 3300 sqr ft

In [10]:

```
reg.predict([[3300]])
```

Out[10]:

```
array([628715.75342466])
```

In [11]:

```
reg.coef_
```

Out[11]:

```
array([135.78767123])
```

In [12]:

```
reg.intercept_
```

Out[12]:

```
180616.43835616432
```

Y = m * X + b (m is coefficient and b is intercept)

In [13]:

```
3300*135.78767123 + 180616.43835616432
```

Out[13]:

```
628715.7534151643
```

(1) Predict price of a home with area = 5000 sqr ft

In [14]:

```
reg.predict([[5000]])
```

Out[14]:

```
array([859554.79452055])
```

Generate CSV file with list of home price predictions

In [15]:

```
area_df = pd.read_csv("areas.csv")
area_df.head(3)
```

Out[15]:

	area
0	1000
1	1500
2	2300

In [16]:

```
p = reg.predict(area_df)
p
```

Out[16]:

```
array([ 316404.10958904,  384297.94520548,  492928.08219178,
        661304.79452055,  740061.64383562,  799808.21917808,
        926090.75342466,  650441.78082192,  825607.87671233,
        492928.08219178, 1402705.47945205, 1348390.4109589 ,
       1144708.90410959])
```

In [17]:

```
area_df['prices']=p
area_df
```

Out[17]:

	area	prices
0	1000	3.164041e+05
1	1500	3.842979e+05
2	2300	4.929281e+05
3	3540	6.613048e+05
4	4120	7.400616e+05
5	4560	7.998082e+05
6	5490	9.260908e+05
7	3460	6.504418e+05
8	4750	8.256079e+05
9	2300	4.929281e+05
10	9000	1.402705e+06
11	8600	1.348390e+06
12	7100	1.144709e+06

In [18]:

```
area_df.to_csv("prediction.csv")
```

Exercise

Predict canada's per capita income in year 2020. There is an exercise folder here on github at same level as this notebook, download that and you will find canada_per_capita_income.csv file. Using this build a regression model and predict the per capita income fo canadian citizens in year 2020

Answer

41288.69409442