```
import io
import sys
import pandas as pd
from google.colab import files
```

```
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving HousePrices canada.csv to HousePrices canada.csv
```

```
df2 = pd.read_csv(io.BytesIO(uploaded['HousePrices canada.csv']))
```

```
print("There are",len(df2.columns),"columns:")
```

```
There are 13 columns:
```

```
for x in df2.columns:
 sys.stdout.write(str(x)+", ")
```

```
Unnamed: 0, price, lotsize, bedrooms, bathrooms, stories, driveway, recreation, fullbase, gasheat, aircon, garage, prefer,
```

```
print("\n************")
print("Dataset Info:")
print(df2.info())
print("\n************")
print(df2)
print("\n************")
```

```

************
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 546 entries, 0 to 545
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  546 non-null    int64  
 1   price       546 non-null    float64
 2   lotsize     546 non-null    int64  
 3   bedrooms    546 non-null    int64  
 4   bathrooms   546 non-null    int64  
 5   stories     546 non-null    int64  
 6   driveway    546 non-null    object 
 7   recreation  546 non-null    object 
 8   fullbase    546 non-null    object 
 9   gasheat     546 non-null    object 
 10  aircon      546 non-null    object 
 11  garage      546 non-null    int64  
 12  prefer      546 non-null    object 
dtypes: float64(1), int64(6), object(6)
memory usage: 55.6+ KB
None

************
     Unnamed: 0     price  lotsize  bedrooms  ...  gasheat  aircon garage prefer
0             1   42000.0     5850         3  ...       no      no      1     no
1             2   38500.0     4000         2  ...       no      no      0     no
2             3   49500.0     3060         3  ...       no      no      0     no
3             4   60500.0     6650         3  ...       no      no      0     no
4             5   61000.0     6360         2  ...       no      no      0     no
..          ...       ...      ...       ...  ...      ...     ...    ...    ...
541         542   91500.0     4800         3  ...       no     yes      0     no
542         543   94000.0     6000         3  ...       no     yes      0     no
543         544  103000.0     6000         3  ...       no     yes      1     no
544         545  105000.0     6000         3  ...       no     yes      1     no
545         546  105000.0     6000         3  ...       no     yes      1     no

[546 rows x 13 columns]

************
```

```
exp=df2.iloc[:,0:1].values
print(exp)
```

```
[[  1]
 [  2]
 [  3]
 [  4]
 [  5]
 [  6]
 [  7]
 [  8]
 [  9]
 [ 10]
 [ 11]
 [ 12]
 [ 13]
 [ 14]
 [ 15]
 [ 16]
 [ 17]
 [ 18]
 [ 19]
 [ 20]
 [ 21]
 [ 22]
 [ 23]
 [ 24]
 [ 25]
 [ 26]
 [ 27]
 [ 28]
 [ 29]
 [ 30]
 [ 31]
 [ 32]
 [ 33]
 [ 34]
 [ 35]
 [ 36]
 [ 37]
 [ 38]
 [ 39]
 [ 40]
 [ 41]
 [ 42]
 [ 43]
 [ 44]
 [ 45]
 [ 46]
 [ 47]
 [ 48]
 [ 49]
 [ 50]
 [ 51]
 [ 52]
 [ 53]
 [ 54]
 [ 55]
 [ 56]
 [ 57]
 [ 58]
 [ 59]
```

```
sal=df2.iloc[:,1].values
print(sal)
```

```
[ 42000.  38500.  49500.  60500.  61000.  66000.  66000.  69000.  83800.
  88500.  90000.  30500.  27000.  36000.  37000.  37900.  40500.  40750.
  45000.  45000.  48500.  65900.  37900.  38000.  42000.  42300.  43500.
  44000.  44500.  44900.  45000.  48000.  49000.  51500.  61000.  61000.
  61700.  67000.  82000.  54500.  66500.  70000.  82000.  92000.  38000.
  44000.  41000.  43000.  48000.  54800.  55000.  57000.  68000.  95000.
  38000.  25000.  25245.  56000.  35500.  30000.  48000.  48000.  52000.
  54000.  56000.  60000.  60000.  67000.  47000.  70000.  45000.  51000.
  32500.  34000.  35000.  36000.  45000.  47000.  55000.  63900.  50000.
  35000.  50000.  43000.  55500.  57000.  60000.  78000.  35000.  44000.
  47000.  58000. 163000. 128000. 123500.  39000.  53900.  59900.  35000.
  43000.  57000.  79000. 125000. 132000.  58000.  43000.  48000.  58500.
  73000.  63500.  43000.  46500.  92000.  75000.  75000.  85000.  93000.
  94500. 106500. 116000.  61500.  80000.  37000.  59500.  70000.  95000.
 117000. 122500. 123500. 127000.  35000.  44500.  49900.  50500.  65000.
  90000.  46000.  35000.  26500.  43000.  56000.  40000.  51000.  51000.
  57250.  44000.  61000.  62000.  80000.  50000.  59900.  35500.  37000.
  42000.  48000.  60000.  60000.  60000.  62000.  63000.  63900. 130000.
  25000.  50000.  52900.  62000.  73500.  38000.  46000.  48000.  52500.
  32000.  38000.  46000.  50000.  57500.  70000.  69900.  74500.  42000.
  60000.  50000.  58000.  63900.  28000.  54000.  44700.  47000.  50000.
  57250.  67000.  52500.  42000.  57500.  33000.  34400.  40000.  40500.
  46500.  52000.  53000.  53900.  50000.  55500.  56000.  60000.  60000.
  69500.  72000.  92500.  40500.  42000.  47900.  52000.  62000.  41000.
 138300.  42000.  47000.  64500.  46000.  58000.  70100.  78500.  87250.
  70800.  56000.  48000.  68000.  79000.  80000.  87000.  25000.  32500.
  36000.  42500.  43000.  50000.  26000.  30000.  34000.  52000.  70000.
  27000.  32500.  37200.  38000.  42000.  44500.  45000.  48500.  52000.
  53900.  60000.  61000.  64500.  71000.  75500.  33500.  41000.  41000.
  46200.  48500.  48900.  50000.  51000.  52500.  52500.  54000.  59000.
  60000.  63000.  64000.  64900.  65000.  66000.  70000.  65500.  57000.
  52000.  54000.  74500.  90000.  45000.  45000.  65000.  55000.  62000.
  30000.  34000.  38000.  39000.  45000.  47000.  47500.  49000.  50000.
  50000.  52900.  53000.  55000.  56000.  58500.  59500.  60000.  64000.
  67000.  68100.  70000.  72000.  57500.  69900.  70000.  75000.  76900.
  78000.  80000.  82000.  83000.  83000.  83900.  88500.  93000.  98000.
  98500.  99000. 101000. 110000. 115442. 120000. 124000. 175000.  50000.
  55000.  60000.  61000. 106000. 155000. 141000.  62500.  70000.  73000.
  80000.  80000.  88000.  49000.  52000.  59500.  60000.  64000.  64500.
  68500.  78500.  86000.  86900.  75000.  78000.  95000.  97000. 107000.
 130000. 145000. 175000.  72000.  84900.  99000. 114000. 120000. 145000.
  79000.  82000.  85000. 100500. 122000. 126500. 133000. 140000. 190000.
  84000.  97000. 103500. 112500. 140000.  74700.  78000.  78900.  83900.
  85000.  85000.  86000.  86900.  94500.  96000. 106000.  72000.  74500.
  77000.  80750.  82900.  85000.  92500.  76000.  77500.  80000.  80000.
  86000.  87000.  87500.  89000.  89900.  90000.  95000. 112000.  31900.
  52000.  90000. 100000.  91700. 174500.  94700.  68000.  80000.  61100.
  62900.  65500.  66000.  49500.  50000.  53500.  58550.  64500.  65000.
  69000.  73000.  75000.  75000. 132000.  60000.  65000.  69000.  51900.
  57000.  65000.  79500.  72500. 104900. 114900. 120000.  58000.  67000.
  67000.  69000.  73000.  73500.  74900.  75000.  79500. 120900.  44555.
  47000.  47600.  49000.  49000.  49000.  49500.  52000.  54000.  55000.
  55000.  56000.  60000.  60500.  50000.  64900.  93000.  85000.  61500.
  88500.  88000.  89000.  89500.  95000.  95500.  51500.  62900. 118500.
  42900.  44100.  47000.  50000.  50000.  53000.  53000.  54000.  58500.
  59000.  60000.  62900.  64000.  65000.  67900.  68500.  70000.  70500.
  71500.  71900.  75000.  75000.  87000.  64000.  70000.  47500.  62600.
  66000.  58900.  53000.  95000.  96500. 101000. 102000. 103000. 105000.
 108000. 110000. 113000. 120000. 105000. 106000. 107500. 108000. 113750.
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(exp, sal, test_size = 0.2, random_state=0)
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
y_pred_train = regressor.predict(X_train)
```

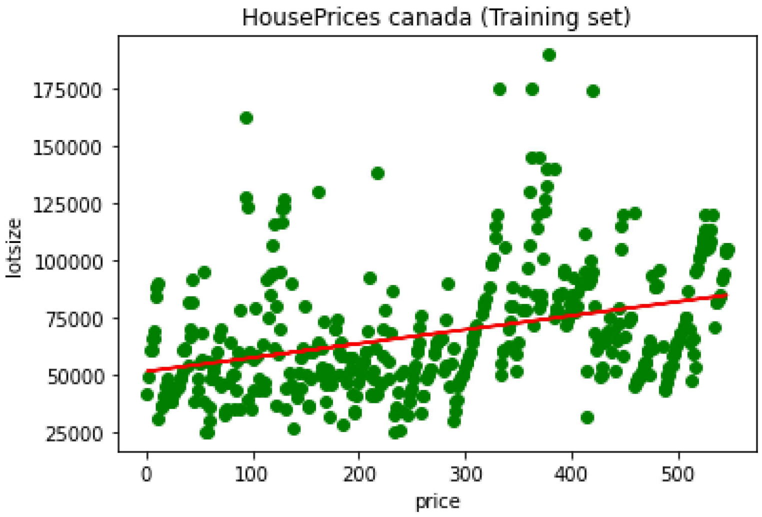
```
print("Model Score: ", regressor.score(X_test, y_test))
```

```
Model Score:  0.18623549897840064
```

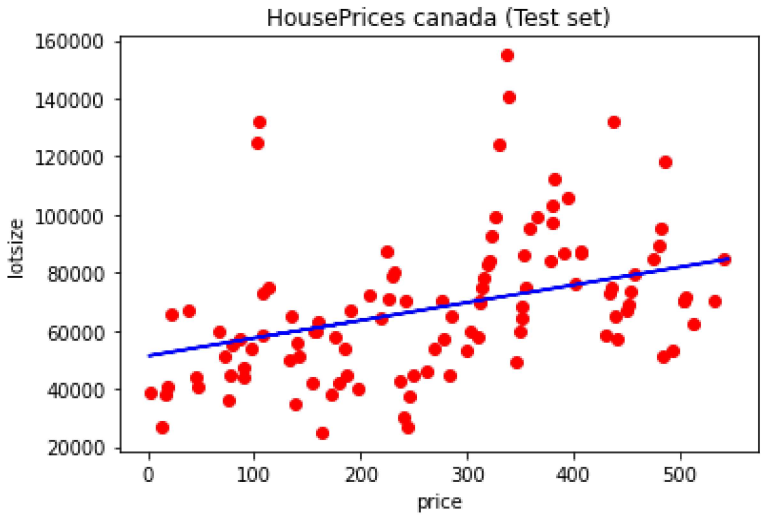
```
from sklearn.metrics import r2_score
print("R_square score: ", r2_score(y_test,y_pred))
```

```
R_square score:  0.18623549897840064
```

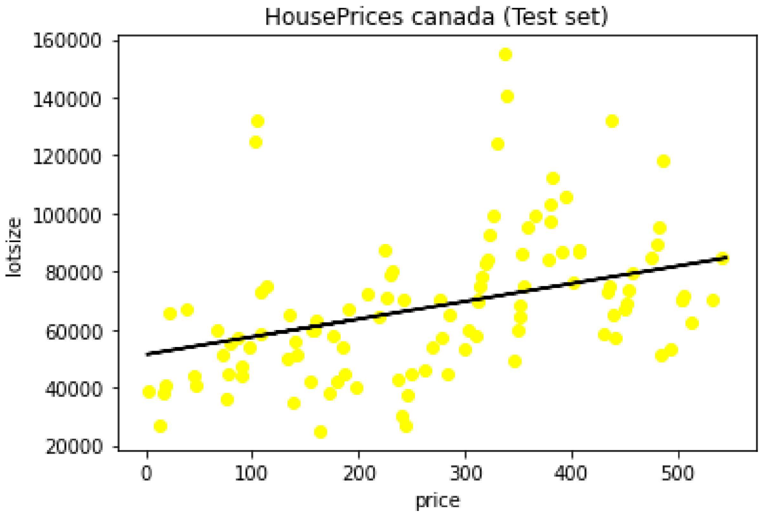
```
import matplotlib.pyplot as plt
plt.scatter(X_train, y_train, color = 'green')
plt.plot(X_train, regressor.predict(X_train), color = 'red')
plt.title('HousePrices canada (Training set)')
plt.xlabel('price')
plt.ylabel('lotsize')
plt.show()
```



```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('HousePrices canada (Test set)')
plt.xlabel('price')
plt.ylabel('lotsize')
plt.show()
```



```
plt.scatter(X_test, y_test, color = 'yellow')
plt.plot(X_train, regressor.predict(X_train), color = 'black')
plt.title('HousePrices canada (Test set)')
plt.xlabel('price')
plt.ylabel('lotsize')
plt.show()
```


HousePrices canada (Test set)
160000
140000
120000
100000
80000
60000
40000
20000
lotsize
0
100
200
300
400
500
price