```
# Load the dataset

from sklearn import tree
from pandas import read_csv
import os
import numpy as np
df = read_csv("sales_data_sample.csv")
```

```
df.head(8)
```

	QTR_ID	MONTH_ID	YEAR_ID	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES
0	1	2	2003	10107	30	95.70	2	2871.00
1	2	5	2003	10121	34	81.35	5	2765.90
2	3	7	2003	10134	41	94.74	2	3884.34
3	3	8	2003	10145	45	83.26	6	3746.70
4	4	10	2003	10159	49	100.00	14	5205.27
5	4	10	2003	10168	36	96.66	1	3479.76
6	4	11	2003	10180	29	86.13	9	2497.77
7	4	11	2003	10188	48	100.00	1	5512.32

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   QTR_ID           2823 non-null   int64  
 1   MONTH_ID         2823 non-null   int64  
 2   YEAR_ID          2823 non-null   int64  
 3   ORDERNUMBER      2823 non-null   int64  
 4   QUANTITYORDERED  2823 non-null   int64  
 5   PRICEEACH        2823 non-null   float64
 6   ORDERLINENUMBER  2823 non-null   int64  
 7   SALES            2823 non-null   float64
dtypes: float64(2), int64(6)
memory usage: 176.6 KB
```

```
# Separate the target variable (Outcome) from the features
x = np.array(df.drop(["SALES"], 1))
y = np.array(df["SALES"])
```

```
<ipython-input-24-b7d750e52d3b>:2: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument
  x = np.array(df.drop(["SALES"], 1))
```

```
df.describe()
```

	QTR_ID	MONTH_ID	YEAR_ID	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES
count	2823.000000	2823.000000	2823.00000	2823.000000	2823.000000	2823.000000	2823.000000	2823.000000
mean	2.717676	7.092455	2003.81509	10258.725115	35.092809	83.658544	6.466171	3553.889072
std	1.203878	3.656633	0.69967	92.085478	9.741443	20.174277	4.225841	1841.865106
min	1.000000	1.000000	2003.00000	10100.000000	6.000000	26.880000	1.000000	482.130000
25%	2.000000	4.000000	2003.00000	10180.000000	27.000000	68.860000	3.000000	2203.430000
50%	3.000000	8.000000	2004.00000	10262.000000	35.000000	95.700000	6.000000	3184.800000
75%	4.000000	11.000000	2004.00000	10333.500000	43.000000	100.000000	9.000000	4508.000000
max	4.000000	12.000000	2005.00000	10425.000000	97.000000	100.000000	18.000000	14082.800000

```
df.shape
```

```
(2823, 8)
```

```
(2823, 8)
```

```
array([2871.  , 2765.9 , 3884.34, ..., 5417.57, 2116.16, 3079.44])
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 100)
```

```
from sklearn.linear_model import LinearRegression
# Create a linear regression model
model = LinearRegression()
```

```
# Fit the model to the training data
model.fit(x_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
# Make predictions on the test data
y_pred = model.predict(x_test)
```

```
y_pred
```

```
            5427.78104615,  -181.23494125,  3925.13144598,  2872.94511489,
            3448.43462208,  5172.89336449,  5057.91488843,   318.79054048,
             419.57824489,  1005.09028005,  1824.99676902,  3807.53732403,
            5080.22934793,  4193.21724204,  3297.84209357,  2932.62041759,
            5318.62217784,  3377.87393923,  4711.74124575])
```

```
# to test the sanmple data
import numpy as np
reshaped_test=np.array([3,7,2003,10134,41,94.74,2])
reshaped_test = test.reshape(1, -1)
reshaped_test
y_pred = model.predict(reshaped_test)
```

```
y_pred
```

```
array([4878.09917784])
```

```
(-22.52453303*3)+(7.02788889 *7)+(-36.16766958*2003)+(0.38153327*10134)+(105.13172685*41)+(60.38264346*94.74)+(-15.22005571*2)
```

```
-58595.15005658959
```

```
# Get the intercept and coefficients
intercept = model.intercept_
coefficients = model.coef_
print("intercept",intercept)
print("coefficients",coefficients)
```

```
intercept 63473.249190095165
coefficients [-22.52453303   7.02788889 -36.16766958   0.38153327 105.13172685
  60.38264346 -15.22005571]
```

```
intercept
```

```
63473.249190095165
```

```
# Y= a+bX
# a - intercept, b - coefficient

y=-58595.15005658959+63473.249190100796
print(y)
```

```
4878.099133511205
```