Machine learning

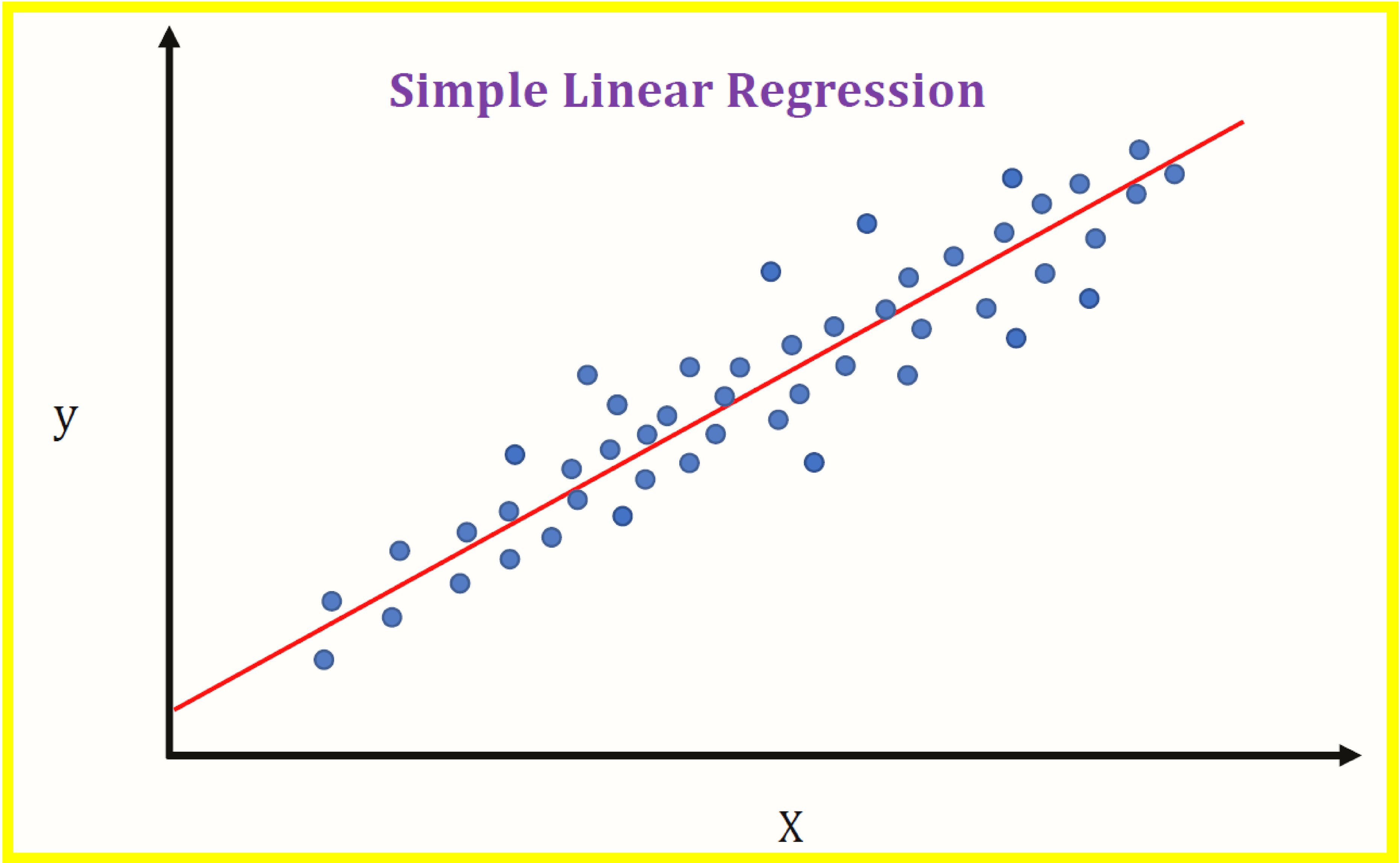
Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behavior.

Supervised Learning

Supervised learning is defined by use of labeled datasets to train algorithms that to classify data or predict outcomes accurately.

Linear Regression

Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values.



Formula For Linear Regression With single class

$$Y = a + bX$$

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

Here we are using the linear regression over a single class

We have a dataset of home price where we are going to predict the price of the plot using areas.

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

In [3]:
```
df
```

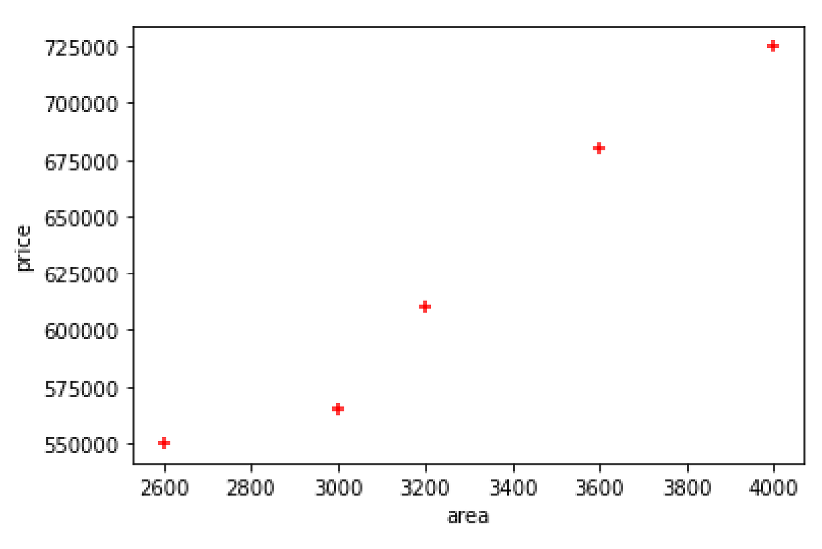
Out[3]:

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

Ploting the dataset

In [4]:
```
plt.xlabel('area')
plt.ylabel('price')
plt.scatter(df.area, df.price, color='red', marker='+')
```

Out[4]:
```
<matplotlib.collections.PathCollection at 0x1c37b6a52b0>
```



Seperating x and y labels. Our x labels would by an area while y label would be a price. We want to predict the price.

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 [6]:
```
price = df.price
price
```

Out[6]:
```
0    550000
1    565000
2    610000
3    680000
4    725000
Name: price, dtype: int64
```

Applying linear regressing using a fit function on x and y labels

In [7]:
```
reg = linear_model.LinearRegression()
reg.fit(new_df, price)
```

Out[7]:
```
LinearRegression()
```

Predict function is used to predict the values of price. We give area as input

In [8]:
```
reg.predict([[3300]])
```

Out[8]:
```
array([628715.75342466])
```

In [9]:
```
reg.predict([[5000]])
```

Out[9]:
```
array([859554.79452055])
```

Linear regression for multiple variable

Multiple linear regression refers to a statistical technique that uses two or more independent variables to predict the outcome of a dependent variable.

Formula for linear regression on multiple variable

In [11]:
```
df1 = pd.read_csv('homepricess.csv')
```

Our dataset contain multiple independent variables such as area, bedroom, and age. All of these variables effects on hourse price. So through all of these we would be predicting house price

In [12]:
```
df1
```

Out[12]:

	area	bedrooms	age	price
0	2600	3.0	20	550000
1	3000	4.0	15	565000
2	3200	NaN	18	610000
3	3600	3.0	30	595000
4	4000	5.0	8	760000
5	4100	6.0	8	810000

In [13]:
```
import math
```

We have a one empty sample. To fill out it, we would be taking a median of that column and floor function is used for round off. Like we have a bedrooms and bedroom could never be in decimals.

In [14]:
```
empt = math.floor(df1.bedrooms.median())
```

In [15]:
```
empt
```

Out[15]:
```
4
```

After taking a round off and median, we fill that empty sample by using a fillna function.

In [16]:
```
df1.bedrooms = df1.bedrooms.fillna(empt)
```

In [17]:
```
df1
```

Out[17]:

	area	bedrooms	age	price
0	2600	3.0	20	550000
1	3000	4.0	15	565000
2	3200	4.0	18	610000
3	3600	3.0	30	595000
4	4000	5.0	8	760000
5	4100	6.0	8	810000

Applying linear regression.

In [18]:
```
reg = linear_model.LinearRegression()
```

In [23]:
```
reg.fit(df1[['area', 'bedrooms', 'age']], df1.price)
```

Out[23]:
```
LinearRegression()
```

In [25]:
```
reg.predict([[3000, 3, 20]])
```

Out[25]:
```
array([563042.60975297])
```