# Linear Regression

Created By

## What is regression?

Regression analysis is the form of modeling technique that defines a relationship between a dependent variable and the independent variable.

In terms of machine learning, the regression analysis draws a line on the data points in such a way that it could fit maximum number of points. The regression analysis is used for following:

1. **Determining the strength of predictors:** The regression analysis is used to find the strength of the relationship between the independent variable and the dependent variable. For e.g. what is the relationship between employee salary and revenue generated by that employee, or how is the relationship between sales and marketing of a product.
2. **Forecasting an effect**: It estimates how much the dependent variable changes with the change in the independent variable, for e.g., how much additional income (dependent variable) can be generated if an extra 1000 dollars is spent on marketing(independent variable).
3. **Trend forecasting:** Regression analysis could be used to find the trend in the data and might be used to forecast the next values. For e.g. it could find the trend in the daily number of steps walked by a person in the last 6 months and can predict the next number of steps a person might take the next day.

## What is linear regression v/s logistic regression?

The two most common types of regression are linear and logistic.

| | Linear regression | Logistic regression |
|---|---|---|
| Concept | Models data by straight line | models data by sigmoid function |
| x-y relationship | y = mx + c | Sigmoid function- gives probability |
| Data type | Used with continuous data i.e for continuous x we have continuous values of y | Used for categorical data i.e. for continuous values of x we have categories in y (dependent variable) |
| Output | Continuous value of the variable | Probability of occurrence of an event |
| Metrics | Loss, R square, adjusted R square | Classification report : confusion matrix, ROC |

## When to choose linear regression?

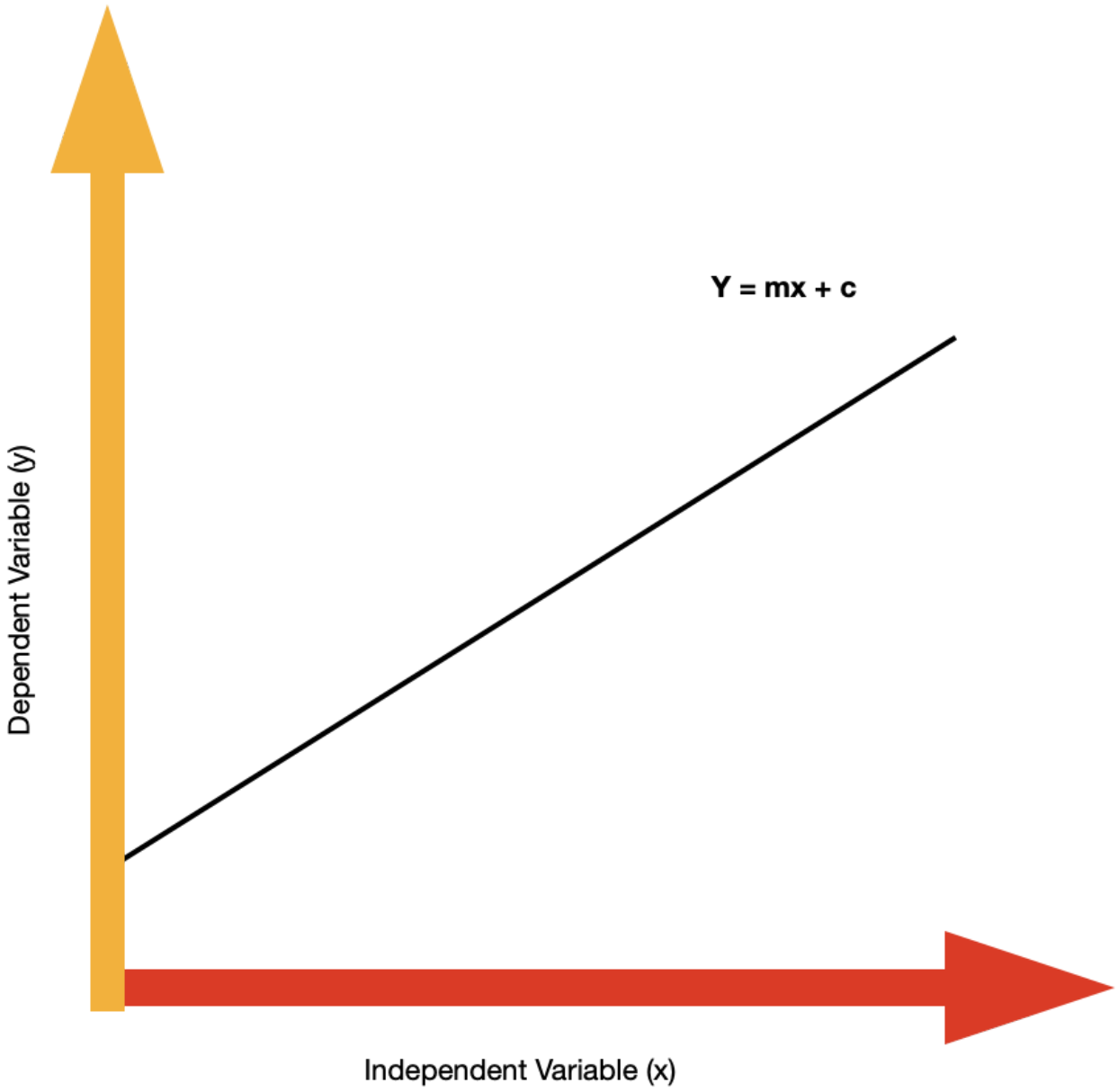
The selection of this model should be based on following key points:

1. **Classification and regression capabilities:** Linear regression is best for regression problems as it will give continuous variables as the output and is suitable for prediction and forecasting. For e.g. it could be used to forecast the next day's temperature, the stock price of a company, projected sales in the next week, and many more. But for the classification problems, whenever a new data point is added, the model has to be redefined so that it can fit max number of data points, hence it is not suitable for classification problems.

2. **Data Quality**: Outliers play an important role in regression analysis, the presence of outliers might change the position of the linear regression line and could actually not fit the data points nicely. Hence, if we have data free of outliers only then the results of linear regression could be trusted. Similarly, if the data has too many missing values then linear regression might not be the best choice to use.
3. **Computational complexity:** This algorithm is not very expensive as compared to decision trees or random forest. For n  number of instances and x number of features, the big O is equal to the product of x and n. i.e. O(xn)
4. **Explainable:** Because its mathematical model is simple and easy to understand, it is more explainable than other complex algorithms.

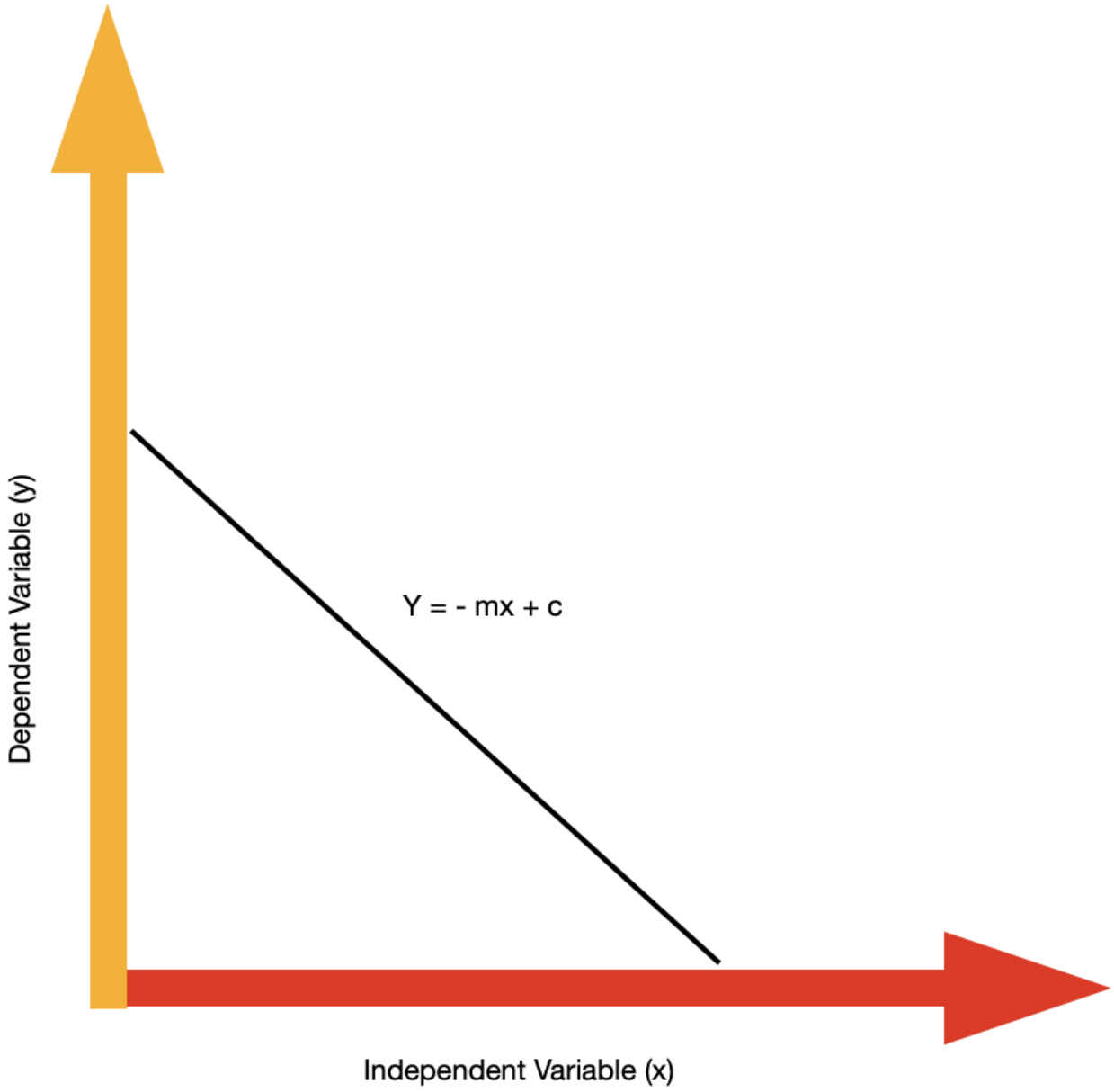
# Mathematical basis of linear regression

Let’s understand linear regression slope first. The equation of a line is y = mx + c , where m is the slope and c is the y-intercept. The slope m could be positive or negative depending on the relationship between independent variable ( x ) and dependent variable ( y ). The figure below shows the regression line with a positive slope.



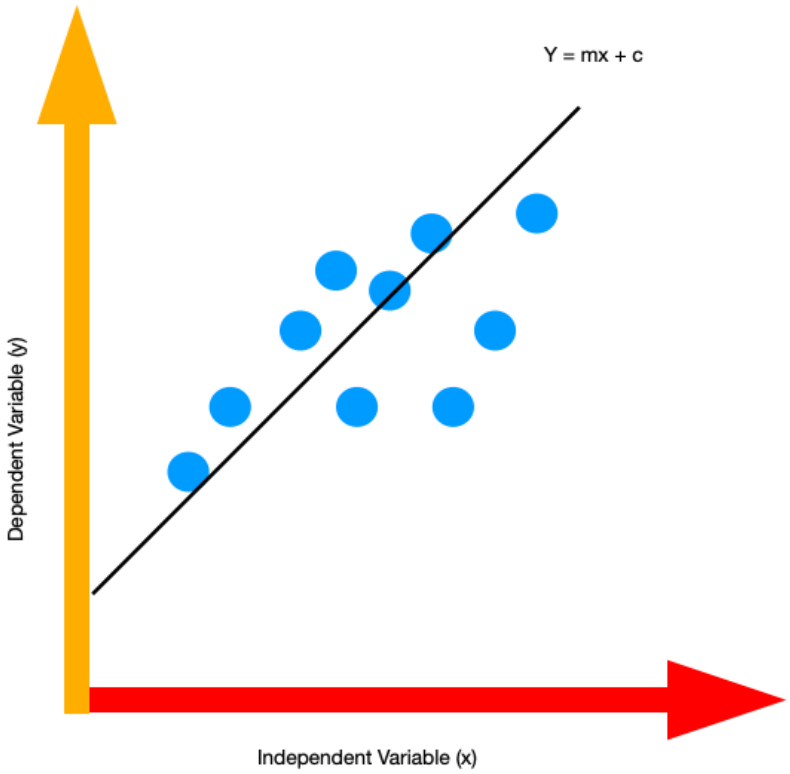

The positive slope shows the directly proportional relationship between x and y i.e y will increase with the increase in x and vice Versa. For e.g. speed v/s distance graph must have a positive slope line keeping time constant because as we increase the speed, the distance covered in a particular time will increase.

The figure below shows the regression line with a negative slope:




In this figure, the y is inversely proportional to x which means, that when x is increased y decreases and vice versa. For. e.g speed v/s time graph will have a negative slope line for a fixed distance. This is because, as we increase the speed the time taken to cover a certain distance decreases.

To understand the linear regression in-depth see the graph below. This graph shows the data points and line of regression, having some positive slope m with y-intercept c. Now, based on this line we will have a predicted value, which could be quite a way from the actual value and hence give us some error. Our goal is to integrate the algorithms a few times in order to minimize the error, hence with every iteration, we will have a new regression line, and a new set of error values. The error minimization is done by using the gradient descent method.




Let's see the mathematics behind linear regression. The following steps explain the algorithm in detail:

1. Given is set of x values and y values, we need to find the regression line based on these data points.

| X | Y |
|---|---|
| 1 | 3 |
| 2 | 4 |
| 3 | 2 |
| 4 | 4 |
| 5 | 5 |

2. To find the regression line we need to find the slope and c. The slope m is calculated by following the statistical formula:

$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$m = \frac{(\sum x - x_{mean})(\sum y - y_{mean})}{(\sum x - x_{mean})^2}$$

$$m = \frac{\sum(x - x_{mean}}{y - y_{mean}} \sum(x - x_{mean})^2$$

$$m = \frac{\sum(x - x_{mean}}{y - y_{mean}} \sum(x - x_{mean})^2$$

$$m = \frac{\sum(x - x_{mean}}{y - y_{mean}} \sum(x - x_{mean})^2$$

$$m = \frac{\sum(x - x_{mean}}{y - y_{mean}} \sum(x - x_{mean})^2$$

3. Next step is to find x-xmean, y - ymean, and (x - xmean)^2.

| X | Y | X-Xmean = A | y-ymean = B | (x-xmean)^2 | AB |
|---|---|---|---|---|---|
| 1 | 3 | -2 | -0.6 | 4 | 1.2 |
| 2 | 4 | -1 | 0.4 | 1 | -0.4 |
| 3 | 2 | 0 | -1.6 | 0 | 0 |
| 4 | 4 | 1 | 0.4 | 1 | 0.4 |
| 5 | 5 | 2 | 1.4 | 4 | 2.8 |
| Sum | | | | 10 | 4 |

Now the slope is 4 / 10 or 0.4 . Now we know the slope of the linear regression line, the next step is to find the value for the y-intercept.

4. Calculate c: we will use the line equation, use y = 3.6 (mean value) and x = 3 (its mean value) to find C.

y = mx + c

3.6 = 3 (0.4) + c

c = 3.6 - 1.2 = 2.4

So the y-intercept is 2.4.

5. Predict the values for y by putting values of x, m = 0.4 and c = 2.4

y = 0.4 (1) + 2.4 = 2.8

y = 0.4 (2) + 2.4 = 3.2

y = 0.4 (3) + 2.4 = 3.6

y = 0.4 (4) + 2.4 = 4.0

y = 0.4 (5) + 2.4 = 4.4

6. Now, we know the predicted values and the actual values, we can find the error relative to each predictive data point.

| Y_pred | Y_actual | Error = y_actual - Y_pred |
|---|---|---|
| 2.8 | 3 | 0.2 |
| 3.2 | 4 | 0.8 |
| | | |

| | | |
|---|---|---|
| 3.6 | 2 | -1.6 |
| 4.0 | 4 | 0 |
| 4.4 | 5 | 0.6 |

7. Now our goal is to minimize this error by choosing different values of slope m. The best fit line will be the line having minimum error between the actual values and predicted values. With every iteration, a new value of m is chosen and error is calculated as we have shown in the above steps.

# How to check the goodness of fit?

To see how our model is performing or how well the linear regression line is fitting the data points, we have a method called the R square method.

R-Square is a statistical parameter that shows how close the line is to the data points. It is also called a coefficient of determination, or the coefficient of multiple determination.

$$R^2 = \frac{SSR}{SST} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}$$

R-square is the ration of sum of squared values of (y_pred - y_mean) to the sum of squared values of (y_actual - y_mean). Let's see on the table , y_mean = 3.6

| x | Y_pred | y_actual | y_pred - y_mean = D | D^2 | y_actual - y_mean = E | E^2 |
|---|---|---|---|---|---|---|
| 1 | 2.8 | 3 | -0.8 | 0.64 | -0.6 | 0.36 |
| 2 | 3.2 | 4 | -0.4 | 0.16 | 0.4 | 0.16 |
| 3 | 3.6 | 2 | 0 | 0 | -1.6 | 2.56 |
| 4 | 4.0 | 4 | 0.4 | 0.16 | 0.4 | 0.16 |
| 5 | 4.4 | 5 | 0.8 | 0.64 | 1.4 | 1.96 |
| Sum | | | | 1.6 | | 5.2 |

As per the formula, our R-square = 1.6 / 5.2 = 0.3

As the value of R-Square is quite low, it suggests that the regression line is not a good fit and data points are quite far from the line.

The ideal value of R-Square = 1 which means all the predicted data points are lying on th Eline itself and the error is zero. And vice versa, a very low value of R-square means a very poor fit of line on the data.

# Python implementation using Numpy, pandas

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('all_week.csv')
data.head().   # see first 5 rows of dataframe
data = data.dropna() # frop the rows having missing values
training_data = data[:600] #choose subset of the data
test_data = data[601:]
```

```
X = training_data['depth'].values
Y = training_data['mag'].values
```

```
# Find slope and y-intercept

X_mean = np.mean(X). # mean value of x
Y_mean = np.mean(Y) # mean value of y
# total number of values
n = len(X). # number of rows in X

# Find the slope m
numer = 0
denom = 0
for i in range(n):
    numer =  numer + ( X[i]-X_mean ) * (Y[i]-Y_mean) # by using above formula
    denom = denom + (X[i]-X_mean) ** 2
m = numer / denom

# find y-intercept C = y - mx, y = y_mean, x = X-mean
C = Y_mean - (m * X_mean)
print(f"The slope of the line is {m} and the intercept is {C}")
```

```
#%% Now we know the slope, we can find Y_pred now
Y_pred = []

for i in range(n):
    Y_pred.append(m * X[i] + C)

# Error between Y_pred and Y
error = []
for i in range(n):
    error.append(Y_pred[i]-Y[i])

# Lets find R-Square
D_sum = 0
for i in range(n):
    D_sum = D_sum +  Y_pred[i] - Y_mean
D_sum_squ = D_sum ** 2
E_sum = 0
for i in range(n):
    E_sum = E_sum +  Y[i] - Y_mean
E_sum_squ = E_sum ** 2

R = (D_sum/E_sum) ** 2
print(f"The R-square value is {R}")
```