FORMERLY Industries Helping Hands

THE iSCALE

Data Science | 30 Days of Machine Learning | Day - 23

Educator Name: Nishant Dhote
Support Team: **+91-7880-113-112**

----Today Topics | Day 23----

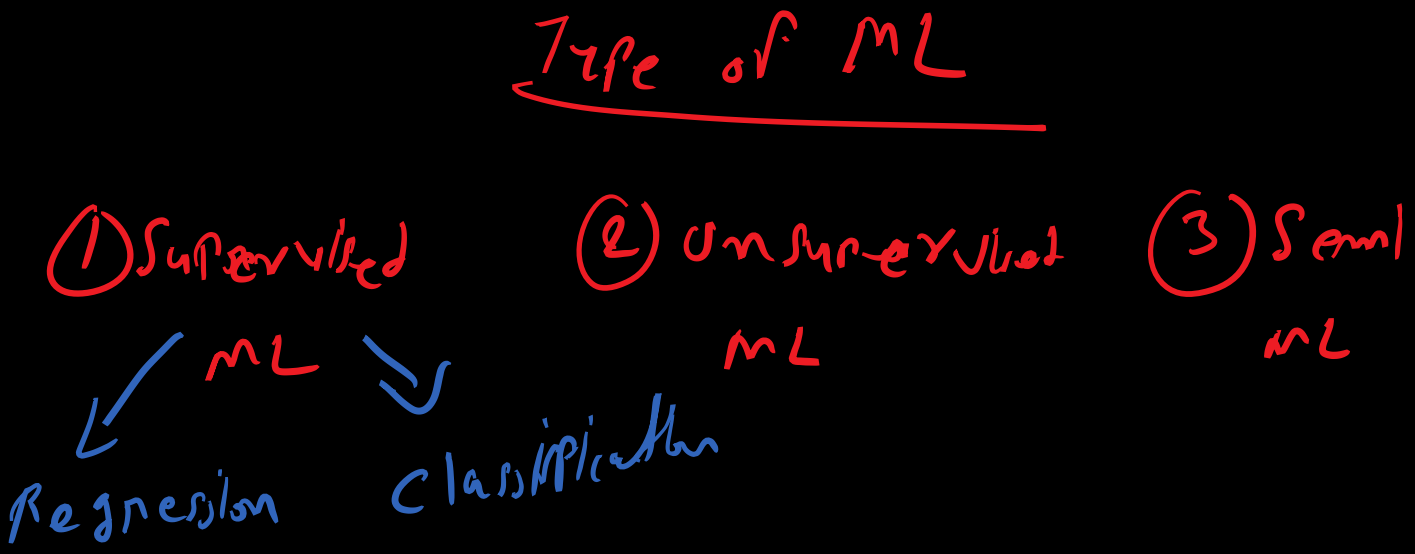
Linear Regression

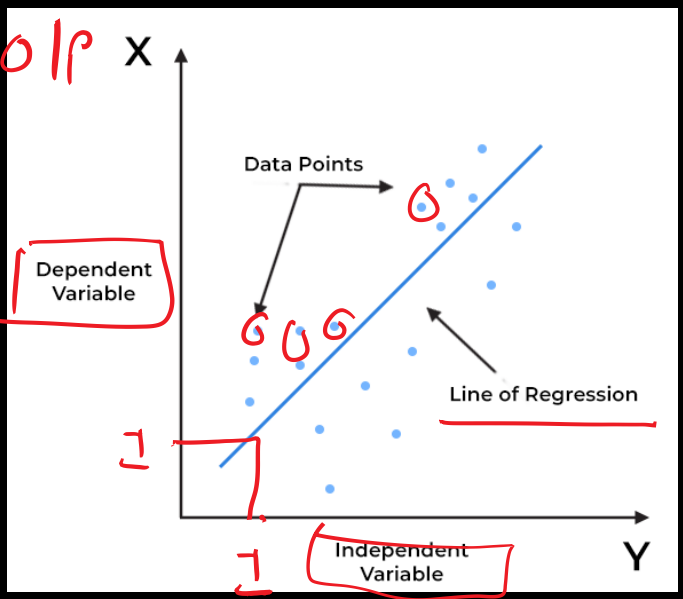
- What Is Linear Regression?
- Key Benefits of Linear Regression
- Type of Linear Regression
- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Linear Regression

- Dataset Link GitHub: https://github.com/TheiScale/30_Days_Machine_Learning/

- What Is Linear Regression?

Linear regression is an algorithm that provides a linear relationship between an independent variable and a dependent variable to predict the outcome of future events. It is a statistical method used in data science and machine learning for predictive analysis.





In the above figure,

X-axis = Dependent variable

Y-axis = Output / Independent variable

Line of regression = Best fit line for a model

========

Key Benefits of Linear Regression

Linear regression is a popular statistical tool used in data science, thanks to the several benefits it offers, such as:

1. Easy implementation

The linear regression model is computationally simple to implement as it does not demand a lot of engineering overheads, neither before the model launch nor during its maintenance.

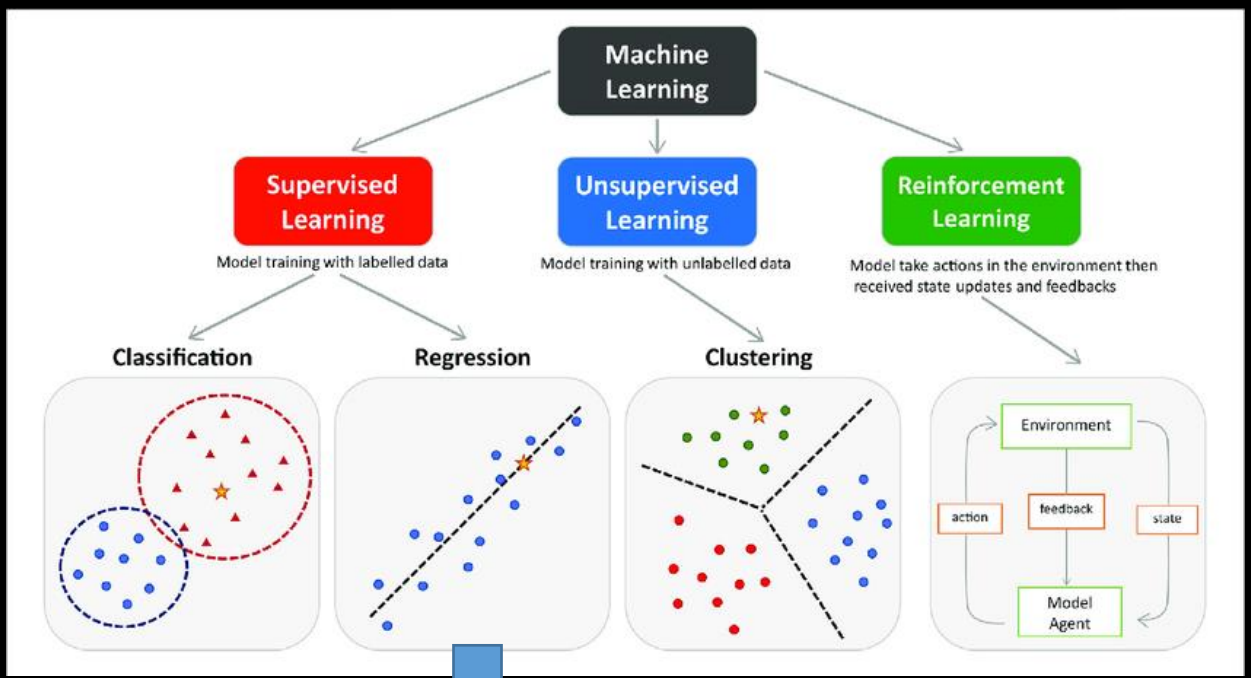
2. Interpretability

Unlike other deep learning models (neural networks), linear regression is relatively straightforward. As a result, this algorithm stands ahead of black-box models that fall short in justifying which input variable causes the output variable to change.

3. Scalability

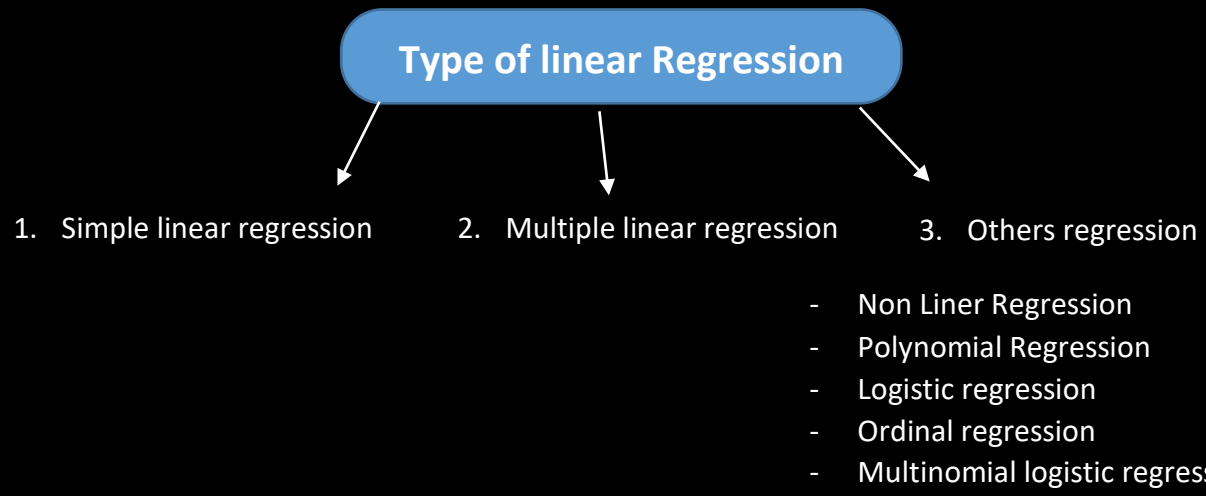
Linear regression is not computationally heavy and, therefore, fits well in cases where scaling is essential. For example, the model can scale well regarding increased data volume (big data).

Type of Machine Learning:



Linear Regression

Type of Linear Regression

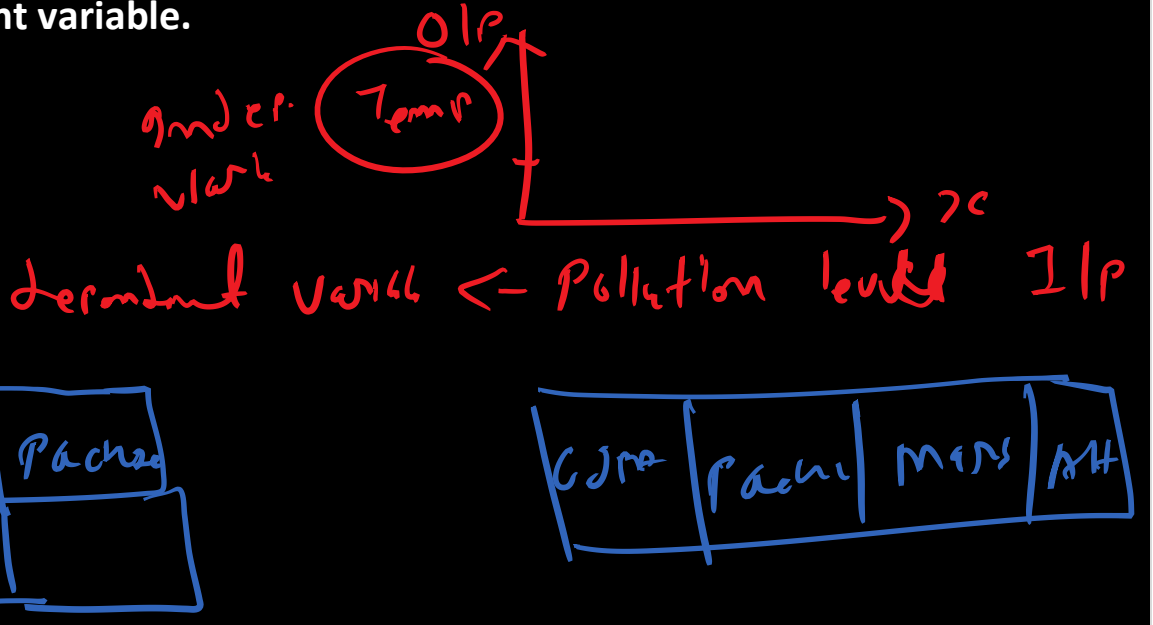


1. Simple linear regression

Simple linear regression reveals the correlation between a dependent variable (input) and an independent variable (output). Primarily, this regression type describes the following:

Relationship strength between the given variables.

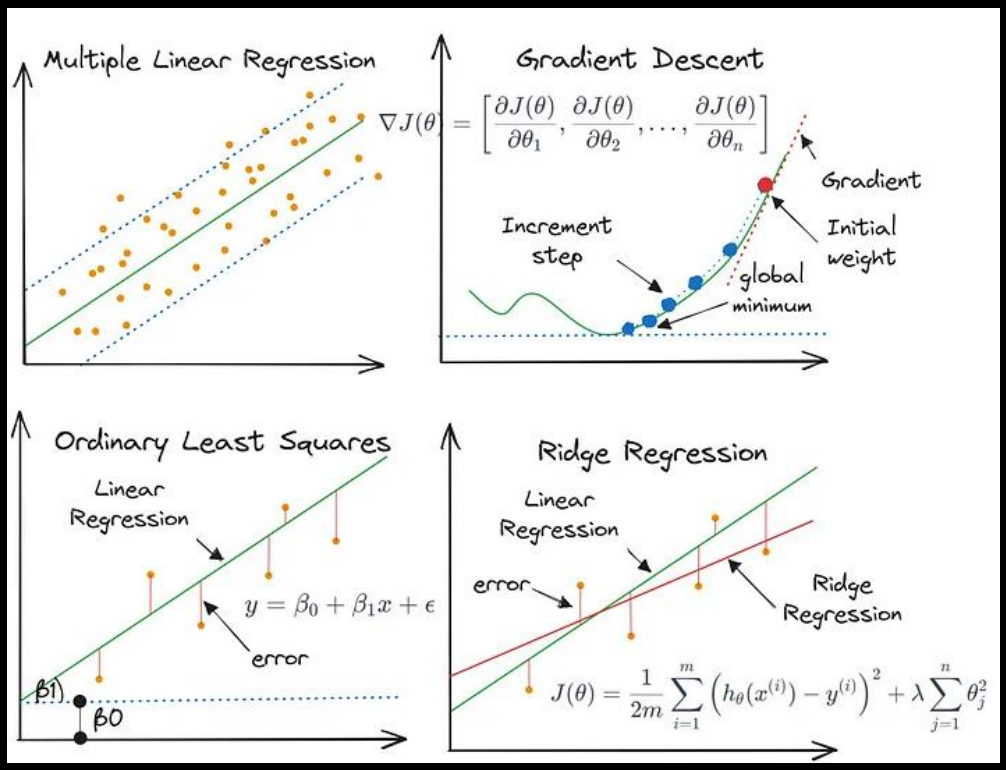
Example: The relationship between pollution levels and rising temperatures.
The value of the dependent variable is based on the value of the independent variable.



2. Multiple linear regression

Multiple linear regression establishes the relationship between independent variables (two or more) and the corresponding dependent variable. Here, the independent variables can be either continuous or categorical. This regression type helps foresee trends, determine future values, and predict the impacts of changes.

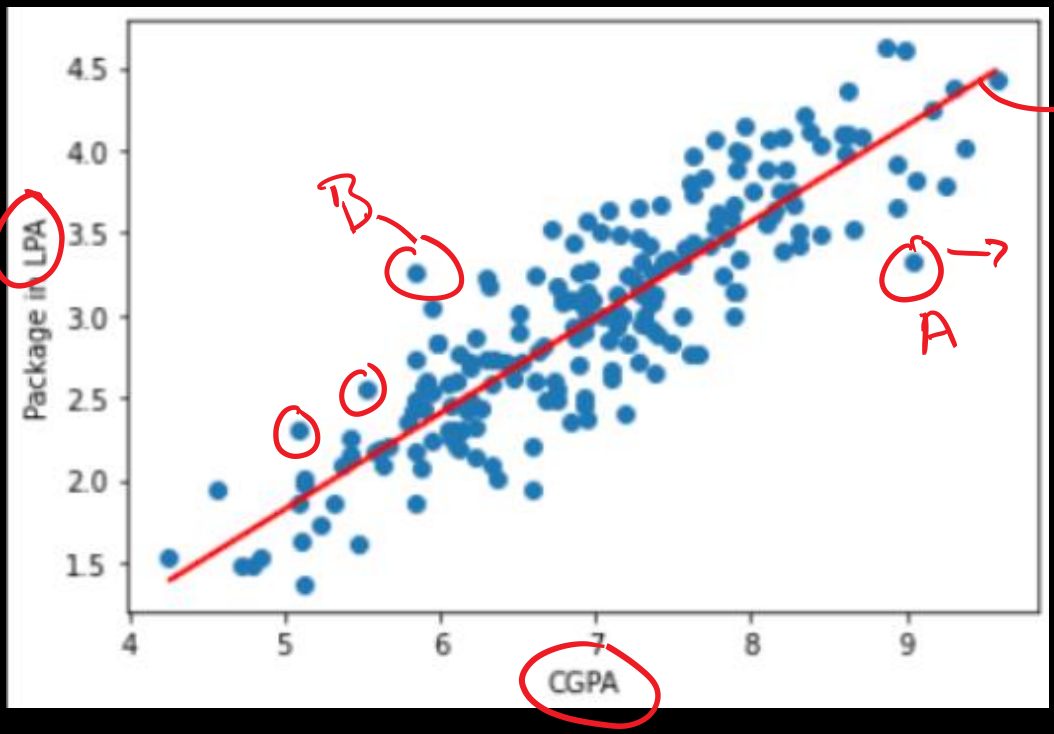
Example: Consider the task of calculating blood pressure. In this case, height, weight, and amount of exercise can be considered independent variables. Here, we can use multiple linear regression to analyse the relationship between the three independent variables and one dependent variable, as all the variables considered are quantitative.


Multiple Linear Regression
Gradient Descent
$\nabla J(\theta) = \left[\frac{\partial J(\theta)}{\partial \theta_1}, \frac{\partial J(\theta)}{\partial \theta_2}, \ldots, \frac{\partial J(\theta)}{\partial \theta_n}\right]$
Gradient
Increment step
Initial weight
global minimum
Ordinary Least Squares
Linear Regression
$y = \beta_0 + \beta_1 x + \epsilon$
error
β1
β0
Ridge Regression
Linear Regression
error
Ridge Regression
$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$

Today's Example **Simple linear regression**

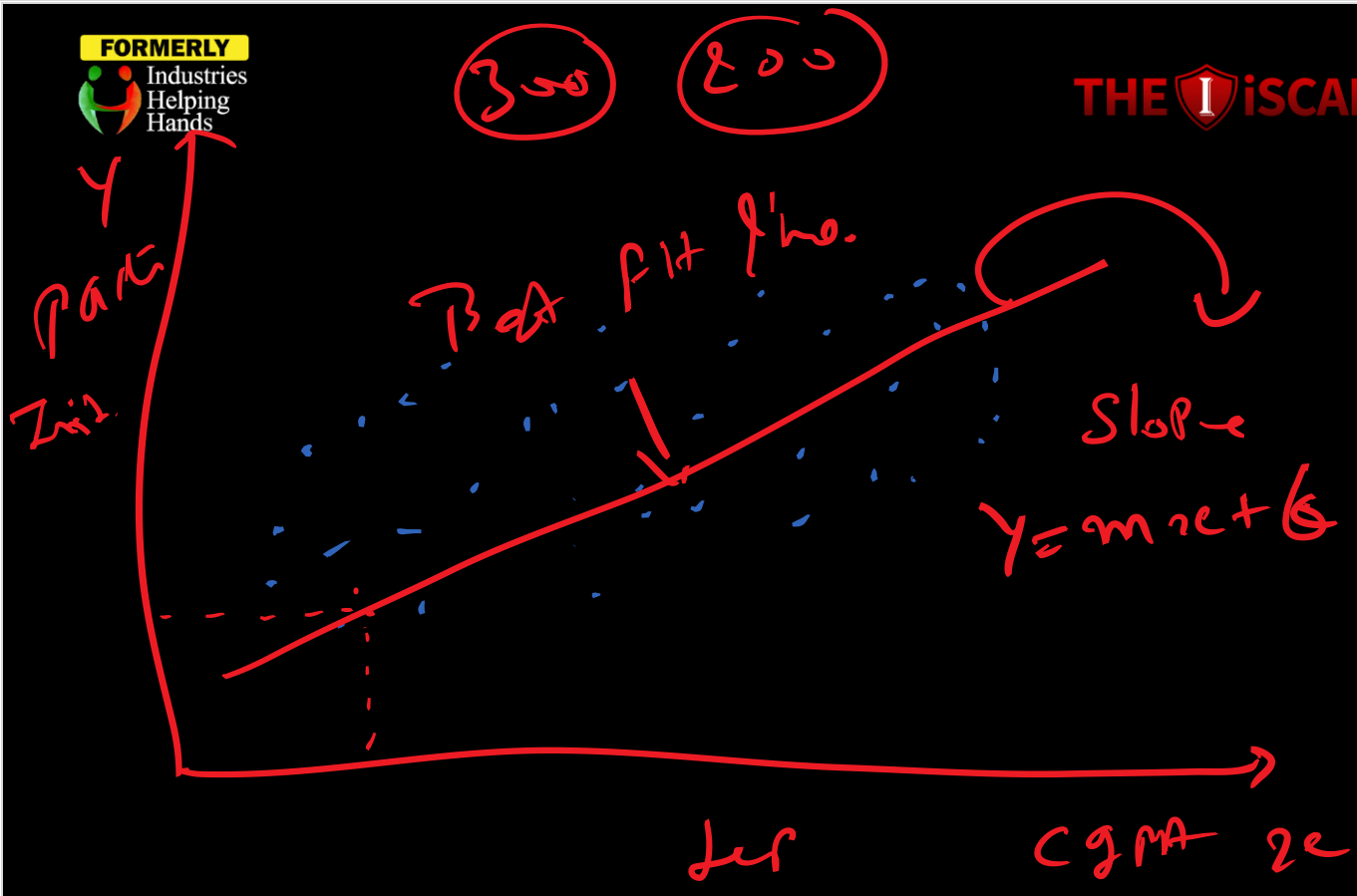
Package LPA and CGPA

CgPA | Package



Simple linear regression

1 Input Column	1 Output Column
CGPA	Package
6.0	3.8
7.25	4.2
8.0	3.0



OP, $\hat{Y}$ = Package.

x = CGPA

I/P

$Y = mx + b$

$\downarrow$

Package = (m) x CGPA + (b)

Best fit line

$\downarrow$

Intercept

<Start Coding>

```
#Import Library

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

#Import Dataset

df = pd.read_csv('placement.csv')
----
df.head()


#Data Plot in Graph

plt.scatter(df['cgpa'],df['package'])
plt.xlabel('CGPA')
plt.ylabel('Package(in lpa)')


#Define X and y as a Input and Output Column

X = df.iloc[:,0:1]
y = df.iloc[:,-1]

----
X
----
y
```

```
#Train test split

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=2)

#Import linear regression

from sklearn.linear_model import LinearRegression


#Define LR

lr = LinearRegression()


#Use "fit" for train the model

lr.fit(X_train,y_train)

---

X_test

---

y_test


#Predict the value

lr.predict(X_test.iloc[0].values.reshape(1,1))

---
lr.predict(X_test.iloc[1].values.reshape(1,1))
```

```
#Again plot the "X_train" data

plt.scatter(df['cgpa'],df['package'])
plt.plot(X_train,lr.predict(X_train),color='red')
plt.xlabel('CGPA')
plt.ylabel('Package(in lpa)')


#Find the value of "m" and "b"


m = lr.coef_

----

b = lr.intercept_


#Calculate the slope line


# y = mx + b

m * 8.58 + b

----

m * 9.25 + b
```

FORMERLY Industries Helping Hands

THE iSCALE

Day 23 | Data Curious Minds

Suggest topic – Next class