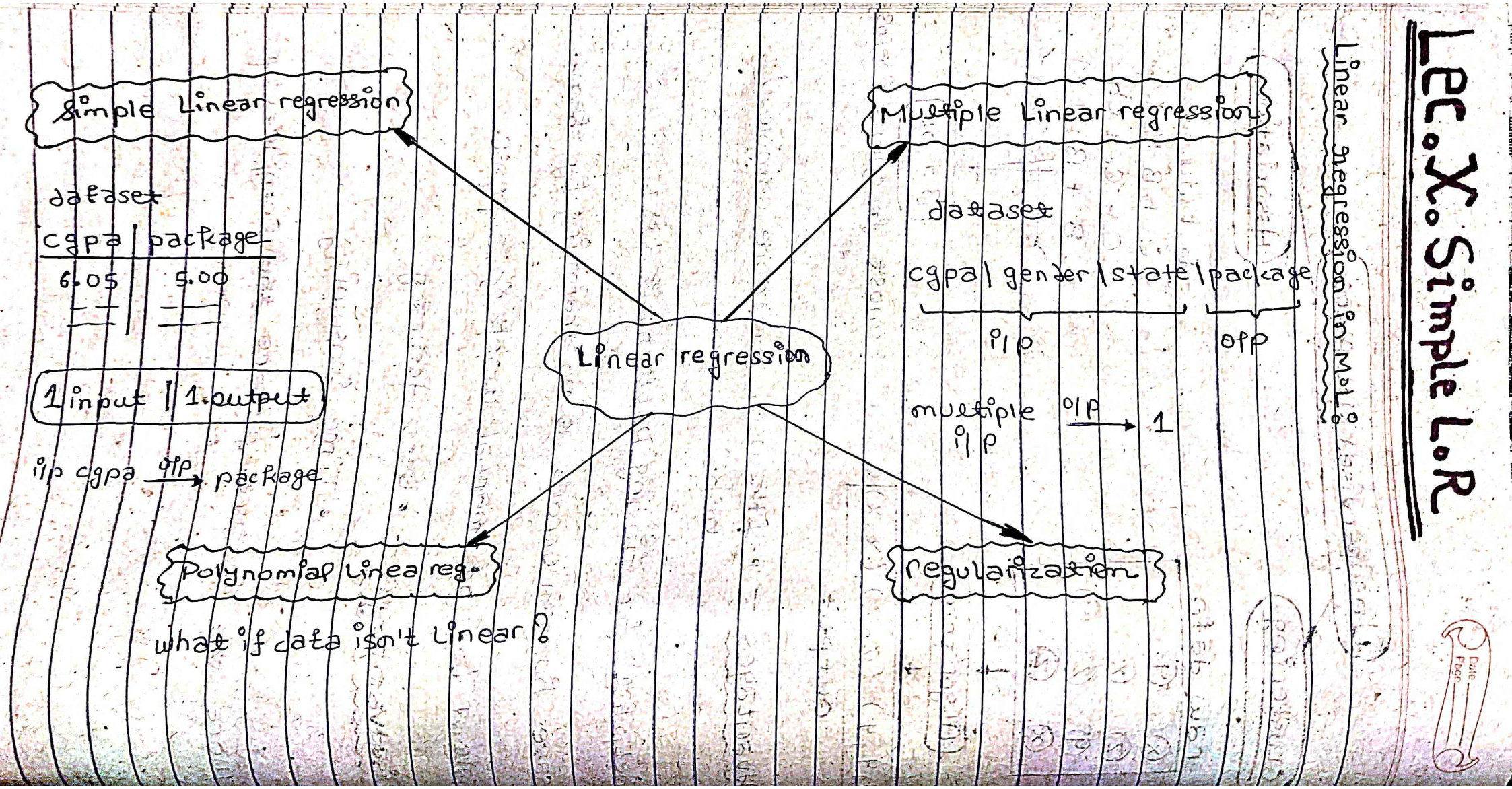
Lec. X. Simple L.R

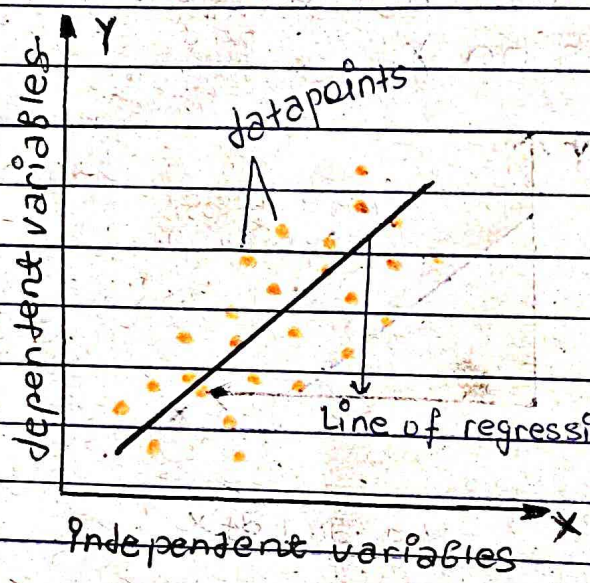
Linear regression in ML



● Linear Regression is a supervised M.L Algorithm

L.R algo. shows a linear relationship b/w a dependent (y) and one or more independent (y) variables, hence called a linear regression.

Since, L.R shows linear relationship, which means it finds how the value of the dependent variable is changing according to independent variable.



$y = a_0 + a_1 x + \epsilon$

y : Dependent (Target) var.
x : Independent var. (Predictor variable)
a_0 : intercept of Line
a_1 : L.R coefficient (scale factor to each input value).
ϵ : random error

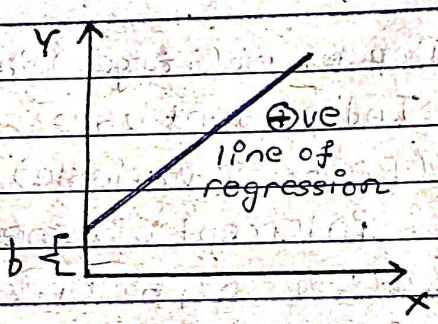
● Simple Linear Regression, If a single independent variable is used to predict the value of a numerical dependent variable.

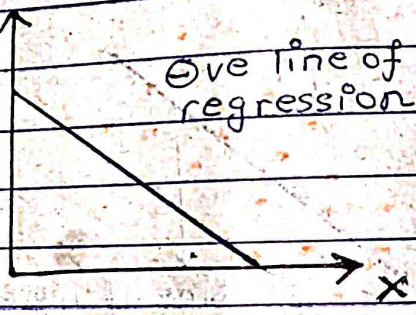
i.e

cgpa	package
7.1	3.5
4.7	1.2
8.9	4.3
8.1	8.9

cgpa → model → package.

- Linear regression Line, a line showing relationship b/w dependent and independent variables.
  - positive Linear relationship, if the dependent var. increases on y-axis, & independent increases on X.
  - negative Linear relationship, if the dependent var. decreases on y-axis, & independent increases on X.





$y = a_0 + a_1 x$

or

$mx + b$

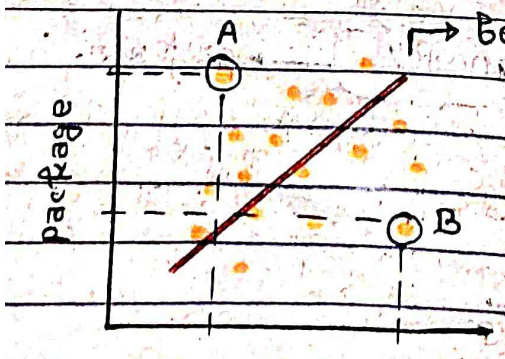
$y = -a_0 + a_1 x$

or

$mx - b$

- Finding the "Best-fit-Line", while working with linear regression that means error between predicted values and actual values should be minimized.

The best fit line will have the least error.



A: bad cgpa (good package)
B: good cgpa (bad package)

Outliers

Stochastic (or Undeterministic) Errors

Qualitative error, i.e may be interviews gone bad.

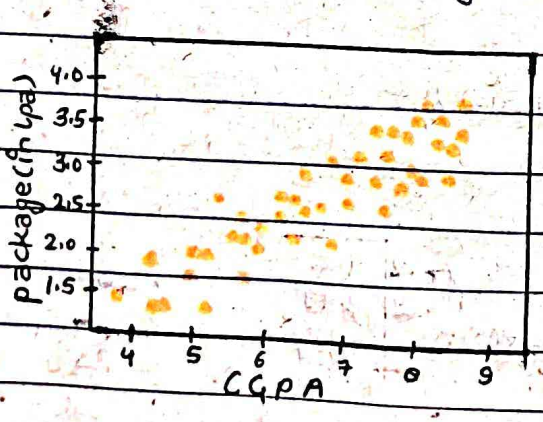
Code implementation:

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

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

	cgpa	package

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




```
X = df.iloc[:, 0:1]
```

	cgpa
0	6.89
1	5.12
2	7.82
3	7.42

```
y = df.iloc[:, -1]
```

0	3.26
1	1.98
2	3.25
3	3.67

```
from sklearn.model_selection import
                        train_test_split

# now we define the data into
# training-set & examine_set

# dividing data in 4-arrays

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
                 random_state=2)
```

20% of data goes to testing array.

```
# Algo

from sklearn.linear_model import
                        LinearRegression

lr = LinearRegression()

# model training starts

lr.fit(X_train, y_train)
```

X_test

	cgpa
112	8.58
29	7.15
182	5.88
...	...
53	6.47

y_test

112	4.10
29	3.49
182	2.08

testing

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

↓

112 के corresponding cgpa क्या है, after training.

op→ array([3.8911601])

→ similar to 4.10

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

↓

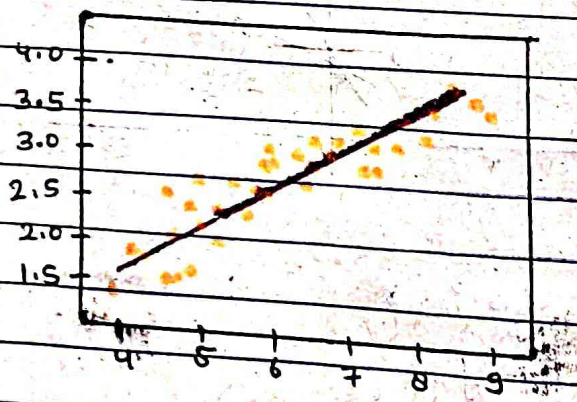
29 के corresponding cgpa क्या है, ~~after training~~

op→ array([3.09324469])

→ similar to 3.49.

plotting best-fit-line

plt.plot(X_test, lr.predict(X_test), color='red')



slope & intercept of line

m = lr.coef_ # slope-value(m)

op→ array([0.55795197])

b = lr.intercept_ # y-intercept

op→ 0.9611922242914 4

$y = mx + b$ → $m * 8.58 + b$

package ← cgpa → cgpa

o/p → array([3.09111601])

package.

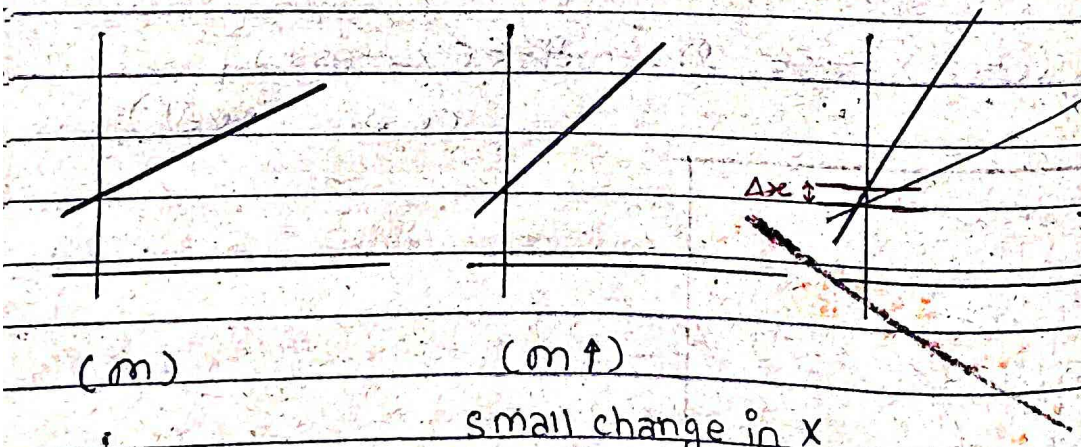
Linear reg is finding the
Best-fit-line for your model
↳ finding value of m & b.

$y = mx + b$

⇓

package = m x cgpa + b

m → weightage



(m) (m↑)

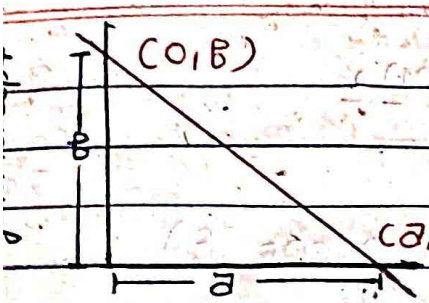
small change in x
scale up in y

ie Let, x = experience

package = m x experience + b

if experience = 0

package = (+b) → some fix sort of salary to fresher



$$\frac{x}{a} + \frac{y}{b} = 1$$

cgpa = 0
package = b

x-intercept

$$y = mx + b$$

How to find 'm' & 'b'?

closed-form expression
mathematical expression expressed using finite nos of standard operns
It may contain const, variable, opern (+, -, /, *) func (log, exp)

Non-closed form solution
but uses limit, differentiation, integration.

i.e. Gradient Descent

method: (OLS)
ordinary least squares
used by Scikit learn

$$b = \bar{y} - m\bar{x}$$

$y \rightarrow$ cgpa package
$\bar{y} \rightarrow$ mean of cgpa
$x \rightarrow$ cgpa
$\bar{x} \rightarrow$ mean of cgpa

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