(51) Gradient Descent from Scratch:

A first-order iterative optimization algorithm for finding a local minimum of a differentiable function.

→ considering m const.

$$\beta_{new} = \beta_{old} - \eta \cdot (\text{slope})$$

(η = 0.01)

learning rate

slope = $\frac{dL}{d\beta}$

no. of iterations = epochs

↑

when to stop.

$\eta \cdot \frac{dL}{d\beta}$ → step size.

→ Now m & β are both unknown.

(1) initialize random values of m & β.

say, $m = 1$ $\beta = 0$

(2) decide epochs & learning rate.

say epochs = 100 lr = 0.01

→ Apply same formula for both m & β.

But now, L will depend on both m & β. Therefore, in both equation, slope will be partial derivative of L(m,β)

$$\beta_{new} = \beta_{old} - \eta \frac{dL}{d\beta}$$

$$m_{new} = m_{old} - \eta \frac{dL}{dm}$$

$$\frac{dL}{d\beta} = \frac{d}{d\beta}\left(\sum (y_i - mx_i - \beta)^2\right)$$

$$\boxed{\frac{dL}{d\beta} = -\frac{2}{n}\sum (y_i - mx_i - \beta)}$$

$\sum (y_i - mx_i - \beta) = \sum (y_i - \hat{y}_i)$

and

$$\boxed{\frac{dL}{dm} = -\frac{2}{n}\sum (y_i - mx_i - \beta)x_i}$$

→ right learning rate is necessary to get optimum epochs, otherwise model will take too long to process.

(52) Variations / variants of GD:

(1) Batch GD
↓
considering whole data as 1 batch and the new values of m & β_0 are updated after batch processing

Stochastic GD
↓
updating after every row.

Mini-batch GD
↓
considering whole data into some no. of mini-batches & updating every time each mini-batch gets processed.

→ for multiple linear regression with n inputs.

$$\boxed{\frac{dL}{d\beta_0} = -\frac{2}{n}\sum (y_i - \hat{y}_i)}$$

$$\boxed{\frac{dL}{d\beta_j} = -\frac{2}{n}\sum (y_i - \hat{y}_i)x_{ij}}$$

doing this summation in 1 go, we need to use dot product.

→ Normal gradient descent is Batch gradient descent

- problem with batch GD:
  - for higher dim data with high epochs to run, there will large no. of derivative calculation increasing the model time.
  - → Hardware problem due to memory overflow.

(2) Stochastic GD: (faster)

→ 1 he epoch ke andar 1 row ke basis pe update hoga.

↳ minimize no. of epochs.

→ 1 row ke use ke karan memory overflow nhi hota.

→ random row selection & update.

→ Slightly imprecise prediction due to randomness.

→ Due to randomness, after reaching near to soln, it fluctuates. So to reduce this fluctuations, we use Learning schedules

↑ varying learning rate with epochs

(3) Mini-Batch Gradient Descent:

↳ creating batches in rows.

Say n = 1000

batches → 100 → size = 10

So, every epoch we do 100 updates.

* Session 1 & 2 on Regression Analysis

↳ The process of studying the relationship between X and y using stats and with the help of computed coefficients.

- steps :-

(1) Identify dependent and independent variables
(2) Collect and prepare the data.
(3) Visualize the data
(4) Check assumptions
(5) fit the LR model
(6) Interpret the model : Analyzing estimated regression coefficient, their std errors, t-values, p-values to determine statistical significance of relationship between dependent & independent variables.
(7) Validate the model
(8) Report results.

$$Y = f^*(X_1, X_2, \dots) + \varepsilon$$

dependent variables. → Y

$f^*(X_1, X_2, \dots)$ ↑ independent variables

ε ↑ irreducible error

$f(X_1, X_2, X_3 \dots)$ → true fn.

But there is deviation in fn. due to sample (as we don't have whole population) This also creates some error

i.e. $f() - f'()$

↳ reducible error.

True parameters → β_0 β_1 β_2 ....

computed parameters → b_0 b_1 b_2 .... based on sample

So, as we try to bring the values of b_i close to β_0 the error $f() - f'()$ reduces.

⇒ $Y = f'(X_1, X_2 \cdots) +$ reducible error + irreducible error

estimated fn. using model

* * * * *

→ Now, we will use hypothesis testing to know whether the estimated parameters are useful or not

→ Then we will calculate confidence intervals to get range of values of parameters.

* * * * *

* * * * *

→ To study how much each feature (X_i) contribute to Y.

- Prediction (what) ⇒ To give output of particular individual based on trained data.
- Inference ⇒ To give relation b/w features due to which we get the predicted value. (why)

$$\text{Prediction} \propto \frac{1}{\text{inference}}$$

Jiska prediction accha unka inference sahi nahi and vice versa.

- Statsmodel LR:

$$X \longrightarrow Y$$

① Is there a relationship (f-statistic)
② If yes, is it linear
③ If yes, how strong. (R2 score)
④ If it is strong, then how is the relationship of every individual component of X with y. (individual t-test)

```
import statsmodels.api as sm
X = sm.add_constant(X)   → for intercep.
model = sm.OLS(y, X).fit()
print(model.summary())
```

- Summary :-

→ Upper right corner ⇒ relationship b/w X & Y

→ Middle part ⇒ Individual relation b/w feature & y colm.

→ bottom part ⇒ assumptions.

① relationship b/w X & Y

- TSS, RSS and ESS :-

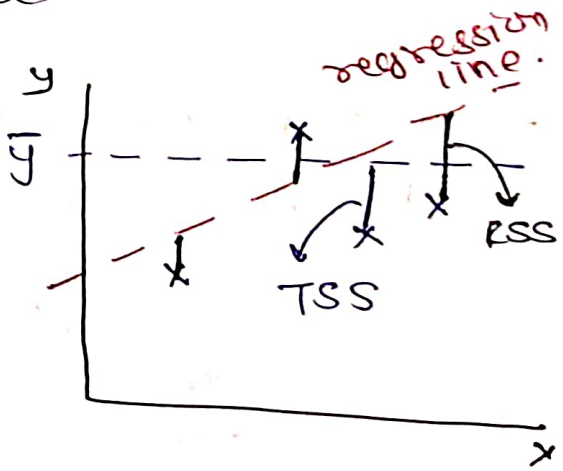
TSS → Total → variance around mean

RSS → Residual → variance around regression line.

ESS → Explained → TSS − RSS → utna variance jitna regression line ne explain kiya.

Sum of squares.

model itna variance explain nhi kar pa rha.



$$TSS = \sum (y_i - \bar{y})^2$$

$$RSS = \sum (y_i - \hat{y}_i)^2$$

$$ESS = TSS - RSS$$

due to both reducible & irreducible error.

- Degree of freedom.. $(n-1)$

$df\_total = df\_model + df\_residual$

$df\_model \rightarrow k$: no. of input cols or no. of independent

$df\_residual \rightarrow n-k-1$: no. of rows

* f-statistic :-

- f-test used to determine whether a linear regression model is statistically significant, meaning it provides a better fit to data than just using mean of dependent variable.

↳ varient of ANOVA test.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots$$

→ f-test for overall significance :-

(1) $H_0 : \beta_1 = \beta_2 = \cdots \beta_k = 0$

$H_a : \beta_1 \neq \beta_2 \neq \cdots \beta_k \neq 0$

↓

at least one independent variable contributes.

(2) fit LR model to get estimated regression coefficients.

(3) calculate TSS, ESS, RSS.

(4) Compute → MSR (Mean square regression)

$$MSR = \frac{ESS}{k}$$

→ MSE (Mean square Error)

$$MSE = \frac{RSS}{n-k-1}$$

(5) calculate f-statistic : $\frac{MSR}{MSE}$

(6) Determine p-value

(7) Derive results by comparing p-value $\in \alpha$.

* R^2 score :- (Coefficient of determination)

→ quantifies the proportion of variance in dependent variable that can be explained by independent variables.

$R^2 \longrightarrow 1$ (better fit of a model)

$$R^2 = \frac{ESS}{TSS}$$

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$\text{Adjusted } R^2 = 1 - \left[\frac{(1-R2)(n-1)}{(n-k-1)}\right]$$

→ R^2 always increases or stays same with addition of new predictor variables, regardless of those variables which do not contribute.

→ Adjusted R^2 score penalizes the model for adding unnecessary complexity.

* T-test :-

① $H_0 : \beta_k = 0$
$H_a : \beta_k \neq 0$ Same for β_0

② Estimate β_0, β_k using LR model → b_0, b_1

③ calculate std. errors for slope and intercept coefficients $[SE(\beta_0) \& SE(\beta_k)]$

$$SE(b_0) = \sqrt{\frac{\Sigma (y_i - \hat{y}_i)^2}{(n-2)} \left[\frac{1}{n} + \frac{\bar{X}^2}{\Sigma (X_i - \bar{X})^2} \right]}$$

$$SE(b_1) = \sqrt{\frac{\Sigma (y_i - \hat{y}_i)^2}{(n-2) \Sigma (X_i - \bar{X})^2}}$$

④ Compute t-teststic for b_0 & b_1

$$(t\_value)_{b_0} = \frac{b_0 - 0}{SE(b_0)}$$

$$(t\_value)_{b_1} = \frac{b_1 - 0}{SE(b_1)}$$

As taken in Null hypothesis.

* confidence interval:

$$CI\ b_0 = b_0 \pm t\text{-value} * SE(b_0)$$
$$CI\ b_1 = b_1 \pm t\text{-value} * SE(b_1)$$

calculate using

$(df = n-2)$

(α) &