Linear Regression - 4

Assumptions of Linear Regression :

a. **Assumption of Linearity**:
   linear relationship between the features x and target variable y

b. **Features are not multi-collinear:**

   What is collinearity?
   Ans: 2 features (f_1, f_2), have a linear relationship between them. $f_1 = \alpha f_2$

   What is Multicollinearity?
   Ans: the feature f_1 has collinearity across multiple features f_2, f_3, f_4

$$f_1 = \alpha_1 f_2 + \alpha_2 f_3 + \alpha_3 f_4$$

   Why is multi-collinearity a problem?

   Ans: MultiCollinearity → non-reliability on the feature importance and model interpretability. (instability in feature weights.)

   How to resolve multi-collinearity?
   Ans: Using Variance Inflation factor (VIF), defined as:

$$VIF\ for\ f_j = \frac{1}{1-R^2_j};\ where\ R^2\ is\ Rsquared$$

   VIF algorithm works as :

   - Calculate the VIF of each feature

   - if VIF >= 5 → **high Multicollinearity**

     - Remove the feature having the highest VIF

     - Recalculate the VIF for the remaining feature

- Again remove the next feature having the highest VIF

- Repeat till all VIF<5 or some number of iterations is reached

c. **Errors are normally distributed:**

   Used to ensure there are no outliers present in the data

d. **Heteroskedasticity should not exist:**

   Heteroskedasticity → unequal scatter of the error term → not having the same variance

   Why Heteroskedasticity is a problem?
   Ans: model inaccurate or outliers in the data.

   How to check Heteroskedasticity?

   Ans: Plotting a Residual plot → Errors $(y - \hat{y})$ vs prediction $(\hat{y})$

e. **No AutoCorrelation:**

   What is AutoCorrelation?
   Ans: When the current feature value depends upon its previous value

   Why is AutoCorrelation a problem?

   Ans: Linear regression assumes $\hat{y}_1 = f(x)$ has to be independent of $\hat{y}_2 = f(x + 1)$ → AutoCorrelation contradicts this assumption.

Is there any other way to solve Linear Regression?

Ans: Closed Form/ Normal Equation

Why use Normal Equations?

Ans: Finds the optimal weights without any iterating steps as done in Gradient Descent.

The optimal weights: $W = (X^T X)^{-1} X^T Y$

Where X → feature matrix: $R^{n\ (Sample\ size)\ \times d\ (d\ dimensional)}$, and Y → target vector: $R^{n\ (Sample\ size)\ \times 1}$

Why even use gradient descent?

Ans: $(X^T X)^{-1}$ → computationally expensive operation ⇒ Not used when the number of features is high.
Dimension of matrix multiplication: [(d x n) * (n x d) => (d x d)]