Interview Questions

1.What assumptions does linear regression make?

2.How do you interpret the coefficients?

3.What is R^2 score and its significance?

4.When would you prefer MSE over MAE?

5.How do you detect multicolinearity?

6.What is the difference between simple and multiple regression?

7.Can linear regression be used for classification?

8.What happens if you violate regression assumptions?

---

1. What assumptions does linear regression make?

Linear Regression makes several key assumptions that must be satisfied for the model to be valid and reliable. Violating these can lead to incorrect or misleading results.

✅ 1. Linearity

- **Assumption**: The relationship between the independent variables (X) and the dependent variable (y) is **linear**.
- **Why it matters**: If the relationship is nonlinear, predictions will be biased.
- **Check**: Scatter plots, residual vs. fitted plots.

✅ 2. Independence of Errors

- **Assumption**: The residuals (errors) are **independent** of each other.
- **Why it matters**: Violations (e.g., autocorrelation in time-series data) lead to underestimated standard errors.
- **Check**: Durbin-Watson test.

✅ 3. Homoscedasticity (Constant Variance of Errors)

- **Assumption**: The variance of the residuals is **constant** across all levels of the independent variables.
- **Why it matters**: If the variance changes (heteroscedasticity), predictions become unreliable.

- **Check**: Plot residuals vs. predicted values – the spread should be uniform.

✅ **4. No Multicollinearity**

- **Assumption**: Independent variables are **not highly correlated** with each other.
- **Why it matters**: High multicollinearity makes it difficult to determine the individual effect of each variable.
- **Check**: Correlation matrix or Variance Inflation Factor (VIF) – VIF > 5 indicates a problem.

✅ **5. Normality of Residuals**

- **Assumption**: The residuals (errors) follow a **normal distribution**.
- **Why it matters**: Important for valid hypothesis testing (e.g., confidence intervals, p-values).
- **Check**: Histogram or Q-Q plot of residuals.

✅ **6. No Significant Outliers or High Leverage Points**

- **Assumption**: Data points should not unduly influence the model.
- **Why it matters**: Outliers can distort regression coefficients and reduce model performance.
- **Check**: Cook's distance, leverage plots.

📌 **Summary Table:**

Assumption	What to Check	Tools / Techniques
Linearity	Linear pattern in scatter/residual plots	Scatter plot, Residual plot
Independence	No autocorrelation	Durbin-Watson test
Homoscedasticity	Constant spread of residuals	Residual vs. Fitted plot
No Multicollinearity	Low correlation between predictors	Correlation matrix, VIF

Assumption	What to Check	Tools / Techniques
Normality of Residuals	Bell-shaped residual distribution	Histogram, Q-Q plot
No Influential Points	No extreme outliers or leverage points	Cook's Distance, Leverage statistics

2. How do you interpret the coefficients?

In **linear regression**, interpreting the **coefficients** means understanding how each feature (independent variable) affects the target (dependent variable), while holding other variables **constant**.

Let's break it down clearly:

✅ **General Equation of Linear Regression:**

$y=\beta0+\beta1X1+\beta2X2+\cdots+\beta nXn$

- y: predicted value (e.g., house price)
- β0 : intercept (value of y when all X=0X = 0)
- βi : coefficient of the ithi^{th} feature
- Xi : value of the ithi^{th} feature
- ϵ : error term

🔍 **How to Interpret the Coefficients:**

Term	Interpretation
Intercept β0	The expected value of the target variable when all predictors are zero. Often has no practical meaning unless zero is within the range of your data.

Term	Interpretation
Slope / Coefficient Bi	The **change in the target variable (y)** for a **one-unit increase in Xi**, assuming all other variables remain constant.

🔢 **Example:**

Suppose your regression model is:

price=50000 + 100 × area + 5000 × bedrooms

- **Intercept (50000)**: Predicted price when area and bedrooms are 0 (may not make sense practically, but mathematically necessary).
- **Area Coefficient (100)**: For every **1 sq ft** increase in area, price increases by **₹100**, keeping bedrooms constant.
- **Bedrooms Coefficient (5000)**: For each additional bedroom, price increases by **₹5000**, assuming area is fixed.

🎯 **Important Notes:**

1. **Sign of Coefficient:**
    - Positive: Feature increases the target value.
    - Negative: Feature decreases the target value.
2. **Magnitude Matters:**
    - Larger absolute value = greater impact on the target.
    - But be careful if features are not standardized — scales can mislead.
3. **Units Matter:**
    - Always interpret coefficients **in the units of the variables**.
    - E.g., a coefficient of 0.003 on a “size in sq ft” feature means that per sq ft increase has small impact.
4. **Standardized Coefficients (optional):**

- If features are **standardized** (mean = 0, std = 1), you can compare coefficients directly to see which variable has the **most influence**.

---

3 . What is R^2 score and its significance?

R^2 score (pronounced *R-squared*) is a statistical measure that explains how well the **independent variables** in a regression model explain the **variability of the target variable**.

📘 **Definition (Formula):**

$$R2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

- SS_{res}: Sum of Squares of Residuals (prediction errors)
- SS_{tot}: Total Sum of Squares (total variance in the target)

🎯 **Interpretation:**

R^2 Value	Interpretation
1	Perfect fit — model explains **100%** of the variability.
0	Model explains **none** of the variability — no better than just predicting the mean.
< 0	Model is worse than predicting the mean (usually means a poor fit or inappropriate model).

📊 **Significance:**

- **High R^2**: Good model fit — most of the variance in the target is explained by the predictors.
- **Low R^2**: Poor model fit — much of the target variance is unexplained.

⚠️ **Important Considerations:**

1. **R^2 increases** with more features, even if they're not useful. So don't rely on it alone.
2. Use **Adjusted R^2** when comparing models with different numbers of features.
3. **R^2 doesn't tell you**:
    - Whether your model is biased
    - Whether the predictions are accurate in absolute terms
    - Whether the model overfits or underfits

🧠 **Example:**

Let's say your housing model returns:

R^2 Score: 0.85

This means **85% of the variance in house prices** can be explained by the variables in your model (e.g., area, bedrooms, bathrooms, etc.), and **15% is unexplained** (likely due to noise, unknown variables, or random error).

---

4. When would you prefer MSE over MAE?

Both **MSE (Mean Squared Error)** and **MAE (Mean Absolute Error)** are common metrics to evaluate regression models, but they behave differently.

🔍 **Definitions Recap:**

- **MAE (Mean Absolute Error):**

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Measures **average magnitude** of errors — treats all errors equally.

- **MSE (Mean Squared Error):**

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Squares the errors — gives more weight to **larger errors.**

🧠 **Prefer MSE over MAE when:**

Scenario	Why MSE is better
You want to penalize large errors more	MSE increases exponentially with large deviations, helping the model avoid big mistakes.
You are optimizing with gradient descent	MSE is smooth and differentiable, making it easier to compute gradients.
Outliers are important	If large errors must be avoided (e.g., in finance, medical), MSE highlights them more.
You're working with normally distributed errors	MSE works best under this assumption — aligns with Maximum Likelihood Estimation (MLE).

📌 **Example Use Cases:**

- **Predicting medical dosages:** Large errors could be life-threatening — use MSE.
- **Loan default prediction:** Large prediction mistakes can cost money — MSE helps penalize those.
- **Training deep learning models:** MSE is smoother for optimization.

⚠️ **Be Careful:**

- **MSE is sensitive to outliers** (can skew model).

- If **you want a robust metric against outliers**, use **MAE** or **Huber Loss**.

---

4. How Do You Detect Multicollinearity?

Multicollinearity occurs when **two or more independent (predictor) variables** in a regression model are **highly correlated**, meaning they carry **similar information**.

🔍 **Why is it a problem?**

- It **inflates standard errors** of coefficients.
- Makes it **hard to determine which variable** is affecting the dependent variable.
- **Unstable coefficients** — small changes in data → big changes in model.
- Can **mislead the model's interpretation**.

---

5. How to Detect Multicollinearity

✅ **1. Correlation Matrix (Heatmap)**

- If **correlation between features** > 0.8 or < -0.8, it's a red flag.
  - import seaborn as sns
  - import matplotlib.pyplot as plt
  - import numpy as np

  - # Compute correlation matrix
  - corr_matrix = df.corr()

  - # Plot heatmap
  - plt.figure(figsize=(10, 6))
  - sns.heatmap(corr_matrix, annot=True, cmap="coolwarm")

- plt.title("Correlation Matrix Heatmap")
- plt.show()

✅ **2. Variance Inflation Factor (VIF)**

- **VIF > 5 or 10** indicates high multicollinearity.
- VIF quantifies how much the **variance of a regression coefficient** is inflated due to multicollinearity.

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.tools.tools import add_constant
import pandas as pd



X = df[['feature1', 'feature2', 'feature3']]  # replace with your features
X = add_constant(X)



vif = pd.DataFrame()
vif["Variable"] = X.columns
vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
print(vif)
```

✅ **3. Condition Number (from statsmodels)**

- Large values (e.g., > 30) indicate multicollinearity.

```
import statsmodels.api as sm
model = sm.OLS(y, X).fit()
print(model.condition_number)
```

🖌️ **How to Fix Multicollinearity**

Solution	Description
❌ **Remove one of the correlated features**	If two features are very similar, drop one.

Solution	Description
🧠 **Combine features**	Use **PCA** or create a new variable (e.g., average of correlated ones).
⚙️ **Use Regularization**	**Ridge Regression** (L2 penalty) helps reduce impact of multicollinearity.
✅ **Feature selection**	Use techniques like **Backward Elimination**, **LASSO**, etc.

6. What is the difference between simple and multiple regression?

Both **simple** and **multiple** linear regression are used to model the relationship between **independent variables (features)** and a **dependent variable (target)**. However, they differ in the **number of predictors** used.

🔹 **1. Simple Linear Regression**

Feature	Description
📌 Predictors	Only **one** independent variable

Feature	Description
📈 Equation	$Y = \beta_0 + \beta_1 * X + \varepsilon$
🔍 Goal	Find the best-fitting **line** through the data
📊 Use Case	Predicting house price based on **area** only

Example:

Predicting price from area only

price = $\beta_0 + \beta_1$ * area

🔹 2. Multiple Linear Regression

Feature	Description
📌 Predictors	**Two or more** independent variables
📈 Equation	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n * X_n + \varepsilon$
🔍 Goal	Find the best-fitting **hyperplane** in n-dimensional space
📊 Use Case	Predicting house price based on **area, bedrooms, location, etc.**

Example:

Predicting price from area, bedrooms, and age

price = $\beta_0 + \beta_1$*area + β_2*bedrooms + β_3*age

🔁 Key Differences

Feature	Simple Regression	Multiple Regression
📌 Number of Predictors	1	2 or more
📊 Complexity	Low	Higher
🧠 Captures interactions	No	Yes
📈 Output Shape	Straight line (2D)	Hyperplane (multi-D)

Feature	Simple Regression	Multiple Regression
📉 Risk of Overfitting	Low	Higher (if too many features)

🔎 **When to Use Which?**

- ✅ **Simple Regression**: When you are confident that **one variable** is a strong predictor.
- ✅ **Multiple Regression**: When **multiple features** contribute to the outcome (real-world problems).

---

7. Can Linear Regression Be Used for Classification?

Linear Regression **is not suitable** for classification tasks.

✅ **Why?**

Linear Regression is designed to predict **continuous numerical values**, not discrete class labels. Classification problems, on the other hand, require predicting **categorical outcomes** like:

- Spam or not spam (0 or 1)

- Pass or fail
- Cat or dog

⚠️ What Happens If You Use Linear Regression for Classification?

- It can **output values like 0.42, -0.3, 1.8**, etc.
- These are not valid **class labels**.
- You'd need to apply a **threshold manually**, such as:
  - if prediction > 0.5:
  - return 1
  - else:
  - return 0
- This approach is **not robust** and performs poorly, especially with **imbalanced data**.

✅ What Should You Use Instead?

Use **classification models** that are designed for categorical outputs:

Model	Use Case
Logistic Regression	Binary Classification (0/1)
Decision Trees	Multi-class or binary
Random Forest	Ensemble classifier
SVM (Support Vector Machines)	Binary or multi-class
K-Nearest Neighbors	Classification by similarity

🔁 Logistic Regression vs Linear Regression

Feature	Linear Regression	Logistic Regression
🔢 Output	Continuous values	Probability between 0 and 1
📊 Use Case	Regression problems	Classification problems
➗ Loss Function	Mean Squared Error	Log Loss (Cross-Entropy)
📈 Output Transformation	None	Uses **Sigmoid function**

---

8. What Happens If You Violate Regression Assumptions?

Violating the key assumptions of linear regression can lead to **biased, misleading, or unreliable** results. Here's what happens for each assumption if it's not met:

- **1. Linearity Assumption**

📌 **Assumption:**
The relationship between independent variables and the dependent variable is linear.

🚫 **Violation Consequence:**

- Model will **underfit** the data.
- Predictions and coefficients will be **inaccurate**.
- R^2 will be low even if variables are related **non-linearly**.

✅ **Solution:**

- Use **polynomial regression**, **log transformation**, or **non-linear models**.

🔹 **2. Independence of Errors**

📌 **Assumption:**
The residuals (errors) are independent of each other.

🚫 **Violation Consequence:**

- Usually occurs in **time series data** (autocorrelation).
- Leads to **overconfident predictions** and **underestimated standard errors**.

✅ **Solution:**

- Use **Durbin-Watson test** to detect autocorrelation.
- Consider **Time Series Models** like ARIMA if violated.

🔹 **3. Homoscedasticity (Constant Variance of Errors)**

📌 **Assumption:**
The variance of residuals is constant across all levels of the independent variables.

🚫 **Violation Consequence:**

- Leads to **inefficient estimates**.

- Makes standard errors **wrong**, leading to **invalid confidence intervals** and p-values.

✅ **Solution:**

- Use **log transformation**.
- Use **Weighted Least Squares (WLS)** if necessary.

🔹 **4. Normality of Errors**

📌 **Assumption:**
The residuals are normally distributed (especially important for inference).

🚫 **Violation Consequence:**

- Affects **hypothesis testing**, p-values, and **confidence intervals**.
- Predictions may still be okay, but interpretation is flawed.

✅ **Solution:**

- Use **log/sqrt transformation** of the target variable.
- Try **robust regression** or non-parametric models.

🔹 **5. No Multicollinearity**

📌 **Assumption:**
Independent variables are not too highly correlated.

🚫 **Violation Consequence:**

- Coefficient estimates become **unstable and unreliable**.
- Hard to determine the **individual effect** of predictors.

✅ **Solution:**

- Use **VIF (Variance Inflation Factor)** to detect it.
- Drop or combine correlated variables.
- Try **regularization** (like **Ridge** or **Lasso** regression).

⚠️ **Summary Table:**

Assumption	Violation Effect	Fix
Linearity	Inaccurate predictions	Add non-linear terms, transform
Independence	Biased SEs, wrong inference	Time series models, check autocorrelation
Homoscedasticity	Wrong SEs, inefficient estimates	Transform or use WLS
Normality	Inaccurate inference	Transform target or errors
No Multicollinearity	Unstable coefficients	Remove/merge vars, use Ridge/Lasso