**Interview Questions for Multiple linear Regression:**

**1. What is Normalization & Standardization and how is it helpful?**  
**Ans**.

**Normalization** and **Standardization** are two commonly used techniques in data preprocessing, particularly in the context of machine learning and statistical modeling. Both techniques are used to adjust the values of numerical features to ensure that they have a consistent scale, which can significantly improve the performance of models.

**Normalization**

**Normalization** refers to the process of rescaling the values of features so that they fall within a specific range, typically [0, 1] or [-1, 1].

**Formula:**

The most common method of normalization is **Min-Max Normalization**:

X′=X−XminXmax−XminX' = \frac{X - X\_{min}}{X\_{max} - X\_{min}}X′=Xmax​−Xmin​X−Xmin​​

* XXX is the original value of the feature.
* XminX\_{min}Xmin​ and XmaxX\_{max}Xmax​ are the minimum and maximum values of the feature, respectively.
* X′X'X′ is the normalized value.

**When to Use:**

* **When the features have different scales**: Normalization ensures that no single feature dominates the model just because it has a larger range of values.
* **For algorithms like K-Nearest Neighbors (KNN) and Neural Networks**: These algorithms are sensitive to the scale of data and can benefit from normalization.

**Standardization**

**Standardization** (or Z-score normalization) transforms the data such that it has a mean of 0 and a standard deviation of 1. This process centers the data and scales it according to the standard deviation.

**Formula:**

The standardization process is typically done using the following formula:

X′=X−μσX' = \frac{X - \mu}{\sigma}X′=σX−μ​

* XXX is the original value of the feature.
* μ\muμ is the mean of the feature.
* σ\sigmaσ is the standard deviation of the feature.
* X′X'X′ is the standardized value.

**When to Use:**

* **When the data follows a Gaussian (normal) distribution**: Standardization works well when the features are normally distributed.
* **For algorithms that assume data is normally distributed**: Algorithms like Linear Regression, Logistic Regression, and Support Vector Machines (SVMs) often assume or perform better when the data is standardized.

**How Normalization and Standardization are Helpful**

1. **Improving Model Performance**:
   * **Speed of Convergence**: In optimization-based algorithms like Gradient Descent (used in Neural Networks, Linear Regression), normalized or standardized data can lead to faster convergence.
   * **Accuracy**: Some machine learning algorithms, especially distance-based ones like KNN or SVMs, can perform poorly if the features are on different scales. Normalization and standardization help ensure that each feature contributes equally to the model's predictions.
2. **Handling Different Scales**:
   * **Consistency**: Features in a dataset might have vastly different scales (e.g., age might range from 0 to 100, while income might range from thousands to millions). Scaling them helps maintain consistency.
   * **Prevention of Dominance**: Without scaling, features with larger ranges could dominate the model’s objective function, leading to suboptimal results.
3. **Improving Interpretability**:
   * **Ease of Comparison**: After normalization or standardization, the features are on a comparable scale, making it easier to interpret coefficients in models like linear regression.
4. **Dealing with Outliers**:
   * **Normalization**: It compresses the range of features and can reduce the impact of outliers on the model.
   * **Standardization**: It keeps the data in a standard normal distribution, which may be less affected by outliers depending on the context.

2. What techniques can be used to address multicollinearity in multiple linear regression?  
Ans.   
**Multicollinearity** in multiple linear regression occurs when two or more independent variables are highly correlated, meaning that they provide redundant information. This can cause problems because it makes it difficult to determine the individual effect of each variable on the dependent variable, leading to unstable coefficient estimates and reduced model interpretability.

**Techniques to Address Multicollinearity**

1. **Remove Highly Correlated Predictors**:
   * **Identify and Remove**: If two or more predictors are highly correlated (e.g., correlation coefficient > 0.8), consider removing one of them. This simplifies the model without losing much information.
   * **Variance Inflation Factor (VIF)**: Calculate the VIF for each predictor. A VIF value greater than 10 indicates high multicollinearity. Removing predictors with high VIF values can help reduce multicollinearity.
2. **Combine Correlated Predictors**:
   * **Principal Component Analysis (PCA)**: PCA is a dimensionality reduction technique that combines correlated variables into a smaller number of uncorrelated components. These components can then be used as predictors in the regression model.
   * **Factor Analysis**: Similar to PCA, factor analysis can be used to identify underlying factors that explain the correlations among variables. These factors can replace the original correlated variables.
3. **Ridge Regression (L2 Regularization)**:
   * **Ridge Regression**: This technique adds a penalty to the regression model proportional to the square of the magnitude of the coefficients. This penalty term shrinks the coefficients of correlated predictors, helping to reduce multicollinearity. Ridge regression is particularly useful when all predictors are potentially valuable and you do not want to remove any of them.
4. **Lasso Regression (L1 Regularization)**:
   * **Lasso Regression**: Lasso adds a penalty equal to the absolute value of the magnitude of coefficients, which can shrink some coefficients to exactly zero. This effectively removes some predictors from the model, thus addressing multicollinearity while also performing feature selection.
5. **Elastic Net Regression**:
   * **Elastic Net**: This technique combines both L1 (Lasso) and L2 (Ridge) regularization. Elastic Net is useful when there are multiple correlated predictors and you want to apply both feature selection (Lasso) and coefficient shrinkage (Ridge).
6. **Centering the Variables**:
   * **Mean-Centering**: Subtract the mean of each predictor from the predictor itself, which can help reduce multicollinearity, especially when it is caused by interactions between variables or polynomial terms. Centering can make the model more interpretable by reducing the correlation between predictors.
7. **Collect More Data**:
   * **Increase Sample Size**: If possible, collecting more data can help reduce the effect of multicollinearity. With more data, the model has more information to distinguish between the effects of correlated predictors.
8. **Partial Least Squares (PLS) Regression**:
   * **PLS Regression**: PLS is a technique that finds the linear combinations of predictors that have maximum covariance with the dependent variable. It is useful when dealing with multicollinearity, as it reduces the predictors to a smaller set of uncorrelated components while retaining the most relevant information.
9. **Remove One of the Variables in Pairs**:
   * **Drop One Predictor**: When two variables are highly correlated, you can choose to drop one of them. The decision on which one to drop can be based on domain knowledge, importance to the model, or practical considerations.

**Summary:**

To address multicollinearity in multiple linear regression, techniques such as removing highly correlated predictors, combining them using PCA or factor analysis, and applying regularization methods like Ridge or Lasso regression can be employed. Additionally, increasing the dataset size, centering variables, and using methods like PLS regression can also help mitigate the issue. The choice of technique depends on the specific context and the goals of the analysis.

3. Ensure to properly comment your code and provide explanations for your analysis. 🡪 Done in code itself

4. Include any assumptions made during the analysis and discuss their implications. 🡪 Done in code itself