Favorite Regression QUESTIONNAIRE

By Shivani Bhatia

What is Linear Regression?

It is a type of Supervised Learning where we compute a linear relationship between the predictor and response variable.

The equation for linear regression:

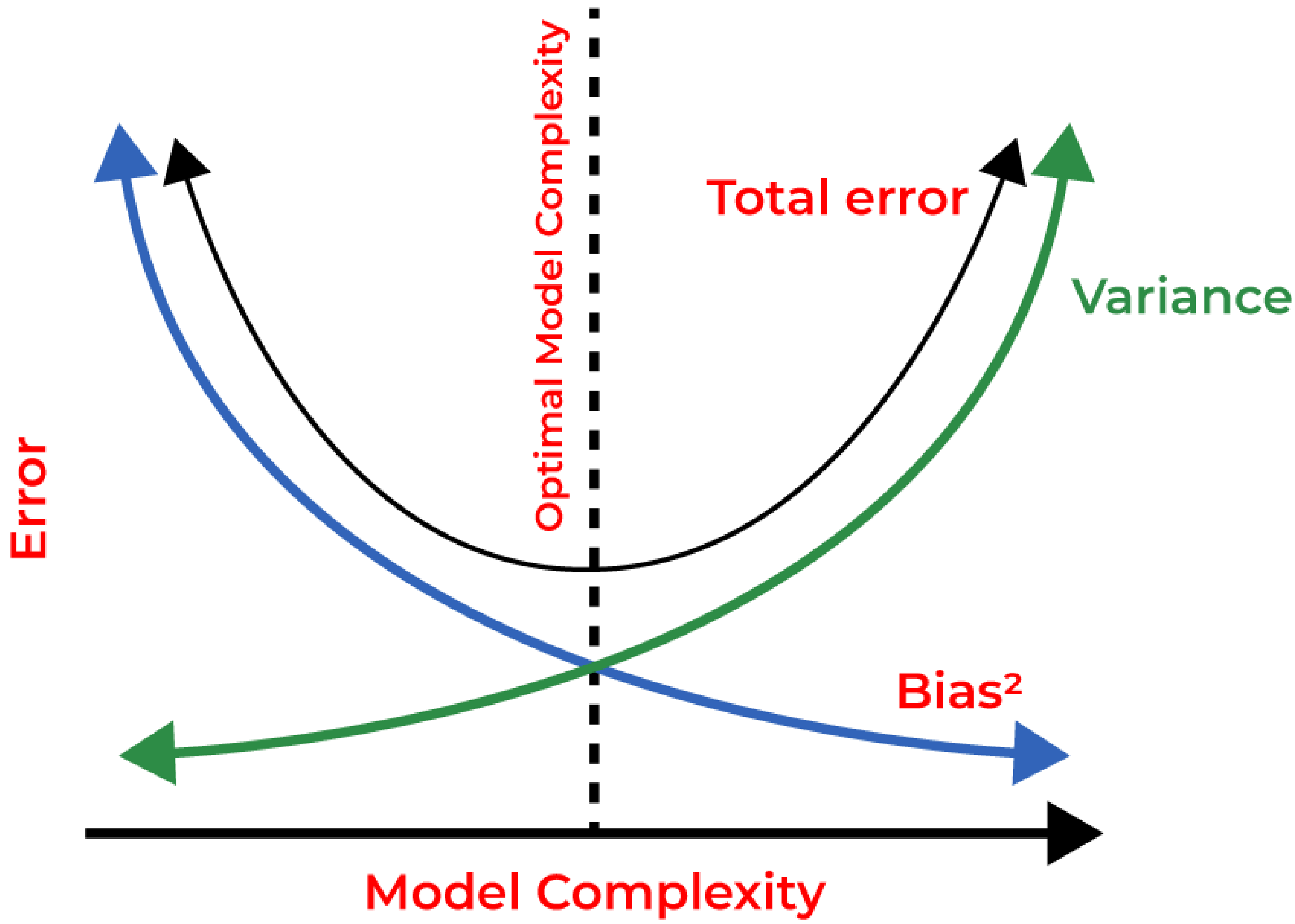
$$\underbrace{Y_i}_{\text{Dependent Variable}} = \overbrace{\beta_0}^{\text{Constant/Intercept}} + \underbrace{\beta_1}_{\text{Slope/Coefficient}} \overbrace{X_i}^{\text{Independent Variable}}$$

What are the assumptions made in Linear Regression?

1. **Linearity:** The relationship between the feature set and the target variable is linear.
2. **Homoscedasticity:** The variance of residuals is constant.
3. **Independence:** All observations are independent of one another.
4. **Normality:** The distribution of Y is assumed to be normal.

Explain Bias Variance Trade-off.

The goal of any supervised machine learning algorithm is to have low bias and low variance to achieve good prediction performance.



Bias

- Error from overly simplistic models
- High Bias: Underfitting (model misses patterns)

Variance

- Error from overly complex models
- High Variance: Overfitting (model captures noise)

Tradeoff

- Balance complexity to minimize both bias and variance
- Goal: Achieve good generalization on unseen data

Normally, as we increase the complexity of your model, we will see a reduction in error due to lower bias in the model. However, this only happens until a particular point. As we continue to make your model more complex, we end up overfitting our model and hence our model will start suffering from high variance.

What is R-Square and how is it used?

An R-square value shows how well the model predicts the outcome of the dependent variable. R-Square values range from 0 to 1.

An R-square value of 0 means that the model explains or predicts 0% of the relationship between the dependent and independent variables.

A value of 1 indicates that the model predicts 100% of the relationship, and a value of 0.5 indicates that the model predicts 50%, and so on.

The formula below is mostly used to find the value of R-Square:

$R^2 = 1 - RSS/TSS$

where,

- R^2 = coefficient of determination
- RSS = sum of squares of residuals
- TSS = total sum of squares

Explain Linear Regression Pitfalls.

When we fit a linear regression model to a particular data set, many problems may arise. The most common among these are the following:

Non-constant variance: The variability of the residuals is not the same across all levels of the independent variables, violating the homoscedasticity assumption.

Autocorrelation and time series issue: The residuals are correlated with each other, often occurring in time series data, leading to biased estimates.

Multicollinearity: Independent variables are highly correlated with each other, making it difficult to isolate the individual effect of each predictor.

Overfitting: The model is too complex, capturing noise in the data and performing well on training data but poorly on unseen data.

Extrapolation: Making predictions outside the range of the training data, where the model's assumptions may no longer hold true.

How to check multicollinearity in the dataset?

Variance Inflation Factor (VIF): Calculate the VIF for each predictor. A VIF value greater than 10 indicates high multicollinearity.

$$\text{VIF} = \frac{1}{1 - R^2}$$

Correlation Matrix: Compute the pairwise correlation coefficients between the predictors. High absolute values (close to 1 or -1) indicate potential multicollinearity.

Tolerance: Tolerance is the reciprocal of VIF (1/VIF). A tolerance value below 0.1 indicates high multicollinearity.

What is Regularization?

Regularization is a technique used in machine learning to prevent overfitting by adding a penalty to the model's complexity in the form of additional terms in the objective function. In the context of hyperparameters, regularization involves tuning these additional terms to achieve a balance between fitting the training data well and keeping the model as simple as possible.

Explain L1 and L2 Regularization.

L1 Regularization (Lasso)

- **Definition**: Adds a penalty equal to the absolute value of the coefficients.
- **Mathematical Form**: $\text{Loss}+\lambda\sum | wi |$
- **Use**: Effective for feature selection as it can shrink some coefficients to exactly zero, effectively removing irrelevant features.

L2 Regularization (Ridge)

- **Definition**: Adds a penalty equal to the square of the coefficients.
- **Mathematical Form**: $\text{Loss}+\lambda\sum wi^2$
- **Use**: Tends to distribute the error among all features, preventing any single feature from dominating the model.

Combined Use

- **Elastic Net**: Combines both L1 and L2 regularization to leverage the benefits of both methods.

Both regularizations help prevent overfitting by penalizing large coefficients, ensuring the model generalizes better to unseen data.

How to test the LinearityAssumption in Regression?

To test the linearity assumption in regression:

- **Residual Plot**: Plot the residuals (differences between predicted and actual values) against the predicted values. If the plot shows a random scatter without any clear pattern, the linearity assumption is likely satisfied. Patterns like curves suggest non-linearity.

- **Component+Residual Plot (CERES)**: Plot the residuals of the model against each predictor. Look for a random scatter. If you see patterns, it suggests that the relationship might not be linear.

- **Fit a Polynomial Model**: Add polynomial terms to your model and compare its performance. If the polynomial model improves significantly, it might indicate that the original relationship wasn't purely linear.

Difference between homoscedasticity and heteroskedasticity.

Homoskedasticity means constant variance of errors, while heteroskedasticity means varying variance.

In the case of homoskedasticity, residuals scatter randomly around zero in a residual plot.

In the case of heteroskedasticity residuals display a pattern (e.g., funnel shape) in a residual plot, indicating that the variance of errors changes systematically with the predictors.

Difference between logistic and linear regression.

Linear Regression: Predicts a continuous outcome variable (e.g., predicting house prices).

Logistic Regression: Predicts a categorical outcome variable, often binary (e.g., predicting whether an email is spam or not).

Explain different approaches to feature selection while building a regression model.

Try All Possible Combinations:

Concept: Evaluate every feature subset.

Challenge: Impractical for large datasets due to high computation.

Manual Feature Elimination:

- **Process:**
  - Build a model with all the features.
  - Remove the least helpful or redundant features based on p-values or correlations.
  - Rebuild and repeat.

- Pros: Detailed, hands-on approach.
- Cons: Time-consuming.

Automated Approaches:

- Recursive Feature Elimination (RFE): Iteratively removes the least important features.

- Forward/Backward/Stepwise Selection: Adds or removes features based on criteria like AIC.

Balanced Approach

- Combine Methods: Use automated methods for initial selection and manual methods for fine-tuning to get the best features for your model.

How to handle categorical variables in multiple linear regression?

Handling categorical variables in Multiple Linear Regression (MLR) involves converting them into a format that the model can interpret.

1. **One-Hot Encoding**:
    - **Concept**: Create a binary column for each category in the variable. Each column indicates the presence (1) or absence (0) of the category.

    - **Example**: For a "Color" variable with categories "Red," "Blue," and "Green," create three columns: "Color_Red," "Color_Blue," and "Color_Green."

2. **Label Encoding**:
    - **Concept**: Assign a unique integer to each category. This method is simpler but may imply an ordinal relationship that doesn't exist.

    - **Example**: "Red" = 1, "Blue" = 2, "Green" = 3.

3. **Dummy Encoding**:
    - **Concept**: Similar to one-hot encoding but excludes one category to avoid multicollinearity (i.e., the dummy variable trap).

    - **Example**: For "Color" with "Red," "Blue," and "Green," use "Color_Red" and "Color_Blue," and omit "Color_Green."

4. **Frequency Encoding**:
    - **Concept**: Replace categories with their frequency or occurrence count in the dataset.

    - **Example**: If "Red" appears 50 times, "Blue" 30 times, and "Green" 20 times, replace "Color" with these frequencies.

5. **Target Encoding (Mean Encoding)**:
    - **Concept**: Replace categories with the mean of the target variable for each category.

    - **Example**: For a binary outcome, calculate the mean outcome for each category and replace the category with this mean.

6. **Ordinal Encoding**:
    - **Concept**: Used for categorical variables with a meaningful order. Assign integers based on the order.

    - **Example**: "Low" = 1, "Medium" = 2, "High" = 3.

Choosing the Method

- **One-Hot Encoding**: Best for nominal variables without intrinsic order.

- **Label Encoding**: Suitable for ordinal variables with a clear order.

- **Dummy Encoding**: Prevents multicollinearity but can create more columns.

- **Frequency and Target Encoding**: Useful for handling high-cardinality categories but may require careful handling to avoid introducing bias.

Mention some ways to handle non-linearity.

To handle non-linear relationships in regression, you can use the following methods:

1. **Polynomial Regression:**

    - Concept: Include polynomial terms (e.g., x2,x3x^2, x^3x2,x3) in the regression model.

    - Usage: Captures curved trends in the data.

2. **Data Transformation:**

    - Concept: Apply transformations like logarithm, square root, or exponential to variables.

- Usage: Makes relationships linear, allowing the use of linear regression on transformed data.

3. **Non-Linear Regression Models:**

    - Concept: Fit non-linear equations directly to the data (e.g., exponential, logistic).

    - Usage: Suitable when the specific non-linear relationship is known and needs to be modeled directly.

------------------**THE END** ------------------