**All Topics to Cover Linear Regression**

**Beginner Level:**

1. Introduction to Linear Regression

* What is Linear Regression?
  + Linear Regression is a **supervised learning** algorithm used to model the relationship between a **dependent variable (target)** and one or more **independent variables (features)**. The goal is to fit a line (or hyperplane) that minimizes the distance between the predicted and actual values.
* Use cases of Linear Regression
  + Predicting housing prices, stock market trends, sales forecasting, etc.
* Types of Linear Regression (Simple, Multiple)
  + **Simple Linear Regression**: One independent variable.
  + **Multiple Linear Regression**: More than one independent variable.

2. Simple Linear Regression

* Linear equation: y = mx + c
  + Where
    - y = dependent variable (the variable you want to predict)
    - x = independent variable (the variable used for prediction)
    - m = slope of the line (how much y changes for a unit change in x)
    - c = intercept (the value of y when x=0)

3. Assumptions of Linear Regression

* **Linearity:** The relationship between the independent and dependent variables is linear (straight line).
* **Independence:** The observations should be independent of each other.
* **Homoscedasticity (Constant Variance):** The variance of the errors (residuals) is constant across all levels of the independent variable.
* **Normality of Errors:** The errors (differences between predicted and actual values) should be normally distributed.
* **No multicollinearity (for multiple linear regression):** The independent variables should not be too highly correlated with each other.

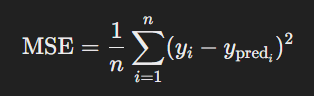
4. Ordinary Least Squares (OLS)

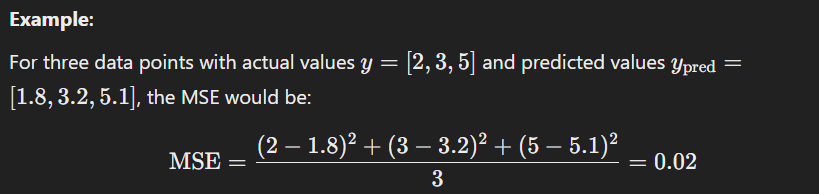
* - Derivation of OLS
* - Concept of minimizing residuals
* - Error term interpretation

5. Metrics to Evaluate Linear Regression

Once the model is built, we need to evaluate how well it performs. Here are common evaluation metrics:

**a. Mean Squared Error (MSE):**

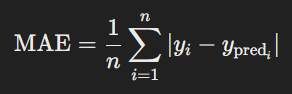
It measures the average squared difference between the actual and predicted values.

* **Lower MSE** means better model performance.

**b. Root Mean Squared Error (RMSE):**

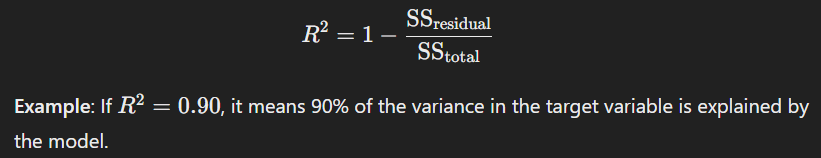
It is the square root of MSE, which brings the error metric back to the original scale of the data.

**c. Mean Absolute Error (MAE):**

It measures the average absolute difference between actual and predicted values.

**d. R-squared (R²): (need clarity)**

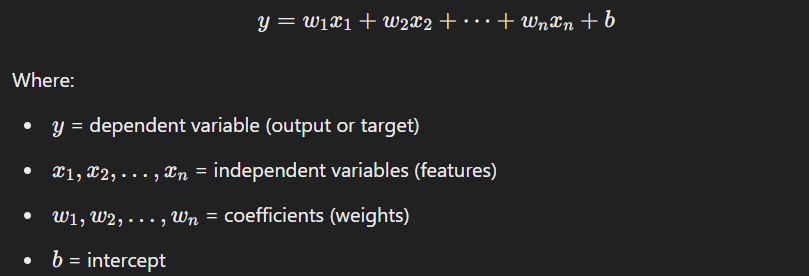
This metric indicates the proportion of the variance in the dependent variable that is predictable from the independent variable(s). R-squared ranges from 0 to 1.

****

**Intermediate Level:**

6. Multiple Linear Regression

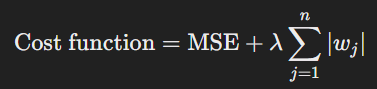
* Expanding to multiple features: When there are multiple independent variables, we extend the simple linear regression to **Multiple Linear Regression**.
* Matrix formulation of linear regression



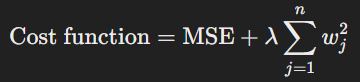
* Interpretation of multiple coefficients
  + In **Multiple Linear Regression**, the goal is to find the optimal values of w1, w2, …, wn and b that minimize the error (difference between predicted and actual values).

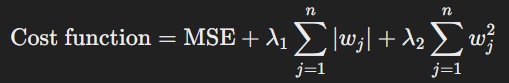
7. Regularization in Linear Regression

Regularization techniques are used to **prevent overfitting** by adding a penalty to the size of the coefficients. This helps the model generalize better to unseen data.

* L1 Regularization (Lasso)
  + In **Lasso (Least Absolute Shrinkage and Selection Operator)**, we add the **absolute value** of the coefficients as a penalty to the cost function.

Where λ is the regularization parameter controlling the strength of the penalty.

* + **Effect**: Lasso regression can shrink some coefficients to zero, effectively performing **feature selection**.
  + Example
    - In predicting house prices, if the distance to the city becomes irrelevant, Lasso might set its coefficient to 0.
* L2 Regularization (Ridge)
  + In **Ridge Regression**, we add the **square of the coefficients** as a penalty.
  + **Effect**: Ridge regression doesn’t set coefficients to 0 but makes them smaller, which reduces the risk of overfitting.
* Elastic Net Regularization
  + **Elastic Net** is a combination of **L1** and **L2** regularization:



* Use cases for regularization (handling overfitting)

8. Feature Selection and Engineering

Feature engineering involves **creating new features** or transforming existing ones to improve the model’s performance.

* Interaction terms
  + Sometimes the relationship between the dependent variable and independent variables is not purely linear. Interaction terms allow for the relationship between two variables to be modeled.
  + **Example:** Predicting house prices where the effect of the number of bedrooms depends on the size of the house. The interaction term could be Size × Number of Bedrooms.
* Polynomial features
  + Polynomial regression is a type of regression in which the relationship between the independent variable xxx and the dependent variable y is modeled as an n-degree polynomial.
  + **Example:** If the relationship between house price and size isn’t linear but quadratic, you could use polynomial features: **Size** and **Size^2**.

9. Multicollinearity and Its Effects

**Multicollinearity** occurs when independent variables are highly correlated, which can make the model unstable and cause issues in interpreting the coefficients.

* Variance Inflation Factor (VIF)
  + VIF is a measure used to detect multicollinearity. A high VIF (>10) indicates a high degree of correlation between independent variables.
  + **Example:** If house size and number of bedrooms are highly correlated (e.g., larger houses usually have more bedrooms), the model might struggle to distinguish their individual effects on house prices.
* How to detect and handle multicollinearity

10. Bias-Variance Trade-off

The **Bias-Variance Trade-off** is a key concept in model performance

* **Bias**: Error due to overly simplistic models (underfitting).
* **Variance**: Error due to overly complex models (overfitting).

The goal is to find the right balance between **bias** and **variance** to improve generalization.

#### ****Example****:

* **Underfitting**: A straight line (simple model) might not capture the trend in the house prices properly.
* **Overfitting**: A model with too many polynomial terms might fit the noise in the data and not generalize to new houses.

**Advanced Level:**

11. Gradient Descent Optimization

* - Batch Gradient Descent
* - Stochastic Gradient Descent (SGD)
* - Mini-Batch Gradient Descent
* - Convergence criteria, learning rate, and its impact

12. Advanced Assumptions Testing

* - Breusch-Pagan Test (for heteroscedasticity)
* - Durbin-Watson Test (for autocorrelation)
* - Shapiro-Wilk Test (for normality of residuals)

13. Diagnostics in Linear Regression

* - Residual plots
* - Cook’s distance and leverage points
* - Outliers and influential observations

14. Generalized Linear Models (GLMs)

* - Moving beyond linearity (Logistic regression, Poisson regression)
* - Link functions and exponential family distribution

15. Handling Complex Data with Linear Regression

* - Interaction terms and higher-order polynomial regression
* - Robust regression techniques (e.g., Huber regression)
* - Ridge regression, Lasso, and Elastic Net

16. Bayesian Linear Regression

* - Bayesian interpretation of regression
* - Priors and posteriors in linear models

17. Sparse Linear Regression Models

* - Understanding LASSO as sparse regression
* - Use cases for feature selection using L1 regularization

18. Advanced Optimization Techniques

* - Regularization paths (LARS algorithm)
* - Coordinate descent optimization for Lasso

19. Implementing Linear Regression in Python

* - Using libraries like `scikit-learn`, `statsmodels`, `pandas`, and `numpy`
* - Manual implementation from scratch

20. Applications of Linear Regression in Real-World Problems

* - Time series forecasting (e.g., ARIMA models)
* - Economic modeling, marketing analysis, etc.

Terms

1. **Residual:** Difference between the actual value and predicted value