**Linear Regression**

Linear Regression is a statistical method used in machine learning to model the relationship between a dependent variable (target) and one or more independent variables (predictors). It fits a line (in simple regression) or a hyperplane (in multiple regression) that minimizes the error between predicted and actual values.

**Assumptions of Linear Regression**

Linearity: The relationship between the dependent and independent variables is linear.

Independence: Observations are independent of each other.

Homoscedasticity: Constant variance of residuals (errors) across all levels of the independent variables.

Normality: Residuals are normally distributed.

No Multicollinearity: Independent variables should not be highly correlated.

**The formula for Multiple Linear Regression with OLS (Ordinary Least Square Method) is:**

**𝑌=𝑏1𝑋1+𝑏2𝑋2+...+𝑏𝑝𝑋𝑝+c**

**Here's what the variables in the formula represent:**

* Y: The dependent variable, or predictive value
* X1, X2, and so on through Xp: The independent variables, or predictive values, that cause a change in Y
* c: Intercept . It is the Y-value when all the independent variables are equal to zero
* b1, b2, and so on through bp: The regression coefficients

**Multiple Linear Regression with SGD**

Stochastic Gradient Descent (SGD) is an iterative optimization technique to find the best-fit line. Instead of minimizing the cost over the entire dataset in one step (as OLS does), it updates SGD updates parameters iteratively by minimizing the loss function. Also ,Instead of computing the gradient for the entire dataset (batch gradient descent), SGD updates the parameters for each data point.

**Advantages of SGD**

Efficient for large datasets as it processes one sample at a time.

Avoids the computational cost of inverting matrices (as in OLS).

Can converge faster for certain problems, especially with appropriate learning rates.

**Limitations of SGD**

1.Sensitive to the learning rate: i.e. if learning rate is too large it may diverge. And if too small it Converges slowly.

2.High variance in updates can lead to instability in convergence.

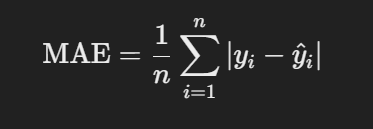
3.Requires careful tuning of hyperparameters.

4.This method is widely used in machine learning for training models where the dataset size is large.

**Evaluation Metrics for Linear Regression**

Linear regression models predict continuous numerical values, and their performance is evaluated using various metrics. Each metric provides unique insights, so selecting the right one depends on the context of your problem.

1. Mean Absolute Error (MAE) : Measures the average absolute difference between actual (yi) and predicted (y^i​) values.

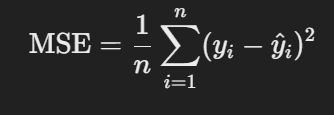


**When to Use**:

* When all errors are equally important.
* Easy to interpret in terms of the unit of the target variable.

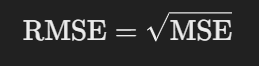
**Limitation**: Does not penalize large errors as heavily as other metrics like MSE

1. **Mean Squared Error (MSE) :** Measures the average squared difference between actual and predicted values. Penalizes larger errors more than MAE.



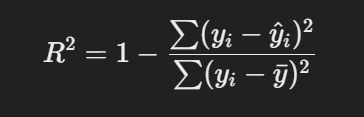
* When to Use:
  + When large errors are especially undesirable.
  + Suitable for applications like finance or engineering where small deviations matter less than large ones.
* Limitation: Sensitive to outliers because of the squaring of errors**.**

1. **Root Mean Squared Error (RMSE) :** Square root of MSE, providing error in the same units as the target variable.



* When to Use:
  + To directly compare prediction errors in the same scale as the target variable.
  + Similar to MSE but more interpretable because it's in the original unit of measurement.
* Limitation: Still sensitive to outliers.

1. **R-Squared () :** Measures the proportion of variance in the target variable explained by the model.

****

* + = 1: perfect prediction i.e. All changes in y can be completely explained by the changes in x.
  + = 0.83: 83% of the variation in y is explained by x, while the remaining 17% is due to other factors or randomness which model is unable to predict.
  + = 0 : The changes in y are not related to x at all, and the model's predictions are no better than simply using the average value of y.
* When to Use:
  + To assess how well the model explains variability in the data.
  + Commonly reported in regression tasks.
* Limitation: 
  + 1. **Assumes independence**: i.e. it assumes that all variables in a model are independent, which is not always true.
    2. **Can be misleading** : it can be misleading when working with small datasets or non-linear relationships.

**Overfitting vs Underfitting**

1. Overfitting: Model performs well on training data but poorly on unseen data. Happens when the model learns noise in the data.

Solution: Use regularization, cross-validation, or simplify the model.

1. Underfitting: Model fails to capture the underlying pattern in the dataHappens when the model is too simple.

Solution: Increase model complexity or improve feature engineering.

**Bias-Variance Tradeoff in Machine Learning**

The **bias-variance tradeoff** is a fundamental concept in machine learning that describes the balance between two sources of error (i.e. bias and variance) that affect model performance:

1. **Bias**: Bias is the error that occurs when the model is too simple and cannot capture the complexity of the data. **Example**: Imagine trying to fit a straight line (simple model) to data that forms a curve. The model is "biased" because it assumes a straight-line relationship that doesn’t exist.
2. **Variance**: Variance is the error that occurs when the model is too complex and tries to fit even the noise in the data. **Example**: Imagine fitting a wavy line (very complex model) that passes through every single data point. The model is "overreacting" to the specific details of the training data.

The goal is to minimize the total error, which is the sum of bias, variance, and irreducible error i.e.

**Total Error Formula**

Total Error=Bias^2+Variance+Irreducible Error where,

* **Bias** contributes to systematic error.
* **Variance** contributes to model instability.
* Irreducible Error is **:-** The noise inherent in the data that cannot be eliminated by any model.

Example: Measurement errors, random variations in the data.

So, In the **Bias-Variance Tradeoff t**he goal is to find a balance between bias and variance because finding the right balance ensures the model generalizes well to unseen data.

* **High Bias** → Model is too simple → Underfits the data.
* **High Variance** → Model is too complex → Overfits the data.
* **Just Right** → Model captures the patterns in the data without overcomplicating things.

**The Tradeoff**

* **High Bias, Low Variance**: Simple models (e.g., linear regression) tend to underfit.
* **Low Bias, High Variance**: Complex models (e.g., deep neural networks) tend to overfit.
* **Optimal Model**: Strikes a balance between bias and variance, minimizing the total error.

1. **How to Address the Tradeoff?**

**Reducing Bias (Underfitting)**

* Use a more complex model (e.g., polynomial regression instead of linear regression).
* Add more features to the dataset.
* Reduce regularization parameters (e.g., lower L1 or L2 penalty).

**Reducing Variance (Overfitting)**

* Use a simpler model (e.g., reduce the depth of a decision tree).
* Add regularization (e.g., Ridge or Lasso regression).
* Increase training data size.
* Use techniques like cross-validation.

**Real-Life Example**

* **Bias-Dominated Model**: Predicting house prices with only the number of bedrooms while ignoring other factors like location or size.
* **Variance-Dominated Model**: Using all features, including irrelevant ones like the color of the house or the type of garden plants, resulting in overfitting.

The bias-variance tradeoff helps guide the choice of model complexity and feature engineering. Striking the right balance is crucial for generalizing well on unseen data.

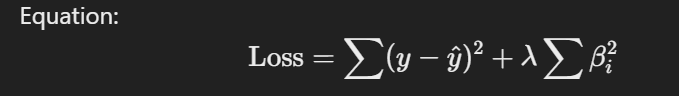
**Regularization** is a technique in machine learning used to reduce overfitting by adding a penalty term to the loss function. This penalty discourages the model from learning overly complex patterns, such as assigning large weights to certain features, which might result in poor generalization to unseen data.

**Regularization in Linear Regression**

These techniques are especially useful for high-dimensional datasets.

**1. Ridge Regression (L2 Regularization)**

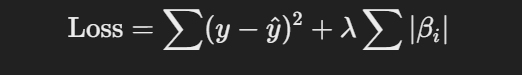
Adds the sum of squared coefficients as a penalty to the loss function. i.e.



Shrinks coefficients towards zero but does not eliminate them.

**2. Lasso Regression (L1 Regularization)**

Adds the sum of the absolute values of coefficients as a penalty.



* Shrinks some coefficients to zero, effectively performing feature selection.

Cross-Validation is a method to detect Overfitting.

Cross-validation splits the dataset into training and validation subsets multiple times to ensure the model generalizes well to unseen data.

Its Common Types are:

K-Fold Cross-Validation: Data is divided into 𝑘 folds; the model trains on k−1 folds and validates on the remaining fold.