# XGBoost
# A Deep Dive into the Most Powerful ML Algorithm

Sreeyakannamala

23094968

**Github:** https://github.com/Sreeya200/XGBoost-for-Machine-Learning.git

## What is XGBoost?

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

It's vital to an understanding of XGBoost to first grasp the machine learning concepts and algorithms that XGBoost builds upon: supervised machine learning, decision trees, ensemble learning, and gradient boosting.

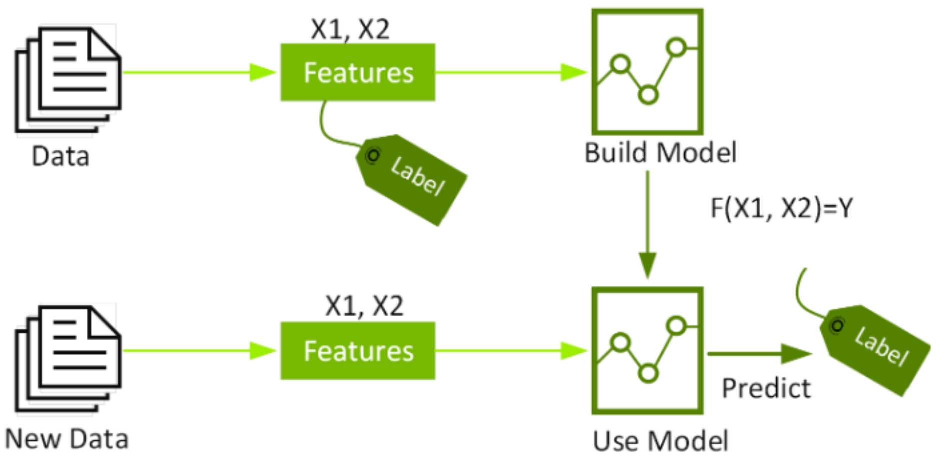
Supervised machine learning uses algorithms to train a model to find patterns in a dataset with labels and features and then uses the trained model to predict the labels on a new dataset's features.

## Introduction to XGBoost

XGBoost (Extreme Gradient Boosting) is an advanced machine learning algorithm based on gradient boosting. It is widely used in machine learning competitions like Kaggle and is preferred for structured/tabular data.

## Why XGBoost?

- **Superior Performance**: Often outperforms deep learning models for structured data.
- **Efficient Computation**: Highly optimized for both CPU and GPU.
- **Feature Importance**: Provides useful insights into which features impact predictions.
- **Handles Missing Values**: Inbuilt capability to manage missing data.
- **Regularization**: Prevents overfitting using L1 (Lasso) and L2 (Ridge) regularization.

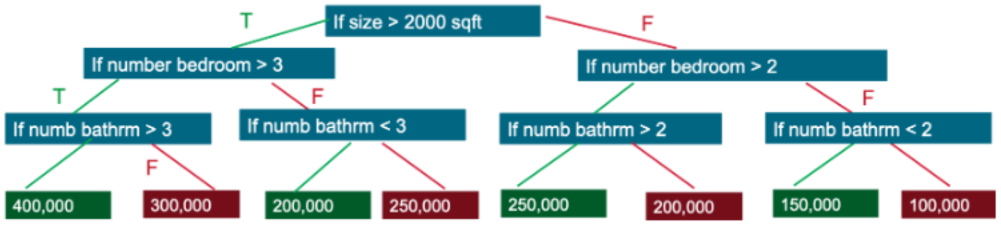


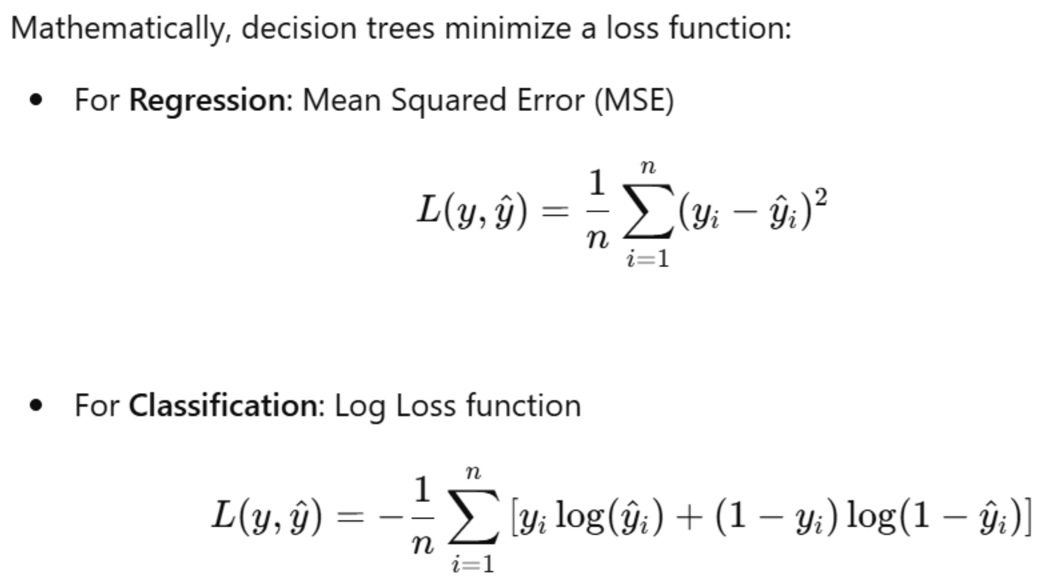
## Understanding Decision Trees:

Decision trees build a model that makes predictions by evaluating a series of true/false conditions on feature values. The goal is to determine the minimum number of such decisions needed to make an accurate prediction. They can be used for:

- **Classification**: Assigning a category label to an input.
- **Regression**: Predicting a continuous numerical value.

For example, in estimating house prices, a decision tree might consider factors like house size and number of bedrooms as features.





Mathematically, decision trees minimize a loss function:

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

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

- For **Classification**: Log Loss function

$$L(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

## Understanding Gradient Boosting

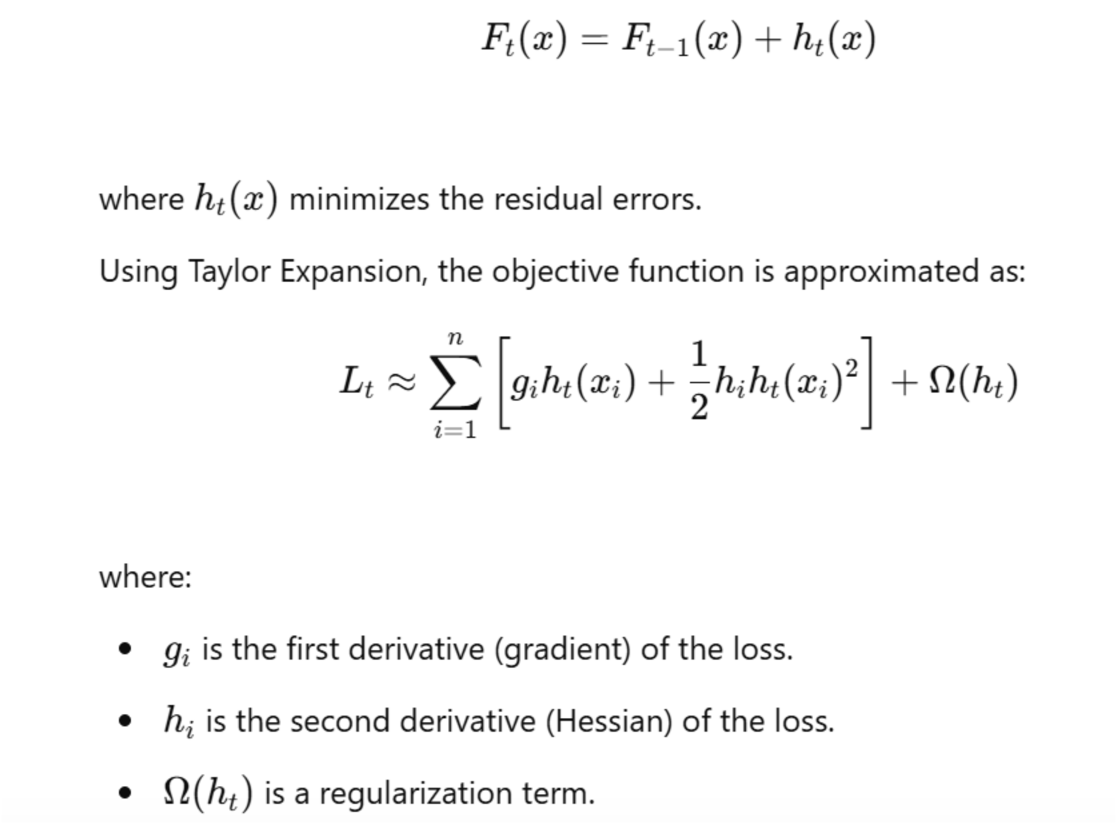
### What is Boosting?

Boosting is an ensemble technique that combines multiple weak learners (usually decision trees) to create a strong learner. It improves model performance iteratively.

### How Gradient Boosting Works

1. Train a weak model (e.g., a small decision tree).
2. Compute residuals (errors between actual and predicted values).
3. Train a new model to predict the residuals.
4. Combine models to reduce errors.
5. Repeat until performance stops improving.

Mathematically, the model at iteration is:



$$F_t(x) = F_{t-1}(x) + h_t(x)$$

where $h_t(x)$ minimizes the residual errors.

Using Taylor Expansion, the objective function is approximated as:

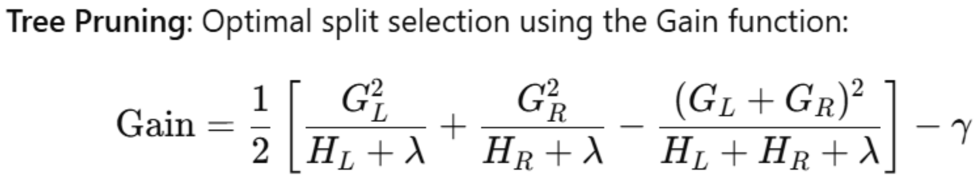
$$L_t \approx \sum_{i=1}^{n} \left[ g_i h_t(x_i) + \frac{1}{2} h_i h_t(x_i)^2 \right] + \Omega(h_t)$$

where:

- $g_i$ is the first derivative (gradient) of the loss.
- $h_i$ is the second derivative (Hessian) of the loss.
- $\Omega(h_t)$ is a regularization term.

**XGBoost Enhancements over Gradient Boosting**

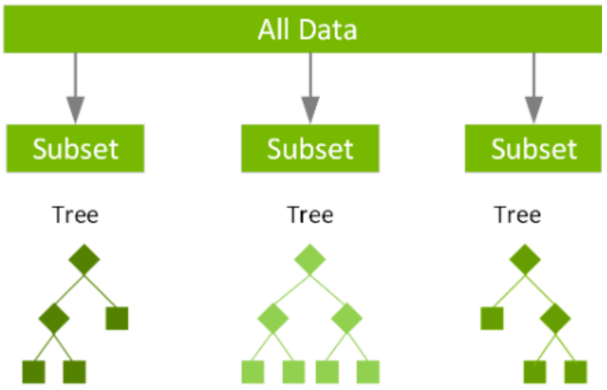
- **Regularization**: L1 (Lasso) and L2 (Ridge) to prevent overfitting.
- **Weighted Quantile Sketch**: Better handling of weighted data.
- **Parallel Processing**: Faster training with multi-core CPUs.
- **Tree Pruning**: Prevents unnecessary complexity in decision trees.



**Tree Pruning**: Optimal split selection using the Gain function:

$$\text{Gain} = \frac{1}{2} \left[ \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma$$

- **Handling Missing Values**: Learns optimal splits even with missing data



## Key Features of XGBoost

### 1. Regularization

- **L1 Regularization (Lasso)**: Shrinks some coefficients to zero, selecting the most important features.
- **L2 Regularization (Ridge)**: Prevents large coefficients, reducing overfitting.

### 2. Feature Importance

XGBoost provides built-in functions to identify the most influential features in your model.

### 3. Cross-Validation

K-fold cross-validation ensures robust model evaluation and helps avoid overfitting.

### 4. Hyperparameter Tuning

Key parameters:

- n_estimators: Number of trees.
- max_depth: Depth of each tree.
- learning_rate: Step size for boosting.
- subsample: Fraction of data used per iteration.
- colsample_bytree: Fraction of features used per tree.

### Comparing XGBoost vs Neural Networks

| Feature | XGBoost | Neural Networks |
|---|---|---|
| **Best for** | Structured Data | Unstructured Data (Images, Text) |
| **Training Speed** | Fast | Slow |
| **Hyperparameter Tuning** | Easier | Harder |
| **Interpretability** | High | Low |
| **Handles Missing Data** | Yes | No (Requires Imputation) |

***Note: Implementation is on the notebook attached to repository***

## Research work :

### 1. XGBoost in Gene Expression Prediction

A study by Wei Li et al. (2019) explored the use of XGBoost for predicting gene expression values. The research demonstrated how XGBoost outperforms traditional models in computational genomics by effectively handling high-dimensional biological data, improving accuracy, and reducing computation time.

- **Benefits of Using XGBoost in Genomics**:
    - Handles missing data efficiently.
    - Provides robust feature importance analysis.
    - Achieves high accuracy compared to other models.
    - Optimized computation for large-scale biological datasets.

### 2. Student Performance Prediction

Another study by Amal Asselman et al. (2021) utilized XGBoost to predict student performance. The research compared XGBoost with other ensemble models like Random Forest and AdaBoost, concluding that XGBoost significantly improved prediction accuracy in educational data analysis.

- **Benefits of Using XGBoost in Education**:
    - Enhances predictive accuracy of student performance.
    - Scalable and efficient for large datasets.
    - Outperforms traditional PFA algorithms.

### 3. Diabetes Prediction Using XGBoost

Mingqi Li et al. (2020) developed a diabetes prediction model leveraging the power of the XGBoost algorithm. The study aimed to improve prediction accuracy by efficiently handling both numerical and text-based features from medical datasets. The researchers implemented an optimized version of XGBoost that separated numerical variables while extracting key insights from textual features. Through extensive experimentation, their model achieved an accuracy of **80.2%**, demonstrating the robustness of XGBoost in medical data analysis. The findings suggest that XGBoost is not only effective in predictive analytics but also enhances early diagnosis and prevention strategies in healthcare.

### References

1. Wei Li et al. (2019). "Gene Expression Value Prediction Based on XGBoost Algorithm." *Frontiers in Genetics*. DOI: 10.3389/fgene.2019.01077
2. Amal Asselman et al. (2021). "Enhancing the Prediction of Student Performance Based on XGBoost." *Interactive Learning Environments*. DOI: 10.1080/10494820.2021.1928235
3. Mingqi Li et al. (2020). "Diabetes Prediction Based on XGBoost Algorithm." *IOP Conference Series: Materials Science and Engineering*. DOI: 10.1088/1757-899X/768/7/072093