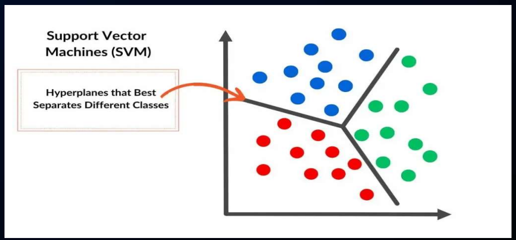
FROM WEAK TO STRONG: HARNESSING THE XGBOOST ADVANTAGE

BY JAYANA SARMA

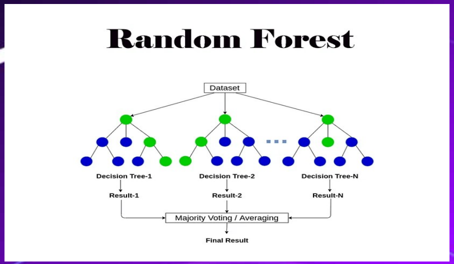
SVM V/S RANDOM FOREST

- A support vector machine (SVM) is a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane that maximizes the distance between each class in an N-dimensional space.



- Effective in high-dimensional spaces but sensitive to parameter tuning.

- A Random Forest is a machine learning algorithm that combines multiple decision trees to make predictions, essentially "voting" on the final result, making it robust and accurate for both classification and regression tasks.



- Robust, avoids overfitting, and works well with large datasets.

XGBOOST, AT A GLANCE!

- **Scalable Gradient Boosting Algorithm**: Improved version of gradient boosting.
- **Focus Areas**: Efficacy, computational speed, and model performance.
- **Open-Source Library**: Part of the Distributed Machine Learning Community.
- **Optimized Design**: Leverages both software and hardware capabilities.
- **Key Strengths**: Enhances boosting techniques for high accuracy in minimal time.

A QUICK FLASHBACK TO BOOSTING

- Boosting generally means increasing performance. In ML, Boosting is a sequential ensemble learning technique to convert a weak hypothesis or weak learners into strong learners to increase the accuracy of the model.
- For example,

  Imagine a class of students learning math:

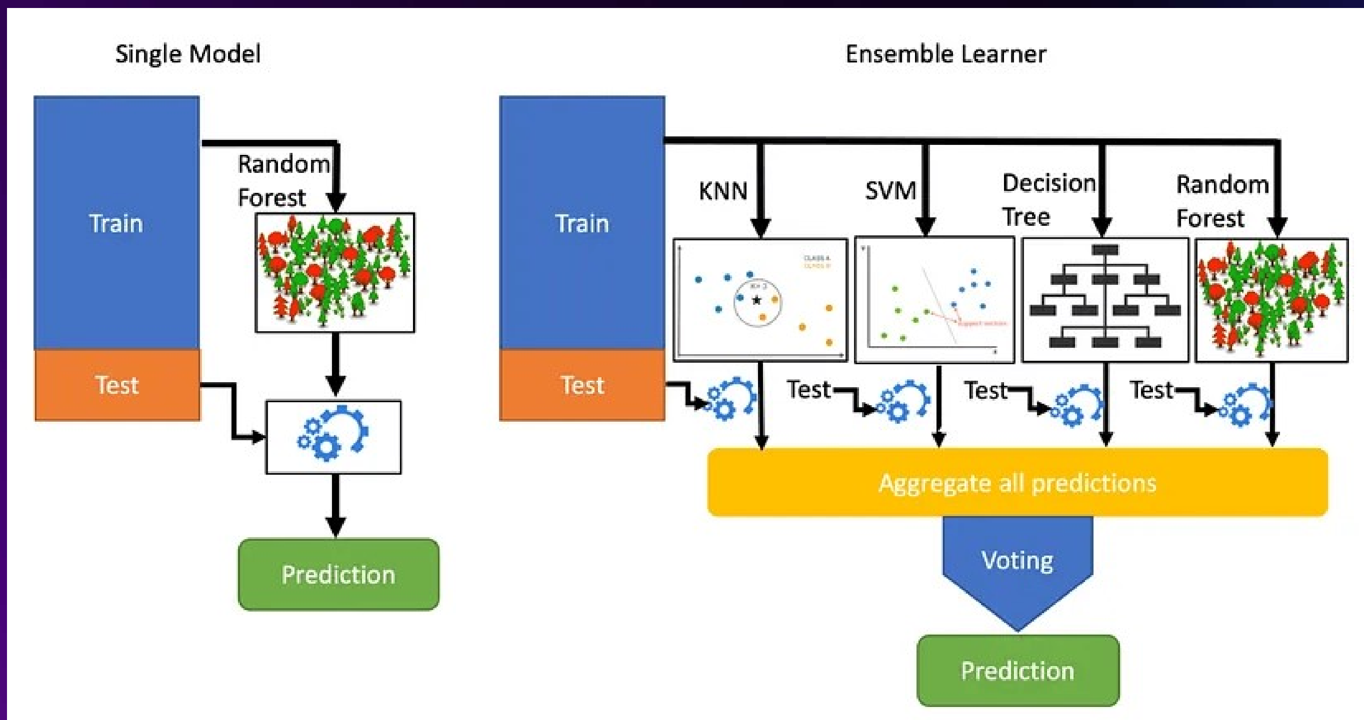
  1. The teacher starts with a simple concept(weak learner)
  2. Reviews mistakes and teaches a slightly advanced lesson to address those errors.
  3. Repeats this process until most students understand (strong learner).

  Similarly, boosting sequentially improves the model's "understanding" of data.

ENSEMBLE LEARNING

- Ensemble Learning combines decisions from multiple machine learning models to improve accuracy compared to using a single model.
- It reduces error by leveraging the strengths of multiple models.
- Maximum voting technique is commonly used for classification tasks, where the majority of votes determine the final prediction.

THIS IMAGE SHOWS A CLEAR DISTINCTION OF A SINGLE ML MODEL WITH RESPECT TO ENSEMBLE LEARNER:



WORKING OF BOOSTING ALGORITHM:

•**Boosting Algorithm Overview**:

•Combines multiple weak learners (models) to improve performance.
•Each new model is trained to correct the errors of the previous model.

•**Learning Process**:

•Misclassified samples receive higher weights, while correctly classified ones have lower weights.
•The final model places more emphasis on the stronger learners (models that perform better).

•**Greedy Nature**:

•Boosting is greedy because it focuses on correcting mistakes sequentially, without revisiting previous models.

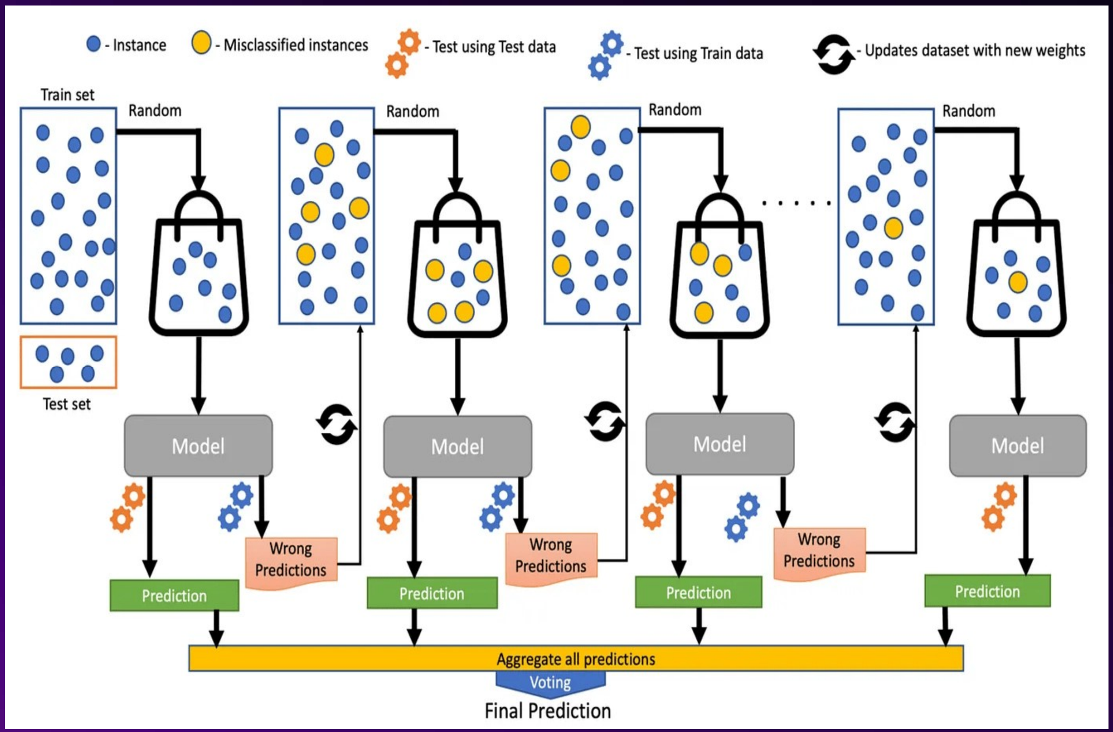
•**Overfitting Prevention**:

•It’s recommended to set a stopping criterion (e.g., early stopping or model performance) to avoid overfitting.

MATHEMATICAL NOTION

$$F_i(x) = F_{i-1}(x) + f_i(x)$$

CAPITAL **F**(I) IS CURRENT MODEL, **F**(I-1) IS PREVIOUS MODEL AND *SMALL* **F**(I) REPRESENTS A WEAK MODEL



Internal working of boosting algorithm

GRADIENT BOOSTING:

- Gradient Boosting Overview:

Special case of boosting that minimizes errors using the gradient descent algorithm.
Produces models composed of weak prediction learners (e.g., decision trees).

- Key Difference from Boosting:

Gradient Boosting updates weights using gradients of the loss function via gradient descent, optimizing errors iteratively.
Loss represents the difference between predicted and actual values.

- Loss Functions:

Regression problems: Use Mean Squared Error (MSE) as the loss function.
Classification problems: Use Logarithmic Loss as the evaluation metric.

Gradient Boosting Process:

Additive Modeling:

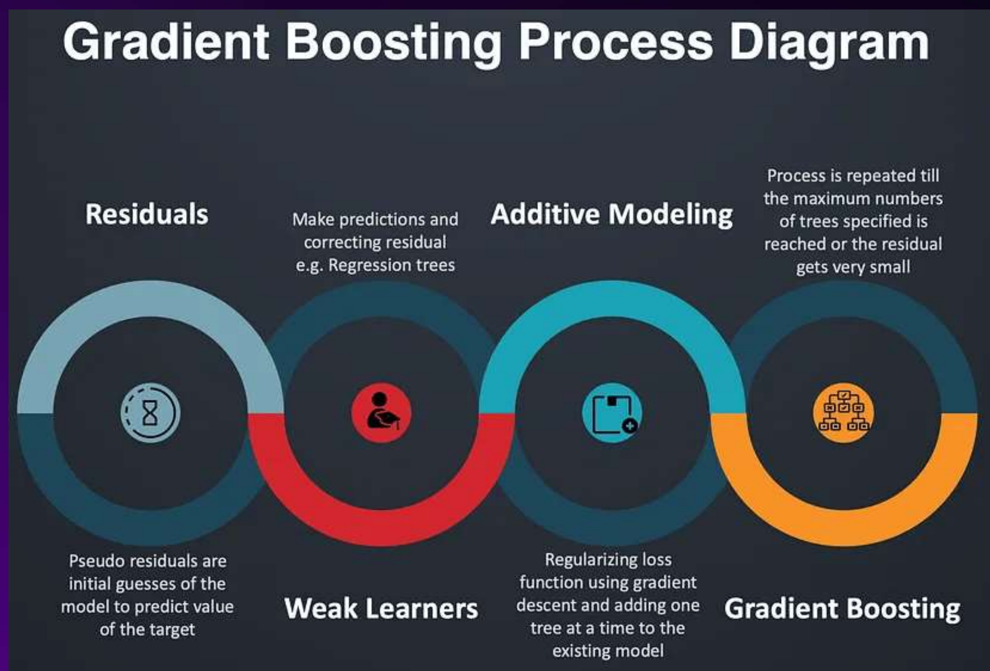
- Builds the model by sequentially adding new decision trees to minimize loss.
- Existing trees are left unchanged to reduce overfitting.

Stops when the loss falls below a specified threshold or a maximum number of trees is reached.

$$w = w - \eta \nabla w$$

$$\nabla w = \frac{\partial L}{\partial w} \text{ where } L \text{ is loss}$$

W REPRESENTS THE WEIGHT VECTOR, ETA IS THE LEARNING RATE



Process flow of Gradient Boosting

XGBOOST IN ACTION

- **Algorithm Enhancements:**
- **Tree Pruning**:
  - Reduces tree size to avoid overfitting.
  - Uses techniques like Cost Complexity or Weakest Link Pruning with MSE, k-fold cross-validation, and learning rate.
  - Prunes backward after reaching the specified max depth, keeping splits if the total loss remains positive.
- **Sparsity-Aware Split Finding**:
  - Handles missing or sparse data by assigning a default direction in trees.
  - Optimizes for sparse data by visiting only missing values, making it much faster (up to 50x).

System Enhancements:

•**Parallelization**:

- Speeds up tree learning by sorting data in compressed blocks and using all CPU cores/threads.
- Efficient for handling frequent node creation.

•**Cache Awareness**:

- Stores gradient statistics in thread-specific buffers, reducing time for read/write operations.
- Optimized block sizes (generally 2^{16}) minimize cache misses.

FLEXIBILITY IN XGBOOST:

- **Customized Objective Function** — An objective function intends to maximize or minimize something. In ML, we try to minimize the objective function which is a combination of the loss function and regularization term. Optimizing the loss function encourages predictive models whereas optimizing regularization leads to smaller variance and makes prediction stable.
  Examples: reg: linear(Regression)
  Binary: logistic(binary classification)
  Multi : softmax(multiclass classification)

- **Customized Evaluation Metric** — This is a metric used to monitor the model's accuracy on validation data.
  ● *rmse* — Root mean squared error (Regression)
  ● *mae* — Mean absolute error (Regression)
  ● *error* — Binary classification error (Classification)
  ● *logloss* — Negative log-likelihood (Classification)
  ● *auc* — Area under the curve (Classification)

RESOURCES:

- https://medium.com/sfu-cspmp/xgboost-a-deep-dive-into-boosting-f06c9c41349
- https://xgboost.readthedocs.io/en/stable/
- https://medium.com/@jyotsna.a.choudhary/mastering-xgboost-a-technical-guide-for-intermediate-machine-learning-practitioners-f7ad167c6865

THANK YOU