AI CLUB

Summer school session

ENSEMBLE METHODS

Table of CONTENTS

01 What are ensemble methodfs

02 Why ensemble methods

03 How many different ensemble methods

04 Ensemble techniques

05 Bagging - Random forests

06 Boosting - Adaboost, gradient boosting, XGboosting

07 Evaluating the performance of ensemble methods

WHAT ARE ENSEMBLE METHODS?

Ensemble methods in machine learning are techniques that combine multiple models to improve the overall predictive performance. These methods work by training a collection of individual models, called base learners, and then combining their outputs to make a final prediction. The idea behind ensemble methods is that the combined model will outperform any of the individual base learners, leading to more accurate and robust predictions.

Take an example,

Say you are a music artist and are planning to release your first album on Lo-fi genre and you wanted to get a general opinion on it before the official release.

WHY ENSEMBLE METHODS?

Improved Accuracy

Ensemble methods combine multiple individual models to create a more powerful and accurate prediction. By leveraging the strengths of different algorithms, ensemble models can outperform any single model, leading to significantly improved performance on complex tasks.

Robustness to Noise

Real-world data is often noisy and contains outliers. Ensemble methods are more robust to these challenges, as they can cancel out the errors made by individual models. This makes them more reliable and stable in the face of data imperfections.

Reduced Overfitting

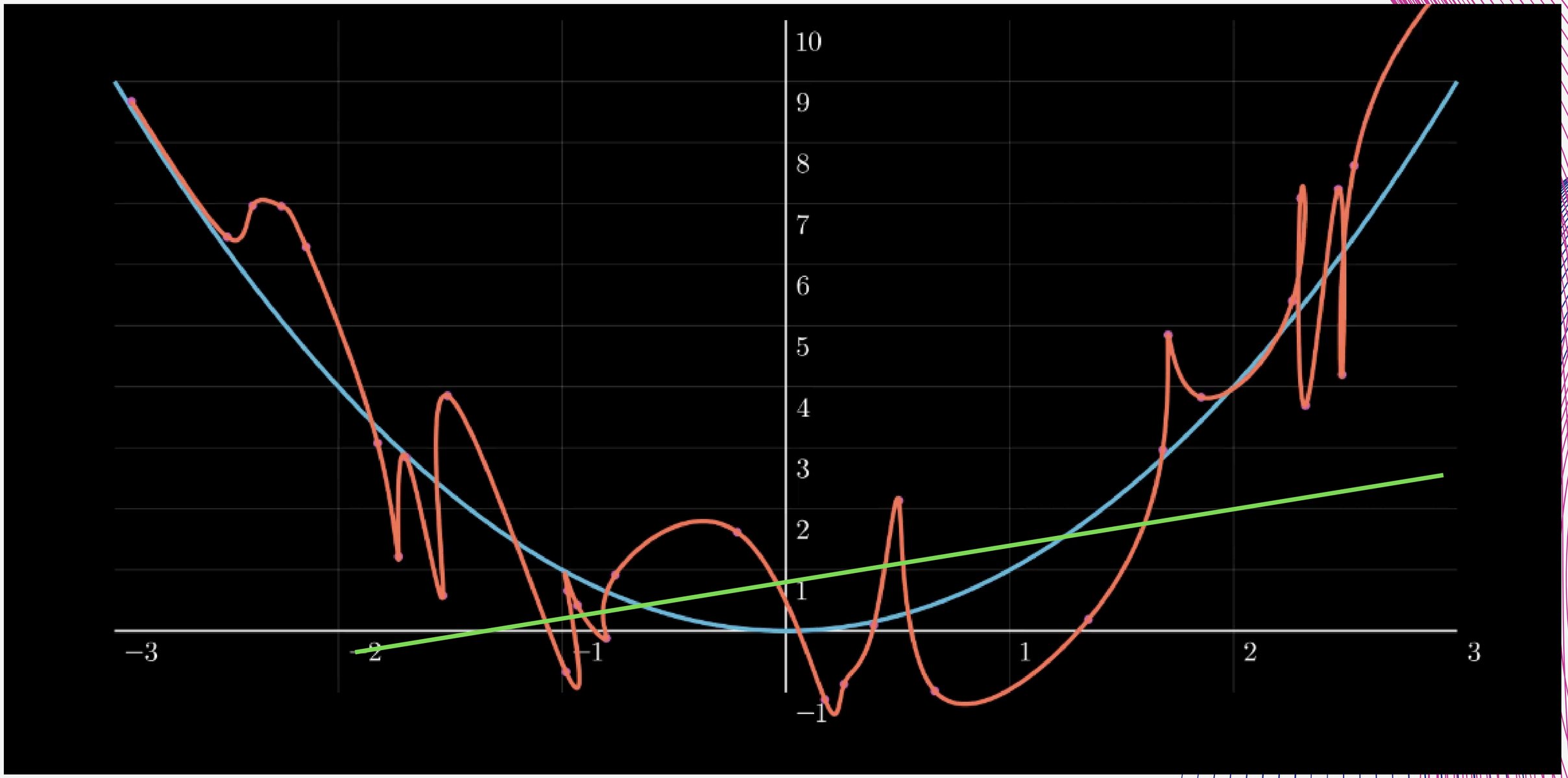
By combining diverse models, ensemble methods can reduce the risk of overfitting to the training data. This results in better generalization performance, allowing the models to perform well on unseen, real-world data.

Increased Flexibility

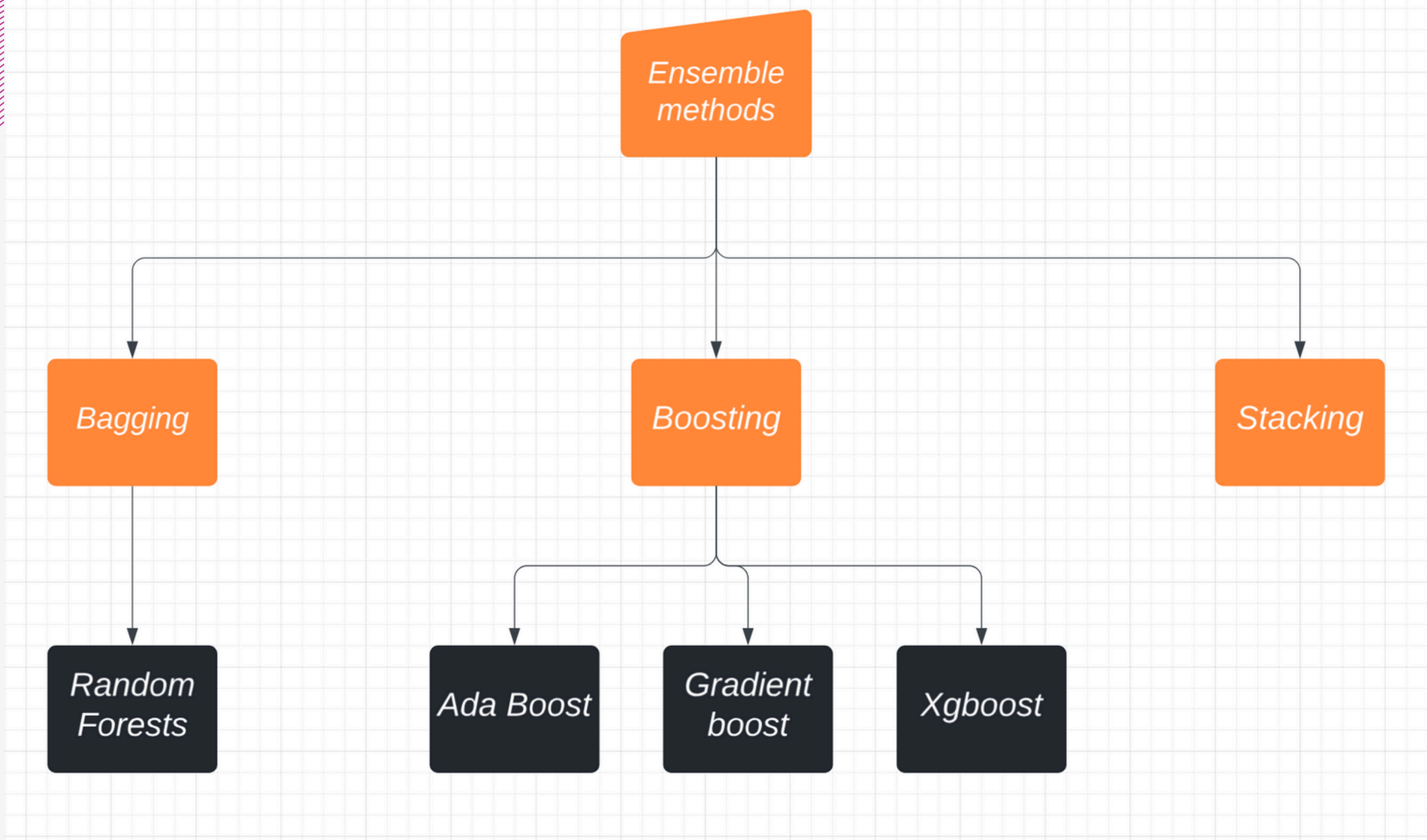
Ensemble methods offer greater flexibility in model selection and configuration. This allows data scientists to explore a wider range of algorithms and techniques, ultimately leading to more effective and customized solutions for their specific problems.

High variance - Overfitting

Low variance - Underfitting

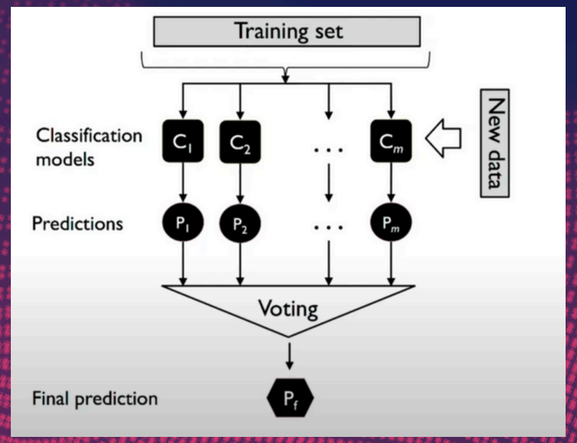


HOW MANY DIFFERENT ENSEMBLE METHODS

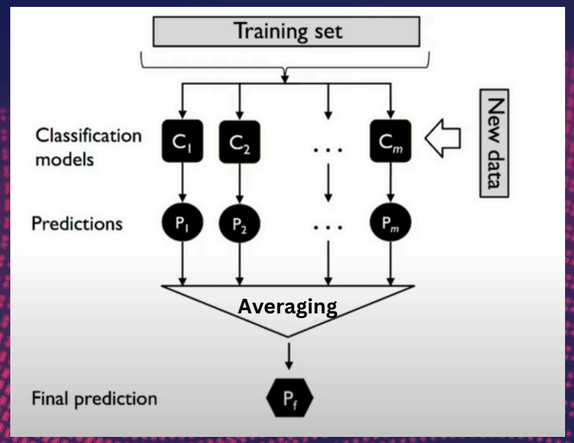


Some simple ensemble techniques

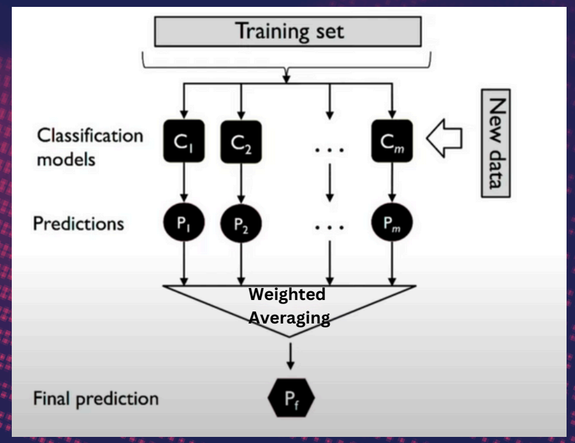
- Before diving into the ensemble methods, lets first gain the basic idea about the ensemble techniques used to implement the aforementioned ensemble methods



Voting



Averaging

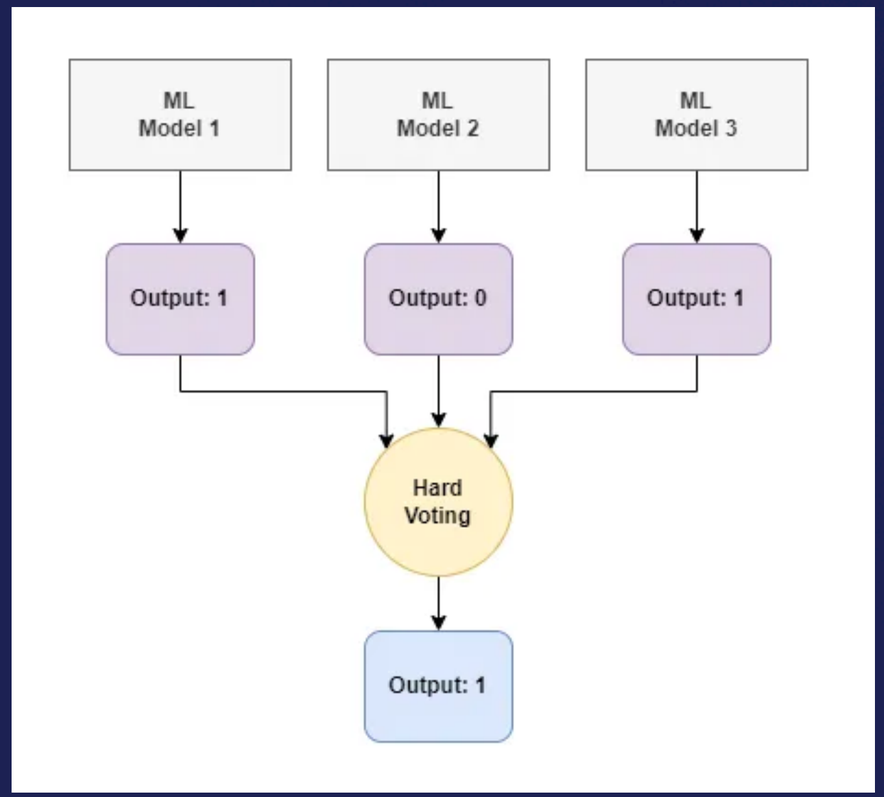


Weighted averaging

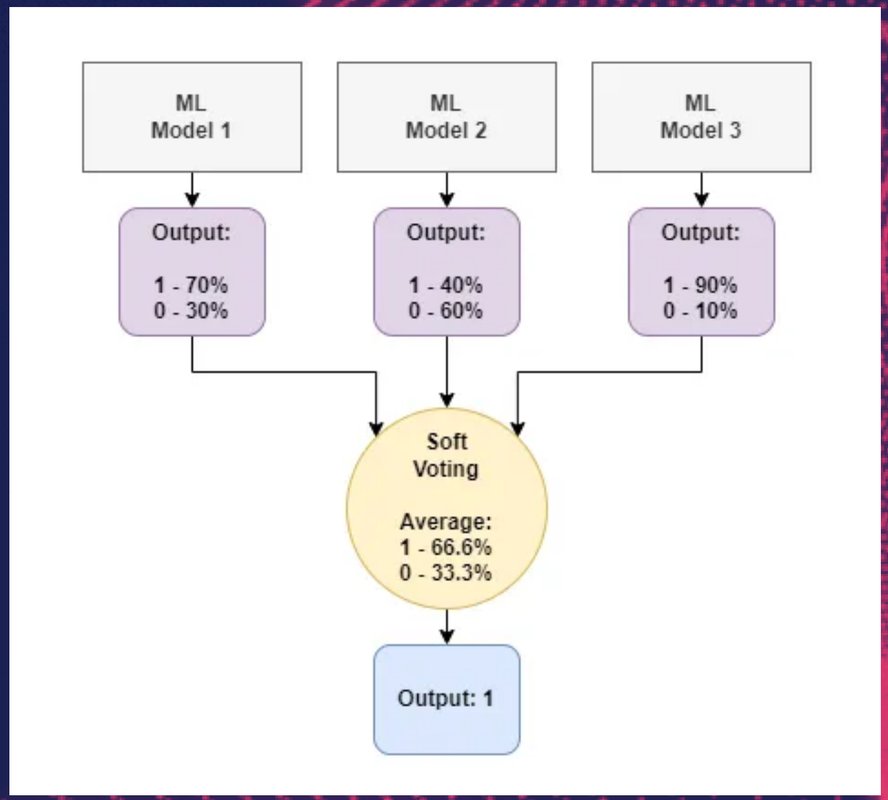
	A	B	C	D
1	Activity	Grade	Weight	
2	Homework	91	10%	
3	Quizzes	65	15%	
4	Assignments	80	20%	
5	Tests	73	25%	
6	Final exam	68	30%	
7				
8	Normal average		75.4	
9	Weighted average		73.5	

- If the data set is continuous or not
- If the data set has models among which certain models are more reliable or accurate

Types of voting

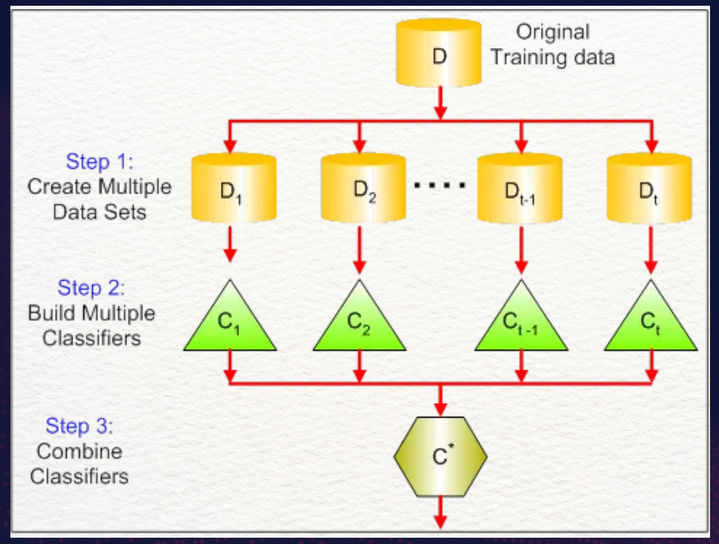


Hard voting



Soft voting

BAGGING - BOOTSTRAP AGGREGATION



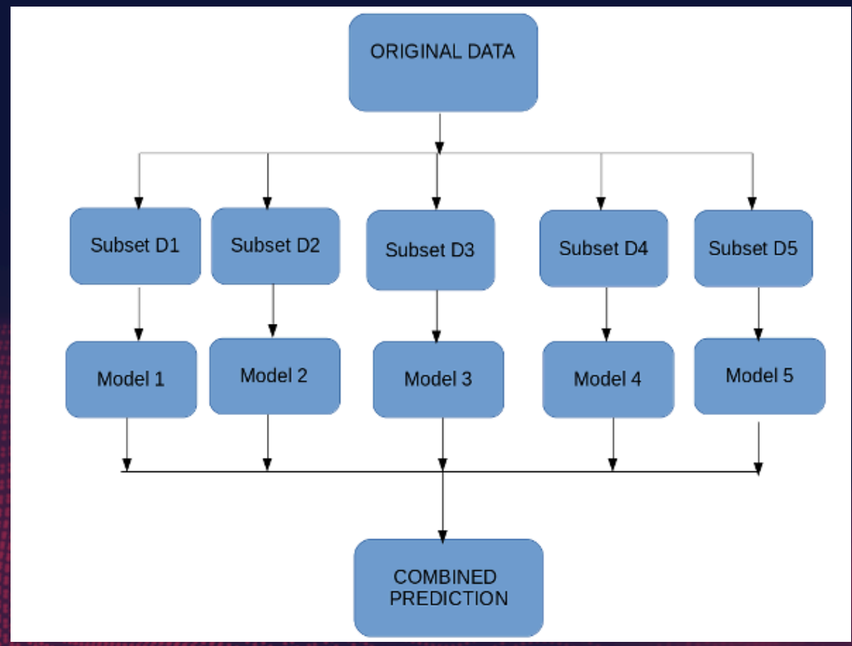
- The idea behind bagging is combining the results of multiple models (for instance, all decision trees) to get a generalized result, often through voting (soft or hard).
- Bagging is split into “Bootstrapping” + “Aggregating” = “Bagging”
- Several individual weak learners are combined together to create a strong learner.
- One such example of a bagging method is random forests.

Bootstrapping :

- Several number of subsets are taken from the data set with replacement and are assigned to different models.

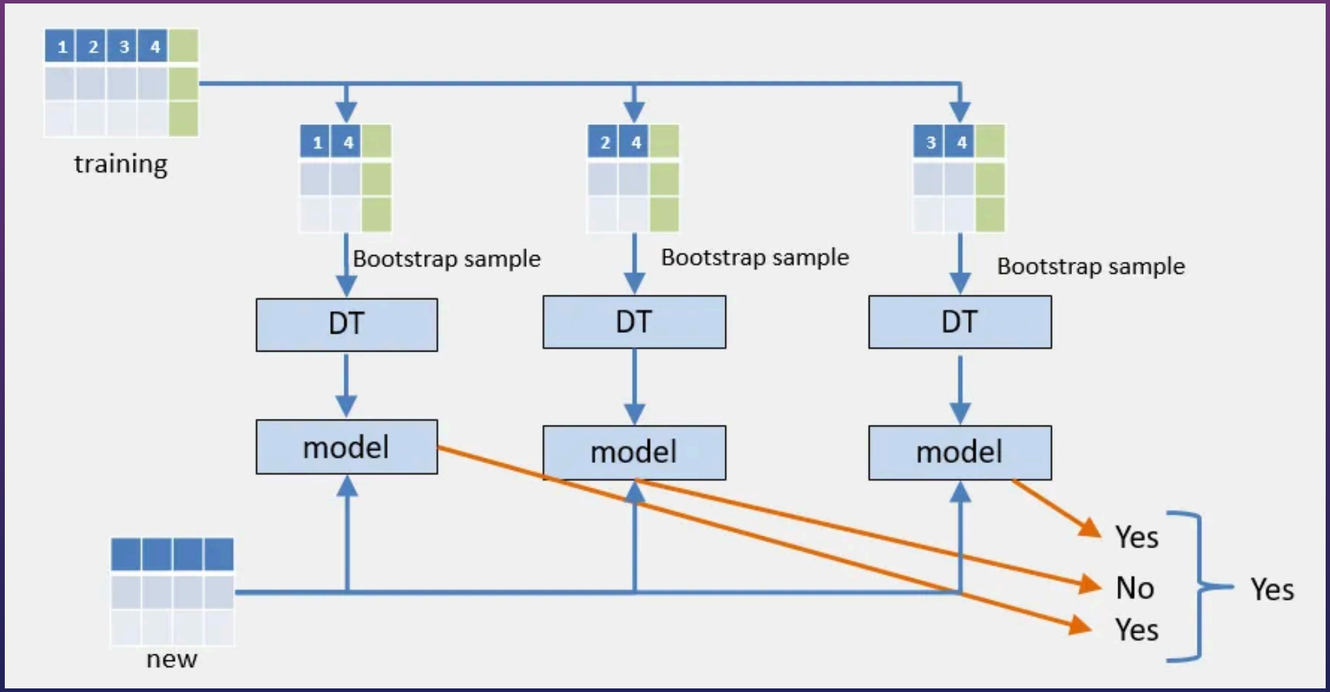
Aggregating :

- Combining the outputs from each subset into a final output, often voting technique is used



RANDOM FORESTS

- Random forests are just decision trees which undergo bagging viz,. the classification models in bagging method replaced with decision trees
- Now, by nature decision trees have a high variance, which means they are prone to overfitting.
- This is mitigated by operating them with bagging so as to introduce averaging among the decision trees and thereby reducing the variance

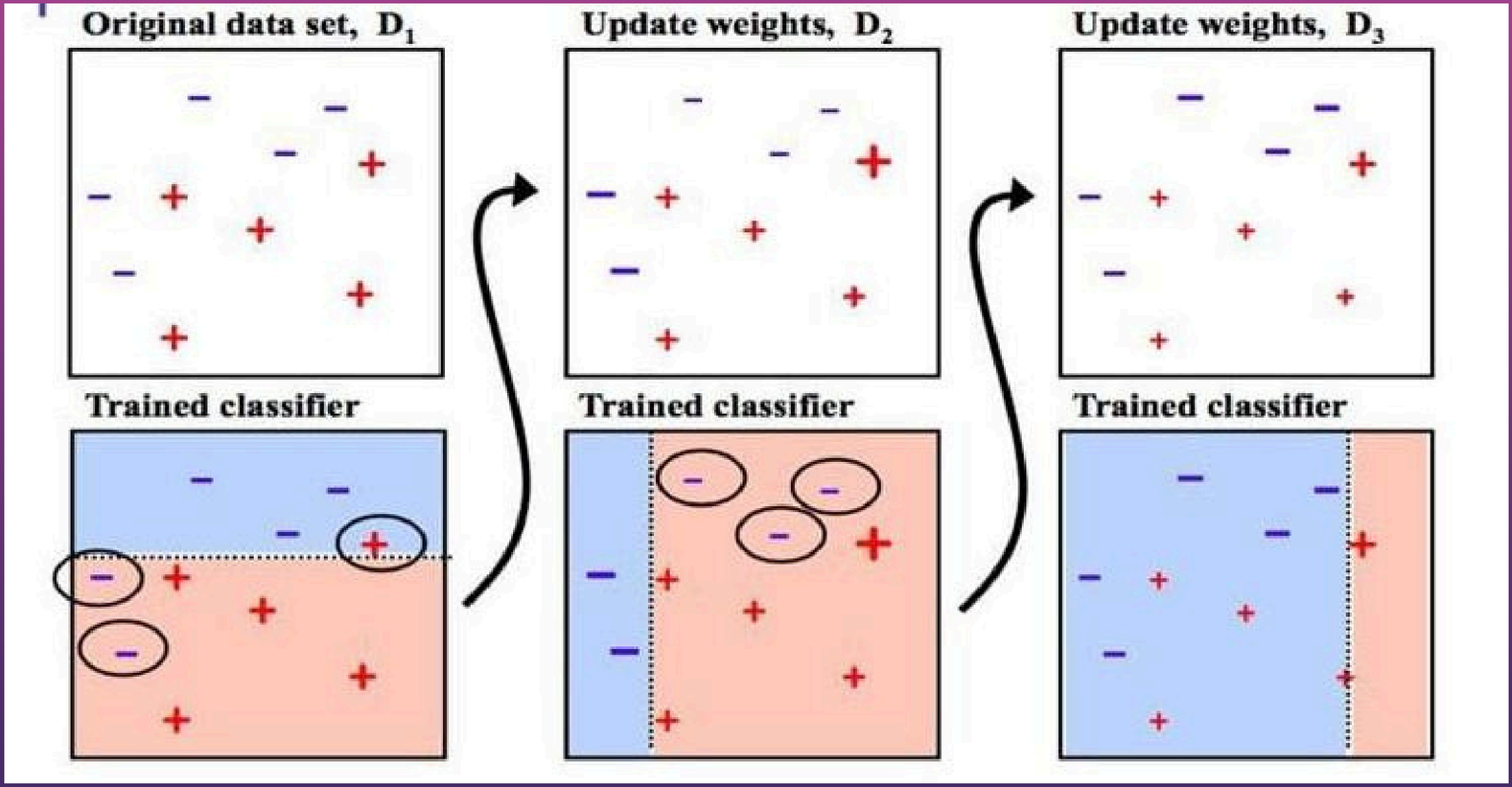


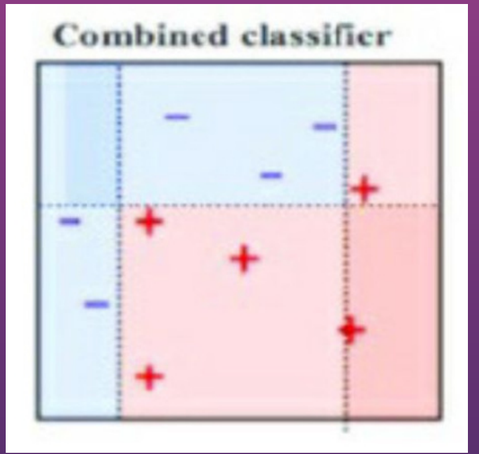
- Random feature selection - Some features are repeated and some do not appear at all
- This makes sure that the model is diverse and the individual trees are less correlated, which ultimately improves the predictions.
- This also helps in filtering out undesirable elements like noise and outliers.

Slido Quiz

If a data point is incorrectly predicted by the first model, and then the next (probably all models); will combining the predictions provide better results?

BOOSTING

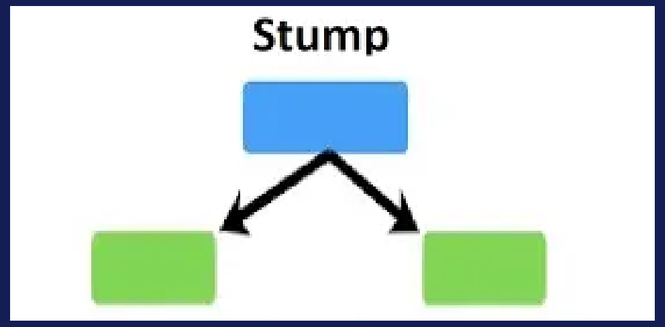




- Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model.
- Here, the ensemble technique used is weighted mean (specific to this classification model.

ADA BOOST

- Ada boost - Adaptive boosting
- Ada boost is a when we replace the classification models with stumps
- Stumps - Decision trees with a depth of 1 ie., one node and “2” leaves

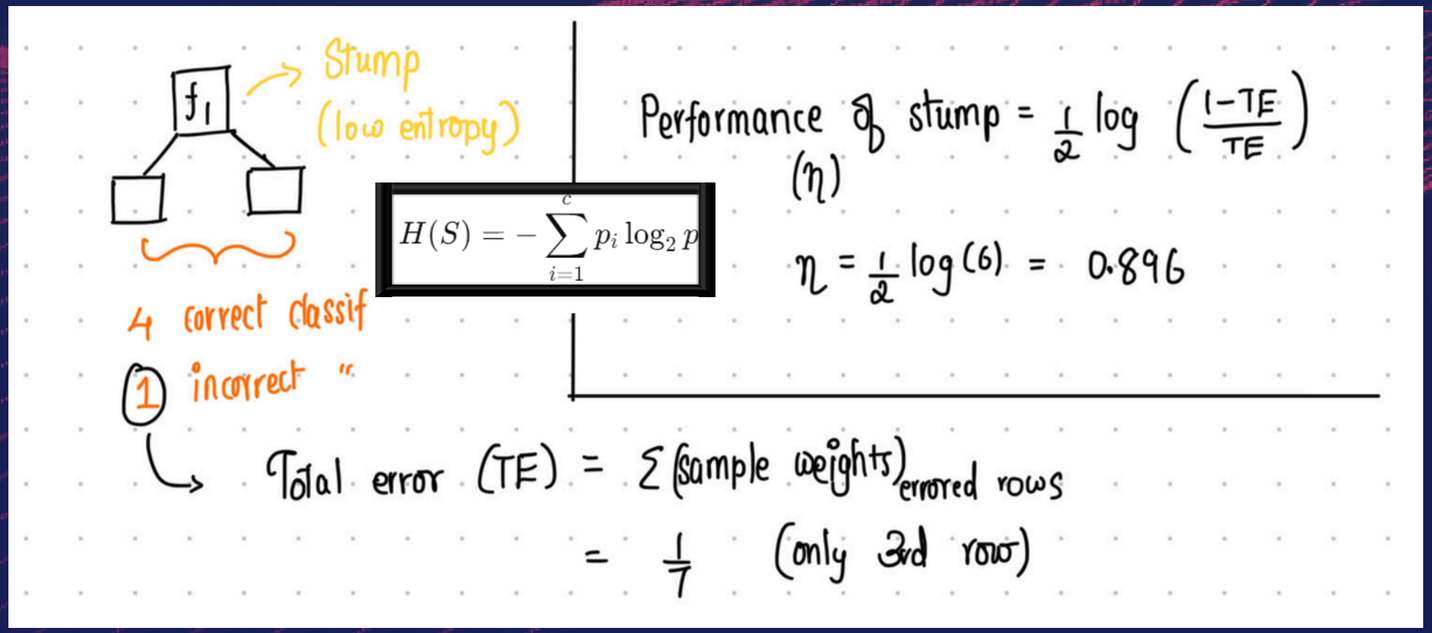


Mathematical functioning :

	f_1	f_2	f_3	Output	Sample weight $= \frac{1}{n}$	Updated weight	Normalised updated weights	Cumulative class
1				yes	1/7	0.06	0.086	0 → 0.086
2				No	1/7	0.06	0.086	0.086 → 0.172
3				yes	1/7	0.349	0.5	0.172 → 0.672
4				No	1/7	0.06	0.086	0.672 → 0.758
5				yes	1/7	0.06	0.086	0.758 → 0.844
6				yes	1/7	0.06	0.086	0.844 → 0.93
7				No	1/7	0.06	0.086	0.93 → 1

Class interval = 0.5 ↓ high probability of next picked up data to fall in this class.

Step 1 & 2- choosing stump & calc of error



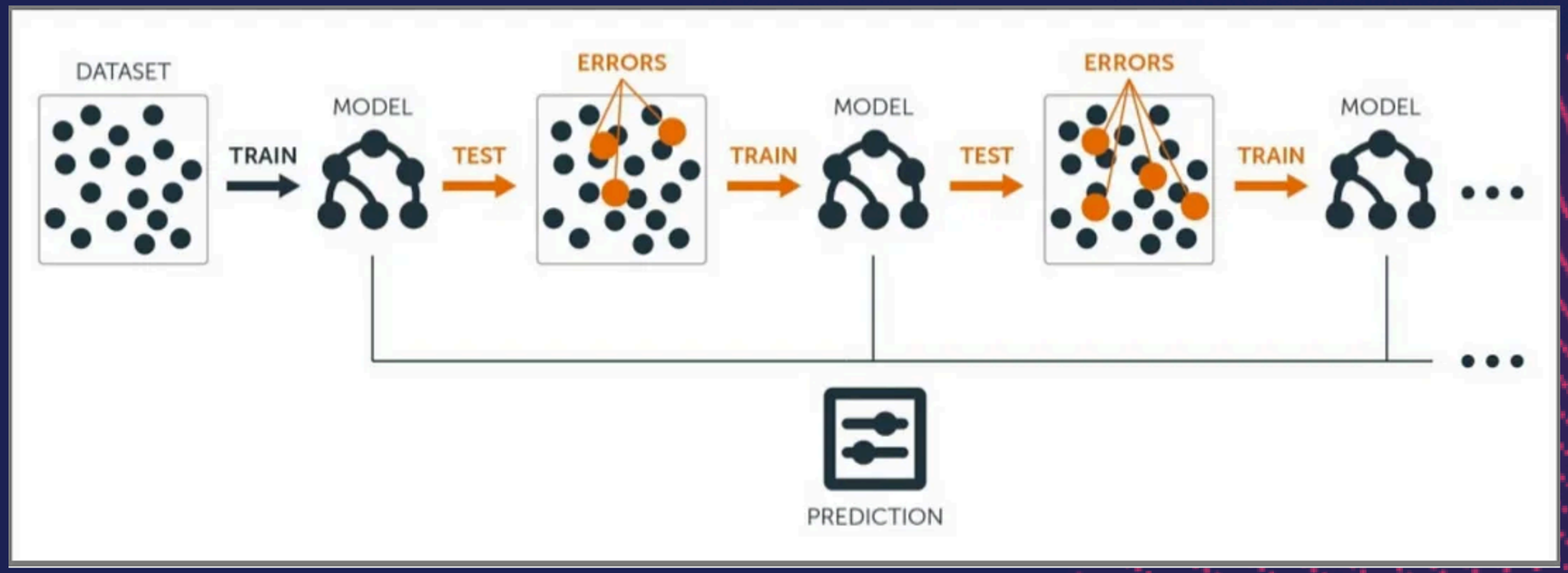
- A stump characteristic to a feature (f1 here) is chosen based on comparing the entropy of all 3 features.
- Total error is calculated.
- Performance of the stump is calculated.

Step 3- Updation of weights

- Update the sample weights with the help of calculated performance.
- Normalise the weights

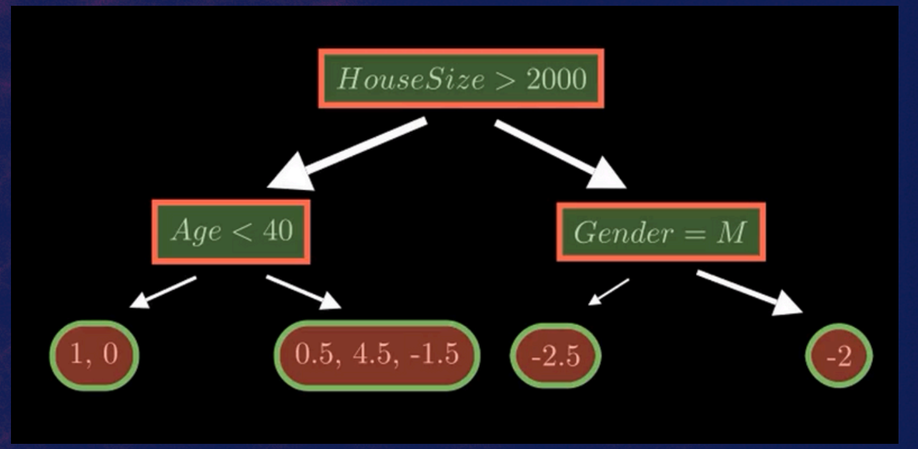
GRADIENT BOOSTING

- Supervised machine learning algorithm
- Final prediction is the summation of the initial model and all the predictions made by the weak learners scaled by learning rate.
- Here the weak learners are decision trees.
- Errors (residuals) are minimised by focusing on the gradient of the loss function.

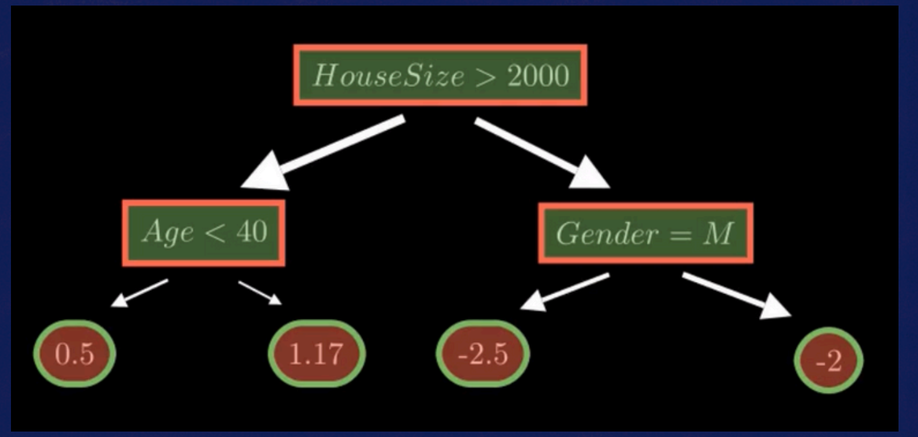


Mathematical implementation :

Gender	Age	House Size	Income	Prediction	Residuals
Male	30	3000	4	3	1
Female	35	5000	3	3	0
Female	41	2000	1	3	-2
Male	25	1000	0.5	3	-2.5
Female	56	2500	3.5	3	0.5
Male	58	6000	7.5	3	4.5
Male	46	2400	1.5	3	-1.5



Creating a decision tree



- Initial prediction = Average of the income
- Let's set the hyperparameter α = 0.1
- Residual = Actual - Predicted (note that the ones in orange colour coded column are predicted residuals)

Updating the predictions :

$$NewPrediction = OldPrediction + (\alpha \times PredictedResidual)$$

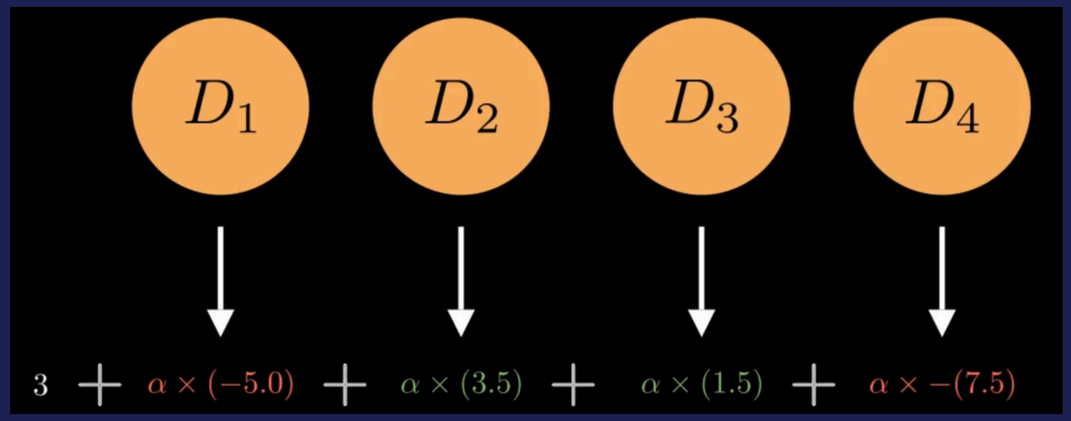
Gender	Age	House Size	Income	Prediction	Residuals
Male	30	3000	4	3.05	0.95
Female	35	5000	3	3.05	0.25
Female	41	2000	1	2.8	-1.8
Male	25	1000	0.5	2.75	-2.25
Female	56	2500	3.5	3.117	0.383
Male	58	6000	7.5	3.117	4.383
Male	46	2400	1.5	3.117	-1.617

Calculating the final prediction for a test data :

Gender	*Age*	*HouseSize*	*Income*	*Prediction*
Male	36	2500	2.5	?

Say the model has to predict the annual income of a person from the test data set given beside

- Say we have 4 decision trees as the classification models, which mean that there will be 4 iterations

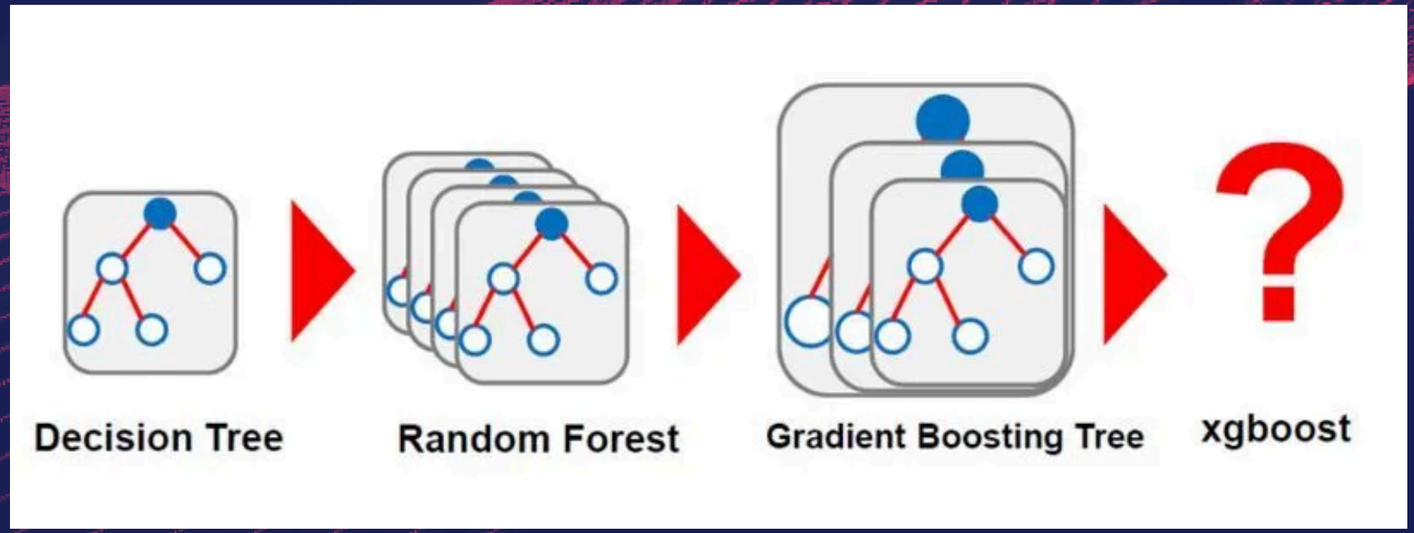


→

2.25

Slido Quiz

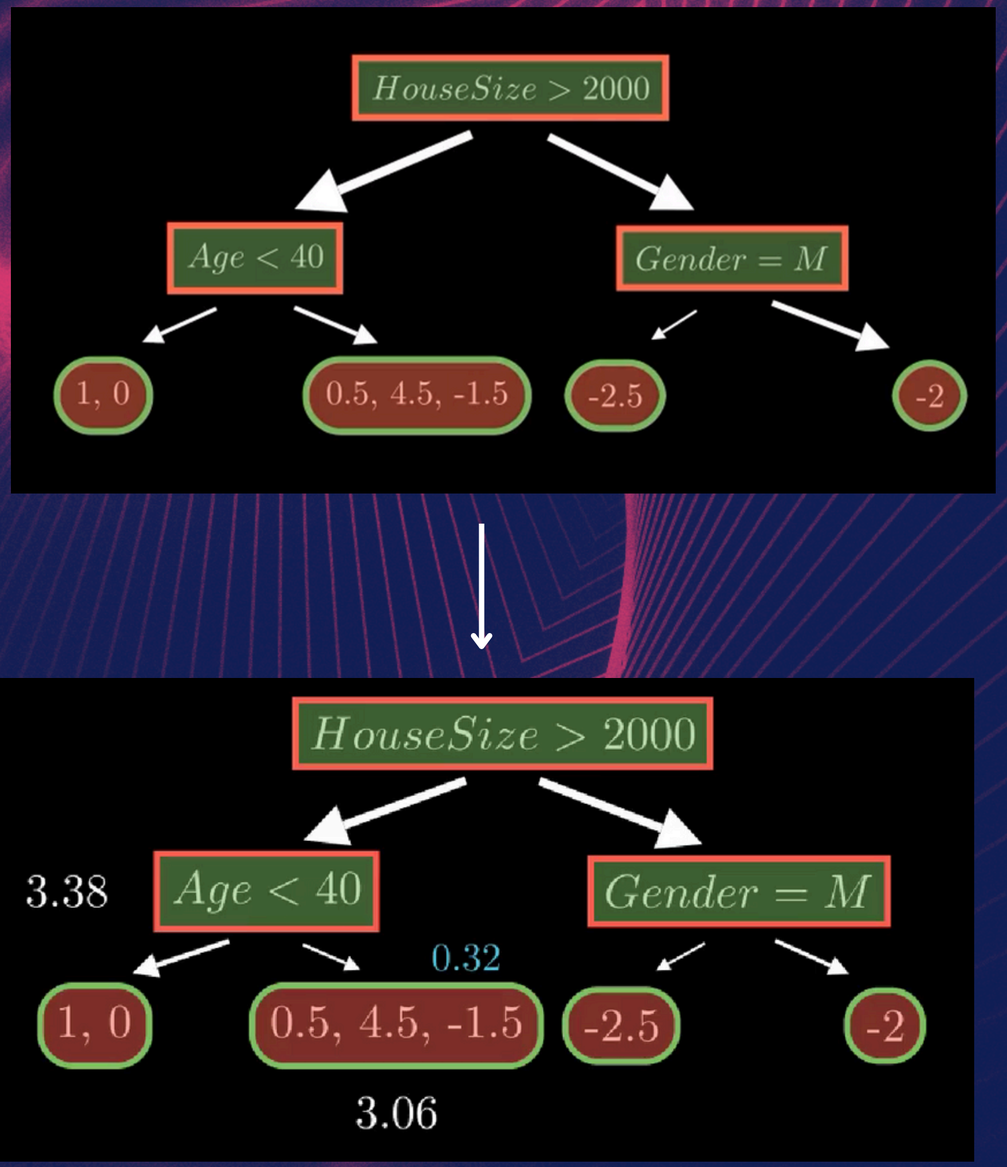
XG BOOSTING



- XGboost - Extreme gradient boosting
- More efficient and scalable than normal gradient boosting
- It’s faster and optimised owing to the fact that the computation done through this method can be distributed among various systems, cpus or cores.
- Salient features include the advanced tree pruning and ability to parallelize the computation.

- Tree pruning - removing the parts of the decision tree which does not contribute significantly towards the performance of the model; just like how excess of the branches of a tree are pruned .
- Regularisation - Simply put, regularisation here is a penalty imposed upon the model for its complexity. It ultimately reduces the risks of overfitting. Some commonly used regularisation methods in xgboost are - ***L1 (lasso regression) and L2 (ridge regression)***

Let us take the same example we have taken for gradient boosting :



- Here a new parameter called Similarity score ***(S)*** is introduced.
- And the pruning parameter (γ) which decides whether or not should the decision tree be pruned.
- λ - regularisation parameter

$$\text{Similarity Score} = \frac{Square of Sum of Residuals}{No. of Residuals + \lambda}$$

$$\text{Information Gain} = S_{parent} - S_{left}$$

- Pruning should be done when the Information gain is lesser than the pruning parameter :

$$\gamma > I.G.$$

EVALUATION OF PERFORMANCE

Metrics

Evaluating the performance of ensemble models requires careful consideration of various metrics, such as accuracy, precision, recall, F1-score, and AUC-ROC (Area Under the Receiver Operating Characteristic Curve). These metrics provide insights into the model's ability to make accurate predictions and its overall effectiveness.

Benchmarking

It's important to benchmark the ensemble model's performance against individual models and other ensemble techniques. This helps assess the added value that the ensemble brings and identify the most effective approaches for the problem at hand.

Cross-Validation

Cross-validation techniques, such as k-fold cross-validation, can provide a robust estimate of the ensemble's generalization performance. This helps ensure that the model is not overfitting to the training data and can perform well on new, unseen data.

NOTE - These are the ***ways*** to evaluate an ensemble method’s performance

CODE IMPLEMENTATION

Attendance QR