Introduction To Bagging and Boosting

DATA FOLKZ

CATAPULT DATA LEADERS

Bagging and Boosting

Bagging and Boosting are ensemble techniques in machine learning .
But what are **ensemble techniques ?**

ENSEMBLE LEARNING
Ensemble methods combine several classifiers to produce one model for better predictive performance.

In technical words, Ensemble learning is a machine learning paradigm where multiple models (often called “weak learners”) are trained to solve the same problem and combined to get better results.

DATA FOLKZ®
#CATAPULT DATA LEADERS

Bagging and Boosting

These methods are designed to improve the stability and the accuracy of Machine Learning algorithms. Combinations of multiple classifiers produce a more reliable classification than a single classifier.

To use Bagging or Boosting you must select a base learner algorithm. For example, if we choose a classification tree, Bagging and Boosting would consist of a pool of trees.

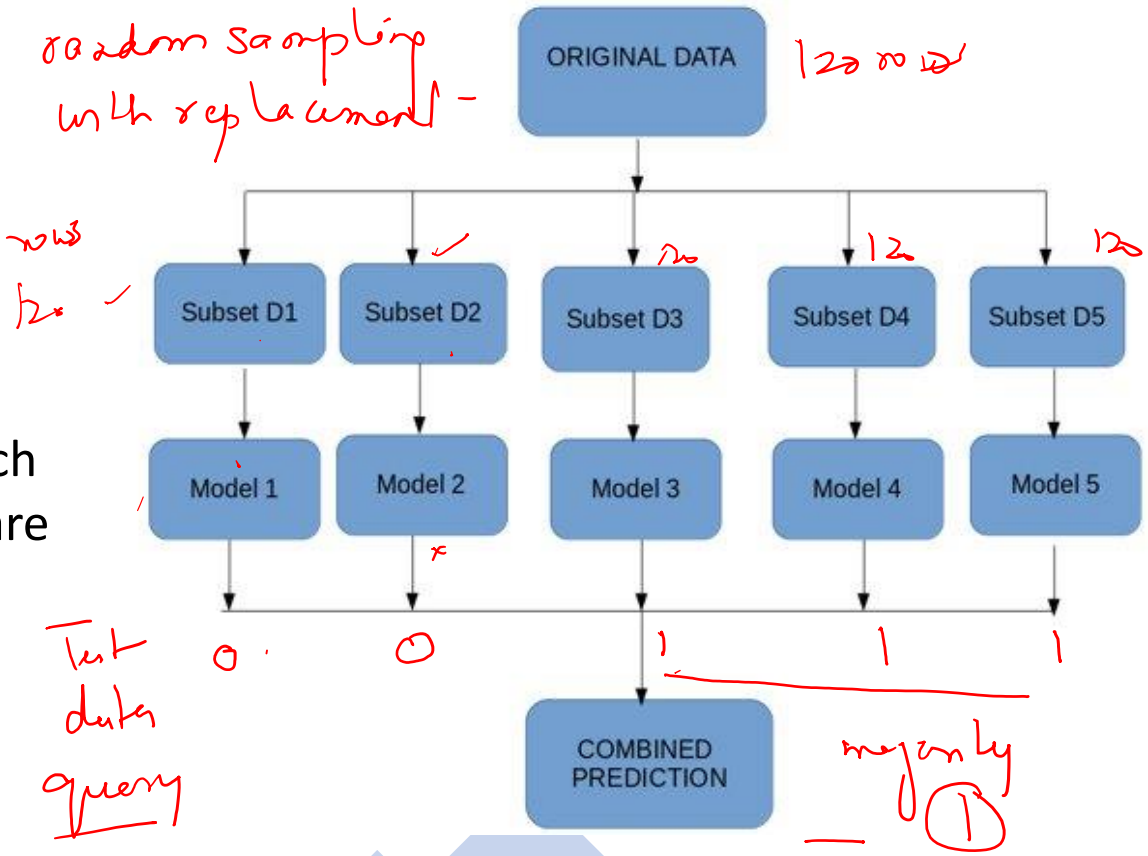
DATA FOLKZ®
#CATAPULT DATA LEADERS

Bagging

In Bagging ,
Firstly , Multiple subsets are generated from the original dataset, with replacement.

Then , A base model (weak model) is created on each of these subsets. These models run in parallel and are independent of each other.

The final predictions are determined by combining the predictions from all the models.



How Bagging works ?

Our dataset has 1490 datapoints .

- In the first step of bagging multiple subsets are created with replacement from the original dataset . **Suppose from our dataset , 120 subsets are created** .
- Then a base learner i.e a machine learning algorithm will be applied to all these 120 subsets . **Suppose decision tree is applied to all the subsets.**
- Then all the models will run parallelly and the **final predictions will be made using the predictions of all the 120 models .**

In our classification problem , the final predictions will be made by using majority voting , the most frequent class out of Yes and No will be selected .

- For regression problems , the final predictions are made by taking the average of all the values predicted by all the trees .

Bootstrap Aggregation (Bagging)

Bagging is also known as Bootstrap Aggregation

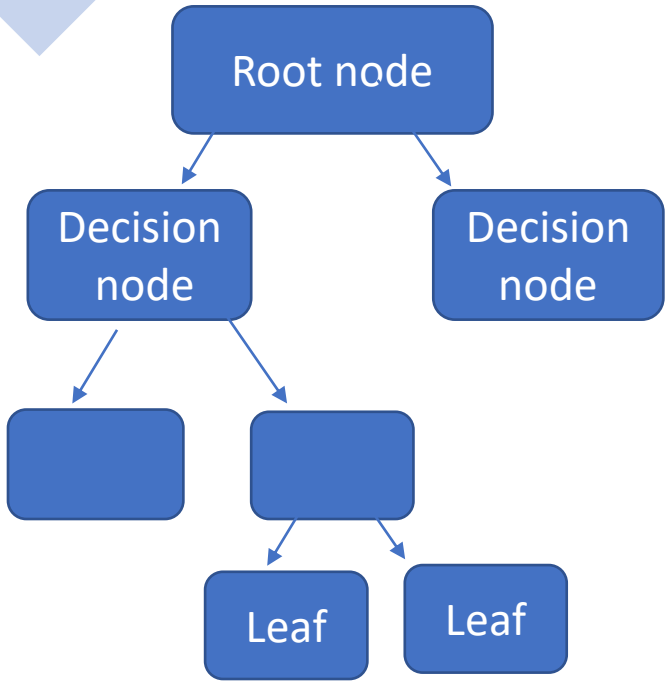
Bootstrapping : Bootstrapping is a sampling technique. Out of the n samples available, k samples are chosen **with replacement**.

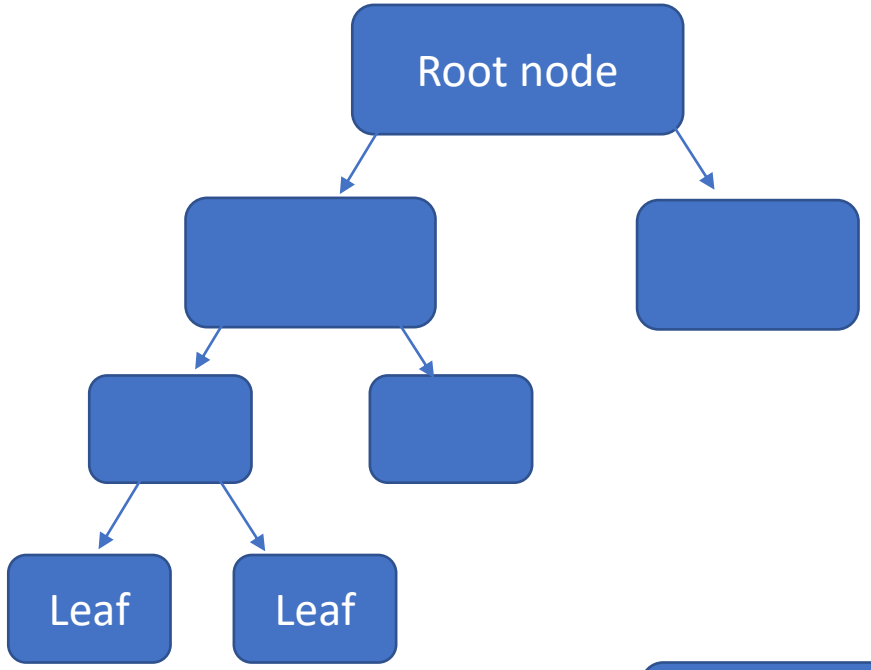
Aggregation : The predictions from the all the models are aggregated to make a final combined prediction. This aggregation can be done on the basis of predictions made or the probability of the predictions made by the bootstrapped individual models.

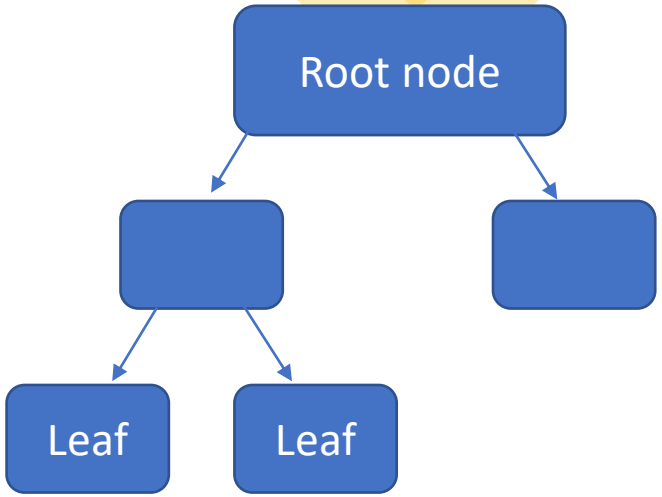
Bootstrapping the data plus using **agg**regate to make the decisions is **BAGGING**

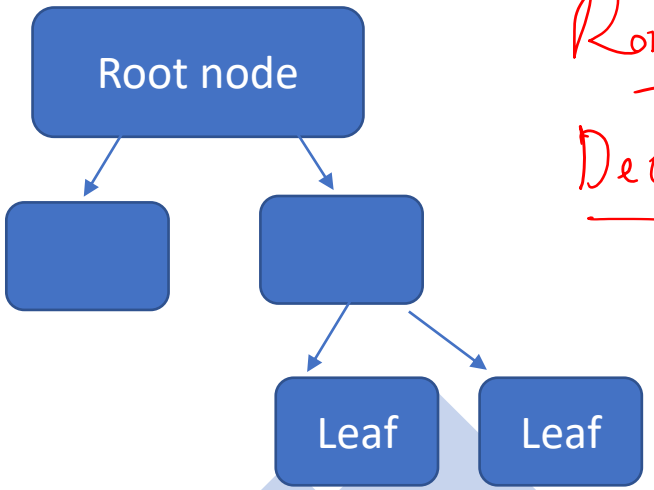
DATA FOLKZ®
#CATAPULT DATA LEADERS

Random Forest









120 trees

- 100

RandomForest

DecisionTree Classifier

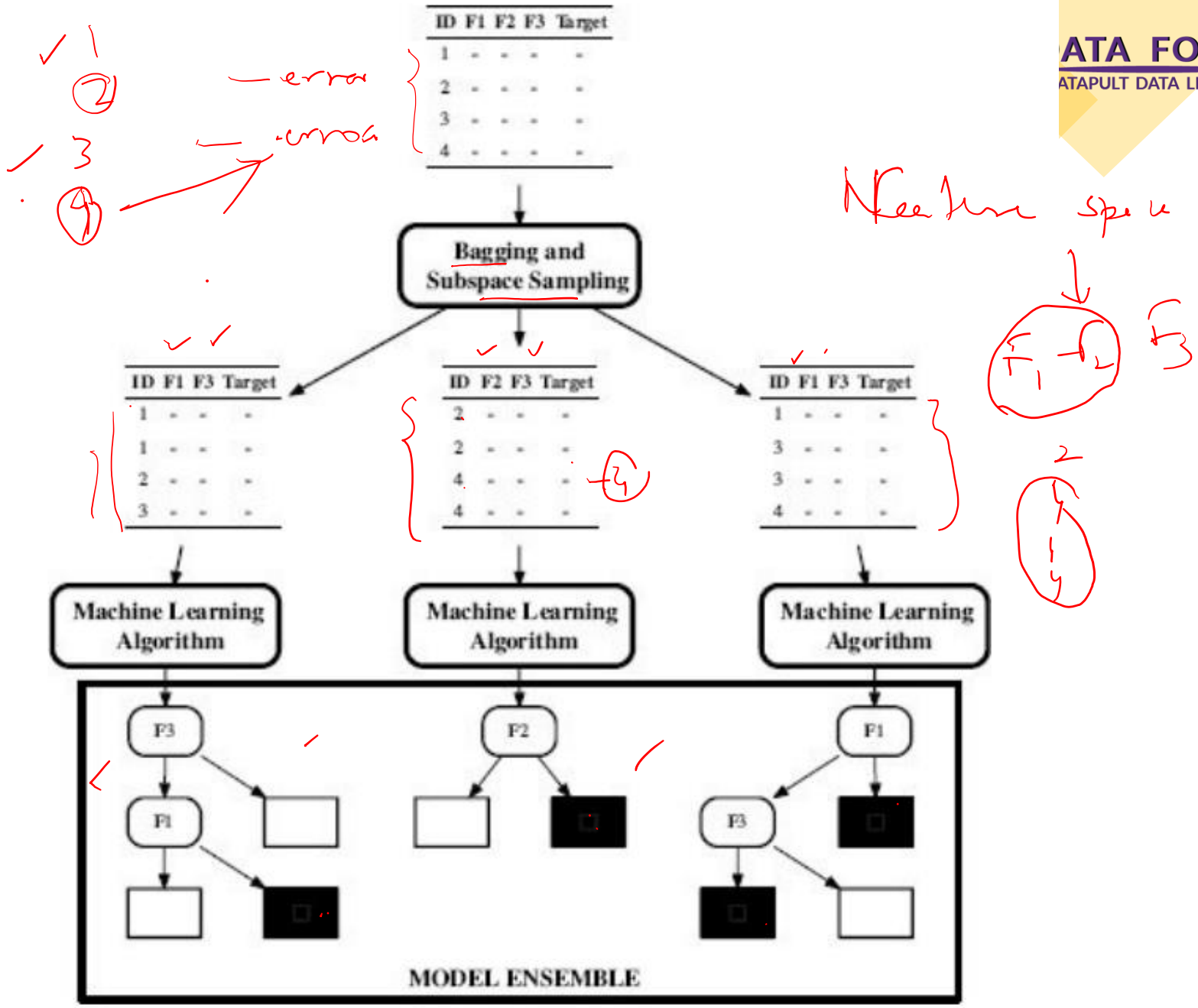
Random Forest- Bagging algorithm

- Random forests are built from Decision trees , random forest is simply a collection of decision trees whose results are aggregated into one final result.
- Random forest algorithm solves the problem of Decision trees of overfitting .
- They are much more stable than Decision tree and is not biased .

DATA FOLKZ®
#CATAPULT DATA LEADERS

Random forest

Random Forest: It is a bagging-based algorithm with a key difference wherein only a subset of features is selected at random



How Random forest classifies a new data point ?

DATA FOLKZ®
#CATAPULT DATA LEADERS

New Data point for prediction

Random forest model

Models	Prediction of Attrition
Decision Tree 1	No
Decision Tree 2	Yes
Decision Tree 3	No
Decision Tree 4	No
Decision Tree 5	Yes
.	
.	
.	
Decision Tree n	No

Will the person attrite ?

YES	NO
2	4

Final Predicition :
The new person will not leave the organisation

Parameters of Random Forests

n_estimators: The number of trees in the forest. Default is 100. This is the number of trees you want to build before taking the maximum voting or averages of predictions. Higher number of trees gives you better performance but makes your code slower. You should choose as high value as your processor can handle because this makes your predictions stronger and more stable.

Criterion: It is the function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “entropy” for the information gain

DATA FOLKZ®
#CATAPULT DATA LEADERS

Parameters of Random Forests

max_depth: The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. While creating Decision Tress for its subset, each decision tree is created to its complete depth by default.

min_samples_leaf: The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression.

DATA FOLKZ®
#CATAPULT DATA LEADERS

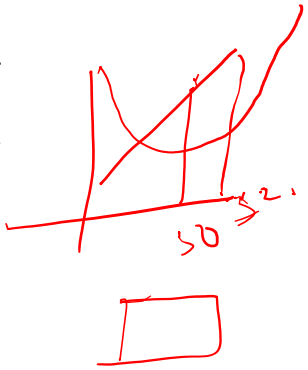
Advantages of Random Forests

- Random forests are popularly applied to both data science competitions and practical problems.
- They are often accurate, do not require feature scaling, categorical feature encoding, and need little parameter tunning.
- They can also be more interpretable than other complex models such as neural networks.
- It has methods for balancing errors in data sets where classes are imbalanced.

Disadvantages of Random Forests

- It surely does a good job at classification but not as good as for regression problem as it does not give precise continuous nature predictions.

In case of regression, it doesn't predict beyond the range in the training data, and that they may over-fit data sets that are particularly noisy.



- Random Forest can feel like a black box approach for statistical modelers – you have very little control on what the model does. You can at best – try different parameters and random seeds.

Boosting

DATA FOLKZ®
#CATAPULT DATA LEADERS

Boosting works a little differently from Bagging .

Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model. The succeeding models are dependent on the previous model.

ensemble models → Sequential model

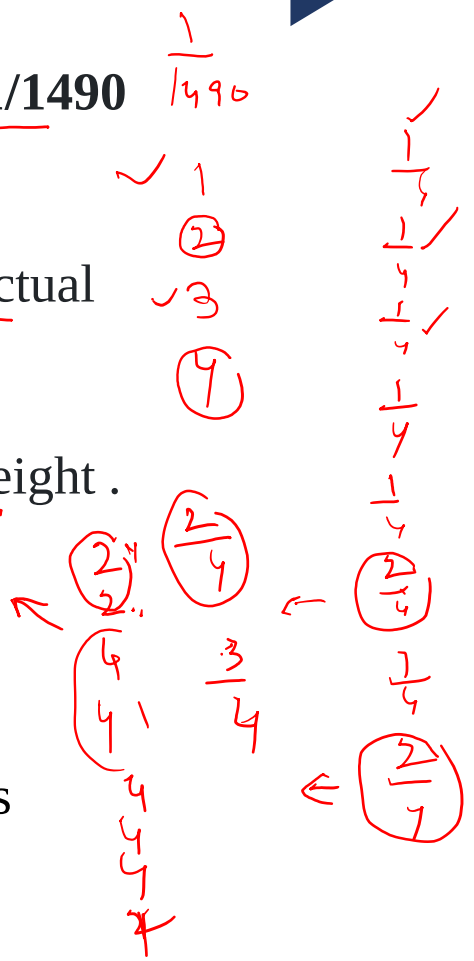
How Boosting works ?

Boosting works in the following steps :

1. All the data points of the dataset are given equal weights in the first step .
As our dataset have 1490 datapoint , each will be given a weight suppose , 1/1490

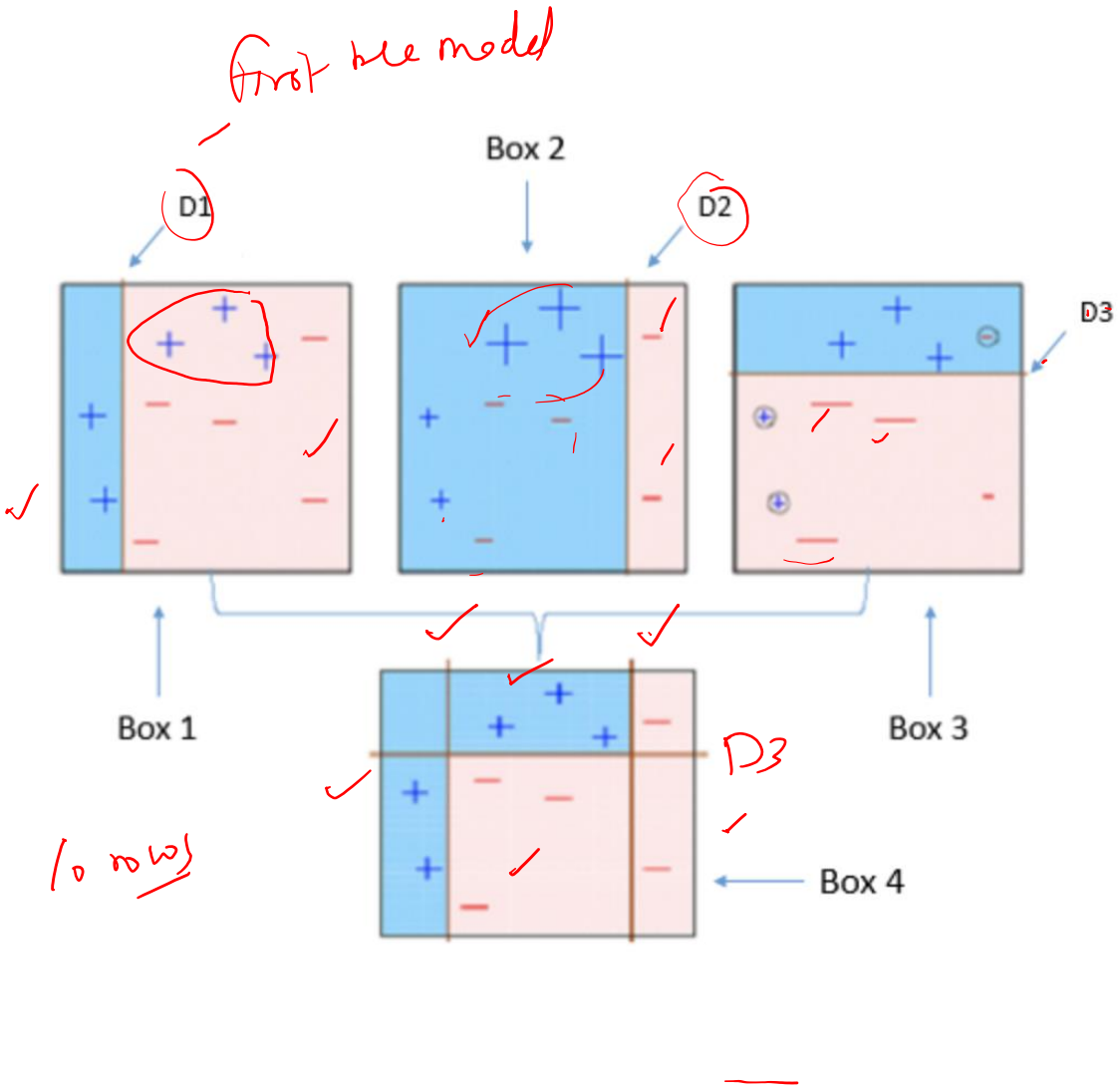
2. Then , the model is build and predictions are made .
3. Then the error is calculated i.e the difference between the predicted and the actual values .

4 . The data points which is incorrectly predicted is given a new and a higher weight .
Another model is created , predictions are made and Errors are calculated .
5. The number of times an instance is replicated is proportional to its weight.



Similarly, multiple models are created, each correcting the errors of the previous model.
The final model is the weighted mean of all the models.

Let's see the working of Boosting with a diagram



In this diagram , we have two classes + and – and D1 is our first model , where all the data points are assigned equal weights .

- In D1 three + datapoints are misclassified (as seen in the D1 box)
- So, while creating second model these three + datapoints are assigned higher weights and we can see in D2 , after assigning higher weights they are correclty classified .
- But again in D2 I.e model 2 , three – datapoints are incorrectly classified so they will be assigned with higher weights for proper classification
- In D3 the previously incorreclty classifed points are classified correclty .

In the D4 model , all the previous models are combined to form a strong correct prediction

XG Boost

DATA FOLKZ®
#CATAPULT DATA LEADERS

XG Boost also known as ***"Extreme Gradient Boosting"***
A special case of boosting where errors are minimized by gradient descent algorithm .
We are doing a weight updation here. That weight updation is done using the gradient descent approach
The main aim of this algorithm is to increase the speed and efficiency of computation.

Features of the model :

- **Regularization**: This is considered to be a dominant factor of the algorithm.Regularization is a technique that is used to get rid of overfitting of the model.
- **Cross-Validation:** We use cross-validation by importing the function from sklearn but XGboost is enabled with inbuilt CV function .
- **Tree pruning** using depth first approach
- **Execution Speed** : XGBoost performs extreme optimization in terms of RAM utilization , CPU core utilization and cache utilization.

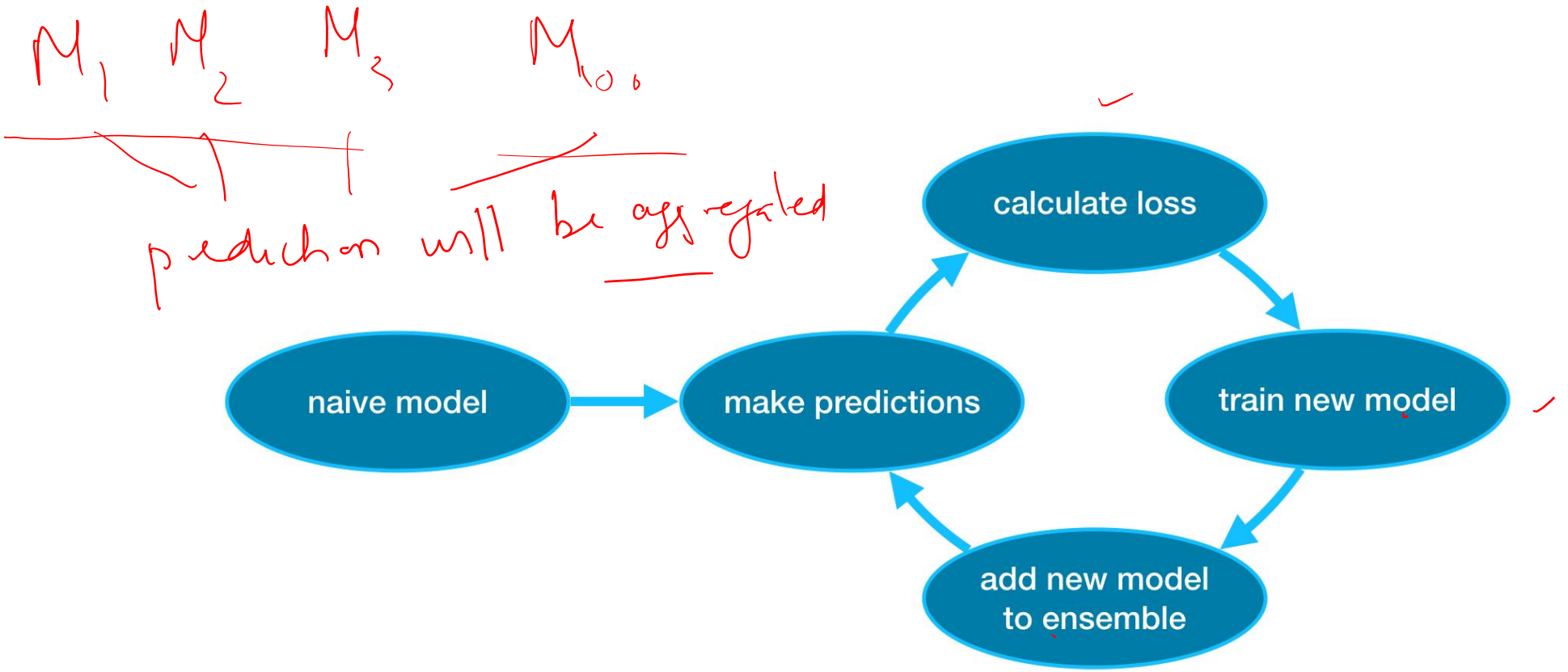
XGBoost

Like **random forests**, gradient boosting is a set of **decision** trees. The two main **differences** are:

How trees are built: **random forests** builds each tree independently while gradient boosting builds one tree at a time.

Outliers can be bad for boosting because boosting builds each tree on previous trees' residuals/errors. **Outliers** will have much larger residuals than non-**outliers**, so gradient boosting will focus a disproportionate amount of its attention on those points.

XG Boost


M1 M2 M3 M100
prediction will be aggregated
calculate loss
naive model
make predictions
train new model
add new model
to ensemble

DATA FOLKZ®
#CATAPULT DATA LEADERS