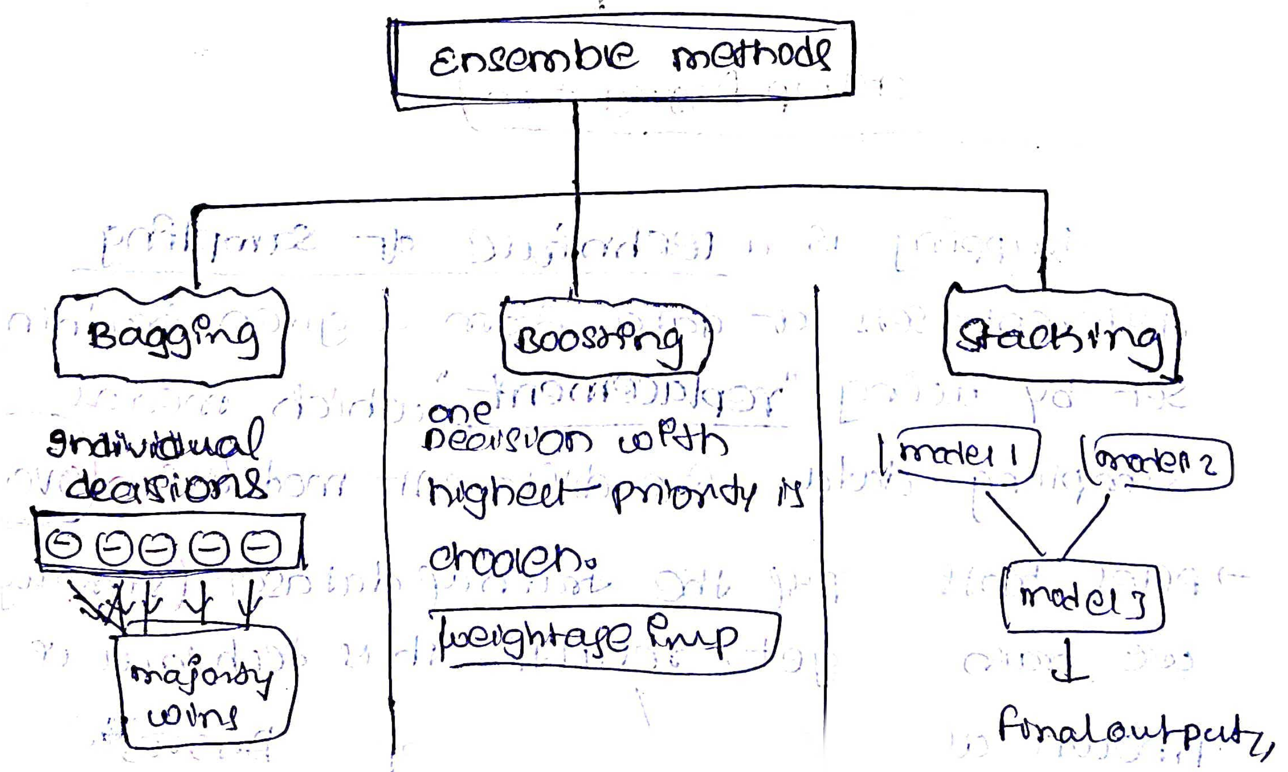
multiple decision makers

Ensemble Techniques & Random Forest

→ until we are predicting using one model, but what if we can take models and predict the output, won't it be better than one. ofcourse it will be. this is the idea behind Ensemble.

→ we regularly came across the option of "audience poll". And constestant mostly win when goes for option which has highest vote from audience. so, we can say, taking opinions from a majority of people is much more preferred than opinion of single person.

→ Ensemble techniques has a similar underlying idea where we aggregate group of predictions from a group of predictors. such algorithms are called "Ensemble methods" and such predictors are called "Ensembles".



(*) Let's suppose we have 'n' predictors (models).

→ $z_1, z_2, \ldots z_n$ with a standard deviation of σ

$\boxed{var(z) = \sigma^2}$ — var of single predictor.

$\boxed{Avg(\mu) = \frac{(z_1+z_2+\ldots z_n)}{n}}$ — Average of predictors. (Expected val)

If we use 'n' as predictor then expected val still remains the same, but **variance** is **reduced** so much.

$\boxed{var(\mu) = \frac{\sigma^2}{n}}$

This is why taking mean is preffered over single predictor

(*) So, Ensemble methods take multiple small models and combine their predictions to obtain a more powerful predictive power.

Bagging (Bootstrap Aggregation)

→ Bootstrapping is a technique of sampling different sets of data from a given training set by using "**replacement**", which means sampling data for different models can overlap

→ After bootstrapping the training dataset (sampling), we train & get results. This technique is known as Bagging / Bootstrapping Aggregations

→ Bagging is a type of ensemble technique in which a single training algorithm is used on a different subsett of training data where samply is done with replacement (bootstrap).

→ In case of Regression, Bagging prediction is simply the mean of all the predictions & in case of classifier, Bagging prediction is the most frequent prediction (majority vote).

(*) Bagging is also known as "parallel mode" since we run all models parallely and combines results at the end.

Advantages	Disadvantages
① Bagging significantly decreases variance without ↑ Bias.	→ It improves the accuracy of model on the expense of "interpretability".
② It works so well bcoz of diversity in training data.	we can't interpret as there will be so many models.
③ works well with smaller dataset.	
④ If training set is very huge, it can save computational time by training model on relatively smaller dataset & still can increase the accuracy of model.	

Pasting :- similar to bagging but no replacement in sampling. So this causes less diversity in the sampled datasets and data ends up being correlated. That's why Bagging is more preferrable than pasting.

Boosting

→ Boosting is an ensemble technique (involves several trees) that starts from a weaker decision & keeps on building the models such that the final prediction is the weighted sum of all the weaker decision makers.

$\boxed{\text{the weights are assigned based on performance of individual tree.}}$

→ In boosting, ensemble parameters are calculated in "stagewise way" which means that while calculating the subsequent weight, the learning from previous tree is considered as well.

i.e) the learning of previous tree boosts the learning of next tree and goes on – so called Boosting

→ we mostly used decision tree as weak classifier. Any other algorithm can be used as a base, but reason for choosing trees are :-

Pros	Cons
1) Robust to outliers	1) Inability to extract a linear comb of features.
2) Feature scaling not required	2) High variance leading to small computational power.
3) Handles missing values	
4) can deal with irrelevant inputs	
5) computational scalability	

→ Boosting minimizes the variance by taking into consideration the results from various trees.

General understanding eg of boosting

① you wanna travel to different place.

→ And you know a friend who is a traveller, so you ask him about the place (gives your info ...) He will tell what he know & he will ask his friends.

→ He asks his friend who visited there to give some phone no or any agency.

→ And that friend asks another friend who lives there.

First, we have to boost other person with info we have, and he will boost next person with info he have & finally get result

② Dad wants to buy car.

→ He asks you & without any knowledge you say 'yes'. (less weightage)

→ Grandpa thinks and says what model should be bought consider grandchildren also wants car. (Good weightage)

→ consider son & grandpa, mom says what to buy, ~~[illegible]~~ new or second hand, based on financial status (Better weightage)

Stacking (Stacked Generalization)

- Stacking is a type of ensemble technique which combines the predictions of 2 or more models, also called base models and use the combination as the input for a new model (metal model) i.e.

> meta model – A new model is trained on the predictions of base models

- Suppose you have a classification problem & you can use several models like LogisticReg, SVM, KNN, Random forest etc.

the idea is to use few models like KNN, SVM as the base model & make predictions using them

- Now predictions made by these models are used as an input feature for Random forest to train on & give prediction.

- Stacking can be multi level;

using base model as level 1 then pass predictions onto another sub-base models at level 2 & so on

> then at last using meta-models which take predictions of last sub-base models as input & does predictions,

working

1. split the dataset into a training set & a holdout set. Generally we do a 50-50 split of training & holdout.
training set = x_1, y_1 ; Holdout set = x_2, y_2 (validation set)

2. split the training set again into training & test set.
like, x1_train, y1_train, x1_test, y1_test

3. train all the base models on training set x1_train, y1_train

4. After training is done, get the predictions of all the base models on the "validation set x_2"

5. stack all the predictions by diff models together as it will be used as input feature for the meta_model.

6. Again, get the predictions for all the base models on the test set x1_test, and stack all these predictions together as it will be used as the

7. prediction dataset for the metamodel.

8. use the stacked data from step5 as the input feature for metamodel & validation set y_2 as the target variable & train the model on these data.

9. once the training is done check the accuracy of metamodel by using data from step7 for prediction & y1_test for evaluation

10. End

visual interpretation