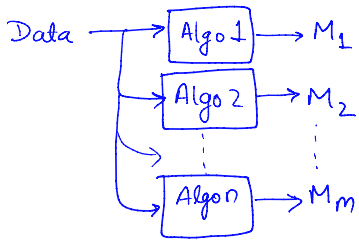
Ensemble Techniques - 1

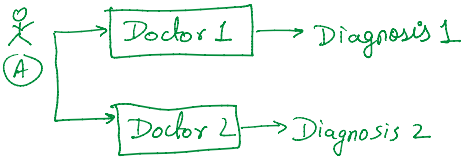
23 July 2023 05:30 PM

Ensembles

→ means a group of objects (models)



Analogy for ensembles ↓
Let say person A is very ill

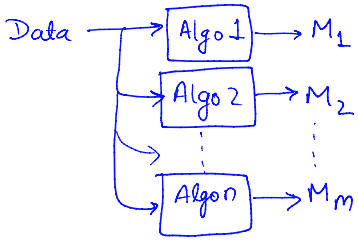


The person goes to multiple Doctors in order to ensure that the Diagnosis is correct

---

Ways to Build Ensembles

① Parallel Ensembles

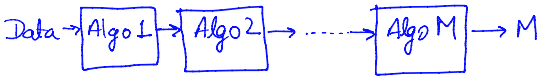


1.1 Voting Ensembles

1.2 Stacking (sklearn)

1.3 Bagging → Random Forest Algo. (sklearn)

② Sequential Ensembles



1.1 Cascading

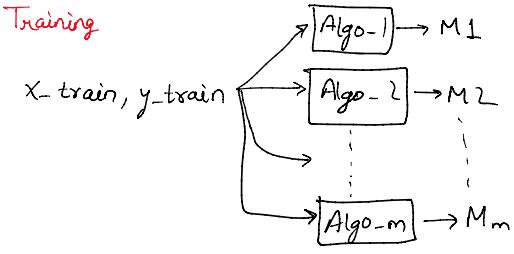
1.2 Boosting

- sklearn → ADA Boost (Adaptive Boosting)
- sklearn → GBDT (Gradient Boosted Decision Tree)
- xgboost → XG Boost. (Extreme Gradient Boosting)

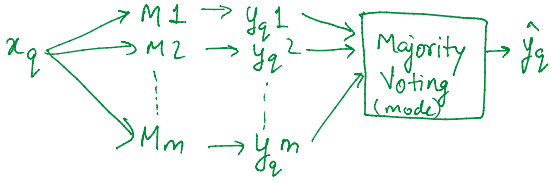
Voting Ensemble ↓

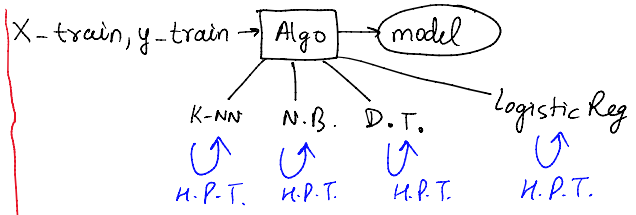
Classification Task →

Training



Prediction

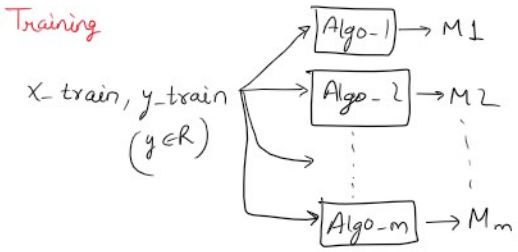


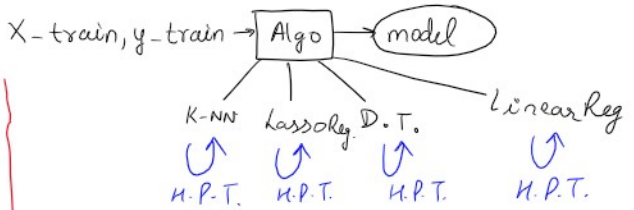


} Final model selection from the best is made on the basis of requirements.

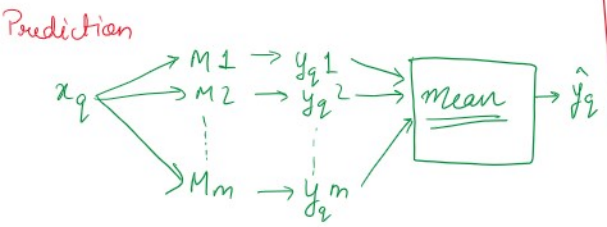
Prediction ↓

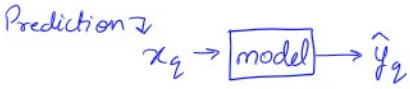
Regression Task



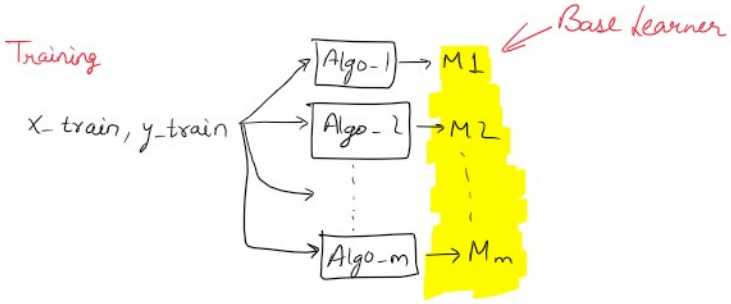


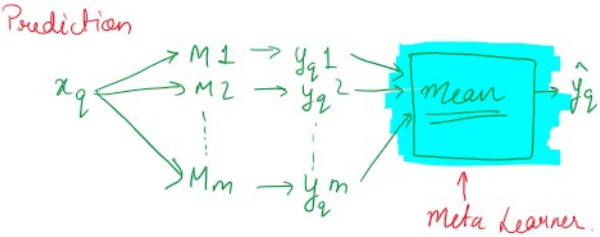
} Final model selection from the best is made on the basis of requirements.





Important Terminologies for Ensembles





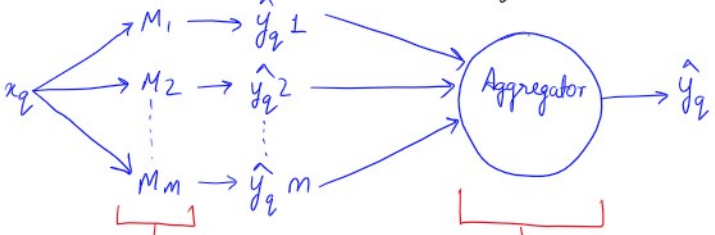
In Voting Ensembles

Classification
- Base Learner → K-NN, D.T., N.B., logistic Reg etc.
- Meta Learner → Mode i.e. Majority Voting

Regression
- Base Learner → K-NN, D.T., Linear Reg., Lasso Reg., Theil-Sen Reg. etc.
- Meta Learner → Mean or Median

Issue with Voting Ensemble

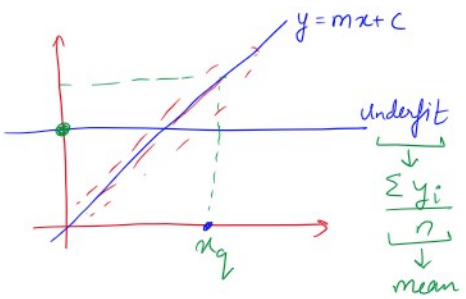
→ It has very high computational complexity because of hyper parameter tuning for each algorithm in the Base Learner.



Base learner → Underfit or overfit → Hyper parameter tuning

Meta Learner = Mean or Mode → Underfit ← Bound to happen

Eg



Voting ensemble has the problem of → overfitting & underfitting + Hyperparameter Tuning of multiple models which made it computationally very complex

As a "solution" to the problems of Voting Ensembles → we go to a stronger ensemble method

↓

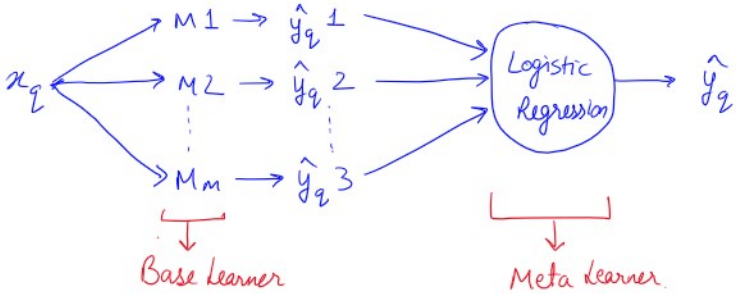
Stacking

Stacking

Classification →

Training → Same as Voting Ensembles (Same in Regression as Well)

Prediction →

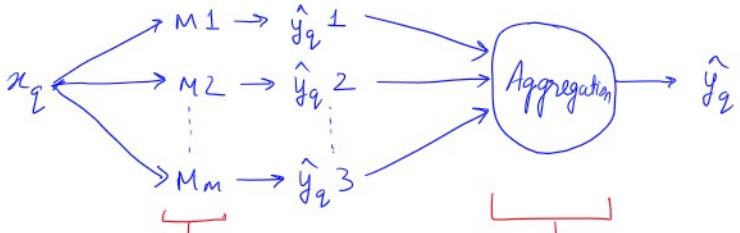


Base Learner → Same as voting ensemble

Meta Learner → Classification → Logistic Reg., D.T., N.B., etc.

→ Regression → Linear Reg., D.T., K-NN, etc.

Now in Base Learner as well as Meta Learner, in both we can have overfitting or underfitting and as a result we have just ended up introducing more complexity than voting ensemble.



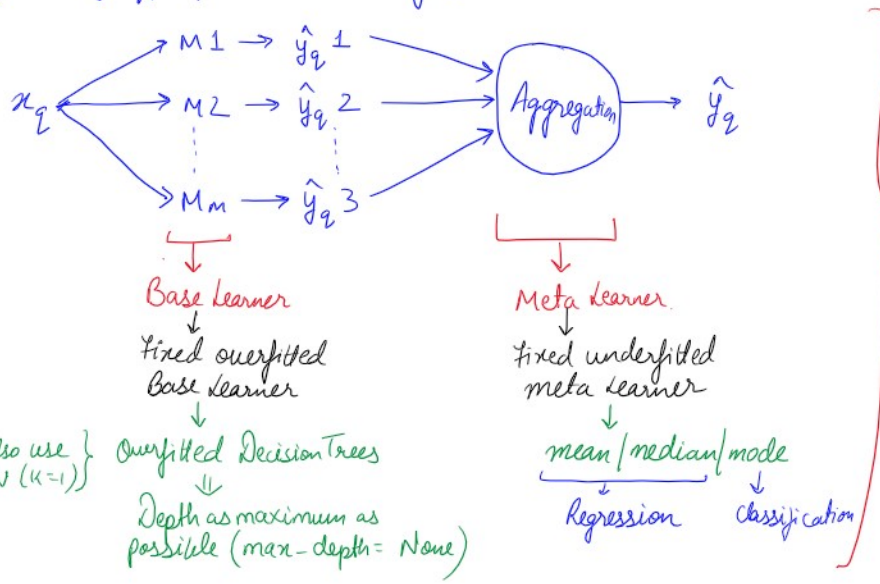
Base Learner	Meta Learner	
Overfit	Overfit	→ Overfit ✗ (Problem of Stacking)
Underfitted	Underfitted	→ Underfit ✗ (Stacking & Voting Ensemble)
Overfit	Underfit	→ Best fit { Stacking, Voting Ensemble }

↓

But requires a lot of hyperparameter tuning

Bagging (Random Forest)

One of the best parts about bagging is that it is able to take care of the underfitting & overfitting problem without doing a lot of hyperparameter tuning (Little to None).



BAGGing [RANDOM FOREST ALGORITHM]

- Bootstrap Sampling of Training Data
- Base Learner → Overfitted Decision Tree
- Meta Learner → Classification → Mode; Regression → mean/median } Underfitting

{Random Forest also utilizes Central Limit Theorem internally} → Ref. Statistics

* Bootstrap Sampling {The sampling is done on rows as well as columns}
  - With Replacement → Any datapoint can be Randomly selected for Random No. of times.

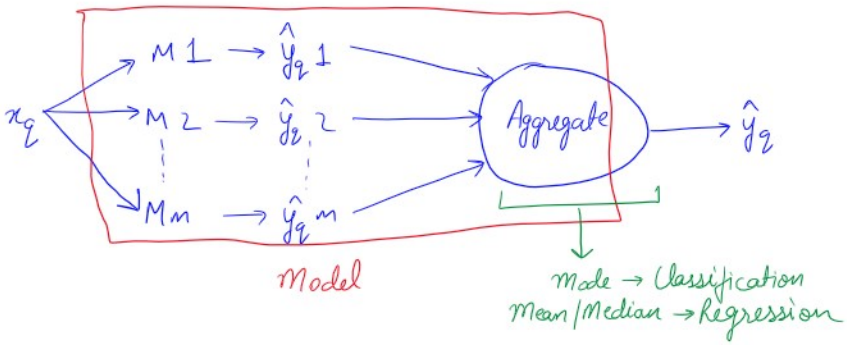
No. of rows & columns in sample: $r \leq R$, $c \leq C$ } No. of rows & columns in training data

{Bootstrap Sampling of training data with replacement} → {It is done so that we don't end up exhausting the data}

This is responsible for generating randomness.

* All the samples will be different from one another as the Bootstrap Sampling will be done on rows as well as columns

Prediction



Random Forest

↳ Base Learner → Decision Tree (Overfitted)

↳ Meta Learner → Mean/Median [Regression]
Mode [Classification]

Training Random Forest → Bootstrap Sampling with replacement for rows & columns of training data.

→ On Each Sample → Train an overfitted Decision Tree Model

Prediction with Random Forest

→ Pass x_q to all models and get their predictions

→ Aggregate all the predictions of model using: mean/median [Regression]
mode [Classification]

[Code Example]