Classifier => Classifier ko training data diya jata hy which has labels already. Then it predicts new data labels.

Bagging & Boosting

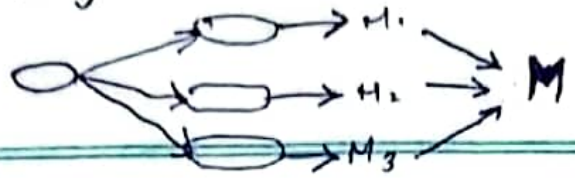
- We use Multi-classifier Combination (Mcc) to solve classification problem.
- When data complex, Basic or 1 classifier does not perform well. By random guess result may be better.
- If multiple independent classifiers are used together, they strengthen each other.
- Idea originates from bootstraping (Iteratively improves 4 classifiers performance)

Baron's story
↓
Tries to pull himself and his horse from a swamp by his hair.

Apne limited data resources ko use krky huway apni problem ka solution khud sey improve krna, no external help.

- Bootstraping
- Seed AI is strong AI capable of improving itself. bot no such system exists. which can improve itself efficiently.
- Example → • We will calculate Avg height of people in world.
  - Take sample size N. + make samples with sampling with replacement. (Same person measured twice)
  - Repeat 1000 times + calculate Avg + Variences.
- Bagging (Bootstrap Aggregating) (Leo Breiman introduces)
  - Starting from training set, draw n samples with replacement (Supervised approach) n > no. of records.
  - Train classifier on resulting samples.
  - Repeat process m times to learn m classifiers.
  - Bagging gives multiple models equal vote for classification.
  - Models vary bcz training data kay chote hissay ko select kr kay model banaya jata hy. so model varies.

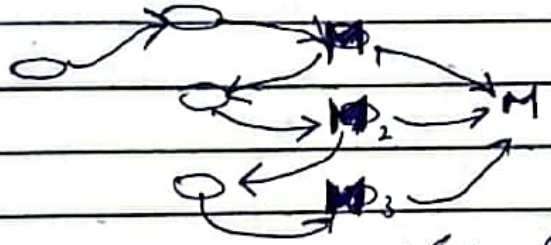
Bagging → Advantages • Increases classifier stability + reduces variance.



- When new record comes, perform majority votes.
- Har sample ka select hony ka probability $1-\left(1-\frac{1}{n}\right)^n$.
- Bagging Algo → • Randomly selecting training sets.
  - Train classifier on each dataset.
  - Classification of new data record (more votes ↓ selected)

weights are fixed.

- Boosting (Michael Kearns) (Bootstrap Aggregating)
  - Classifiers are formed in series.
  - Har classifier us data pr zada focus krta hy jo phly waley classifier ne galat classify kiye ho.
  - Can use basic classifier such as decision trees or SVM.
  - At start all records are assigned equal weights.
  - Weights may change. (After model trains for 1st time, data points which model predicted wrong are assigned higher weights bcz next round din pr focus). Which are corrected predicted are given less weight.



- For hard classification problems, usual classifier tends to be weak learners. ↓ weak learner is only slightly better than random guessing.
- Set of weak learners create strong learner.
- Naive Approach for boosting (Majority vote).
  - Every classifier learns from different training data when new classifier comes all classifier votes and that class is selected as final.
  - Problems → • what if majority classifiers are wrong.
- Adaptive Boosting (Yoav Freund, Robert E Schapire) (AdaBoost)
  - Train 1st base classifier on training set.
  - See which are not well explained by this classifier.
  - Assign weight. (wrong ko zada)
  - Train new base classifier on weighted training set.
- Reweight as in step 3 + Repeat step 4 and 5 to create set of base classifiers.

Bootstrap aggregating (introduces voting principle)
Boosting (weights for falsely classified objects)
Adaptive Boosting (weights also for classifiers)

- Adaptive boosting → • Multiple base classifiers are created. So give importance weight to them.
  - Weight depends that every classifier kitni achi tarah training data ko fit krta hy.
  - High importance → Agr koi classifier aur examples ko galat predict krta hy jin ki weights kam hain, tou aus classifier ko zaida importance.
- So, jo classifier zaida important hta so aus key vote ko more importance. So, ais tarah weighted majority vote se final decision liya jata hy (more good).
- Balanced weights means that votes of classifiers & training examples weights ko aisay adjust krna key majority vote should be right (more accurate)
- AdaBoost is good as new classifier pechlay classifier ki kamzori pr work krta hy. + kamzori compensate and strengths used.