meaning :- taking knowledge from the crowd
— same concept used in Random forest

Ensemble learning - Wisdom of the crowd

In Ensemble learning, the wisdom of the crowd is the idea that a group of diverse models, when combined, can make accurate predictions than any single model alone — similar to how a group of people can often make a better decision than one person

In ML:
— Each decision tree in a Random Forest is like one "person" in the crowd.
— By combining their predictions, the final result is more reliable

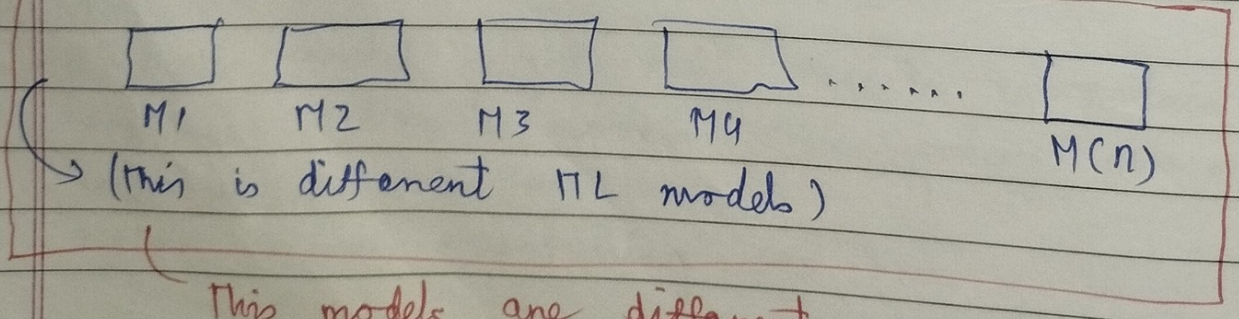
- For Classification → Take the majority vote
- For Regression → Take the average prediction

Ensemble Learning — Core Idea

- Machine learning algorithm : Training
  \ Testing (Prediction)

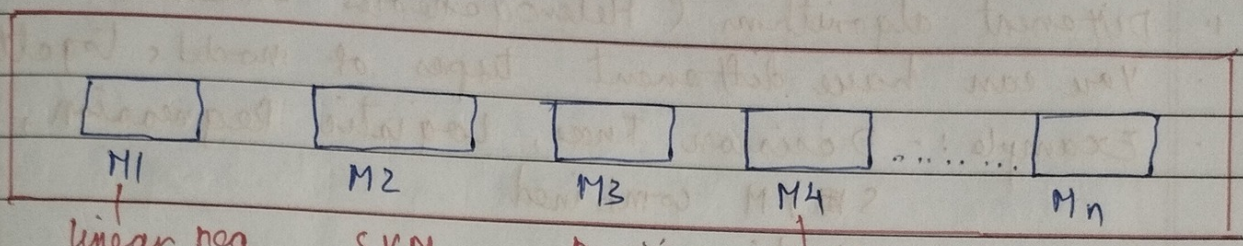
What happens in prediction stage :
For eg we have Ensemble learning Algorithm :-
↳ collection of smaller basemodels

this is called Ensemble learning model



↳ (This is different ML models)

This models are different
& we always need different model in Ensemble learning



→ This are Different models

(The model in Ensemble learning needs to be different or at least behave differently because if they are all the same and make the same mistakes, combining them won't help — you'd just be repeating the same error)

Why diversity is important:

1. Error cancellation
2. Better coverage of patterns:
3. Reduced Overfitting

→ (1) If one model makes wrong prediction, another model might make the correct one
When combined, the mistakes cancel out, improving accuracy

→ (2) Different models capture different aspects of the data
Eg: A decision tree might capture non-linear patterns while a logistic regression might capture linear patterns

→ (3) If all models are trained the same way, they might all overfit to the training data
Different models or randomization make them generalize better

— In Ensemble model 3 core ideas

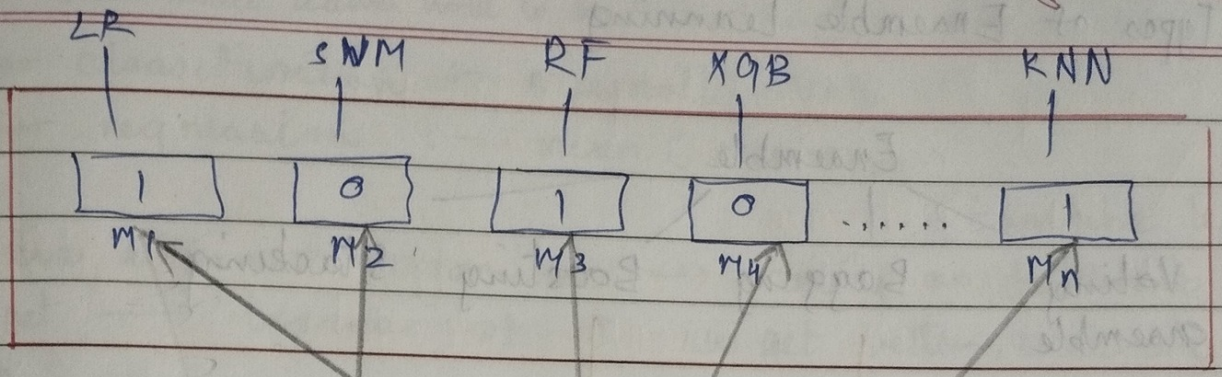
1. Different algorithms (Heterogeneous ensemble)
   - You can have different types of models together
   - Example :- Decision Tree, Logistic Regression, SVM combined
   - Common in stacking

2. Same Algorithm, trained differently (Homogeneous Ensemble)
   — on different data
   - You can use the same type of model but train them on different subsets of data or with different settings
   - Example :- Many decision trees trained on different random subsets → Random Forest

→ we use the same algorithm but we train it on different data
↓
model train differently
↓
variety difference

1. Heterogeneous Ensemble :- Different Algorithm same data
2. Homogeneous Ensemble :- same Algorithm but different subset of data
   ↓→ this is mostly used in Bagging & Boosting
3. Hybrid Ensemble :- Hybrid Ensemble is a mixture of both where we train on different algorithms (Heterogeneous) and also train on different subsets of data (Homogeneous)

1 - Yes
0 - No



- A person with 85 IQ and 7.5 CGPA :- Prediction will be placed or not ? Classification Problem

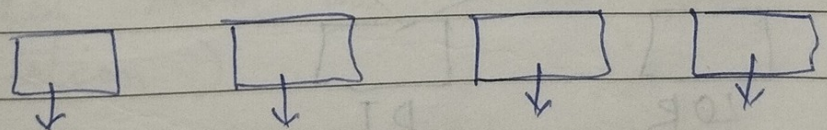
We give this data to different algorithms for eg we have (5)

1 :- ~~Encoder Dec~~ Logistic Regression (1)
2 :- SVM (0)
3 :- Random Forest (1)
4 :- XGBoost (0)
5 :- KNN (1)

out of 5 → 3 said Yes → and 2 said No
:- ~~out~~ final output will be :- Yes because in Ensemble learning — majority vote model

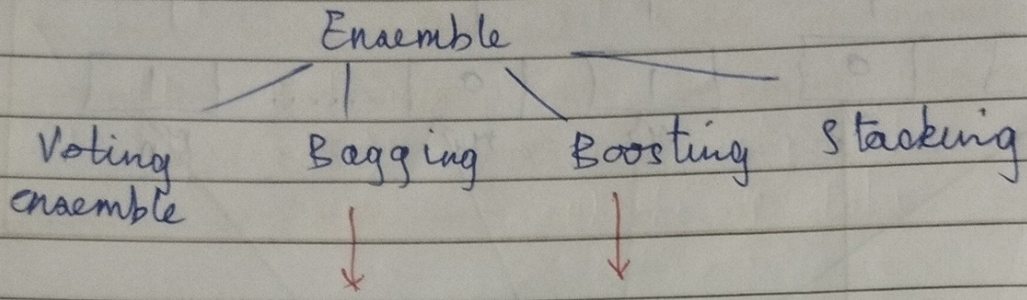
In Regression :- How much LPA? we will get ?

Eg



$$\frac{8.1 + 3.2 + 4.1 + 7.5}{4} = 6.6 \text{ LPA}$$

Types of Ensemble Learning



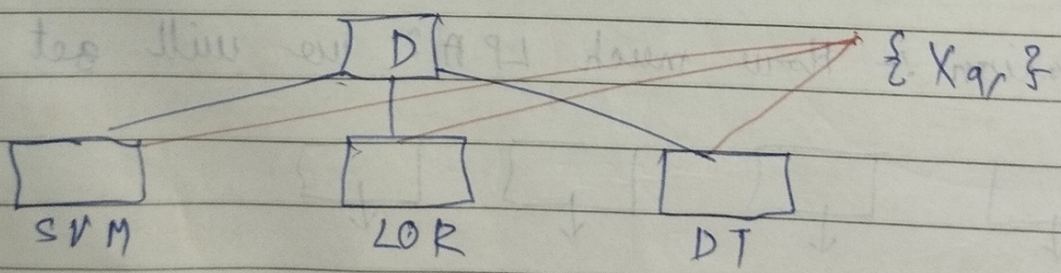
- XG - Extreme Gradient Boosting
- Ada — Adaptive Boosting
- Light GBM :- Light Gradient Boosting Machines

Why exist different types

→ because of 3 core Ensemble Learning model ideas

1) Hetenogeneous Ensemble
2) Homogeneous Ensemble
3) Hybrid Ensemble

1) Voting Ensemble :- Whether our base models have different Algorithms (Hetenogeneous)



Same Dataset D :- Used for different algorithms such as SVM, LOR and DT and now we train it on our data D

→ In stacking :- Instead of directly voting, the predictions from the base models becomes the input features for the meta-model

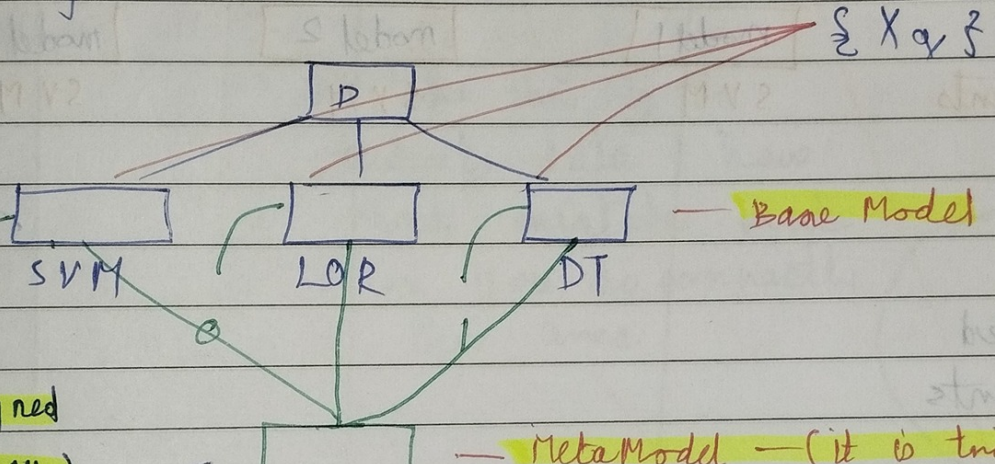
→ The meta-model learns how to best combine base models output.

for classification :- majority vote

for regression :- mean (average)

Since different algorithms → variety or difference we get → because of it we get better, accurate result (Ensemble learning mode)

2) stacking (Heterogeneous Ensemble Learning)



(weight is assigned based on accuracy)

↓

more accurate (more weight)

↓

less accurate (less weight)

(This model does one work:
— assigning weights to the above algorithm based on the data who has more weight or correct (accurate))

from where we get data
(We get from the model itself which makes prediction)

For Eg

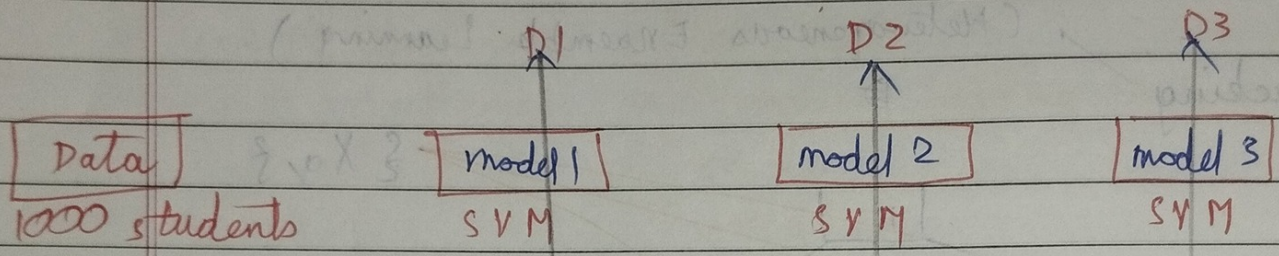
SVM	LOR	DT	Result
1	0	1	1
0	0	1	1

based on this result it assigns weightable or weight to model (like who's more correct or accurate or rich) (This could be of any reason)

3) Bagging → Bootstrap Aggregation

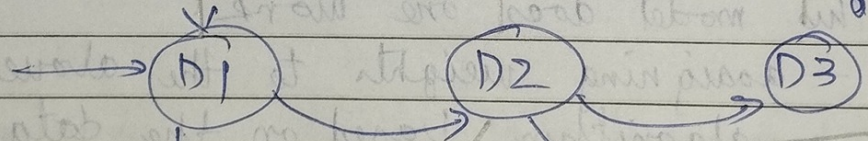
↘ this follows (Homogeneous) → same algorithm and different subsets of data

{ X_q }



For eg we gonna feed 500 students data

↗ Random forest → base model → D trees → a collection of trees



In Final round I took random 500 students data

after this I again returned the 500 students data and again took 500 random students data

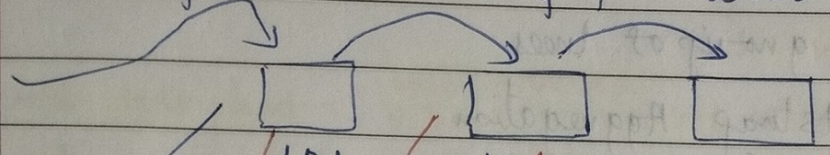
returned the data and again took random 500 students data

this is called Bootstrapping

Probability of getting the same data is relatively lesser in Percentage or number

most powerful machine learning Algorithm

4) Boosting :-



Boosting your data :- tells the other model everytime

↓

in series

↓

it reduces the mistake

during training it remembers what mistake it has made

↓

Data

↓

1000 students

it tells this is the students data I have made mistakes make sure you do correctly in this area

It tries to make sure not do the same mistake and then it does something happens it tells the other model to make sure it doesn't repeat the same mistake and do it correctly

Random Forest

Random Forest (Bootstraping)

Random Forest
↑ ↳ group of trees
Bagging → Bootstrap Aggregation
↳ Sampling
(Means: selecting Data Randomly)

Bootstrap Aggregation comes in this fact called sampling

(because of this the word Random came out)

Decision Tree

D_1 = [1, 18, 32, 43, 103 ..... 998]
(any random values choosen randomly)

D = Dataset
D = 1000 rows → (we only give 500 rows out of 1000 rows)

(we give our data to decision tree but not fully) → DT1 DT2 DT3 DT4 ..... DT100

→ what we do is :- we create samples out of it :

* — 3 Sampling techniques

1 — row sampling

here also 2 types of selection

1. ~~with sampling~~
2. ~~without~~

1. with replacement
2. without replacement

we can have duplicate values ← because — one data comes & and it again goes back to the Database which can repeat again & again :- D = [1, 2, 91, 1001, 2, 1, 91]

with replacement → duplicate values

without replacement → no duplicate values

Data Subset → Creation Ideas

without replacement → no duplicate values

increases diversity

1. row sampling < with replacement / without replacement — classic bootstrapping — less diversity
2. column sampling also called as feature sampling < with replacement / without replacement

eg (randomly choosing 5 columns out of 10) — D1

↓

for another (decision tree we again randomly choose 5 columns)

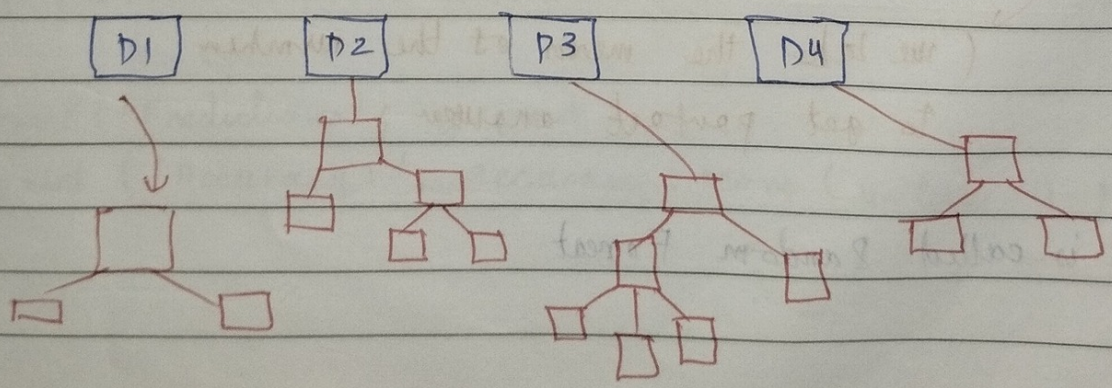
↓

same we do for other decision tree

This is called as Random subspace method

3. hybrid sampling (row sampling + column sampling) — or combining
   - with replacement (duplicate values)
   - without replacement (no duplicate values)

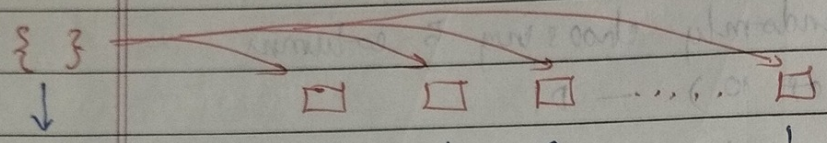
* Since we give different subsets of data to each base model their structure can also be different



→ This whole thing is called as bootstraping

Random Forest (Aggregation)

Aggregation :- When we want to predict
— (classification problem)
For eg 1000 decision trees are trained → now they can predict → think this as a classification problem



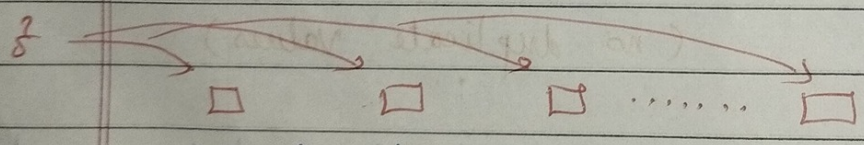
this is our ~~query~~ query point or testing data

0 - 43/100
1 - 57/100

(Since 57 is greater than 43, most of the decision tree says that :- incoming query point should be ~~the~~ 1 → label with be 1)

(This is what aggregation is called as)

— If it is a regression point



query point

(we take the mean of the numbers to get perfect answer)

→ This is called Random Forest