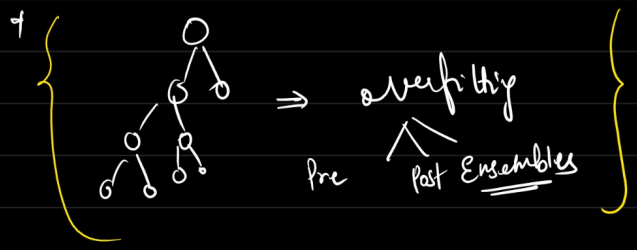
Till now

- Linear data → MLR, Log Regression
- Non linear → DT, SVM

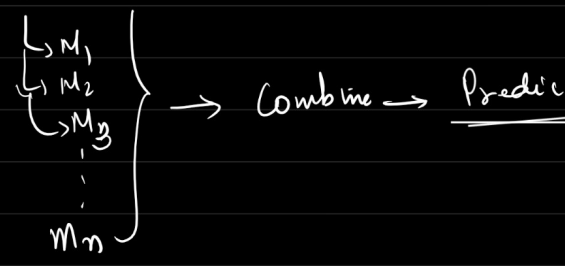


* Ensembles and its technique

→ one person might give you wrong advice.

* data → Model → train → Prediction → multiple mentors → Chance of getting wrong is minimized.

* data



* Ensembles: Combine multiple models

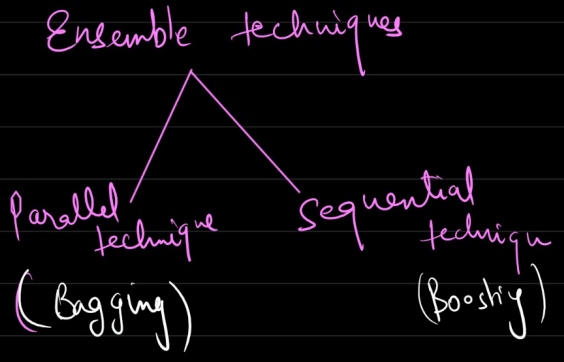
: Prediction : more stable and accurate as compared to individual model.

of same Algorithm	different algorithms
① DT_1 → max depth = 5	→ Logistic Reg
DT_2 → max depth = 10	→ SVC
DT_3 → max dep = 12	→ DTC_1 (5)

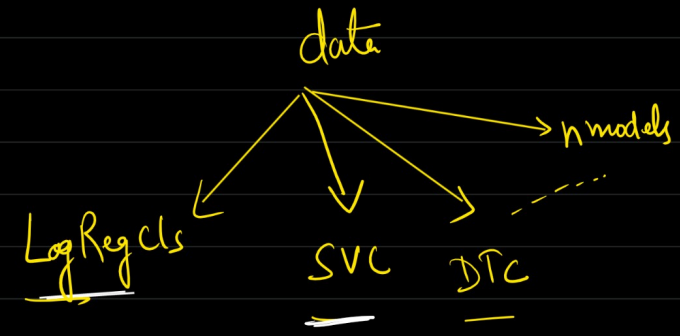
→ DTC2 (15) //
→ NB_C

n

* Ensemble → No necessarily only one type of Algorithms

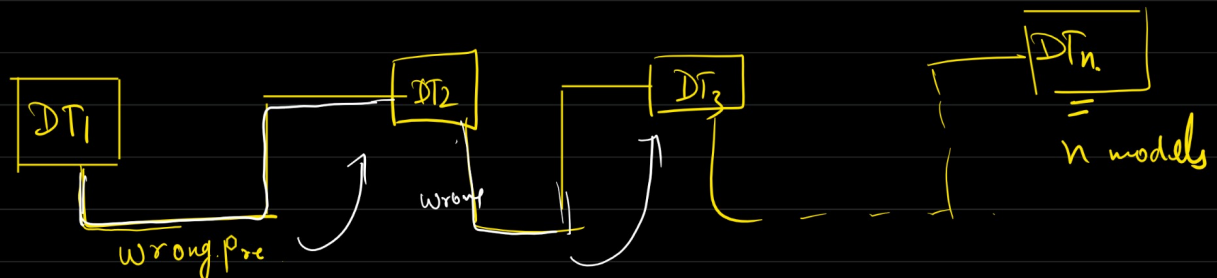


* Parallel technique of Ensembles



→ All the models here are built Parallely and independent of each other.

* Sequential technique of Ensembles

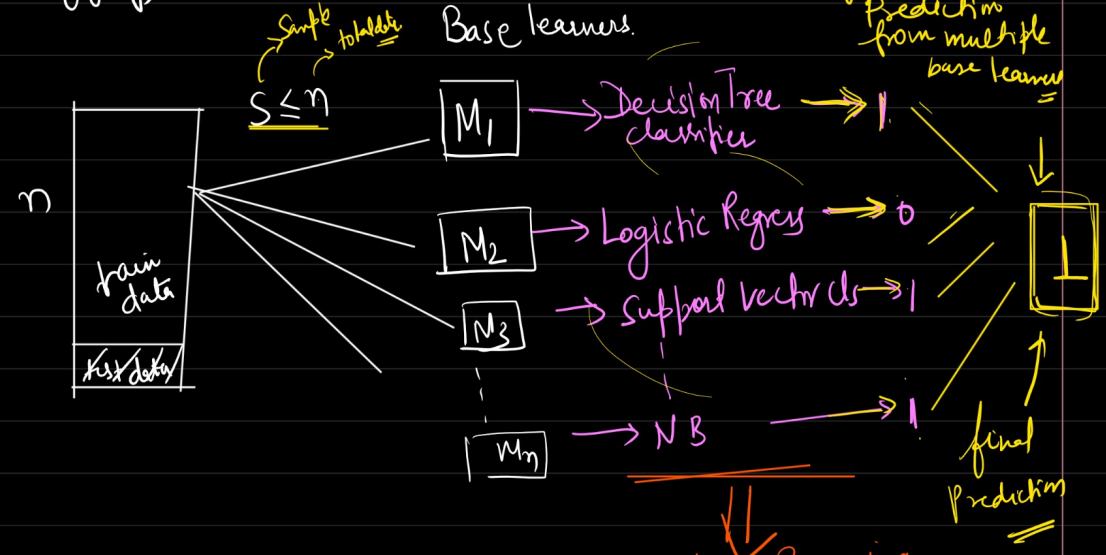


Ensemble models

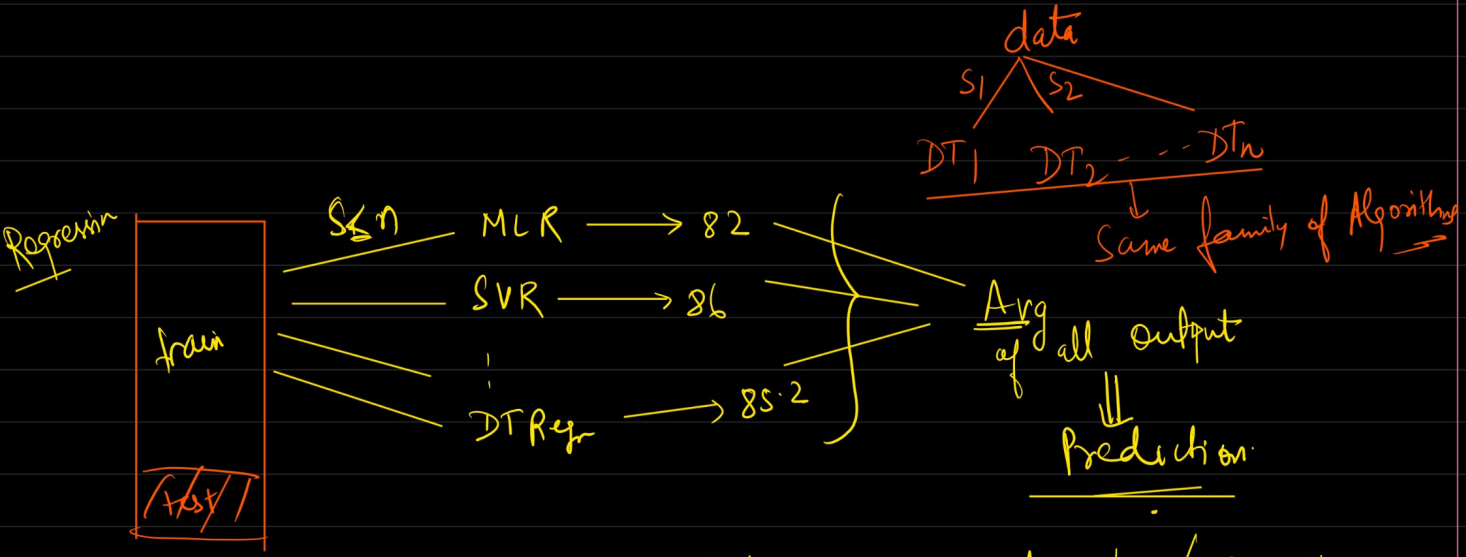
- Bagging (Random Forest)
- Boosting (AdaBoost, Gradient B, XG Boost)

* Bagging (Custom Bagging)

* Parallel models



Custom Bagging technique.



→ Custom Bagging Classifier / regression

→ Pipeline & column transformer

S-1 * missing value treatment
S-2 * One hot encoding
S-3 * Scaling.

* Pipeline

Pipeline

Pipeline ⇒ Encoding | Scaling | Imputation

→ A sequence of data transformation

→ f_1 f_2 f_3 ... f_{100}

Set of all feature.

⇓

ColumTransform.

⇓

Group all the pipeline steps for each of the column

* Random Forest classifier and regressor

Random forest

⇓

Bagging technique.

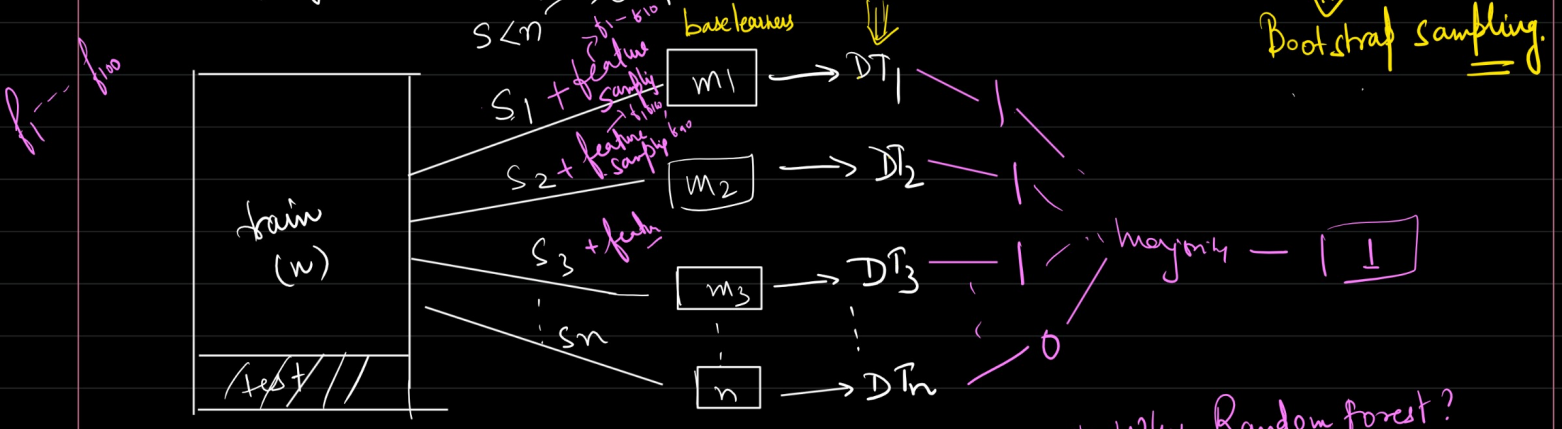
⇒ random samples with replacement → data can be repeated across the sample.

⇓

Bootstrap sampling.

Random Forest

↳ All the base learners are DT.



* Why Random forest?

→ Multiple DT's in parallel

→ row & features will be randomly selected.

Bagging?

↓ Bootstrap ↓ Aggregating.

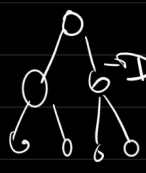
↓ ↓

different samples with replacement Bootstrap sample

Aggregating the Prediction

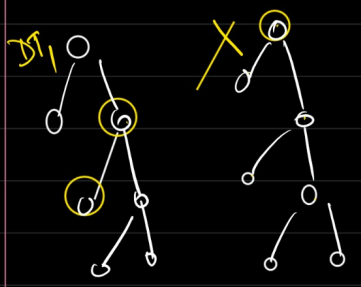
Random forest { * Classifier → Majority is the prediction
* Regressor → Avg will be the prediction

* Random forest reduces the variance



DT → overfitting → high variance.

Random forest
↳ random sampling of rows & features
⇓
data is splitted into small chunks (randomly)
⇓
due to random subset of rows and features, each of the subset is different representation of itself
→ Each of random subset trains a diff. model.

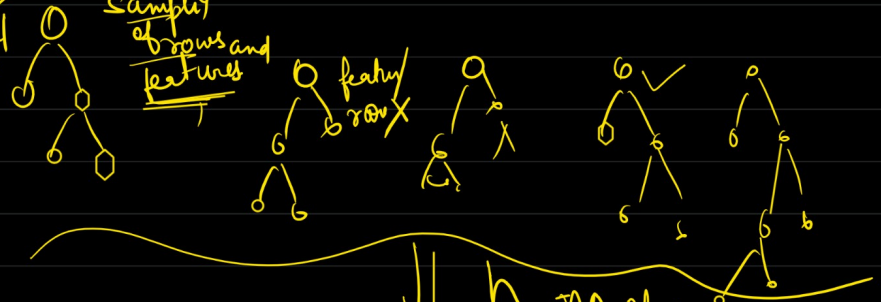


→ feature/row which was causing an overfitted model in one DT, might not be contributing to Overfitting in other DTs
⇓
reduces the variance.

- Scen-1

→ high var
⇓
feature/rows

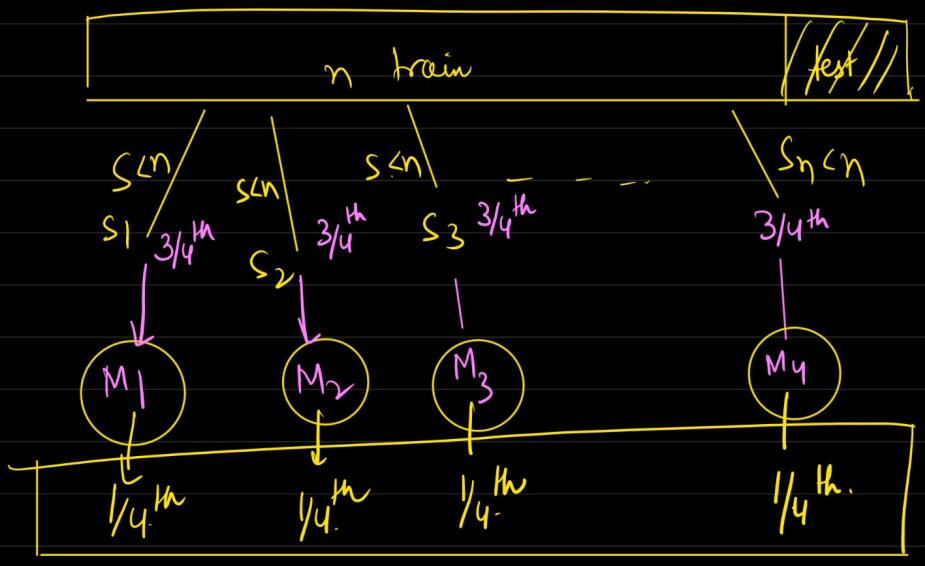
Scen-2 random sampling of rows and features
feature/row



feature/row

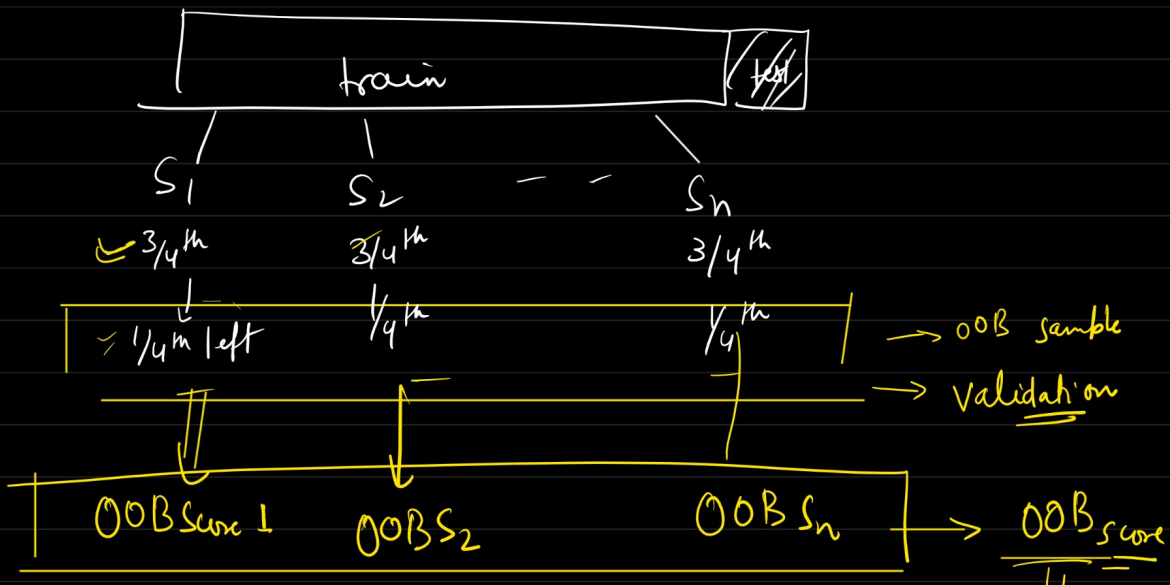
⇓
h no of trees
⇓
final prediction
→ overfitting.

OOB Score



→ 1/4th acts as validation for each specific Individual DT.

OOb score



→ In training itself OOB is low, model is not performing well on train data.