15 Oct 2023

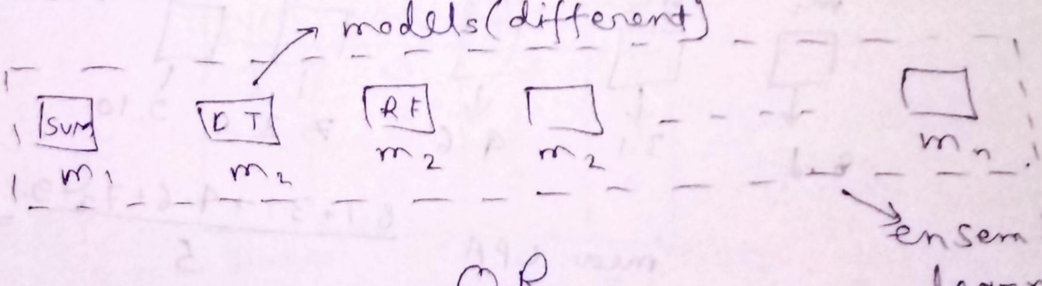
Ensemble Learning

Ensemble learning:
It is a collection of machine learning models.

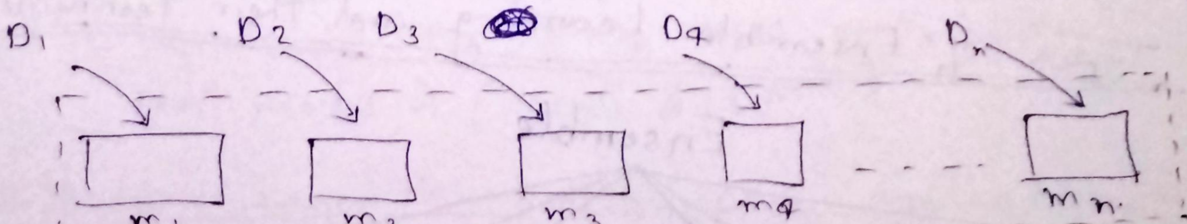
wisdom of the crowd

- in KBC the life line audience poll
- in democratic country the winner of election
- in amazon product reviews
- in movie or OTT platform movies rating.

core idea:
prediction:



OR



m_i = linear Regression

D_i = different data set

According to the above the ensemble learnig same data set with different models or different dataset with same model. Because here we used the concept wisdom of the crowd.

(i) for example in classification model:

cgpa	iq	placement
8	120	1
7	130	0

predict the for the input {8.1. 85}
placement is done or not.

let us, $\left\{\begin{array}{l} 3 \text{ model} \rightarrow 1 \\ 2 \text{ model} \rightarrow 0 \end{array}\right\}$

Hence placement is done.

(ii) for example in Regression model:

calculat/predict the LPA of the student

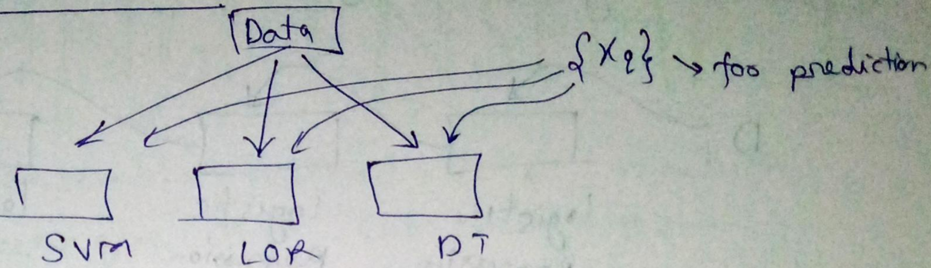
m_1	m_2	m_3	m_4	m_5
8.1	3.1	4.6	7.3	9.10

$$\text{mean LPA} = \frac{8.1+3.1+4.6+7.3+9.1}{5} =$$

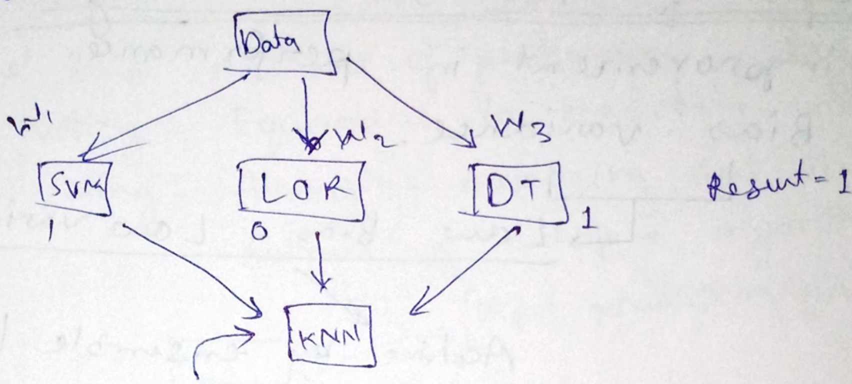
Types of Ensemble Learning and their Techniques:

Ensemble
- voting ensemble
- Cascading
- Bagging
  - Random Forest
- Boosting
  - adaboosting
  - Gradient Boosting
  - XgBoot
- Blending
- stacking

(I) voting ensemble: base models are different



(II) Stacking ensemble:

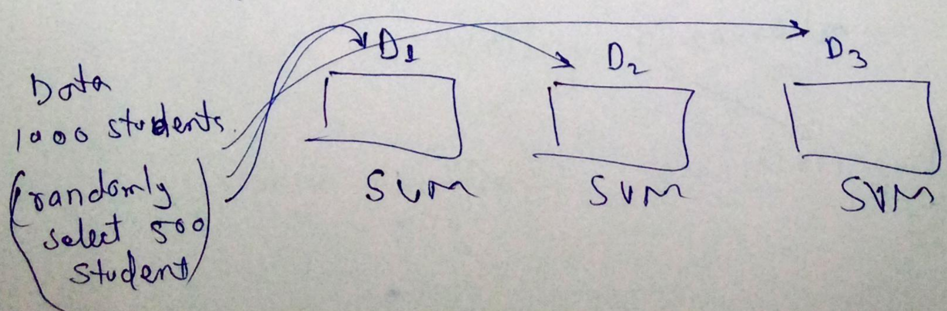


SVM	LOR	DT	R
1	0	0	1
0	0	1	1
0	0	0	0

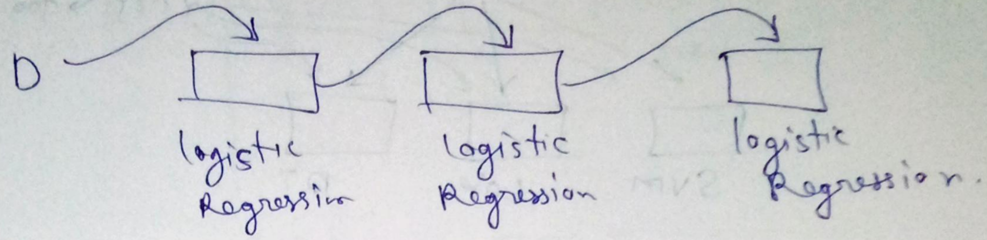
Data

मतलब Stacking ensemble में model के weightage एक decide होता है कि उसका contribution Next model में (KNN) कितना है.

(III) Bagging ensemble: Bootstrapped Aggregation
(same model with different data)



(IV) Boosting: Boosting our data.



Benefits of Ensemble Learning:

- improvement in performance
- Bias variance.
  → Low Bias + Low variance.
  Achive by ensemble learning
- Robusness

When to use Ensemble Learning:

- Always use it.
- Also use in ~~ensemble~~ kaggle. competition

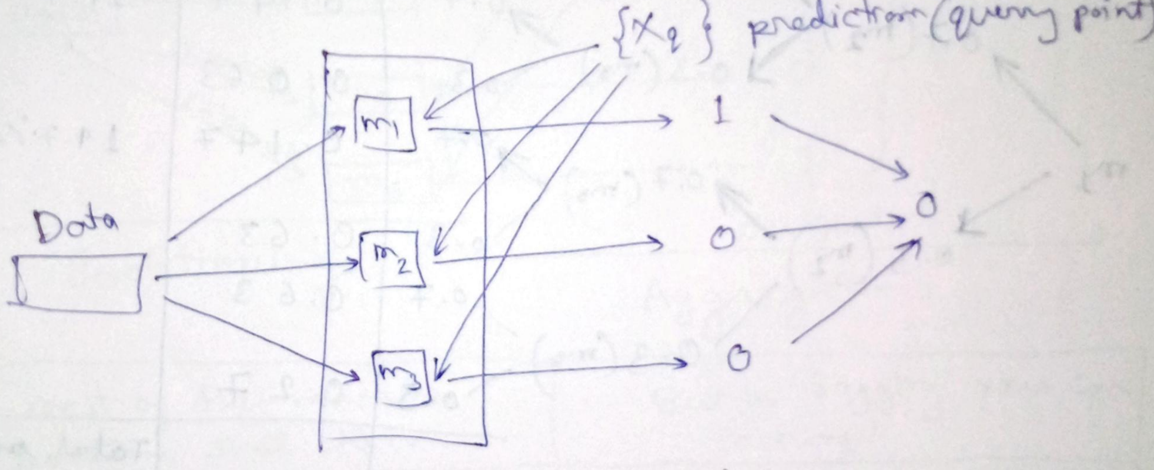
VOTING ENSEMBLE

The voting ensemble method is a machine learning technique that combines the prediction of multiple models. it can be used for classification and Regression.

There are two methods of performing voting:

(i) Hard voting: Equivalent to majority vote

(ii) Soft voting: ~~Averging~~ Avaraging out the output of multiple algorithms.



Assumption of voting ensemble:

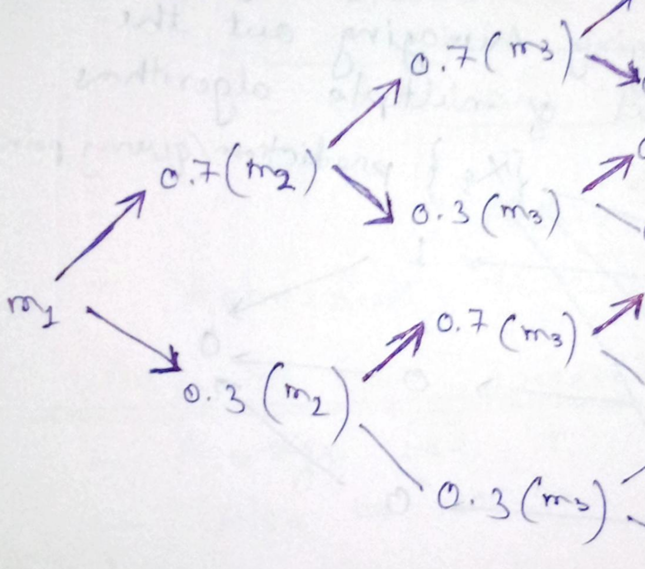
- all the models are independent
- The accuracy of each model must be greater then 50 %.

voting ensemble

voting classifier | voting Regresor

models

	m_1	m_2	m_3
probability of correct	0.7	0.7	0.7
probability of incorrect	0.3	0.3	0.3



		select majority
0.7	0.343	34.3%
0.3	0.147	14.7%
0.7	0.147	14.7%
0.3	0.063	
0.7	0.147	14.7%
0.3	0.63	
0.7	0.63	
0.3	0.27	
		Total accuracy 78% approx.

If the probability of correct is 0.3 for each model then total/overall model accuracy is

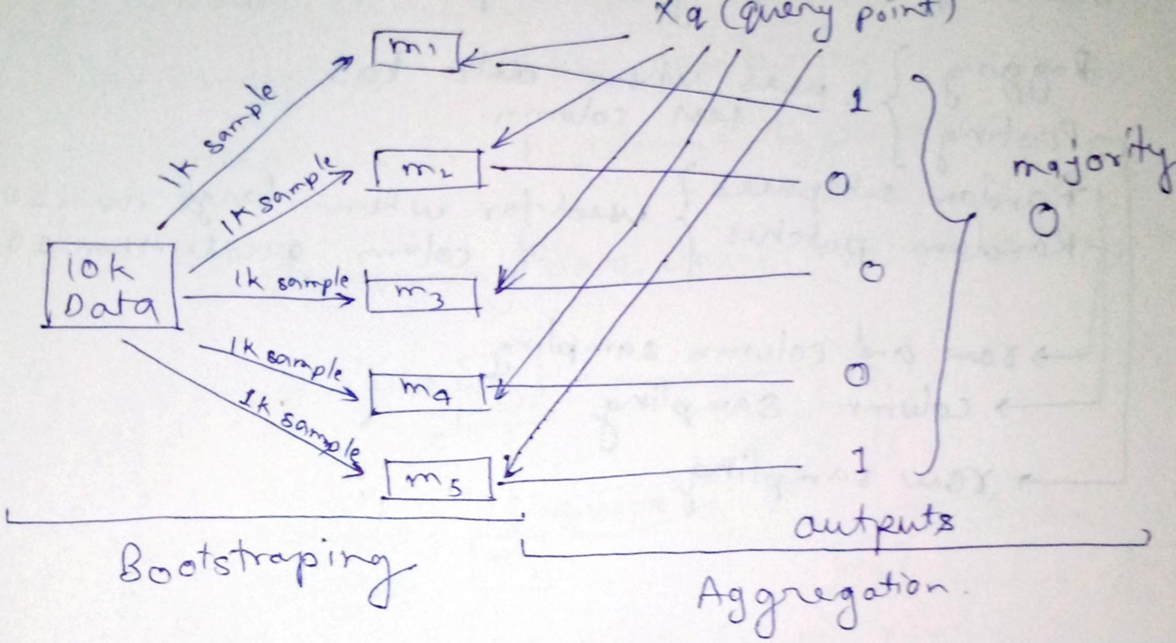
100 − 78 = 22%

Hence we say that the model accuracy for each model is greater than 50%.

BAGGING ENSEMBLE

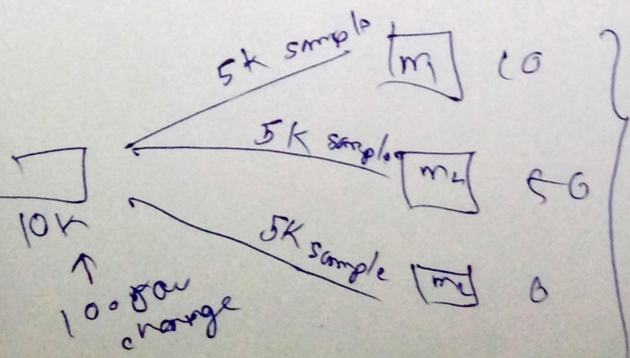
Bagging

Bootstraping   Aggregation.



most of the machine learning that has	but in Bagging you can achive
low bias, High varience High bias, Low variance.	Low Bias, High varience

→ DT, SVM, KNN (overfitting)



Low variance

hence we can say that low Bias, High varience (The out put is consistent) if aur model is ~~High~~ low bias . High varience then we can only use the Bagging Technique exampl is Random forest

~~Types of Bagging~~

~~Ensemble~~.

Bootstrap: sampling with replacement

(i) Bagging
(ii) Posting

uses when data has less column.

(iii) Random subspaces
(iv) Random patches

used for when large numbes of colum greater than 10

→ raw and column sampling

→ column sampling

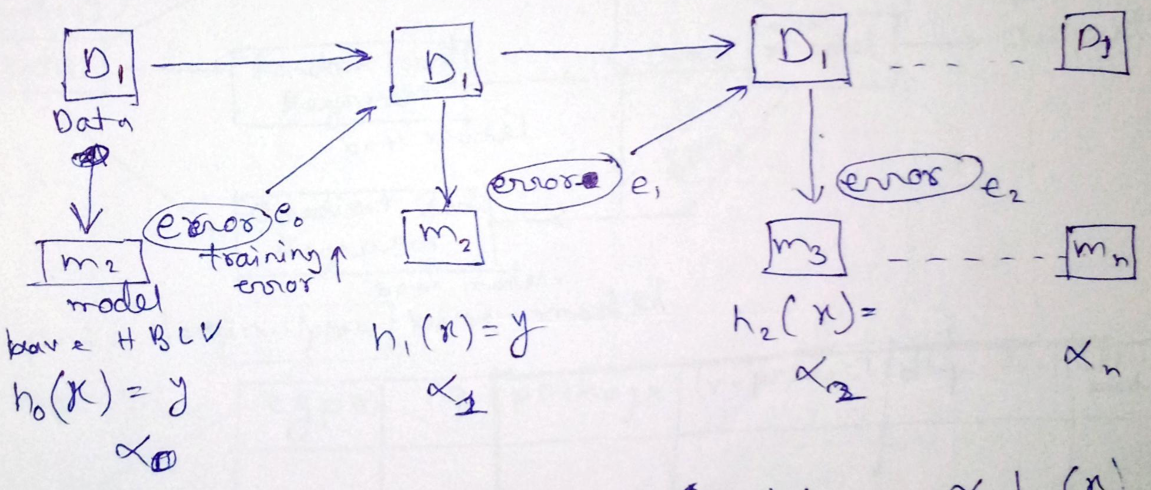
→ raw sampling

Boosting

* Bagging: convert to Low bias High variance (model) to Low bias Low variance (model)
* Boosting: convert the High bias low variance (model) to Low bias. Low variance (model)
  → concept use: additive combining

→ concept use is Bootstrom + Aggregation.

(x_i, y_i) training data. where input and output column



$$f = \alpha_0 h_0(x) + \alpha_1 h_1(x) + \dots \alpha_n h_n(x)$$

Boosting Techniques / Algorithm.

- Gradient Boosting
- Xgboot
- AdaBoot

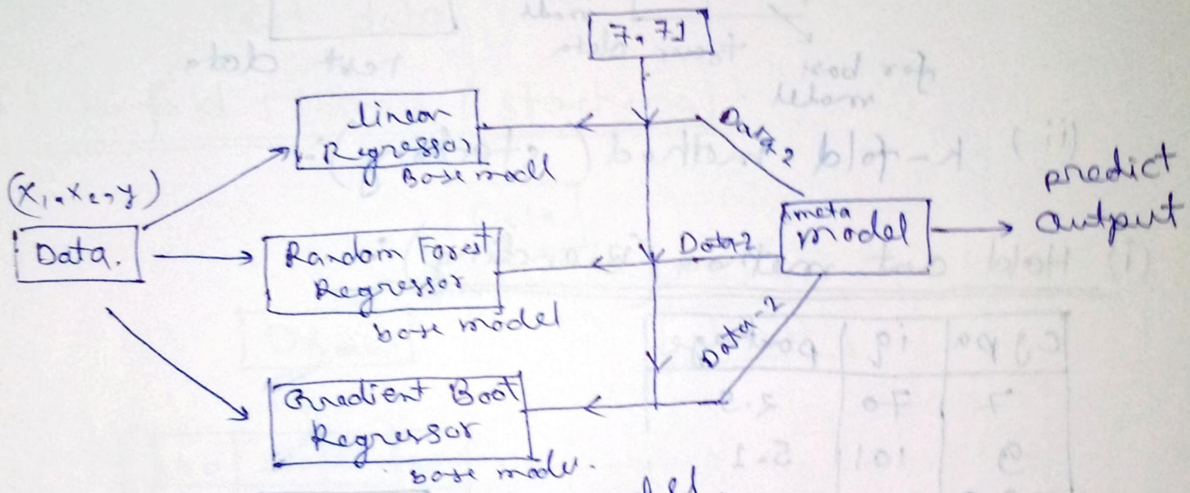
STACKING ENSEMBLE

- it is very similar to voting ensemble.
- we can also say that it is extended version of voting ensemble.
- let us take an example

cgpa	iq	package
7	75	3.5

1000 Raw

Data.1



step-1: train your base model.

step-2:

cgpa	iq	package	lr-predict	dt-predict	GBR predict

1000 Rows

Data-2.

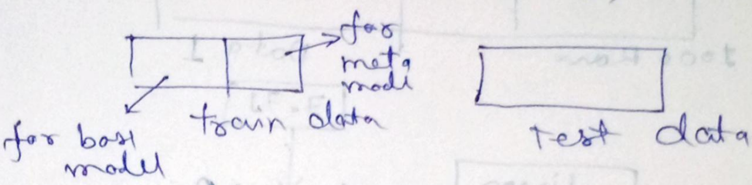
$\underbrace{\text{package}}_{\text{output}}$ $\underbrace{\text{lr-predict, dt-predict, GBR predict}}_{\text{input}}$

Step. 3 train the meta. model on new data (Data-2) then we get the final predicted output

The problem with stacking ensemble is our model is over fit because is for base model model we use same ~~for~~ data for training as well as prediction and data goes for meta model is also overfit hence our output is overfit.

to resolve this problem we have a two solution

(i) Hold out method (Blending):

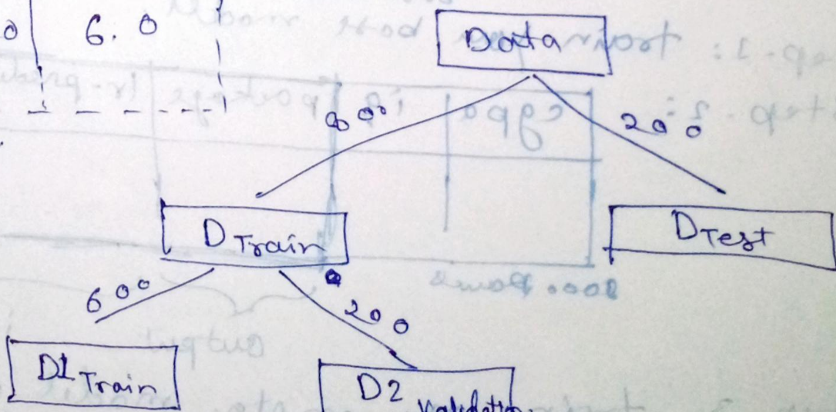


(ii) k-fold method (stacking):

(i) Hold out method (Blending):

cgpa	iq	package
7	70	2.5
9	101	5.1
7.90	120	6.0
...	...	...

1000 rows.



Step-1. train 3 base models using $D1_{Train}$ data.

Step. 2 you form a new data set.
calculate the prediction generated by your base model by using $D2_{validation}$ data set

output data

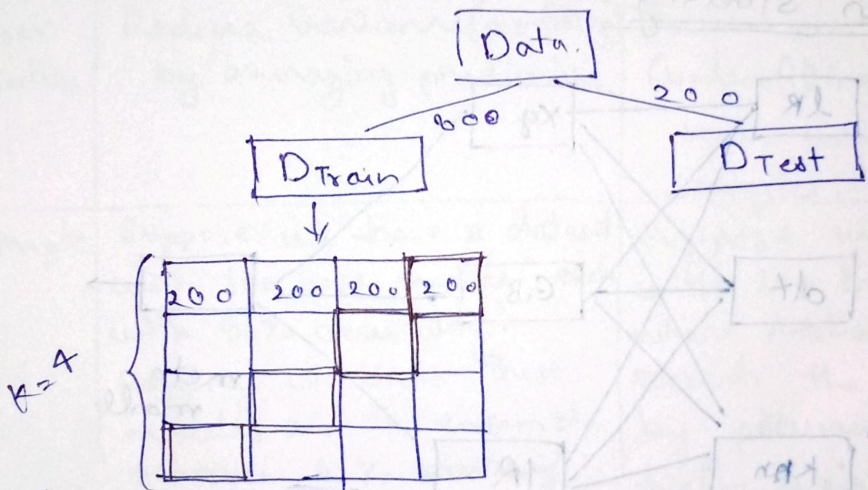
package	lr-predict	dt-pridict	knn-predict

(200, 4)
size of data.

Step-3. Train the meta model ~~on~~ using output data in the above.

Step-4. For the final prediction we use test data.

(ii) k-fold Method (stacking):



default k = 5

→ for train the our base model

→ for make prediction on base models (prediction for each base model)

output data.

package	lr-predict	dt-predict	knn-predict

800 rows

Step 1. defin k value.

Step-2.
Now train your meta model using output/predict data see in th above.

Step-3.
retrain the our ~~meta~~ base model on x-train, y-train for ~~each~~ base modll.

step-4. For the final prediction. use the D_{Test} (Data). on meta model.

(iii) Multi layer stacking: