CM2604 Machine Learning

Ensemble Techniques

Week 08 | Prasan Yapa

ROBERT GORDON UNIVERSITY ABERDEEN

INFORMATICS INSTITUTE OF TECHNOLOGY

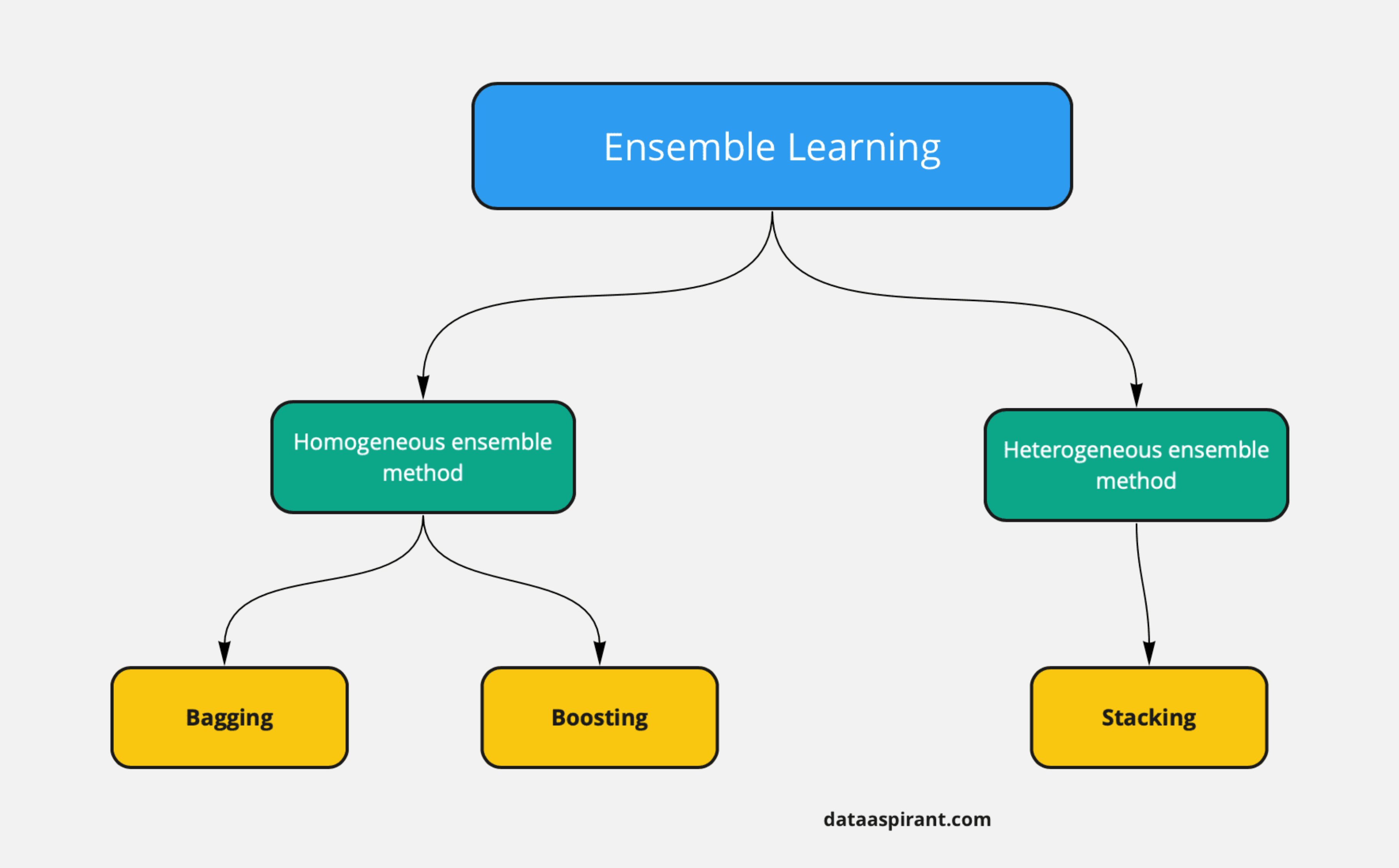
Content

- Ensemble methods in Machine Learning
- Ensemble types
- Bagging
- Boosting
- Bagging Vs Boosting
- Stacking

Ensemble Methods in Machine Learning

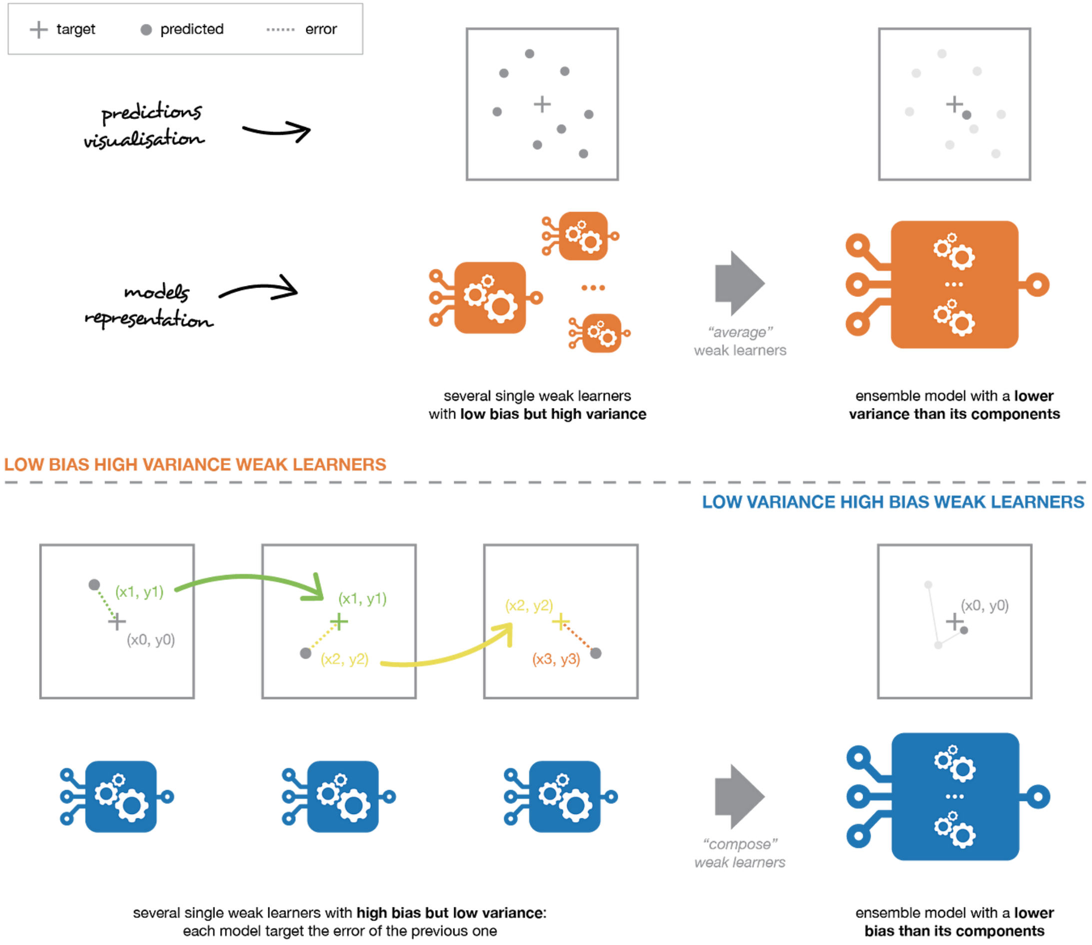
- Machine learning models are often trained with a variety of different methods in order to create a more accurate prediction.
- Ensemble methods are one way to do this and involve combining the predictions of several different models in order to get a more accurate result.
- It combines low performing classifiers and combine individual model prediction for the final prediction.
- Based on type of base learners, ensemble methods can be categorized as homogeneous and heterogeneous ensemble methods.

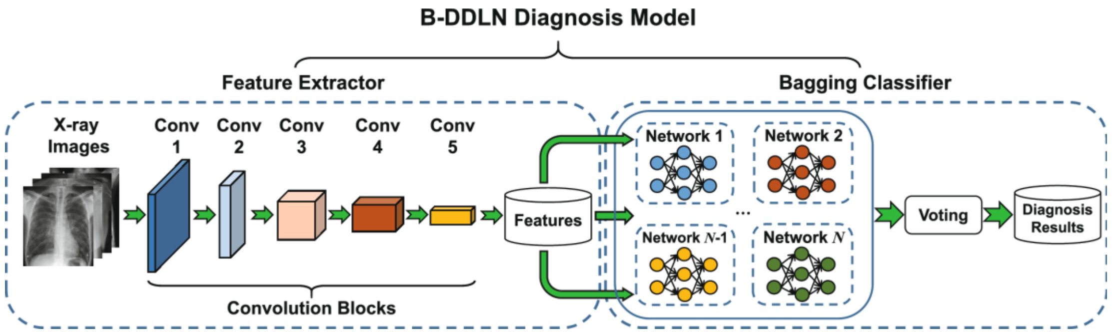
Ensemble Methods in Machine Learning



Ensemble Learning Methods

- Bagging
- Boosting
- Stacking

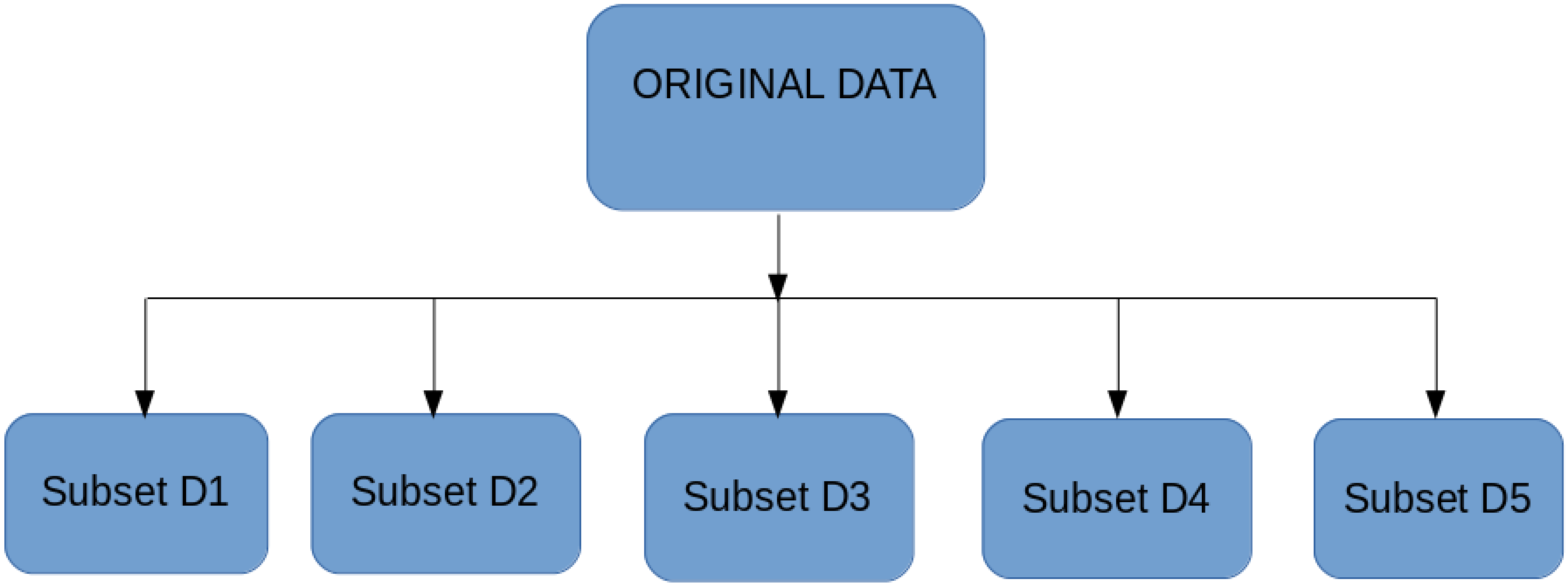




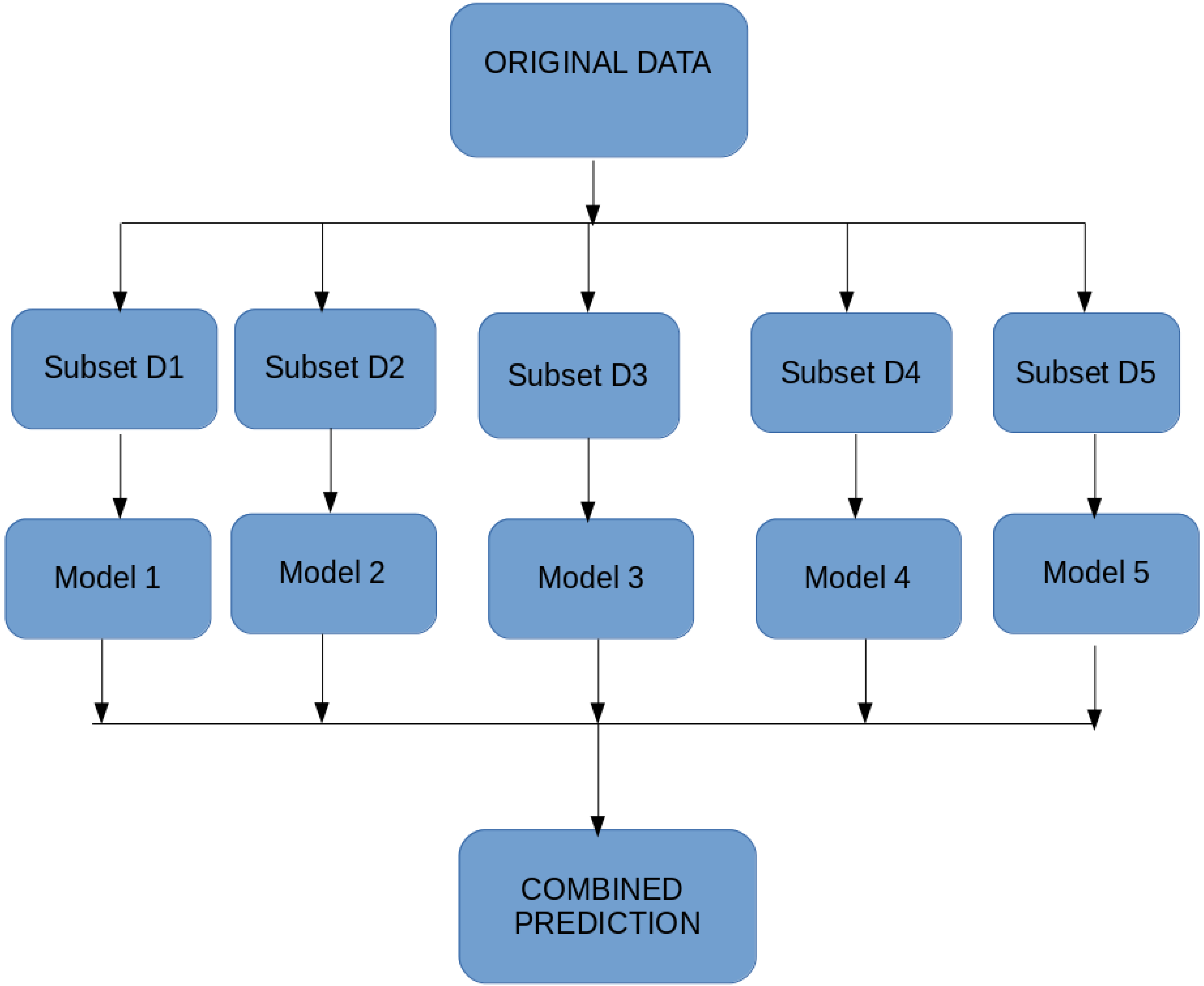
Bagging

- Bagging is an acronym for Bootstrapped Aggregation.
- Bootstrapping means random selection of records with replacement from the training dataset.
- The idea behind bagging is combining the results of multiple models to get a generalized result.
- Bagging technique works by creating several models, called “bags”, each of which is based on a different randomly-selected sample of the data.

Bagging


ORIGINAL DATA
Subset D1
Subset D2
Subset D3
Subset D4
Subset D5

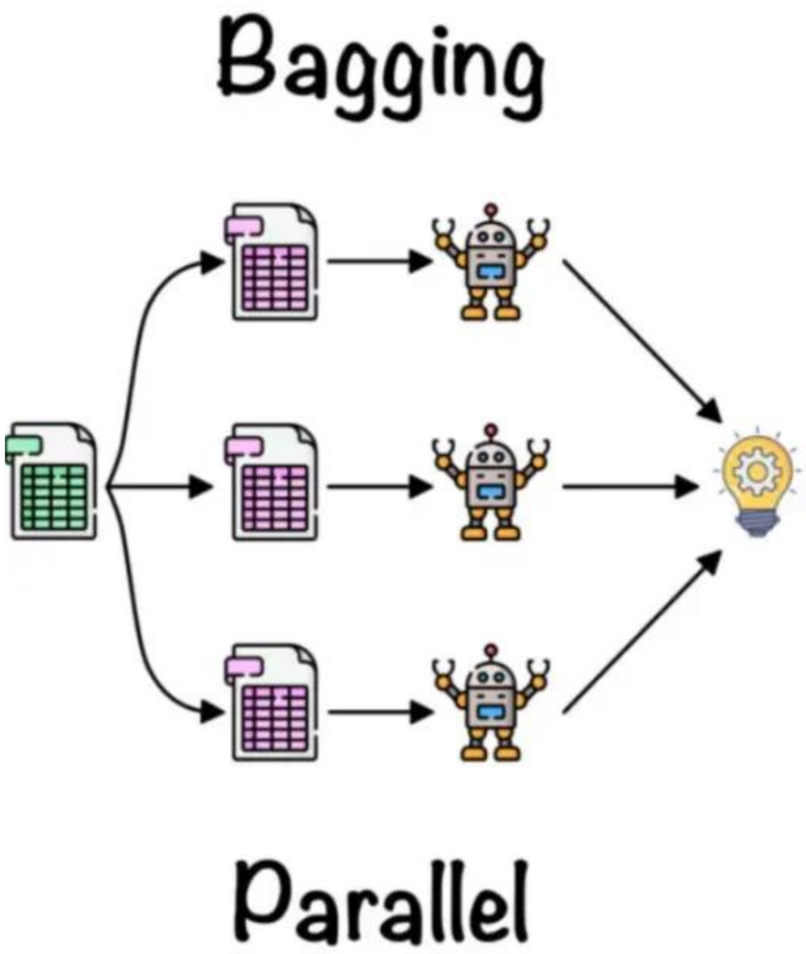
Bagging

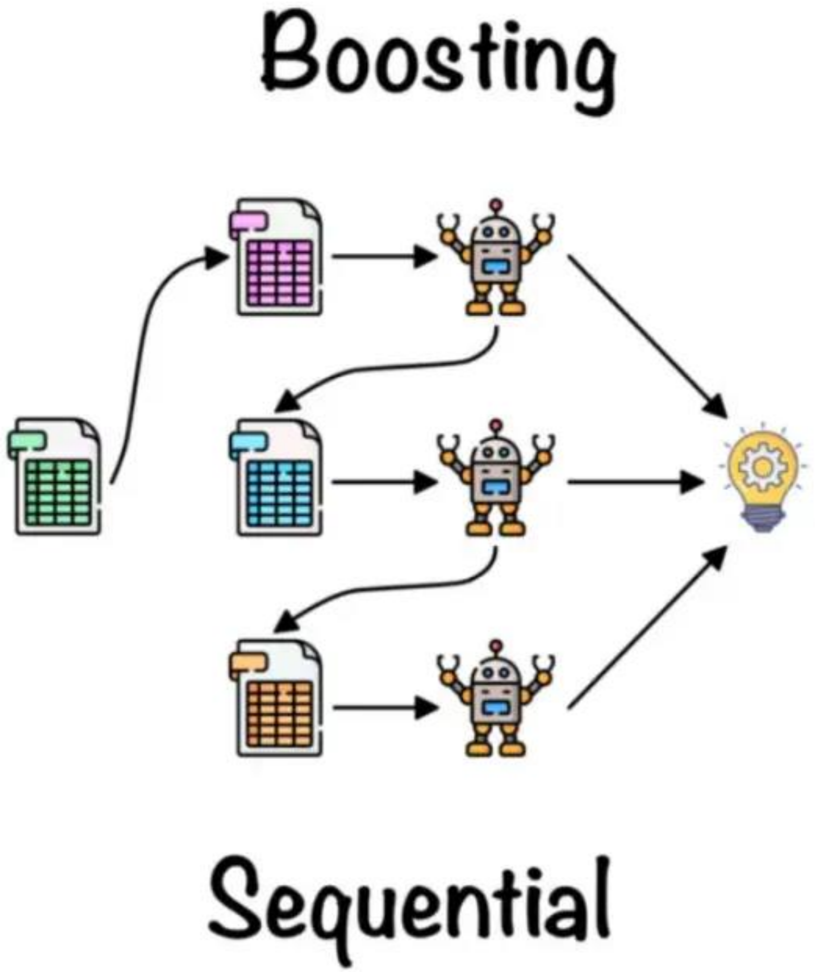

ORIGINAL DATA
Subset D1
Subset D2
Subset D3
Subset D4
Subset D5
Model 1
Model 2
Model 3
Model 4
Model 5
COMBINED PREDICTION

Boosting

- If a data point is incorrectly predicted by the first model, and then the next will combining the predictions provide better results? Such situations are taken care of by boosting.
- Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model.
- After classification, sample weights are changed. Weight of correctly classified sample is reduced, and weight of incorrectly classified sample is increased.
- Boosting can help data scientists adjust the weights of the models so that the better models have more influence on the final prediction.

Bagging Vs Boosting





Boosting

- Consider below training data for heart disease classification.

Blocked Arteries	Chest Pain	Weight	Heart Disease
Y	Y	200	Y
Y	N	185	Y
N	Y	200	Y
Y	Y	160	Y
Y	N	140	N
Y	N	130	N
N	Y	170	N
Y	Y	170	N

How Adaboost Algorithm Works?

- Step 1: Initialize Weights To All Training Points
- First step is to assign equal weights to all samples as all samples are equally important. Always, sum of weights of all samples equals 1.
- There are 8 samples, so each sample will get weight = 1/8 = 0.125
- Since all samples are equally important to start with, all samples get equal weight: 1 / total number of samples = 1/8

How Adaboost Algorithm Works?

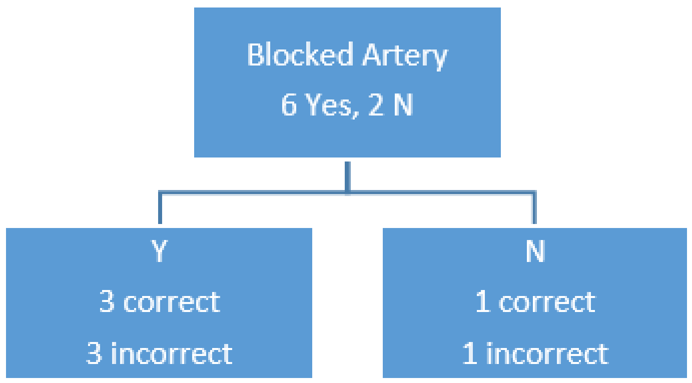
Blocked Arteries	Chest Pain	Weight	Heart Disease	Weights Assigned
Y	Y	200	Y	1/8
Y	N	185	Y	1/8
N	Y	200	Y	1/8
Y	Y	160	Y	1/8
Y	N	140	N	1/8
Y	N	130	N	1/8
N	Y	170	N	1/8
Y	Y	170	N	1/8

How Adaboost Algorithm Works?

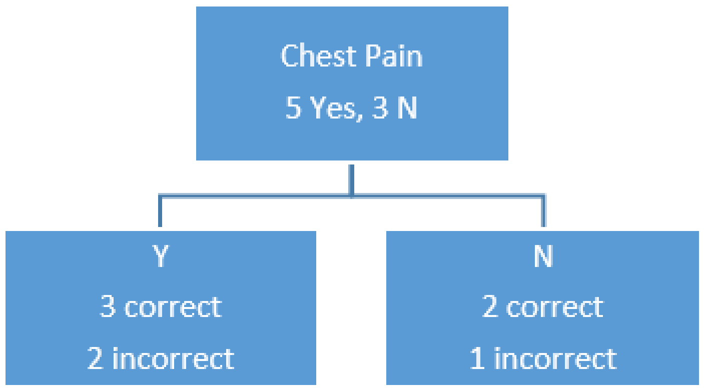
- Step 2: Create Stump
- Stump is a decision tree with one node and two leaves.
- To create stump, only one attribute should be chosen. But it is not randomly selected.
- The attribute that does the best job of classifying the sample is selected first.

How Adaboost Algorithm Works?

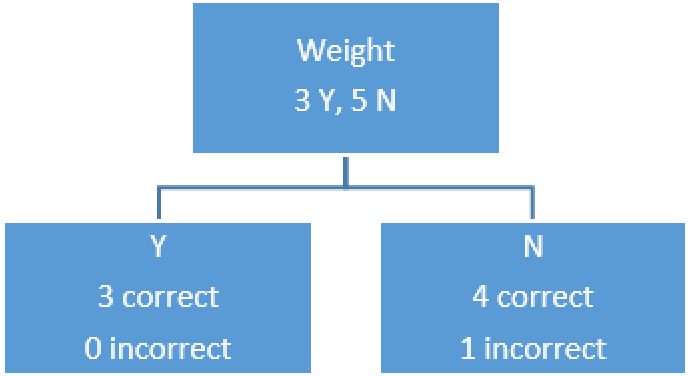
- <u>Blocked Artery:</u>



- <u>Chest Pain:</u>



How Adaboost Algorithm Works?



- <u>Weight:</u>
- Weight is continuous quantity.
- Set the threshold weight as 175.
- <u>Summary:</u>
- ‘Blocked Arteries’ as a stump made 4 errors.
- ‘Chest Pain’ as a stump made 3 errors.
- ‘Weight’ as a stump made 1 error.
- Since, the attribute ‘Weight’ made least number of errors, ‘Weight’ is the best stump to start with.

How Adaboost Algorithm Works?

- <u>Step 3: Calculate Total Error And Voting Power (Amount Of Say) Of The Stump</u>
- $Amount\ of\ say = \frac{1}{2} * \ln((1 - Total\ Error)/Total\ Error)$
- Where $\ln$ is Natural Log
- $Total\ Error = Number\ of\ misclassified\ samples * Weight\ of\ sample$
- Since the stump misclassified only 1 sample, Total Error made by stump = weight of 1 sample = 1/8 = 0.125
- Substituting Total Error in the equation:
- $Amount\ of\ say\ =\ ½\ *\ \ln((1 - 0.125)/0.125)$ = 0.9729

How Adaboost Algorithm Works?

- Step 4: Start To Build Final Classifier Function
- The Adaboost classifier then starts building a function that will assign this 'amount of say' as voting power of the stump.
- It will assign 0.9729 as voting power to the stump created for 'Weight' attribute.
- The function $h(x) = 0.9729 * \text{'}Weight\text{'}$ as stump.

How Adaboost Algorithm Works?

- Step 5: Update New Weights For The Classifier
- After calculation of amount of say, next step is to create another stump.
- However, before that adjust the weights assigned to each sample.
- The samples that were correctly classified in previous stump should get less importance as their classification is over, and the samples that were not correctly classified should get more weight in the next step.

How Adaboost Algorithm Works?

- The formula to assign new weight for correctly classified samples is:
- $W_{new} = \frac{W_{old}}{2(1-Total\,Weight)}$ = (0.128) / 2(1-0.125) = 0.07
- The formula to assign new weight for incorrectly classified samples is:
- $W_{new} = \frac{W_{old}}{2*Total\,Weight}$ = (0.125) / (2*0.125) = 0.5

How Adaboost Algorithm Works?

Blocked Arteries	Chest Pain	Weight	Heart Disease	Weights Assigned
Y	Y	200	Y	0.07
Y	N	185	Y	0.07
N	Y	200	Y	0.07
Y	Y	160	Y	0.5
Y	N	140	N	0.07
Y	N	130	N	0.07
N	Y	170	N	0.07
Y	Y	170	N	0.07

How Adaboost Algorithm Works?

- From above classification, it can be seen that:
  - ‘Blocked Arteries’ as a stump made 4 errors. Therefore, Total Error for ‘Blocked Arteries’ = 0.07*4 = 0.28
  - ‘Chest Pain’ as a stump made 3 errors. Therefore, Total Error for ‘Chest Pain’ = 0.07*3 = 0.21
  - ‘Weight’ as a stump made 1 error. Therefore, Total error for weight = 1*0.5 = 0.5.
- We have already considered ‘Weight’ attribute for first stump creation. So, this attribute need not be considered again.

How Adaboost Algorithm Works?

- Step 6: Calculate Voting Power (Amount of Say) and Add to Final Classifier Function
- Since Chest Pain classifier has least Total Error, this attribute is selected as second stump.
- The Amount of Say = ½ $\ln(1 - 0.21)/0.21 \;= 0.66$
- Add this classifier with its weight to the function.
- The function $h(x) \;=\; 0.9729 *$ '$Weight$' as stump $+ \; 0.66 *$ '$Chest\ Pain$' as stump

How Adaboost Algorithm Works?

- <u>Step 7: Assign New Weights and Repeat</u>
- Again, assign new weights to the correctly classified samples with formula:
- $W_{new} = \frac{w_{old}}{2(1-Total\ Weight)}$
- And incorrectly classified samples with the formula:
- $W_{new} = \frac{w_{old}}{2*Total\ Weight}$
- We repeat this process until
  - Enough rounds of this process are done to create strong classifier
  - No good classifiers left for making predictions.

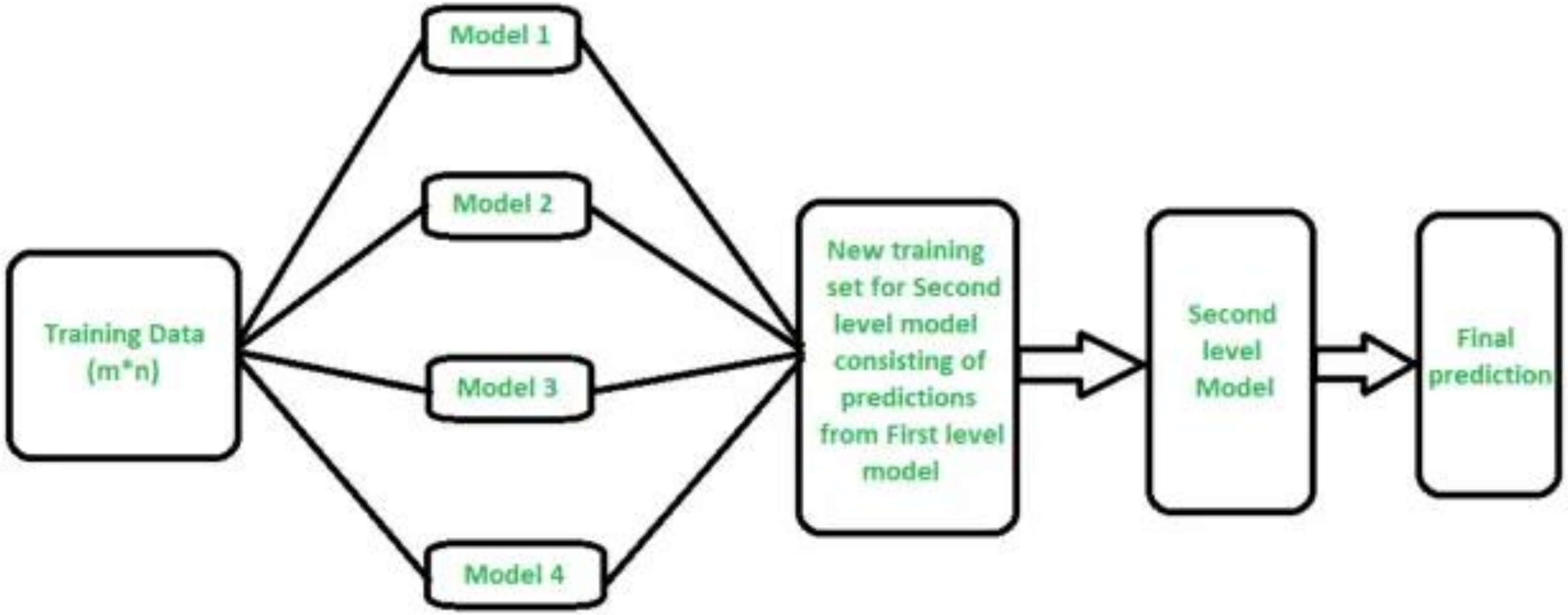
Boosting

- Summary:
  - Adaboost is a sequential method where one weak learner runs after other.
  - Weak learners are stumps i.e., decision tree with 2 nodes.
  - Voting power is assigned to stumps based on how correctly it classifies the data.
  - Boosting reduces bias
- Apart from Adaboost, there is other boosting algorithm – Gradient Boosting or G
- The main idea is to create decision trees with smaller leaves and scale the tree with a learning rate.

Stacking

- Stacking uses predictions from multiple models (for example decision tree, KNN or SVM) to build a new model.
- The models are “stacked” on top of each other, and the predictions from each model are combined to produce a final prediction.
- This can be used for regression or classification tasks, and it has been shown to outperform traditional machine learning methods.
- The ultimate model is powerful because it can combine the strengths of different models to produce a more accurate prediction.
- Overall, it is difficult to implement, and it requires a great deal of tuning to get the best results.

Stacking


Training Data
(m*n)
Model 1
Model 2
Model 3
Model 4
New training set for Second level model consisting of predictions from First level model
Second level Model
Final prediction

Stacking

- The stacking ensemble method can be used for solving a variety of machine learning problems, such as regression and classification.
- For example, imagine that you are trying to predict the price of a house.
- You could use a model to predict the price of a house based on its size and location, and then use a second model to predict the price of a house based on its age and condition.
- The predictions from each model would be combined to produce a final prediction for the price of the house.

Questions