Tree Based Models

Ensemble Models

- An ensemble is a composite model, combining a series of low performing models (based models) with the aim of creating an improved classifier.
- The based model in ensemble models are usually decision trees

Ensemble Models

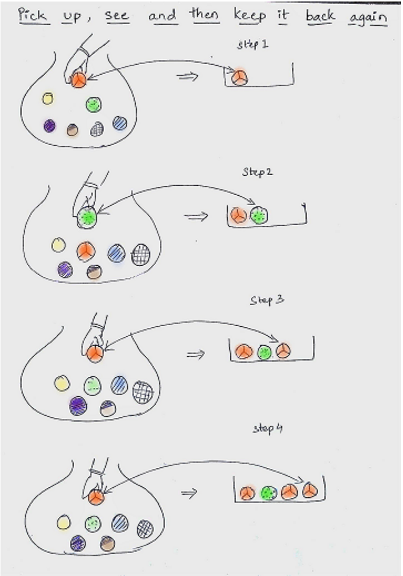
- Two common ensembles:
  - Bagging
  - Boosting

Bagging

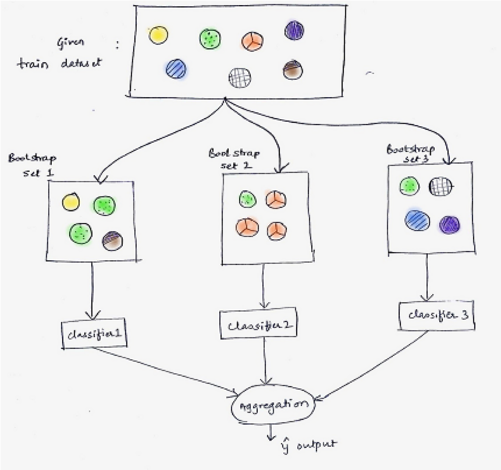
Bagging

- Step 1: From the original Dataset create k boostrap dataset (Boostrap sample)
- Step 2: Train k models (decision trees for example) on the k boostrap sample
- Step 3: After training, use the k models to make k predictions
- Step 4: The final prediction is
  - the majority vote of the k predictions in Step 3 for a categorical target or
  - the average of the k predictions for a continuous target.

How to make a boostrap sample?



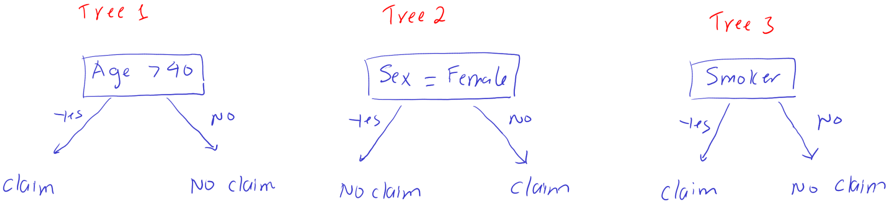
Bagging



Given train dataset :
Bootstrap set 1
Bootstrap set 2
Bootstrap set 3
classifier 1
classifier 2
classifier 3
Aggregation
ŷ output

Example 1: Classification

A bagging model uses three boostraping samples to train a decision with one split (called a stump). The response variable is whether a customer make a claim on a policy. The three trees after training are as follows.



- Use the bagging model to predict if a 30 year-old male customer who is a non-smoker would make claim on the policy. The customer does not have children.

Solution

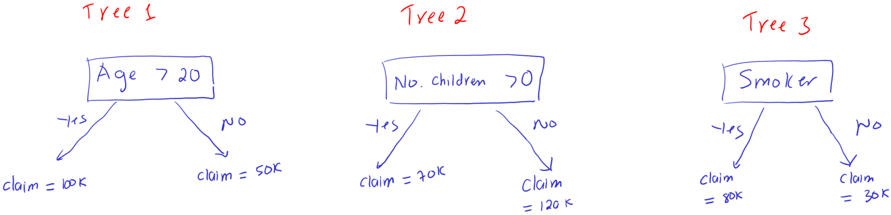
The predictions of the three trees on the customer as follows. Notice that the customer is a 30 year-old male, who is a non-smoker and does not have children.

- Tree 1 predicts: Claim
- Tree 2 predicts: Claim
- Tree 3 predicts: No Claim

The final prediction is the majority vote between the three trees. Thus the final prediction of the bagging model is: Claim, or the customer would make a claim on the policy.

Example 2: Regression

A bagging model uses three boostraping samples to train a decision with one split (called a stump). The response variable is the claim amount of the customer on a policy. The three trees after training are as follows.



- Use the bagging model to predict the claim amount of a 30 year-old male customer who is a smoker. The customer does not have children.

Solution

The predictions of the three trees on the customer as follows. Notice that the customer is a 30 year-old male, who is a smoker and does not have children.

- Tree 1 predicts: Claim amount of 100k
- Tree 2 predicts: Claim amount of 120k
- Tree 3 predicts: Claim amount of 80k

The final prediction is the average of the three predictions. Thus the final prediction of the bagging model is:
$(100k + 120k + 80k)/3 = 100k$.

Random Forest

- A random forest is a bagging model that used decision trees as the based model
- When training trees, at each split, only a random subset of k variables are considered to decide the best split
- The smaller the k value, the more diverse the forest

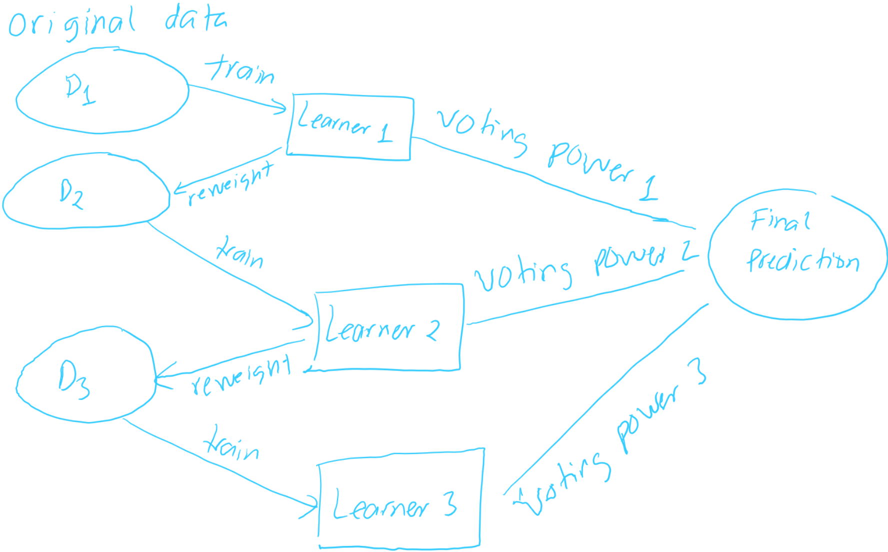
Boosting

The idea

- Train a weak model
- Update the data to address the model's mistakes
- Retrain the model
- Repeat the process

- Train Model A, usually weak model, on the original dataset (D1)
- Obtain Trained Model A - Version 1 (Learner 1)
- Calculate the error of the above model
- Update the Dataset 1 (D1) to Dataset 2 (D2) to `emphasize` the errors
- Train Model A again on Dataset 2 (D2)
- Obtain Trained Model A - Version 2 (Learner 2)
- Calculate the error of the above model
- Update the Dataset 2 (D2) to Dataset 3 (D3) to `emphasize` the errors
- Train Model A again on Dataset 3 (D3)
- Obtain Trained Model A - Version 3 (Learner 3)
- And so on.
- All the learners then called to vote to make the final prediction

Boosting



Original data
D_1
train
Learner 1
voting power 1
D_2
reweight
train
Learner 2
voting power 2
D_3
reweight
train
Learner 3
voting power 3
Final prediction

Boosting

- Different boosting models have different ways to update the data to emphasize the errors
- Some popular boosting models: Gradient Boosting, Adaboost

Gradient Boosting

Gradient Boosting

- Update the data by replacing the original response by the error of the previous model

Gradient Boosting

- Train a weak model on the original data (response variable: y)

x_1	x_2	y
...	...	...

- Calculate the error $\epsilon = y - \hat{y}$
- Retrain model A on this below data (response variable: ϵ)

x_1	x_2	ϵ
...	...	...

- Repeat the process
- Aggregate all the model's prediction to make the final prediction

Example 3 - Gradient Boosting Calculation

Click Here

Adaboost

Adaboost

- Update the data by adding more copies of the observations that the previous model predicts wrongly.

Adaboost

Idea Behind Ada Boost

- Examples of high weight are shown more often at later rounds
- Face/nonface classification problem:

Round 1

	1/7	1/7	1/7	1/7	1/7	1/7	1/7
best weak classifier:	✓	✗	✓	✓	✗	✓	✗
change weights:	1/16	1/4	1/16	1/16	1/4	1/16	1/4

Round 2

best weak classifier:	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
change weights:		1/8	1/32	11/32		1/2		1/8	1/32	1/32

Adaboost

Idea Behind Ada Boost

Round 3

- out of all available weak classifiers, we choose the one that works best on the data we have at round 3
- we assume there is always a weak classifier better than random (better than 50% error)
-  image is half of the data given to the classifier
- chosen weak classifier **has to** classify this image correctly