Ensemble Learning with Random Forest

Session # 3

Preview of last session

- Decision tree intuition
- Growing decision trees
    - Splitting criteria: Entropy/variance in reduction
    - For continuous variable: chose a threshold to split
- Rpart implementation of trees on a dataset
- Accuracy, Precision and Recall of the model
- Confusion Matrix construction
- **Assignment**
    - **Loss_Matrix** to emphasize on business aspect of the problem.
    - Parameters of rpart

Agenda for today

- An example of decision tree working to clear up gini index calculation
- Introduction to the idea of ensemble learning
- Different types of ensemble
- Random forest as bagged decision trees
- How random forests are build
- R implementation with **ranger**

Decision tree working example (solve on copy)

Refund	Marital Status	Cheat
Yes	Single	No
No	Married	No
No	Single	No
Yes	Married	No
No	Divorced	Yes
No	Married	No
Yes	Divorced	No
No	Single	Yes
No	Married	No
No	Single	Yes

Label/Target

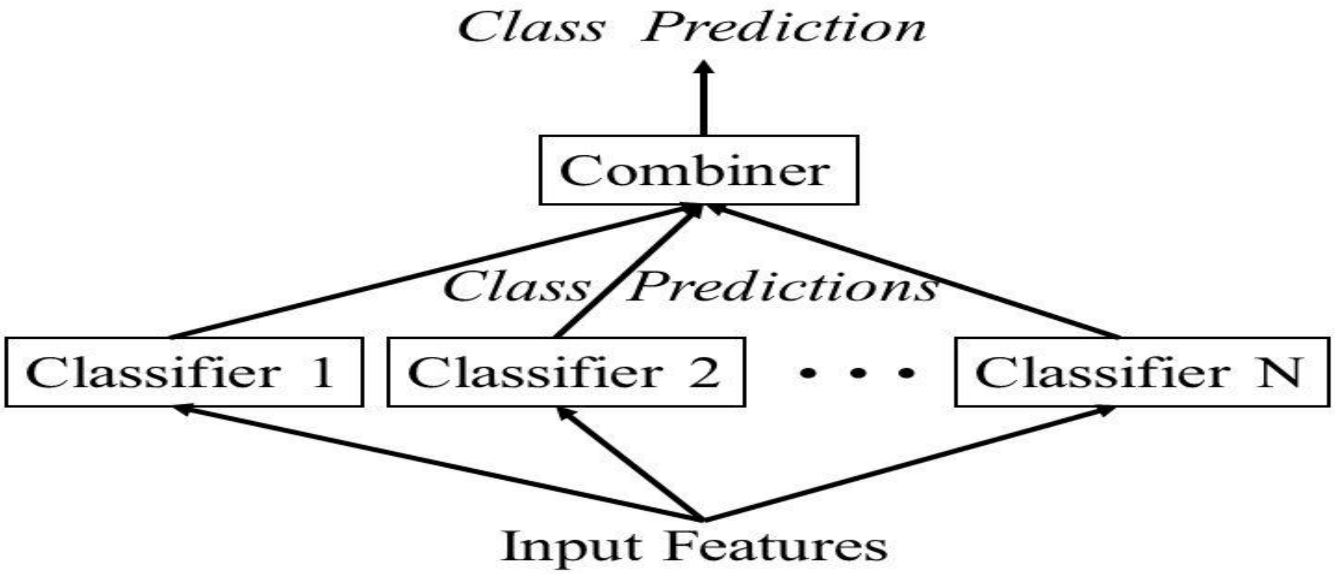
Ensemble Learning

- Combining decisions from multiple models to improve the overall performance.

- A crowd of experts is better than one expert. One model alone can suffer from several issues, which can lead to different issues.

- It improves variance and bias in a desirable way.

A Classifier Ensemble



Ways of Combining results

- Max voting: Train **n** different models on the data. Take majority vote on test sample. Quite common in **classification** setting.

- Average Voting: Train **n** different models on the data. Take average of all predictions from different models.

- Weighted Average: Same as average voting, except for the weight assigned to each model. Each model is given a weight according to its importance.

Types of ensemble in terms of execution

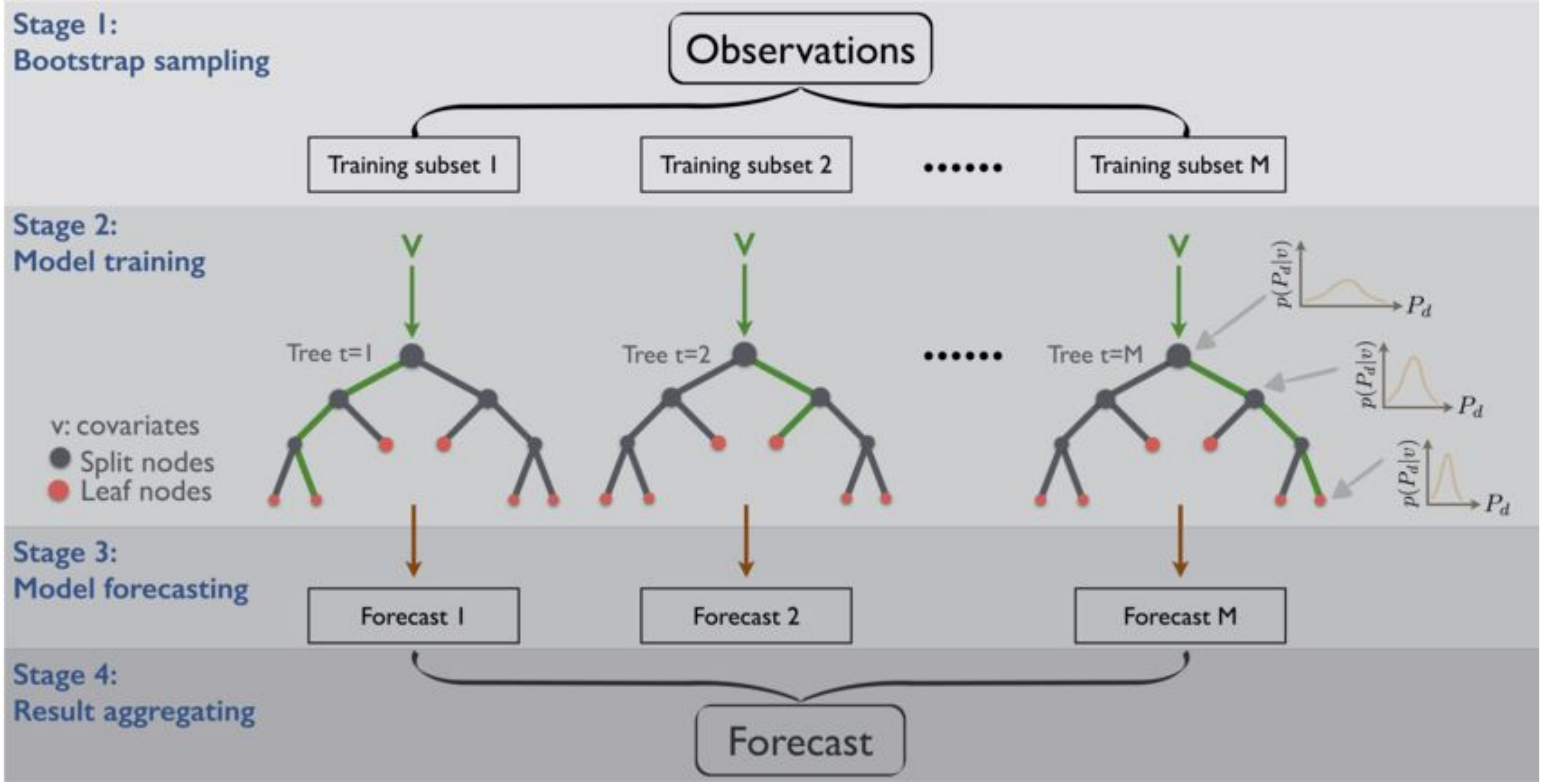
- Sequential Ensembles: Base classifiers are trained sequentially. Purpose is to exploit the dependence between classifiers.
  - Example: Adaboost, Stacking etc

- Parallel Ensembles: Base classifiers are trained parallel and their results are then combined as mentioned before. Purpose is to exploit the independence between classifiers. Classifiers' predictions **must be** decorrelated.
  - Example: Random Forest (topic of today's class)

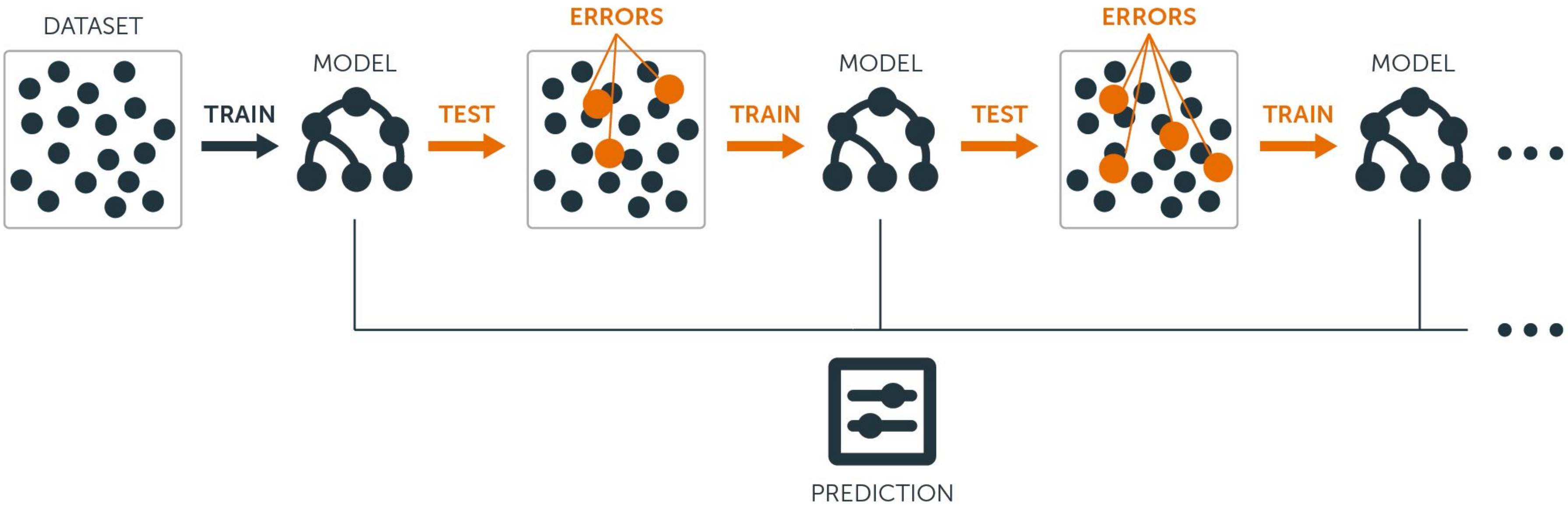
Types of ensembling in terms of training

- Bagging: stands for Bootstrap aggregation. Idea is to have **bootstrapped samples** of the data, train different models on each sample and then aggregate the results
- Boosting: It happens sequentially. Each subsequent model attempts to correct the errors of the previous model. The succeeding models are dependent on the previous model.
- Stacking: Train multiple models on same data. Use predictions of these base models as input features for a final model.

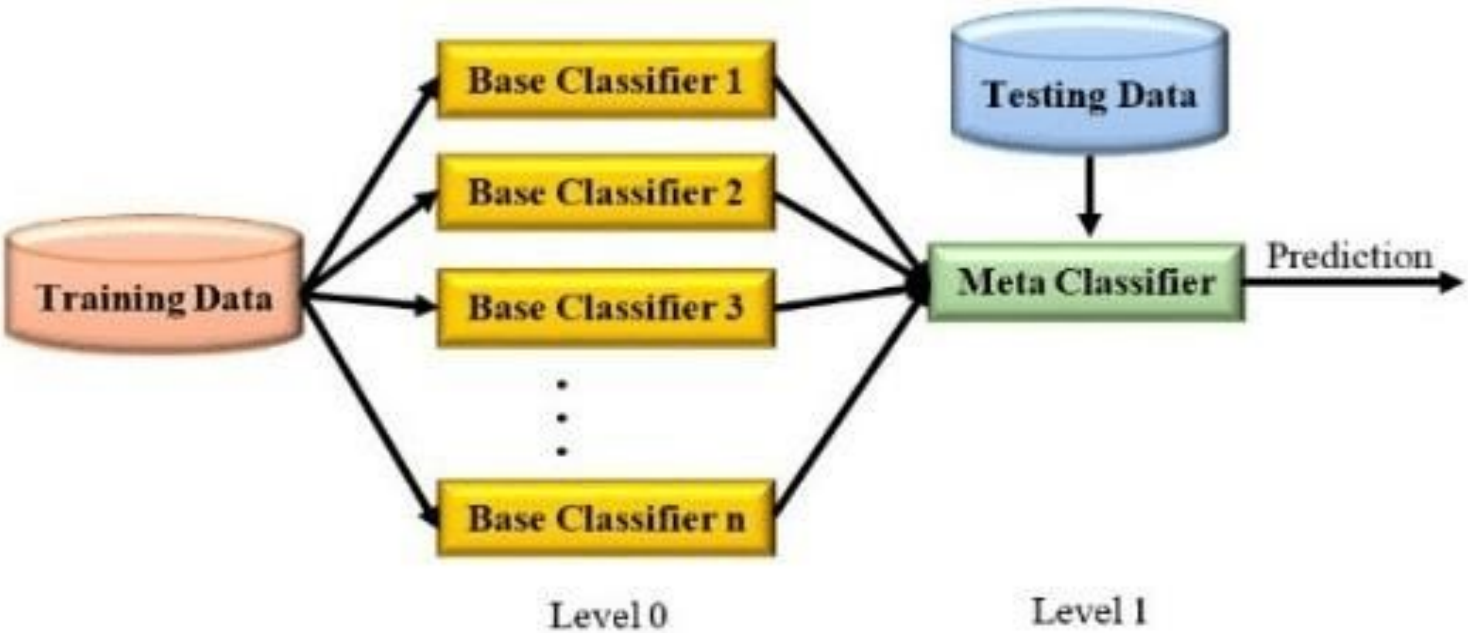
Bagging


Stage 1:
Bootstrap sampling
Observations
Training subset 1
Training subset 2
Training subset M
Stage 2:
Model training
v
Tree t=1
Tree t=2
Tree t=M
v: covariates
Split nodes
Leaf nodes
$p(P_d|v)$
P_d
Stage 3:
Model forecasting
Forecast 1
Forecast 2
Forecast M
Stage 4:
Result aggregating
Forecast

Boosting


DATASET
TRAIN
MODEL
TEST
ERRORS
TRAIN
MODEL
TEST
ERRORS
TRAIN
MODEL
PREDICTION

Stacking


Training Data
Base Classifier 1
Base Classifier 2
Base Classifier 3
Base Classifier n
Testing Data
Meta Classifier
Prediction
Level 0
Level 1

Random Forests
(bagged decision trees)

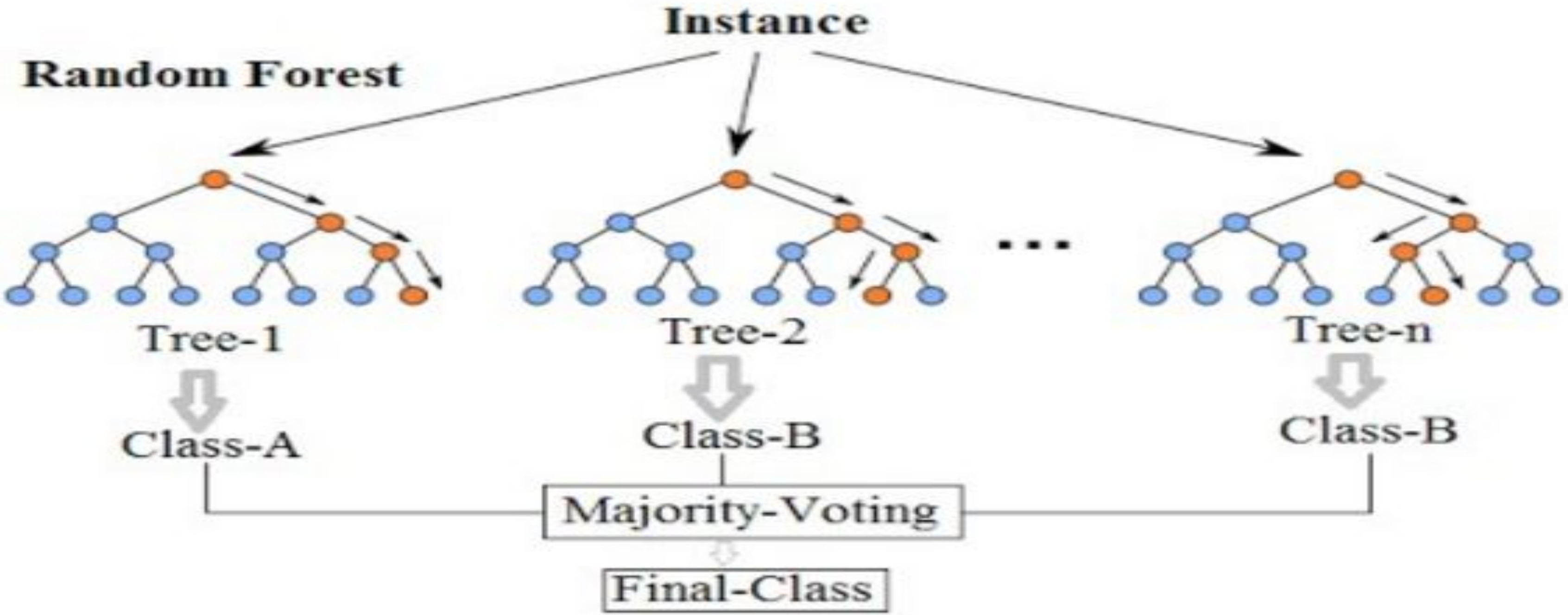
Introduction to random forests

- Inspired by the idea of bagging, which is a type of ensemble learning

- Instead of relying on one tree, we generate different trees for the problem

- Predictions are then made based on majority voting (in classification) and average/mean (in regression)

Random Forest Simplified



How random forest is build?

- As in bagging, we bootstrap samples from data. For generating n trees, we sample n times from the data

- We then build n different trees, but with selecting **some** input features/variables/predictors at random (this selection of input features makes it different than traditional bagging)

- Prediction on test set is then done by majority voting and mean of output

Given a training set S

For i = 1 to k do:

Build subset Si by sampling with replacement from S

Learn tree Ti from Si

At each node:

Choose best split from random subset of F features

Each tree grows to the largest extend, and no pruning

Make predictions according to majority vote of the set of k trees.

Decision Trees

- More interpretable
- More fast
- Higher variance

Random Forest

- Less interpretable
- Computationally much slower than single tree
- Lower variance
- Usually, gives more accuracy than single tree