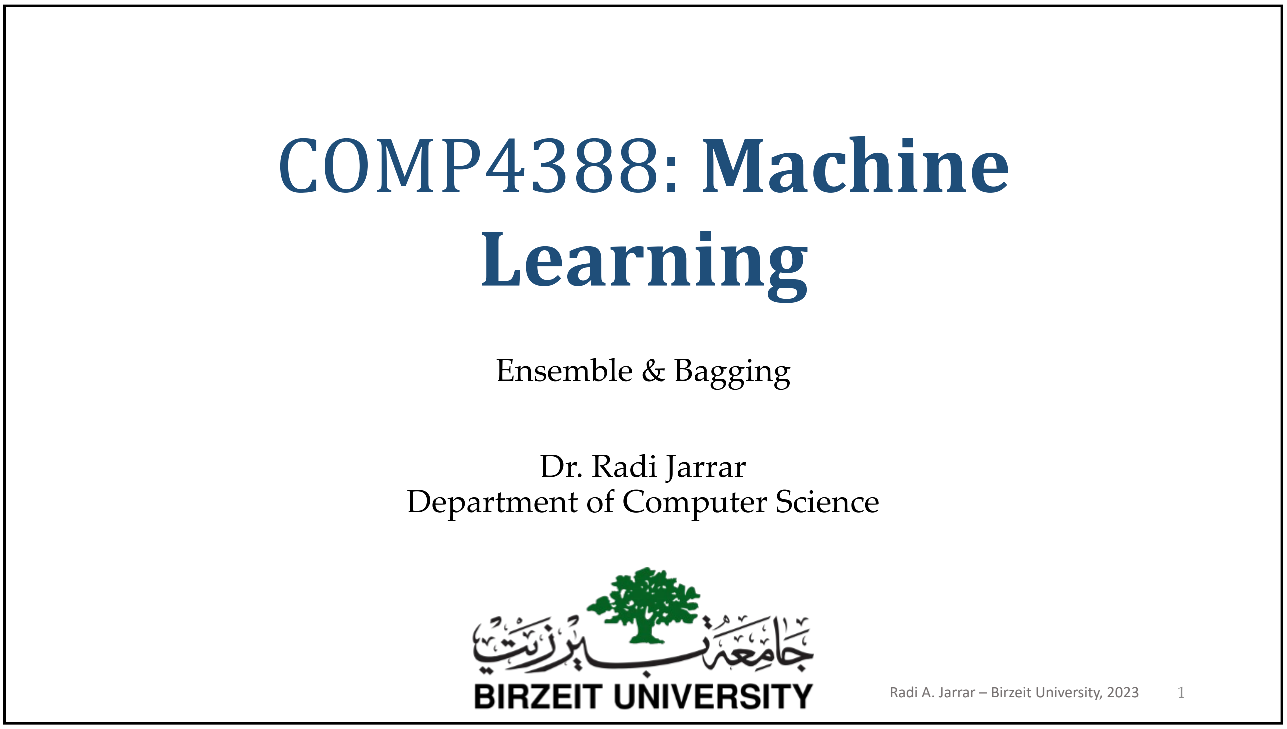
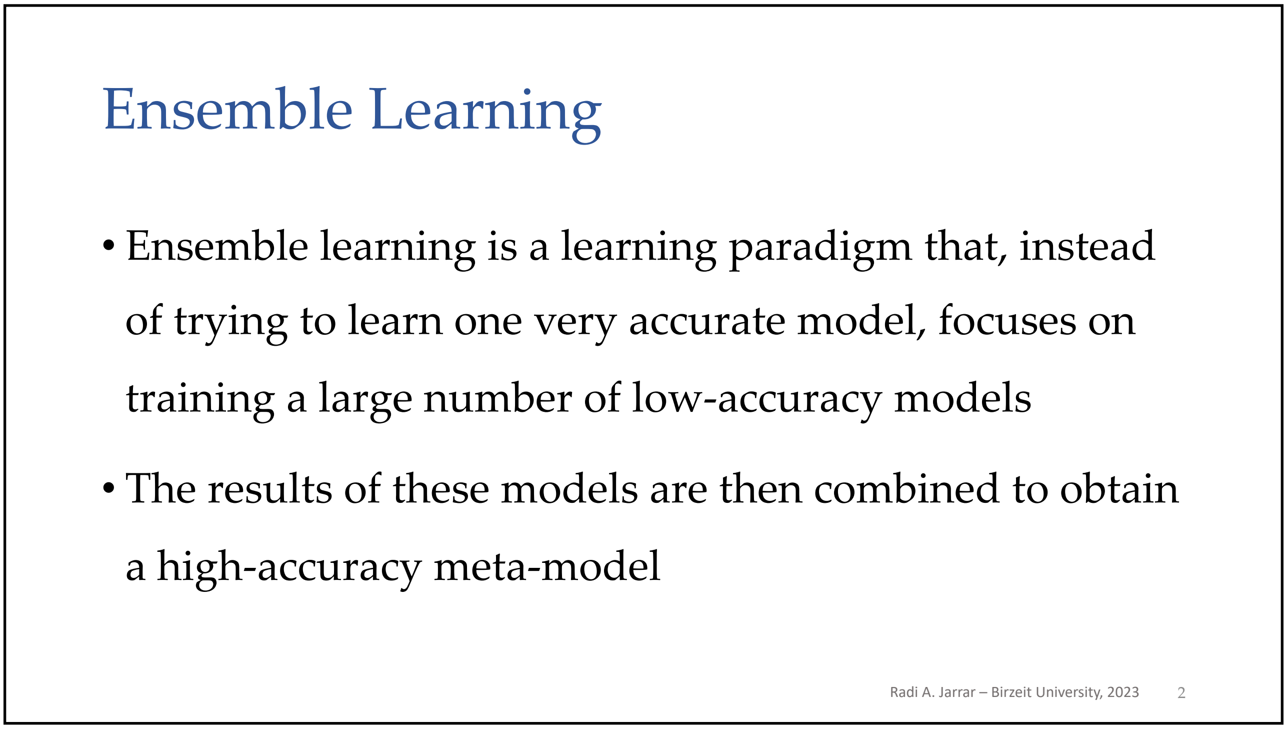


COMP4388: **Machine Learning**

Ensemble & Bagging

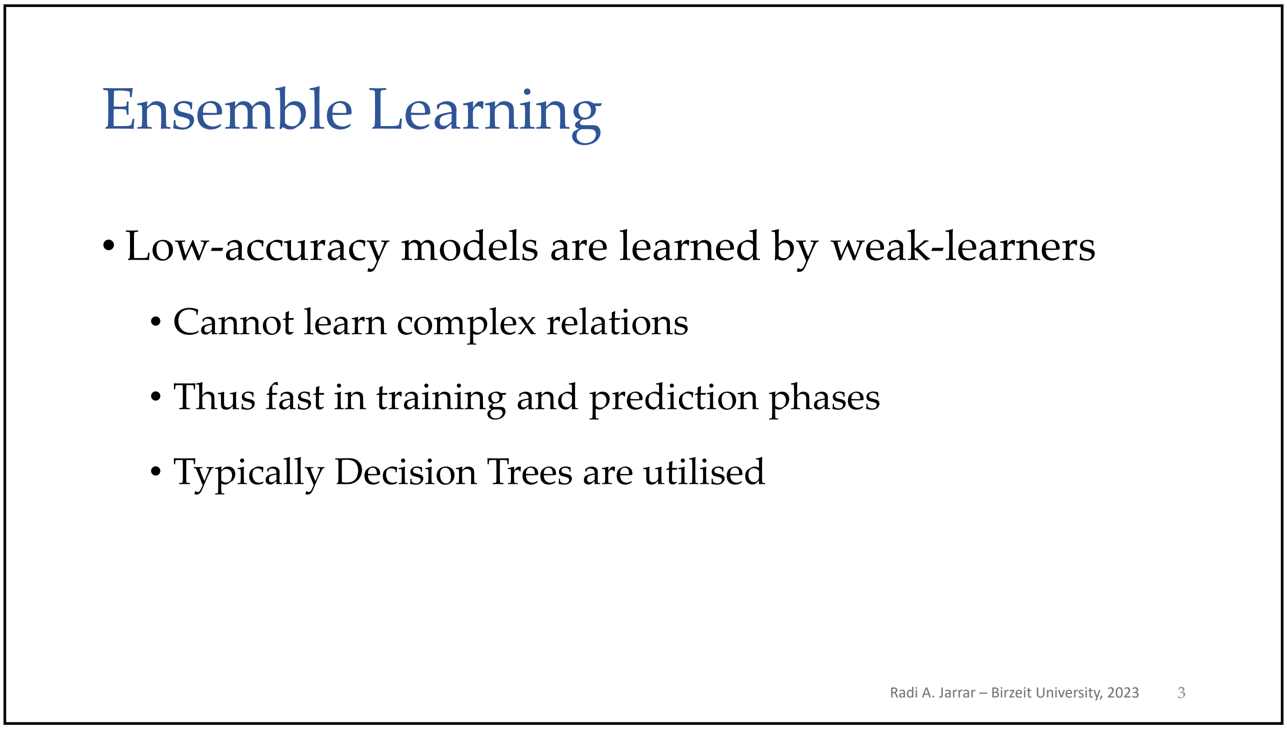
Dr. Radi Jarrar
Department of Computer Science

جامعة بيرزيت
BIRZEIT UNIVERSITY



Ensemble Learning

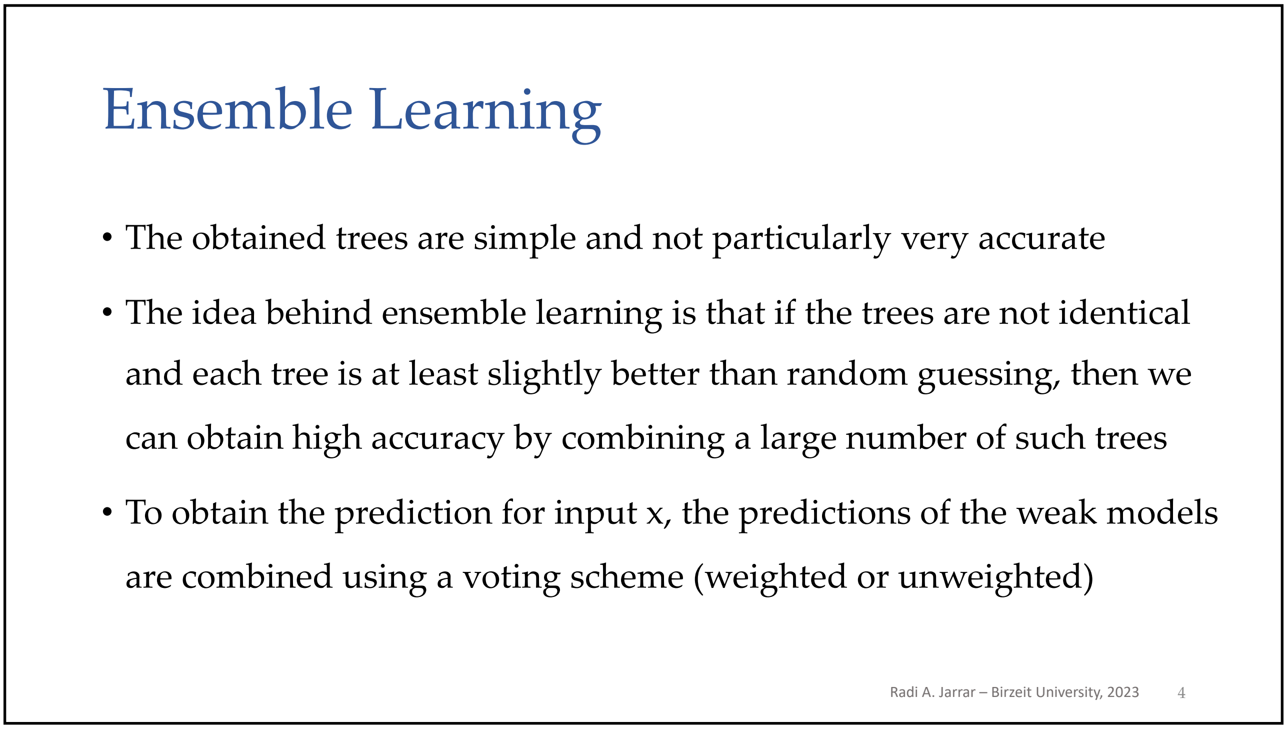
- Ensemble learning is a learning paradigm that, instead of trying to learn one very accurate model, focuses on training a large number of low-accuracy models
- The results of these models are then combined to obtain a high-accuracy meta-model



Ensemble Learning

- Low-accuracy models are learned by weak-learners
  - Cannot learn complex relations
  - Thus fast in training and prediction phases
  - Typically Decision Trees are utilised

3



Ensemble Learning

- The obtained trees are simple and not particularly very accurate
- The idea behind ensemble learning is that if the trees are not identical and each tree is at least slightly better than random guessing, then we can obtain high accuracy by combining a large number of such trees
- To obtain the prediction for input x, the predictions of the weak models are combined using a voting scheme (weighted or unweighted)

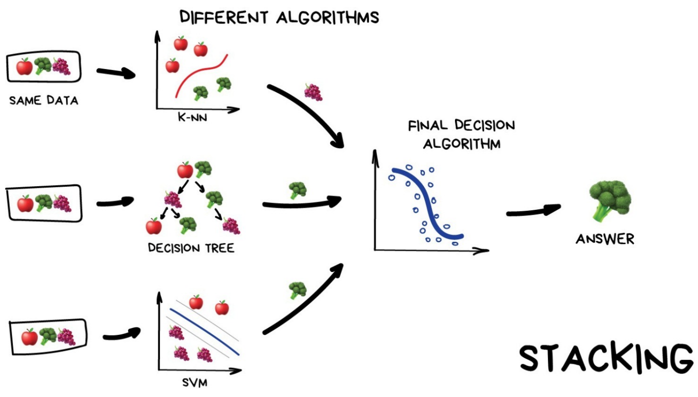
4

Ensemble Learning

- Types of Ensemble Learning
  - Boosting
  - Bagging
  - Stacking

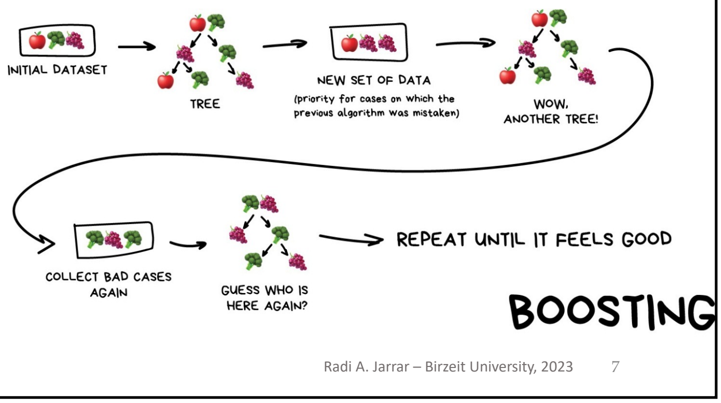
Stacking

- Heterogeneous weak learners (i.e., different learning algorithms)
- Learn from data in parallel
- The final result is determined by training a meta-model to output a prediction based on the different weak models predictions
- For example, the output of these models to be input into a NN to learn the final classification task



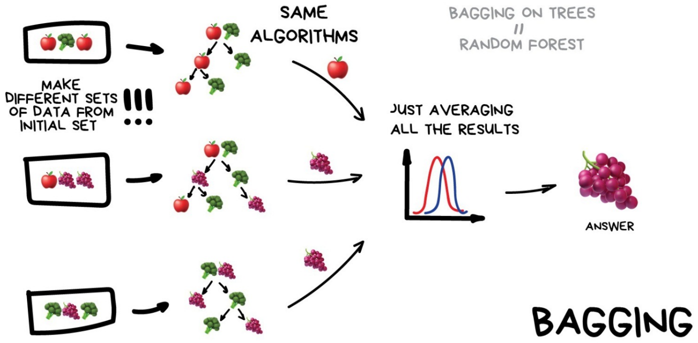
Boosting

- A number of homogeneous weak learners
- The models are learned sequentially in a very adaptative way (a base model depends on the previous ones)
- The final result is determined by combining the results of the models in a deterministic strategy



Bagging

- A number of homogeneous weak learners
- The models are learned in-parallel and independently
- The final result is determined by combining the results of the models in a deterministic averaging strategy


SAME ALGORITHMS
BAGGING ON TREES
=
RANDOM FOREST
MAKE DIFFERENT SETS OF DATA FROM INITIAL SET !!!
...
JUST AVERAGING ALL THE RESULTS
ANSWER
BAGGING

Bagging

- Bagging – **B**ootstrap **agg**regat**ing**
- Bootstrap: generating new datasets by sampling with replacement B times
- Replacement means that we may get some data several times and others not at all
- The bootstrap sample is the same size as the original

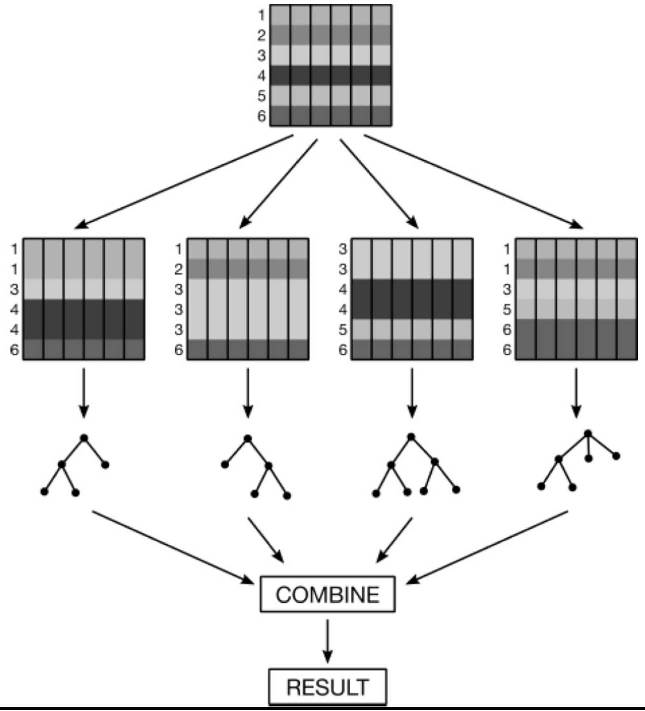
Original Dataset

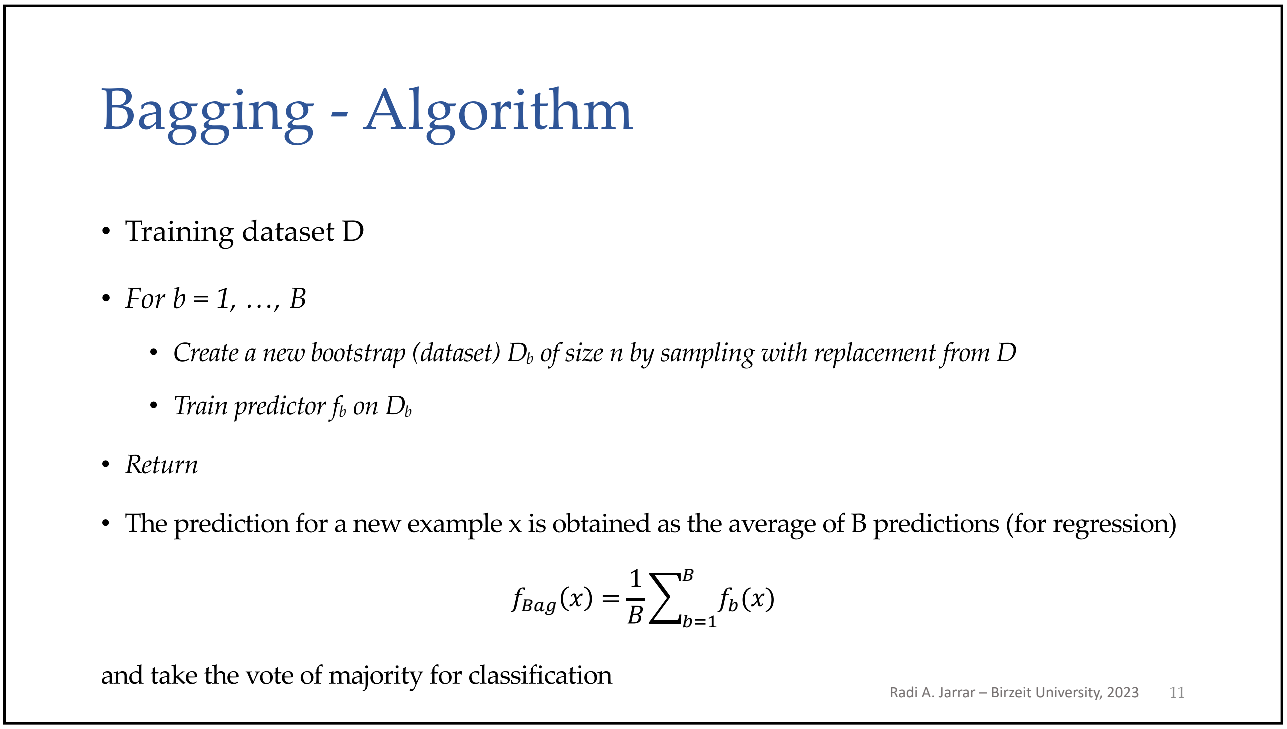
(x_1, y_1) (x_2, y_2)
(x_3, y_3) (x_4, y_4)
(x_5, y_5) (x_6, y_6)

Bootstrap 1

(x_2, y_2) (x_4, y_4)
(x_5, y_5) (x_2, y_2)
(x_2, y_2) (x_1, y_1)

Bagging



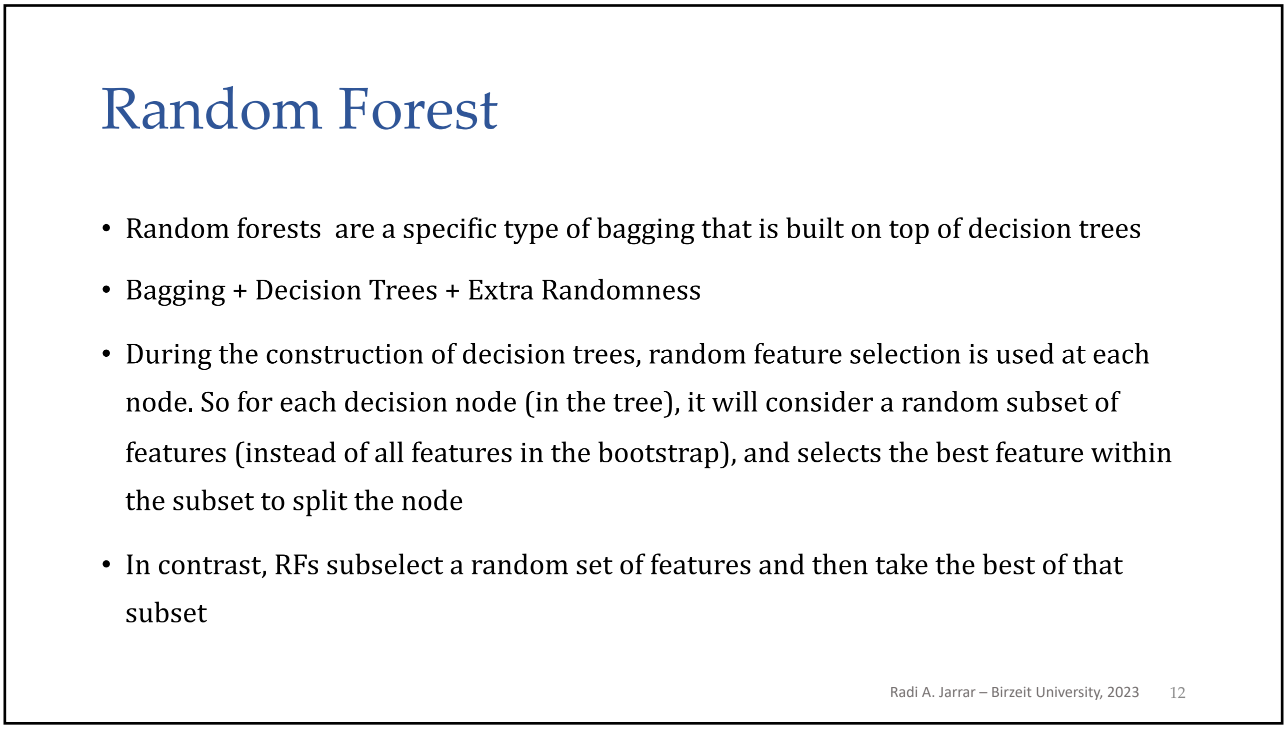


Bagging - Algorithm

- Training dataset D
- *For b = 1, …, B*
  - *Create a new bootstrap (dataset) D_b of size n by sampling with replacement from D*
  - *Train predictor f_b on D_b*
- *Return*
- The prediction for a new example x is obtained as the average of B predictions (for regression)

$$f_{Bag}(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x)$$

and take the vote of majority for classification

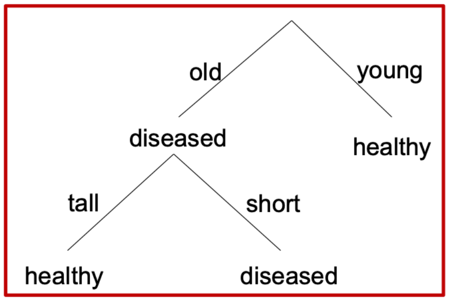


Random Forest

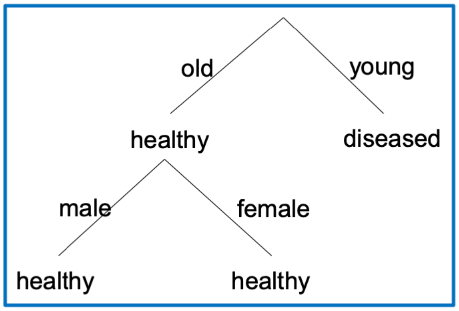
- Random forests are a specific type of bagging that is built on top of decision trees
- Bagging + Decision Trees + Extra Randomness
- During the construction of decision trees, random feature selection is used at each node. So for each decision node (in the tree), it will consider a random subset of features (instead of all features in the bootstrap), and selects the best feature within the subset to split the node
- In contrast, RFs subselect a random set of features and then take the best of that subset

Random Forest

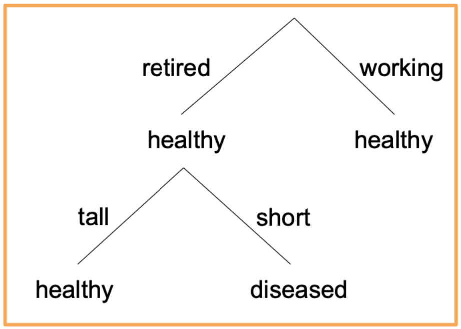
Tree 1



Tree 2



Tree 3



- New input instance:
  - Old, retired, male, short
- Tree predictions:
  - Diseased, healthy, diseased
- Majority votes: diseased

Random Forest

- The goal is to force the trees to be different from each other
- This is done to to avoid the correlation of the trees: if one or a few features are very strong predictors for the target, these features will be selected to split examples in many trees. This would result in many correlated trees in our “forest.”
- It leads to have mode diverse set of models
- Highly correlated features cannot help in improving the accuracy of prediction
- The main reason behind a better performance of model ensembling is that models that are good will likely agree on the same prediction, while bad models will likely disagree on different ones

Random Forest - Algorithm

- Training dataset D
- *For b = 1, …, B*
  - *Create a new bootstrap (dataset)* D_b *of size n by sampling with replacement from D*
  - *Build a tree* T_b *on* D_b *by recursively repeating the following until minimum node size k is reached*
    - *Select m features uniformly at random, without replacement, from the d features*
    - *Compute the information gain (or Gini impurity) only on that set of features, selecting the optimal one*
- *Output the ensemble of trees* $\{T_b\}_1^B$

Random Forest

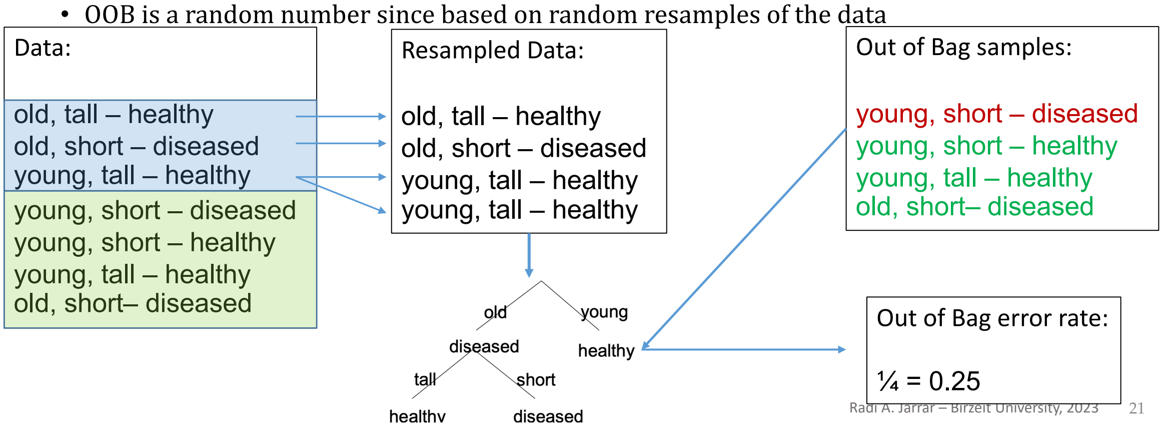
- The main parameters to tune in Random Forest learning are
  - the number of trees, B, and
  - the size of the random subset of the features to consider at each split
    - Max features: the number of columns that are show to each tree
    - Max samples: the maximum number of rows that are passed to each tree
  - Max depth: the number of splits each generated decision tree is allowed to make
    - Caution: too low causes underfitting and too high causes overfitting. Max depth typically set to 3, 5, or 7

Random Forest

- Random Forests help in reducing overfitting (reduce variance)
- When using multiple samples of the original dataset, the **variance is reduced in the** final model
  - Overfitting happens when the model tries to explain small variations in the dataset because the dataset is just a small sample of the population of all possible examples of the phenomenon we try to model
  - If we were unlucky with how our training set was sampled, then it could contain some undesirable (but unavoidable) artifacts: noise, outliers and over- or underrepresented examples, etc.
  - By creating multiple random samples with replacement of our training set, we reduce the effect of these artifacts
- Reduce bias
  - Bias occurs when there is a certain degree if error introduced in the model
- Bias may occur when you are not evenly splitting the instance space during the training. So instead of seeing all of the data points, you might see only half because this is how you set your model up

Estimating Error – Out of Bag (OOB) Score

- Out-of-Bag (OOB) score is a method to estimate the error in RF models
- Similar to cross-validation but here there is much less computation cost
- OOB is a random number since based on random resamples of the data



Data:

old, tall – healthy
old, short – diseased
young, tall – healthy
young, short – diseased
young, short – healthy
young, tall – healthy
old, short– diseased

Resampled Data:

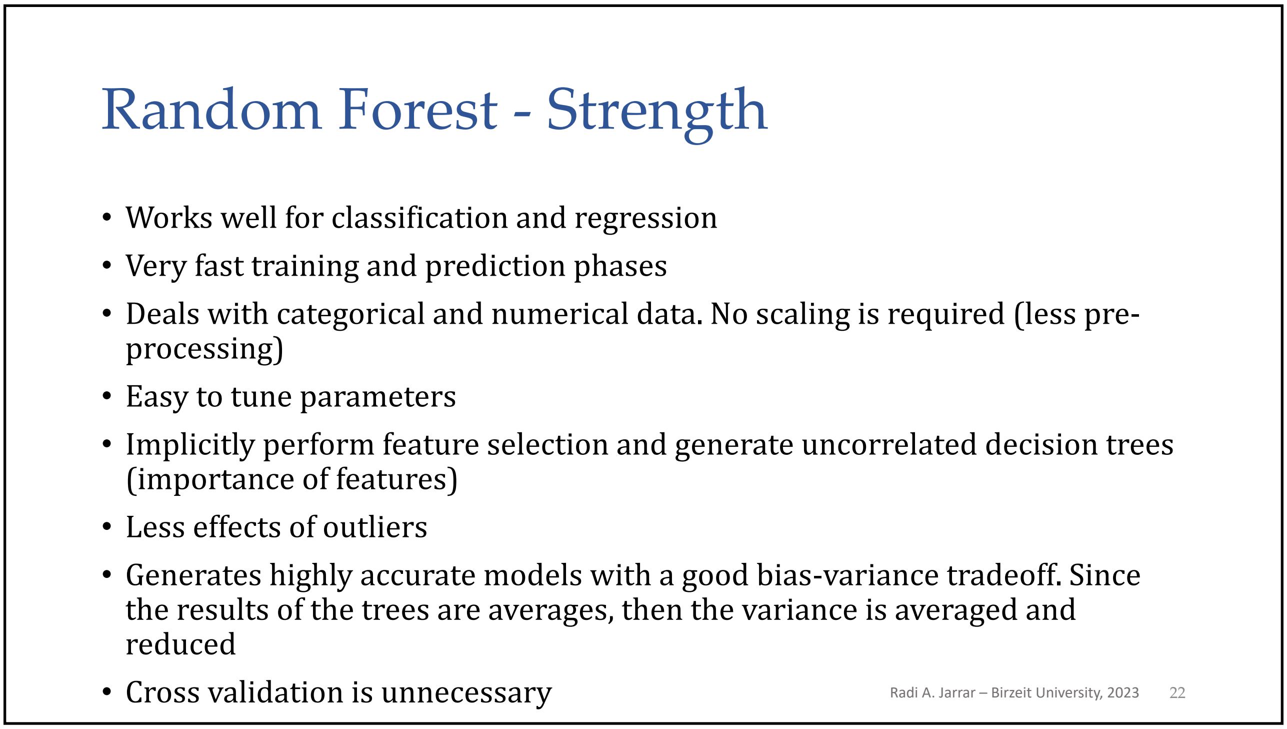
old, tall – healthy
old, short – diseased
young, tall – healthy
young, tall – healthy

Out of Bag samples:

young, short – diseased
young, short – healthy
young, tall – healthy
old, short– diseased

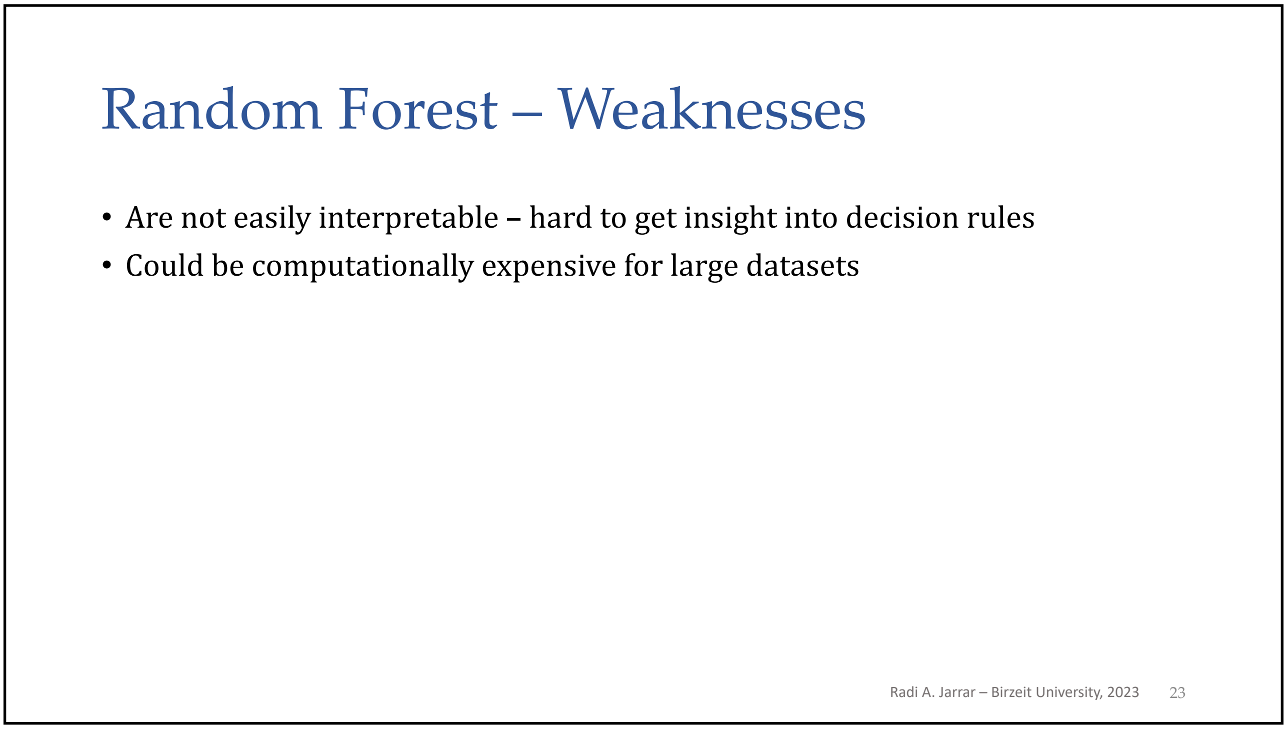
Out of Bag error rate:

¼ = 0.25



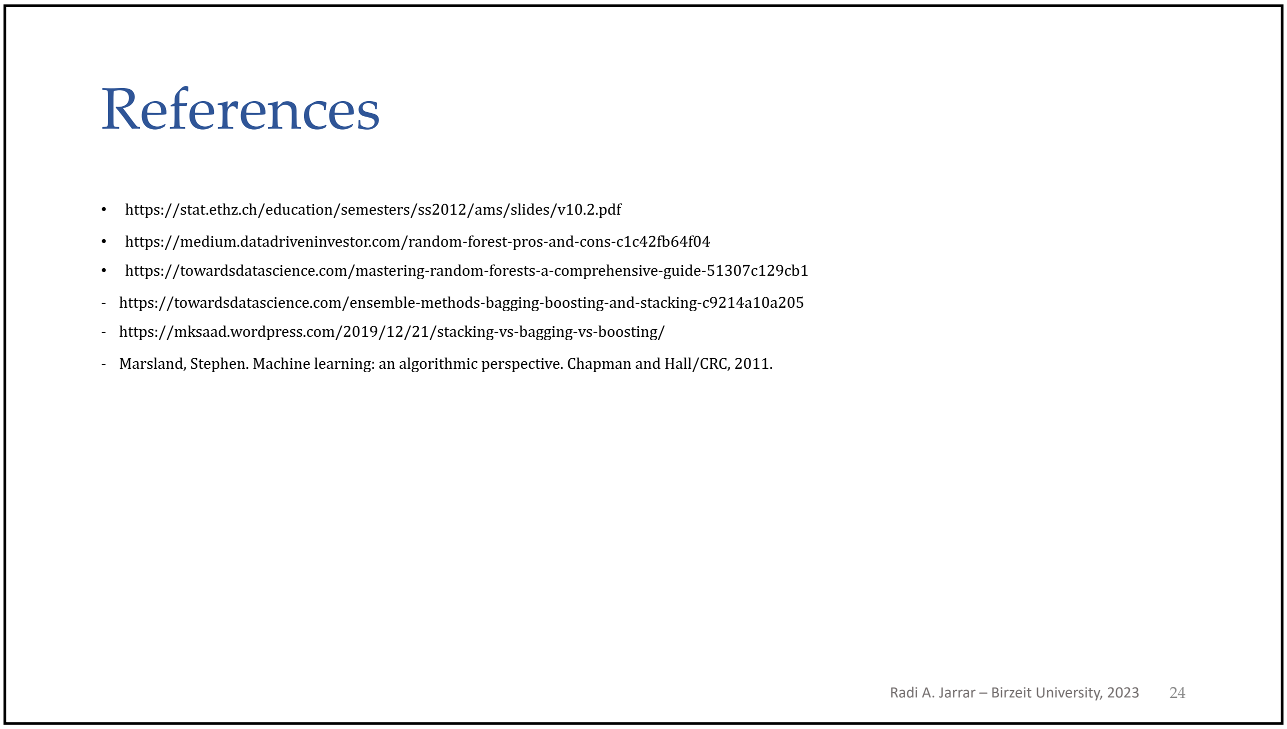
Random Forest - Strength

- Works well for classification and regression
- Very fast training and prediction phases
- Deals with categorical and numerical data. No scaling is required (less pre-processing)
- Easy to tune parameters
- Implicitly perform feature selection and generate uncorrelated decision trees (importance of features)
- Less effects of outliers
- Generates highly accurate models with a good bias-variance tradeoff. Since the results of the trees are averages, then the variance is averaged and reduced
- Cross validation is unnecessary



Random Forest – Weaknesses

- Are not easily interpretable – hard to get insight into decision rules
- Could be computationally expensive for large datasets



References

- https://stat.ethz.ch/education/semesters/ss2012/ams/slides/v10.2.pdf
- https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04
- https://towardsdatascience.com/mastering-random-forests-a-comprehensive-guide-51307c129cb1

- https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205
- https://mksaad.wordpress.com/2019/12/21/stacking-vs-bagging-vs-boosting/
- Marsland, Stephen. Machine learning: an algorithmic perspective. Chapman and Hall/CRC, 2011.