Week 6

Machine Learning and Big Data - DATA622

Fall 2023

CUNY School of Professional Studies

Week 6

1. Discussion Board Week 6: Decision Trees vs Random Forest
2. Reading materials:
   - **Textbook Reading: Practical Machine Learning in R (PMLiR)**
     - **PMLiR Chapter 10: Improving Performance (30 mins)**
       Chapter covers parameter tuning, bagging & boosting of models.
     - ISLR **Section 8.2 (Pages 340-352): Bagging, Random Forests, Boosting, and Bayesian Additive Regression Trees (30 mins)**
       "An Introduction to Statistical Learning" isn't our main textbook but is the definitive source for this topic (see below how to download free copy from author's site). You can skip the text if you watch the videos, or vice versa.
   - **ML Concepts Reading & Videos**
     - **Bagging (also known as Bootstrap Aggregation)**
       Video: Overview of Bagging: https://youtu.be/omSN-shKM1Y (14 mins)
       This summarizes ISLR Section 8.2.1 bagging on Page 340. Slides available here.
     - **Boosting and Variable Performance**
       Video: https://www.youtube.com/watch?v=RSWg_islt9c (14 mins)
       This summarizes ISLR Section 8.2.3 bagging on Page 345. Slides available here.
     - **Lab: Decision Trees**
       Video: Implementing Decision Trees in R: https://youtu.be/YPz2J5lHeVM (10 mins)
     - **Lab: Random Forest Trees**
       Video: Implementing Random Forest in R: https://youtu.be/MpDEU96Ss8E (15 mins)

   Note: "ISLR" refers to the book "An Introduction to Statistical Learning" which you should have from the prerequisite courses. You can buy it here, or it is available for free as a PDF here (author's site here). "ESL" refers to the book "The Elements of Statistical Learning" which you should have from the prerequisite courses. You can buy it here, or it is available for free as a PDF here (author's site here)

Decision Trees

Recap

CUNY School of Professional Studies

Decision Trees

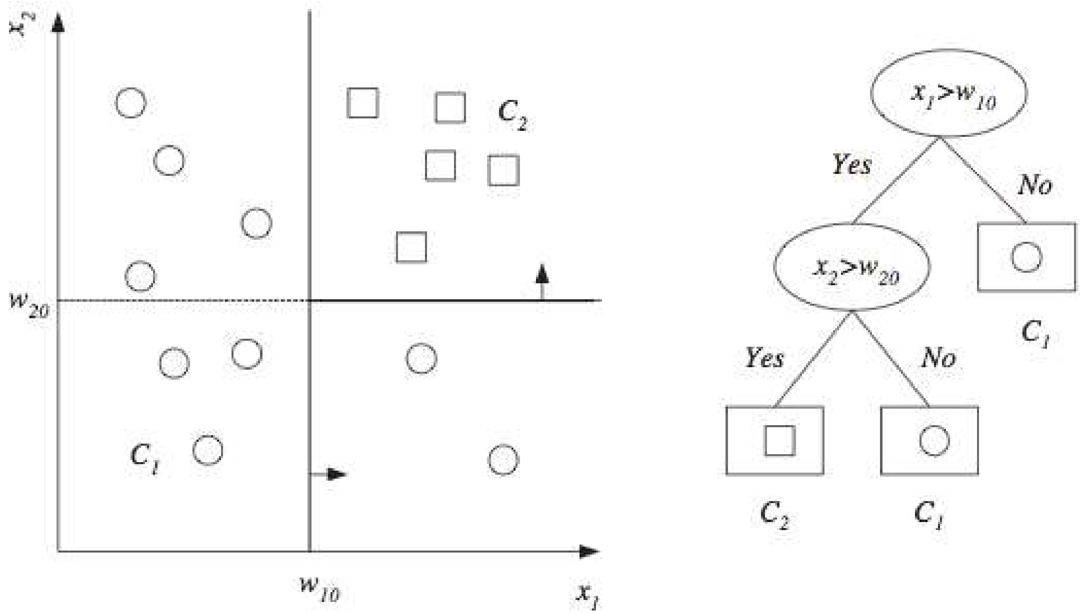


Figure 9.1 Example of a dataset and the corresponding decision tree. Oval nodes are the decision nodes and rectangles are leaf nodes. The univariate decision node splits along one axis, and successive splits are orthogonal to each other. After the first split, $\{\mathbf{x} | x_1 < w_{10}\}$ is pure and is not split further.

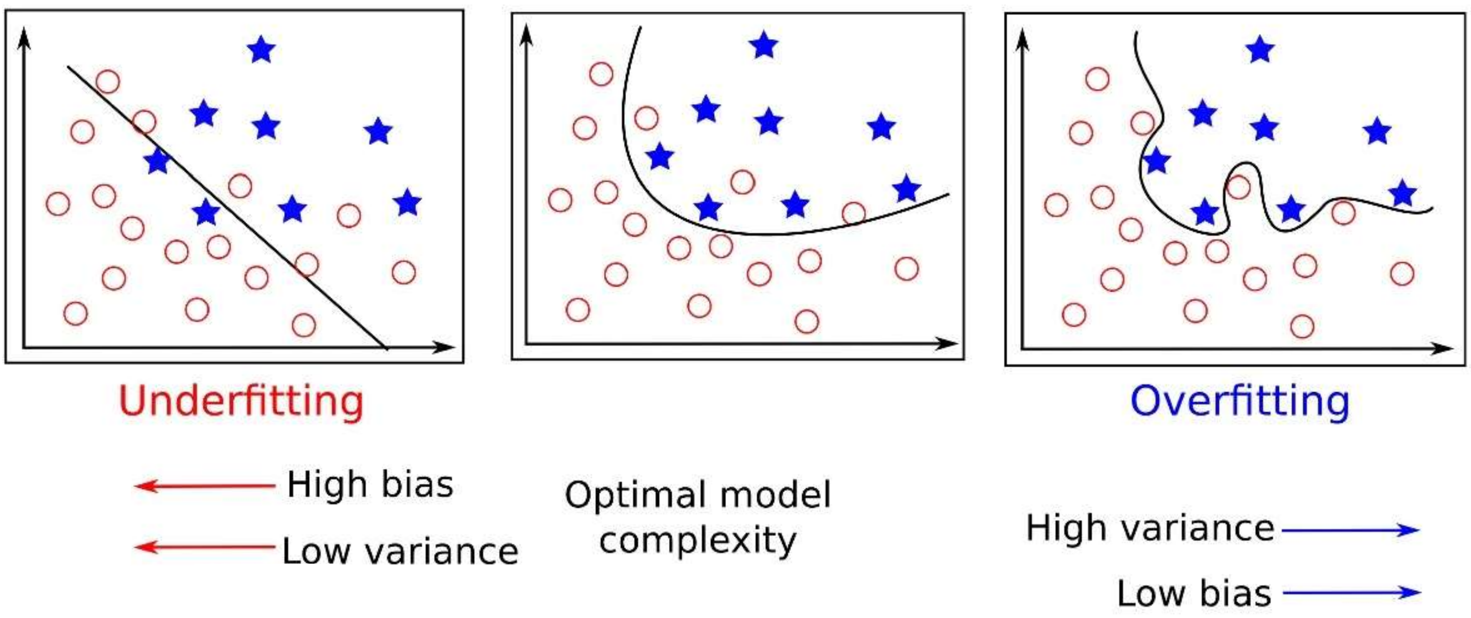
Decision Tree Construction

- For a given dataset there are many trees with no error
- Finding the tree with no error and fewest nodes is NP-complete
- Finding split in one column
  - Goodness of split given by entropy, gini index, or misclassification error.
  - Binary classification: Let *p* be the proportion of instances in class 0.
  - Find split that minimizes weighted sum of impurities of child nodes:
    - Misclassification error: 1 – max(p,1-p)
    - Entropy: plog(p) – (1-p)log(1-p)
    - Gini index: 2p(1-p)
- Similar approach for regression – (regression trees)

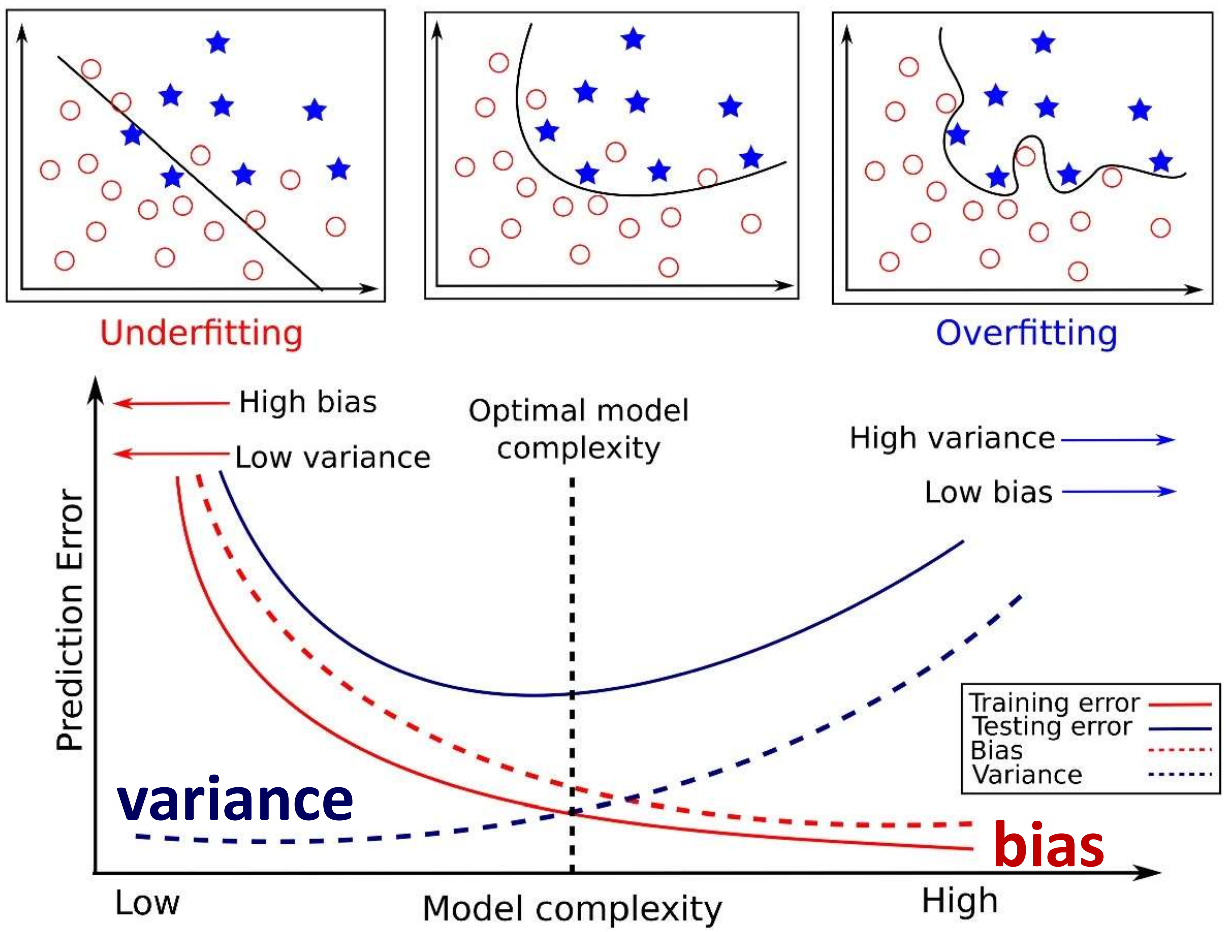
Bias and Variance

Note: We don’t mean societal bias – but model bias.

Bias and Variance



Bias and Variance



CUNY School of Professional Studies

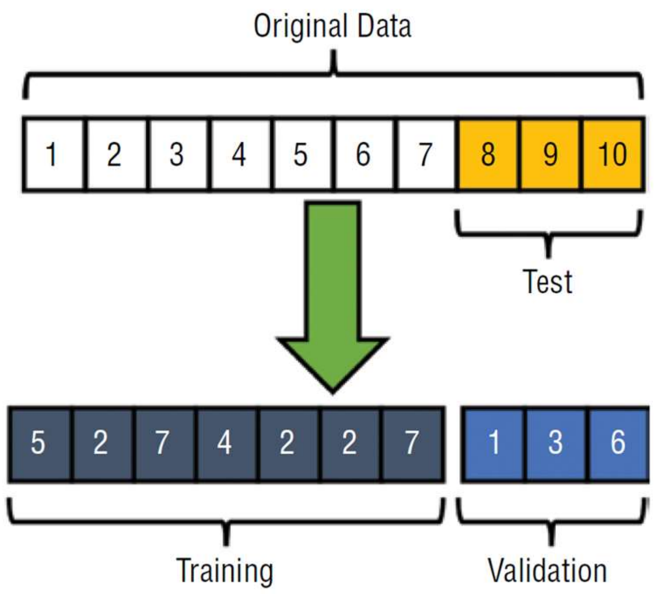
Bootstrapping

Bootstrap Aggregation

CUNY School of Professional Studies

Bootstrapping

- Create a training dataset from the original data
- Randomly sample data, with replacement



Bootstrapping

- The probability of picking one row out of n is *1/n*.
- Therefore the probability of not picking it is *1-(1/n)*
- After n trials the probability of not picking it is *(1-(1/n))n*
- As n approaches infinity *(1-(1/n))n* becomes *e-1=0.368*.
- Hence, approximately 63.2% of datapoints are uniquely selected in a bootstrap

Bagging

Bootstrap Aggregation

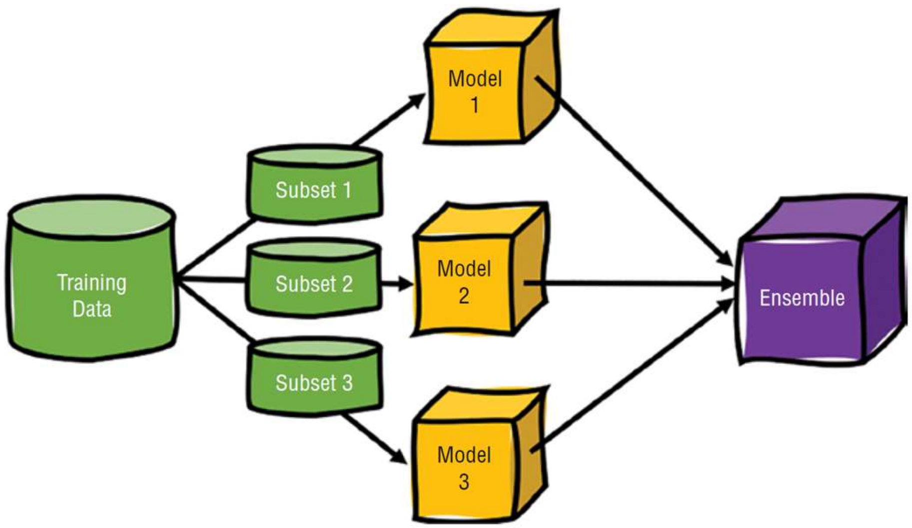
CUNY | School of Professional Studies

Bootstrap Aggregation (Bagging)

- Ensemble method that “manipulates the training set”
- Uses Bootstrapping
- Action: repeatedly sample with replacement according to uniform probability distribution
  - Every instance has equal chance of being picked
  - Some instances may be picked multiple times; others may not be chosen
- Sample Size: same as training set

Bagging

- Boosting works by iteratively generting models and adding them to the ensemble
- Iteration stops when a predefined number of models have been added
- Each new model added to the ensemble is biased to pay more attention to instances that previous models misclassified (weighted dataset).



CUNY School of Professional Studies

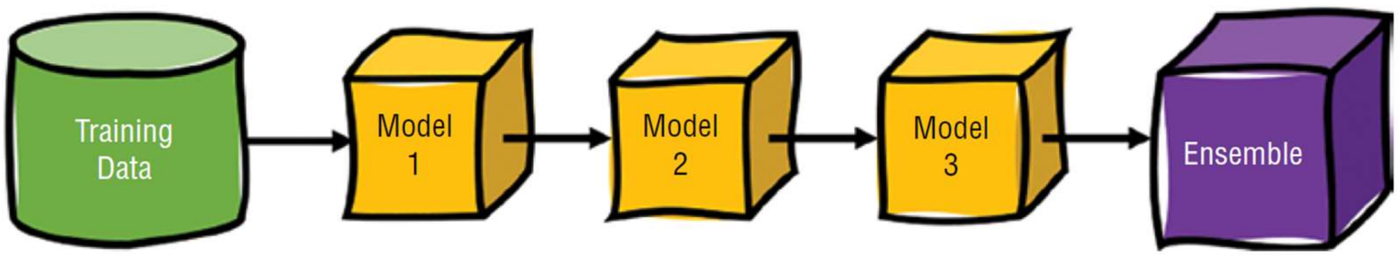
Boosting

Bootstrap Aggregation

CUNY | School of Professional Studies

Boosting

- Boosting works by iteratively creating models and adding them to the ensemble
- Iteration stops when a predefined number of models have been added
- Each new model added to the ensemble is biased to pay more attention to instances that previous models misclassified (weighted dataset).



Boosting

- Sequential algorithm where at each step, a weak learner is trained based on the results of the previous learner.
- Two main types:
  - Adaptive Boosting: Reweight datapoints based on performance of last weak learner. Focuses on points where previous learner had trouble. Example: AdaBoost.
  - Gradient Boosting: Train new learner on residuals of overall model. Constitutes gradient boosting because approximating the residual and adding to the previous result is essentially a form of gradient descent. Example: XGBoost.