Ensemble Learning

COSC 480A: Applied Machine Learning

Spring 2021

Prof. Apthorpe

Outline

- Main Idea
- Voting Classifiers
- Bagging & Pasting
- Random Forests
- Stacking
- Boosting
- Takeaways

Main Idea

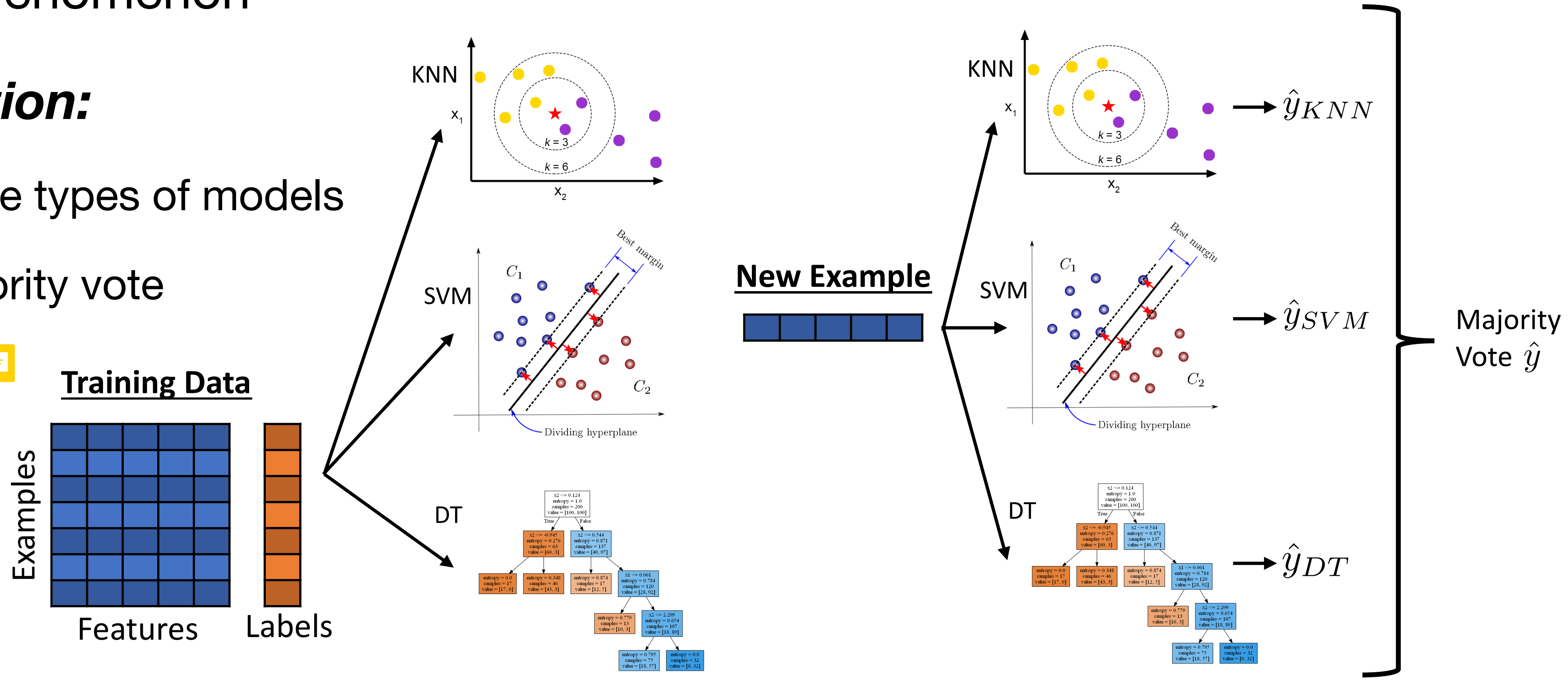
- Improve classification performance by combining many models
  - *Many diverse perspectives better than one opinion*

Thought Experiment

Voting Classifiers

- ***Key Idea:*** Different types of models represent different aspects of underlying phenomenon

- ***Implementation:***
  - Train multiple types of models
  - Predict majority vote



Voting Classifiers

- **"Hard"** voting classifier
  - **Evenly** weight "vote" from all classifiers
- **"Soft"** voting classifier
  - Use **predicted probability** to weight "votes"
- How do KNN, SVM, and decision trees estimate prediction probabilities?

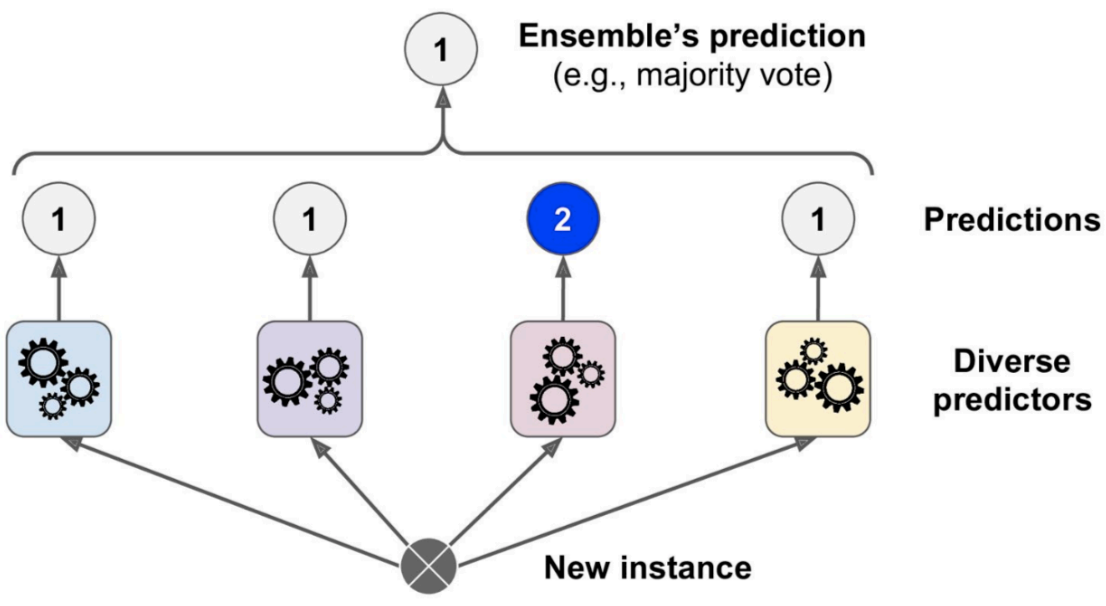
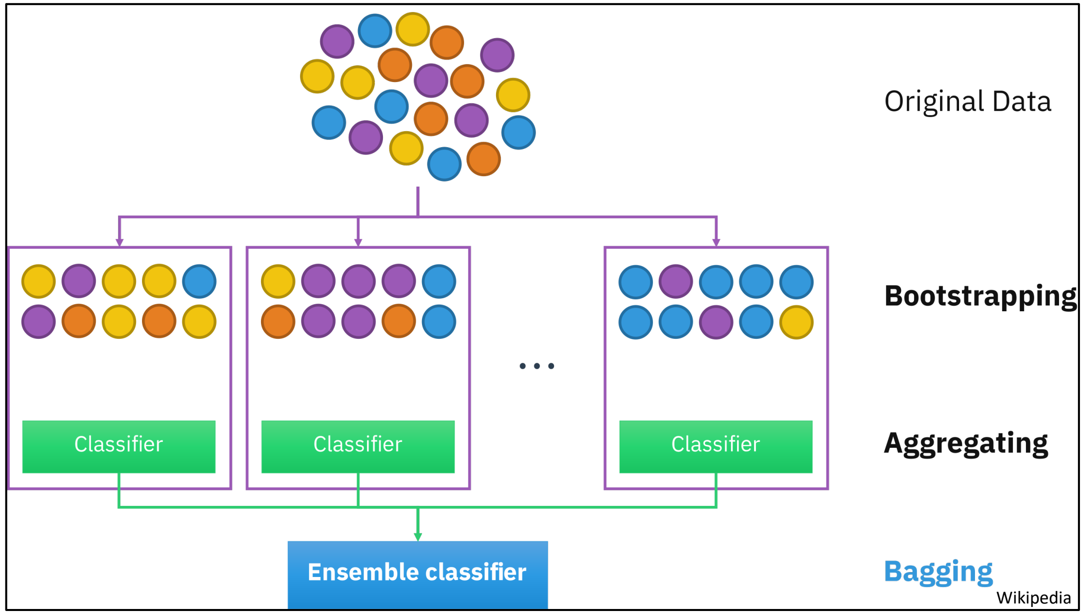


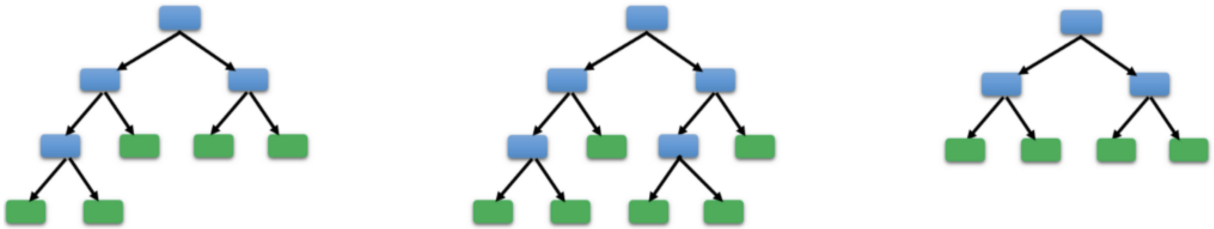
Figure 7-2. Hard voting classifier predictions

Bagging & Pasting

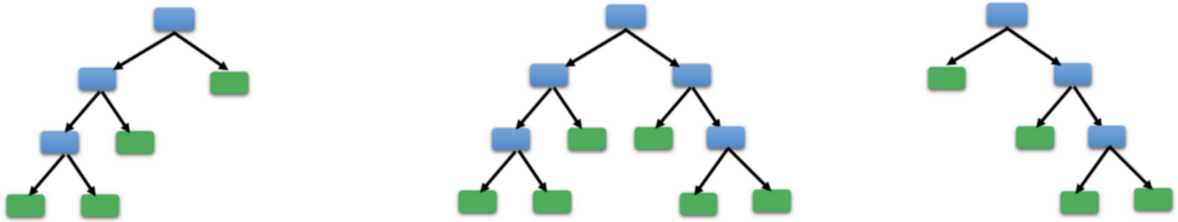
- ***Key Idea:*** Stochastic classifiers may have **high variance**
- ***Implementation:*** Training multiple instances of the same type of classifier on **subsets of training data** will **reduce variance**
  - **Bagging:** Sampling **with** replacement
  - **Pasting:** Sampling **without** replacement
- Hard or soft voting for final prediction



Random Forests



- **Many decision trees trained using bagging or pasting**
    - Limit max depth or number of leaf nodes to increase diversity
    - Reduces variance from stochastic decision tree training (CART or ID3)
    - More robust feature importance metrics than single decision tree
    - **Competes with deep learning when data has obvious features**
    - *Few hyperparameters, robust to overfitting, generally good results!*



Stacking

- ***Key Idea:*** Hard and soft voting can't express that some models may be better or worse than others at prediction task
- ***Implementation:*** Train a meta-model to weight votes of each classifier

Cross-Validation Stacking

$$\hat{y} = \sum_{m \in M} v_m h_m(\mathbf{x})$$

Prediction

Sum of weighted votes of *m* classifiers in ensemble

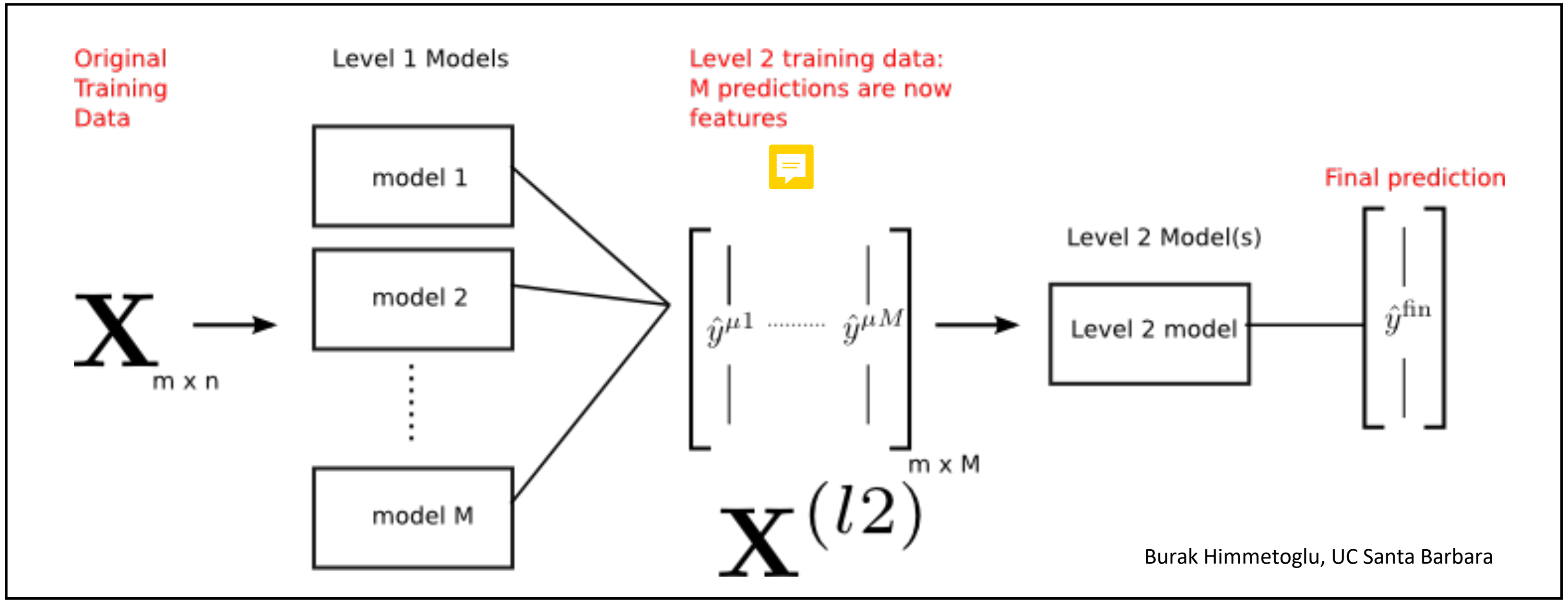
$$\mathbf{v} = \underset{\mathbf{v}}{\operatorname{argmin}} \sum_{i=1}^{N} E(y_i, \sum_{m=1}^{M} v_m h_m^{-1}(\mathbf{x}))$$

Choose weights that minimize the sum of the leave-one-out cross-validation errors across the ensemble

Stacking

- ***Key Idea:*** Hard and soft voting can't express that some models may be better or worse than others at prediction task

- ***Implementation:*** Train a **meta-model** to weight votes of each classifier



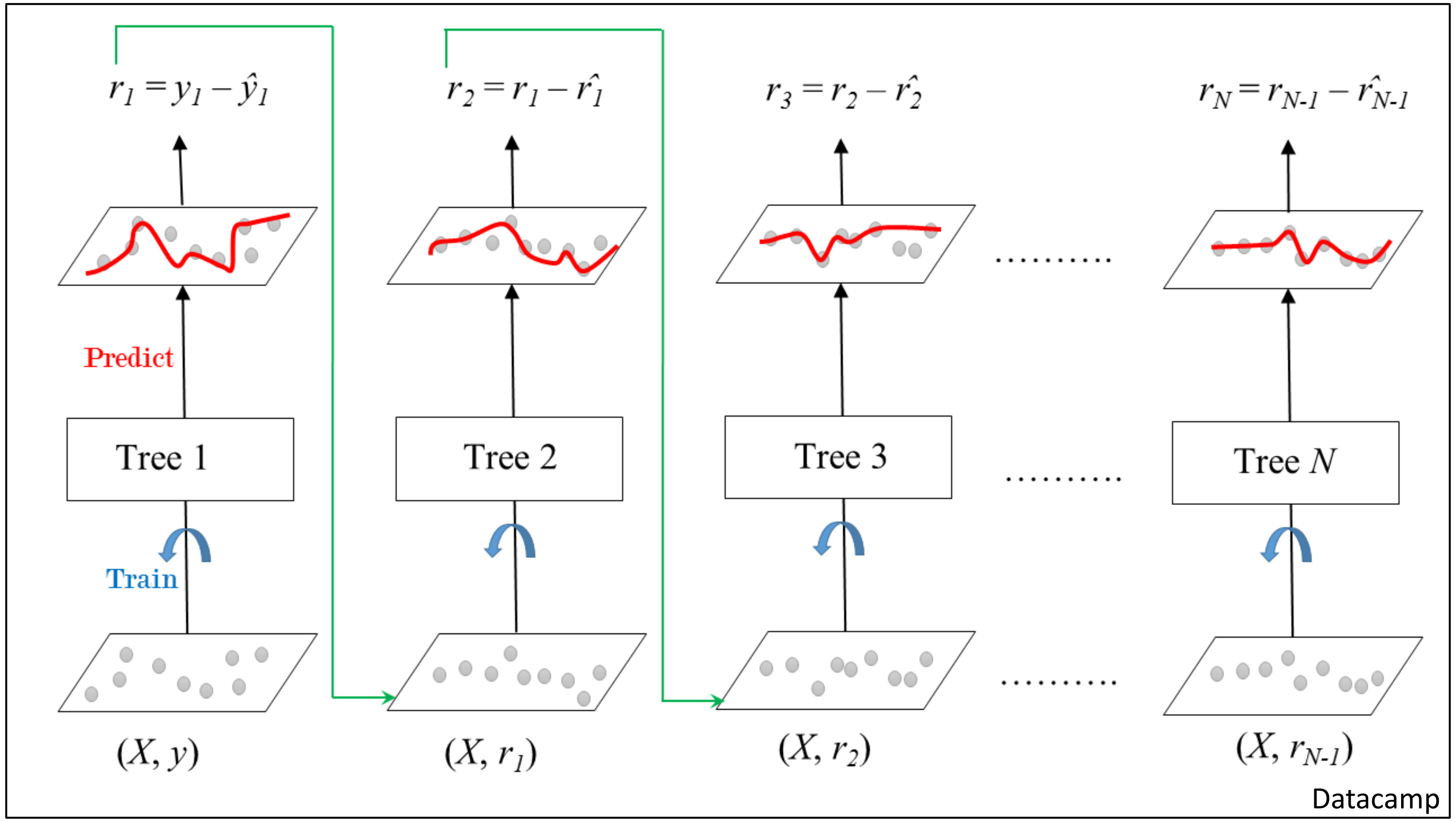
Boosting

- ***Key Idea:*** Shallow ML classifiers can exhibit **bias errors**, i.e. mistakes due to assumptions that simplify learning but miss underlying complexities of data
  - How does a depth 2 decision tree exhibit bias? How about a linear SVM?

- ***Implementation:*** Train multiple classifiers in sequence, each to correct mistakes made by the previous

Gradient Boosting

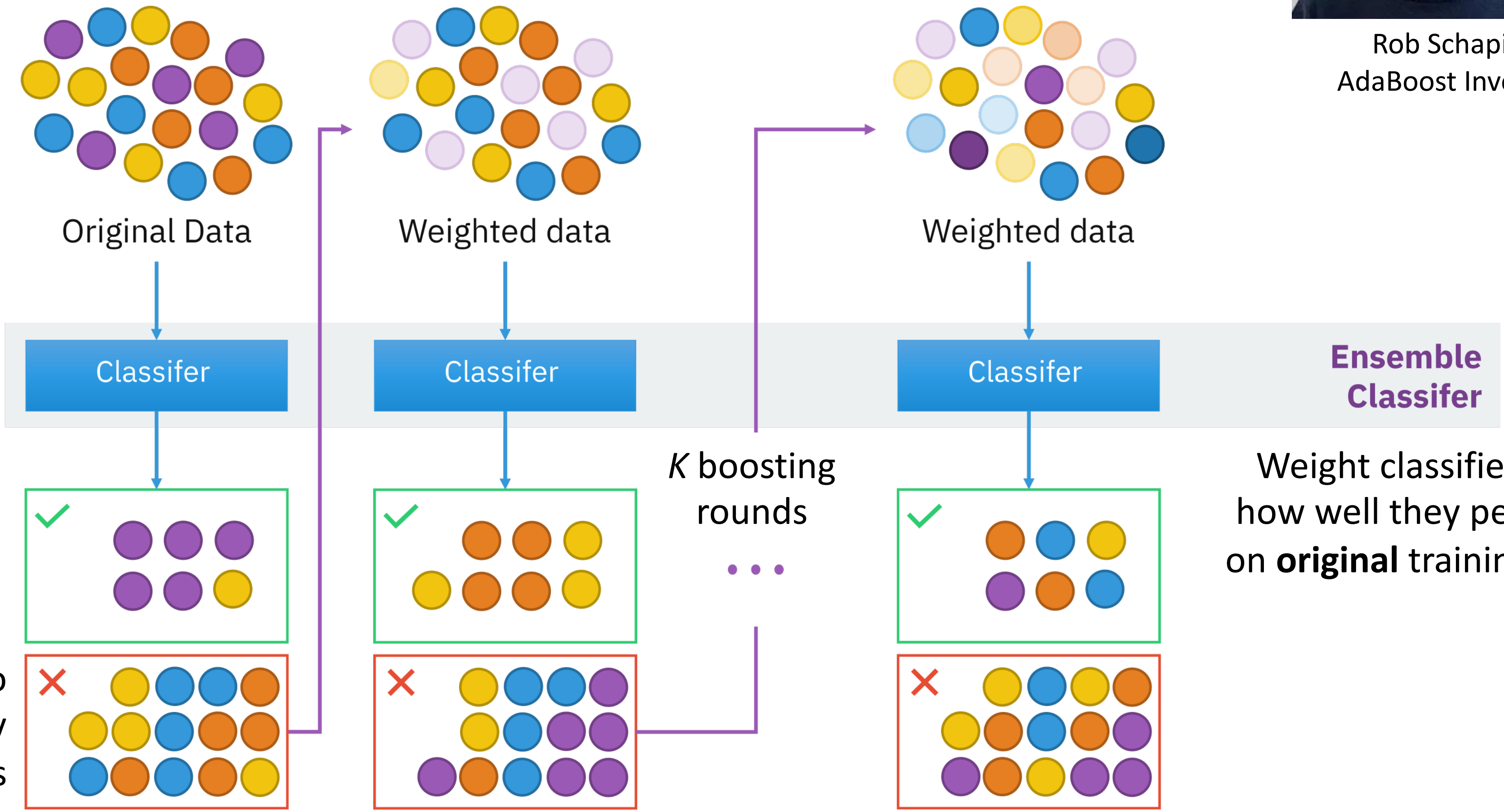
- Train each successive model to predict the **error** of the previous



Final label is sum of sequential predictions

AdaBoost

Rob Schapire
AdaBoost Inventor



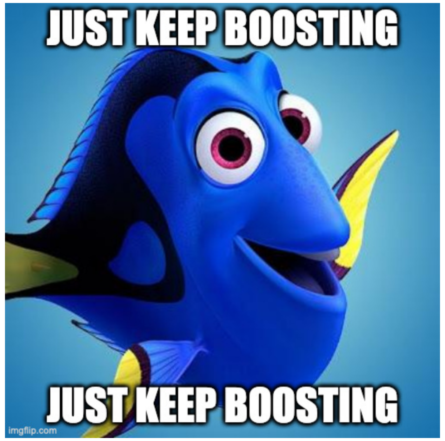
Re-weight training data to prioritize prior incorrectly labeled examples

Weight classifiers by how well they perform on **original** training data

AdaBoost

Weak learner: any classifier that does better (even if only slightly better) than random guessing

- **AdaBoost + any weak learner**
  - Zero training error with enough boosting rounds
  - Improved test error with additional rounds



- **AdaBoost + decision trees**
  - Competes with deep learning when data has obvious features
  - Many fewer parameters and hyperparameters than deep learning
  - Easier to tune with much less chance of overfitting than deep learning

Takeaways

- Ensemble methods can be applied to **any** supervised classifier
  - If computation time permits, give it a try!

- Diverse set of simple models
  - Better than one simple model
  - Often better than one complex model
- Random Forests and AdaBoost are among best supervised ML methods

5-Minute Break

Programming Practice

EnsembleLearning.ipynb