ENSEMBLE METHODS

BAGGING & RANDOM FOREST

CS 334: Machine Learning

WISDOM OF CROWDS

"Wisdom of Crowds" (Surowiecki, 2004) - the collective knowledge of a diverse and independent body of people typically exceeds the knowledge of any single individual, and can be harnessed by voting

ENSEMBLE METHODS

- Use multiple “learners” to solve the same problem
- Reduce bias/variance and improve performance

ENSEMBLE METHODS

"This is how you win ML competitions: you take other peoples' work and ensemble them together."

- Vitaly Kuznetsov, NIPS 2014

GROUP ACTIVITY

ENSEMBLE METHODS

- Given a dataset, how to get multiple learners to ensure diverse opinions?
- How to combine the multiple learners?

ENSEMBLE METHODS

- Same classifier (different datasets)
  - Bagging/averaging: build multiple models independently and then average – reduce variance
  - Boosting: build multiple models sequentially – reduce bias
- Different classifiers (same datasets)
  - Voting: average or weighted average of multiple different classifiers
  - Stacking: predictions of multiple classifiers are used as input to another estimator for final prediction

ENSEMBLE METHODS

- Bagging and Random forest
- Boosting and Gradient boosted tree
- Voting and Stacking

BAGGING/AVERAGING METHODS

- Bootstrapping (resampling with replacement) – bagging
- Random subsets of the dataset (sampling without replacement) – pasting
- Random subsets of the features – random subspaces
- Random subsets of both samples and features – random patches

https://scikit-learn.org/stable/modules/ensemble.html#

BAGGING

- Bootstrap Aggregating: variance reduction technique introduced by Breiman in 1992
- Method: Average predictions over collection of bootstrap samples
  - Create B bootstrap replicates
  - Fits model to each replicate
  - Combines predictions via averaging or voting

BOOTSTRAPPING

- Fundamental resampling tool in statistics
- Resampling with replacement
- General and most widely used tool to estimate measures of uncertainty associated with a given statistical model (e.g., confidence intervals, bias, variance, etc.)

BOOTSTRAPPING

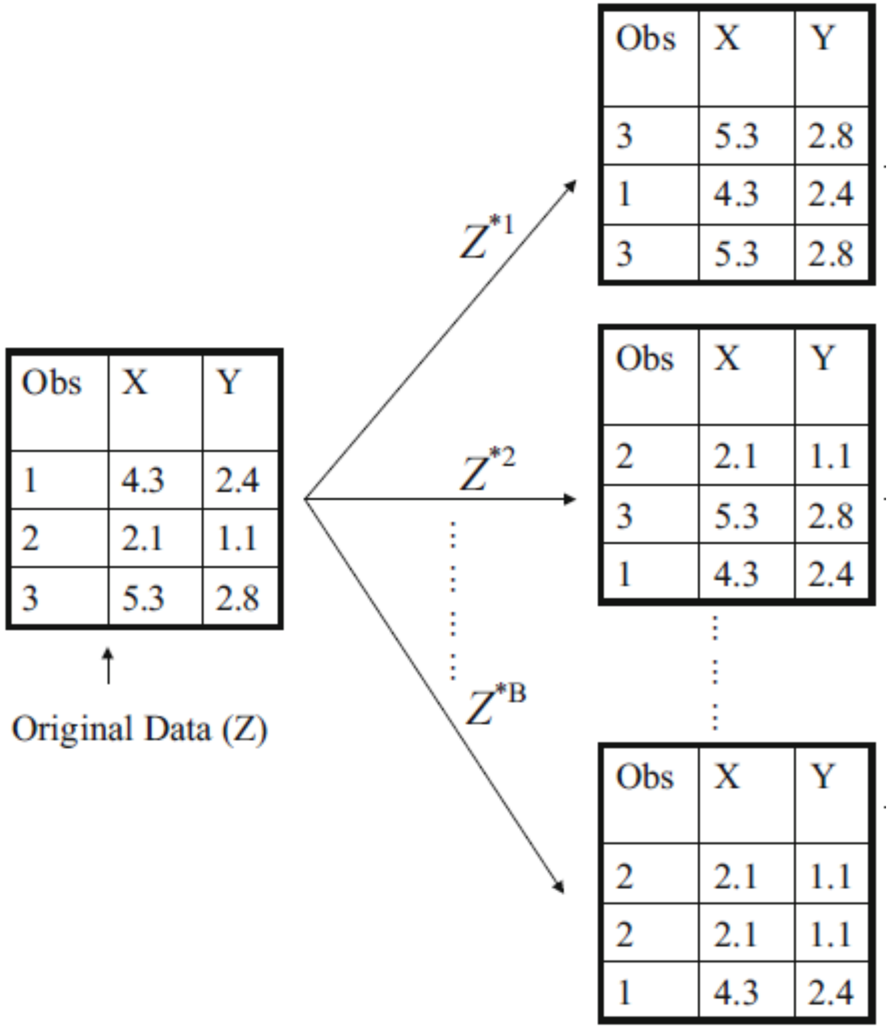
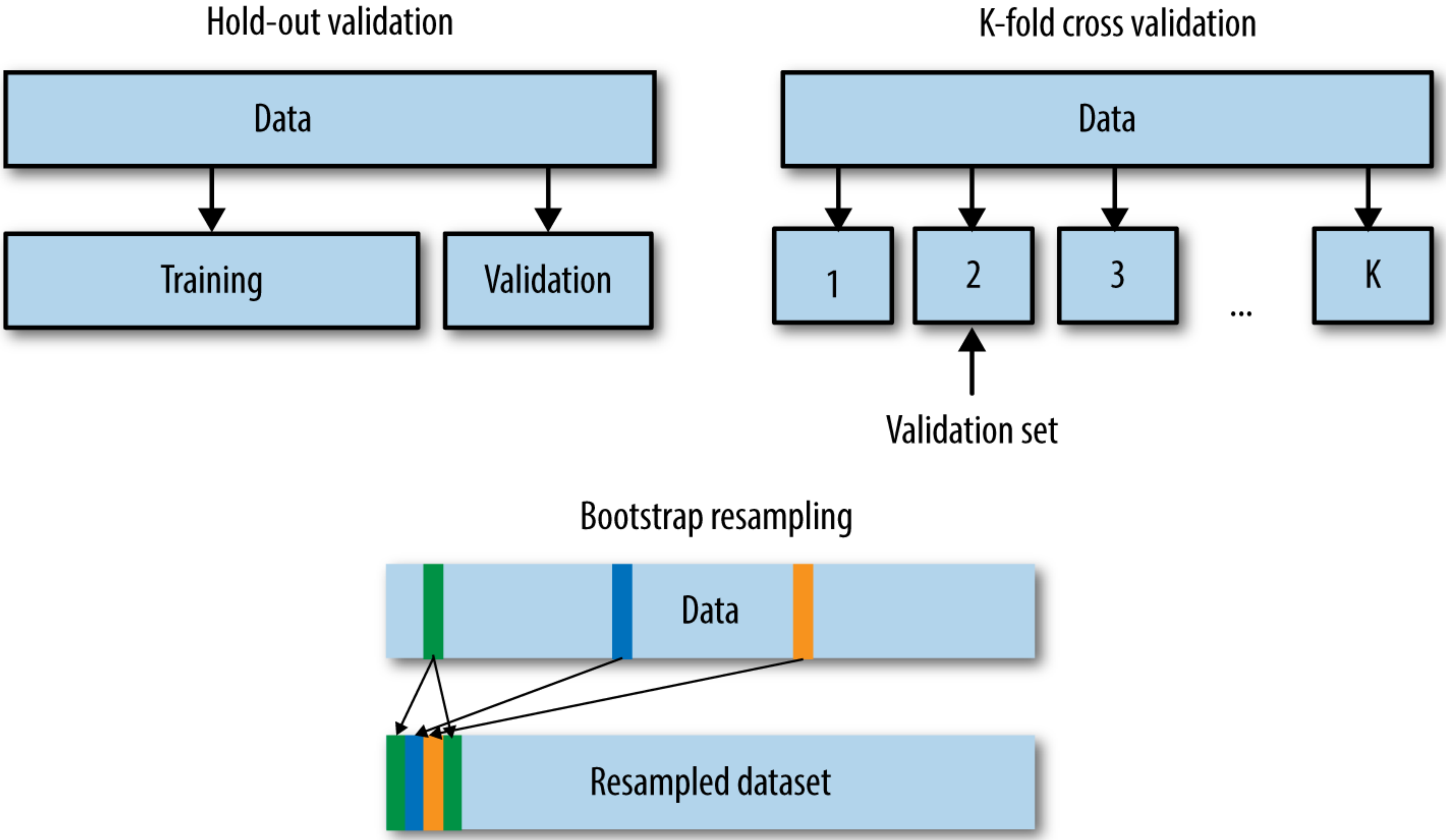


Figure 5.11 James et al


Hold-out validation
Data
Training
Validation
K-fold cross validation
Data
1
2
3
...
K
Validation set
Bootstrap resampling
Data
Resampled dataset

BAGGING: COMBINING MULTIPLE LEARNERS

- Regression: averaging
- Classification: majority voting

$$\hat{f}^{\text{bag}}(\mathbf{x}) = \text{argmax}_G \sum_b \mathbb{1}_{\{\hat{f}_b^{\text{tree}}(\mathbf{x})=g\}}$$

- Classification: average of predicted class probabilities, then choose class with highest probability

$$\hat{p}^{\text{bag}}(y = g|\mathbf{x}) = \frac{1}{B}\sum_b \hat{p}_b^{\text{tree}}(y = g|\mathbf{x})$$

- Classification: averaging probability preferable for estimates of class probabilities and can help overall prediction accuracy

BAGGING: COMBINING MULTIPLE LEARNERS

- What if we were to use the proportion of votes for class g as estimated probability?

$$\hat{p}_g^{\mathrm{bag}}(\mathbf{x}) = \frac{1}{B} \sum_b \mathbb{1}_{\{\hat{f}_b^{\mathrm{tree}}(\mathbf{x}) = g\}}$$

- Why would this not be a good estimate?

BAGGING & TREES

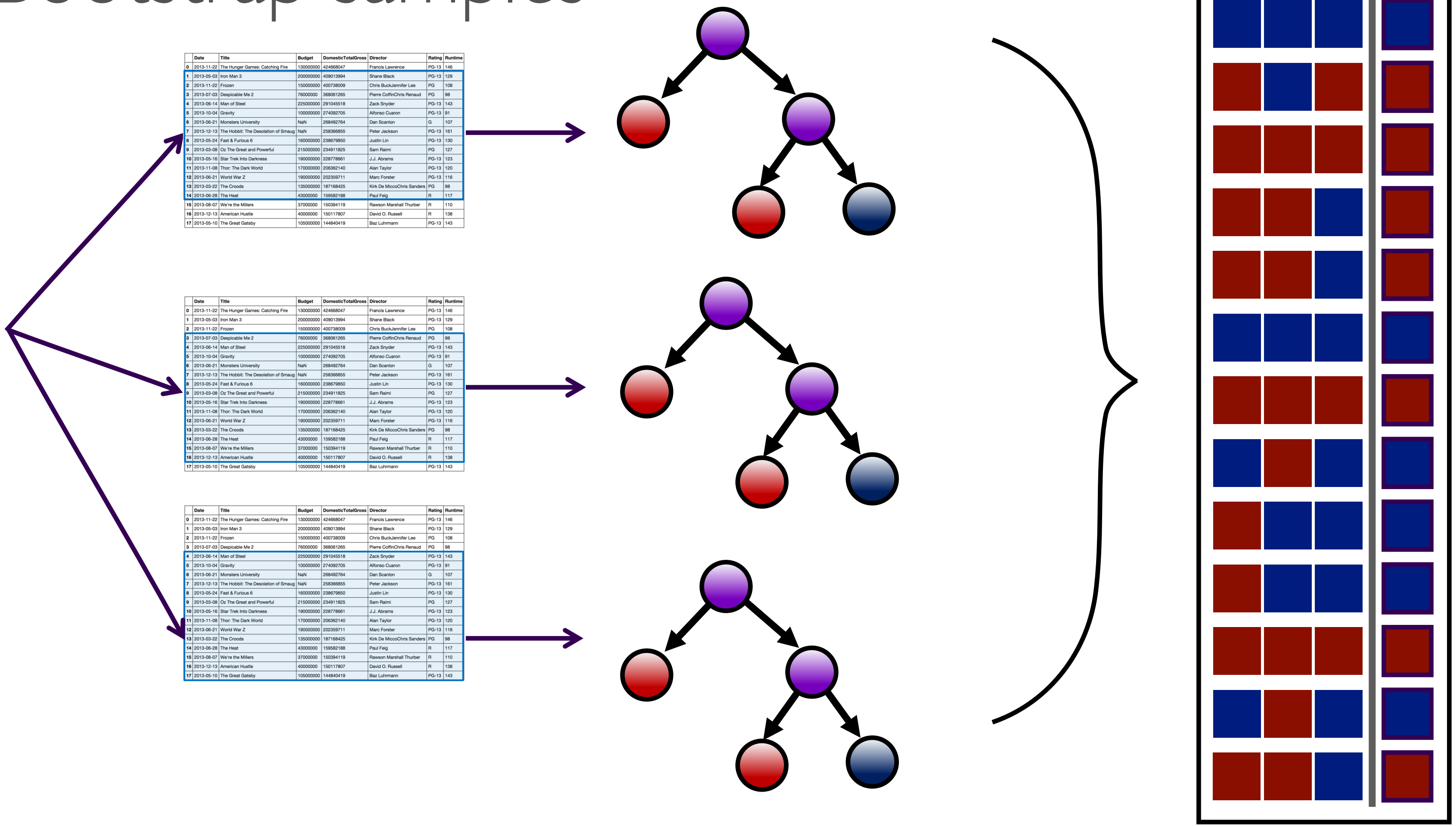
- Trees are ideal candidates for bagging
  - Capture somewhat complex boundaries (with sufficient depth)
  - Low bias but high variance
- Bagging: the bias does not change but variance is reduced

BAGGING: CONCEPTUALLY

Original dataset

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Bootstrap samples



EXAMPLE: BAGGING + DECISION TREE

Simulated data with n=30, two classes, and 5 features (high pairwise correlations)

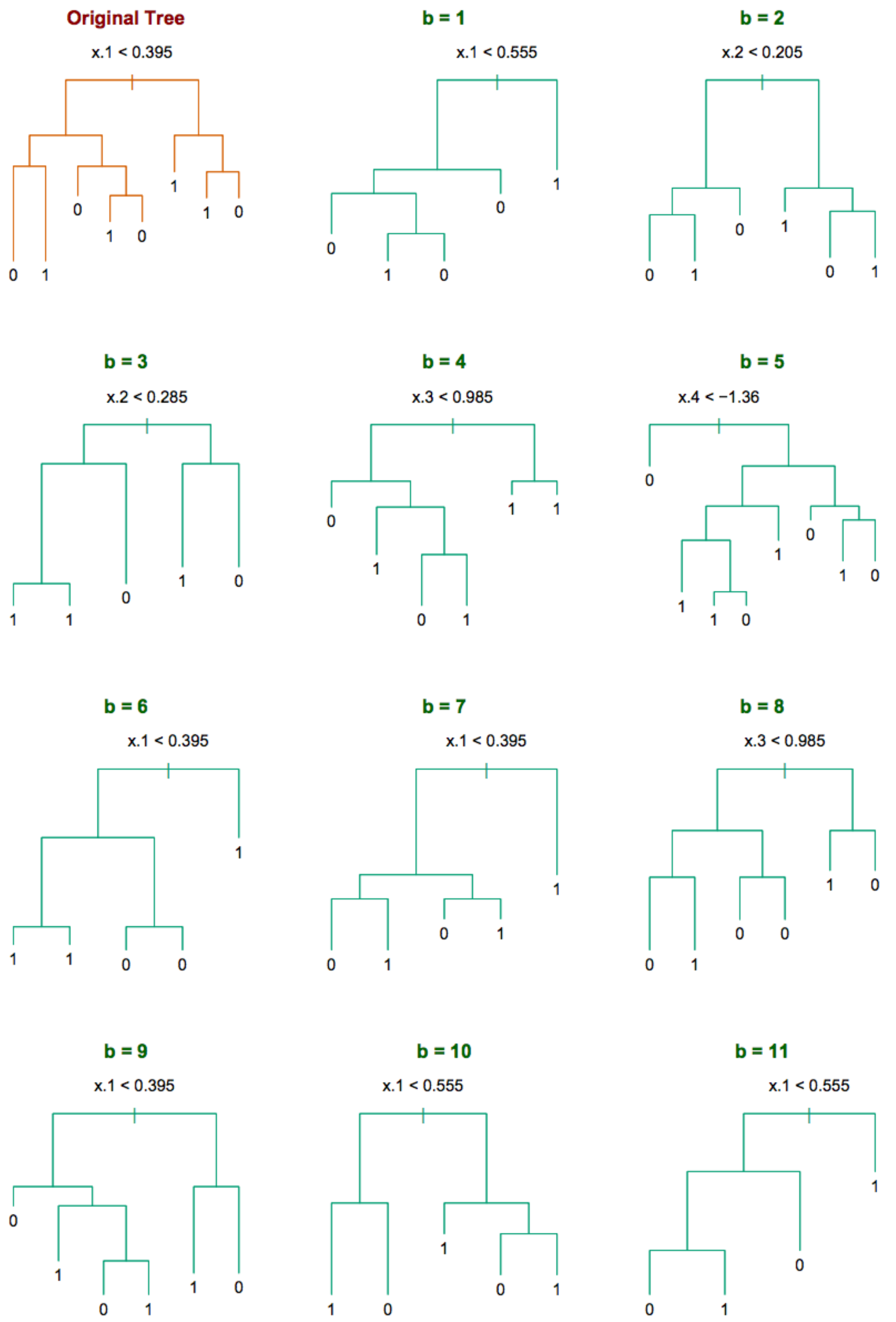
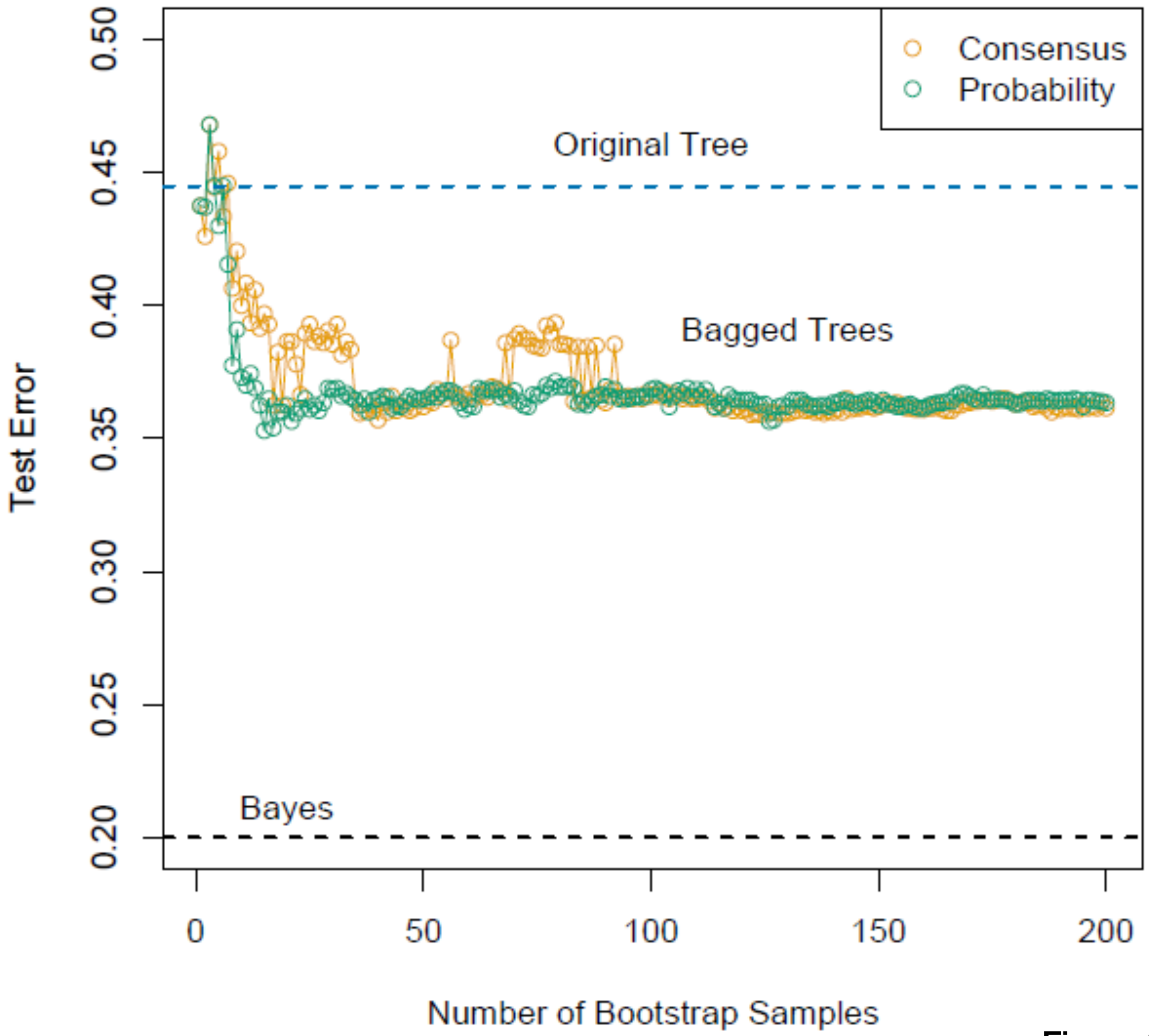


Figure 8.9 (Hastie et al.)

EXAMPLE: BAGGING + DECISION TREE



How many bags to choose?

Figure 8.10 (Hastie et al.)

EXAMPLE: BREIMAN'S EXPERIMENT

Data Set	$\bar{e}_S$	$\bar{e}_B$	**Decrease**
waveform	29.1	19.3	34%
heart	4.9	2.8	43%
breast cancer	5.9	3.7	37%
ionosphere	11.2	7.9	29%
diabetes	25.3	23.9	6%
glass	30.4	23.6	22%
soybean	8.6	6.8	21%

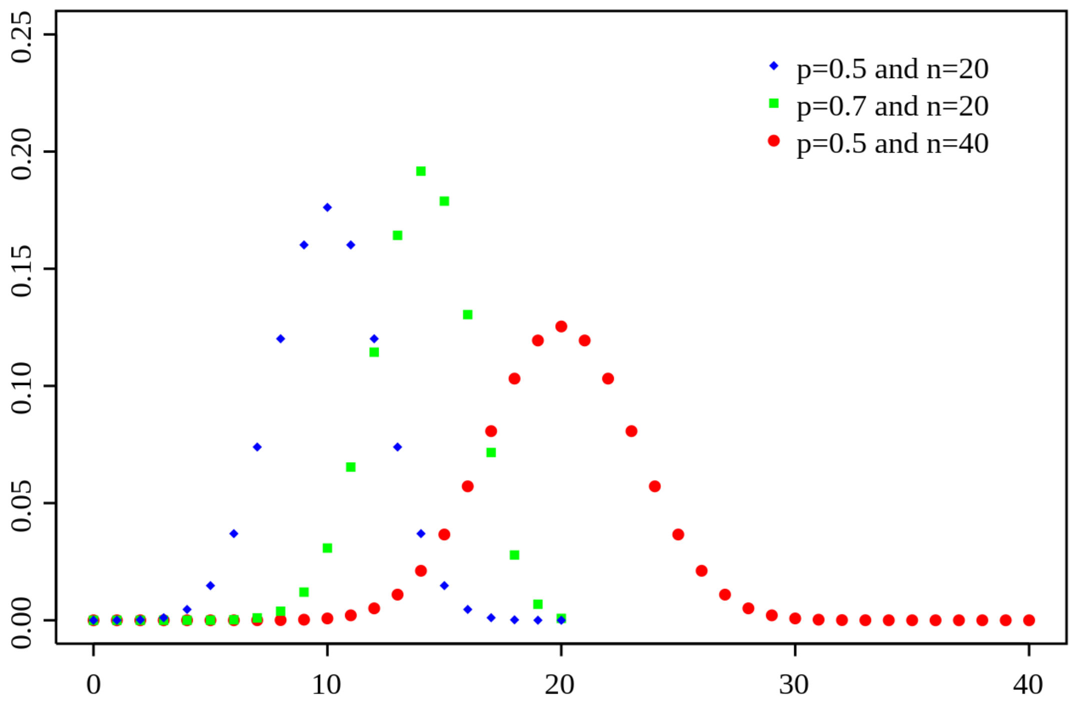
Comparison of misclassification error between CART tree (pruned via cross-validation) and bagging (B = 50)

WHY DOES BAGGING WORK?

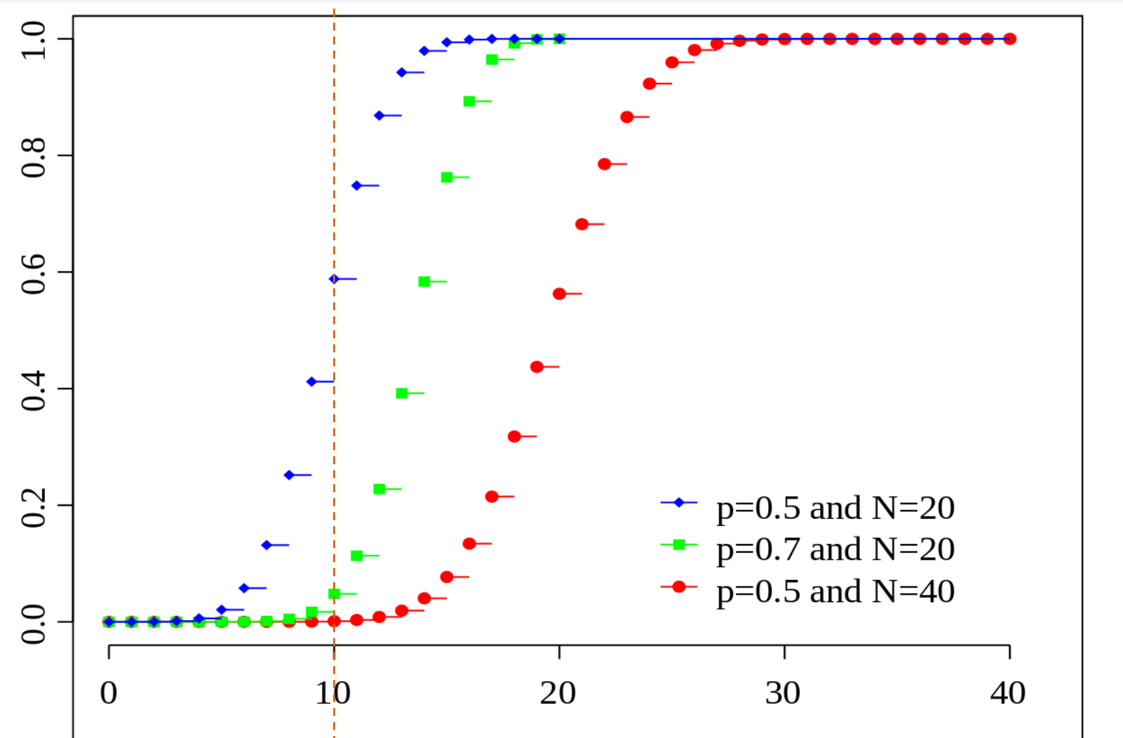
- Suppose a binary classification problem and we have B independent classifiers, each has an accuracy of p (misclassification rate 1-p)
- Our bagged classifier: $\hat{f}(\mathbf{x}) = \mathrm{argmax}_G \sum_b \mathbb{1}_{\{\hat{f}_b^{\mathrm{tree}}(\mathbf{x})=g\}}$
- The number of positive votes of bagged classifier is a Binomial variable with probability p
- Assume without loss of generality that the true class is 1
  - Correct prediction if the number >= B/2, incorrect if < B/2

BINOMIAL DISTRIBUTION

$$\Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$



$$\Pr(X \le k) = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$$



What happens when p increase? When n increases?
When does bagging fail?

BAGGING

- If each classifier has a misclassification rate over 0.5
  - The bagged classifier will fail and become perfectly inaccurate as B approaches infinity
- Assume each classifier has a misclassification rate lower than 0.5
  - As B grows larger, the bagged classifier should be perfect in theory
  - Often this is not the case, since individual classifiers are not independent

RANDOM FOREST: MOTIVATION

- For B independent trees with same variance, bagged variance is:

$$\sigma^2/B$$

- For B trees with positive pairwise correlation ρ , bagged variance is:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- Correlation of bagged trees limits benefits of averaging

How to reduce correlation?

RANDOM FORESTS (BREIMAN, 2001)

- Bagged classifier using decision trees
  - Each split only considers a **random group of features**
  - Tree is grown to maximum size without pruning
  - Final predictions obtained by aggregating over the B trees

$$\hat{f}_{\mathrm{rf}}^{B}(\mathbf{x}) = \frac{1}{B}\sum_{b} T(\mathbf{x}; \theta_b)$$

- Reduce variance (at the cost of slight increase in bias)

RANDOM FOREST: ALGORITHM

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\mathrm{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\mathrm{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

RANDOM FOREST: ALGORITHM

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

What's a good number of trees?

What's a good number of subset of variables?

EXAMPLE: GENE EXPRESSION

15-class gene expression
data set with p = 500 predictors

When m=p, equivalent to bagging

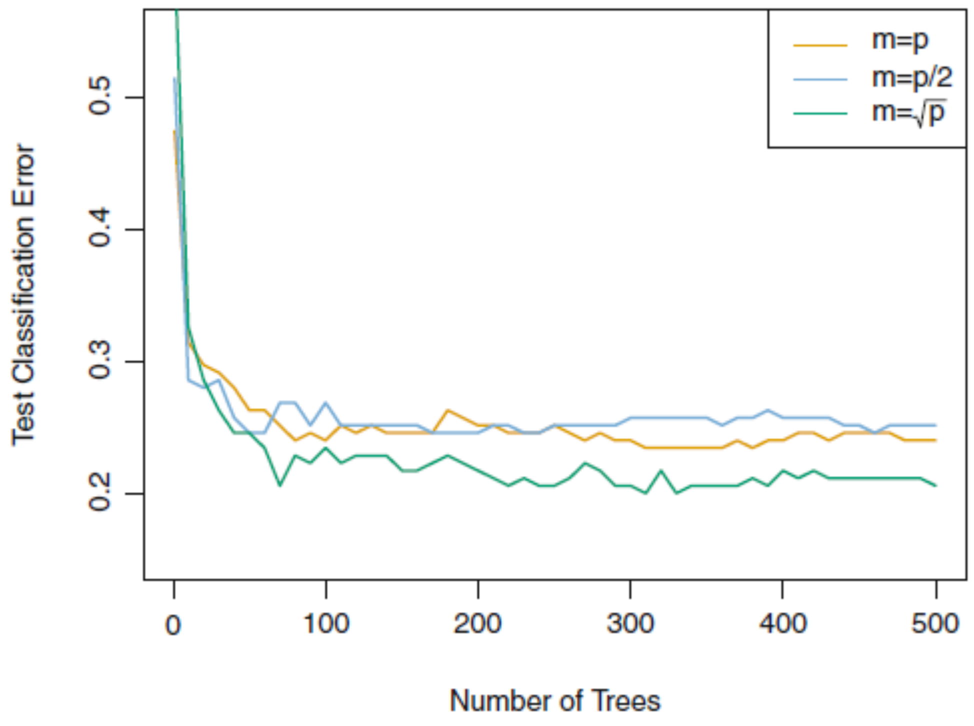


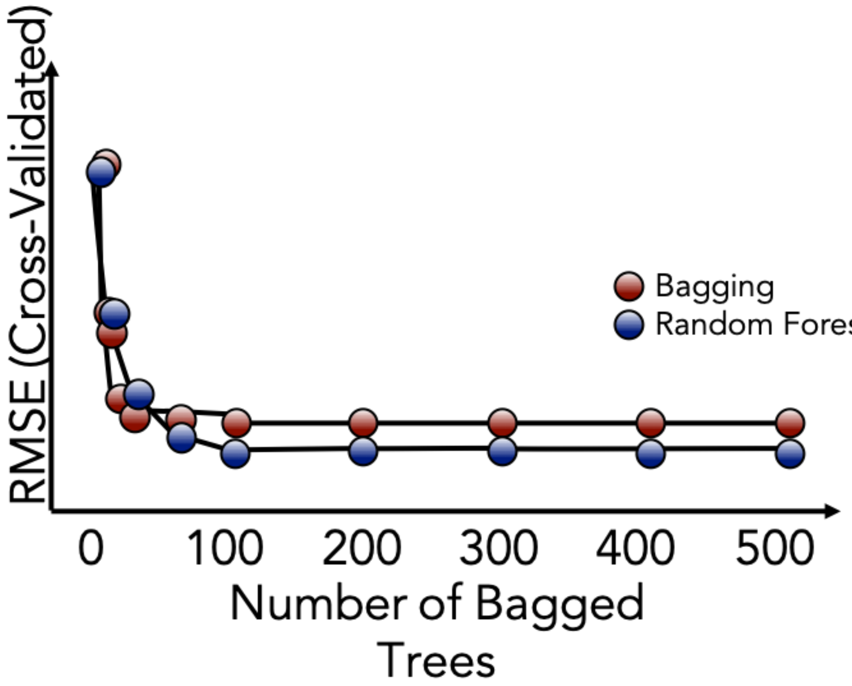
Figure 8.10 (James et al.)

What's a good number of trees?

What's a good number of subset of variables?

RANDOM FOREST VS. BAGGING

- Errors are further reduced for RF compared to Bagging
- Grow enough trees until error settles down
- Additional trees won't improve results

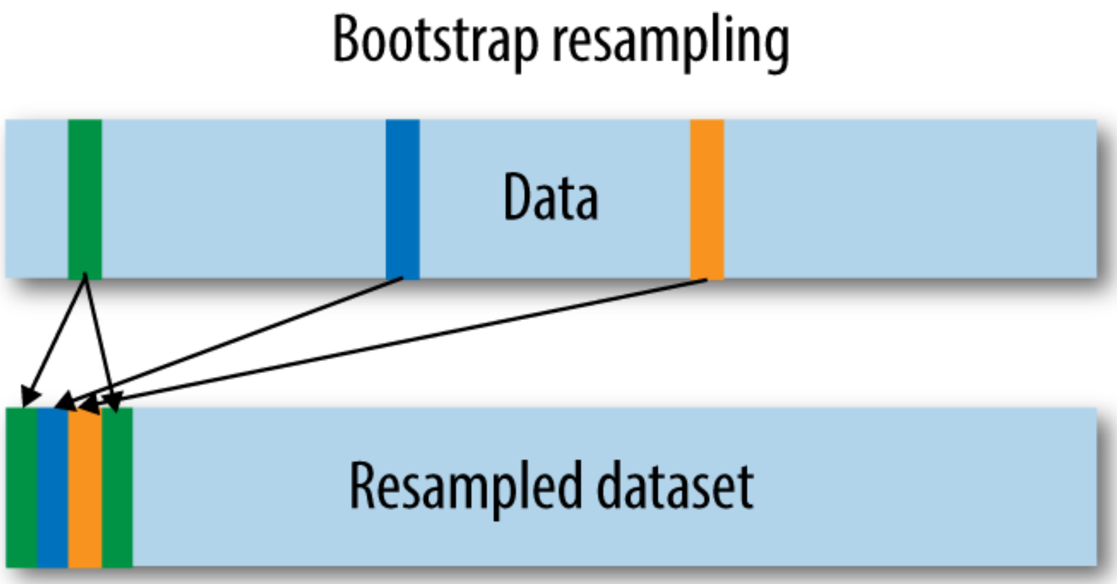


HOW TO EVALUATE RANDOM FOREST?

- Cross-validation error can be expensive to compute
- An alternative method: Out of bag (OOB) error

BOOTSTRAP: NUMBER OF POINTS

What's the probability of a data point belonging to a bootstrap sample/dataset?



BOOTSTRAP: NUMBER OF POINTS

- Sampling with replacement from N samples

$$\Pr(i \in B) = 1 - (1 - \frac{1}{N})^N$$
$$\approx 0.632$$

- Each bootstrap sample will contain roughly 63.2% of the original instances
- Roughly 36.8% samples will not be sampled

OUT OF BAG (OOB) SAMPLES

- Out of Bag (OOB) samples are those not in the bootstrap
- For each observation *i*, construct its prediction by averaging those trees corresponding to bootstrap samples not containing *i* (in which *i* is an OOB)
- OOB error estimates almost identical to k-fold cross-validation (leave-one-out cross validation)
- Once OOB stabilizes, training can be stopped

EXAMPLE: OOB ERROR

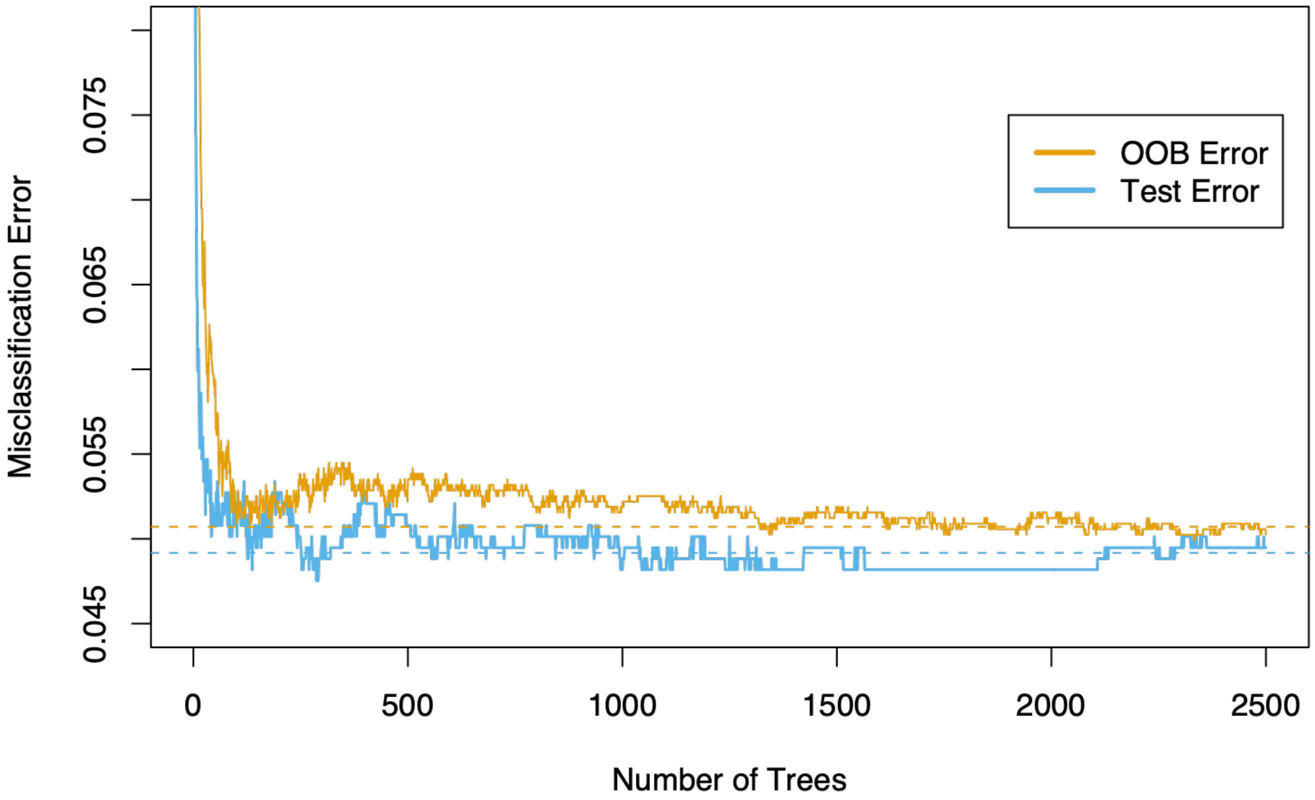


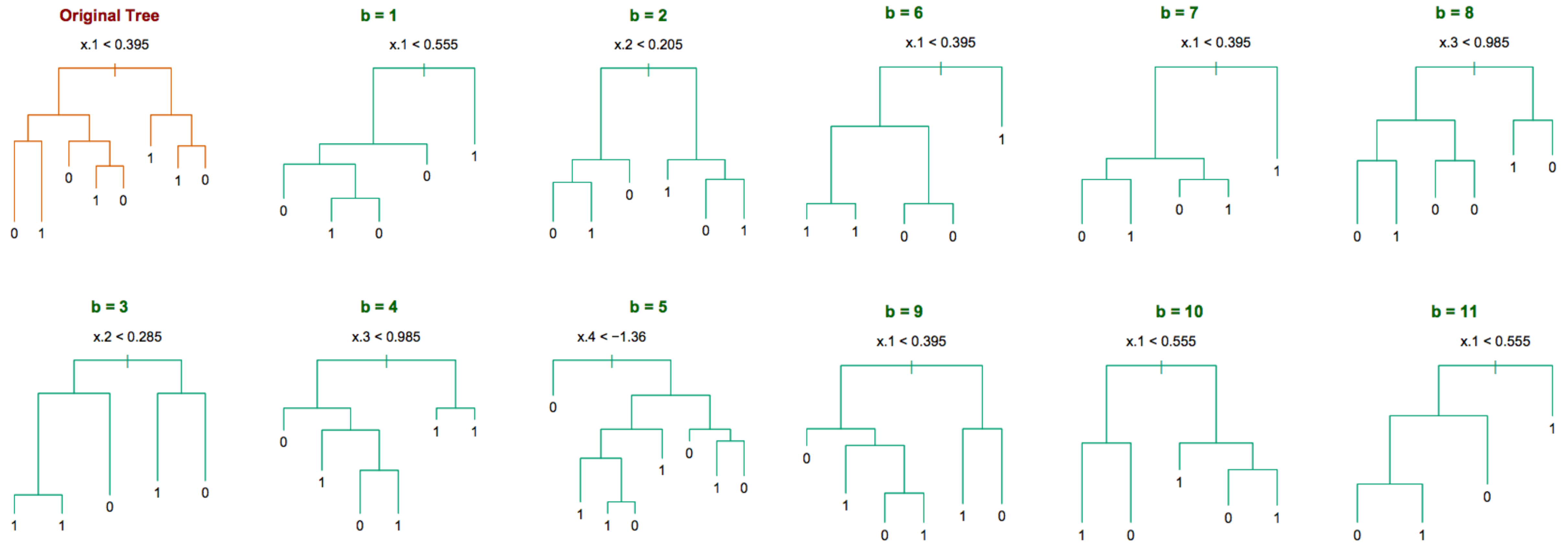
Figure 15.4 (Hastie et al.)

RANDOM FOREST VS DECISION TREE

- Reduced variance and improved performance
- Lose interpretability

How to evaluate the importance of each feature?

WHICH FEATURES ARE MOST IMPORTANT?



TREES: VARIABLE IMPORTANCE

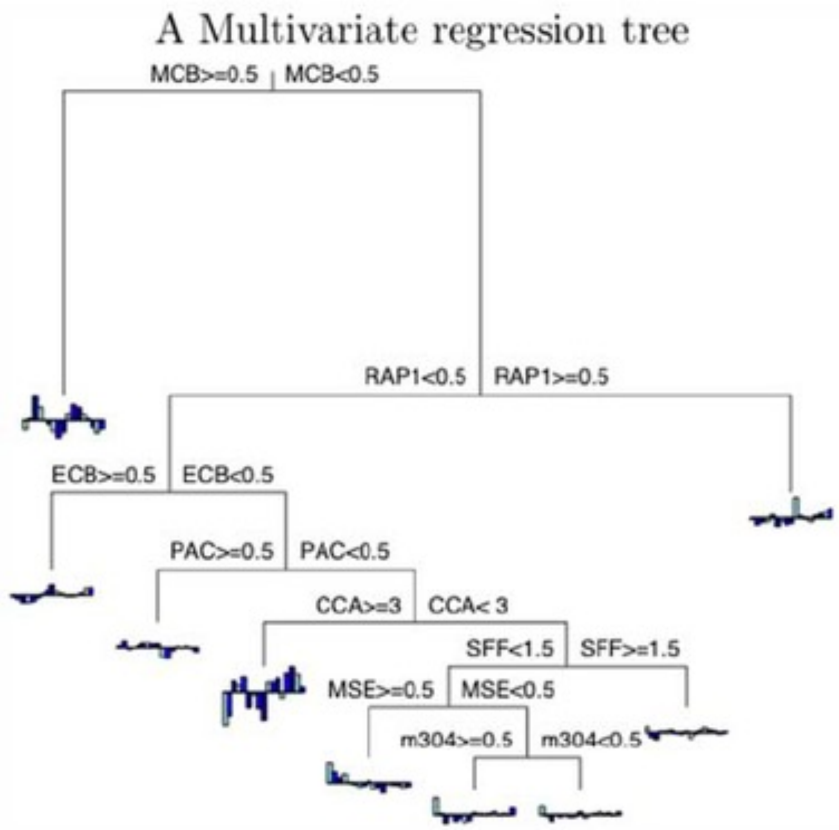
- Squared importance for variable j

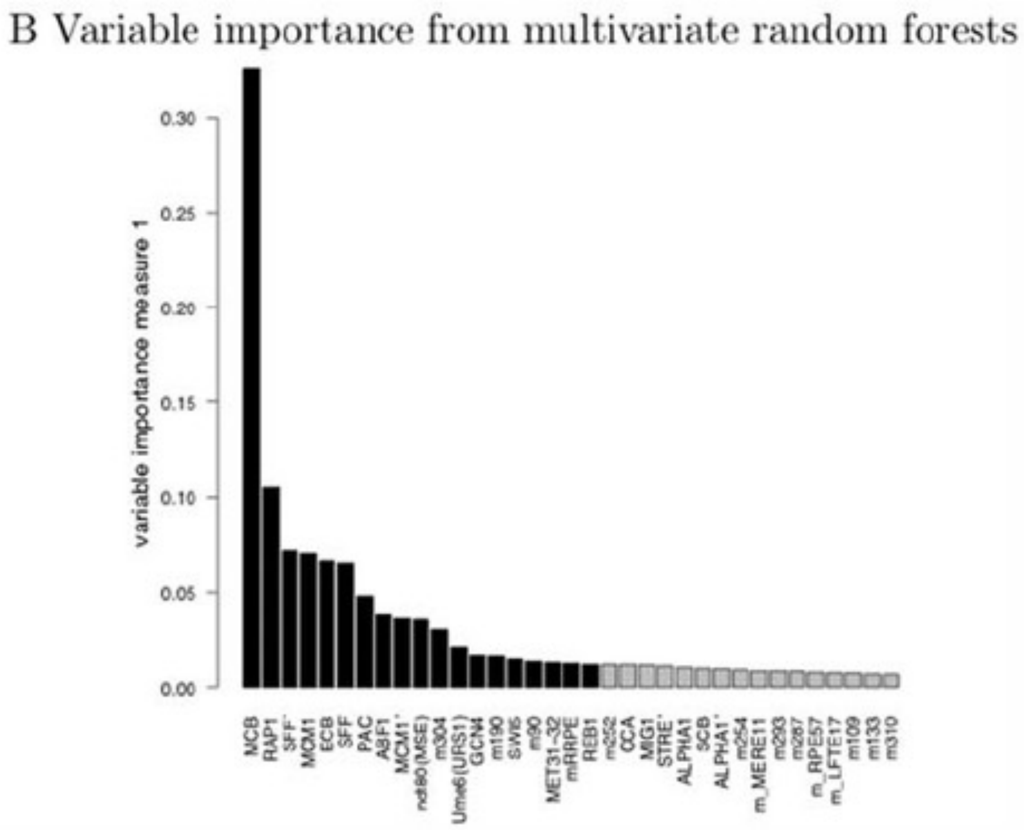
$$\mathrm{Imp}_j^2(\hat{f}^{\text{tree}}) = \sum_{k=1}^{m} \hat{d}_k \mathbb{1}_{\{\text{split at node } k \text{ is on variable j}\}}$$

- m is number of internal modes (non-leaves)

- $\hat{d}_k$ is the improvement in RSS (regression) or misclassification/Gini/Entropy (classification) from making the split

EXAMPLE: VARIABLE IMPORTANCE





https://www.researchgate.net/figure/A-Multivariate-regression-tree-and-the-variable-importance-measures-from-MRF-for-the-cell_fig2_26308068

FOREST: VARIABLE IMPORTANCE

- Average squared importance over all fitted trees

$$\mathrm{Imp}_j^2(\hat{f}^{\text{boost}}) = \frac{1}{M}\sum_{m=1}^{M}\mathrm{Imp}_j^2(\hat{f}_m^{\text{tree}})$$

- Stabilizes variable importances —> more accurate than for single tree

- Relative importance: Scale largest importance to 100 and scale all other variable importances accordingly

What are the drawbacks of this importance?

VARIABLE IMPORTANCE: IMPURITY BASED

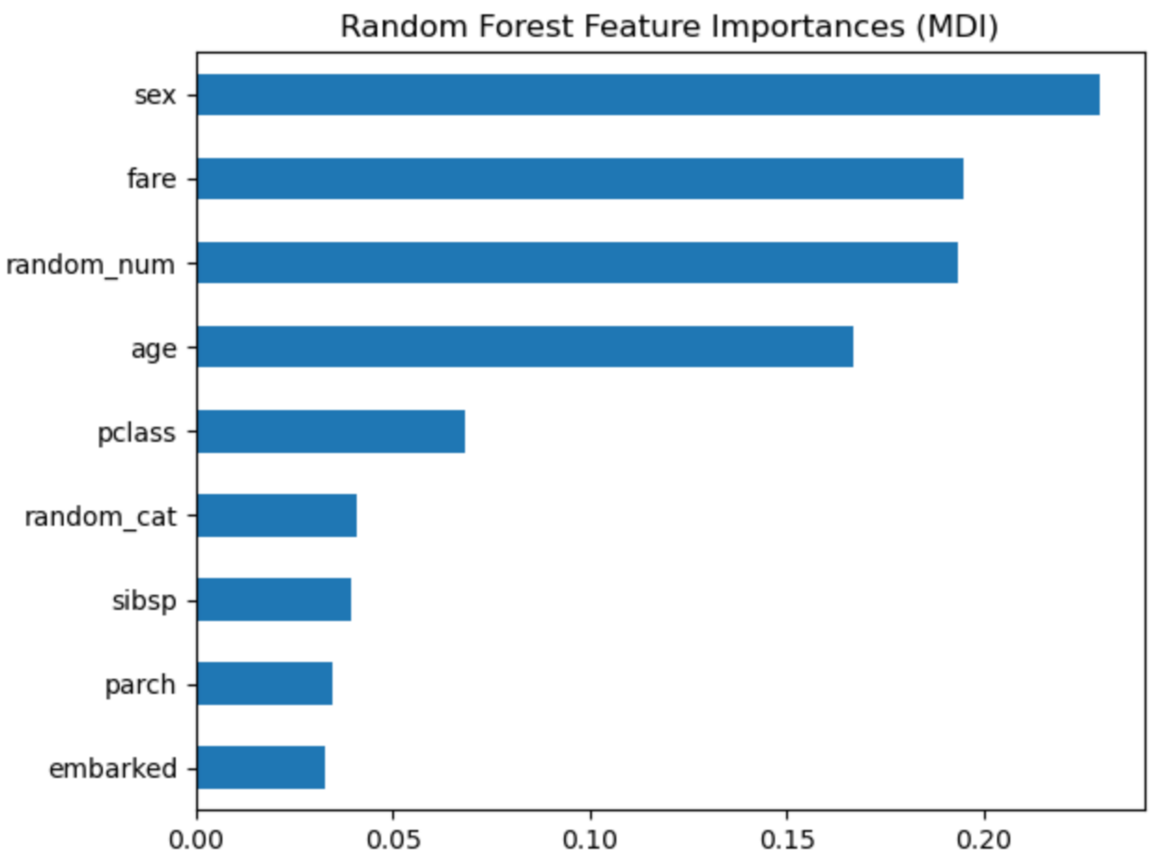
- Biased towards high cardinality features
- Computed on training set statistics and do not reflect the ability of feature to generalize to the test set

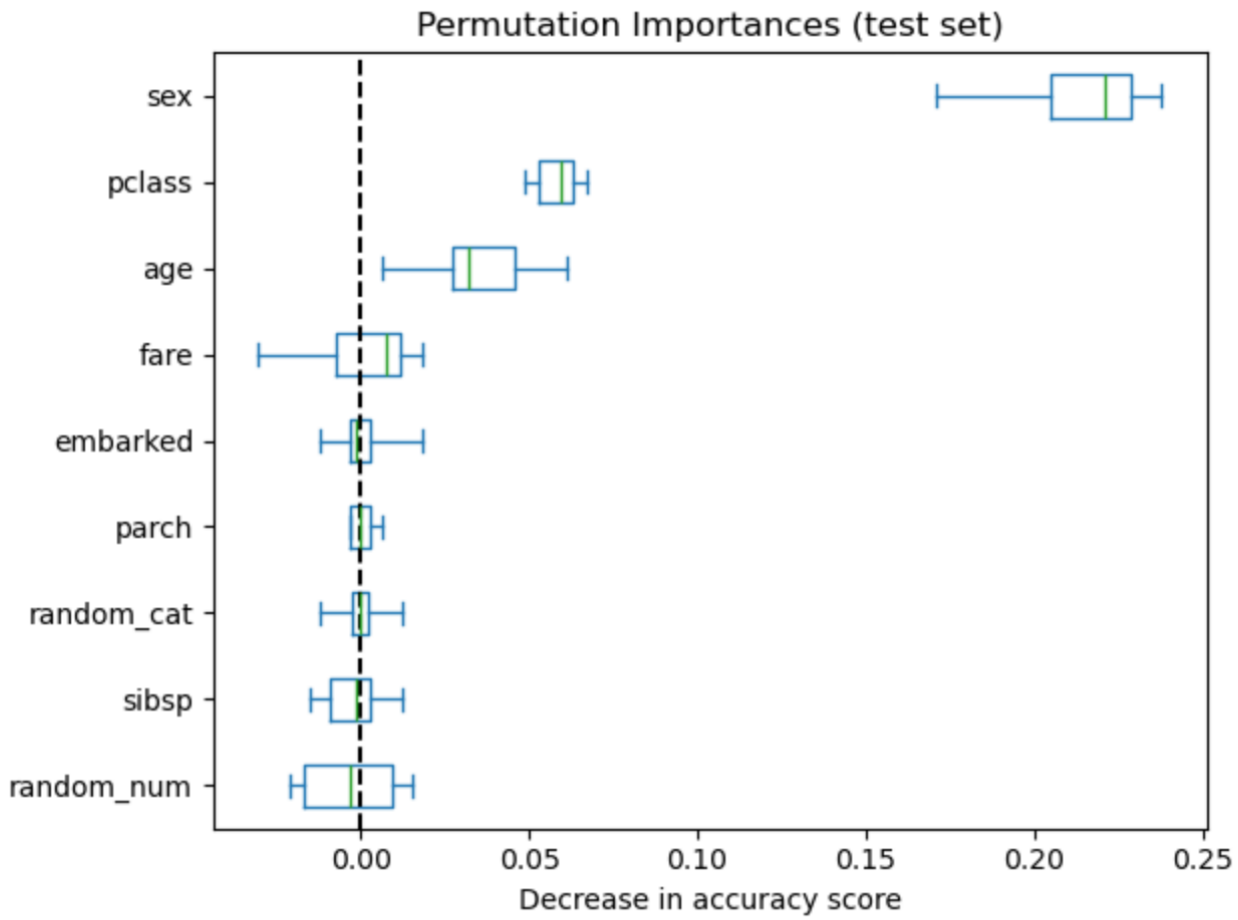
VARIABLE IMPORTANCE: PERMUTATION BASED

- For bth tree, OOB samples are passed down tree and accuracy recorded
- Values for jth variable are randomly permuted in OOB samples and accuracy again computed
- Decrease in accuracy is used as measure of importance (marginal contribution of the feature)

https://scikit-learn.org/stable/modules/generated/sklearn.inspection.permutation_importance.html#sklearn.inspection.permutation_importance

FEATURE IMPORTANCE





https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance.html#sphx-glr-auto-examples-inspection-plot-permutation-importance-py

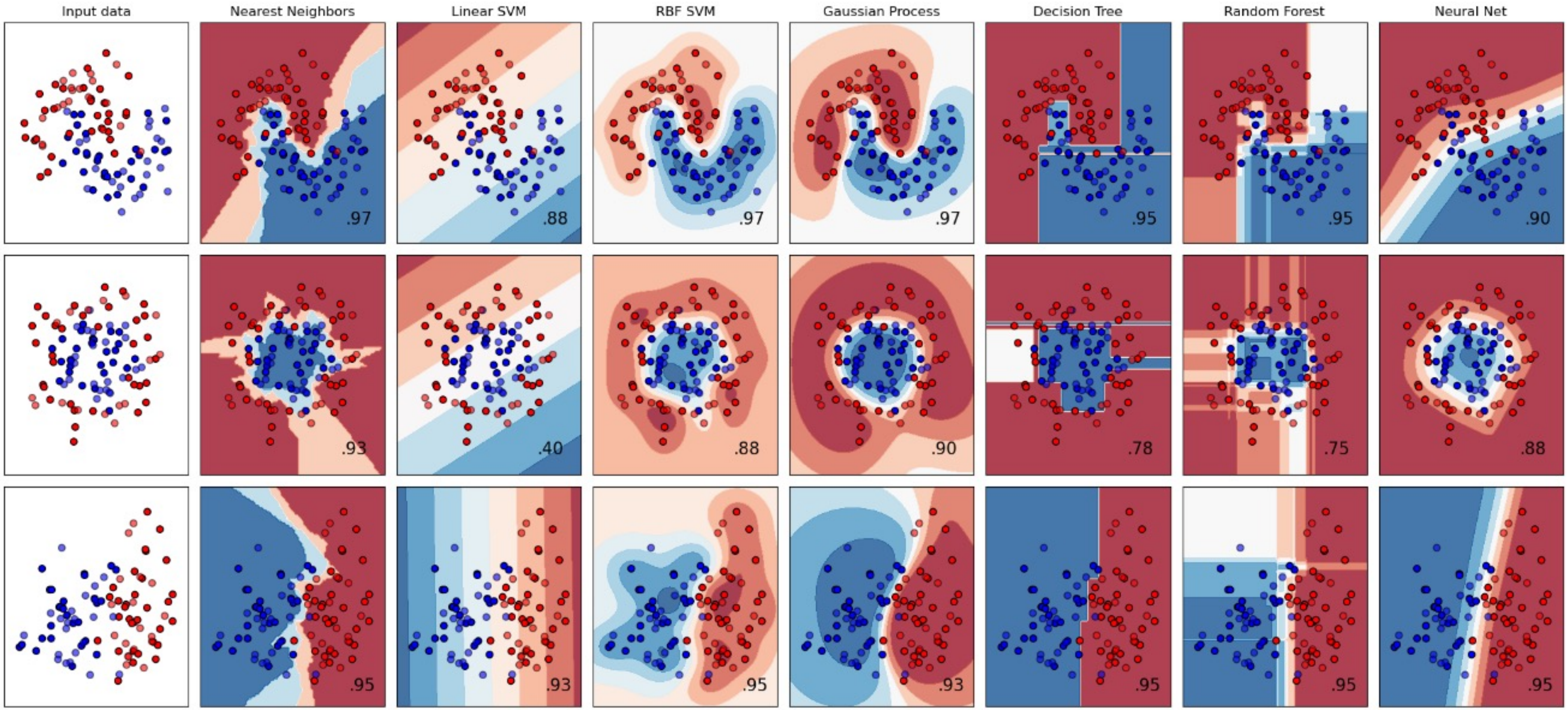
RANDOM FOREST: ADVANTAGES

- State of the art method, one of the most accurate general-purpose learners available
- Handles a large number of input variables without overfitting (variance reduction)
- Robust to errors and outliers
- Can model non-linear boundaries
- Gives variable importance and out of bag error rates
- Easy to train and tune, easily parallelized by training

RANDOM FOREST: DISADVANTAGES

- Loss of interpretability (no decision rules)
- Difficult to analyze as an algorithm and mathematical properties still largely unknown
- Large number of trees is memory-intensive
- Bias towards categorical variables with larger number of levels

RANDOM FOREST



https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html#sphx-glr-auto-examples-classification-plot-classifier-comparison-py

PREVIEW: HOMEWORK #5

- Almost Random Forest
- Instead of choosing a random subset of features for each split, choose a random subset of features that the tree will be created on (the same subset is used as candidates from all splits)

SKLEARN: RANDOM FOREST

- sklearn.ensemble.RandomForestClassifier
  - n_estimators, default=100
  - max_features: {“sqrt”, “log2”, None}, default=”sqrt”

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html