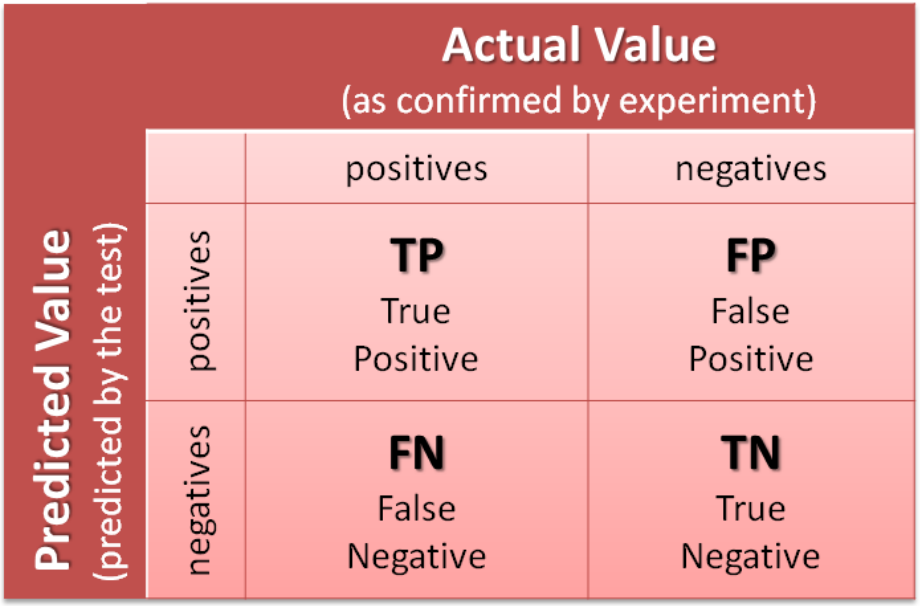
Biology 644

Old Title: Bioinformatics for Molecular Biologists

Potential New Title: Integrated Bioinformatics Using R for Both Wet and Dry Scientists

Classification – Confusion Matrix



FP = "Falsely labelled as a positive"

FN = "Falsely labelled as a negative"

- A classification model (classifier or diagnosis) is a mapping of instances to certain distinct classes or groups.
- The classifier or diagnosis result can be a real value (continuous output), in which case the classifier boundary between classes must be determined by a threshold value (for instance, to determine whether a person has hypertension based on a blood pressure measure).
- Or it can be a discrete class label (discrete output), indicating one of the classes.

Sensitivity, Specificity, Efficiency

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

- Sensitivity
  - The proportion of true positives: the ability of the system to correctly predict the condition in cases it is really present.
  - SENS = POSITIVE HITS / TOTAL POSITIVES
  - = TP / (TP + FN)

- Specificity
  - The proportion of true negatives: the ability of the system to correctly predict the absence of the condition in cases it is not present.
  - SPEC = NEGATIVE HITS / TOTAL NEGATIVES
  - = TN / (TN + FP)

- Efficiency
  - The mean of Sensitivity and Specificity.
  - In practical situations, sensibility and specificity vary in reverse directions.
  - Generally, when a method is too responsive to positives, it tends to produce many false positives, and vice versa.
  - Therefore, a perfect classifier (with 100% sensitivity and 100% specificity) rarely is conceived, and a balance between both must be obtained
  - EFF = (SENS + SPEC) / 2

Accuracy, Positive Predictive Value, Negative Predictive Value

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

- **Accuracy**
  - The proportion of correct predictions, without considering what is positive and what is negative.
  - This measure is highly dependent on the data set distribution and can easily lead to wrong conclusions about the system performance.
  - ACC = TOTAL HITS / NUMBER OF ENTRIES IN THE SET
  - = (TP + TN) / (P + N)

- **Positive Predictive Value**
  - The proportion of true positives in contrast with all positive predictions.
  - This measure is highly susceptible to the prevalence in the data set, but gives an estimate on how good the system is when making a positive affirmation.
  - It also can easily lead to wrong conclusions about system performance.
  - PPV = POSITIVE HITS / TOTAL POSITIVE PREDICTIONS
  - = TP / (TP + FP)

- **Negative Predictive Value**
  - The proportion of true negatives in contrast with all negative predictions.
  - This measure is highly susceptible to the prevalence in the data set but gives an estimate on how good the system is when making a negative affirmation.
  - It can easily lead to wrong conclusions about system performance.
  - NPV = NEGATIVE HITS / TOTAL NEGATIVE PREDICTIONS
  - = TN / (TN + FN)

Confusion Matrix II

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

		Condition (as determined by "Gold standard")		
		Condition positive	Condition negative	
Test outcome	Test outcome positive	**True positive**	**False positive** (Type I error)	Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$
	Test outcome negative	**False negative** (Type II error)	**True negative**	Negative predictive value = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$
		Sensitivity = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	Specificity = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Accuracy

More Confusion

		Condition (as determined by "Gold standard")		
		Condition positive	Condition negative	
Test outcome	Test outcome positive	**True positive**	**False positive** (Type I error)	Precision = Σ True positive / Σ Test outcome positive
	Test outcome negative	**False negative** (Type II error)	**True negative**	Negative predictive value = Σ True negative / Σ Test outcome negative
		Sensitivity = Σ True positive / Σ Condition positive	Specificity = Σ True negative / Σ Condition negative	Accuracy

true positive (TP)

eqv. with hit

true negative (TN)

eqv. with correct rejection

false positive (FP)

eqv. with false alarm, Type I error

false negative (FN)

eqv. with miss, Type II error

---

sensitivity or true positive rate (TPR)

eqv. with hit rate, recall

$$TPR = TP/P = TP/(TP + FN)$$

specificity (SPC) or True Negative Rate

$$SPC = TN/N = TN/(FP + TN)$$

precision or positive predictive value (PPV)

$$PPV = TP/(TP + FP)$$

negative predictive value (NPV)

$$NPV = TN/(TN + FN)$$

fall-out or false positive rate (FPR)

$$FPR = FP/N = FP/(FP + TN)$$

false discovery rate (FDR)

$$FDR = FP/(FP + TP) = 1 - PPV$$

Miss Rate or False Negative Rate (FNR)

$$FNR = FN/(FN + TP)$$

---

accuracy (ACC)

$$ACC = (TP + TN)/(P + N)$$

F1 score

is the harmonic mean of precision and sensitivity

$$F1 = 2TP/(2TP + FP + FN)$$

Matthews correlation coefficient (MCC)

$$\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Informedness = Sensitivity + Specificity - 1

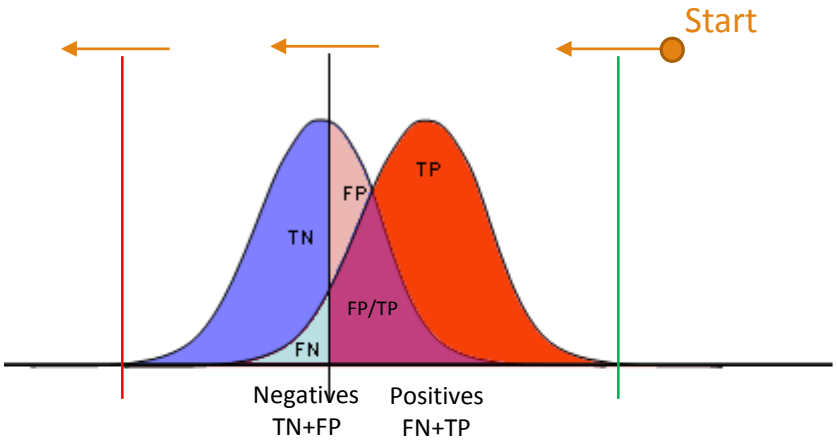
Markedness = Precision + NPV - 1

The Receiver Operating Characteristic (ROC) Curve

- Developed by electrical and radar system engineers during World War II to detect enemy combatants in the air or battlefield –where the "friend or foe" classification is a "characteristic" received by an operator of the machine.
- Used in medicine, radiology, psychology and other areas for many decades. More recently introduced to areas such as machine learning, data mining, and bioinformatics
- Because the output from classification systems are generally continuous, it is necessary to define a cutoff value, or discriminatory threshold, to classify and count the number of positive and negative predictions (such as positive or negative diagnostics in the case of pathology occurrence).
- As this threshold can be arbitrarily determined, the best practice to compare the performance of different systems is to study the effect of selecting the full spectrum of cutoff values over the output data.
- Considering many cutoff values, it is possible to calculate a set of pairs (sensitivity, 1-specificity) which can then be plotted in a curve.
- This curve is called the ROC curve for the system, having sensitivity values on the y-axis and the complement of specificity (1-specificity) on the x-axis.

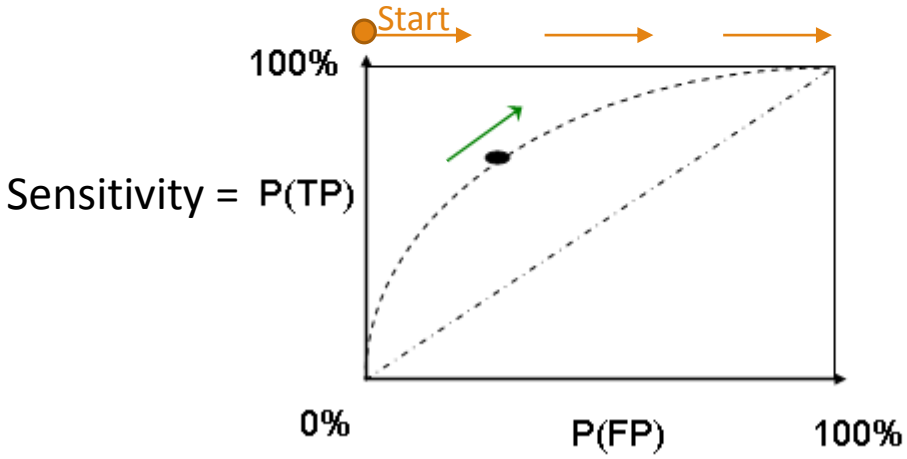
The Receiver Operating Characteristic (ROC) Curve

Different Classifier Thresholds



Confusion Matrix

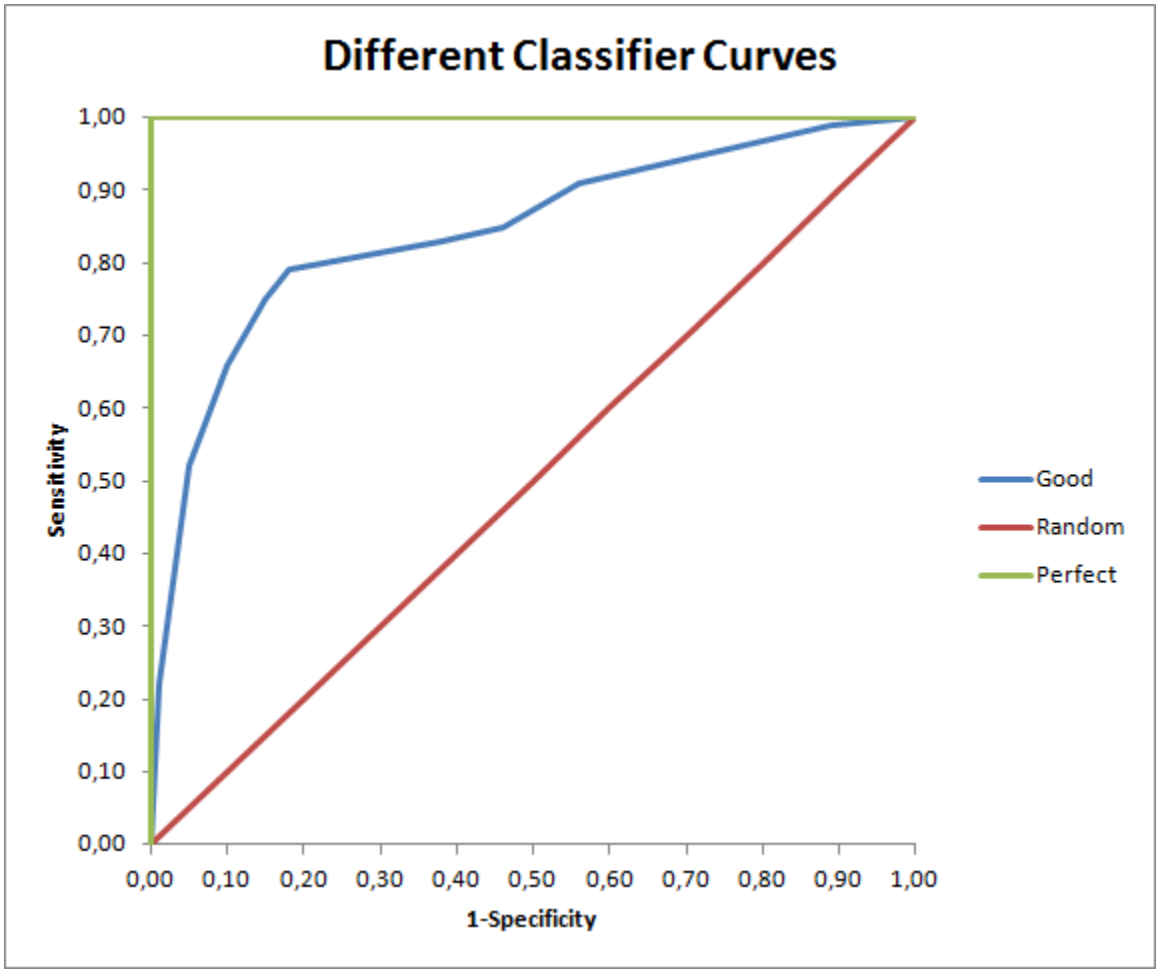
Actual Positive Set	Actual Negative Set	
TP	FP	Predicted Positive
FN	TN	Predicted Negative
1	1	



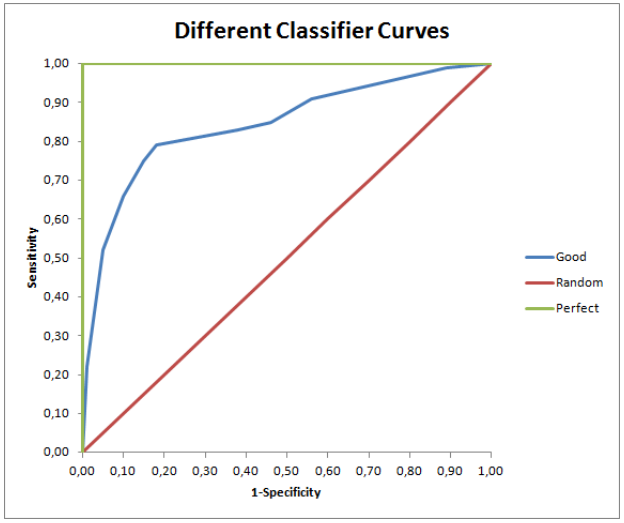
Graphs the change in TP relative to the change in FP as the threshold sweeps from right to left

1 – Specificity = 1 - P(TN) = P(FP)

The Receiver Operating Characteristic (ROC) Curve



The Area Under an ROC Curve (AUROC)



- The accuracy of a classifier depends on how well it separates the group being tested into those with and without the disease/phenotype in question.
- A common non-parametric accuracy measurement is given by the Area Under the ROC curve (AUROC) – with an area of 1 representing a perfect classifier and an area of .5 representing a worthless one.
- A rough guide for classifying the accuracy of a diagnostic test is the traditional academic point system:
  - .90-1.0 = excellent (A)
  - .80-.90 = good (B)
  - .70-.80 = fair (C)
  - .60-.70 = poor (D)
  - .50-.60 = fail (F)

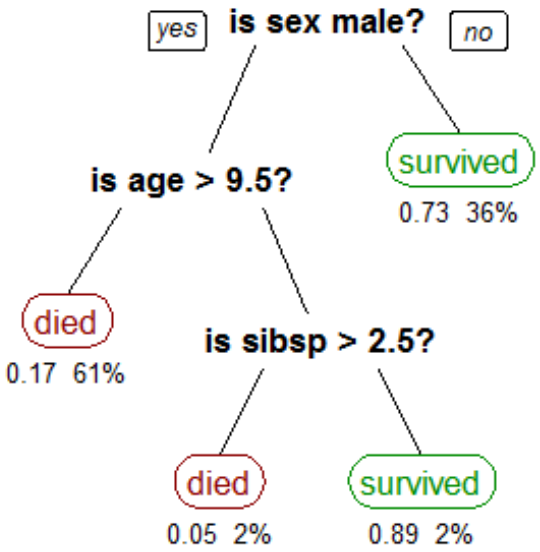
Decision Tree Learning

- A method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables.
- Decision tree learning is one of the most successful techniques for supervised classification learning.
- Each internal (non-leaf) node is labeled with an input feature.
- The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature.
- Each leaf of the tree is labeled with a class or a probability distribution over the classes.
- A tree can be learned by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning.
- The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions.
- This process of top-down induction of decision trees (TDIDT) is an example of a greedy algorithm, and it is by far the most common strategy.

Classification And Regression Trees (CARTs)

- The term Classification And Regression Tree (CART) analysis is an umbrella term used to refer to the two main types Decision Trees :
  - Classification tree analysis is when the predicted outcome is the class to which the data belongs. (e.g. cancerous or non-cancerous)
  - Regression tree analysis is when the predicted outcome is a continuous real number (e.g. rate of tumor cell proliferation, or a patient's blood pressure).
- Regression and classification trees have some similarities but also some differences, such as the procedure used to determine where to split
- Both use Boolean tests at each split to partition the data into classifications

Surviving the Titanic Classification Tree



sibsp = sibling or 2 parents

- The "sibsp" feature is the number of siblings or spouses on board
- Each interior node corresponds to one of the input variables
- There are edges to children for each of the possible values of that input variable.
- Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.
- The stats under the leaves show the probability of survival and the percentage of observations in the leaf.

Advantages of Single Decision Trees

- Small trees are simple to understand and interpret.
- Requires little data preparation. Other techniques often require data normalization or transformation
    - Easily ignores redundant variables
    - Handles missing data elegantly
- Able to handle both numerical and categorical data.
- Uses a white box model where any condition is easily explained by boolean logic.
    - In a black box model (i.e. an artificial neural network) the results are difficult to understand, even if it's a good classifier.
- Possible to validate a model using statistical tests.

Disadvantages of Single Decision Trees

- The problem of learning an optimal decision tree is known to be NP-complete and cannot be solved in sub-polynomial time
- Consequently, practical decision-tree learning algorithms are based on heuristics such as the greedy algorithm where locally-optimal decisions are made at each node.
- Therefore, not guaranteed to return the globally-optimal decision tree.
- Overfitting - can create overly complex single trees that do not generalize well to new data
- Single decision trees do not generalize well and have poor predictive performance
- Large trees are hard to interpret
- Nonlinear relationships among the data can be hard to learn because decision trees do not always capture them easily.
    - In such cases, the decision tree can become prohibitively large

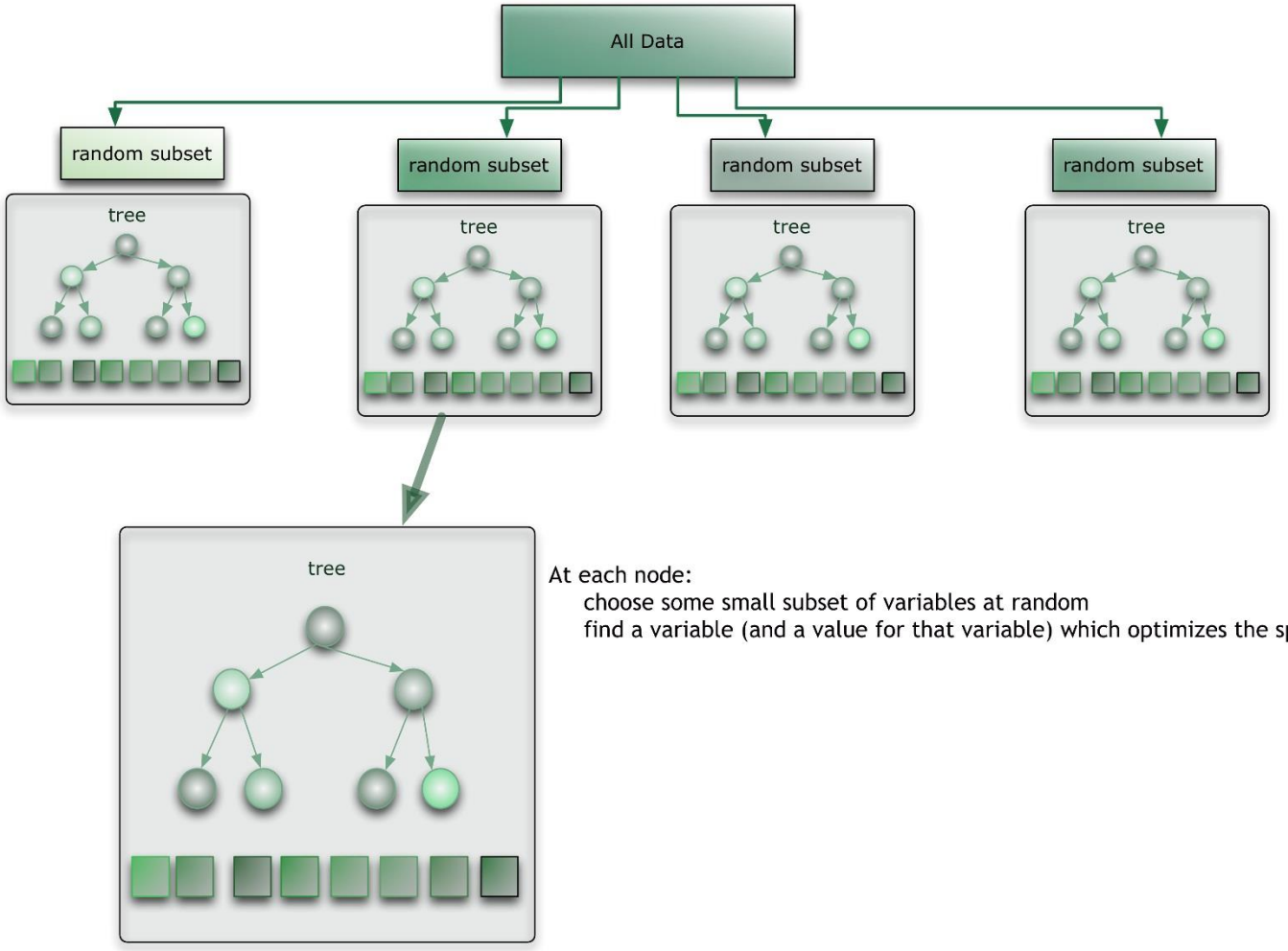
Ensemble Methods

- Ensemble methods construct many simple decision trees and employ a voting scheme with the goal of creating a better aggregate classifier:
  - Bootstrap aggregating (Bagging) decision trees - an early ensemble method, builds multiple decision trees by repeatedly resampling training data with replacement, and voting with the trees for a consensus prediction.
  - Random Forest – similar to bagging trees, but also performs resampling of the feature space (without replacement) at each candidate split in the learning process.
    - Typically, for a dataset with p features, $\sqrt{p}$ features are used in each split
    - Makes the classifier more robust to noisy, false features that just happen to correlate well with the outcome just by chance.
  - (Gradient) Boosted Trees – trees built sequentially while assigning higher weights to (boosting) the training data that was difficult to classify in the previous step
    - Iteratively re-weight the data such that the misclassified cases get higher weights
    - Combine predictions from each model (weighted by accuracy of the models)
  - Rotation Forest – resampling on just features (not data) and first applying PCA on the random subset of the input features before creating each tree

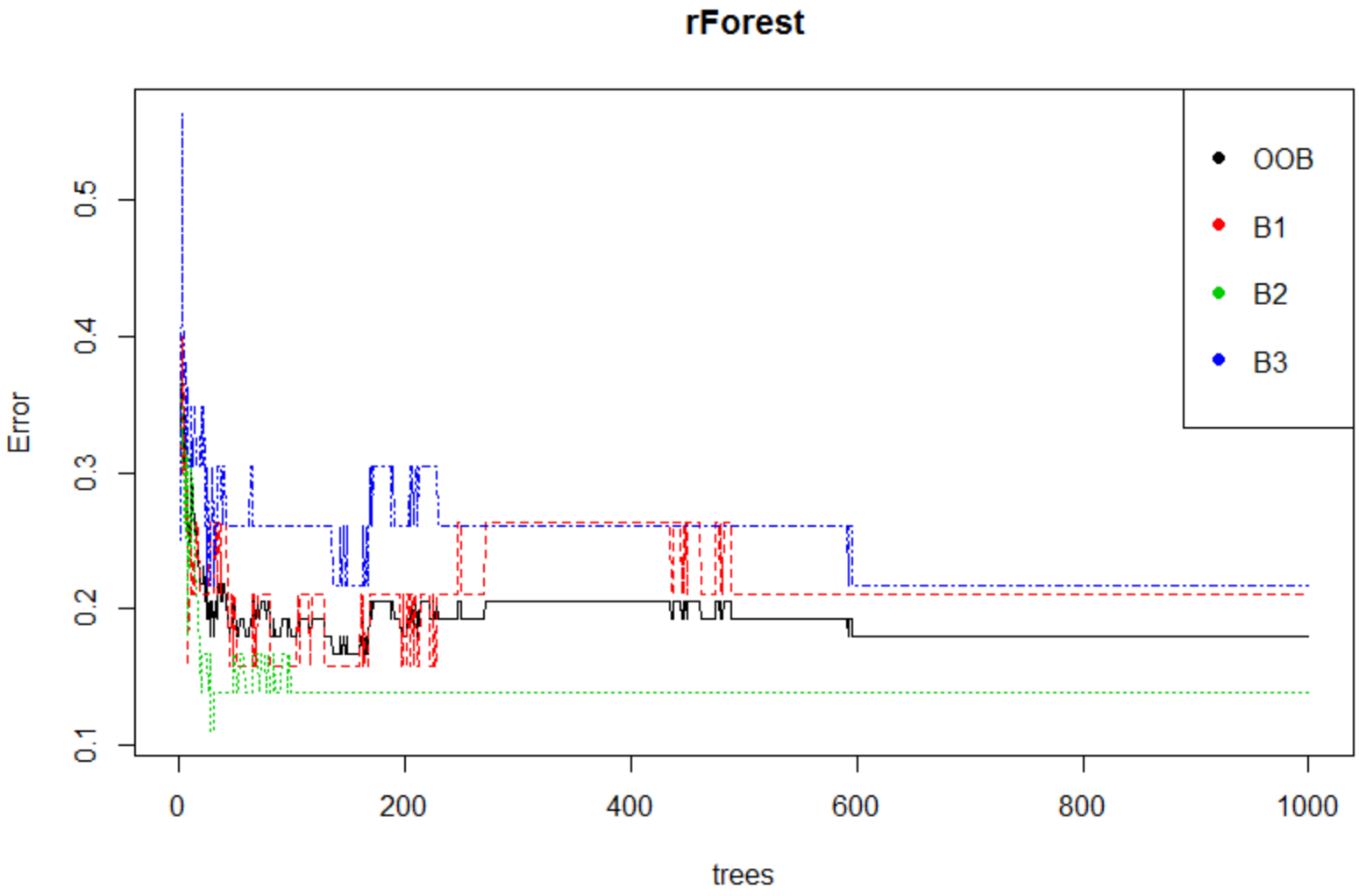
Random Forests

- Each tree gives a classification and submits one "vote" for its predicted class.
- The forest chooses the classification having the most votes.
- Each tree is grown as follows:
  - First, sample N cases at random with replacement, from the original data of size N. (Bagging)
  - Next, at each node m, variables are selected at random (without replacement) out of the M total variables (m << M) and the best split on these m is used to split the node. (Bagging on Features)
- The value of m is held constant during the forest growing.
- Each tree is grown to the largest extent possible. There is no pruning.
- An out-of-bag (oob) error estimate is generated during training:
  - When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample.
  - This oob (out-of-bag) data is used to get a running unbiased estimate of the classification error as trees are added to the forest.
- Also uses the oob data to measure variable importance through permutation
  - randomly permute the values of variable m in the oob cases and then count the change in correct classifications (without changing the decision trees)

Random Forests



Random Forests – on the Chiaretti et al. Data



Advantages to Random Forests

- have good predictive power
- Do not overfit and generalize well to other data sets. You can grow as many trees as you want.
- Robust to noise and perform well with large datasets
- With the out-of-bag (oob) error estimate, there is no need for additional cross-validation or a separate test set to get an unbiased estimate of the test set error
- The out-of-bag (oob) error estimate can also be used as a termination criterion to stop growing more trees
- Also measures the importance of each input variable in the predictions
- Easily parallelizable – can train each tree on a different processor