Biostat 202: Opportunities and Challenges of "Big Data": Machine Learning 4: Cross-Validation, Ensembles, and Feature Importance

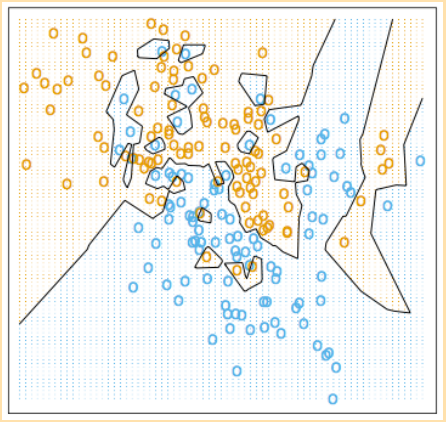
Aaron Wolfe Scheffler

Division of Biostatistics

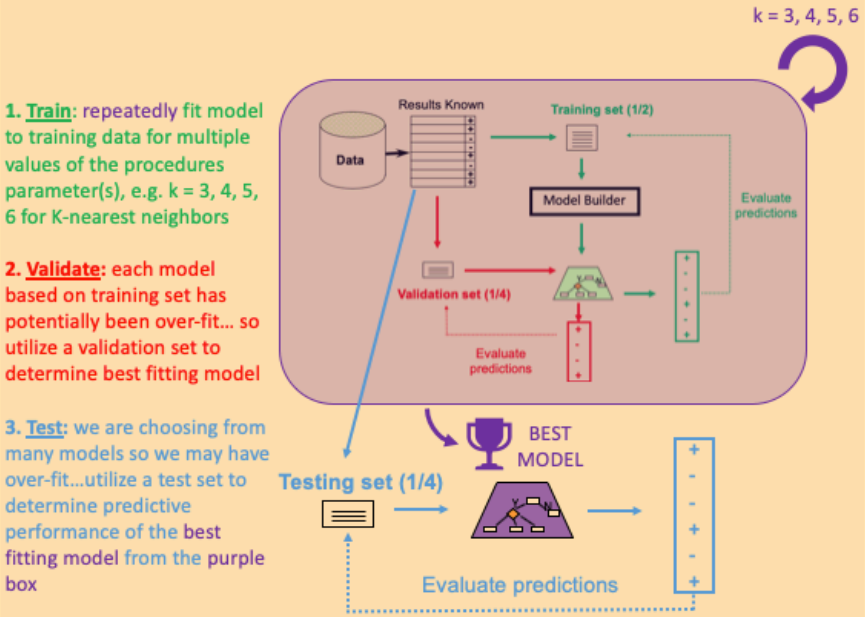
Department of Epidemiology and Biostatistics

Review of last lecture

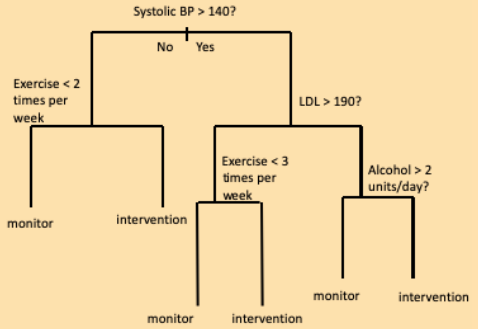
1. K-nearest neighbors



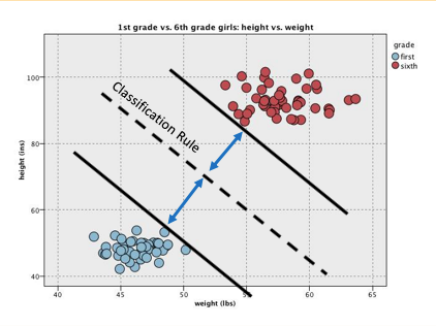
2. Train/validate/test



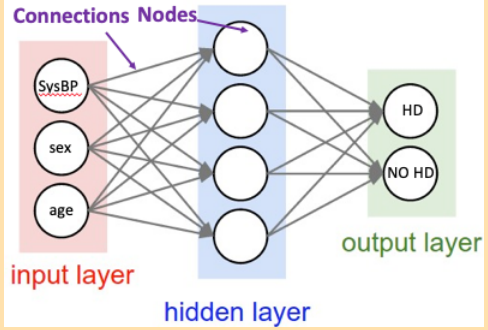
3. Classification trees



4. Support vector machines



5. Artificial neural networks



Outline

- Cross-validation
- Ensemble methods
  - Bagging: *Random Forests*
  - Boosting: *Adaboost*
  - Stacking
- Parameter tuning, continued
- Feature importance

Cross-validation

Cross-validation

- For complex models and ensemble, parameter tuning is almost always required
- What if you don't have enough data to split 3 or even 2 ways for train/validate/test?
- If data is expensive, it is a big sacrifice to reserve a large proportion of data for validation and testing…
- **Solution**: cross-validation

Cross-validation

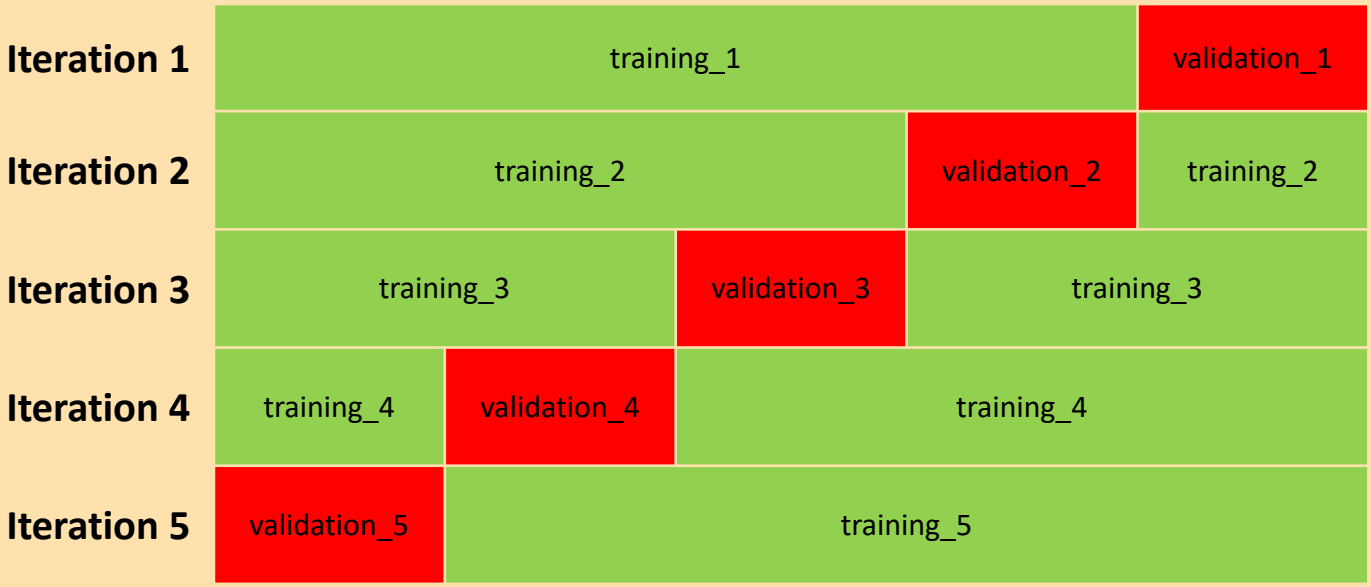
- *Cross-validation*
  - First step: data is split into *k* subsets (folds) of equal size
  - Second step: each subset is used in turn for validation (evaluation of prediction) and the remainder for training
- This is called *k-fold cross-validation* (10-fold is common choice, also 5-fold for smaller datasets approx. <200 and even *N-fold* for very small datasets, called leave-one-out)

E.g. 5-fold cross-validation

- 1. Break up data into 5 “folds” of the same size

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

- 2. Set aside one fold for validation and use the rest to build model, repeat this process for each fold



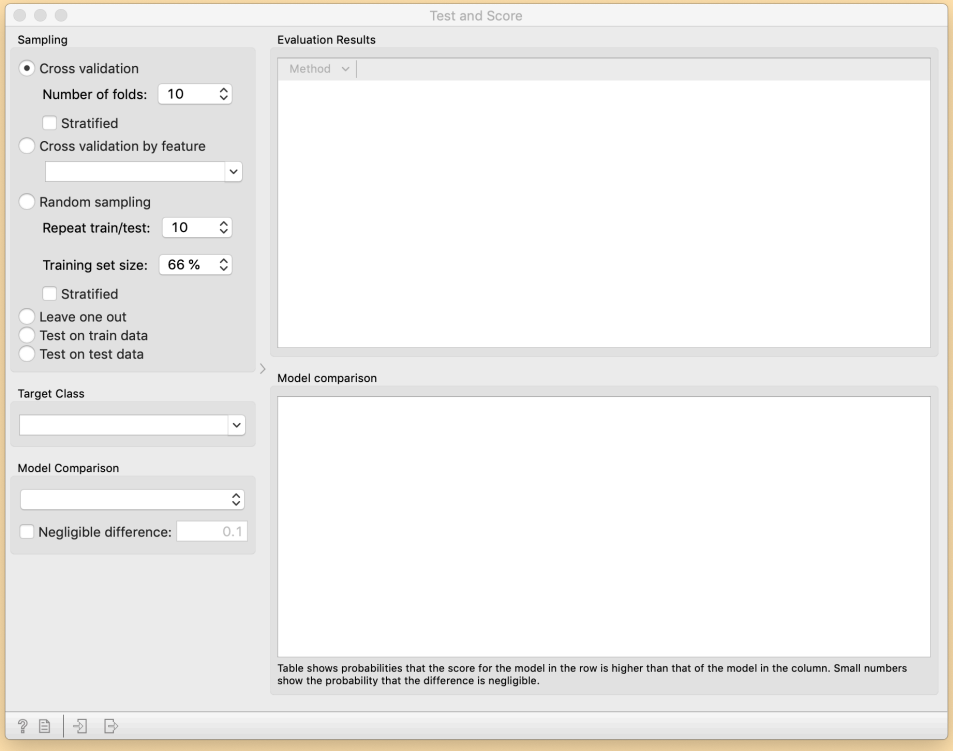
- Estimate the predictive performance of our model on the combined validation data (i.e. the validation_1-5 sets)

Cross-validation

- Notice that at no time is the predictive performance based on data that was used to train the model – so there is no over-fitting
- Use of cross validation
  - Single model assessment: replaces train/test
  - Multiple model assessment (i.e. hyperparameter tuning): replaces train/validation – STILL NEED TO ASSESS FINAL MODEL ON A TEST SET!

Cross-validation - Orange

- The evaluation metrics are averaged across folds to yield an overall assessment of predictive performance
- Often the subsets are stratified before the cross-validation is performed – e.g. same proportion of each class or group in each fold



Ensemble Methods

Ensemble methods

- Train a bunch of predictive models and pool them to form a final prediction.
- **Advantage:**
  - improvement in predictive accuracy
- **Disadvantages:**
  - more computation
  - it is difficult to understand an ensemble of classifiers
- Examples: bagging, boosting, and stacking

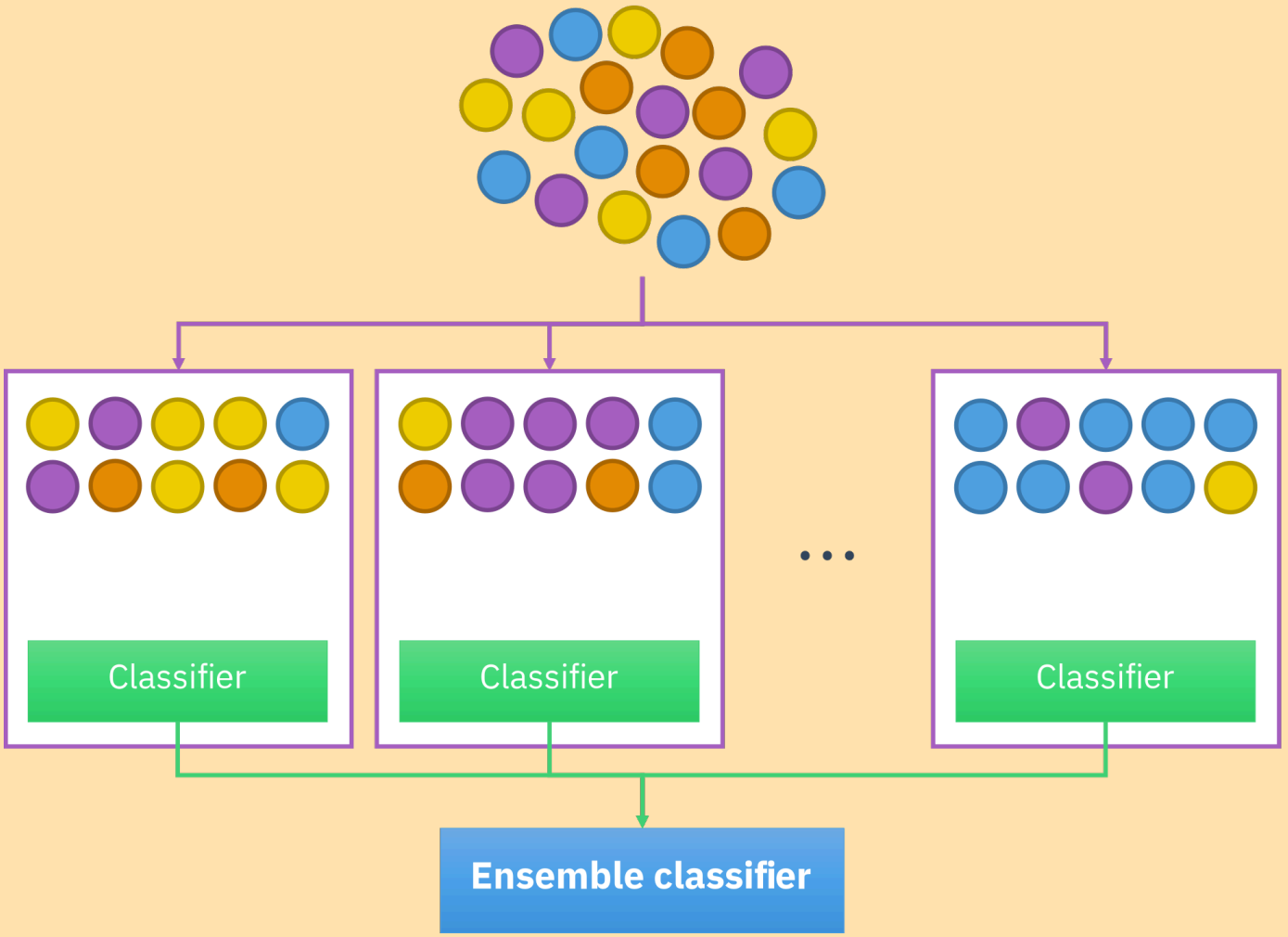
Bagging

- Bagging = “Bootstrap Aggregating”
- Simplest way of combining predictions from a similar class of models
- Train models on slightly different versions of the data and combine via voting or averaging with each model receiving equal weight
- You can bag any model, most common is decision trees (e.g. random forest)

Bagging

- Basic idea of bagging:
  1. Re-sample several training sets of size n from the dataset with replacement (bootstrap sample)
  2. Build a model for each training set
  3. Combine the models predictions (majority vote)
- Improves performance in almost all cases if learning scheme is *unstable* (e.g. decision trees depend heavily on where the first cut occurs)

Bagging



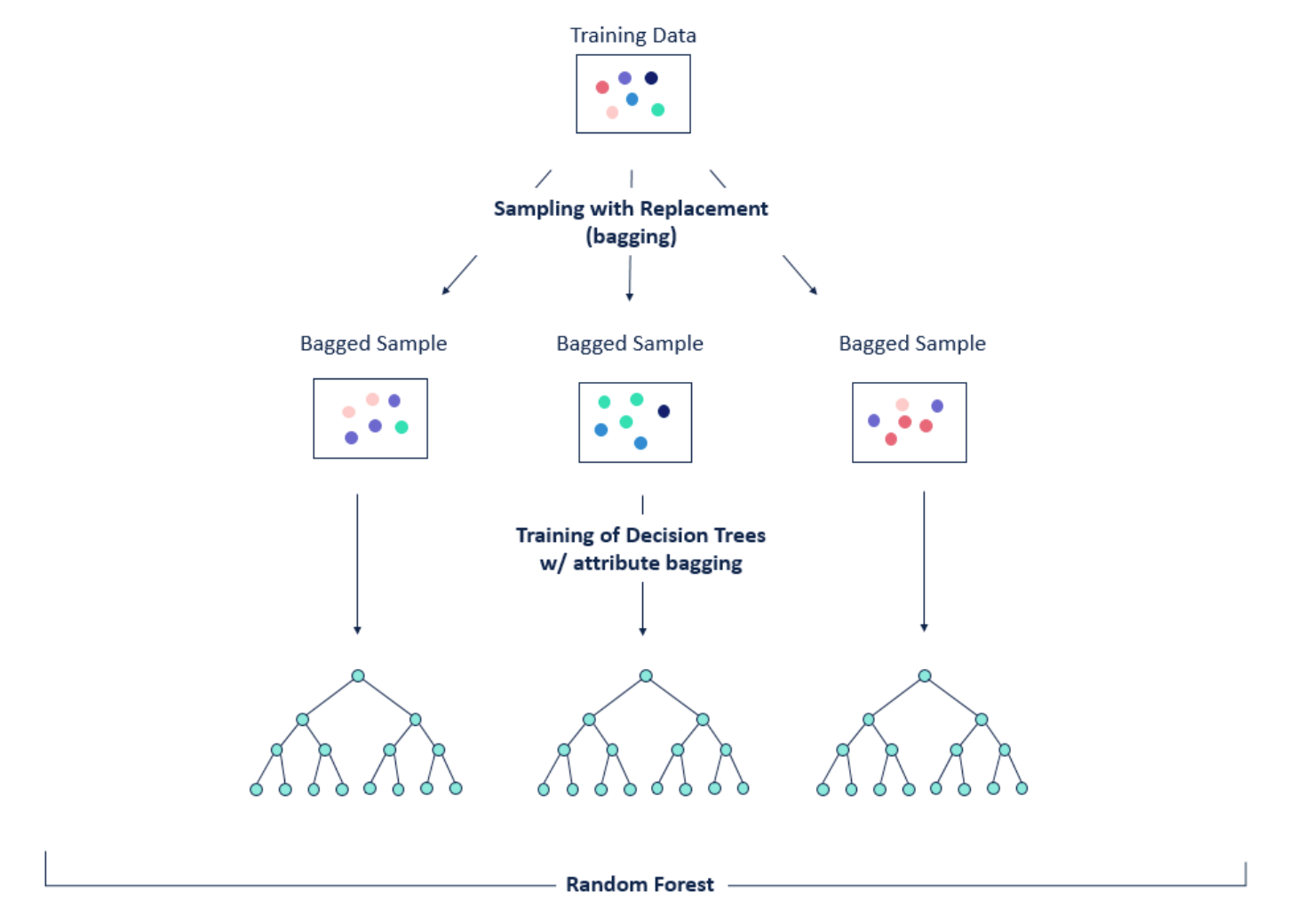
Original Data

Bootstrapping

Aggregating

Bagging

Random Forests



FINAL PREDICTION BY VOTING ACROSS ALL TREES,
THE RANDOM FOREST!

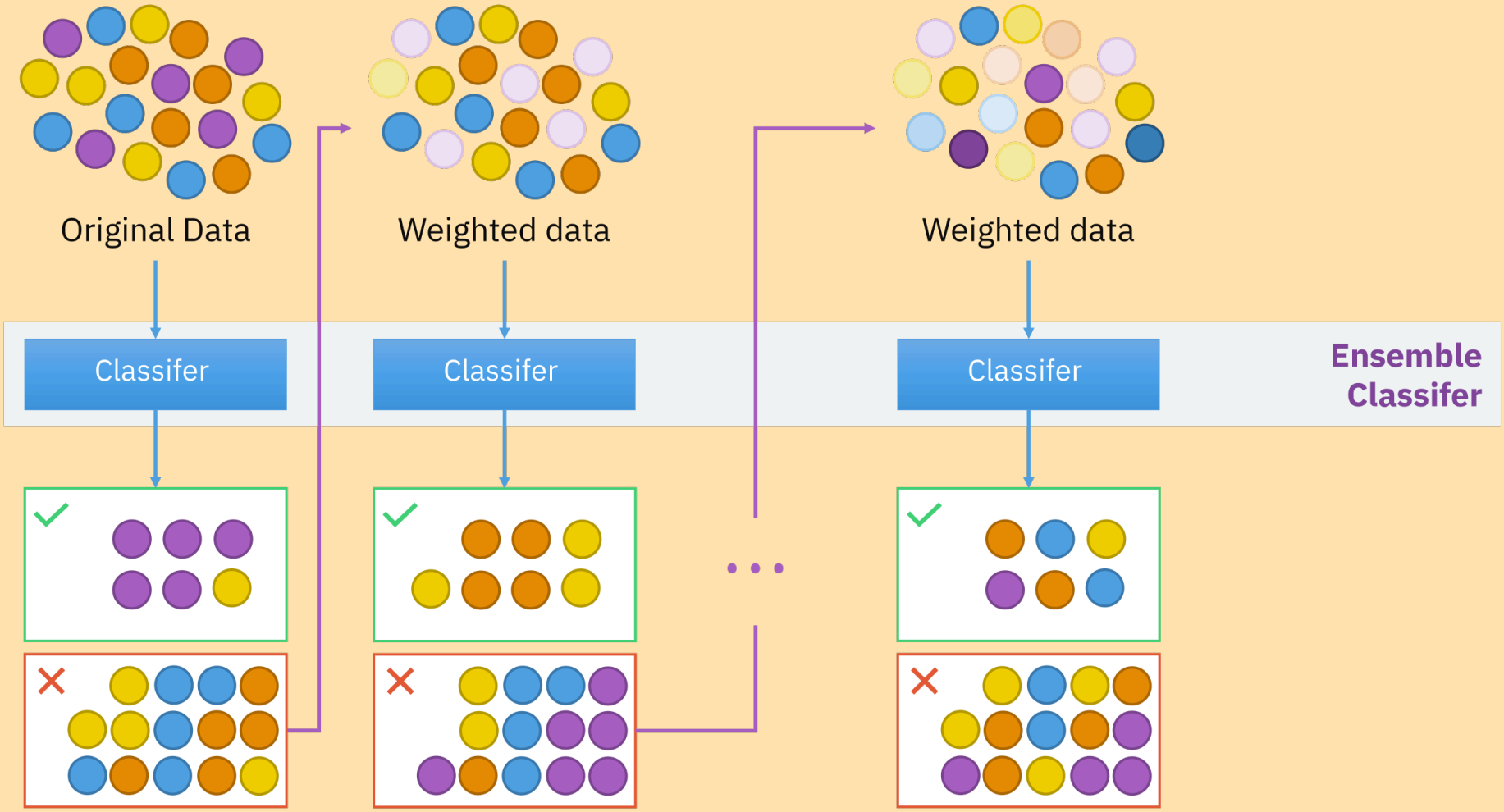
Boosting

- Use a sequence of “weak” learners to produce a strong one
- Uses model voting/averaging to aggregate predictions but models are trained on the same data and re-weighted based on their performance
- Justification: a sequence of weak learners that complement each other

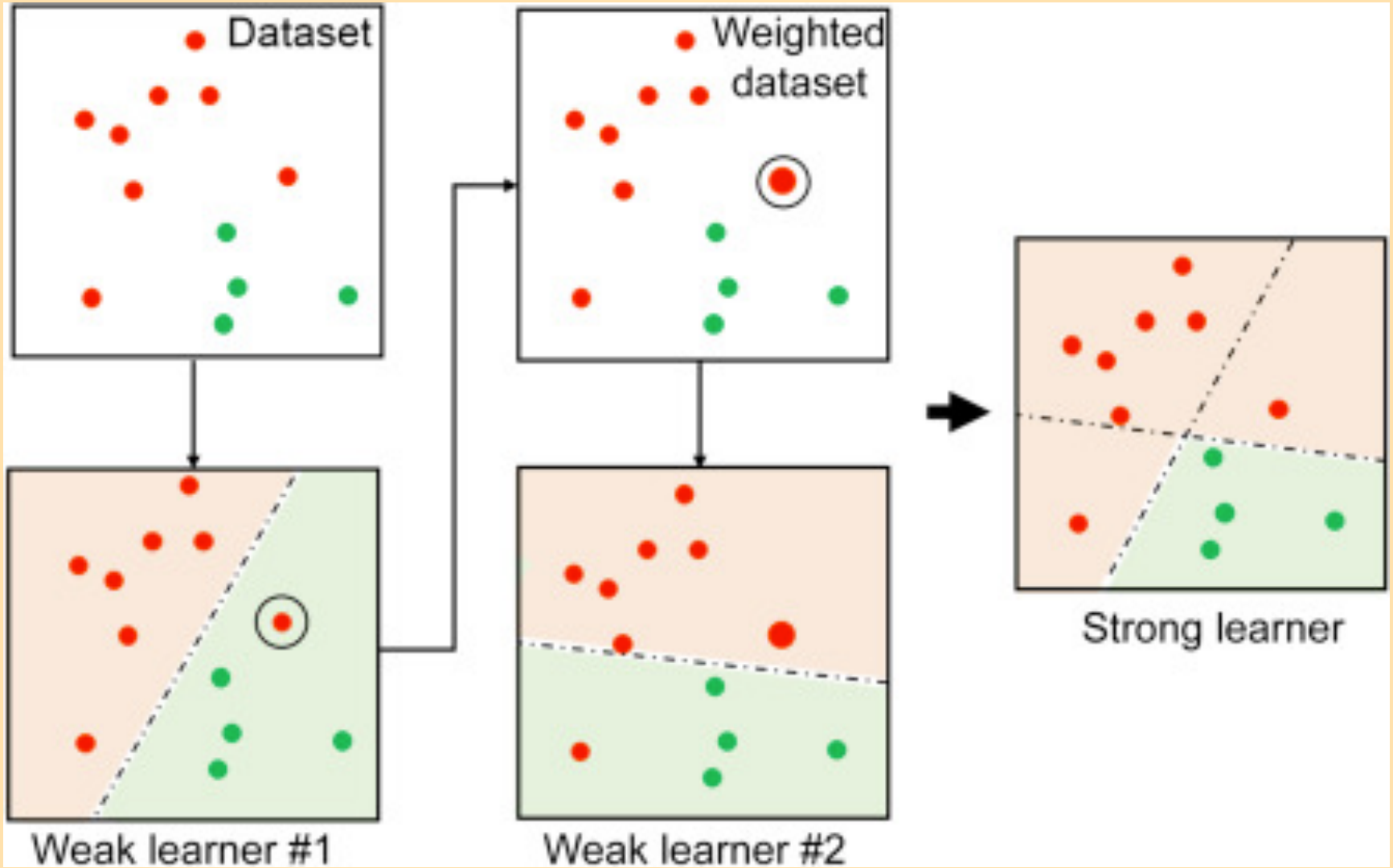
Boosting

- Basic idea of boosting: iterative procedure:
  1. Train a weak model on the data
  2. Determine which observations are incorrectly/correctly predicted
  3. Retrain the weak learner, this time encourage it to try and guess the previously misclassified data points
  4. Repeat process (say 100 times) then vote across all models
- With each iteration, models are encouraged to become expert for instances classified incorrectly by earlier models
- Subtlety comes from in how misclassified instances are reweighted

Boosting


Original Data
Weighted data
Weighted data
Classifer
Classifer
Classifer
Ensemble Classifer

Adaptive Boosting (Adaboost)



Boosting algorithms

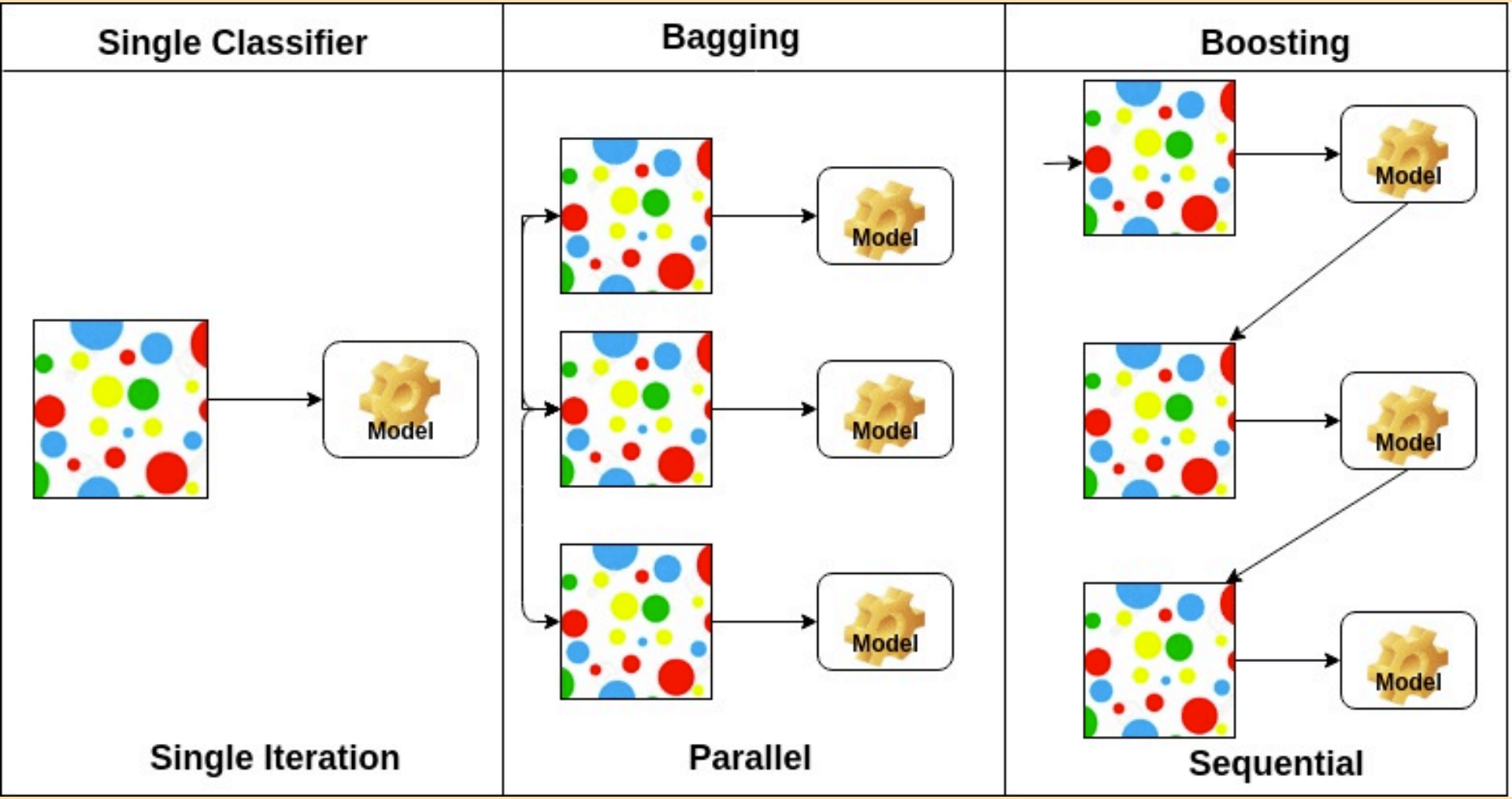
1997

Adaboost – Orange implemented

XGBoost – considered best predictive algorithm for tabular data

2016

Boosting vs Bagging



Stacking

- Uses a "*meta-learner*" instead of voting to combine predictions across models
- Individual initial models can be of many different types
- Complex methodology to build meta-learner by learning again from the initial set of models (machine learning on top of machine learning)
- Can help unlearn bias in each base model

Stacking

initial dataset

L weak learners
(that can be non-homogeneous)

meta-model
(**trained** to output predictions based on weak learners predictions)

Parameter Tuning

Parameter tuning, continued

- Some algorithms have one parameter to tune
  - KNN: k neighbors
  - Linear/Logistic regression: alpha regularization parameter
- Others have many
  - Decision trees: number of splits, number of leaves, minimum number of subject, etc…
  - Random forest: everything about the tree plus number of trees
- Parameter tuning often requires extensive training and evaluation over a grid of possible hyperparameters

Parameter tuning, continued

- Example: decision tree
  - Min. number of instances in leaves [5, 10, 15]
  - Do not split subsets smaller than [15, 20, 25]
  - Limit the maximal tree depth to [20, 60, 80]
  - Produces 3x3x3=27 possible configurations!
- Parameter tuning is often done in an **automated** way by defining the parameter grid and telling the computer to find the configuration that produces a single optimal evaluation metric

Feature Importance

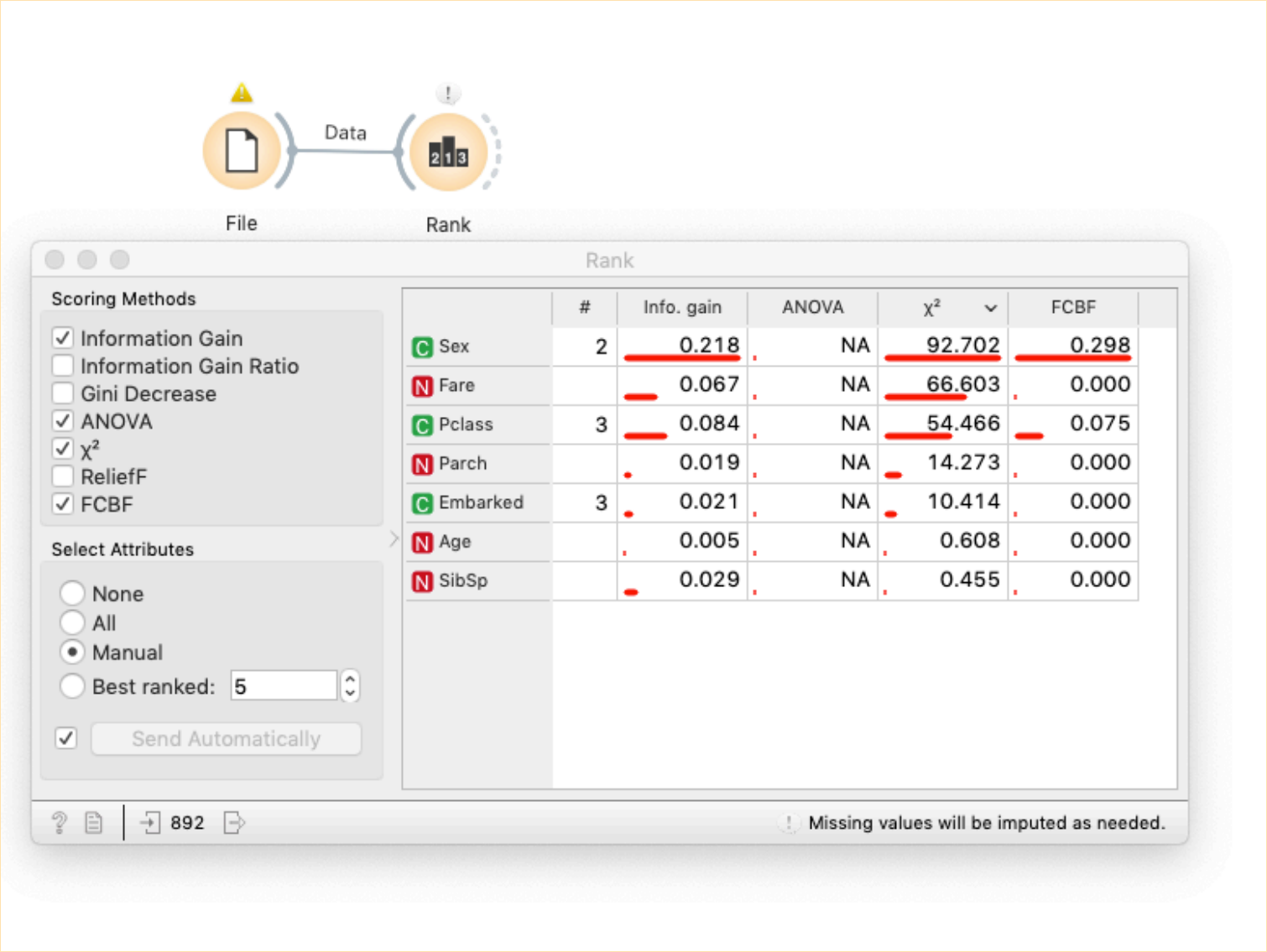
Feature importance

- Even though we fool ourselves into only being concerned with prediction, we often want to understand which features impact our prediction
- Given a target variable, which variables are important for prediction?
- The define this, you can develop measures of **feature importance**
- Feature importance can either be pair wise or model based
  - **Pairwise:** based directly on measures of association between target and individual features (*e.g. correlation*)
  - **Model based**: based on machine learning algorithms (*e.g. regression coefficients, classification tree splits*)

Feature importance – pairwise

- Looking at pairwise association between individual features and target can help identify variables that may be important for subsequent prediction
- Measures in Orange (target type dependent)
  - **Information gain**: expected information (reduction of entropy or chaos) in a feature with respect to a target
  - **ANOVA**: difference between average values of the feature in different classes
  - **Chi2**: dependence between features and outcome class
  - **FCBF:** information based measure, identifies redundancy due to correlation among features when estimating relevance to target

Titanic feature importance – pairwise



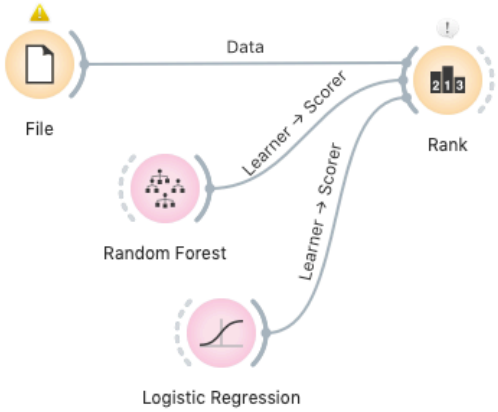
Feature importance – model based

- Model output (if available)
  - Decision trees, regression
  - Rule induction (lab)
- Permutation feature importance – model agnostic
  1. Fit model with data and evaluate predictive performance
  2. Return to data, scramble one feature randomly among observations
  3. Refit model and compare model performance to original data.
  4. Repeat for every variable. Most important variables will produce biggest drops in predictive performance
  - You can do this with any model that has tabular data!

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks
182	155	...	
175	147	...	
...	...	...	
156	142	...	
153	130	...	

Feature importance – permutation

Titanic feature importance – model



Rank

Scoring Methods

- [x] Information Gain
- [ ] Information Gain Ratio
- [ ] Gini Decrease
- [x] ANOVA
- [x] χ^2
- [ ] ReliefF
- [x] FCBF

Select Attributes

- ( ) None
- ( ) All
- (•) Manual
- ( ) Best ranked: 5

[x] Send Automatically

	#	Info. gain	ANOVA	χ^2	FCBF	Rand...rest	Logi...sion
C Sex	2	0.218	NA	92.702	0.298	0.185	1.796
C Pclass	3	0.084	NA	54.466	0.075	0.071	1.183
N Fare		0.067	NA	66.603	0.000	0.240	0.119
N SibSp		0.029	NA	0.455	0.000	0.048	0.335
C Embarked	3	0.021	NA	10.414	0.000	0.013	0.398
N Parch		0.019	NA	14.273	0.000	0.032	0.066
N Age		0.005	NA	0.608	0.000	0.258	0.536

892

Missing values will be imputed as needed.

Review of this lecture

- Cross-validation
- Ensemble methods
  - Bagging: *Random Rorests*
  - Boosting: *Adaboost*
  - Stacking
- Hyperparameter tuning continued
- Feature importance

Next time

- Clustering
  - k-means clustering
  - TwoStep clustering
  - Kohonen self-organizing map (SOM)
- Data reduction
  - Principal components analysis (PCA)
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Guidance for choosing machine learning algorithms