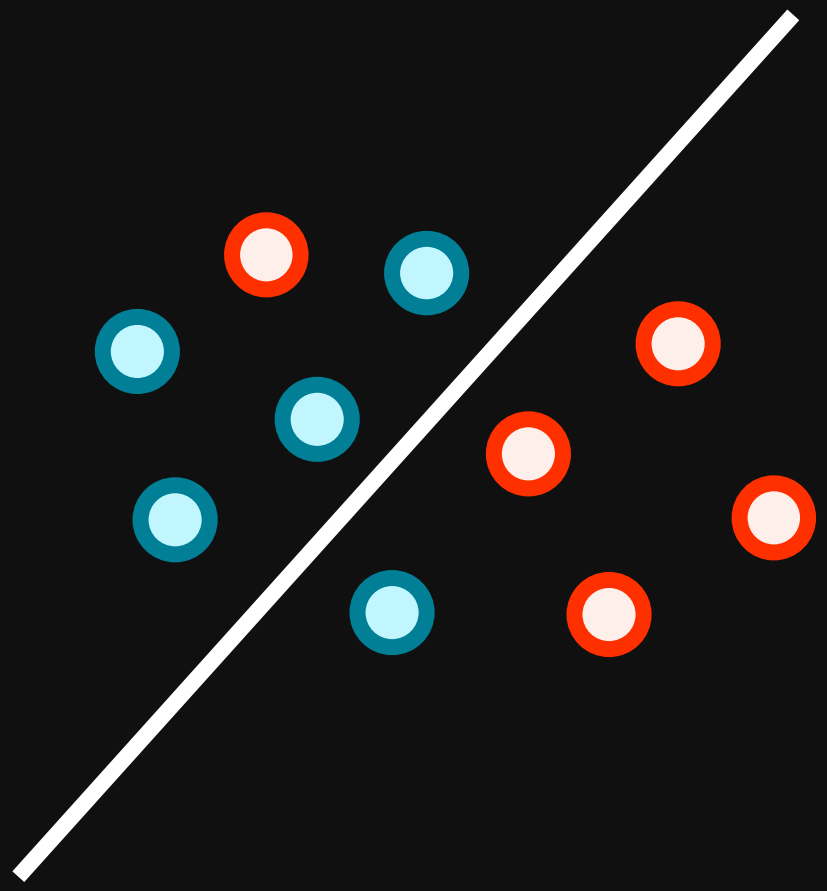
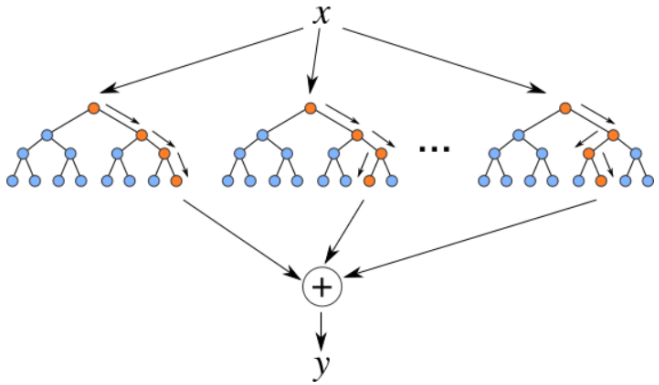
Basics of Machine Learning

Dmitry Ryabokon, github.com/dryabokon



Lesson 15
Ensemble Learning

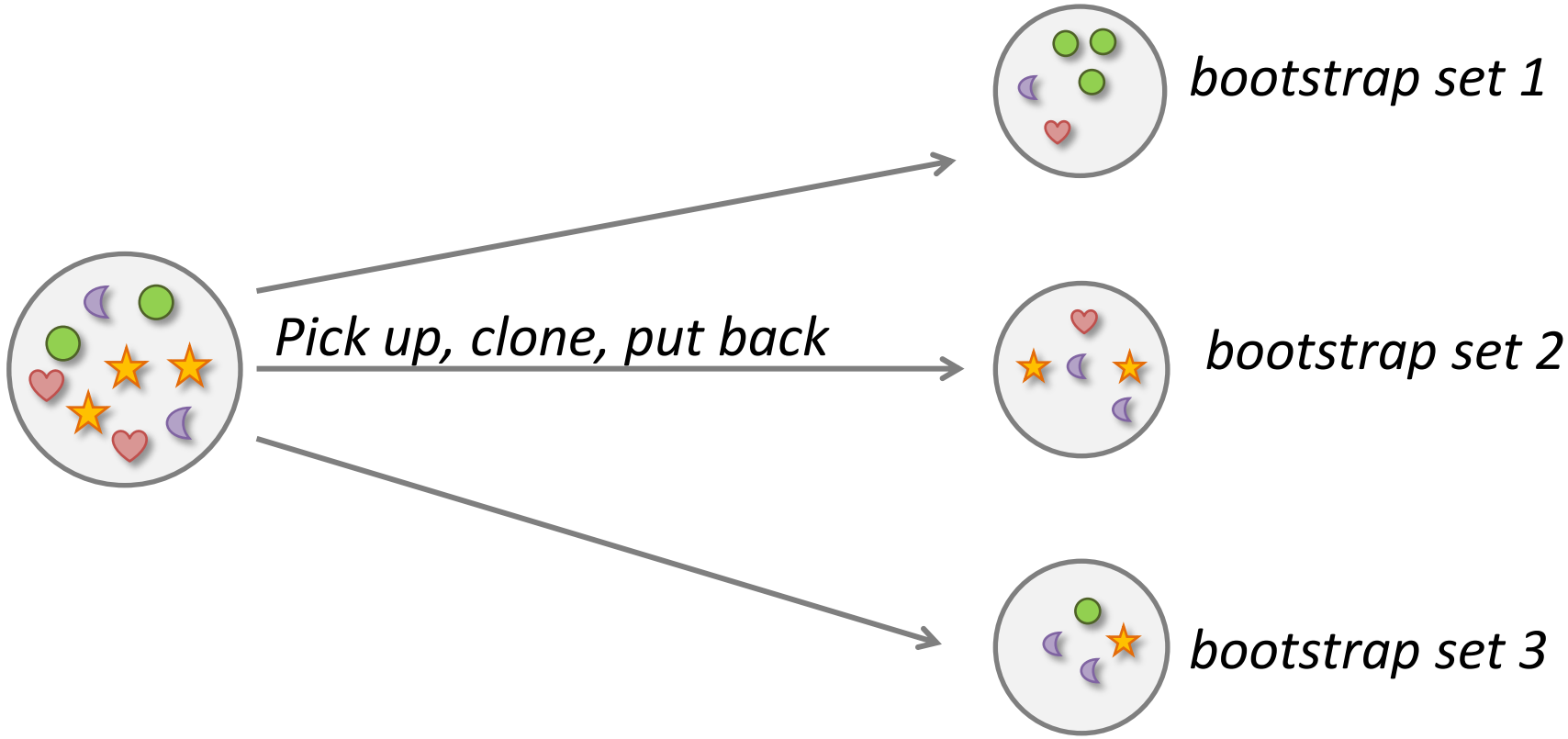


Summary

- Bootstrapping
- Bagging
- Boosting
- Random Forest

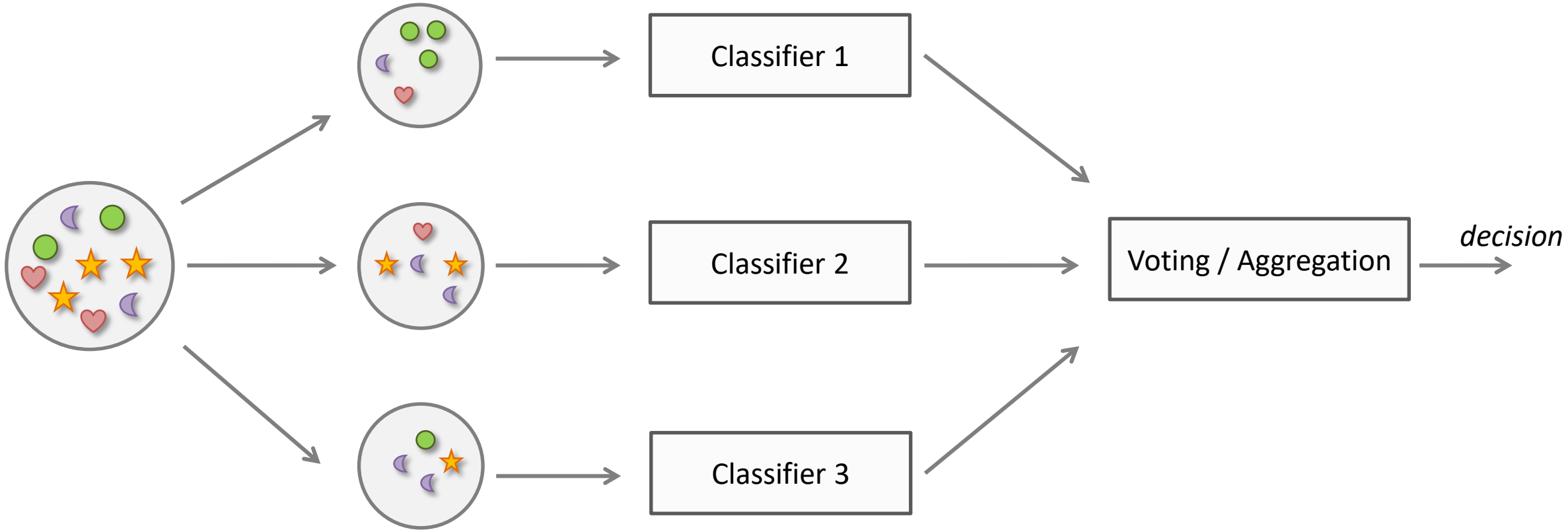
Ensemble Learning

Bootstrapping: random sampling with replacement



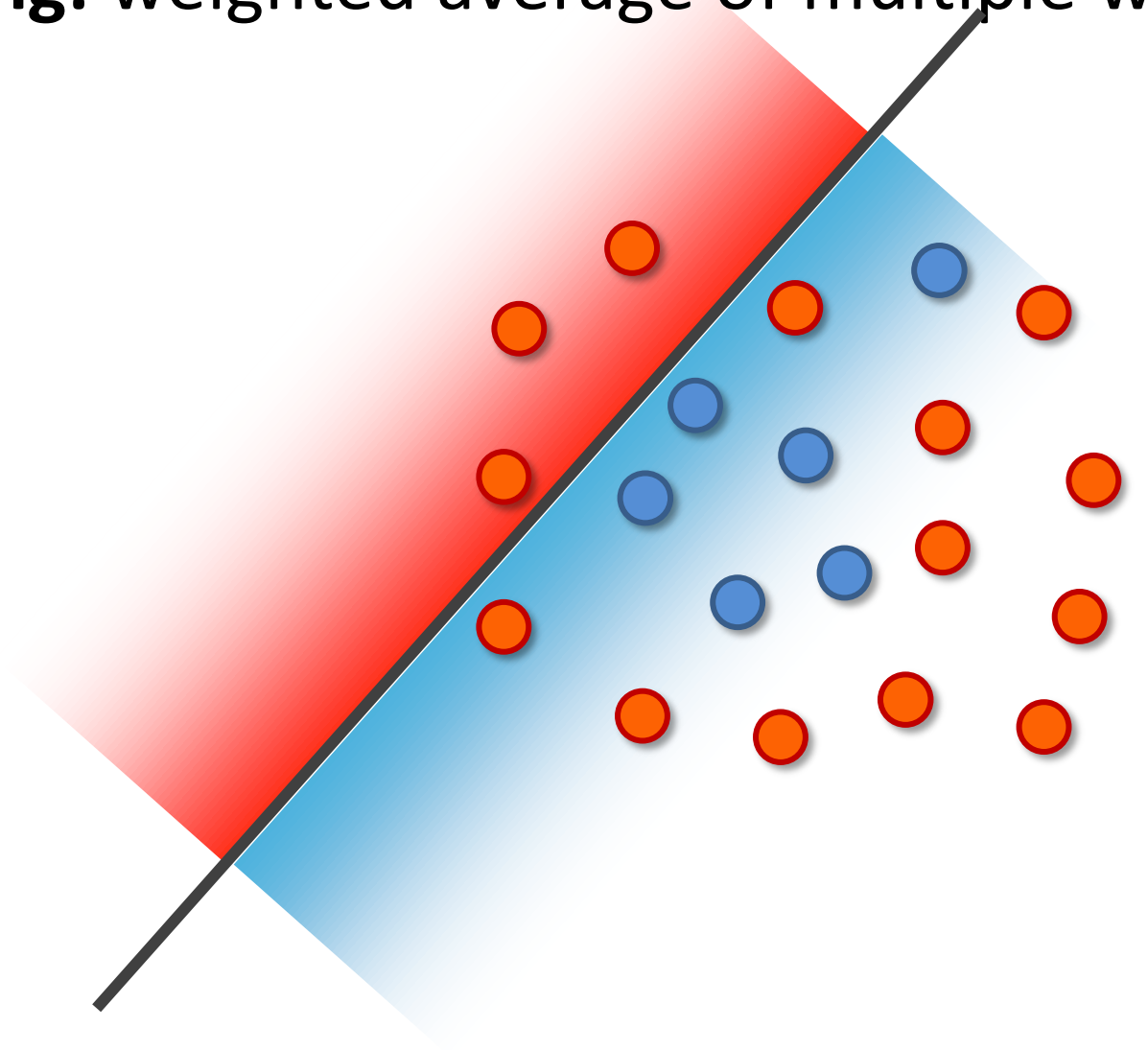
Ensemble Learning

Bagging: Bootstrap **Agg**regation



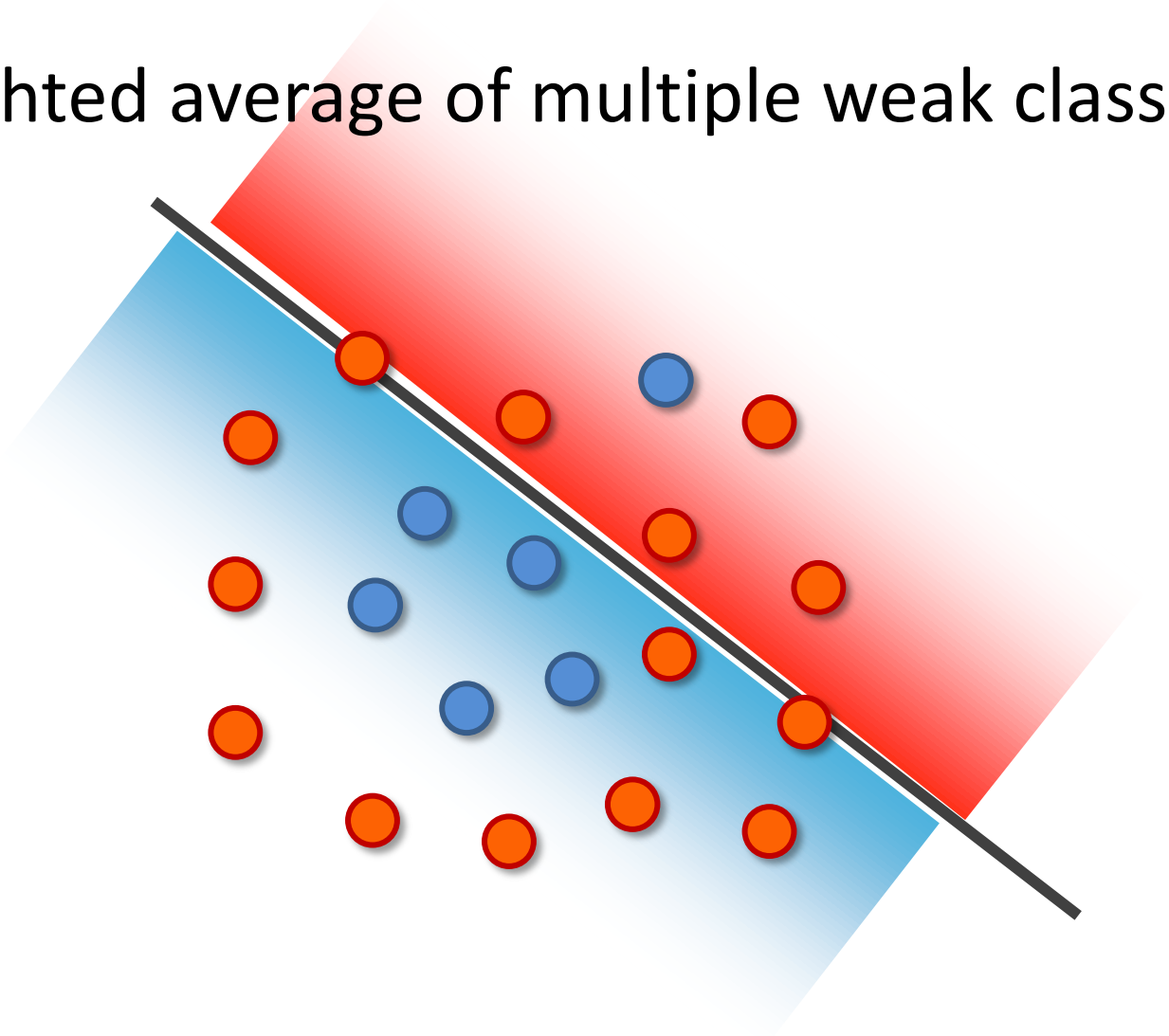
Ensemble Learning

Boosting: weighted average of multiple weak classifiers



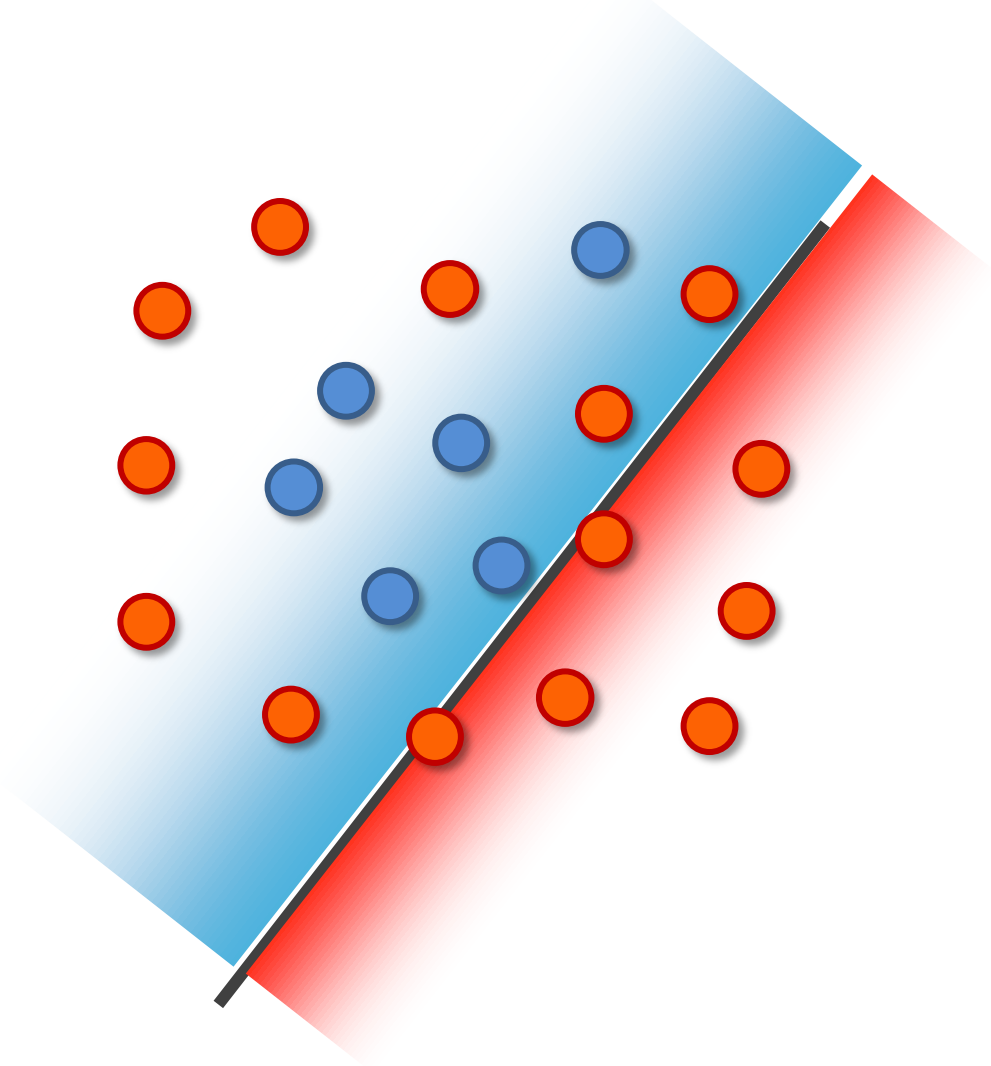
Ensemble Learning

Boosting: weighted average of multiple weak classifiers



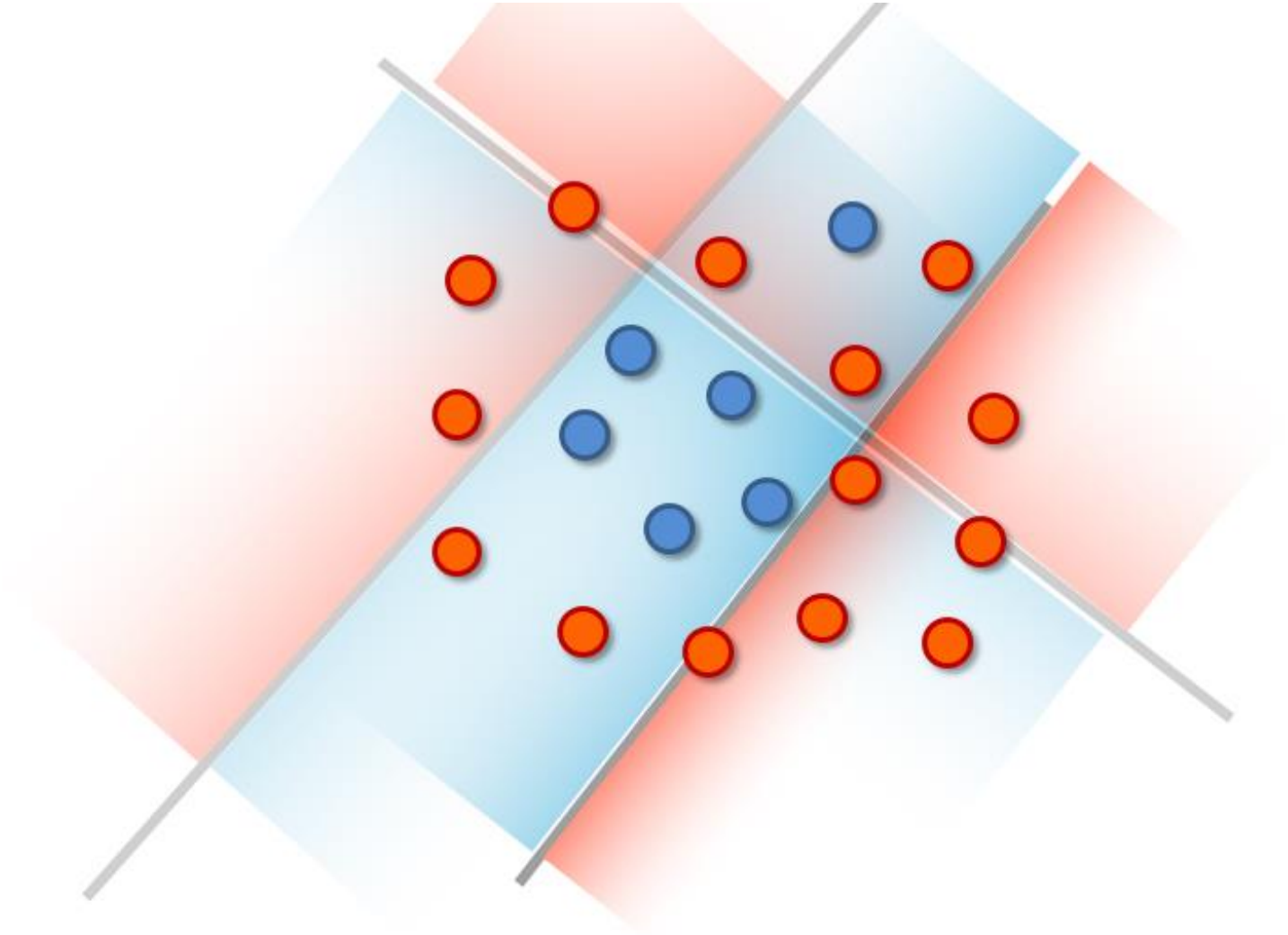
Ensemble Learning

Boosting: weighted average of multiple weak classifiers



Ensemble Learning

Boosting: weighted average of multiple weak classifiers



Ensemble Learning

Bagging

- Aims to decrease variance
- Aims to solve over-fitting problem

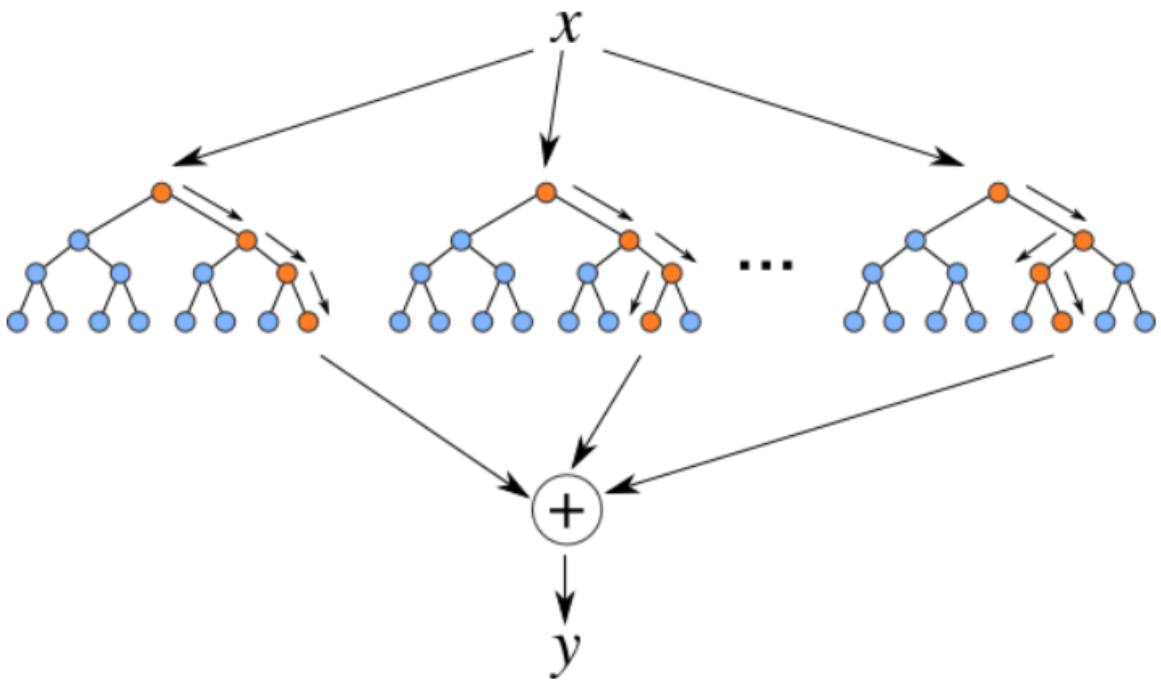
In bagging technique, a data set is divided into n samples using randomized sampling. Then, using a single learning algorithm a model is build on all samples. Later, the resultant predictions are combined using voting or averaging.
Bagging is done in parallel.

Boosting

- Aims to decrease bias

In boosting, after the first round of predictions, the algorithm weighs misclassified predictions higher, such that they can be corrected in the succeeding round. This sequential process of giving higher weights to misclassified predictions continue until a stopping criterion is reached.

Random Forest

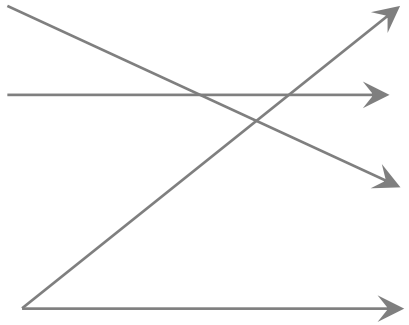


https://www.youtube.com/watch?v=J4Wdy0Wc_xQ

Random forest

Original dataset

F1	F2	F3	Target
A	A	A	0
A	A	B	0
B	B	A	1
A	B	B	1



Bootstrapped dataset

F1	F2	F3	Target
A	B	B	1
A	A	B	0
A	A	A	0
A	B	B	1

Random forest

Bootstrapped dataset

F1	F2	F3	Target
A	B	B	1
A	A	B	0
A	A	A	0
A	B	B	1

candidate candidate

Determine the best root node out of randomly selected sub-set of candidates

Random forest

Bootstrapped dataset

F1	F2	F3	Target
A	B	B	1
A	A	B	0
A	A	A	0
A	B	B	1

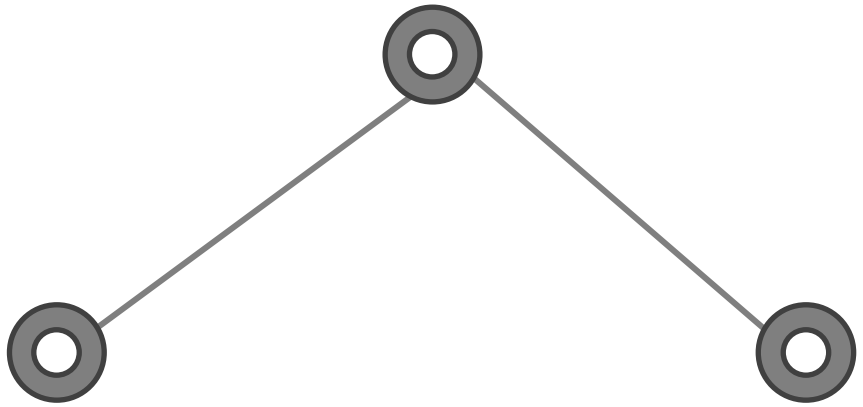
root

Random forest

Bootstrapped dataset

F1	F2	F3	Target
A	B	B	1
A	A	B	0
A	A	A	0
A	B	B	1

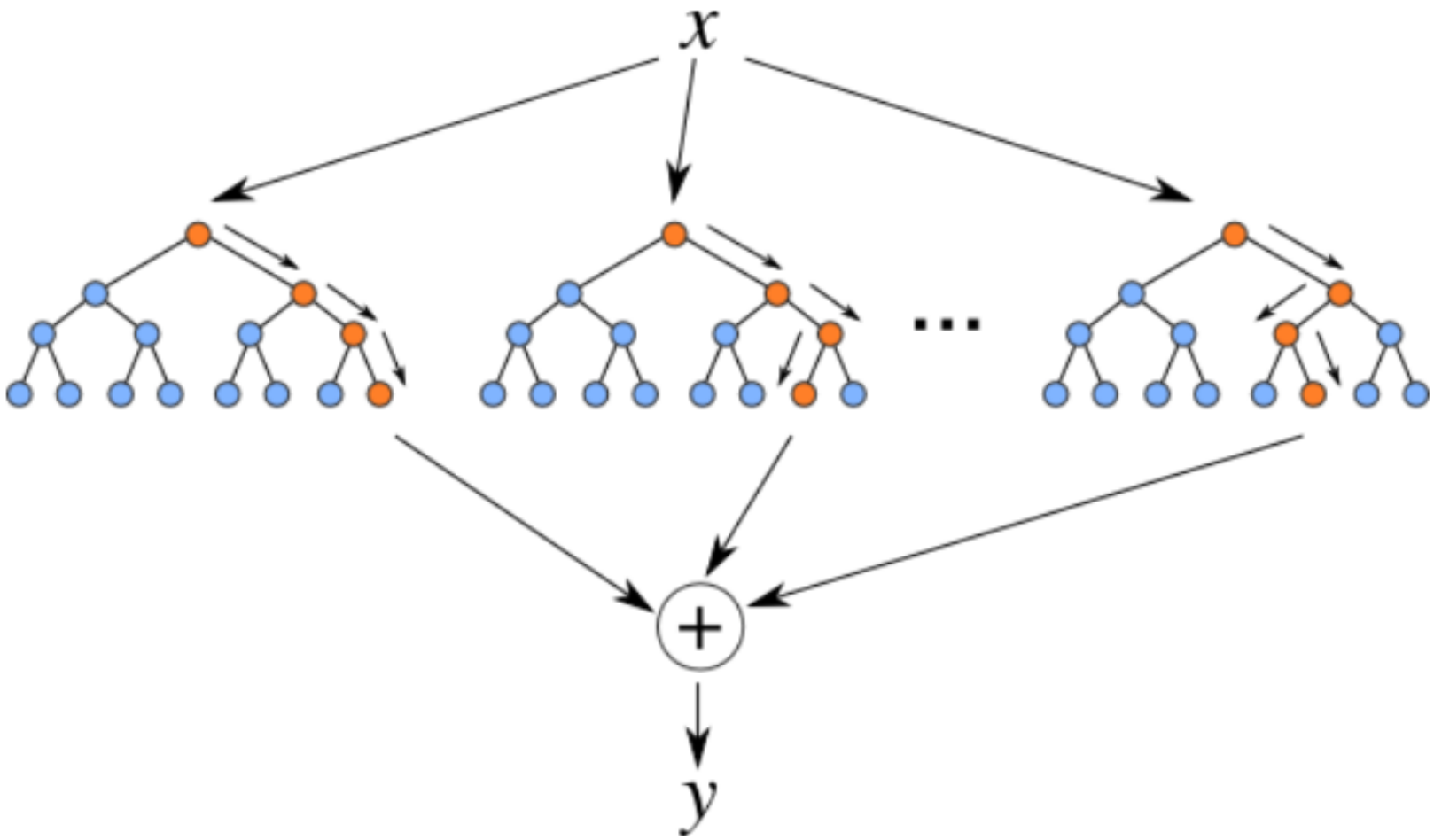
candidate candidate



Build tree as usual, but considering a random subset of features at each step

Random forest

Bagging: do aggregate decisions against the bootstrapped data



Random forest

Validation: run aggregated decisions against out-of-bag dataset

Original dataset

F1	F2	F3	Target
A	A	A	0
A	A	B	0
B	B	A	1
A	B	B	1

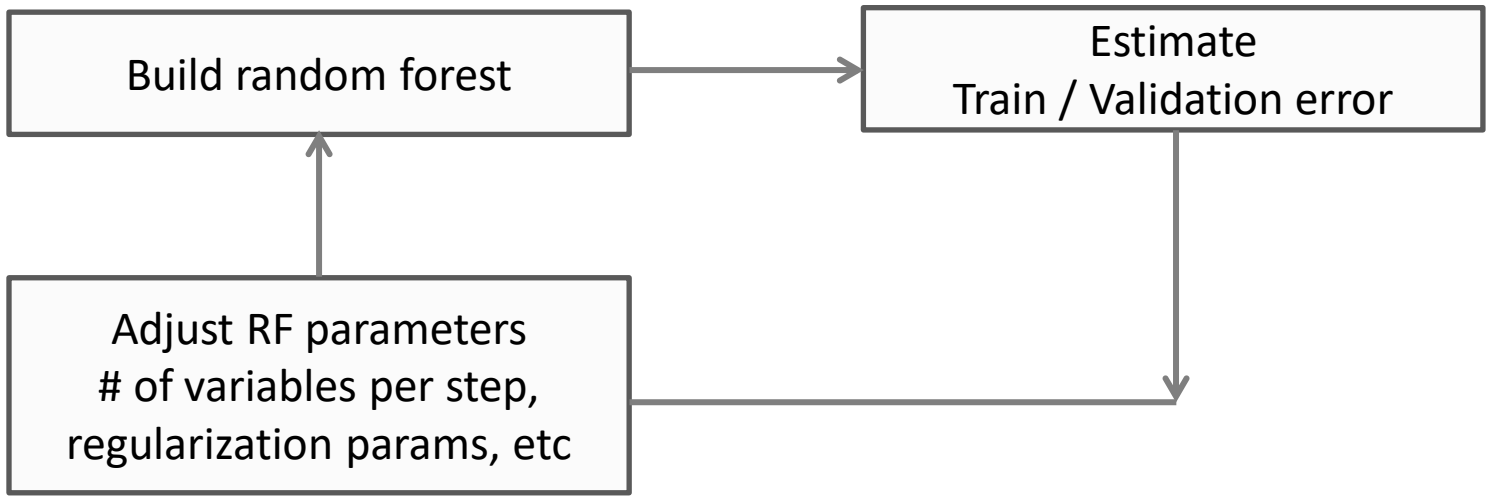
out-of-bag

Bootstrapped dataset

F1	F2	F3	Target
A	B	B	1
A	A	B	0
A	A	A	0
A	B	B	1

Random forest

The process



RF vs GBM

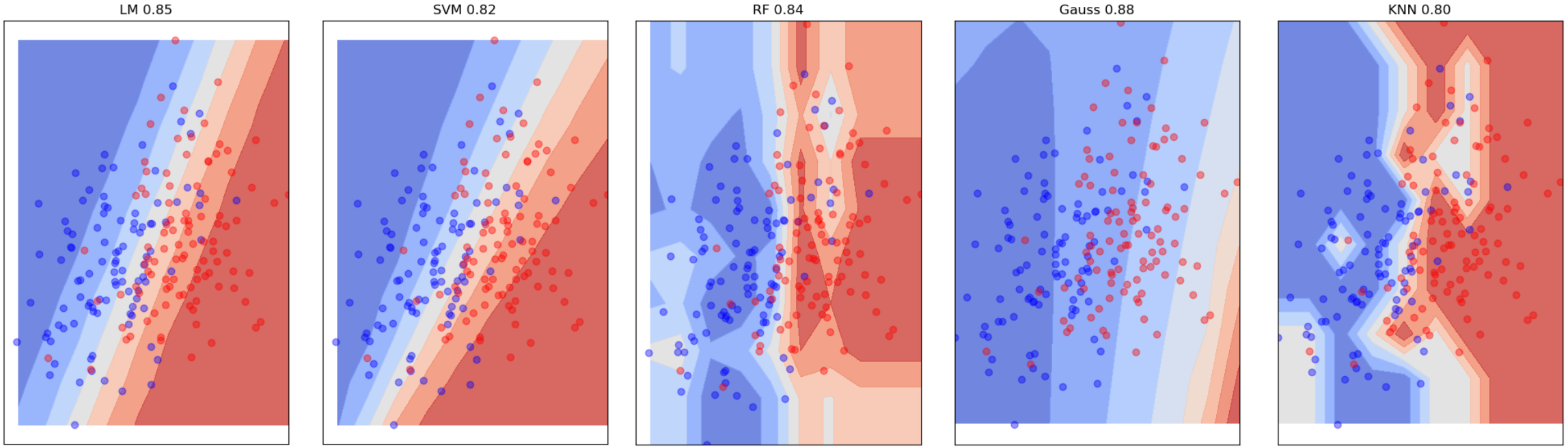
RF uses bagging technique to make predictions.
GBM uses boosting techniques to make predictions.

Random forest improves model accuracy by reducing variance (mainly). The trees grown are uncorrelated to maximize the decrease in variance.

On the other hand, GBM improves accuracy by reducing both bias and variance in a model.

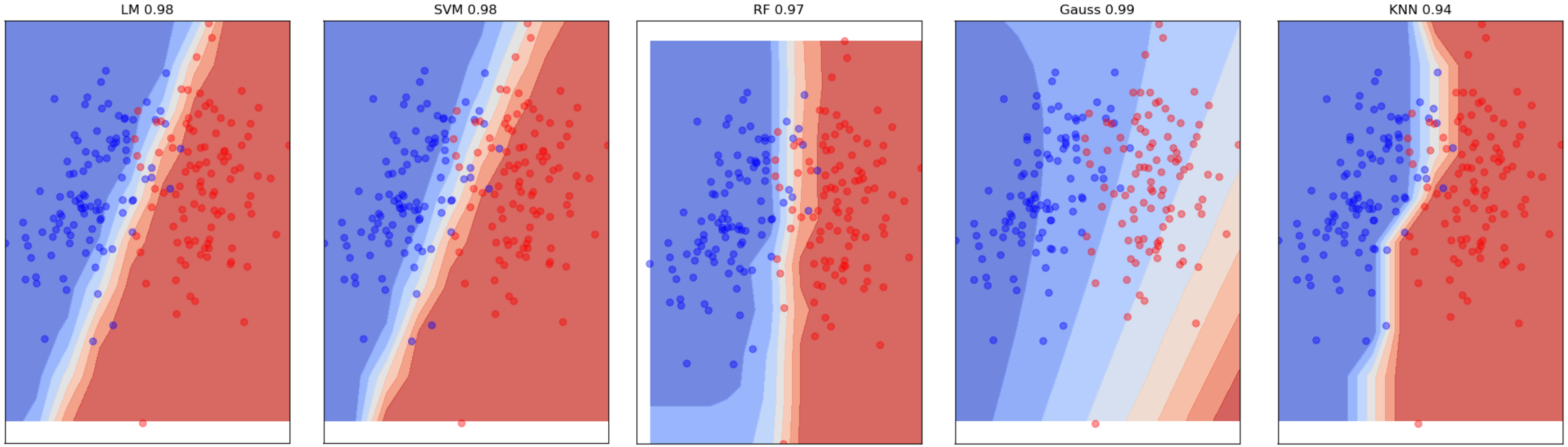
Random forests are a significant number of decision trees pooled using averages or majority rules at the end. Gradient boosting machines also combine decision trees but at the beginning of the process unlike Random forests. Random forest creates each tree independent of the others while gradient boosting develops one tree at a time. Gradient boosting yields better outcomes than random forests if parameters are carefully tuned but it's not a good option if the data set contains a lot of outliers/anomalies/noise as it can result in overfitting of the model.Random forests perform well for multiclass object detection. Gradient Boosting performs well when there is data which is not balanced such as in real time risk assessment.

Classification examples: 2D data, 2 classes



https://github.com/dryabokon/ML/blob/master/ex_02_01_train_test_2_classes.py

Classification examples: 2D data, 2 classes



https://github.com/dryabokon/ML/blob/master/ex_02_01_train_test_2_classes.py