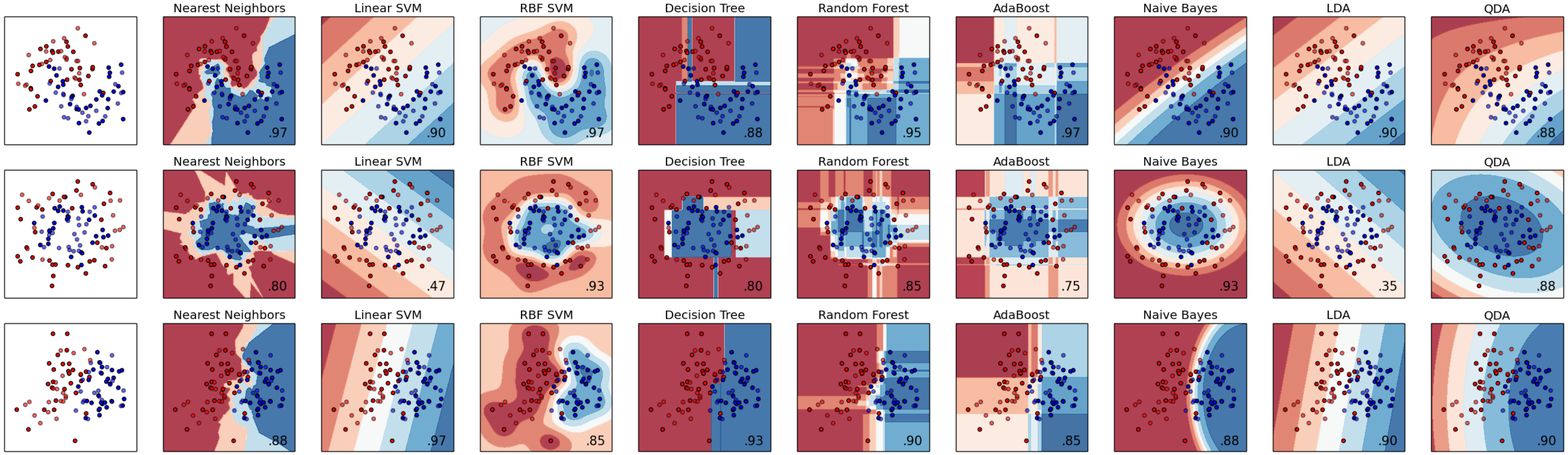


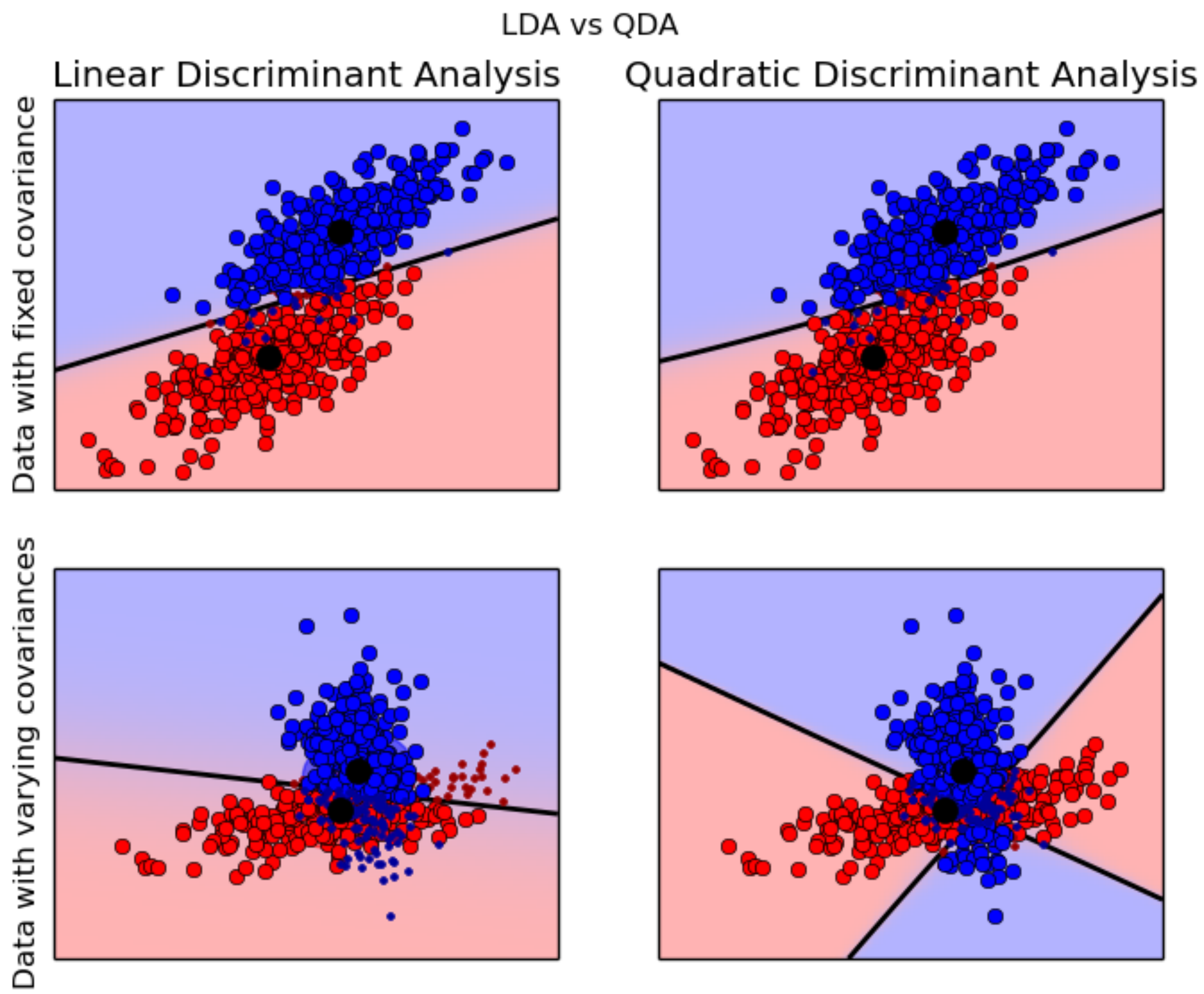
Algo-Palooza 2k15

(BIG) CAVEAT:

Often times choosing/creating good features or gathering more data will help more than changing algorithms…

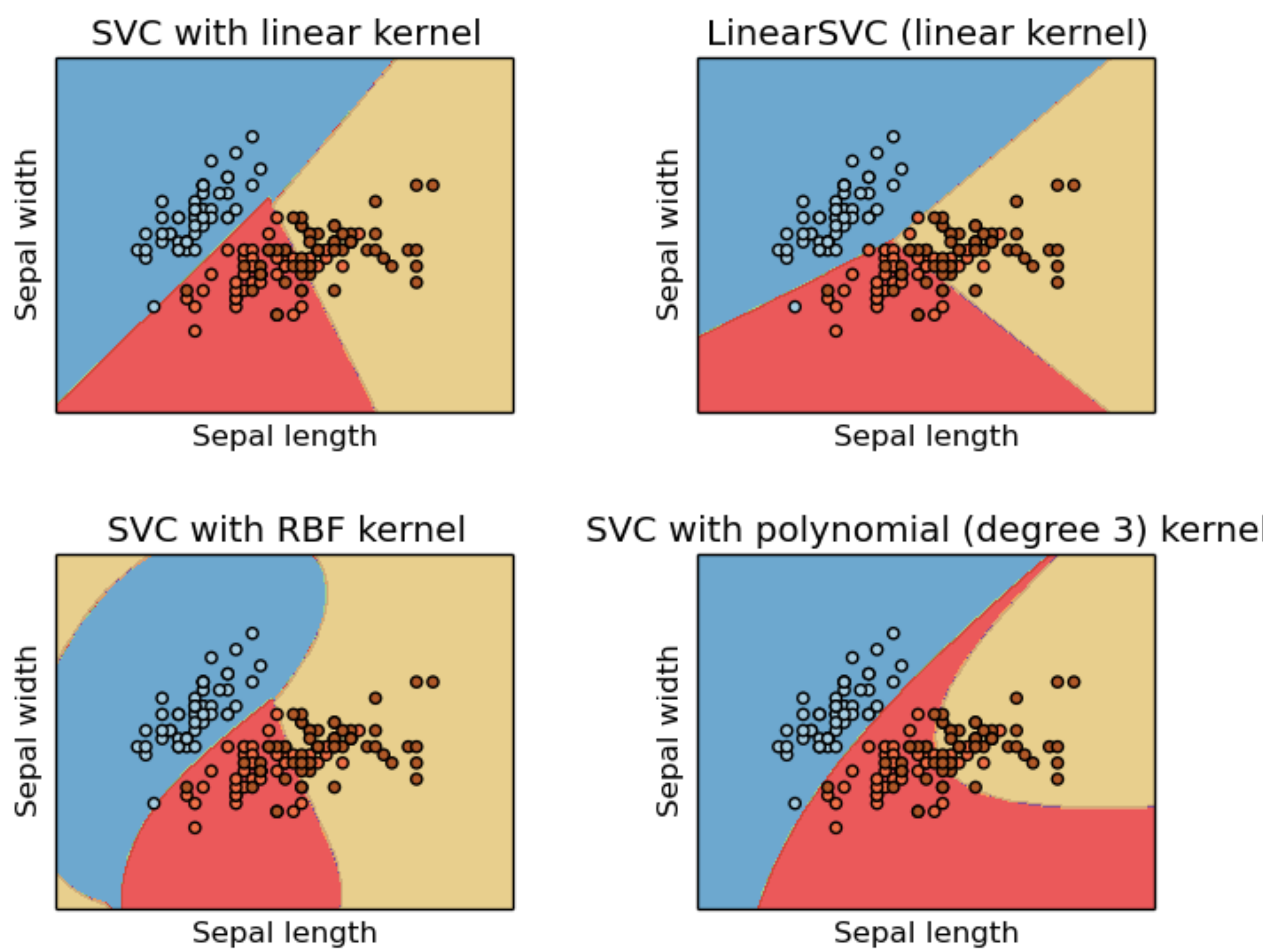
Linear Discriminants

"draw a line through it"



SVMs (Support Vector Machines)

“Advanced draw-a-line-through-it”



Pros:

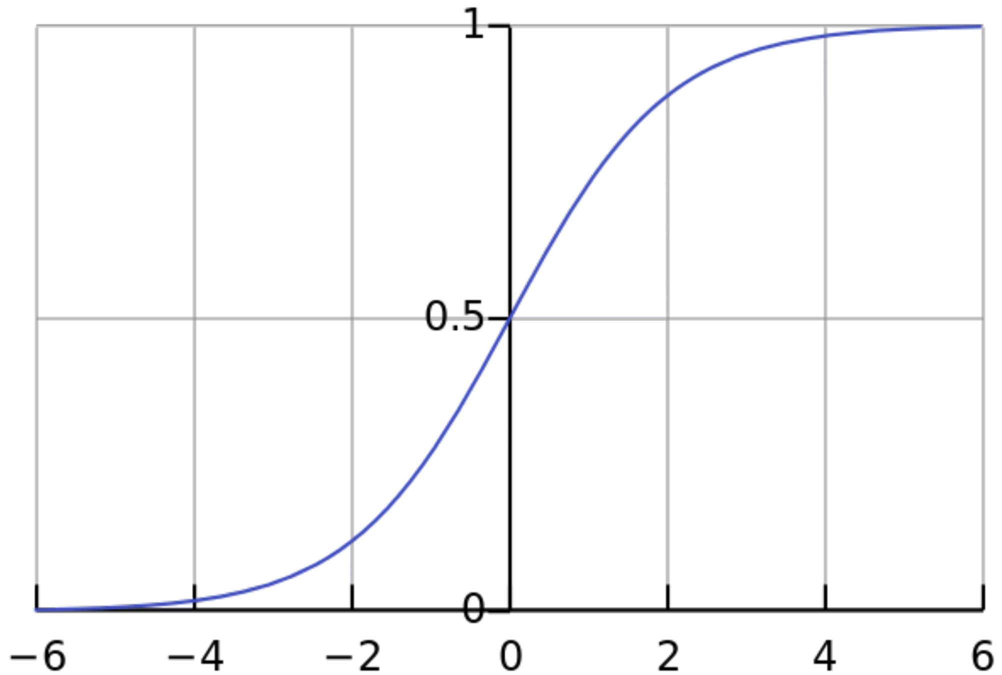
- can work well on many types of data (low or high-dimensional, linearly divisible or not) data thanks to the “kernel trick“

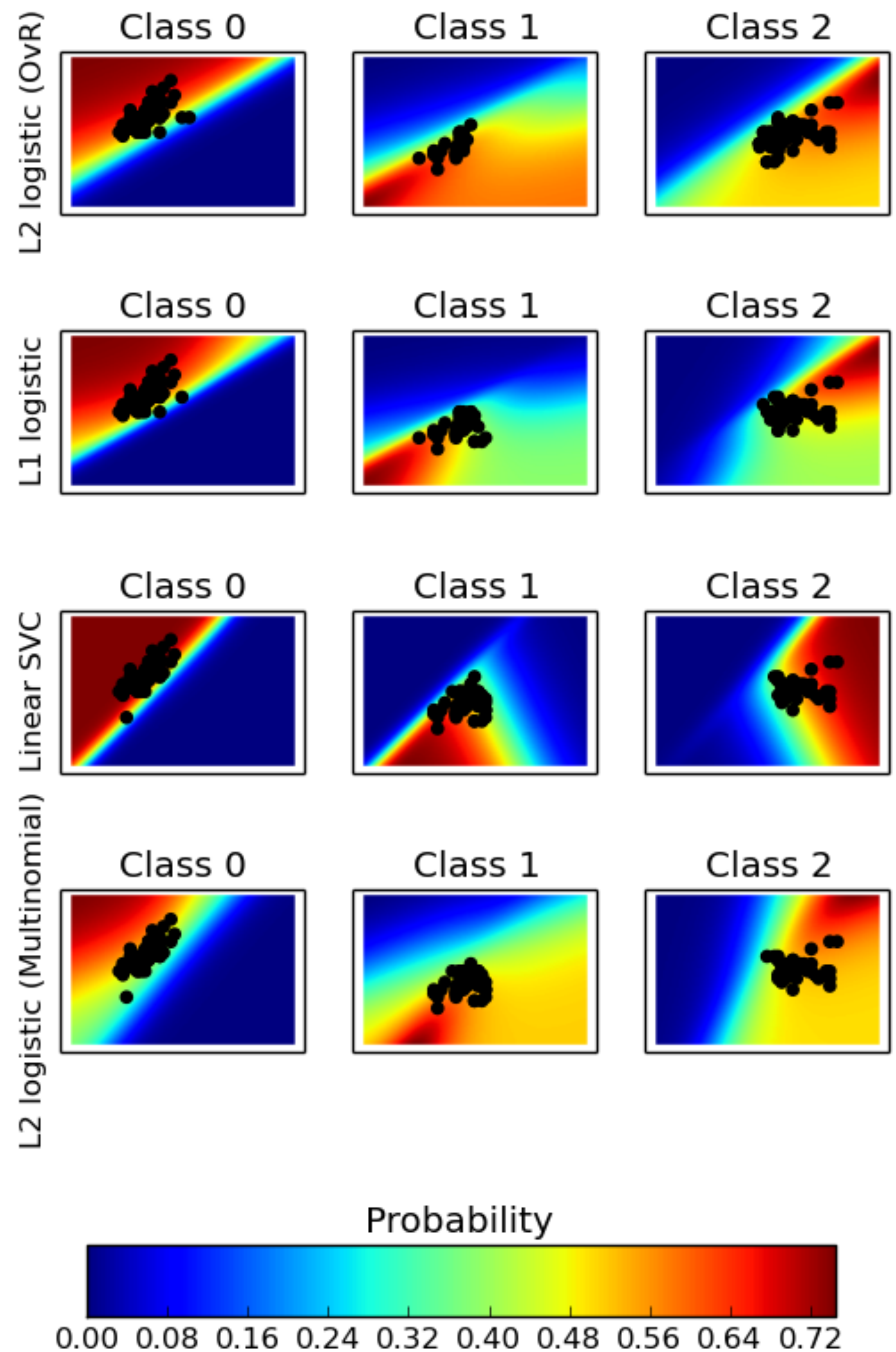
Cons:

- Not particularly easy to explain or tweak
- Can only *kinda* provide probability estimates– computationally intensive

Logistic Regression

"divide it with a log function"





Pros:

- Can provide probabilities (has “fuzzy edges“)
- Can update with new data

Cons:

- Limited to linear decision boundaries

Naïve Bayes

“calculate a probability of it”

- Uses the training data to build conditional probabilities for each feature in X
- When classifying, use the probabilities calculated in training + Bayes Rule to calculate chance of each possible result given the data

Pros:

- Fast + simple
- Can provide estimates
- Works well with small amounts of data
- Can add new data without retraining

Cons:

- Assumes independence between features (though it can do a good job even if this is a 💩-y assumption)
- Can’t learn interactions between features

Decision Tree

“make a flow chart to describe it”

Pros:

- Easy to interpret results, especially at low dimensions/simplistic models
- Very fast
- Can adapt to many shapes of data

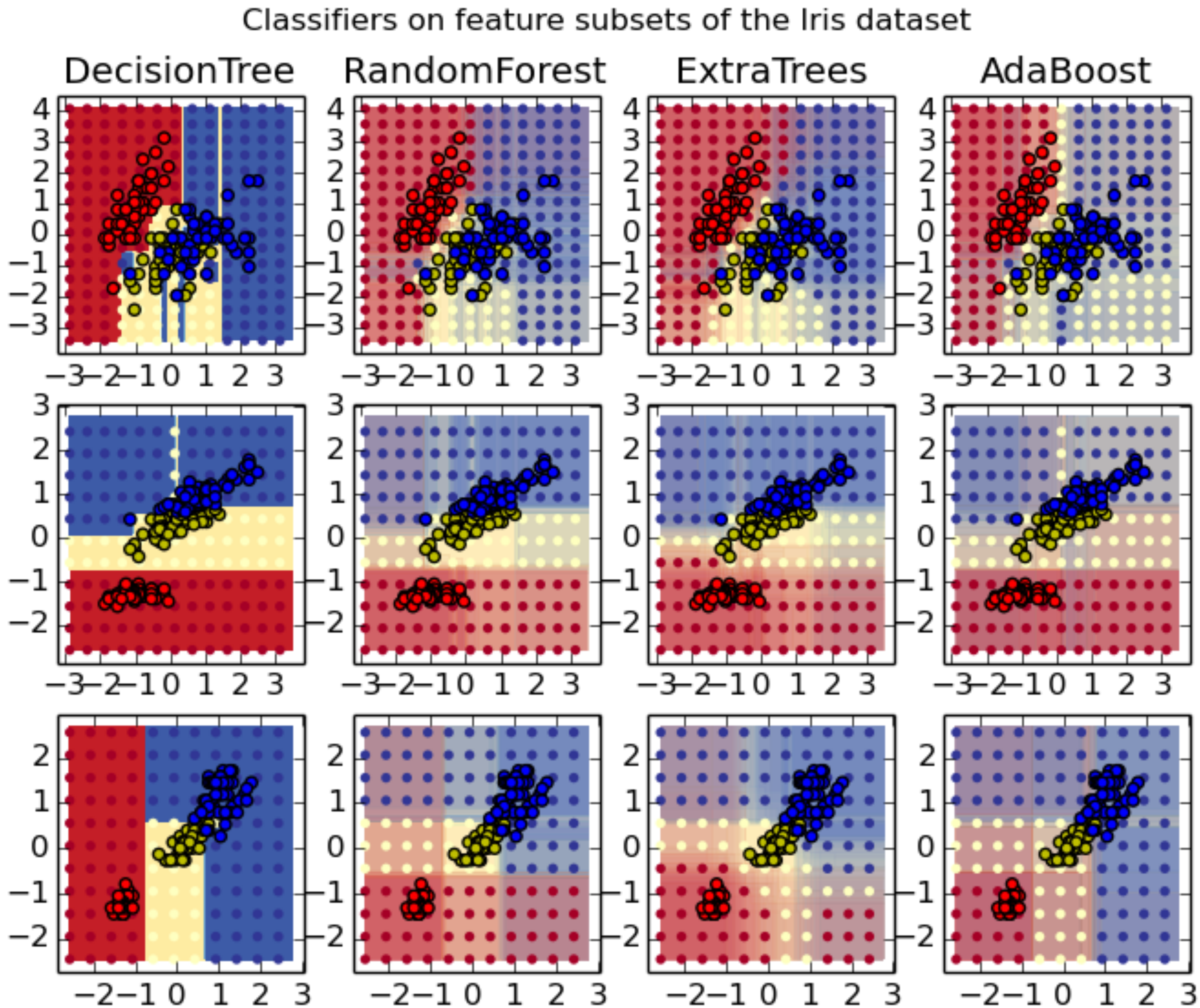
Cons:

- Very easy to overfit with these if you aren’t careful.
- Have to rebuild any time you get new data.
- No probability estimates

Ensemble Methods

"combine results from a bunch of models"

- Random Forest creates a bunch of decision trees, then makes a decision based on the mode of the results. Fixes the overfitting problem of a single decision tree.
- AdaBoost trains a bunch of simplistic models, focusing more over time on the cases failed in the early ones


Classifiers on feature subsets of the Iris dataset
DecisionTree
RandomForest
ExtraTrees
AdaBoost

Pros:

- Optimized to solve the problems present in their component parts
- Generally don’t require much parameter tweaking
- If data doesn’t change very often, you can make them semi-online by just adding new trees to the ensemble
- Can provide probabilities

Cons

- Slower than their component parts (though if those are fast, it doesn’t matter)