Data Science

Deriving Knowledge from Data at Scale

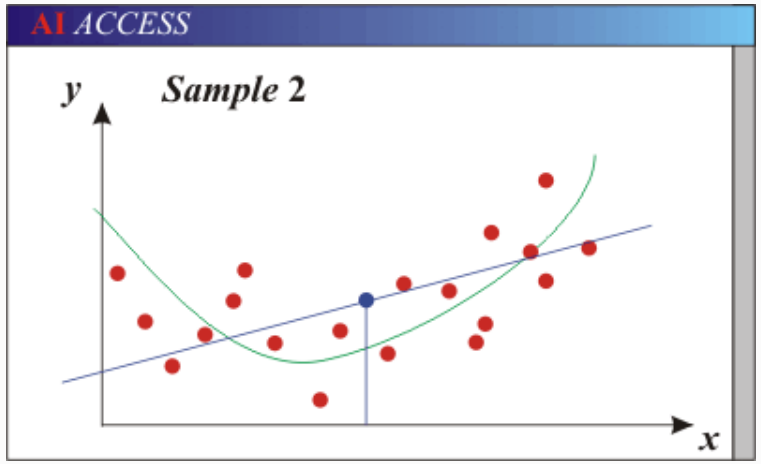
Wee Hyong Tok

July 7, 2016

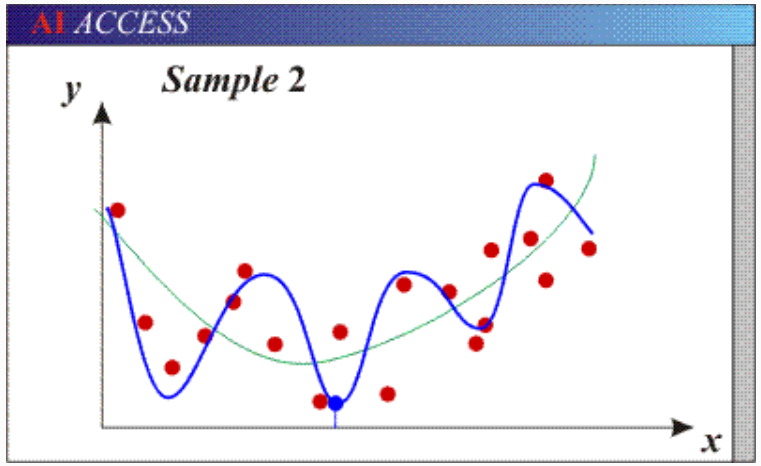
Continue from last lecture

Bias Variance Tradeoff

Bias-Variance Trade-off



- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).



- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).

Bias-Variance Trade-off

$$E(MSE) = noise^2 + bias^2 + variance$$

Unavoidable error

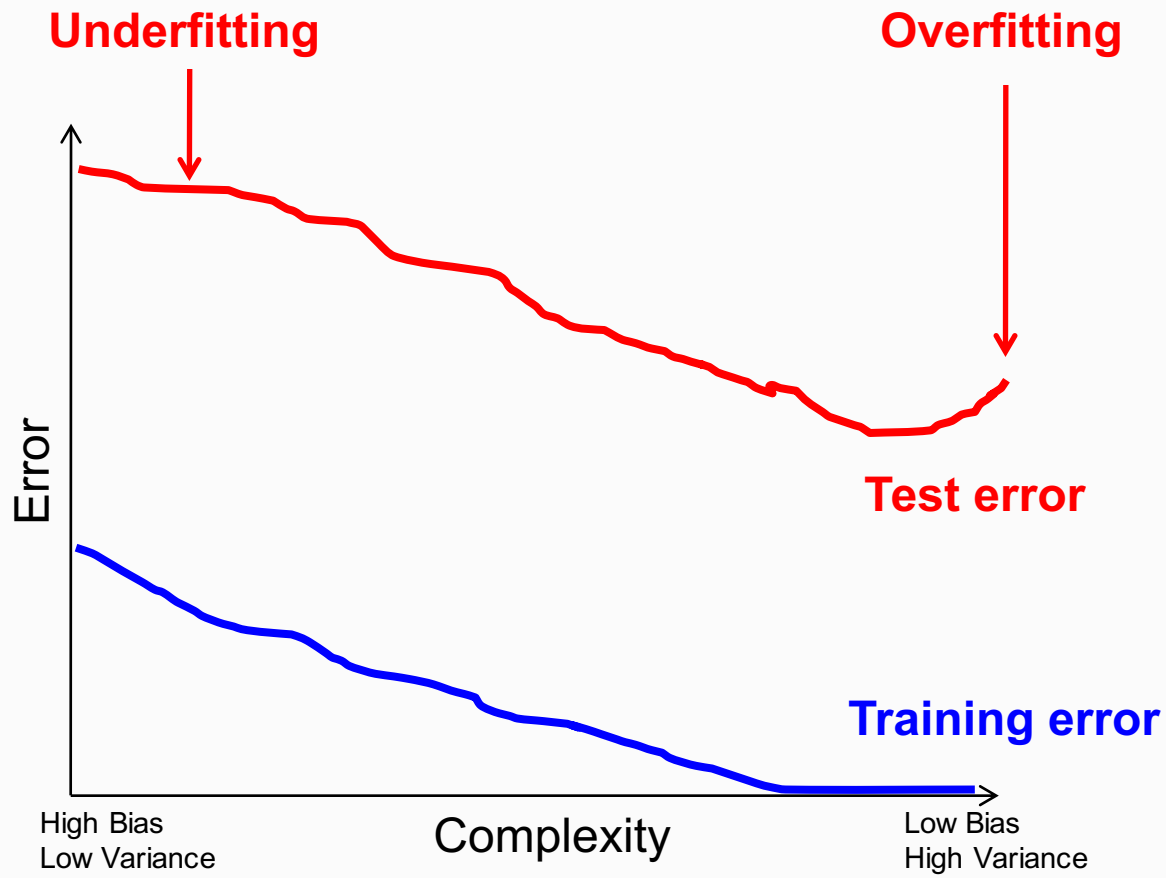
Error due to incorrect assumptions

Error due to variance of training samples

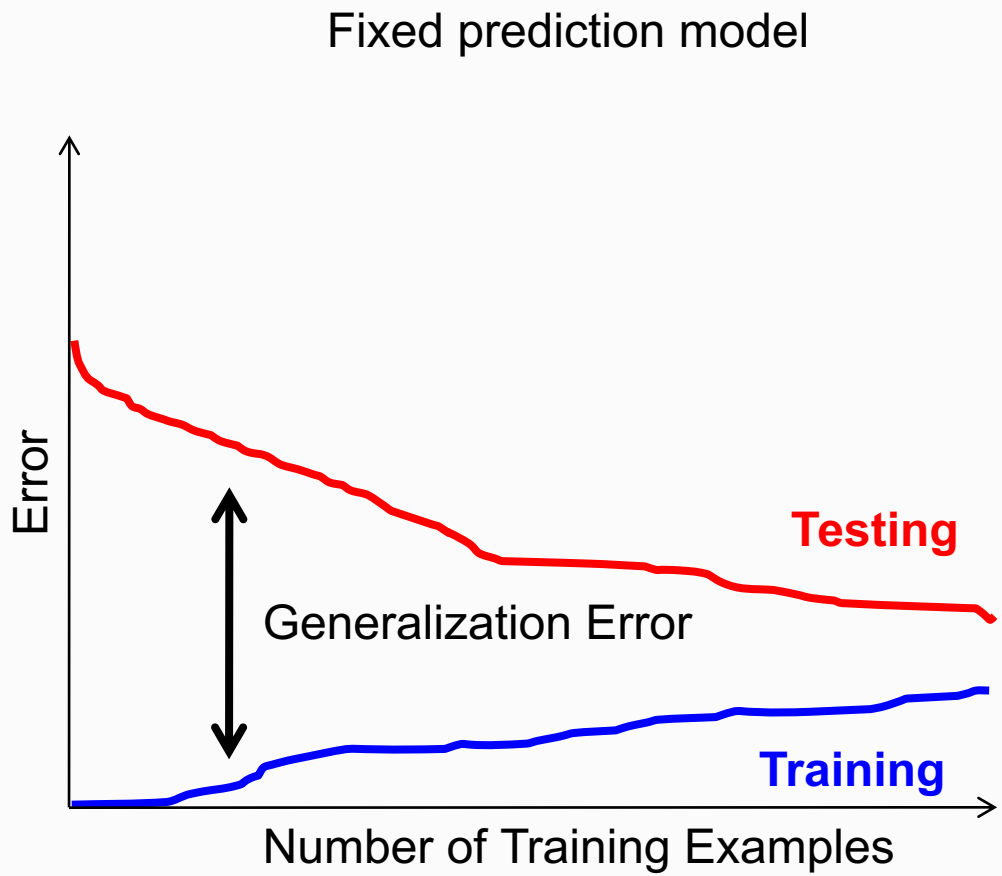
See the following for explanations of bias-variance (also Bishop's "Neural Networks" book):

- http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf

Bias-variance tradeoff



Effect of Training Size



Classification

How to reduce variance?

- Choose a simpler classifier
- Regularize the parameters
- Get more training data

What to remember about classifiers

- Try simple classifiers first
- Better to have smart features and simple classifiers than simple features and smart classifiers
- Use increasingly powerful classifiers with more training data (bias-variance tradeoff)

Lecture Outline

- Assignment 1 Discussion
- Ensembles, Random Forests
- Break
- Data Science Modelling
  Model performance evaluation…

Assignment 1

Design a simple, low-cost sensor that can distinguish between red wine and white wine for at least 95% of the samples. Your technology is capable of sensing the following wine attributes:

- **Fixed acidity**
- **Volatile acidity**
- **Citric acid**
- **Residual sugar**
- **Chlorides**
- **Density**
- **Free sulfur dioxide**
- **Total sulfur dioxide**
- **Sulfates**
- **pH**
- **Alcohol**

Sense as few of these attributes as possible to meet ~95%.

Objective: Gain familiarity with using a machine learning platform

Objective: Experience building and evaluating decision trees;

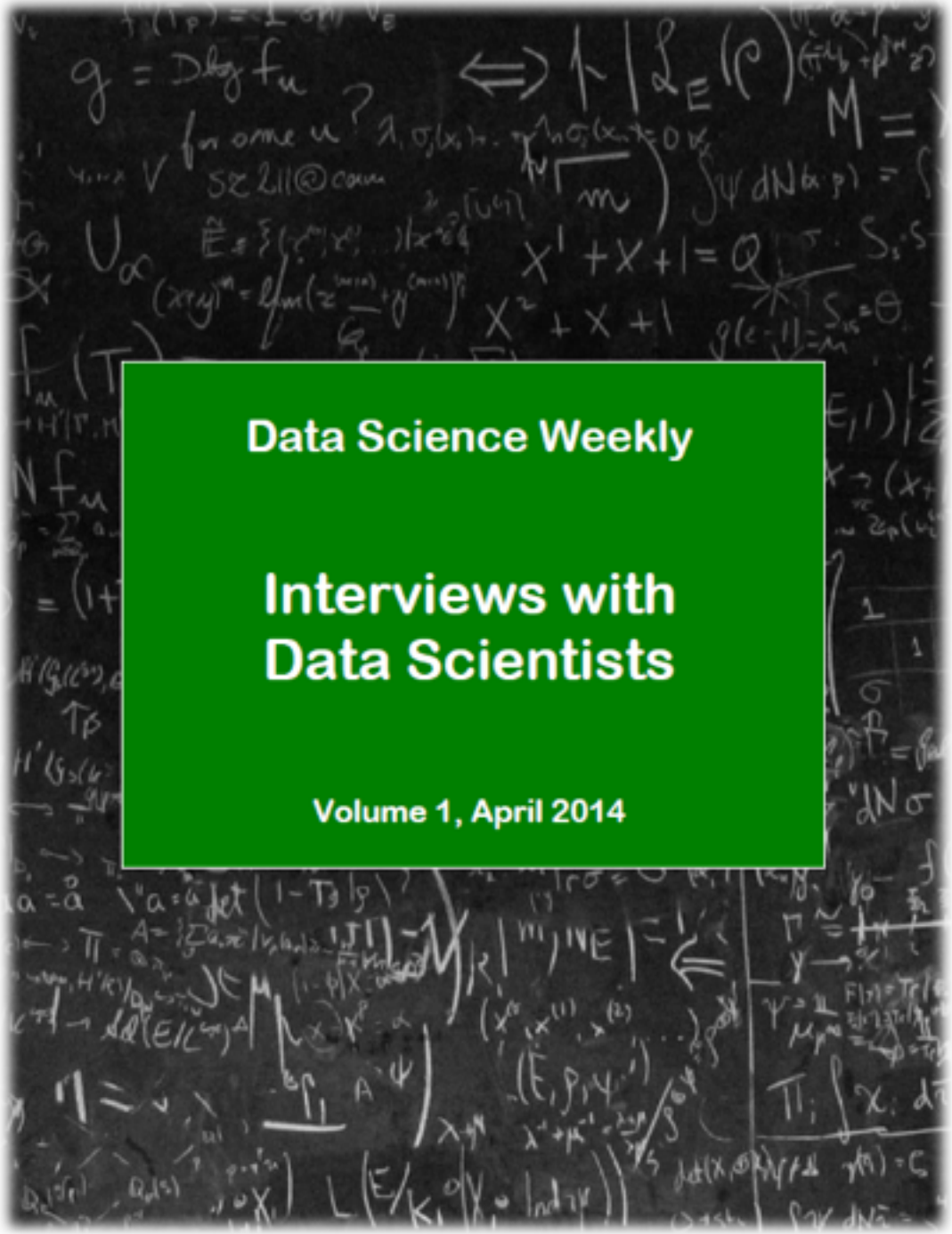
Objective: Interpreting, add/drop attributes – not just for making a prediction;

Robert M. Parker, Jr.

Wine Quality

Orley Ashenfelter (Princeton Economist)

Wine Quality = 12.145 + 0.00117 x (winter rainfall) + 0.0614 x (average growing season temperature)-0.00386 x (harvest rainfall).

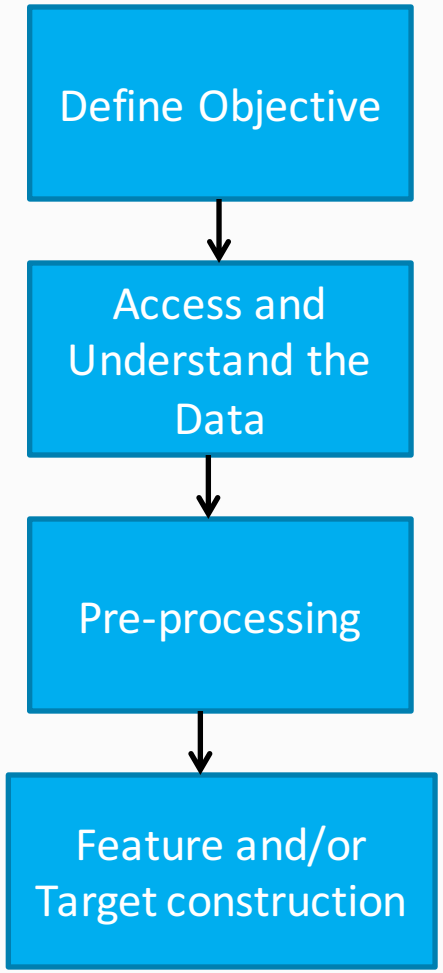


Optional Reading

Light reading, an engaging set of interviews with data scientists about their work, their experience, influences, lessons learned…

Doing Data Science

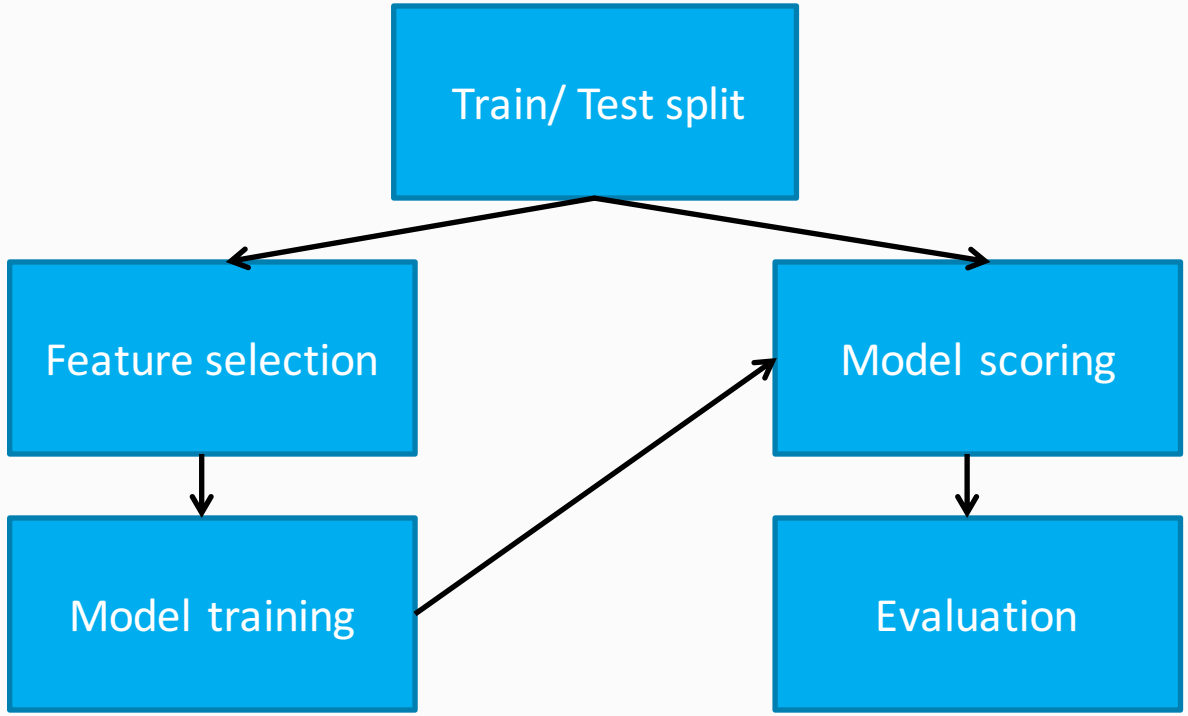
My Process Model



1. Define the objective and quantify it with a metric – optionally with constraints, if any. This typically requires domain knowledge.
2. Collect and understand the data, deal with the vagaries and biases in the data acquisition (missing data, outliers due to errors in the data collection process, more sophisticated biases due to the data collection procedure etc
3. Frame the problem in terms of a machine learning problem – classification, regression, ranking, clustering, forecasting, outlier detection etc. – some combination of domain knowledge and ML knowledge is useful.
4. Transform the raw data into a "modeling dataset", with features, weights, targets etc., which can be used for modeling. Feature construction can often be improved with domain knowledge. Target must be identical (or a very good proxy) of the quantitative metric identified step 1.

Doing Data Science

My Process Model



5. Train, test and evaluate, taking care to control bias/variance and ensure the metrics are reported with the right confidence intervals (cross-validation helps here), be vigilant against target leaks (which typically leads to unbelievably good test metrics) – this is the ML heavy step.

Business and Data Understanding

- What exactly is the business problem to be solved?
- Is the data science solution formulated appropriately to solve this business problem?
  NB: sometimes we have to make judicious approximations.
- What business entity does an instance/example correspond to?
- Is the problem a supervised or unsupervised problem?
  - If supervised,
  - Is a *target* variable defined?
  - If so, is it defined precisely?
  - Think about the values it can take.
- Are the attributes defined precisely?
  - Think about the values they can take.
- For supervised problems: will modeling this target variable actually improve the stated business problem? An important subproblem? If the latter, is the rest of the business problem addressed?
- Does framing the problem in terms of expected value help to structure the subtasks that need to be solved?
- If unsupervised, is there an "exploratory data analysis" path well defined? (That is, *where is the analysis going*?)

Data Preparation

- Will it be practical to get values for attributes and create feature vectors, and put them into a single table?
- If not, is an alternative data format defined clearly and precisely? Is this taken into account in the later stages of the project? (Many of the later methods/techniques assume the dataset is in feature vector format.)
- If the modeling will be supervised, is the target variable well defined? Is it clear how to get values for the target variable (for training and testing) and put them into the table?
- How exactly will the values for the target variable be acquired? Are there any costs involved? If so, are the costs taken into account in the proposal?
- Are the data being drawn from a population similar to that to which the model will be applied? If there are discrepancies, are any selection biases noted clearly? Is there a plan for how to compensate for them?

Modeling

- Is the choice of model appropriate for the choice of target variable?
  - Classification, class probability estimation, ranking, regression, clustering, etc.
- Does the model/modeling technique meet the other requirements of the task?
  - Generalization performance, comprehensibility, speed of learning, speed of application, amount of data required, type of data, missing values?
  - Is the choice of modeling technique compatible with prior knowledge of the problem (e.g., is a linear model being proposed for a definitely nonlinear problem)?
- Should various models be tried and compared (in evaluation)?
- For clustering, is there a similarity metric defined? Does it make sense for the business problem?

Evaluation and Deployment

- Is there a plan for domain-knowledge validation?
  - Will domain experts or stakeholders want to vet the model before deployment?
  - If so, will the model be in a form they can understand?
- Is the evaluation setup and metric appropriate for the business task? Recall the original formulation.
  - Are business costs and benefits taken into account?
  - For classification, how is a classification threshold chosen?
  - Are probability estimates used directly?
  - Is ranking more appropriate (e.g., for a fixed budget)?
  - For regression, how will you evaluate the quality of numeric predictions? Why is this the right way in the context of the problem?
- Does the evaluation use holdout data?
  - Cross-validation is one technique.
- Against what baselines will the results be compared?
  - Why do these make sense in the context of the actual problem to be solved?
  - Is there a plan to evaluate the baseline methods objectively as well?
- For clustering, how will the clustering be understood?
- Will deployment as planned actually (best) address the stated business problem?
- If the project expense has to be justified to stakeholders, what is the plan to measure the final (deployed) business impact?

Data Science Workflow.pdf

Develop your own for defining and evaluating project opportunities…

How to be a good data scientist - Be a good modeler…

***Example 1*: Amazon, big spenders.**

Target of the competition was to predict customers who spend a lot of money among customers **using past purchases**.

Data - Transaction data in different categories.

Winning model identified that 'Free shipping = True' was an **excellent** predictor.

***Leakage**: "Free Shipping = True" was simultaneous with the sale, which is a no-no… We can only use data from beforehand to predict the future…*

Know your Data

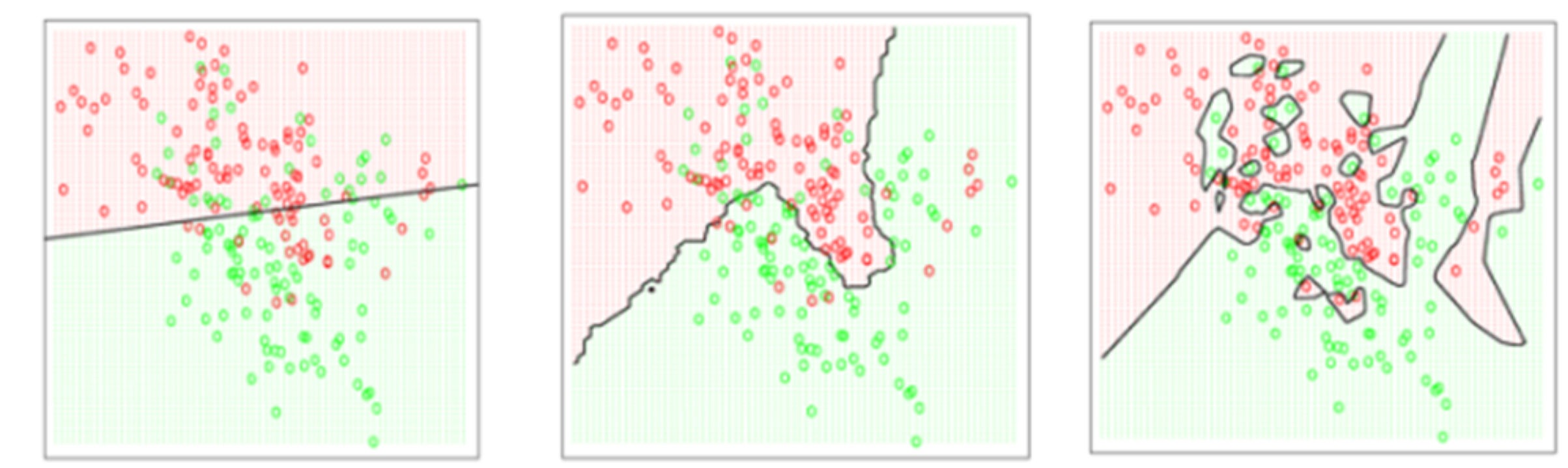
Avoid Leakage

Winning competition on leakage is easier than building good models. But even if you don't explicitly understand and game the leakage, ***your model will do it for you****. Either way, leakage is a huge problem.*

- *You need a strict temporal cutoff: remove all information just prior to the event of interest.*
- *There has to be a timestamp on every entry and you need to keep it*
- *The best practice is to start from scratch with clean, raw data after careful consideration*
- *You need to know how the data was created! I (try to) work only with data I pulled and prepared myself…*

Know Your Model

To avoid overfitting, we cross-validate and we cut down on the complexity of the model to begin with. Here's a standard picture (although keep in mind we generally work in high dimensional space and don't have a pretty picture to look at)



No model is perfect, great models are useful…

The Big Picture

You need to know what the purpose of the model is and how it is going to be used in order to decide how to do it and whether it's actually working…

The art in data science *is translating the problem into the language of data, features, proxy variable(s), a prediction,…*

The science in data science *is given raw data, constraints and a problem statement, you have an infinite set of models to choose from, with which you can use to maximize performance on some evaluation metric. Every design choice you make can be formulated as a hypothesis, upon which you will use* ***rigorous testing and experimentation to either validate or refute****.*

The Big Picture

Given

- data
- a problem, and
- constraints

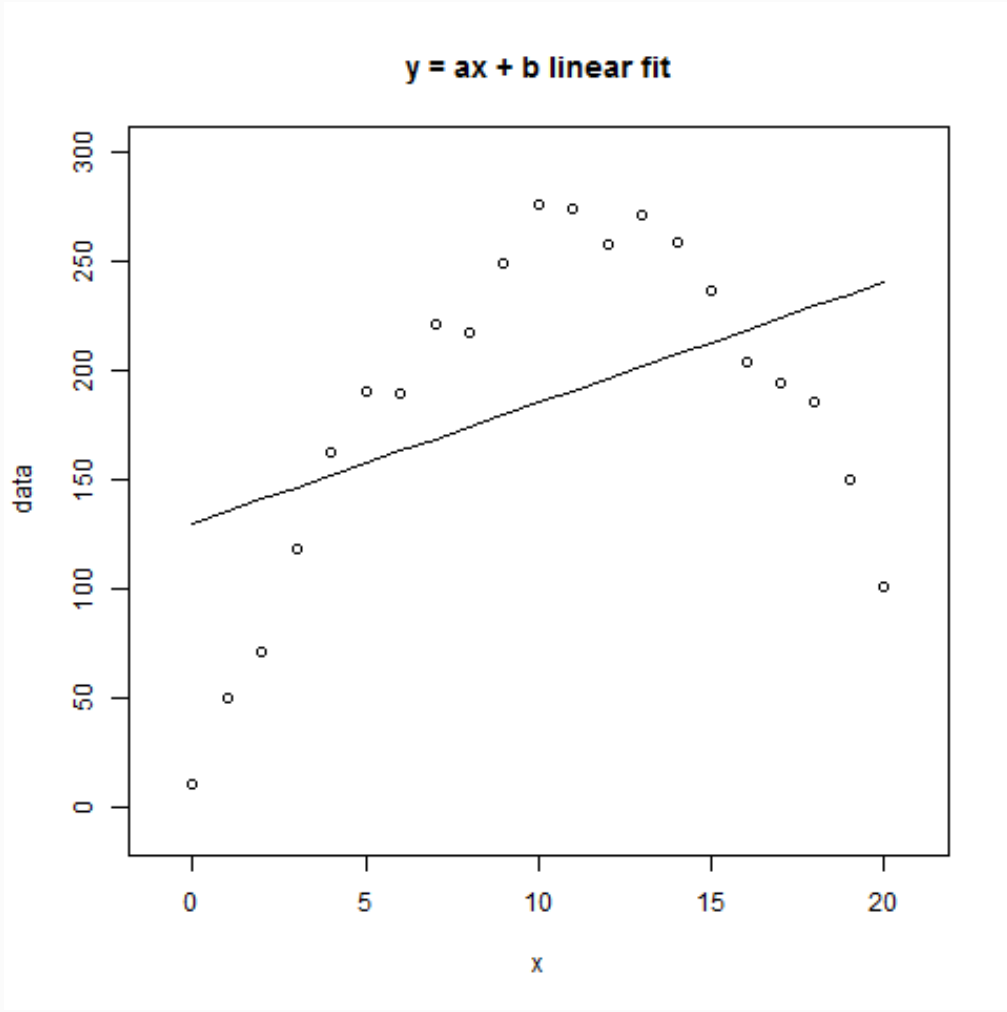
We need to determine

- features,
- a classifier,
- an optimization method, and
- an evaluation metric.

Later today we will focus on **evaluation metrics**...

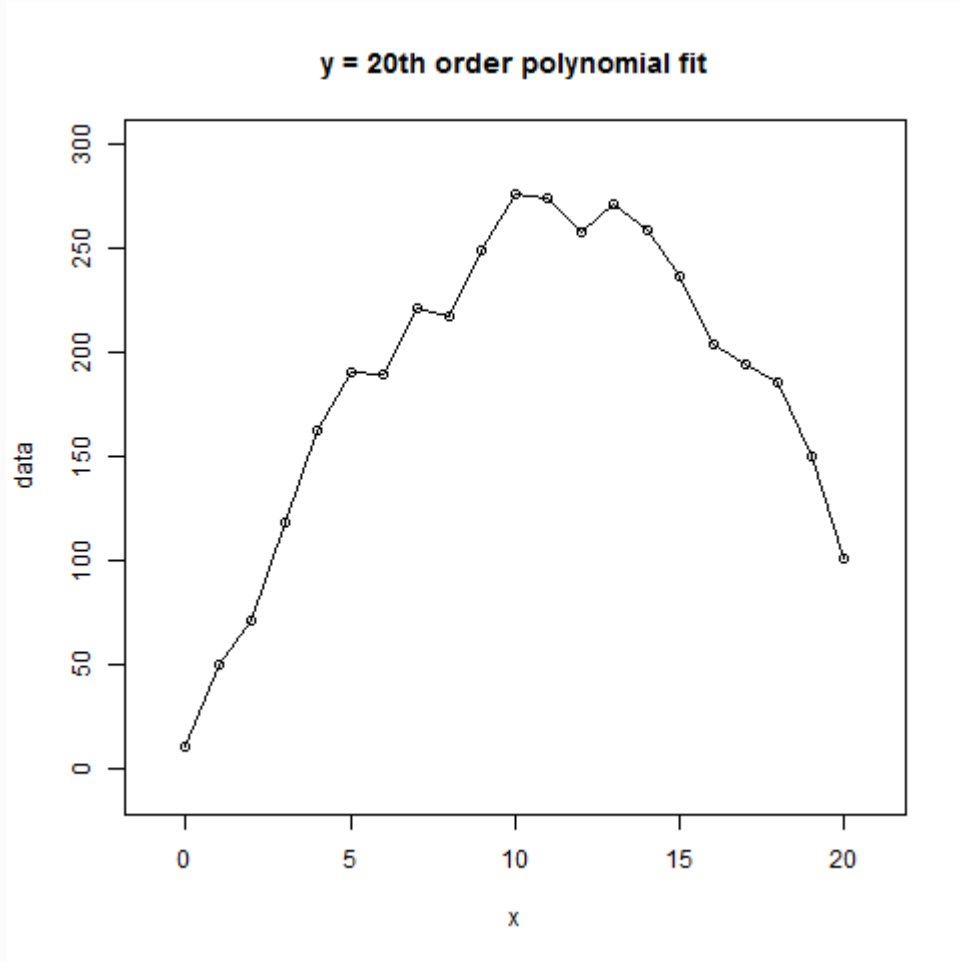
Review: Under-Fitting

- Under fit model to the data
  - Model is not expressive enough to capture the (quadratic polynomial) signal in the data
- Big approximation errors in training data (high bias)
- Little variance between multiple test datasets during prediction (low variance)



Review: Over-Fitting

- Over fit model to the data
  - Model has just learnt every point in the training data
- No approximation errors on training data (low bias)
- Generalize poorly across multiple test datasets during prediction (high variance)

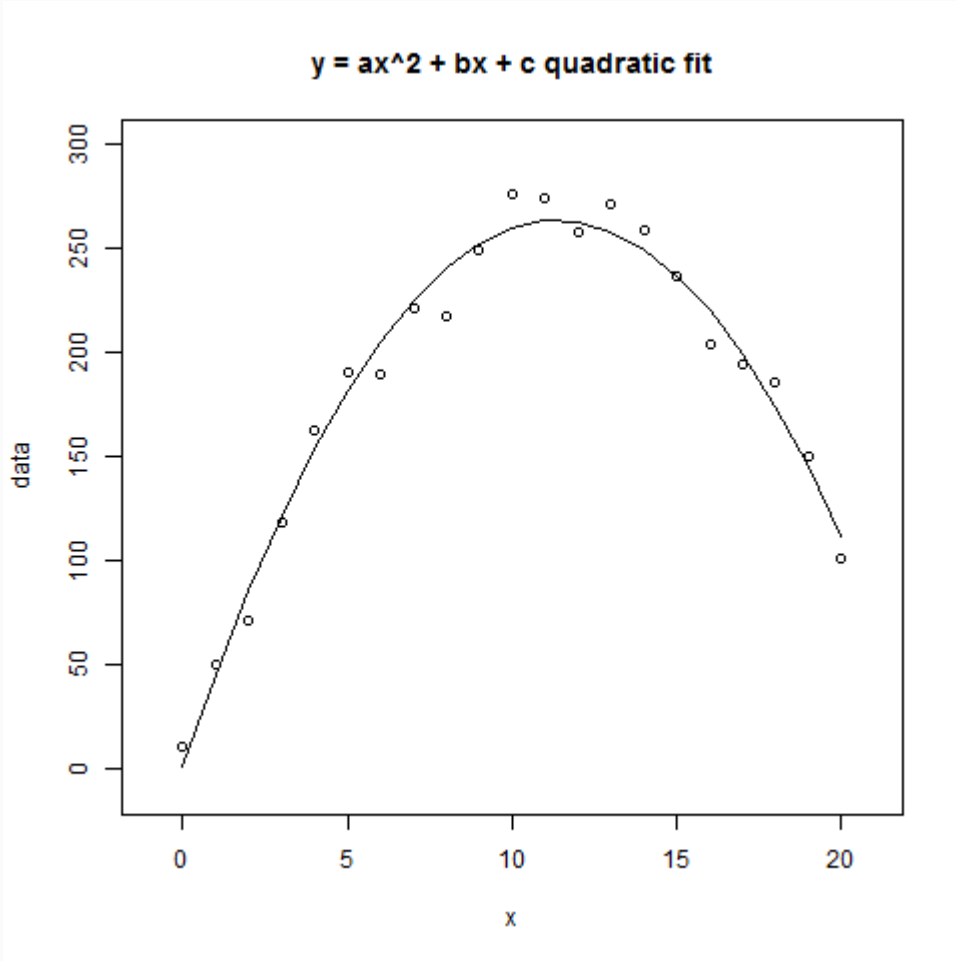


Bias-Variance Trade-Off

Trade-off between

- Capturing high level structure in data
  - Reduce bias
  - Provides some variation in new datasets
- Not memorize the training data
  - Reduce variance
  - Generalize to new datasets

Trade-off is called “Bias-Variance” trade-off



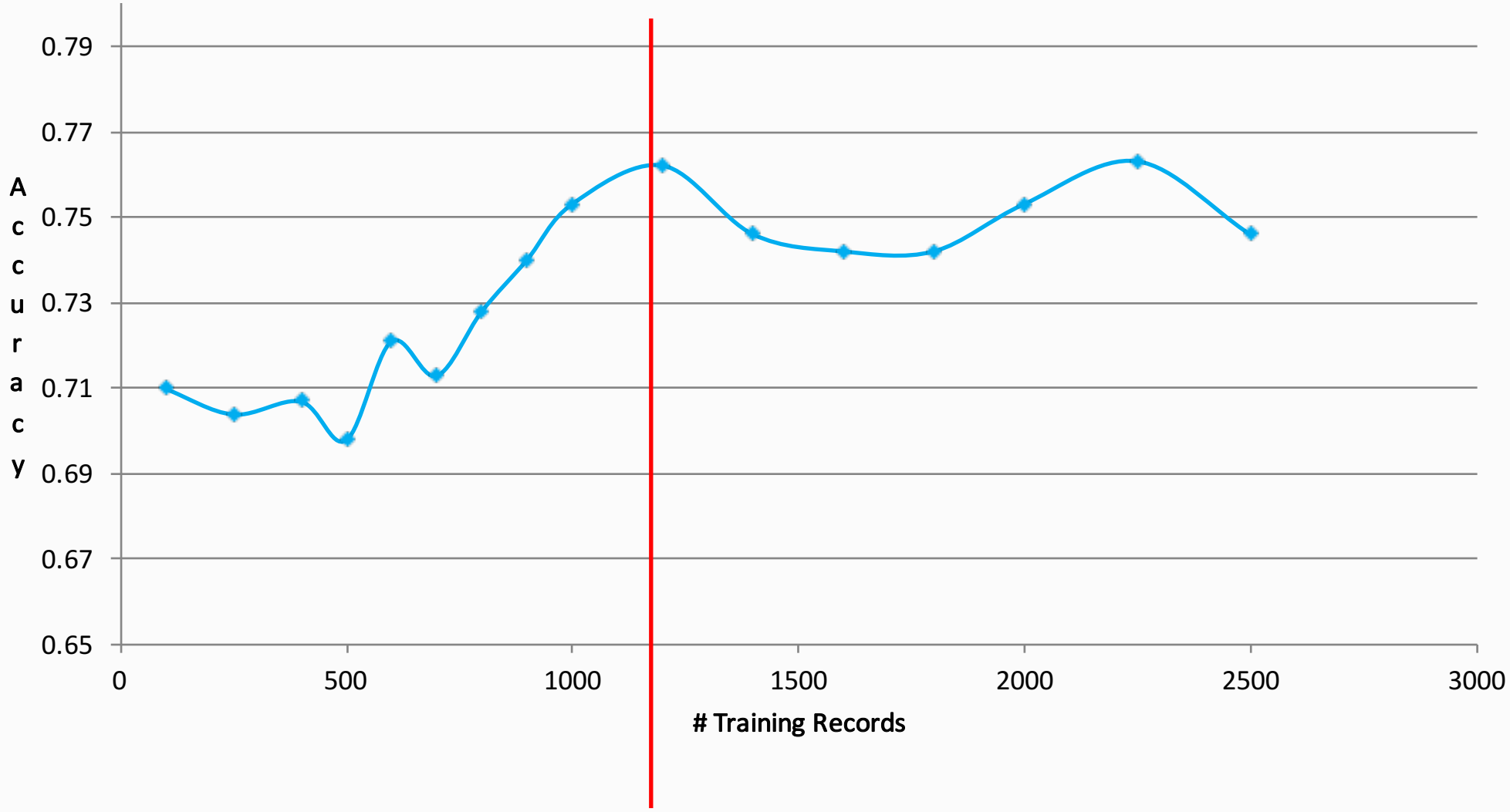
Reducing Under Fitting

- Increase model complexity, for e.g.
  - Increase the number of levels in a decision tree
  - Increase the number of hidden layers in a neural network.
  - Decrease the number of neighbors (k) in k-NN
- Increase the number of features
- In iterative training algorithms, iterate long enough so that the objective function has converged.

Reducing Over Fitting

- Decrease model complexity, for e.g.
  - Prune a decision tree
  - Reduce the number of hidden layers in a neural network.
  - Increase the number of neighbors (k) in k-NN
- Decrease the number of features
  - More aggressive feature selection
- Regularization (control feature complexity)
  - Penalize high weights.
  - L-1 regularization (LASSO) very efficient at pushing weights of non-informative features to 0.
- Gather more training data if possible
- In iterative training algorithms, stop training earlier to prevent "memorization" of training data

Sufficiency of Training Data (Example)



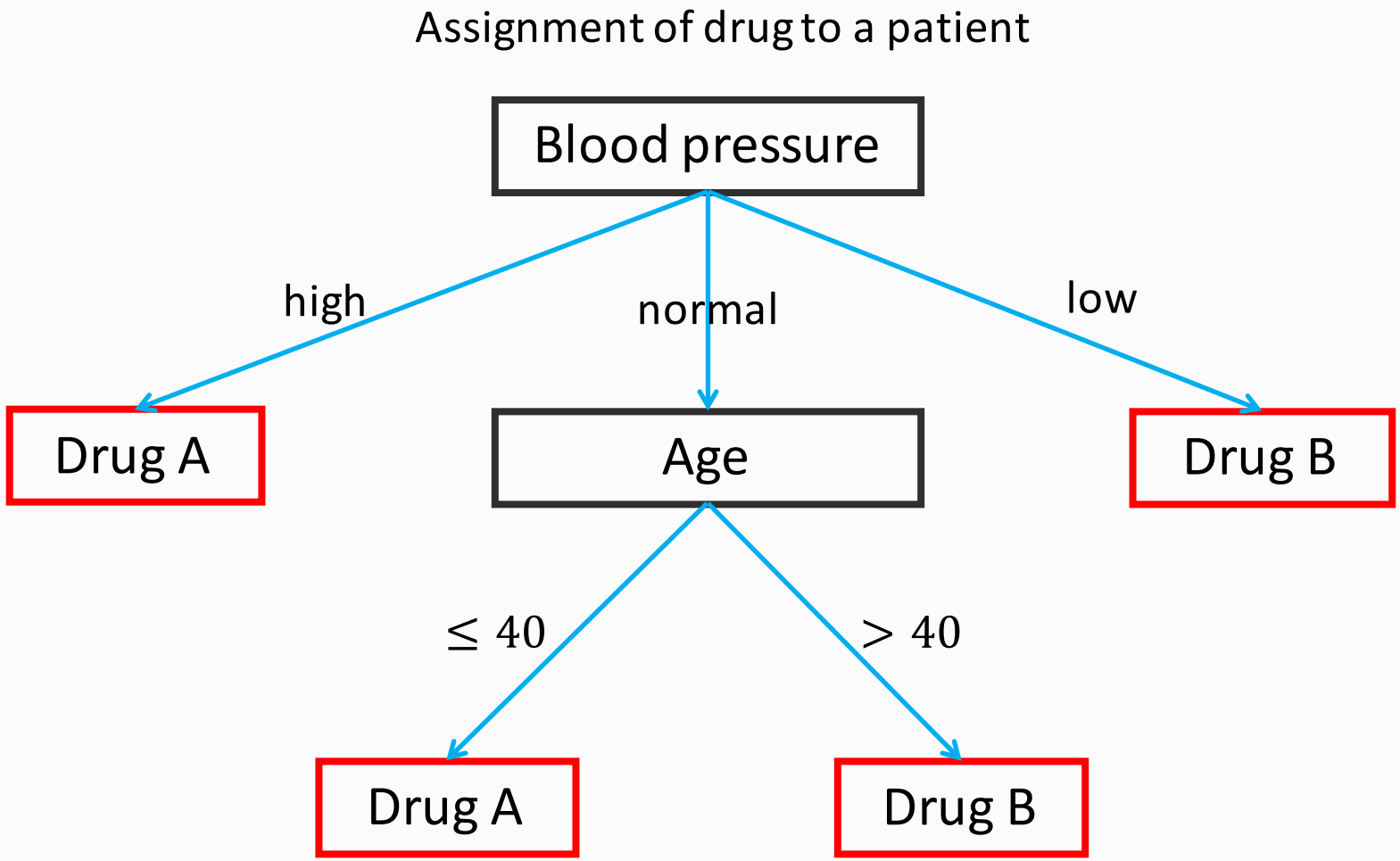
Accuracy on test data stabilizes above 1000 training samples

Review: Decision Tree

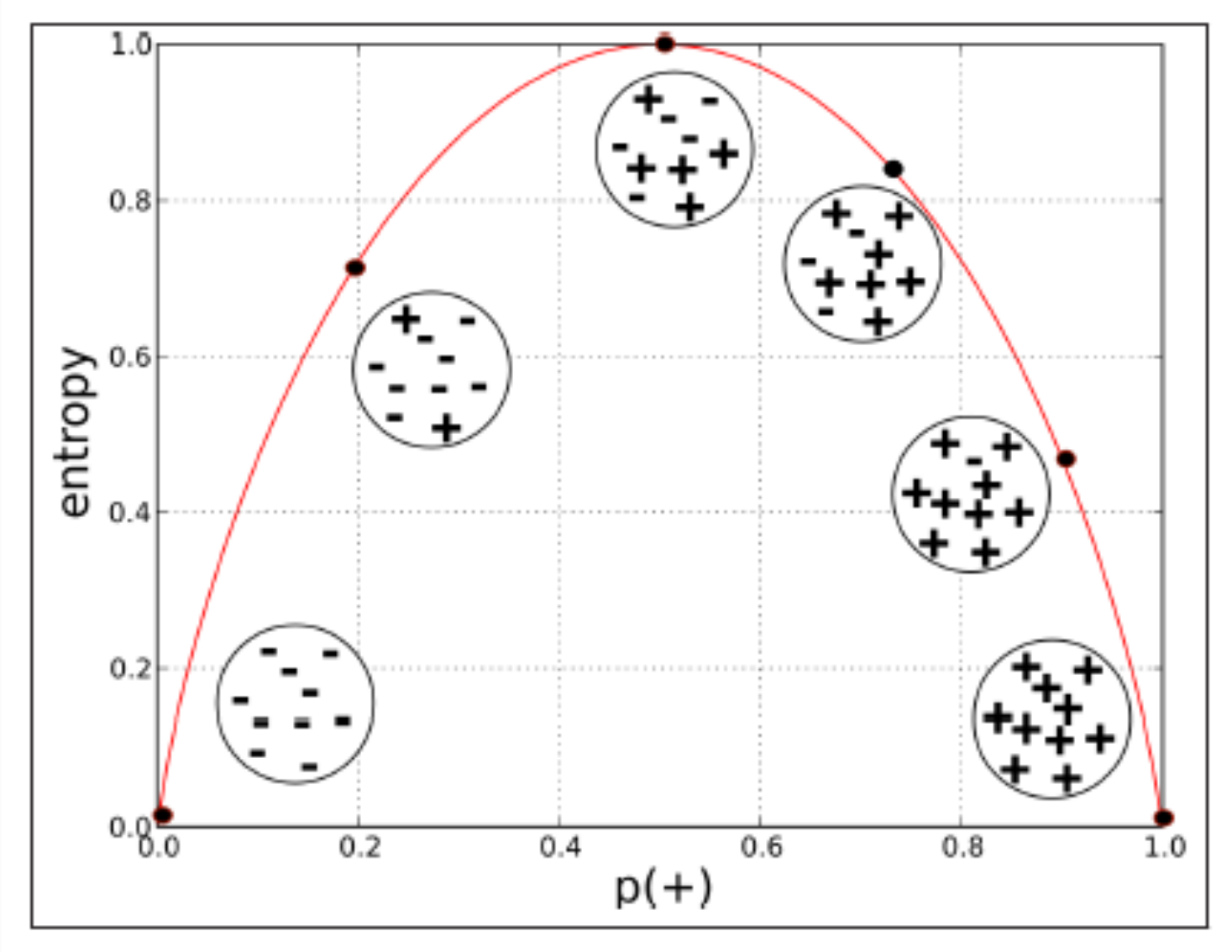
Unique Features

1. Automatically selects features
2. Able to handle large number of features
3. Numeric, nominal, missing
4. Easy to ensemble (Random Forrest, Boosted DT)
5. Transparent and easily explainable☺…

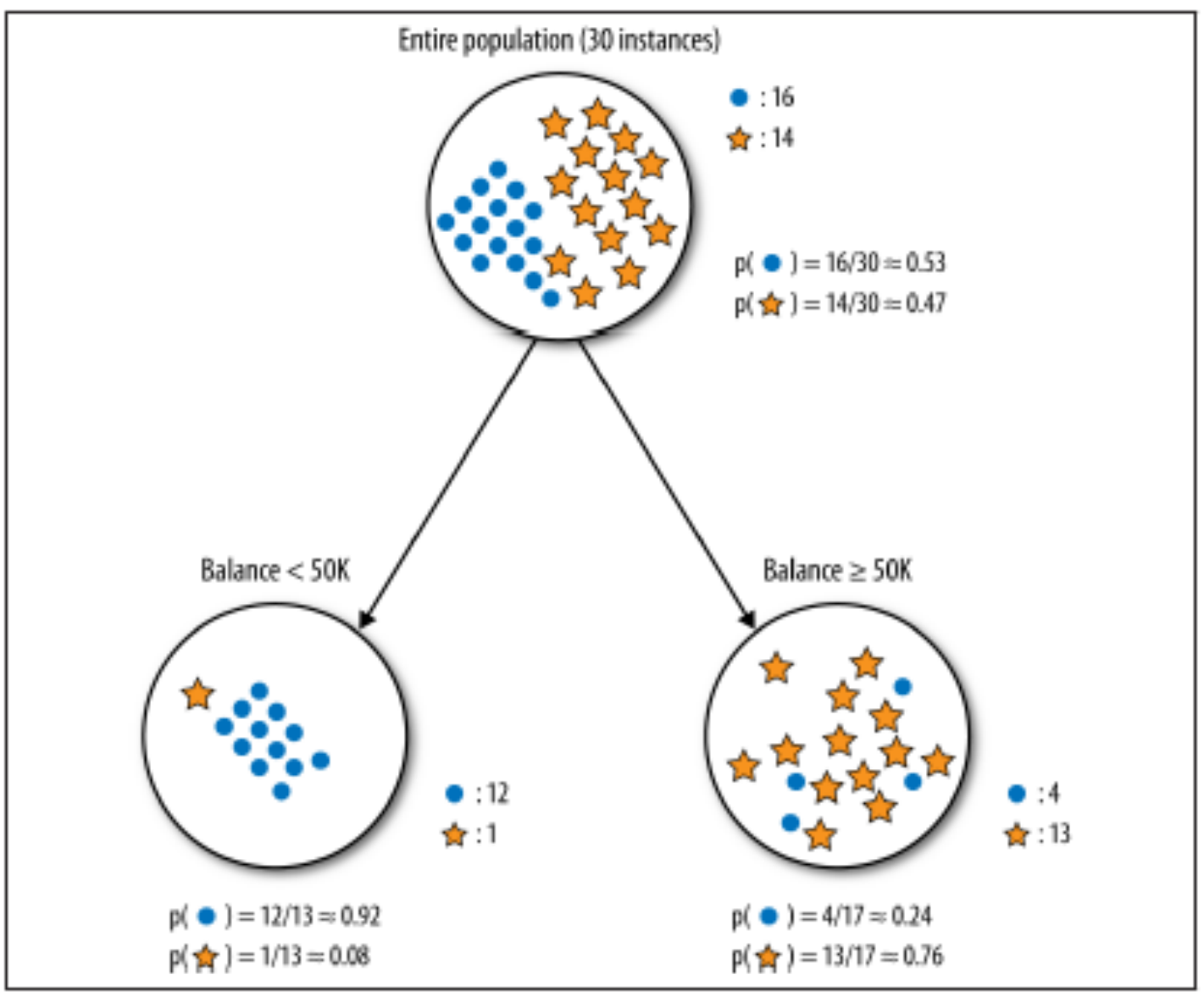
Review: Decision Tree



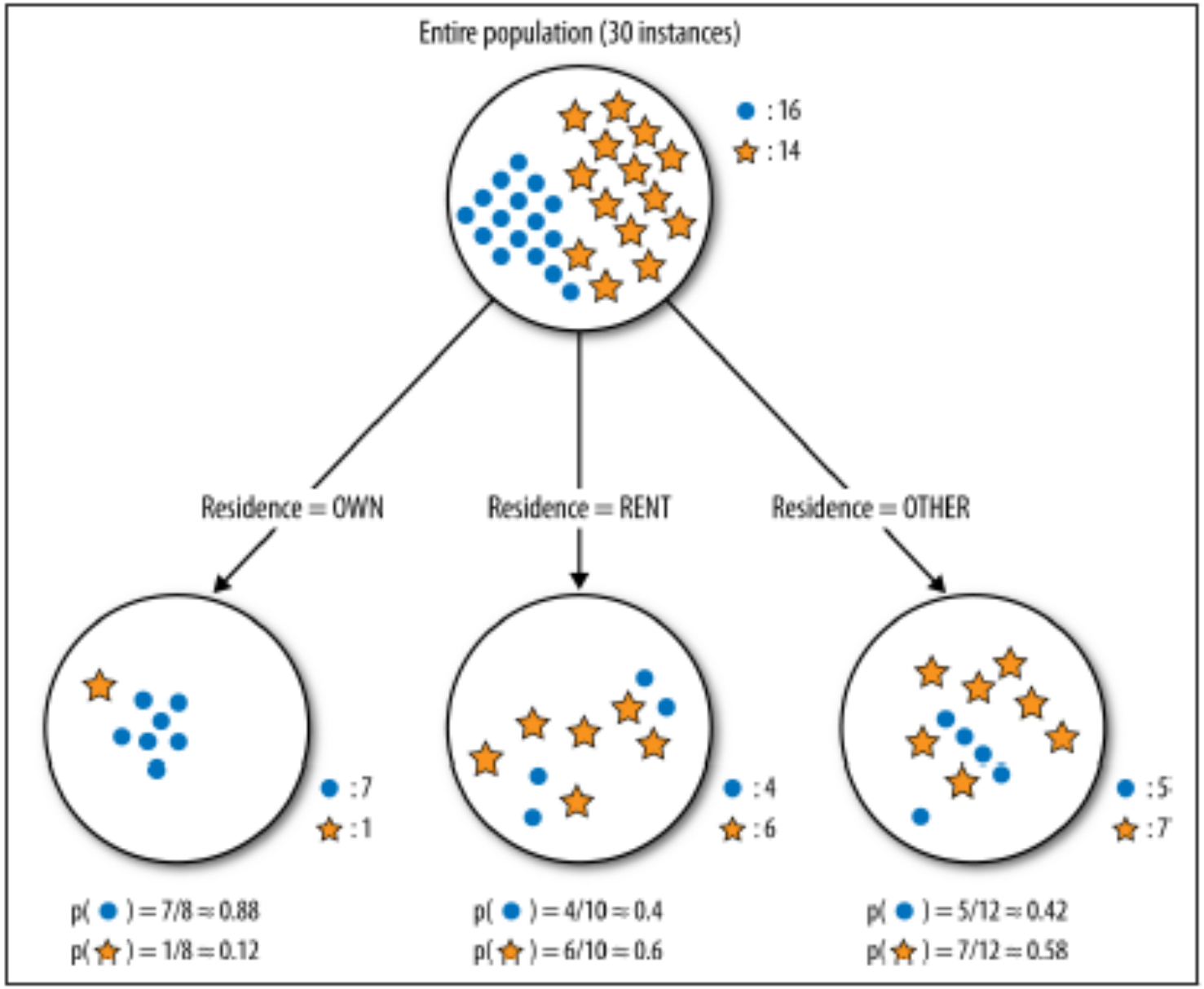
It’s all about minimizing the entropy (variance)…



It’s all about minimizing the entropy (variance)…



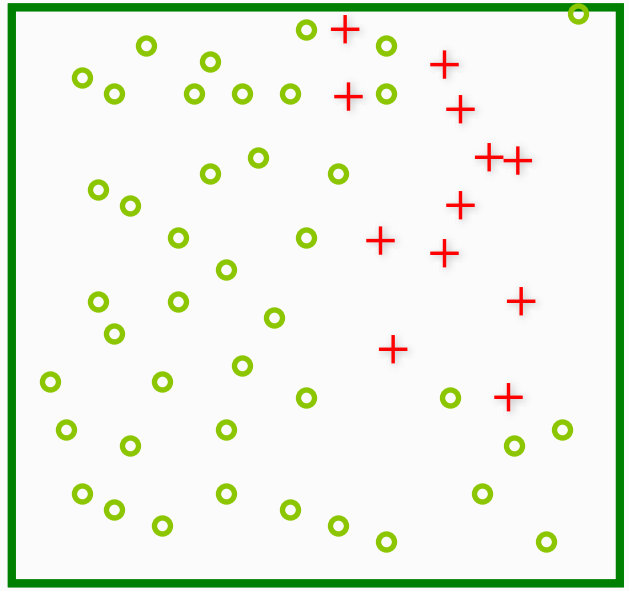
It’s all about minimizing the entropy (variance)…



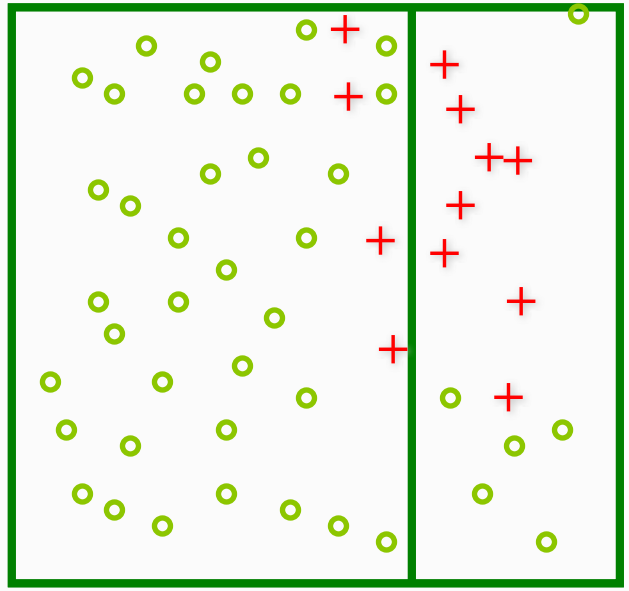
Review: Induction of decision trees

- Top down approach
  - Build the decision tree from top to bottom, from the root to the leaves
- Greedy selection of a test feature
  - Compute an evaluation measure for all features
  - Select the feature with the best measure
- Divide and Conquer/ Recursive Descent
  - Divide examples according to values of test feature
  - Apply the procedure recursively to the subsets
  - Terminate the recursion if
    - All cases belong to the same class, no more examples are available, cutoff condition has been satisfied (minimum node size)

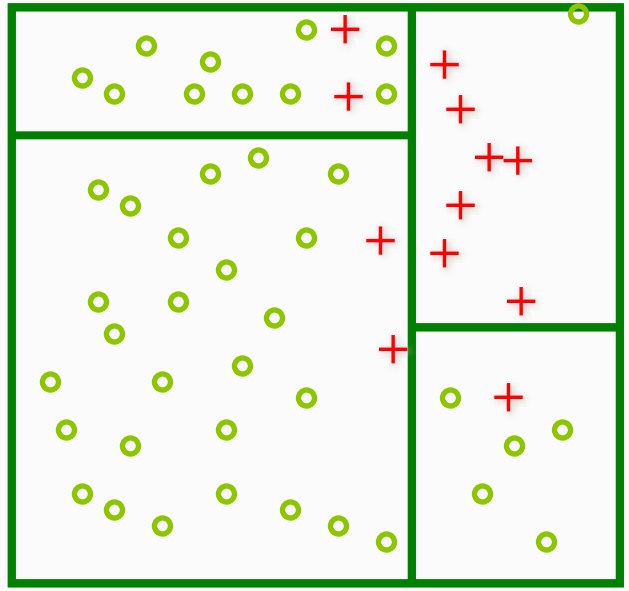
Review: Spatial example, recursive binary splits



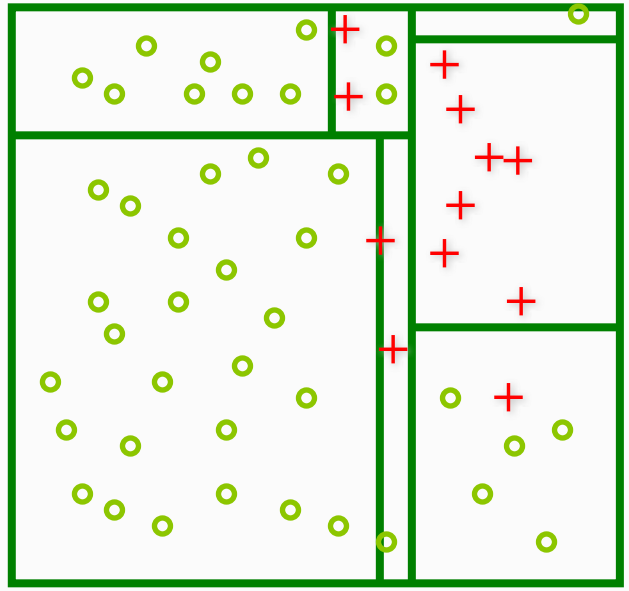
Review: Spatial example, recursive binary splits



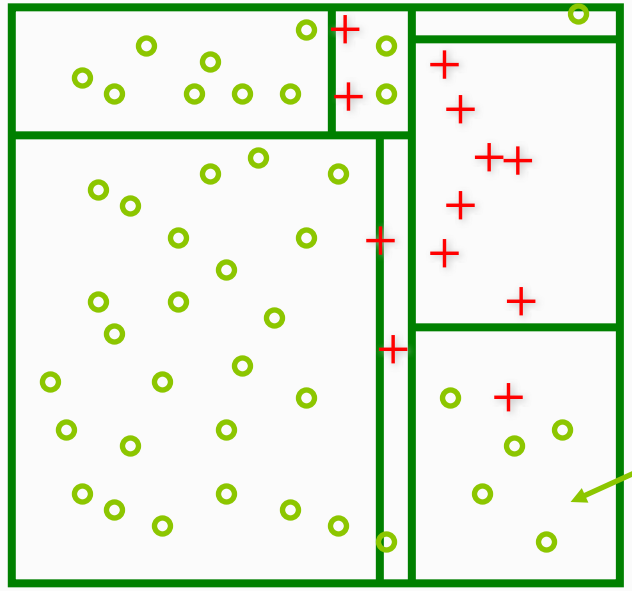
Review: Spatial example, recursive binary splits



Review: Spatial example, recursive binary splits



Review: Spatial example, recursive binary splits



Once regions are chosen class probabilities are easy to calculate

$p_m = 5/6$

Ensemble and Random Forests

Random Forest (Decision Forests)

Learning ensemble consisting of a bagging of un-pruned decision tree learners with a randomized selection of features at each split.

Decision trees are the individual learners that are combined. Decision trees are one of most popular learning methods commonly used for data exploration

- Powerful, explainable (decision trees, that is…)
- Automatic Feature Selection
- Ensembling

Why Ensemble with Decision Trees?

Decision trees: Advantages and limitations

Advantages:

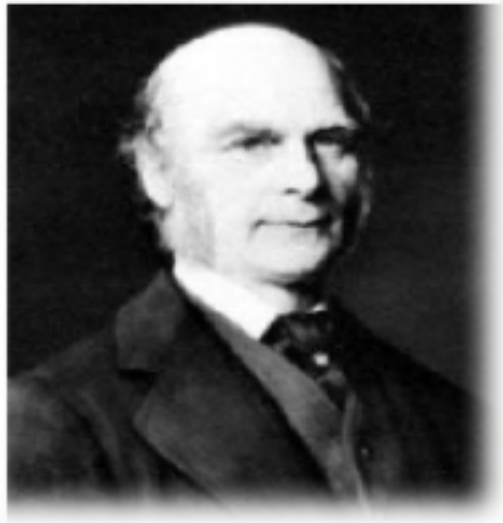
Characteristic	Neural nets	SVM	Trees	MARS	k-NN, kernels
Natural handling of data of "mixed" type	●	●	●	●	●
Handling of missing values	●	●	●	●	●
Robustness to outliers in input space	●	●	●	●	●
Insensitive to monotone transformations of inputs	●	●	●	●	●
Computational scalability (large N)	●	●	●	●	●
Ability to deal with irrelevant inputs	●	●	●	●	●
Ability to extract linear combinations of features	●	●	●	●	●
Interpretability	●	●	●	●	●
Predictive power	●	●	●	●	●

Limitations:

- Low prediction accuracy
- High variance
- Ensemble, to maintain advantages while increasing accuracy !

Francis Galton

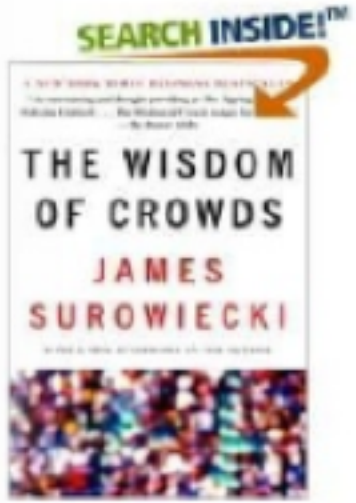
- Galton promoted statistics and invented the concept of correlation.
- In 1906 Galton visited a livestock fair and stumbled upon an intriguing contest.
- An ox was on display, and the villagers were invited to guess the animal's weight.
- Nearly 800 gave it a go and, not surprisingly, not one hit the exact mark: 1,198 pounds.
- Astonishingly, however, the average of those 800 guesses came close - very close indeed. It was 1,197 pounds.



The Wisdom of Crowds

Why the Many Are Smarter Than the Few and How Collective Wisdom

Shapes Business, Economies, Societies and Nations



- Under certain controlled conditions, the aggregation of information in groups, resulting in decisions that are often superior to those that can been made by any single - even experts.
- Imitates our second nature to seek several opinions before making any crucial decision. We weigh the individual opinions, and combine them to reach a final decision

Does it work…

Not all crowds (groups) are wise…

Example: investors in a stock market bubble (swing)

Key Criteria

Diversity of Opinion

– Each person should have private information even if it's just an eccentric interpretation of the known facts.

Independence

– People's opinions are not determined by the opinions of those around them.

Decentralization

– People are able to specialize and draw on local knowledge.

Aggregation

– Some mechanism exists to turn private judgments into a collective decision

So, how good are ensemble methods…

Netflix Prize

Began October 2006

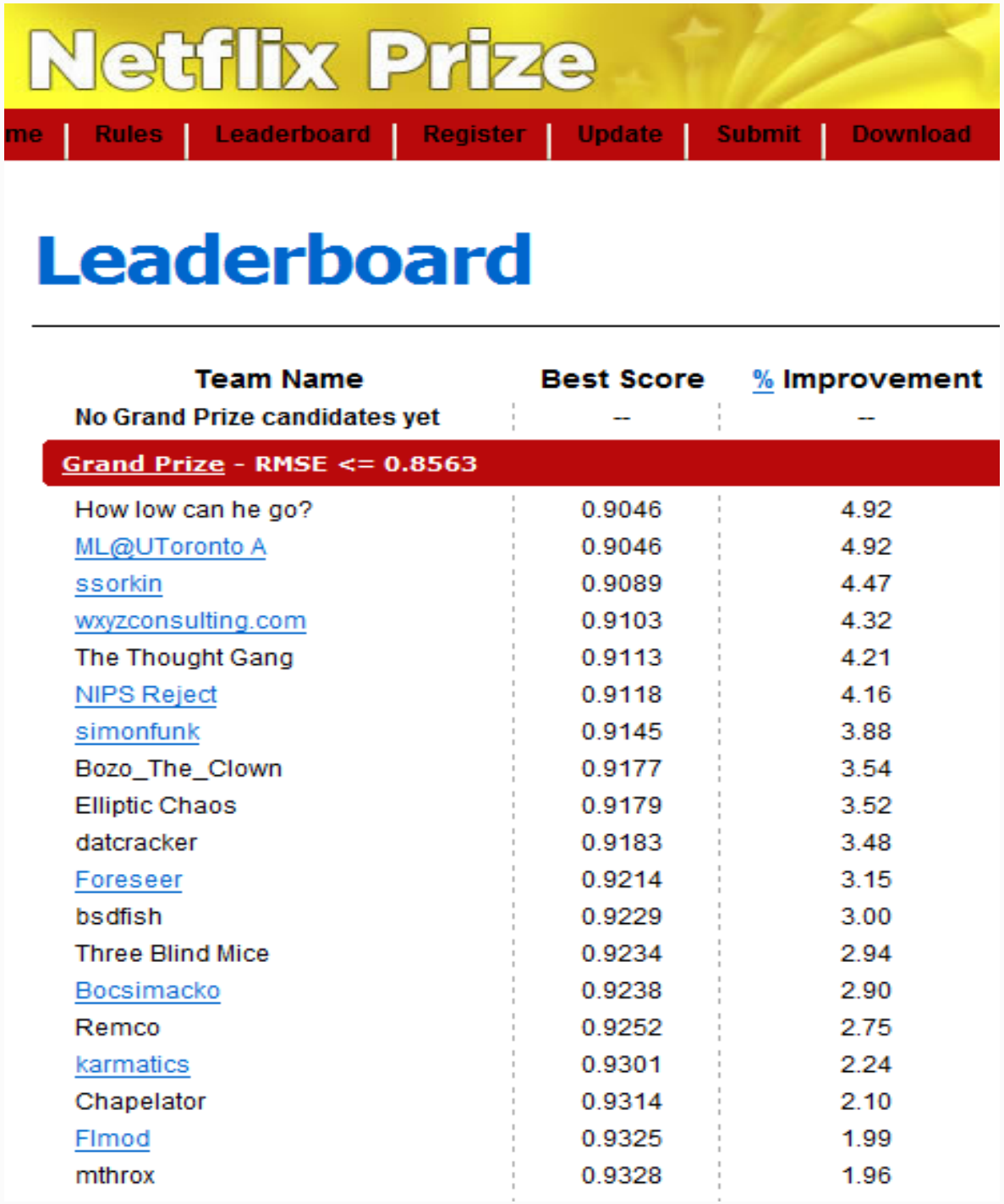
Supervised learning task

- Training data is a set of users and ratings (1,2,3,4,5 stars) those users have given to movies.
- Construct a classifier that given a user and an unrated movie, correctly classifies that movie as either 1, 2, 3, 4, or 5 stars

$1 million prize for a 10% improvement over Netflix's current movie recommender/classifier

(MSE = 0.9514)

http://www.wired.com/business/2009/09/how-the-netflix-prize-was-won/, a light read (highly suggested)

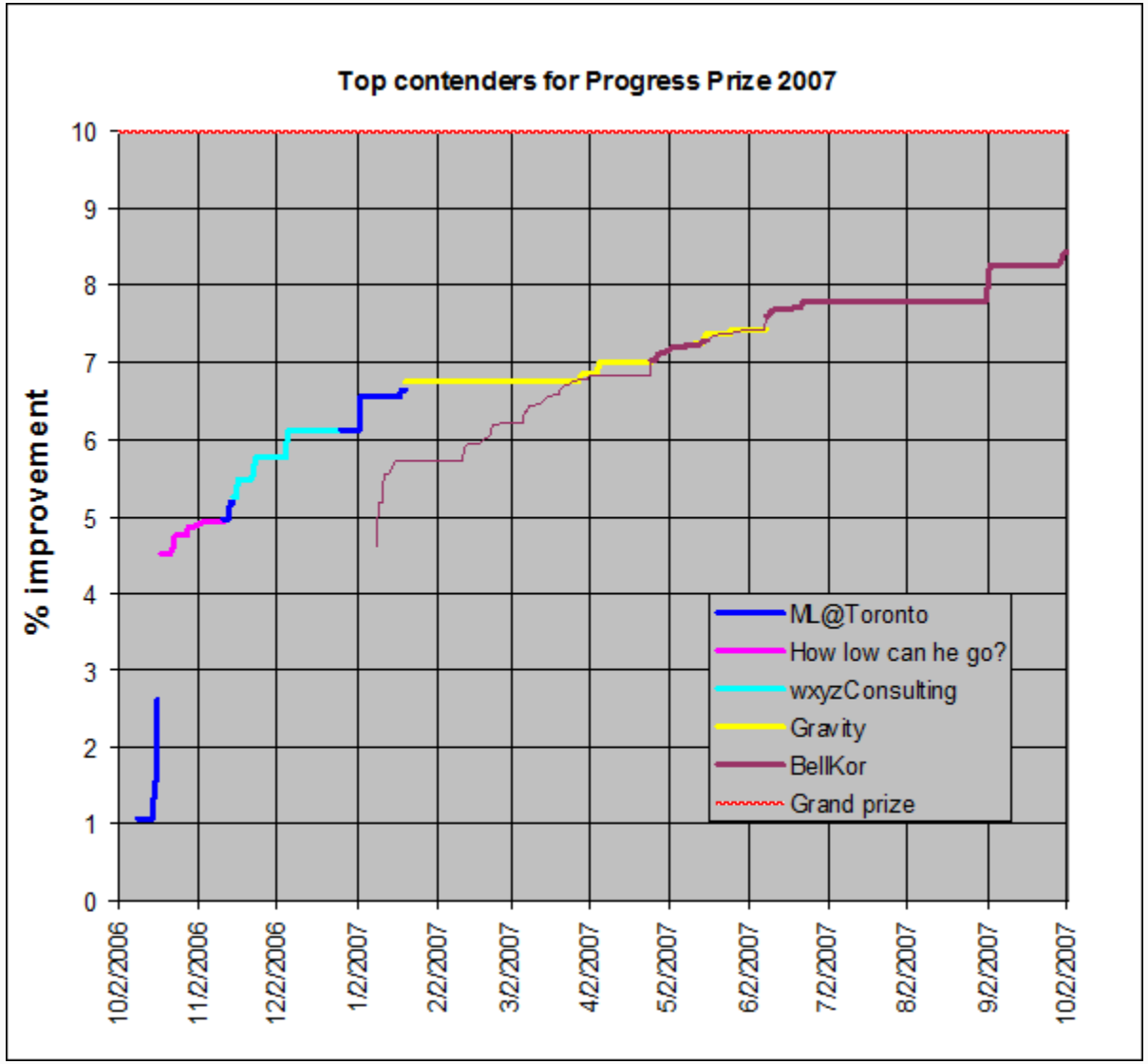


Team Name	Best Score	% Improvement
No Grand Prize candidates yet	--	--
Grand Prize - RMSE <= 0.8563		
How low can he go?	0.9046	4.92
ML@UToronto A	0.9046	4.92
ssorkin	0.9089	4.47
wxyzconsulting.com	0.9103	4.32
The Thought Gang	0.9113	4.21
NIPS Reject	0.9118	4.16
simonfunk	0.9145	3.88
Bozo_The_Clown	0.9177	3.54
Elliptic Chaos	0.9179	3.52
datcracker	0.9183	3.48
Foreseer	0.9214	3.15
bsdfish	0.9229	3.00
Three Blind Mice	0.9234	2.94
Bocsimacko	0.9238	2.90
Remco	0.9252	2.75
karmatics	0.9301	2.24
Chapelator	0.9314	2.10
Flmod	0.9325	1.99
mthrox	0.9328	1.96

Just three weeks after it began, at least 40 teams had bested the Netflix classifier.

Top teams showed about 5% improvement.

However, improvement slowed…



from http://www.research.att.com/~volinsky/netflix/

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

The top team posted a 8.5% improvement.

Ensemble methods are the best performers…

Rookies

“Thanks to Paul Harrison's collaboration, a simple mix of our solutions improved our result from 6.31 to 6.75”

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Arek Paterek

"My approach is to **combine the results of many methods** (also two-way interactions between them) using linear regression on the test set. The best method in my ensemble is regularized SVD with biases, post processed with kernel ridge regression"

http://rainbow.mimuw.edu.pl/~ap/ap_kdd.pdf

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

U of Toronto

"When the predictions of **multiple** RBM models and **multiple** SVD models are linearly combined, we achieve an error rate that is well over 6% better than the score of Netflix's own system."

http://www.cs.toronto.edu/~rsalakhu/papers/rbmcf.pdf

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Gravity

Table 5: Best results of single approaches and their combinations

Method/Combination	RMSE
MF	0.9190
NB	0.9313
CL	0.9606
NB + CL	0.9275
MF + CL	0.9137
MF + NB	0.9089
MF + NB + CL	0.9089

home.mit.bme.hu/~gtakacs/download/**gravity**.pdf

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

When Gravity and Dinosaurs Unite

"Our common team blends the result of team Gravity and team Dinosaur Planet."

Might have guessed from the name…

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

BellKor / KorBell

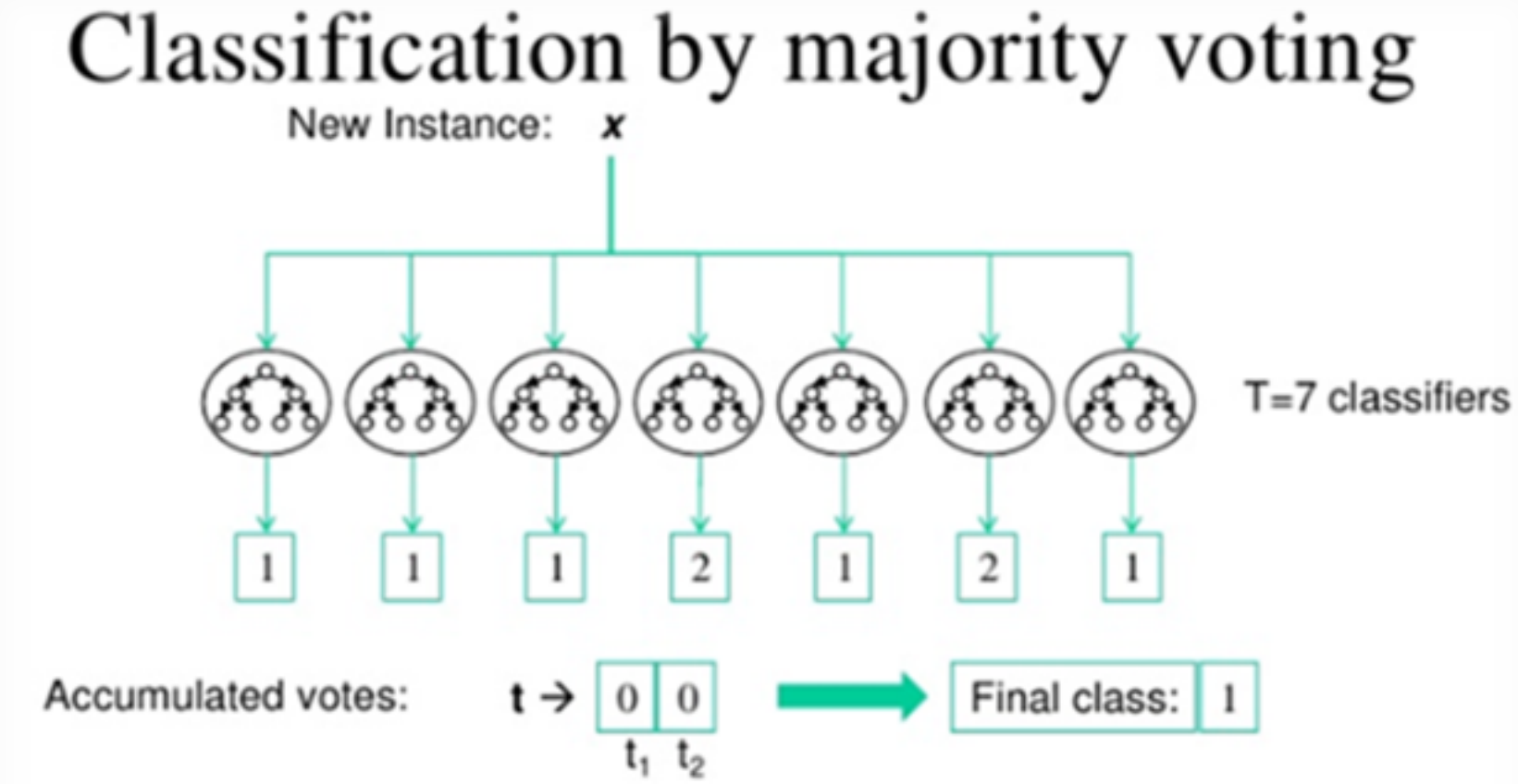
And, yes, the top team which is from AT&T…

“Our final solution (RMSE=0.8712) consists of blending 107 individual results. “

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Ensemble Classification

Aggregation of predictions of multiple classifiers with the goal of improving accuracy.



Bagging (Bootstrap Aggregating)

Given

– Training set of N examples

– A class of learning models (e.g. decision trees, neural networks, …)

Method

– Train multiple (k) models on different samples (data splits) and average their predictions

– Predict (test) by averaging the results of k models

Goal

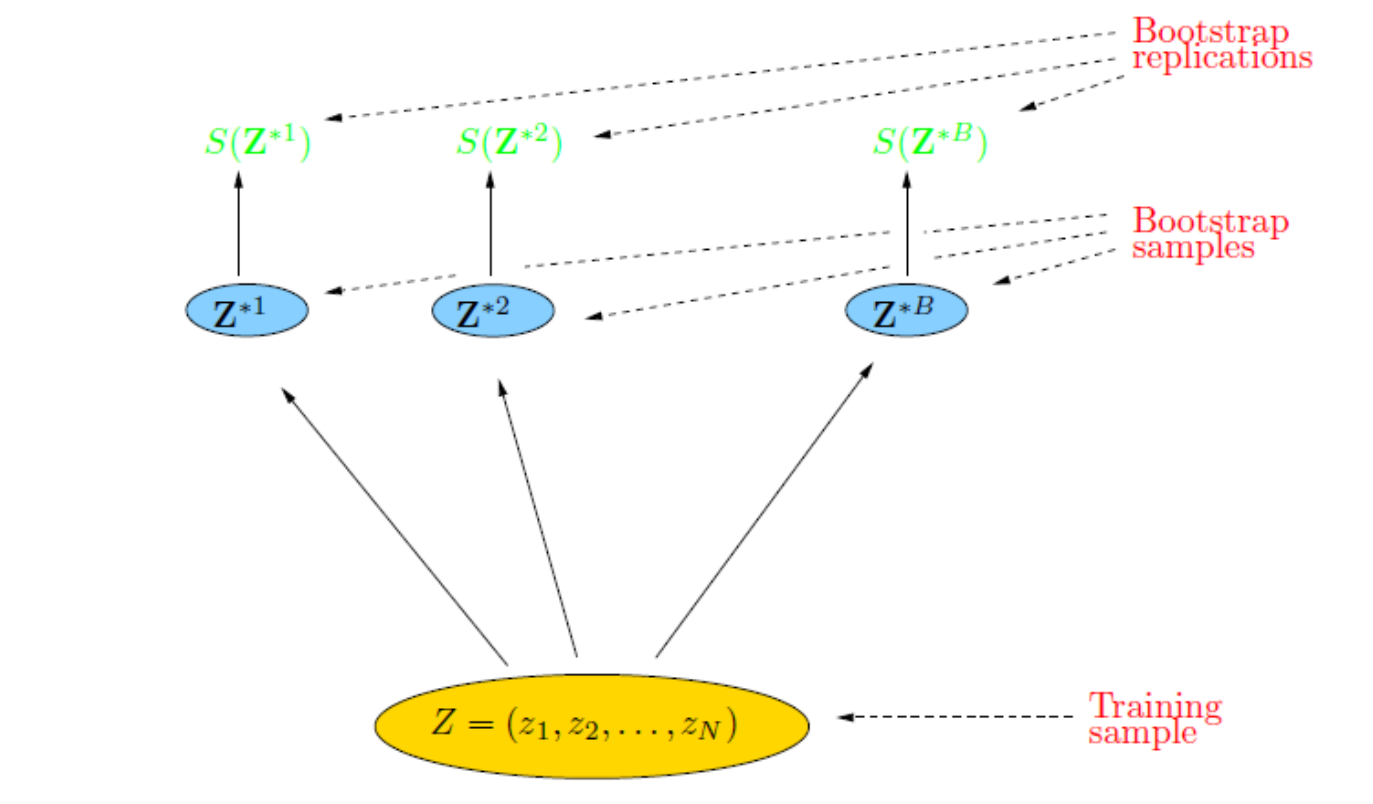
– Improve the accuracy of one model by using its multiple copies

– Average of misclassification errors on different data split gives a better estimate of the predictive ability of a learning method

Bootstrap

The basic idea:

Randomly draw datasets *with replacement* from the training data, each sample *the same size as the original training set*



Bagging Algorithm

Training

– In each iteration t, t=1,…T

- Randomly sample with replacement N samples from the training set
- Train a chosen “base model” (e.g. neural network, decision tree) on the samples

Test

– For each test example

- Start all trained base models
- Predict by combining results of all T trained models:
  - Regression: averaging
  - Classification: a majority vote

More on Bagging

Bagging works because it reduces variance by voting/averaging

- Note: in some hypothetical situations the overall error might increase;
- Usually, the more classifiers the better;

Can help a lot if data is noisy

Can also be applied to numeric prediction

If the learning algorithm is unstable, then bagging almost always improves performance. Bagging stable classifiers is not a good idea

- Which ones are unstable? Neural nets, decision trees, regression trees, linear regression
- Which ones are stable? K-nearest neighbors

What is Boosting?

- Analogy: Consult several doctors, based on a combination of weighted diagnoses—weight assigned based on the previous diagnosis accuracy
- How boosting works?
  - Weights are assigned to each training tuple
  - A series of k classifiers is iteratively learned
  - After a classifier Mi is learned, the weights are updated to allow the subsequent classifier, Mi+1, to pay more attention to the training tuples that were misclassified by Mi
  - The final M* combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy
- Boosting algorithm can be extended for the prediction of continuous values
- Comparing with bagging: boosting tends to achieve greater accuracy, but it also risks overfitting the model to misclassified data

The Basic Idea

- Suppose there are just 5 training examples {1,2,3,4,5}
- Initially each example has a 0.2 (1/5) probability of being sampled
- 1st round of boosting samples (with replacement) 5 examples:
- {2, 4, 4, 3, 2} and builds a classifier from them
- Suppose examples 2, 3, 5 are correctly predicted by this classifier, and examples 1, 4 are wrongly predicted:
  - Weight of examples 1 and 4 is increased,
    Weight of examples 2, 3, 5 is decreased
- 2nd round of boosting samples again 5 examples, but now examples 1 and 4 are more likely to be sampled
- And so on …until some convergence is achieved

Boosting

- Also uses voting/averaging
- Weights models according to performance
- Iterative: new models are influenced by the performance of previously built ones
  - Encourage new model to become an “expert” for instances misclassified by earlier models
  - Intuitive justification: models should be experts that complement each other
- Several variants
  - Boosting by sampling, the weights are used to sample the data for training
  - Boosting by weighting, the weights are used by the learning algorithm

Random forests

- Random forests (RF) are a combination of decision tree predictors
- Each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest
- The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them
- Using a random selection of features to split each node yields error rates that compare favorably to Adaboost, and are more robust with respect to noise

Random forests

```
D = training set                          F = set of tests
k = nb of trees in forest                 n = nb of tests

for i = 1 to k do:
   build data set Di by sampling with replacement from D
   learn tree Ti (Tilde) from Di:
      at each node:
         choose best split from random subset of F of size n
      allow aggregates and refinement of aggregates in tests

make predictions according to majority vote of the set of k trees.
```

Random Forests

- Ensemble method tailored for decision tree classifiers
- Creates k decision trees, where each tree is independently generated based on random decisions
- Bagging using decision trees can be seen as a special case of random forests where the random decisions are the random creations of the bootstrap samples

Two examples of random decisions in RFs

- At each internal tree node, randomly select F attributes, and evaluate just those attributes to choose the partitioning attribute
  - Tends to produce trees larger than trees where all attributes are considered for selection at each node, but different classes will be eventually assigned to different leaf nodes, anyway
    - Saves processing time in the construction of each individual tree, since just a subset of attributes is considered at each internal node

- At each internal tree node, evaluate the quality of all possible partitioning attributes, but randomly select one of the F best attributes to label that node (based on InfoGain, etc.)
  - Unlike the previous approach, does not save processing time

Random forest: first randomization through bagging

Bagging or *bootstrap aggregation is* a technique for reducing the variance of an estimated prediction function.

Bootstrap sample = create new training sets by random sampling the given one $N' \leq N$ times with replacement

Bootstrap aggregation, parallel combination of learners, independently trained on distinct bootstrap samples

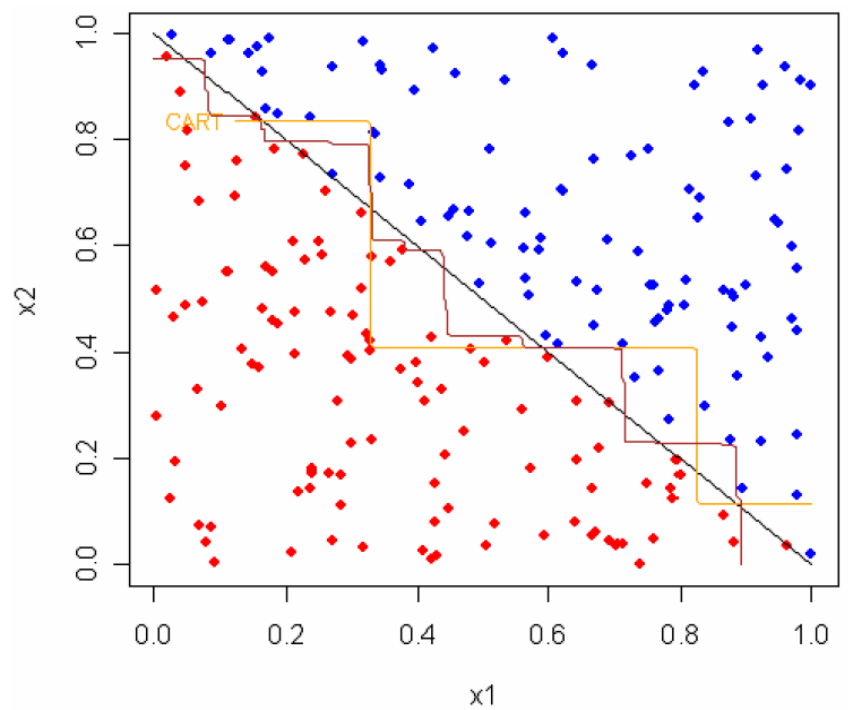
Final prediction is the mean prediction (regression) or class with maximum votes (classification).

Using squared error loss, bagging alone decreases test error by lowering prediction variance, while leaving bias unchanged.

Bagging: reduces variance – Example 1

- Two categories of samples: blue, red
- Two predictors: x1 and x2

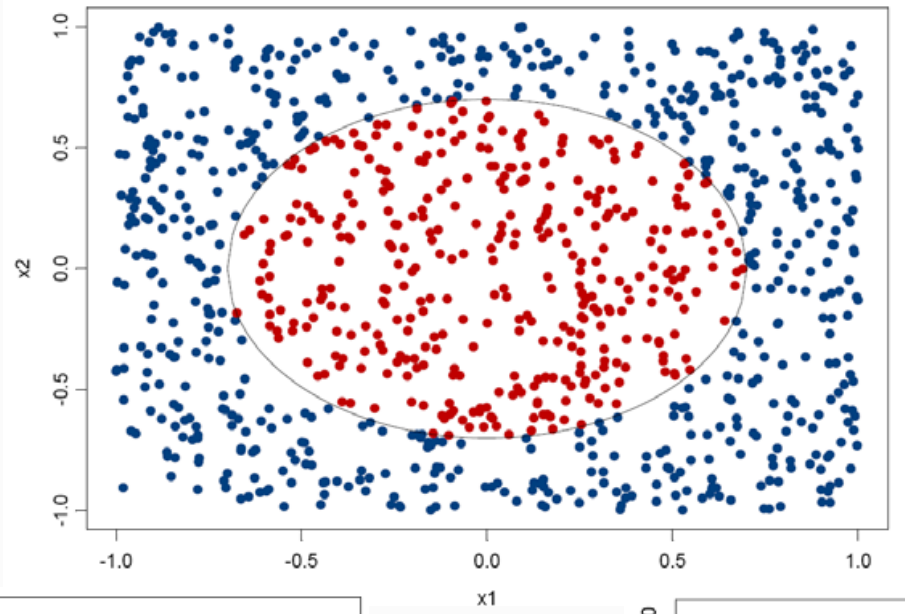
  Diagonal separation…hardest case for tree-based classifier

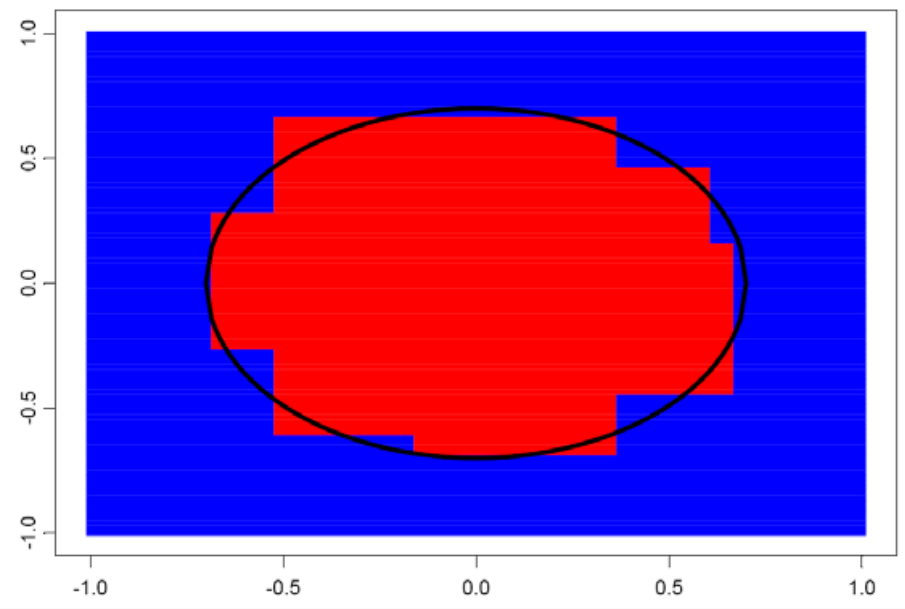


- Single tree decision boundary in orange.
- Bagged predictor decision boundary in red.

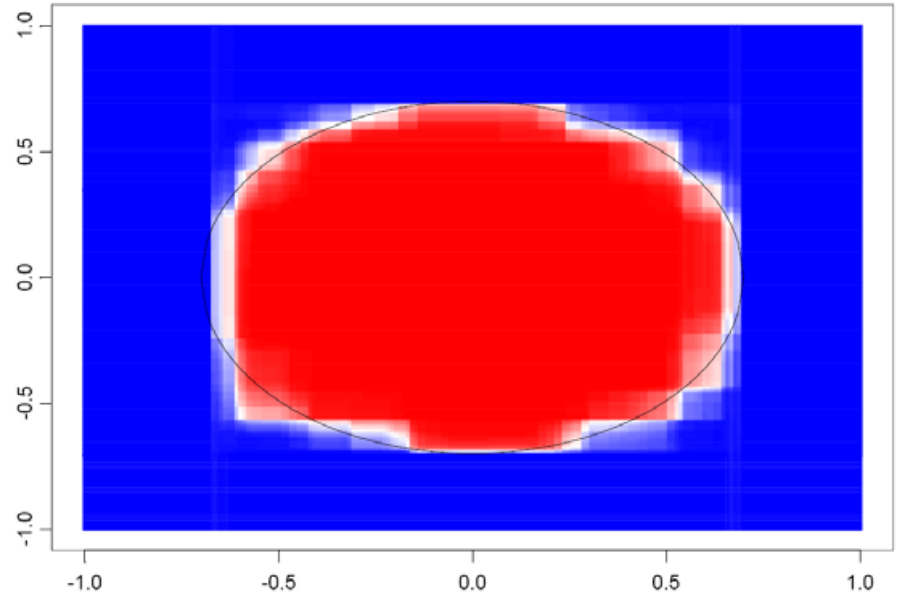
Bagging: reduces variance – Example 2

Ellipsoid separation →
Two categories,
Two predictors





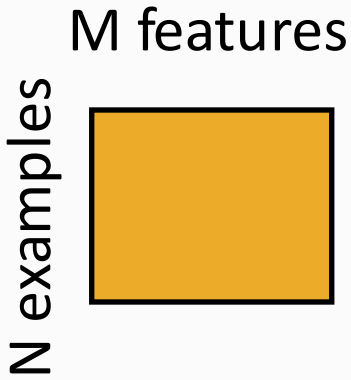
Single tree decision boundary



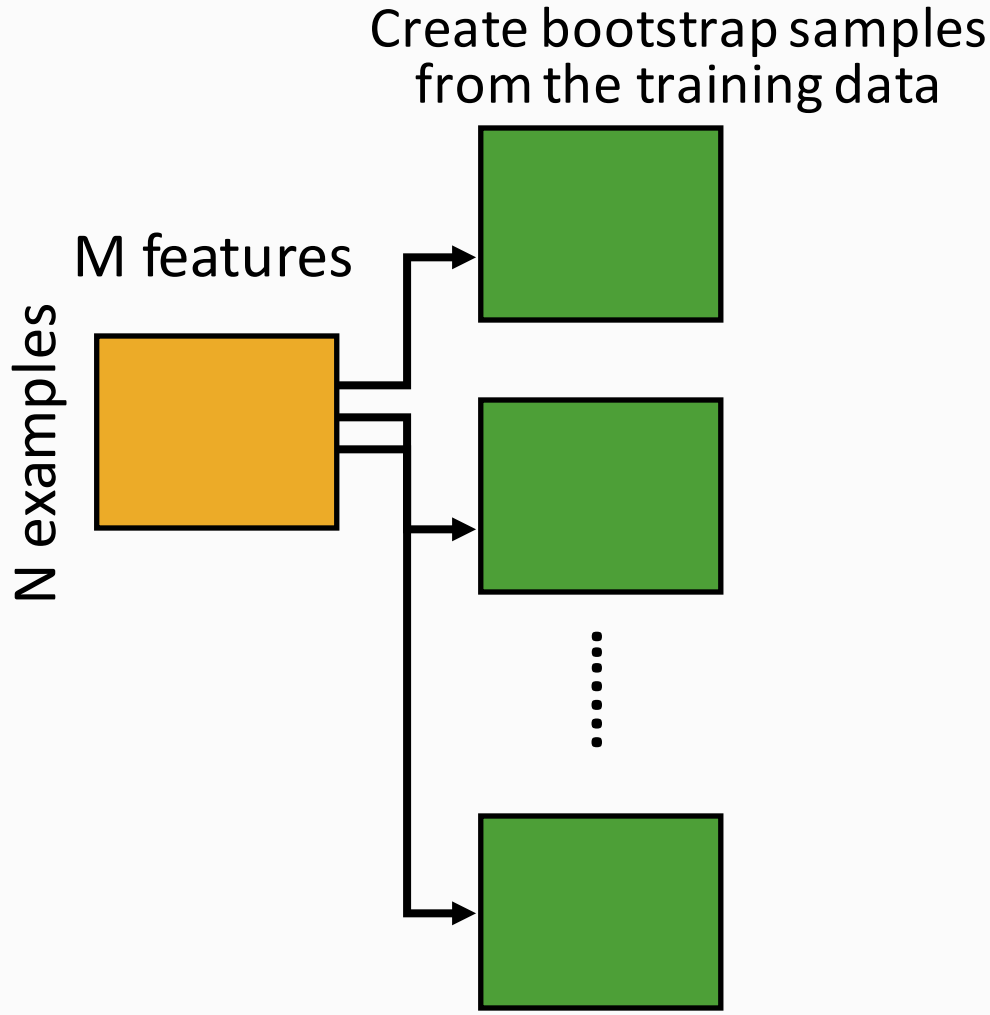
100 bagged trees..

Random Forest Classifier

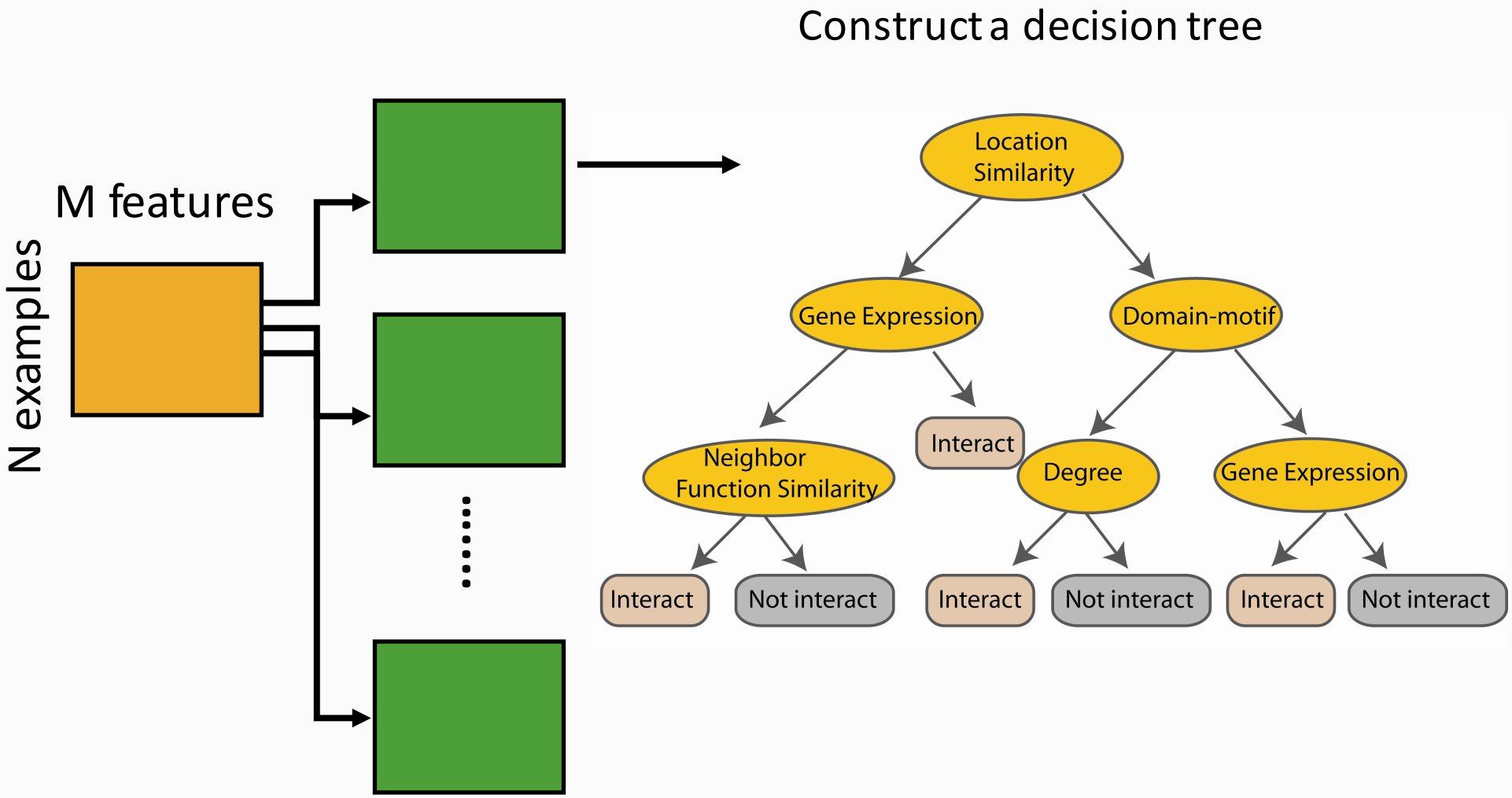
Training Data



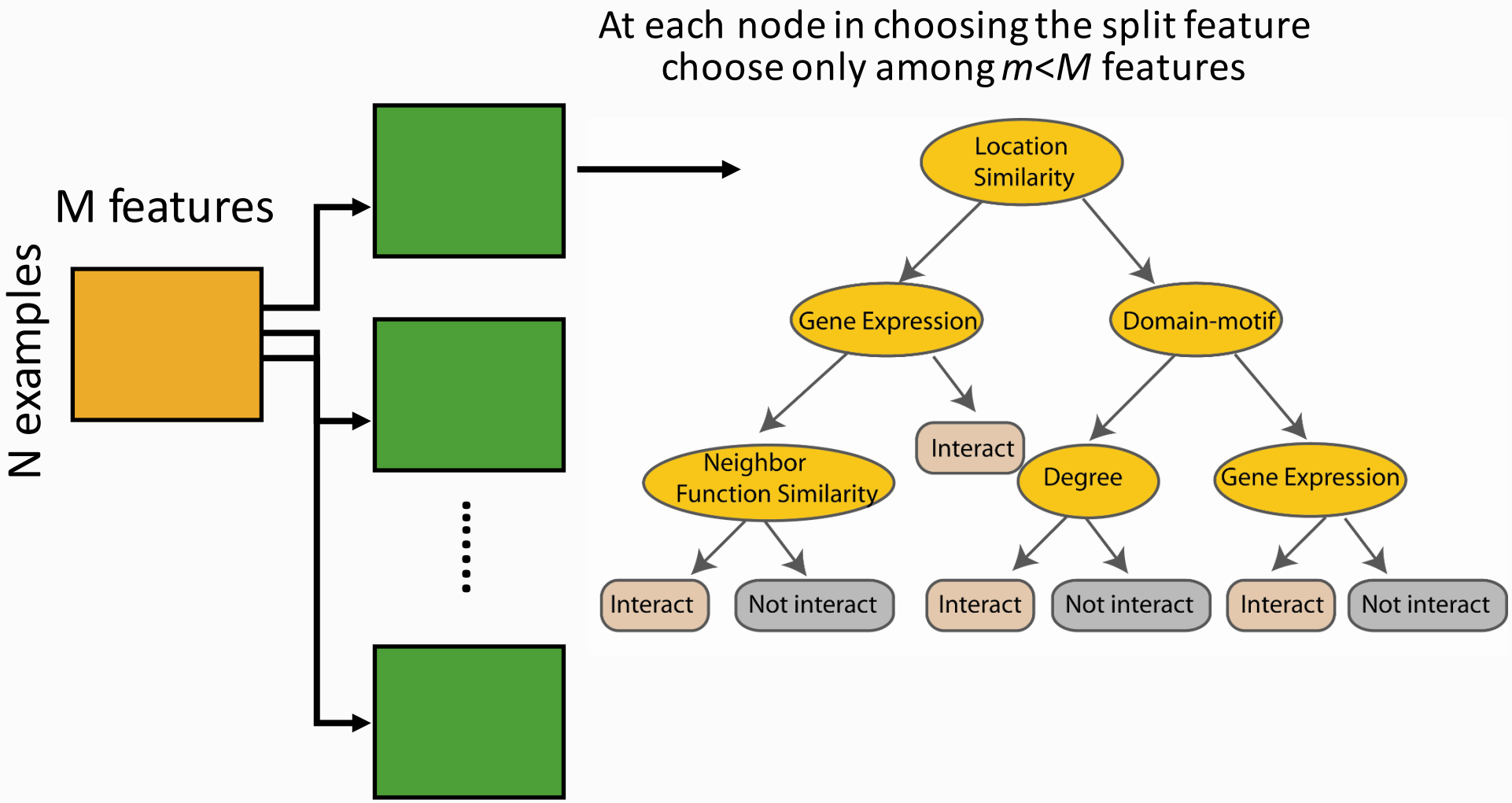
Random Forest Classifier



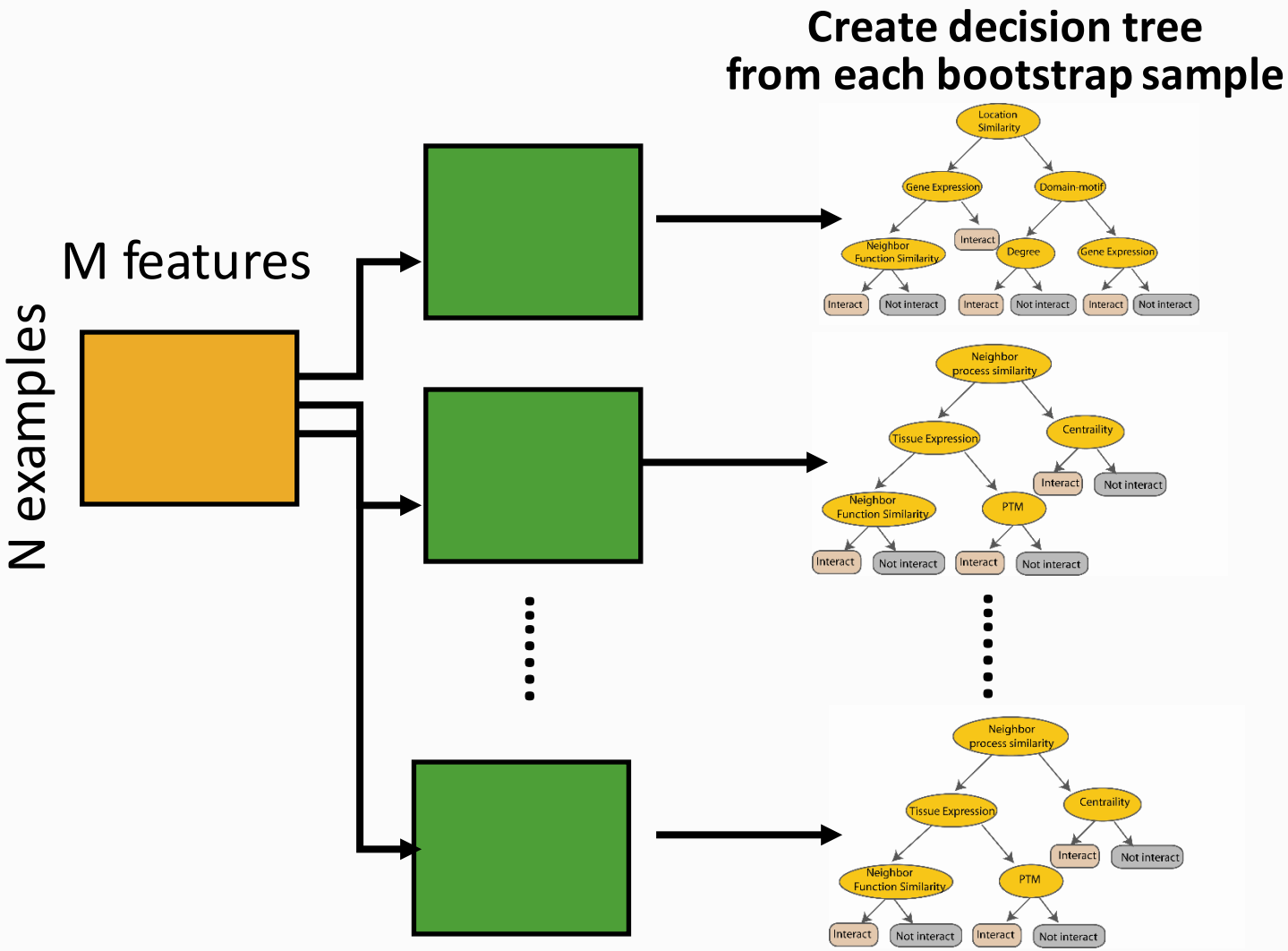
Random Forest Classifier



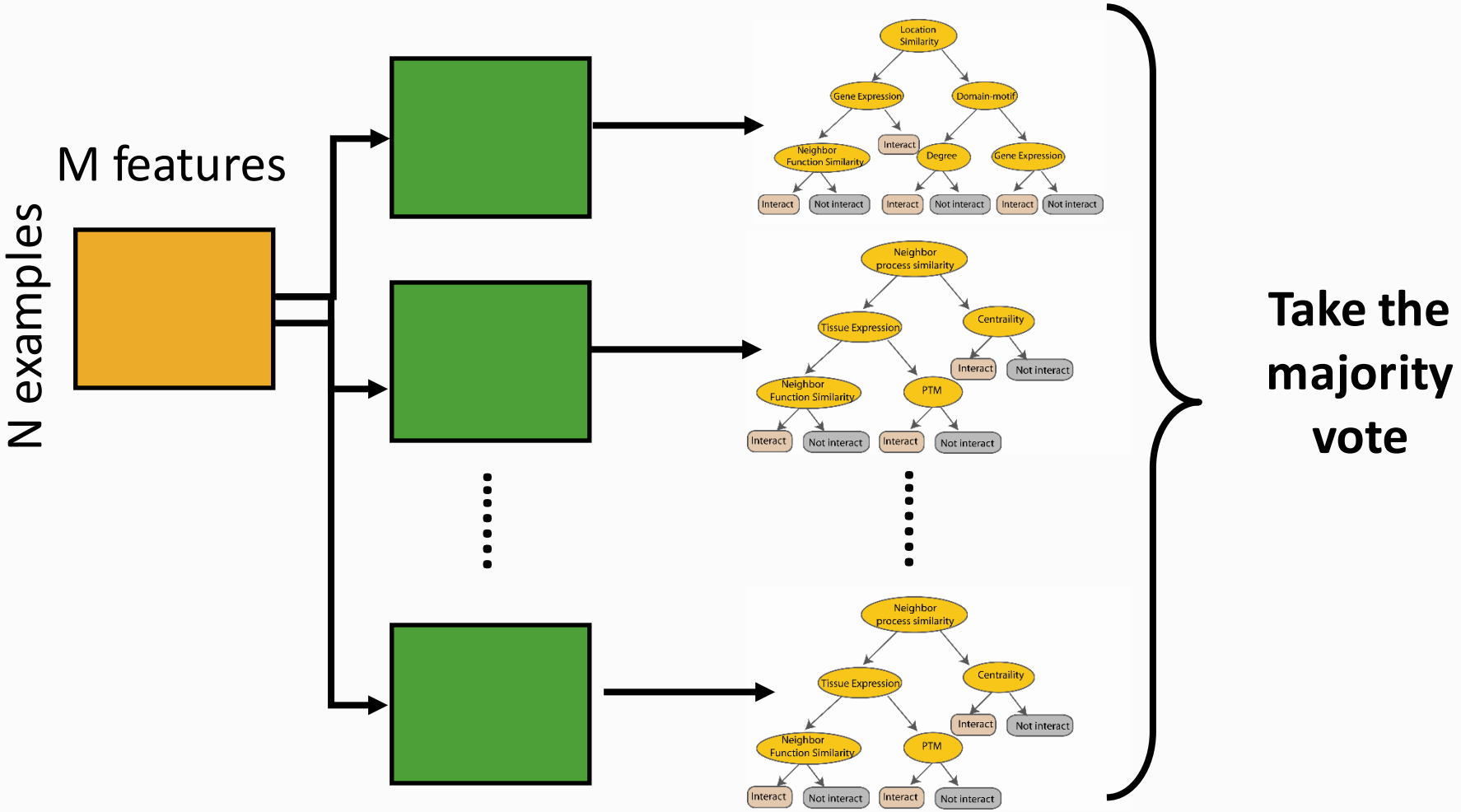
Random Forest Classifier



Random Forest Classifier



Random Forest Classifier



Random Forests

- Combination of decision trees and bootstrapping concepts
- A large number of decision trees is trained, each on a different bootstrap sample
- At each split, only a random number of the original variables is available (i.e. small selection of columns)
- Data points are classified by majority voting of the individual trees

Key Criteria for Ensembling

Consensus

– Each person has access to the same information on which to learn and base their decision.

Independence

– People's opinions are not determined by the opinions of those around them.

Decentralization

– People are able to specialize and draw on local knowledge.

Aggregation

– Some mechanism exists to turn private judgments into a collective decision

Key Criteria for Ensembling

Diversity of Opinion

– Each person should have private information even if it's just an eccentric interpretation of the known facts.

Independence

– People's opinions are not determined by the opinions of those around them.

Decentralization

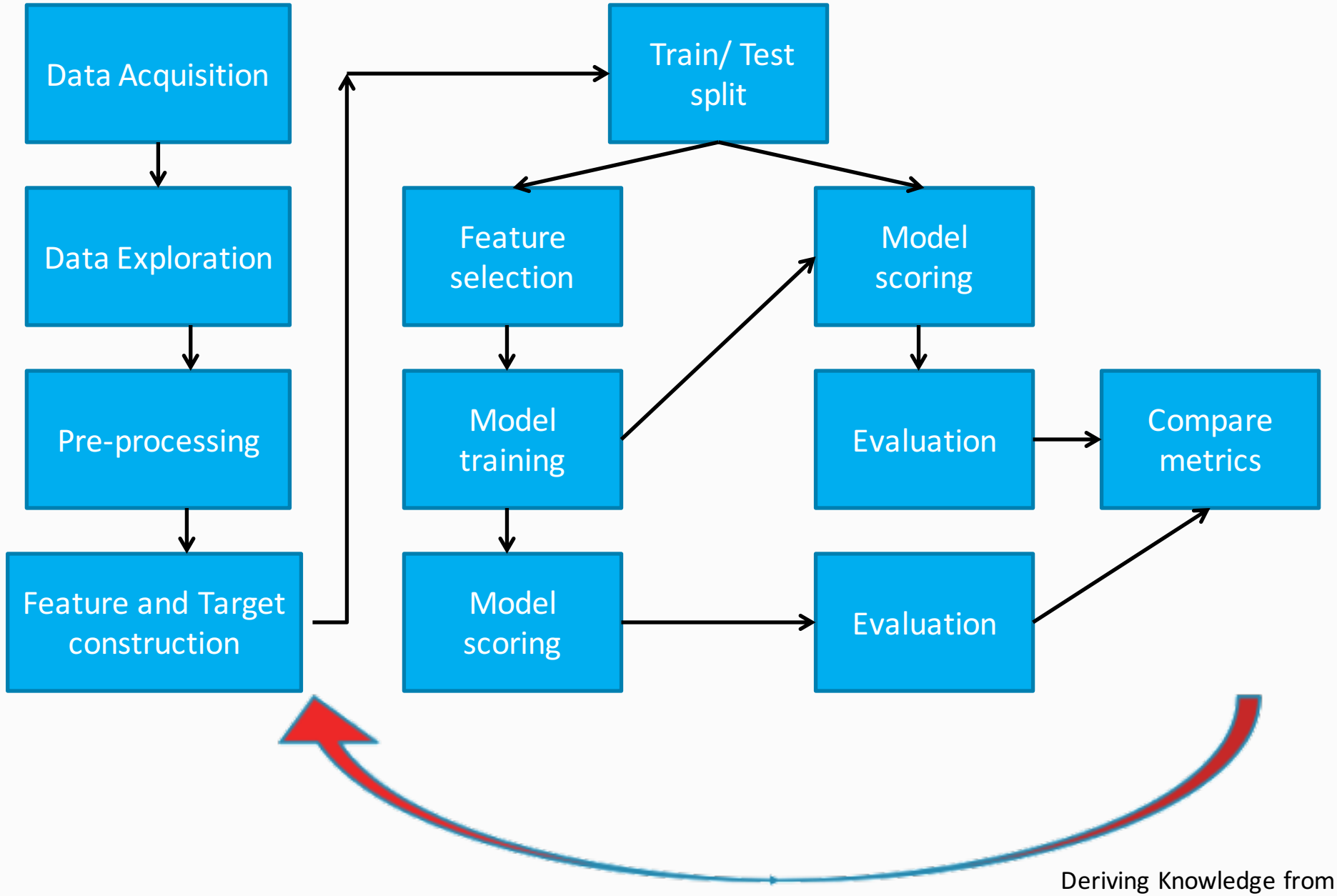
– People are able to specialize and draw on local knowledge.

Aggregation

– Some mechanism exists to turn private judgments into a collective decision

Model Evaluation

Steps in a Predictive Modeling process



Model Scoring (subject for today…)

Process of applying the model parameters to a dataset of features to generate predictions

Some algorithms produce calibrated scores

- i.e. scores represent probability of data point belonging to a class
  e.g. logistic regression, versions of decision trees, Naïve Bayes

Other algorithms produce rank ordering

- i.e. scores can be used for relative ordering of records e.g. SVM

Model Evaluation

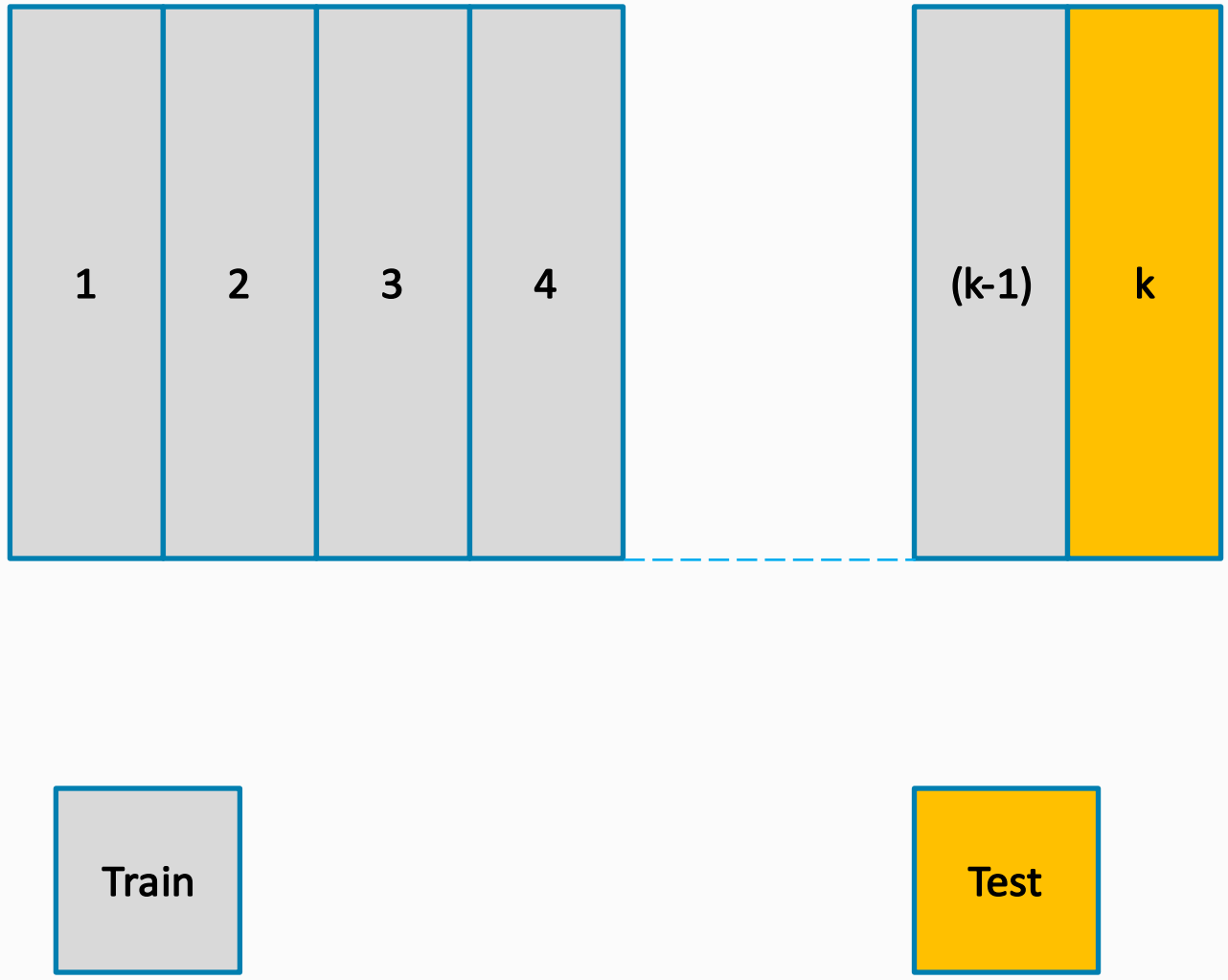
- Compute model performance metrics based on the scores and the true target label

- Metrics measure ability of model to learn the relationship between features and targets

Cross Validation

- Technique for assessing generalization capability of a model, i.e.
  - How well will the model perform on new (unseen) data (drawn from the same distribution as the training data)
- Basic idea is to split the training data into “k” independent pieces (called folds)
  - Train on (k-1) folds and test on the remaining fold
  - Repeat this “k” times, testing once on each fold
  - Average the model and performance metrics from each of these “k” runs
- Typically, k ~ 10

Cross Validation



Performance Metrics

Classification

- Consider a two (2) class classification problem.
- Algorithms predict either a
  - **Class** of the example
    - Decision Trees assign the dominant class of the node where the example falls.
  - **Score** of a class for the example
    - Higher the score, greater is the probability of the data point being positive
    - A better model segregates the classes better
    - E.g. Logistic regression, Naïve Bayes, SVM

Performance Metrics (Example)

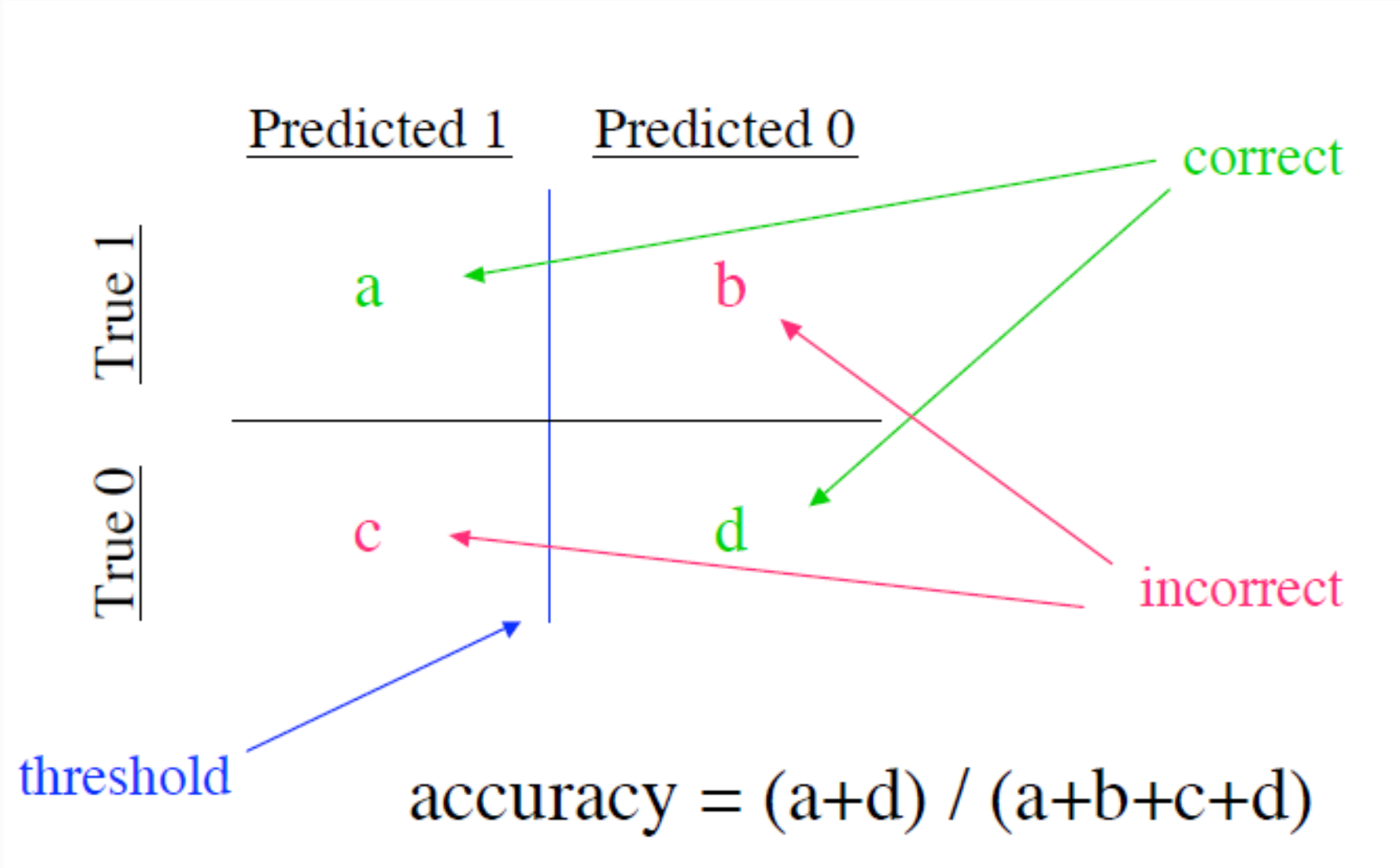
Predicted Label

True Label

Row ID	1	0
1	506	122
0	169	420

Confusion matrix

Performance Metrics



Is Accuracy a good evaluation metric?

Is Accuracy the best evaluation metric?

Accuracy is not the best evaluation metric…

What's wrong with accuracy?

If the vast majority is of binary outcomes are 1, then a stupid model can be accurate but not useful (guess it's always "1"), and a better model might have lower accuracy.

Performance Metrics

Percent Reduction in Error

- 80% accuracy = 20% error
  - *Suppose learning increases accuracy from 80% to 90% error reduced from 20% to 10%*
  - 50% reduction in error

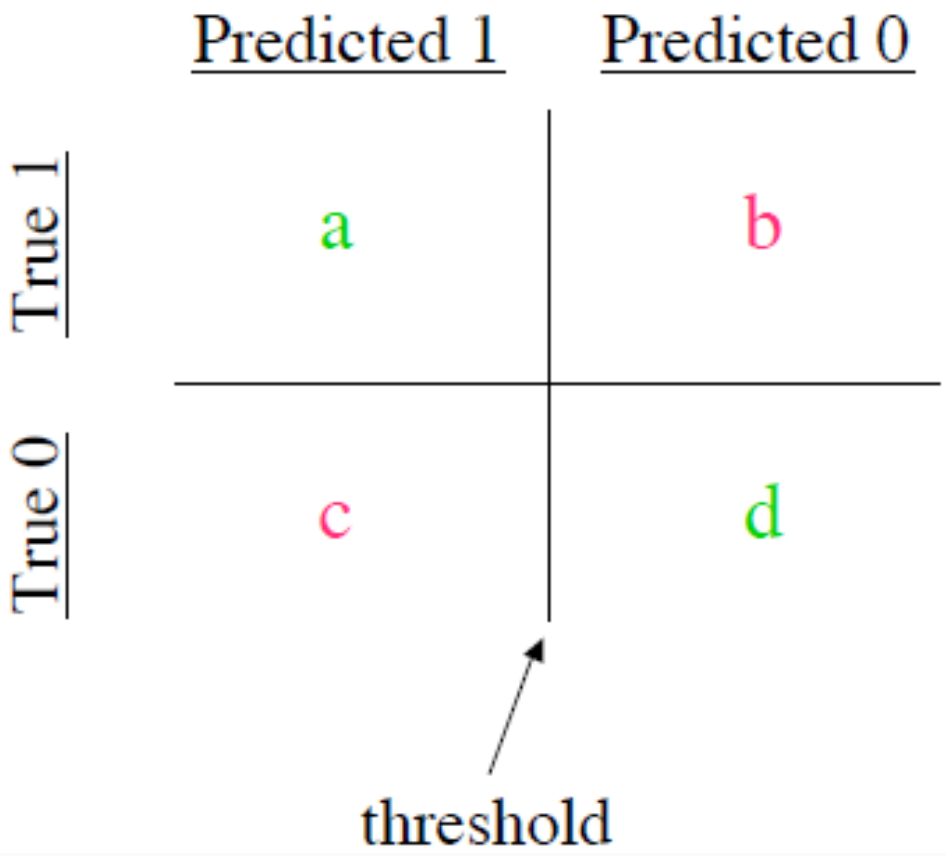
Performance Metrics

Precision and Recall

- Typically used in document retrieval
- Precision:
  - how many of the returned documents are correct
  - precision (threshold)
- Recall:
  - how many of the positives does the model return
  - recall (threshold)

Performance Metrics

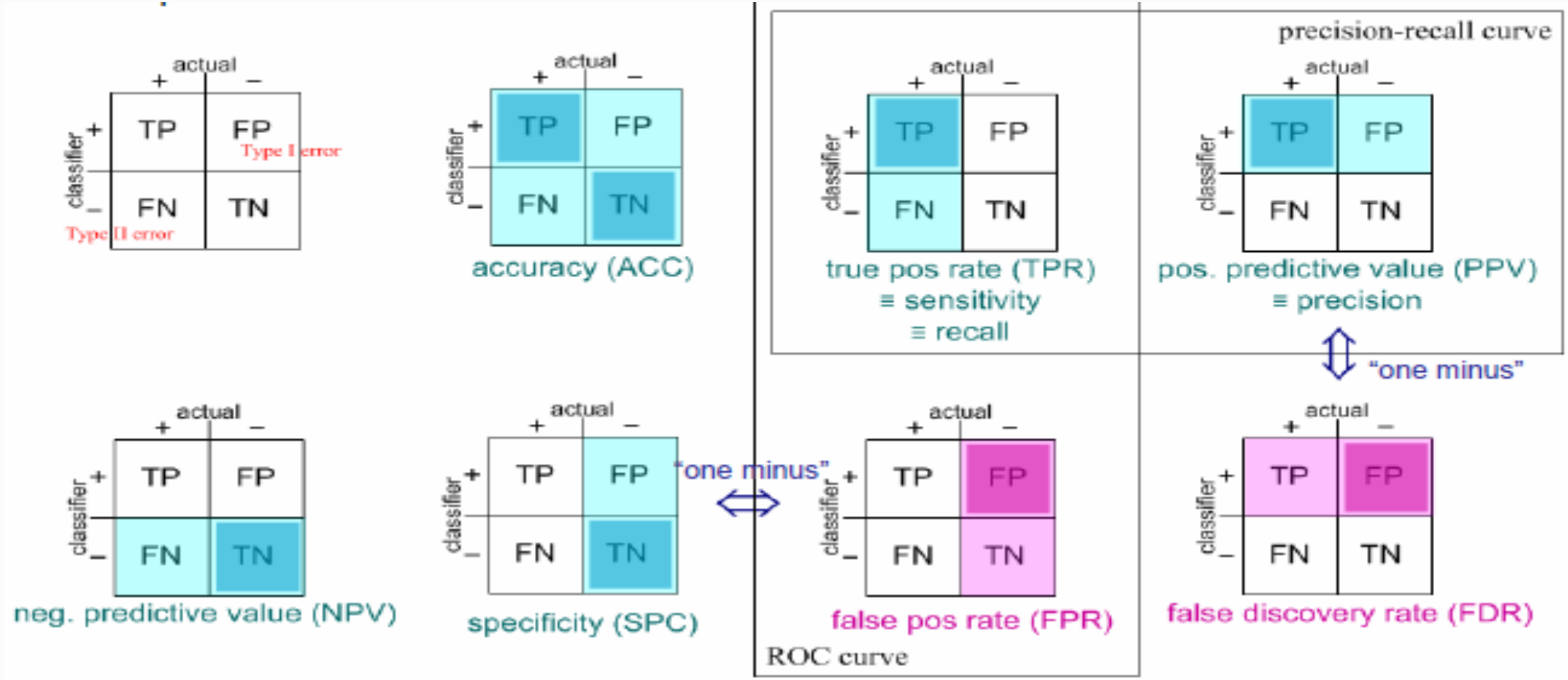
Precision and Recall



$$PRECISION = a/(a+c)$$

$$RECALL = a/(a+b)$$

Performance measures in context…



Dark color is numerator, dark and light color is denominator.

Data Science

Deriving Knowledge from Data at Scale

That's all for tonight….