



Data Science and Business Intelligence

BU.330.780

Session 5

Instructor: Changmi Jung, Ph.D.



Announcement

» Assignment #3 is due next week

- Available on Canvas > Week 6
- Please download the R Markdown and the Excel files (IT-Help-Desk) data in Assignment #3.
- Follow the directions in the markdown file to complete the assignment
- The style is similar to the previous assignments: answer the questions and attach the knitted file – render it to a **Word docx** or **HTML**

» Project status report due next week

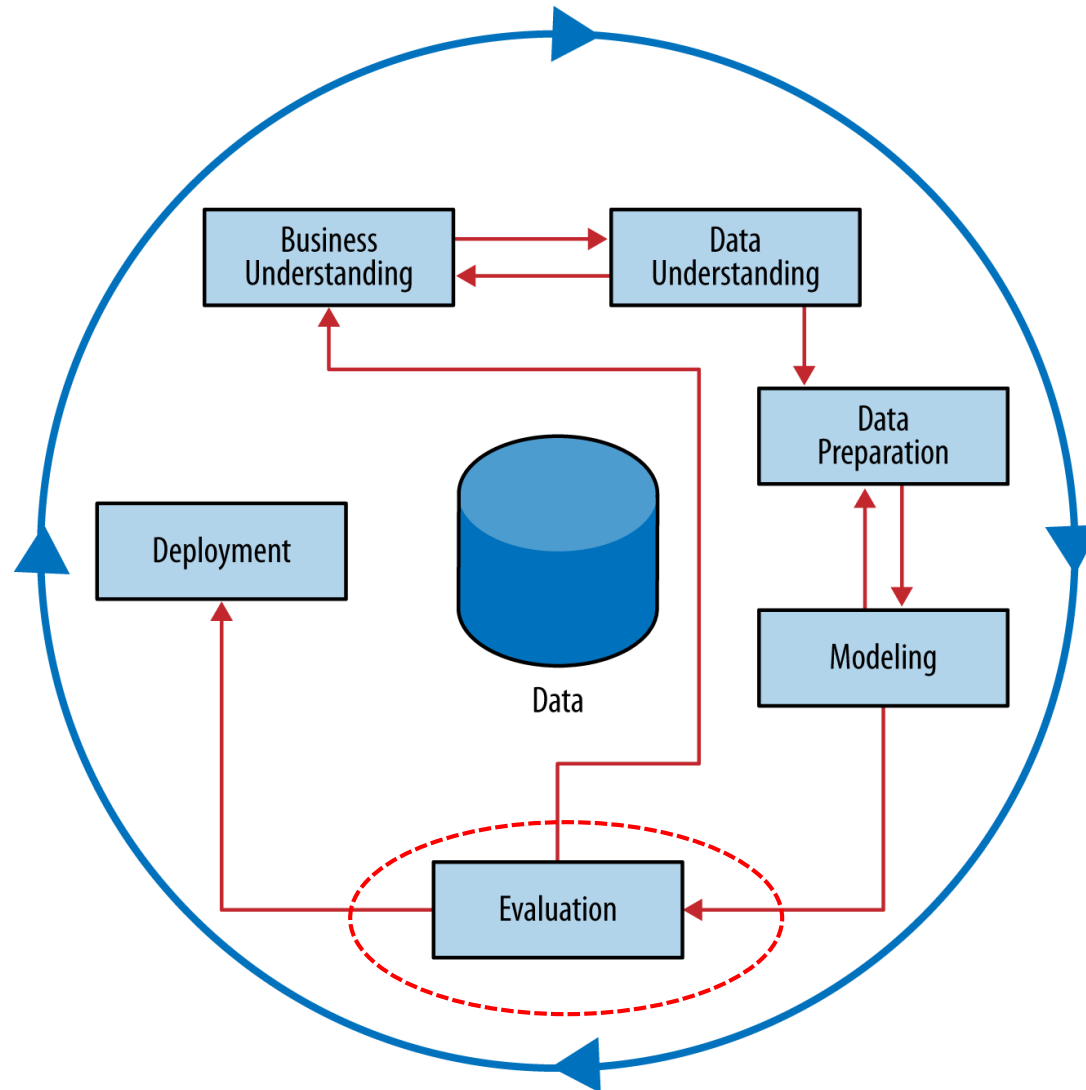
- Submit the file (no more than 3 pages) on Canvas > Week 6
- Details for the content are in Canvas > Project > Team Project Instructions



Next Week: Invited Guest Speaker

- » Tuesday (4/29): We will have our MSIS alumni speaker, Tanush Sharanarathi (MSIS '22), from IBM, joining via Zoom. Prepare your questions for him.

Data science is a process with well-understood stages





Overfitting & Model Validation

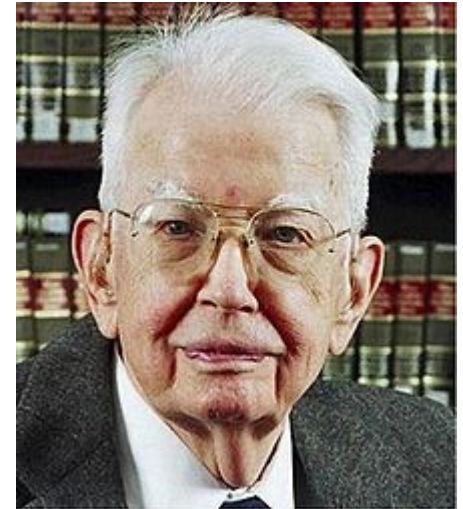


“If you torture the data long enough,
it will confess.”

R.H. Coase "How should economists choose?"

Warren Nutter Lecture, 1981.

Nobel prize winner, Economics 1991

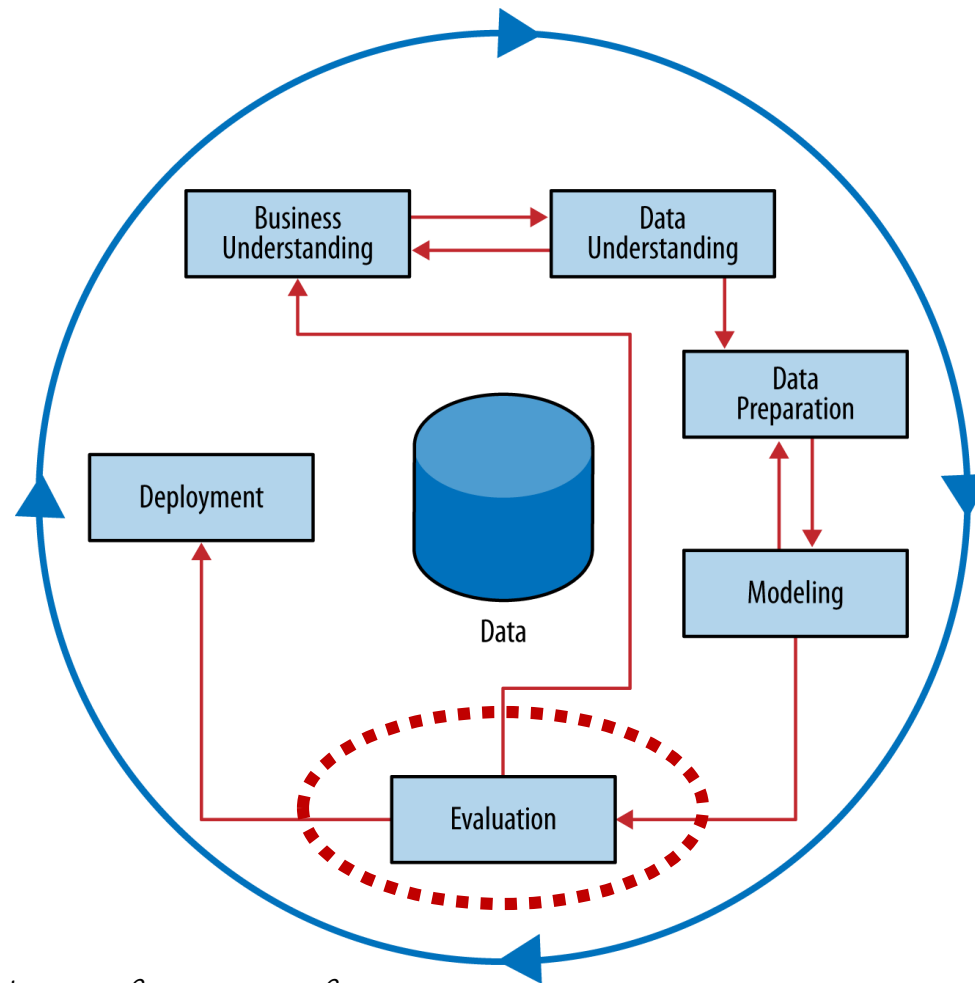




Generalization vs. Overfitting

- » We are interested in patterns that generalize, i.e., that predict well for instances that we have not yet observed.
- » Overfitting: finding chance occurrences in data that **look like** interesting patterns, but which do **not generalize**.
- » An extreme case is building a model that **memorizes** the training data and performs **no generalization**.
- » Generalization: the property of a model or modeling process whereby the model applies to data that **were not used** to build the model.

The Data Science Process



Okay. Then, to evaluate the performance of a model, we need a metric for the performance!



Model accuracy

» Accuracy = $\frac{\text{Number of correct decision made}}{\text{Total number of decisions made}}$ ✓

» Error rate = 1- Accuracy ✓

» Classification accuracy is popular, but usually too simplistic!

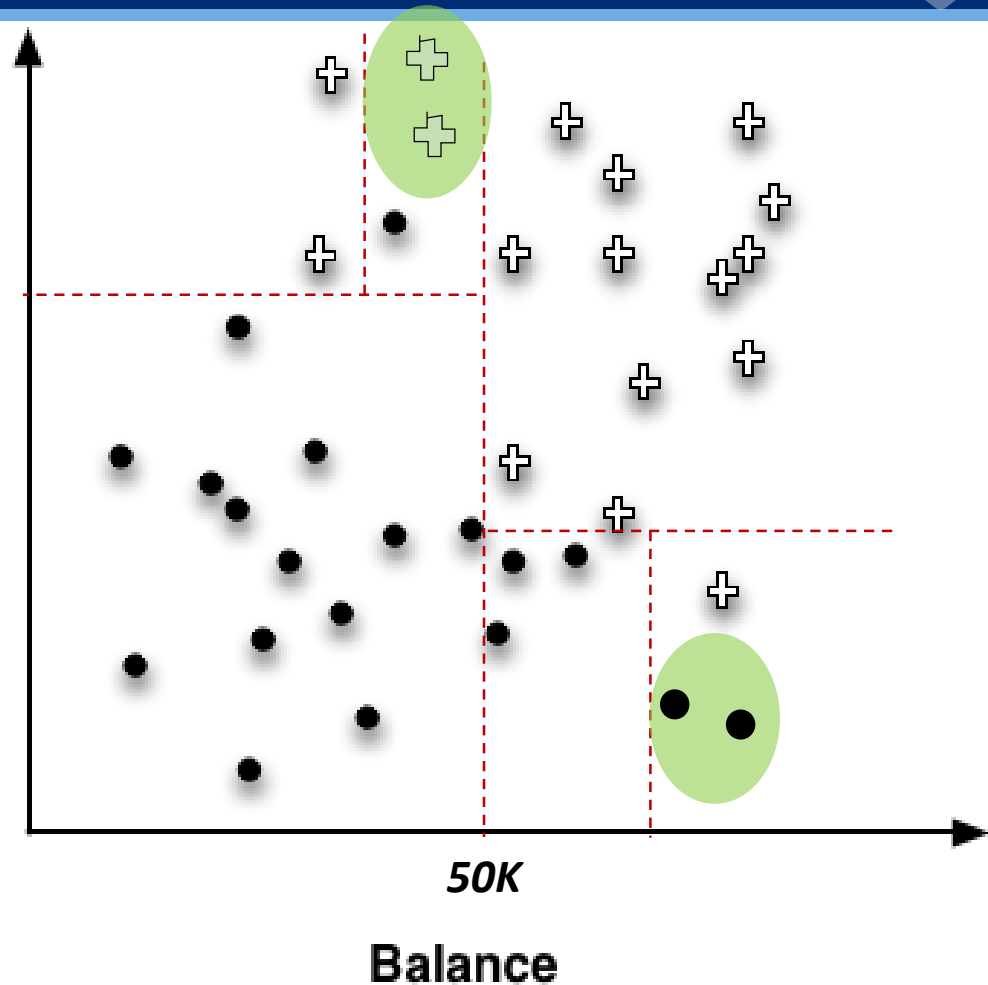
we will talk about the limitation later!



Overfitting in Tree Induction

- » Recall tree induction is finding important, predictive individual attributes recursively and split the data to smaller and smaller data subsets.
- » Eventually, the subsets will be pure – we have found the leaves of our decision tree.
- » The accuracy of this tree will be perfect!
- » What's wrong with this result?

Income



- Not Default – 17 cases
- ⊕ Default – 15 cases



Overfitting in Linear Discriminants

- » There are different ways to allow more or less complexity in mathematical functions.

- $f(x) = w_0 + w_1x_1 + w_2x_2$

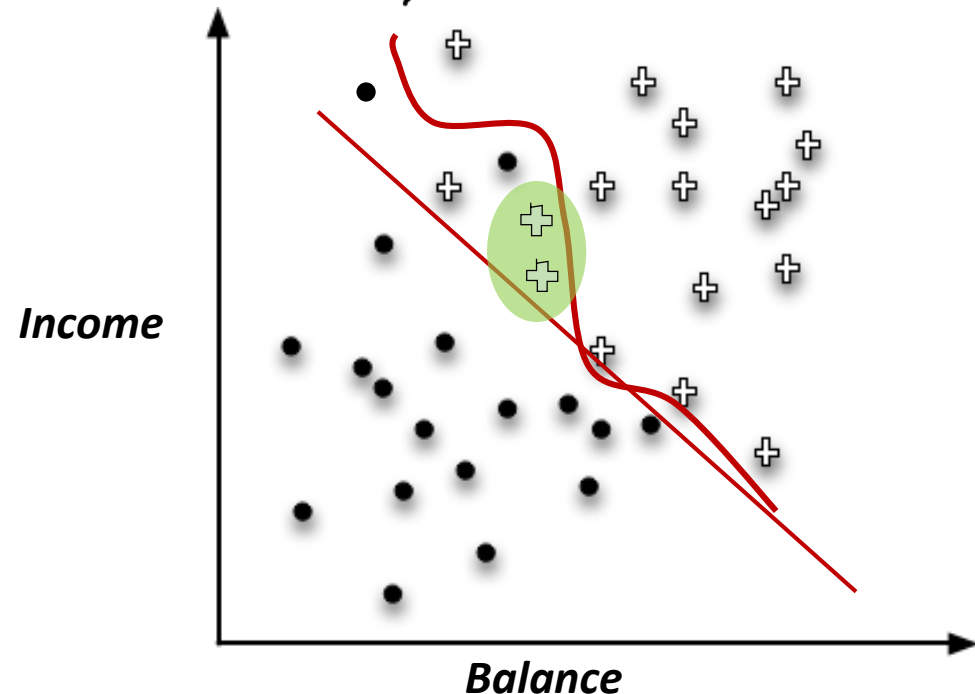
- e.g.) $f(x) = w_0 + w_{Balance} \times BALANCE + w_{INCOME} \times INCOME$

- » Add more variables: $f(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$

- » Add attributes that are non-linear, i.e. x_1^2 or $\frac{x_2}{x_3}$.

- » As you increase the dimensionality, you can perfectly fit larger and larger sets of arbitrary points.

- » Is it a good approach?





Why does performance degrade? (1/5)

» Why does **overfitting** cause a model to become worse? —

- As a model gets more complex, it is allowed to pick up harmful “spurious” correlations, which may produce **incorrect generalizations**.
- These correlations do not represent characteristics of the population in general..

Now let's take a look at an example case of overfitting



Why does performance degrade? (2/5)

Class	r1
leave	A
leave	B
leave	A
leave	B
stay	A
stay	B
stay	B
stay	A

*This table has one predicting attribute (r1) with its value assigned **randomly**. Unless we're very unlucky, we shouldn't see a relationship between the random value and the class (target variable).*




Now, let's add more random attributes!

Class	r1	r2	r3
leave	A	B	B
leave	B	A	A
leave	A	A	A
leave	B	A	B
stay	A	B	A
stay	B	B	A
stay	B	B	B
stay	A	A	A

With more random attributes, we can start to mine "useful" patterns:

"if r2 = A then leave

***else stay"** has accuracy = 75% ! *

*We found a pattern that doesn't exist!
(on these data!) *



Why does performance degrade? (3/5)

Let's keep on adding random variables.... Now with 9 random variables...

Class	r1	r2	r3	r4	r5	r6	r7	r8	r9
leave	A	B	B	A	B	A	B	A	B
leave	B	A	A	B	B	B	B	B	B
leave	A	A	A	A	A	A	B	B	B
leave	B	A	B	A	A	A	A	B	B
stay	A	B	A	A	A	A	A	B	A
stay	B	B	A	A	B	B	B	B	B
stay	B	B	B	A	B	A	B	A	A
stay	A	A	A	B	B	B	A	B	A

If $r9 = B$

if $r2 = B$

if $r1 = A$ then leave

if $r1 = B$ then stay

if $r2 = A$ then leave

If $r9 = A$ then stay

$r9$ just misses 1

Correctly Classified Instances

8

100 %



Why does performance degrade? (4/5)

More random variables? *Better-looking* trees...

Class	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10	r11	r12	r13	r14
leave	A	B	B	A	B	A	B	A	B	B	B	A	B	A
leave	B	A	A	B	B	B	B	B	B	A	A	A	A	B
leave	A	A	A	A	A	A	B	B	B	B	B	A	A	A
leave	B	A	B	A	A	A	A	B	B	A	B	A	B	B
stay	A	B	A	A	A	A	A	B	A	A	A	B	B	B
stay	B	B	A	A	B	B	B	B	B	A	B	B	B	B
stay	B	B	B	A	B	A	B	A	A	A	B	B	B	B
stay	A	A	A	B	B	B	A	B	A	A	A	A	B	A

If $r_9 = B$

if $r_{12} = A$ then leave

if $r_{12} = B$ then stay

If $r_9 = A$ then stay

→ 100% accuracy



Why does performance degrade? (5/5)

Even good *“true”* patterns will be obscured eventually

Class	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10	r11	r12	r13	r14	t1	t2
leave	A	B	B	A	B	A	B	A	B	B	B	A	B	A	A	A
leave	B	A	A	B	B	B	B	B	B	A	A	A	A	B	A	A
leave	A	A	A	A	A	A	B	B	B	B	B	A	A	A	B	B
leave	B	A	B	A	A	A	A	B	B	A	B	A	B	B	A	A
stay	A	B	A	A	A	A	A	B	A	A	A	B	B	B	A	B
stay	B	B	A	A	B	B	B	B	B	A	B	B	B	B	B	A
stay	B	B	B	A	B	A	B	A	A	A	B	B	B	B	A	B
stay	A	A	A	B	B	B	A	B	A	A	A	A	B	A	B	A

If $r9 = B$

if $r12 = A$ then leave

if $r12 = B$ then stay

If $r9 = A$ then stay

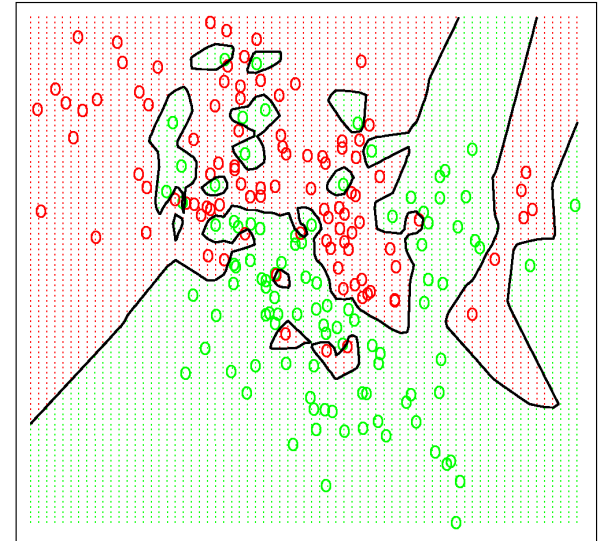
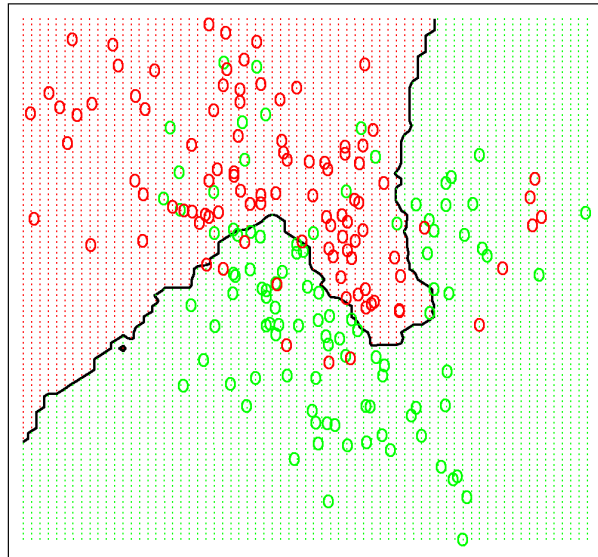
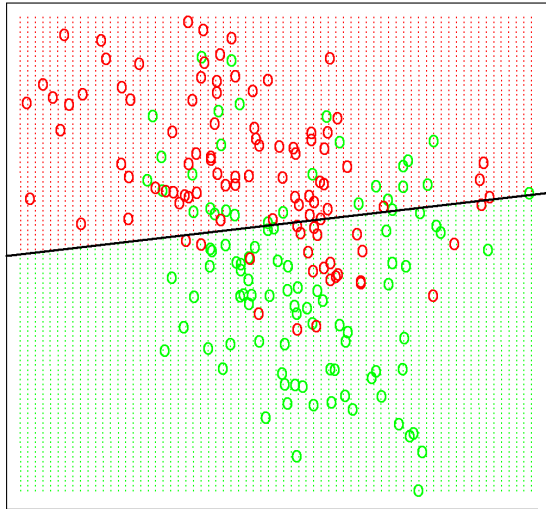
→ 100% accuracy

Together, $t1$ and $t2$ predict the class perfectly

:if $t1 = t2$ then “leave”

But they’re not found by the tree inducer
(here, because neither t variable
in isolation reduces entropy
as much as $r9$ does accidentally)

How can we judge whether our modeling has overfit?





Two important types of model evaluation

» Holdout validation

- e.g., temporal split(s), cross-validation

» Domain knowledge validation

- Sanity checking by modelers
- Using the model as an interface between modelers and stakeholders
 - important to have a model that is comprehensible to stakeholders
 - can get an expert assessment of the model (if experts are available)
- Validate that data (training/testing) and use contexts are acceptably similar



Holdout validation

- » Evaluating the model using training data does not assess how well the model generalizes to unseen cases.
 - We are interested in generalization– the model performance on data not used for training
- » Given only one data set, we hold out some data for evaluation
- » Idea: “**hold out**” some data for which we know the value of the target variable but will not be used to build the model → “lab test”
- » **Predict** the values of the “holdout data” (a.k.a. “test data”) with the model and **compare** them with the true values

Let's move to the week5 R exercise





Cross-validation

- » Cross-validation is a more sophisticated training and testing procedure.
 - Not only a simple estimate of the generalization performance but also some statistics on the estimated performance (mean, variance,..)
 - How does the model performance vary across data sets?
 - Assessing confidence in the performance estimate

- » Cross-validation computes its estimates over all the data by performing multiple splits and systematically swapping out samples for testing → enabling more effective utilization of a given dataset

Alright, then how does it work?

Cross-validation (CV)

» Split a data set into k partitions (**folds**) ($k = 5$ or 10).

» Iterate training and testing k times.

» In each iteration, a different fold is chosen as the test data. The other $k-1$ folds are combined to form the training data.



Original dataset

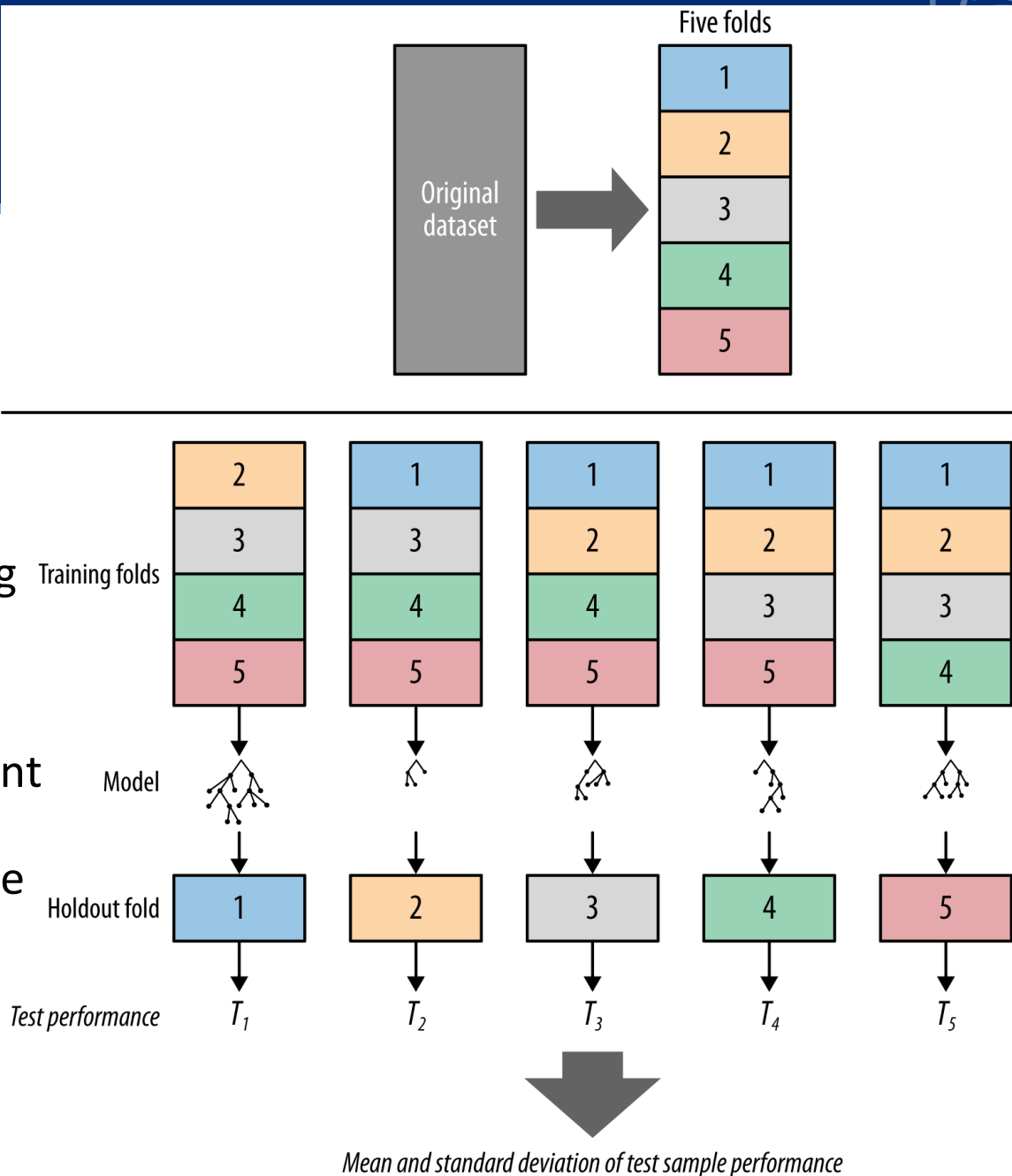
A vertical gray rectangle with a black border, containing the text 'Original dataset' in white. A horizontal line is positioned below the rectangle.

Cross-validation (CV)

» Split a data set into k partitions (**folds**) ($k = 5$ or 10).

» Iterate training and testing k times.

» In each iteration, a different fold is chosen as the test data. The other $k-1$ folds are combined to form the training data.

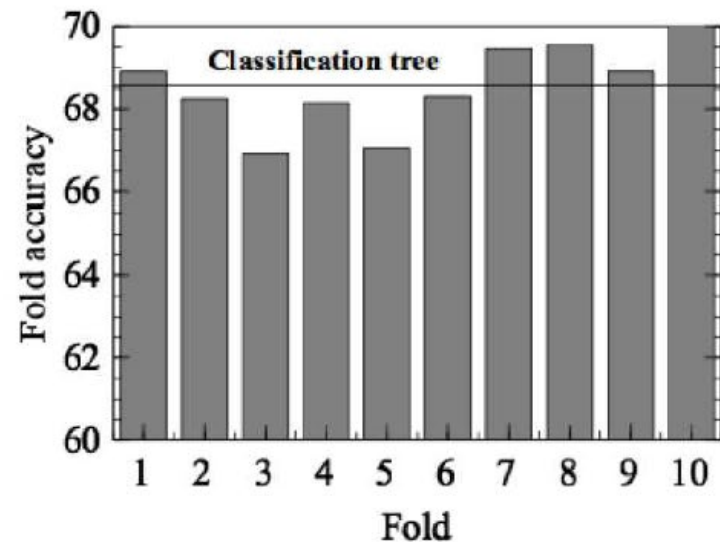
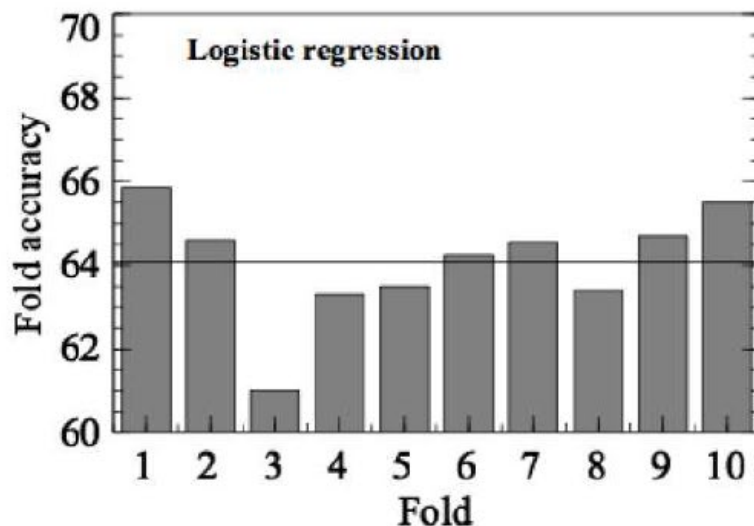




Cross-validation example

» Cross-validation: the dataset was first shuffled, then divided into ten partitions → 10 fold cross validation

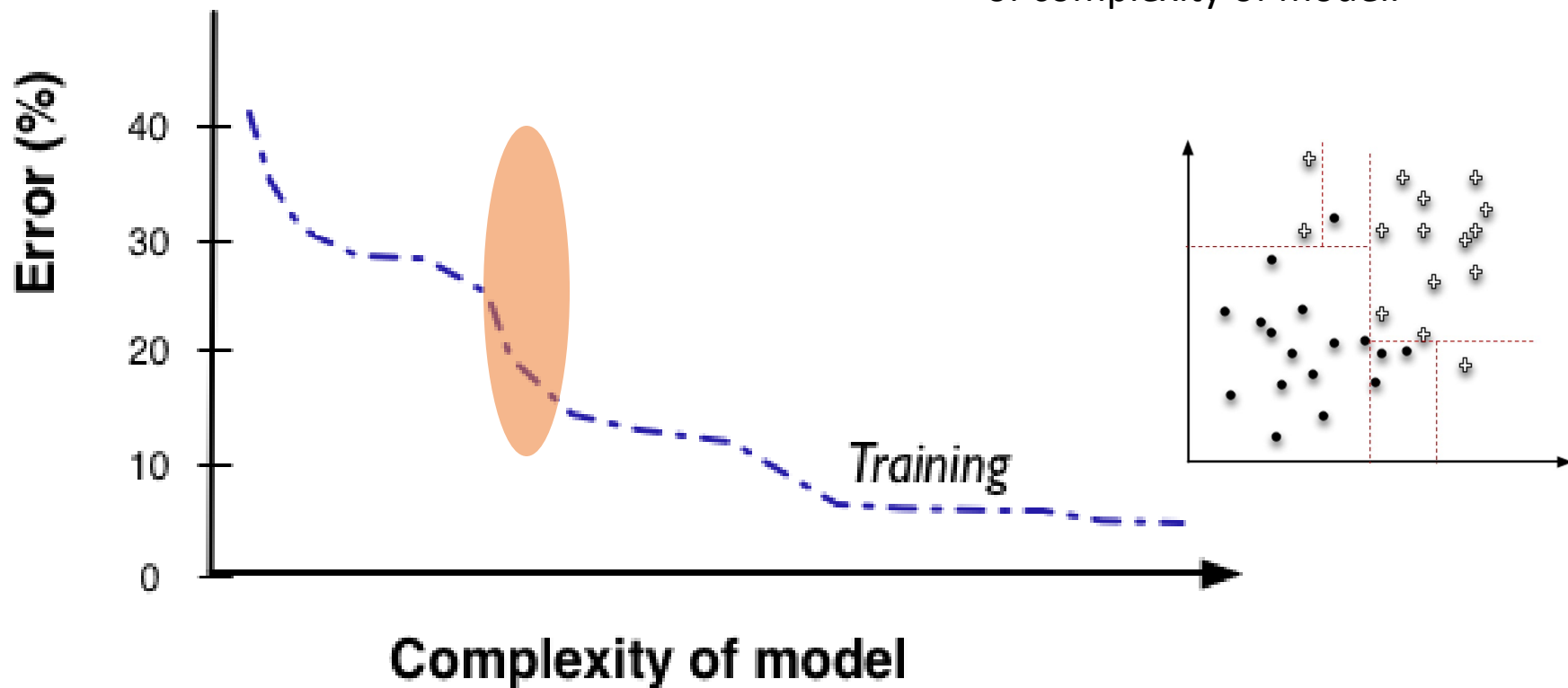
- Classification trees: avg. accuracy is 68.6% (std 1.1)
- Logistic regression models: avg. accuracy is 64.1% (std 1.3)



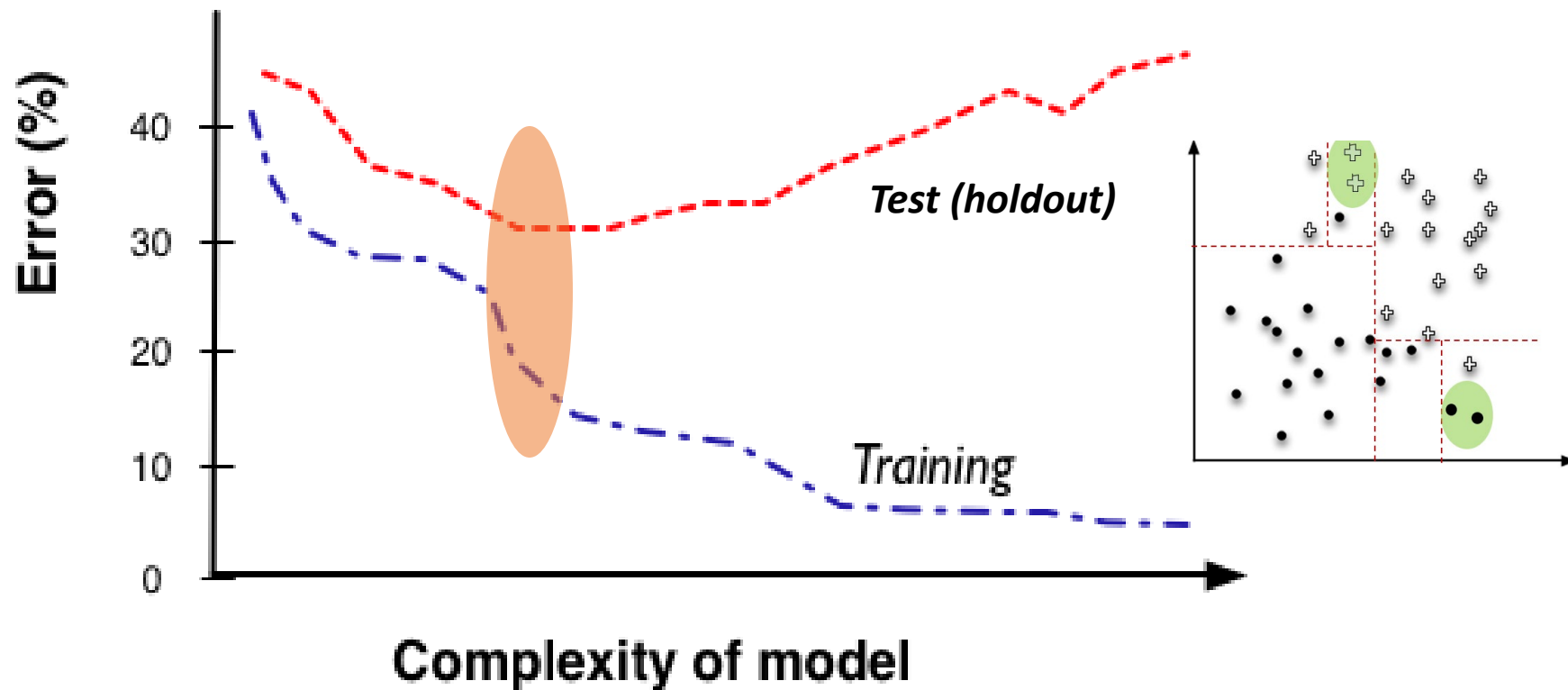
Fitting Curves: Tool for model performance analytics



Fitting curve displays model performance as a function of complexity of model.



Fitting Curves: Tool for model performance analytics

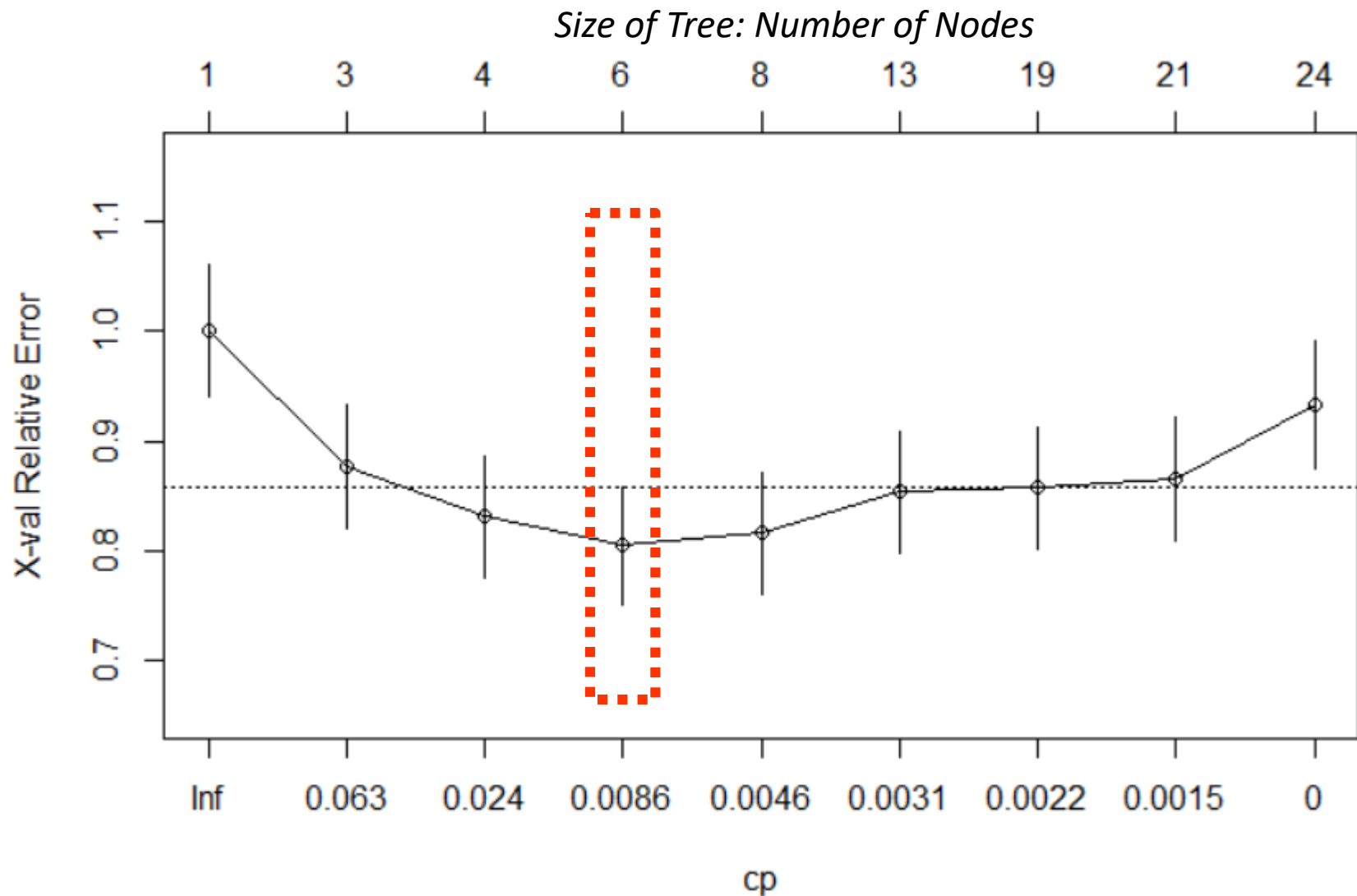


Let's practice with R





Fitting Curve – Credit Card Default Case





Recap: OOB Error from Random Forest

Original Dataset

Age	Student	Income	Balance	Default
40	No	44K	3K	No
32	Yes	12K	1.5K	No
27	No	31K	1K	Yes
51	No	35K	0.5K	No
35	No	63K	2K	No

Bootstrapped Dataset

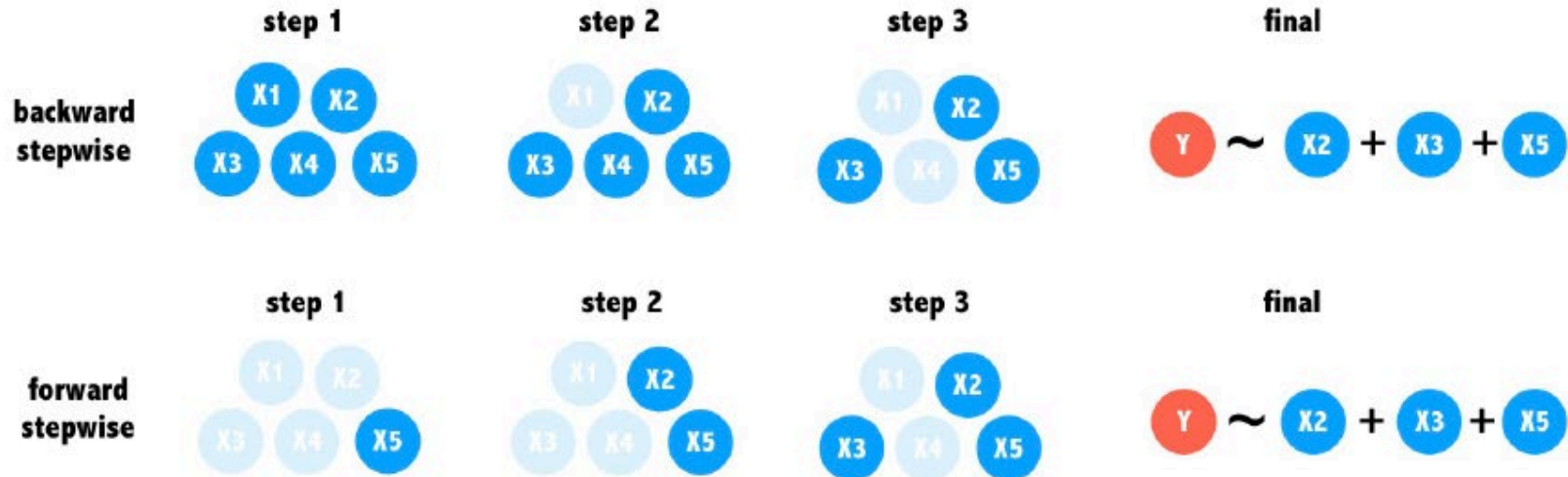
Age	Student	Income	Balance	Default
40	No	44K	3K	No
40	No	44K	3K	No
27	No	31K	1K	Yes
51	No	35K	0.5K	No
35	No	63K	2K	No

32	Yes	12K	1.5K	No
----	-----	-----	------	----

Out-Of-Bag Dataset

- » OOB (Out-Of-Bag) datasets are great test data
- » OOB Error: Performance of Random Forest is assessed by the proportion of OOB samples that are classified incorrectly by the model

Automatic feature selection – Stepwise regression



Note: We don't always get the same results from the two approaches.



Generalization Performance

- » Different modeling procedures may have different performance on the same data
- » Different training sets may result in different generalization performance
- » Different test sets may result in different estimates of the generalization performance
- » If the training set size changes, you may also expect different generalization performance from the resultant model



Model Performance Evaluation – Confusion Matrix



What is a good model? (1/2)

- » Accuracy = $\frac{\text{Number of correct decision made}}{\text{Total number of decisions made}}$
- » Classification accuracy is popular, but usually too simplistic!
- » What is a potential problem with this measure? What are we missing?



What is a good model? (2/2)

- Measuring accuracy and its problems

» Classification terminology: Bad positive and harmless negatives

- A **bad** outcome → a “positive” example [alarm!]
- A **good** outcome → a “negative” example [uninteresting, normal]

» Further Examples





- Medical test
 - positive example → disease is present
- Fraud detector
 - positive example → unusual activity on account

» A classifier tries to distinguish the majority of cases (**negatives**, the uninteresting) from the small number of alarming cases (**positives**, alarming).



Confusion matrix

- » A confusion matrix for a problem involving 2 classes

		<i>Actual classes</i>	
		positive (+)	negative (-)
<i>Predicted classes</i>	Y (+)	<i>True positives</i> 	<i>False positive</i> 
	N (-)	<i>False negative</i> 	<i>True negative</i> 

- » Each example in a test set has an **actual class** label and a **predicted class** by the classifier!

Confusion matrix represents those different types of predictions in the matrix format and demonstrates how one class is being confused for another.



Building a Confusion Matrix

Let's regard: '0' as negative (No Default)
and '1' as positive (Default)

Actual Default	Model Prediction
0	0
1	1
0	1
0	1
0	0
1	1
0	0
0	0
1	1
1	0



Actual Predicted	Default	No Default	Total
Default	3	2	5
No Default	1	4	5
Total	4	6	10



False positive and False negative

There could be a big difference in the associated cost and benefit for different types of errors.

Type I error
(false positive)

Type II error
(false negative)

Type I Error

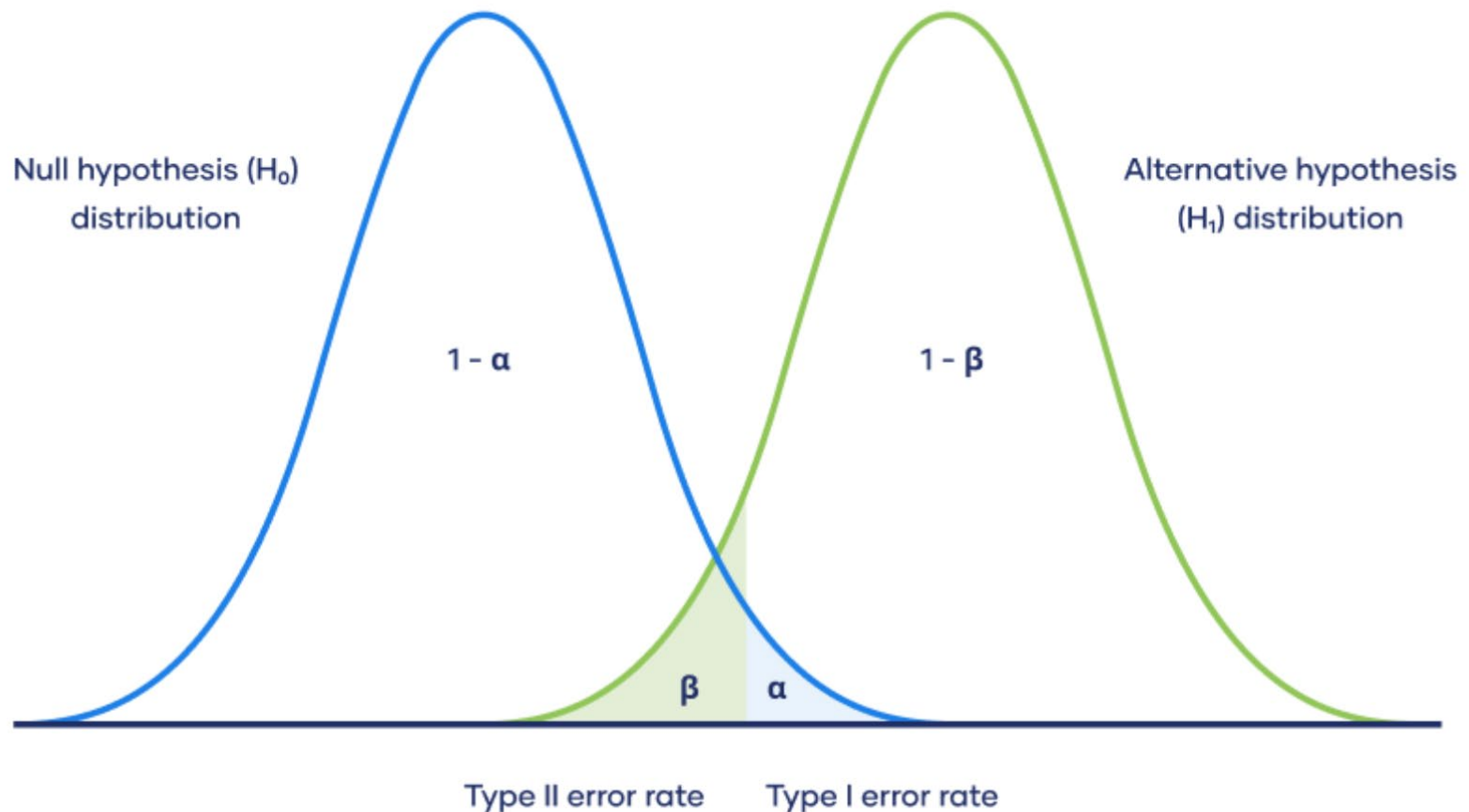


Type II Error





Probability of making Type I and Type II errors





Unequal costs and benefits

» How much do we care about the different errors and correct decisions?

- Classification accuracy makes no distinction between false positive and false negative errors
- In real-world applications, different kinds of errors lead to different consequences!

» Errors should be counted separately

- Estimate the cost or benefit of each decision for the business

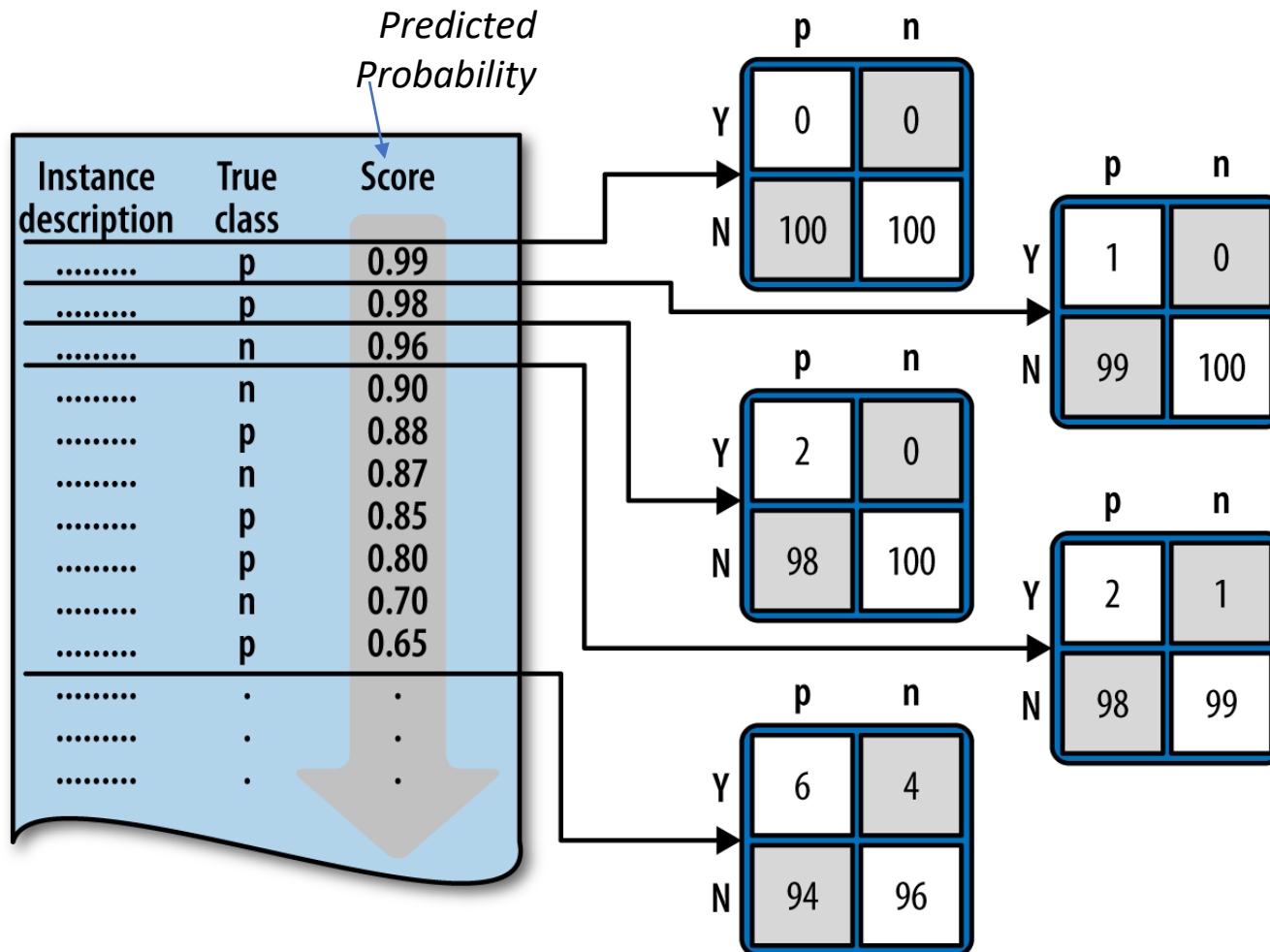


Visualizing Classifier Performance



Recall: Different Threshold Results Different Confusion Matrix!

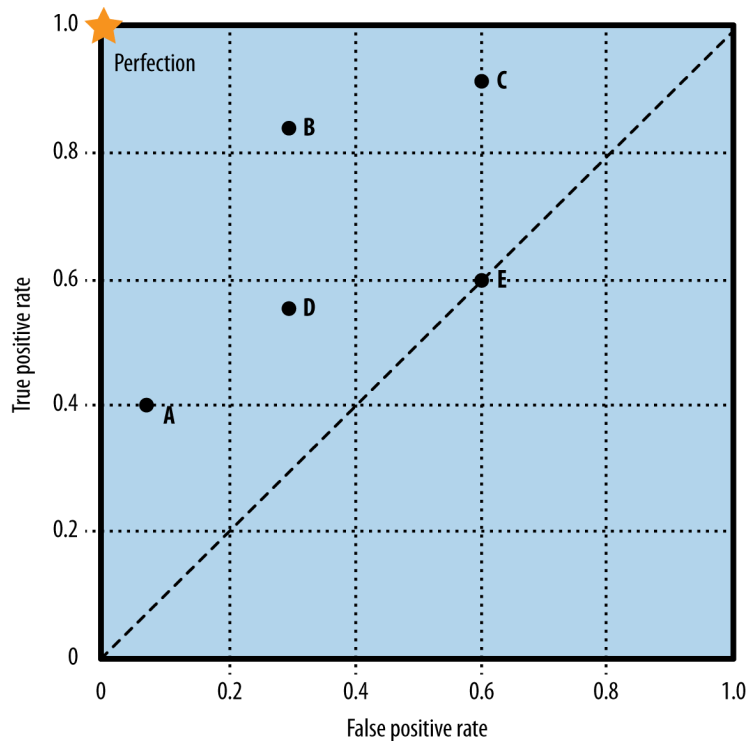
» Consider the case that consists of **100 positive** and **100 negative** cases.





ROC Space

- » “Receiver Operating Characteristic” analysis (introduced in 1950s)
- » Each confusion matrix is represented by a point - plotting its (TPR,FPR) pair

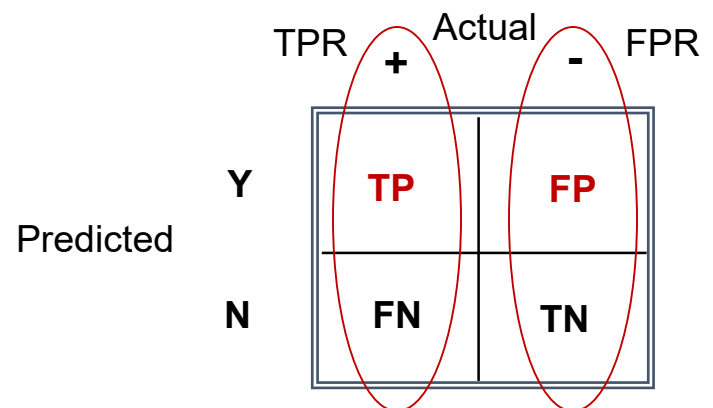


- True Positive Rate (a.k.a sensitivity) =

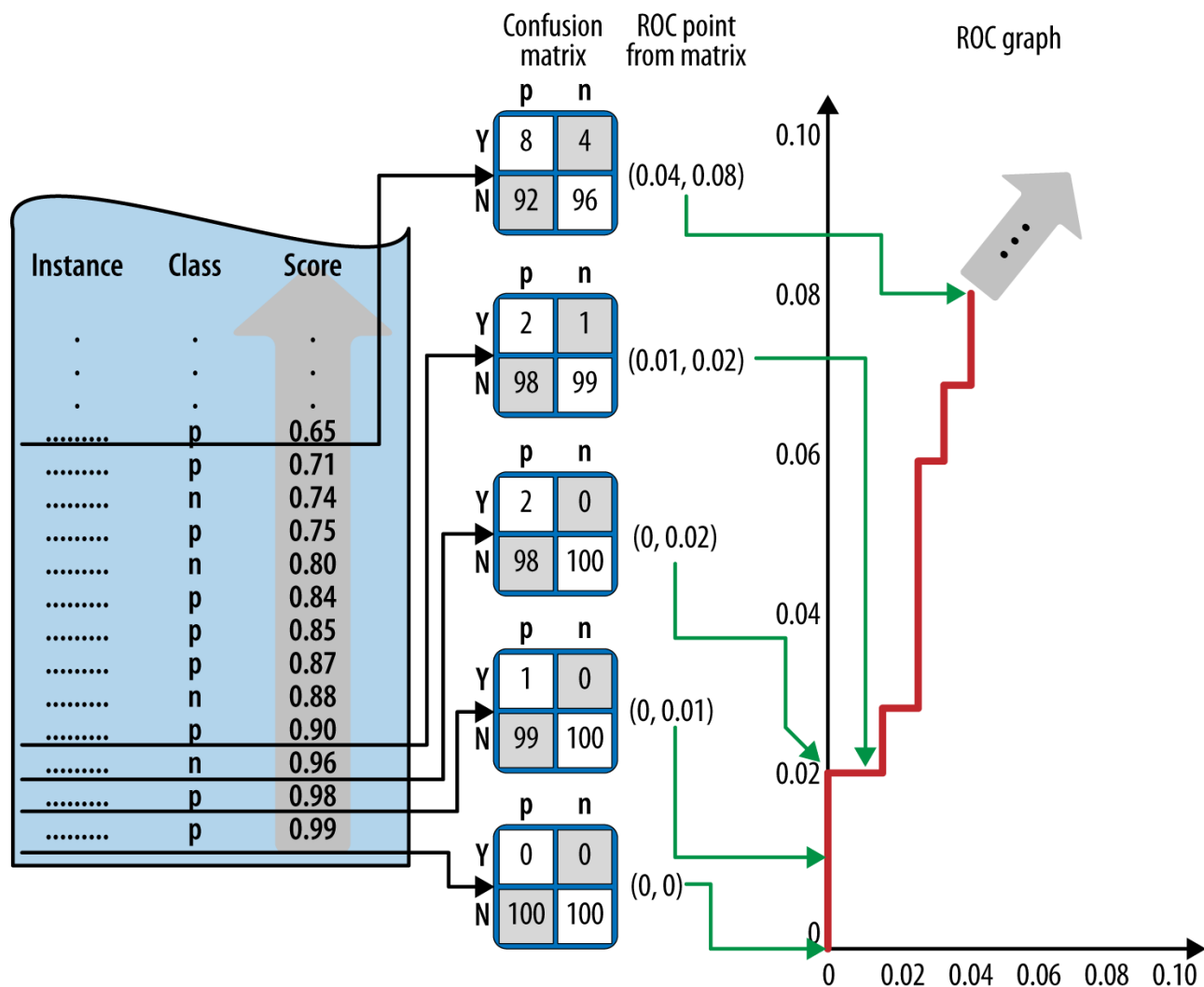
$$\frac{\text{True positive}}{\text{True Positive} + \text{False Negative}}$$

- False Positive Rate (a.k.a. 1- specificity) =

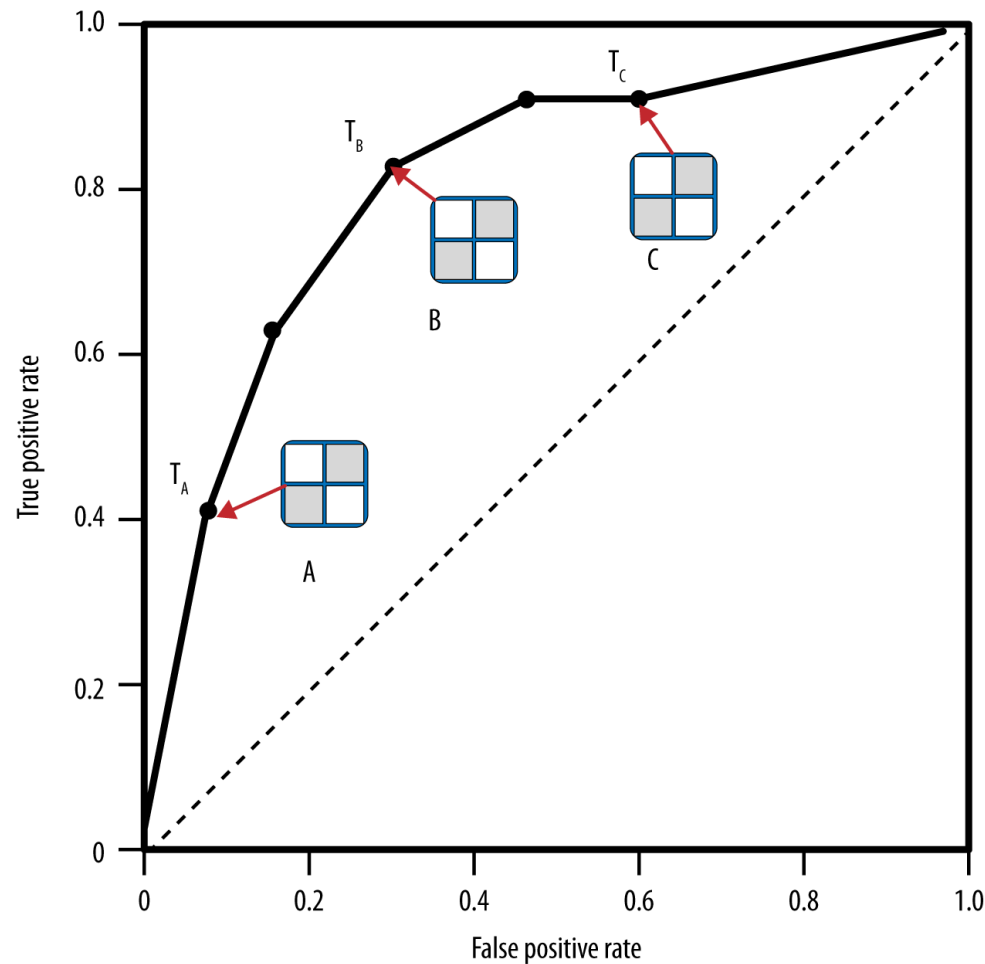
$$\frac{\text{False positive}}{\text{True Negative} + \text{False Positive}}$$



ROC Graphs and Curves



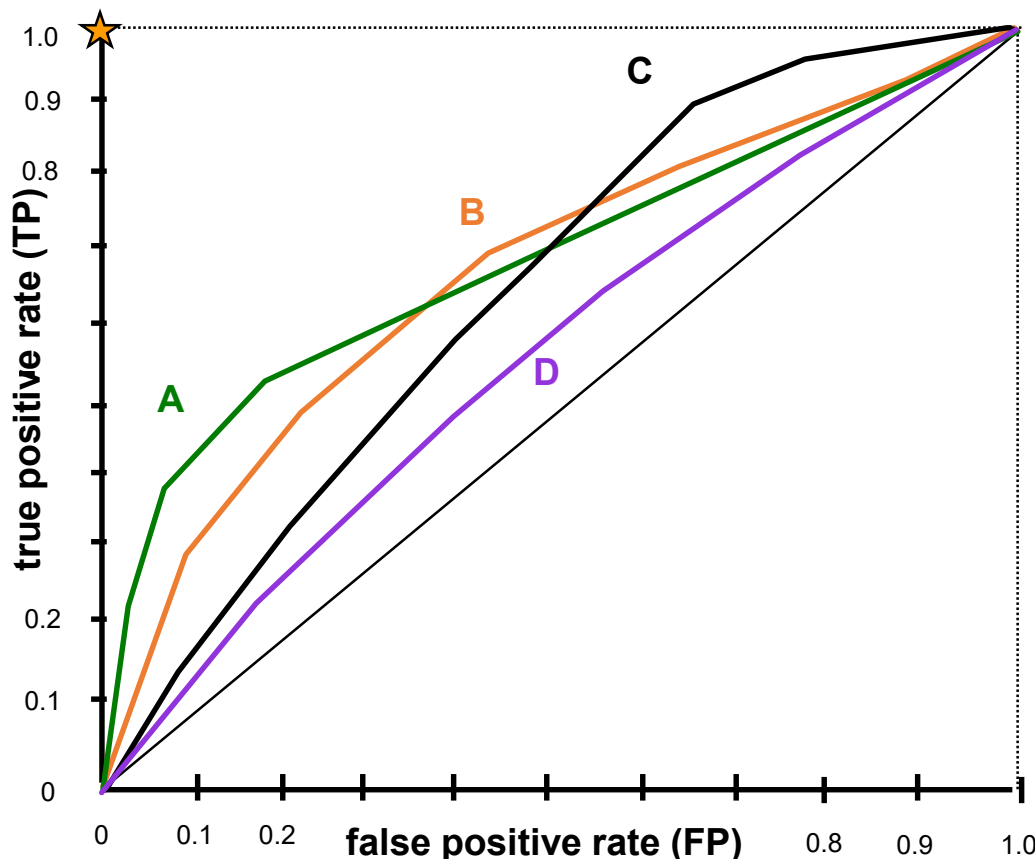
ROC Graphs and Curves





ROC Curves and Area under ROC curve (AUC)

- » ROC curves separate classifier performance from costs, benefits, and target class distributions



- Area Under Curve (AUC)
 - The area under a classifier's curve expressed as a fraction of the unit square
 - Measures the quality of a ranking/probability estimate model
 - The AUC is useful when a single number is needed to summarize performance, or when nothing is known about the operating conditions