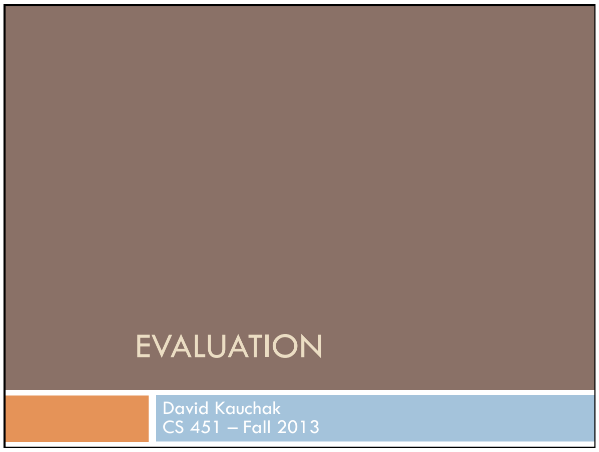
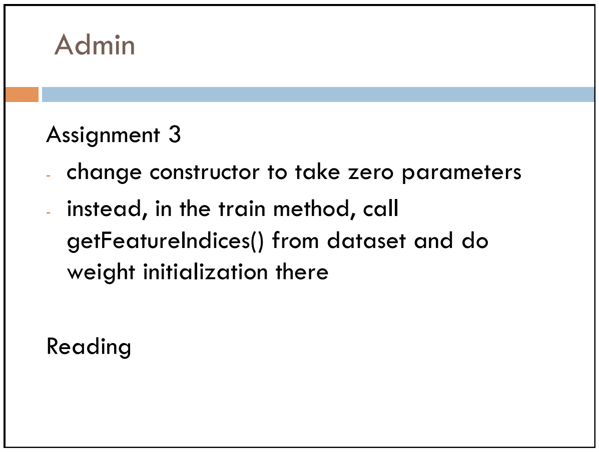


EVALUATION

David Kauchak
CS 451 – Fall 2013

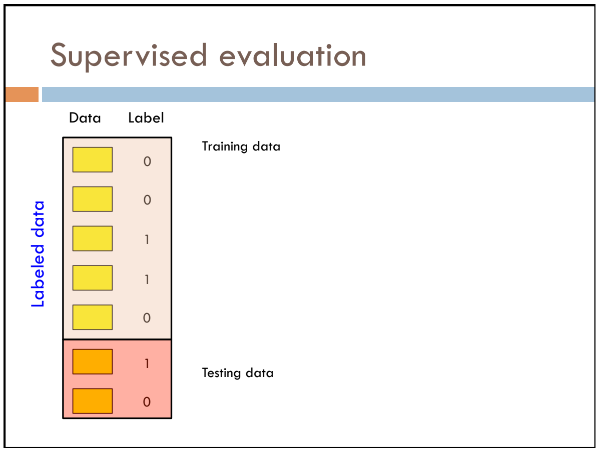


Admin

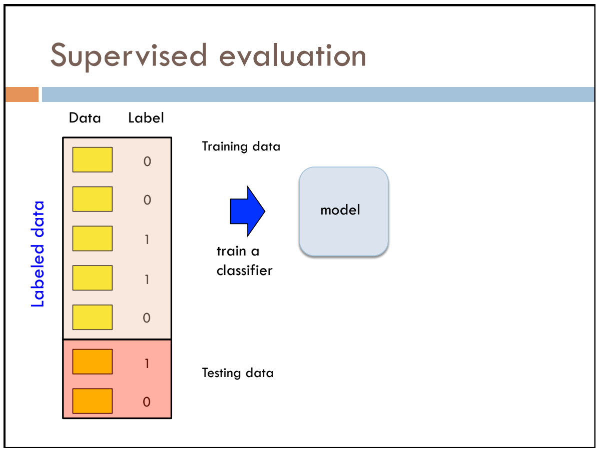
Assignment 3

- change constructor to take zero parameters
- instead, in the train method, call getFeatureIndices() from dataset and do weight initialization there

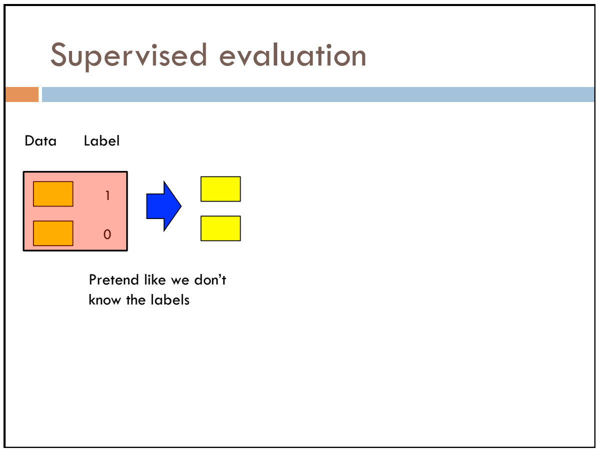
Reading



Supervised evaluation



Supervised evaluation



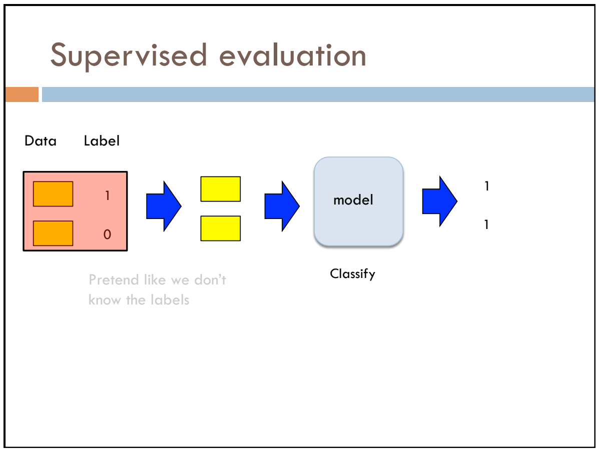
Supervised evaluation

Data Label

1

0

Pretend like we don't know the labels



Supervised evaluation

Data Label

1

0

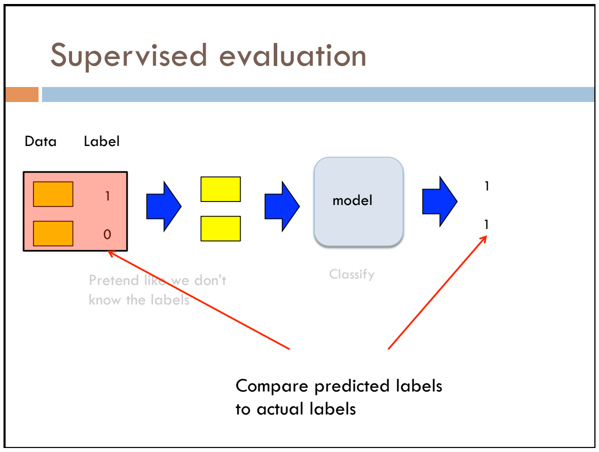
model

1

1

Pretend like we don't know the labels

Classify



Supervised evaluation

Data Label

1

0

model

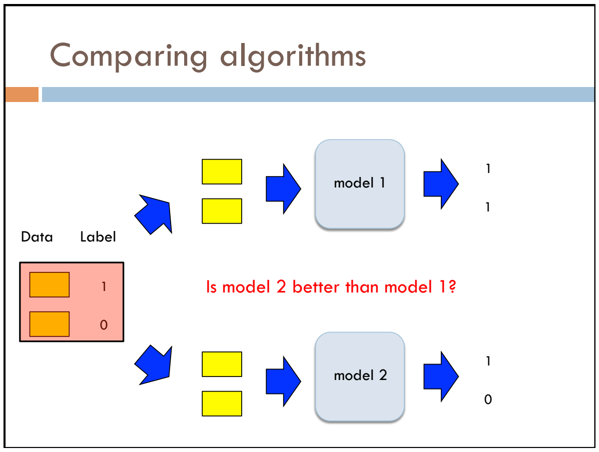
1

1

Pretend like we don't know the labels

Classify

Compare predicted labels to actual labels



Comparing algorithms

Data Label

1

0

model 1

1

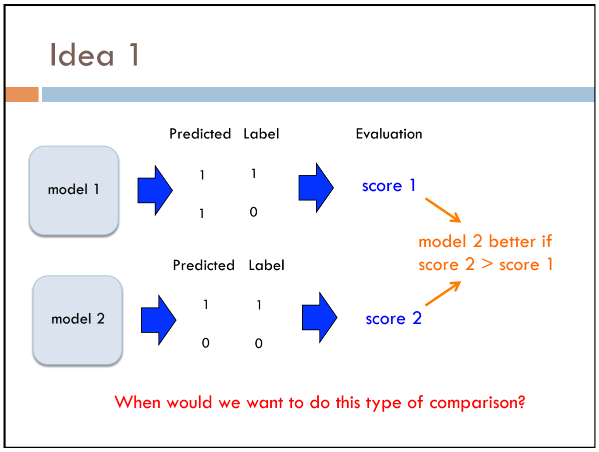
1

Is model 2 better than model 1?

model 2

1

0



Idea 1

model 1 →

Predicted	Label
1	1
1	0

→ Evaluation: score 1

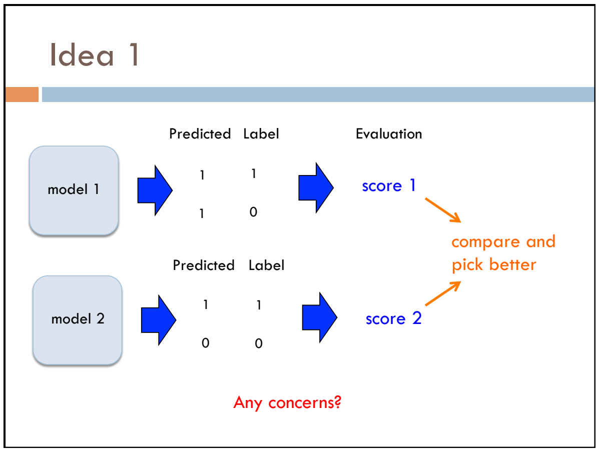
model 2 →

Predicted	Label
1	1
0	0

→ score 2

model 2 better if score 2 > score 1

When would we want to do this type of comparison?



Idea 1

model 1 →

Predicted	Label
1	1
1	0

→ Evaluation: score 1

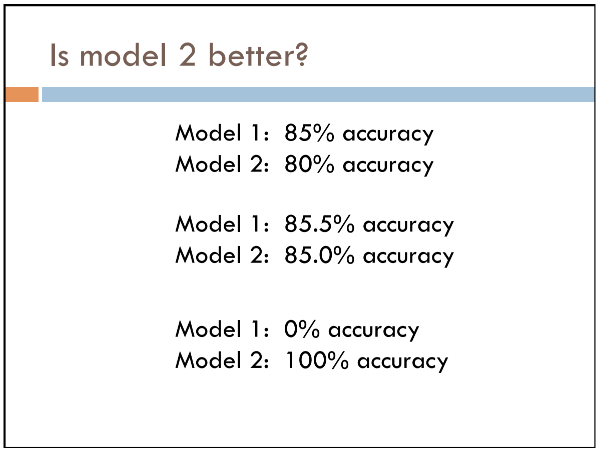
model 2 →

Predicted	Label
1	1
0	0

→ score 2

compare and pick better

Any concerns?

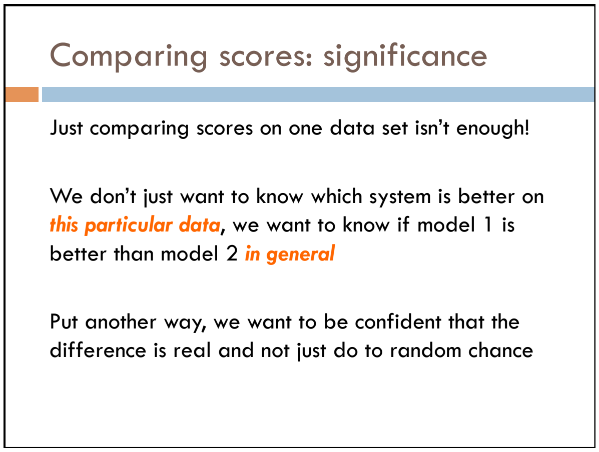


Is model 2 better?

Model 1: 85% accuracy
Model 2: 80% accuracy

Model 1: 85.5% accuracy
Model 2: 85.0% accuracy

Model 1: 0% accuracy
Model 2: 100% accuracy



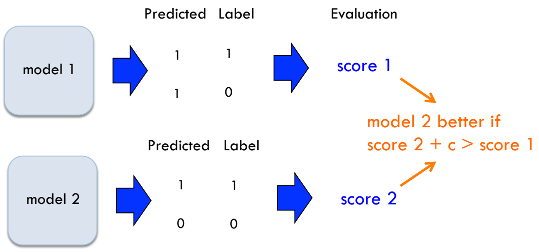
Comparing scores: significance

Just comparing scores on one data set isn't enough!

We don't just want to know which system is better on ***this particular data***, we want to know if model 1 is better than model 2 ***in general***

Put another way, we want to be confident that the difference is real and not just do to random chance

Idea 2



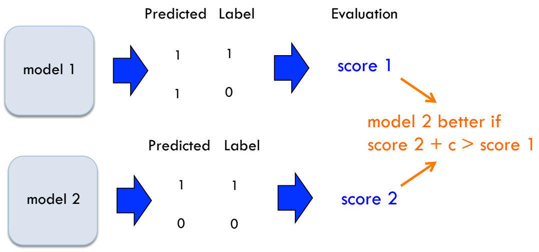
	Predicted	Label	Evaluation
model 1	1	1	score 1
	1	0	

	Predicted	Label	
model 2	1	1	score 2
	0	0	

model 2 better if
score 2 + c > score 1

Is this any better?

Idea 2



	Predicted	Label	Evaluation
model 1	1	1	score 1
	1	0	

	Predicted	Label	
model 2	1	1	score 2
	0	0	

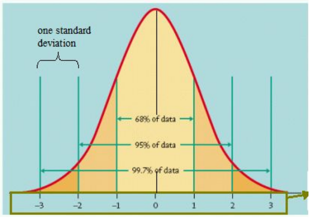
model 2 better if
score 2 + c > score 1

NO!

Key: we don't know the variance of the output

Variance

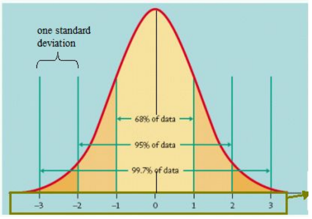
Recall that variance (or standard deviation) helped us predict how likely certain events are:



How do we know how variable a model's accuracy is?

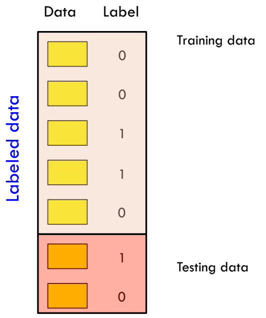
Variance

Recall that variance (or standard deviation) helped us predict how likely certain events are:



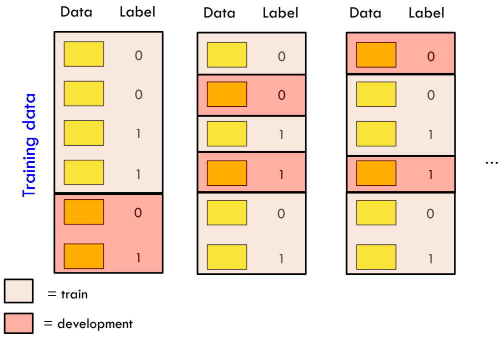
We need multiple accuracy scores! Ideas?

Repeated experimentation

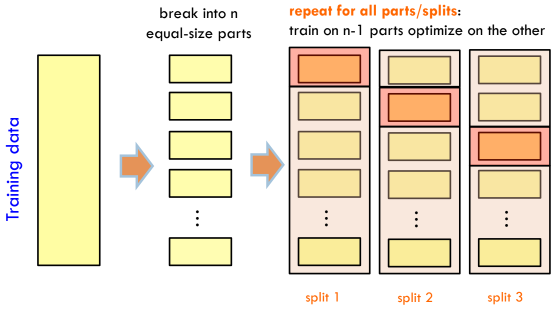


Rather than just splitting once, split multiple times

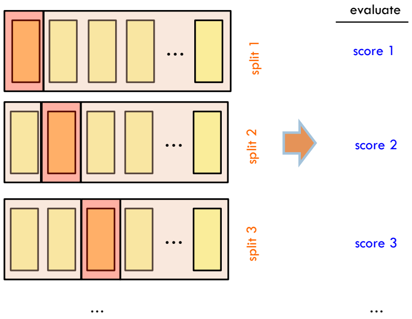
Repeated experimentation



n-fold cross validation


Training data
break into n
equal-size parts
repeat for all parts/splits:
train on n-1 parts optimize on the other
split 1
split 2
split 3

n-fold cross validation



n-fold cross validation

better utilization of labeled data

more robust: don’t just rely on one test/development set to evaluate the approach (or for optimizing parameters)

multiplies the computational overhead by n (have to train n models instead of just one)

10 is the most common choice of n

Leave-one-out cross validation

n-fold cross validation where n = number of examples

aka “jackknifing”

pros/cons?

when would we use this?

Leave-one-out cross validation

Can be very expensive if training is slow and/or if there are a large number of examples

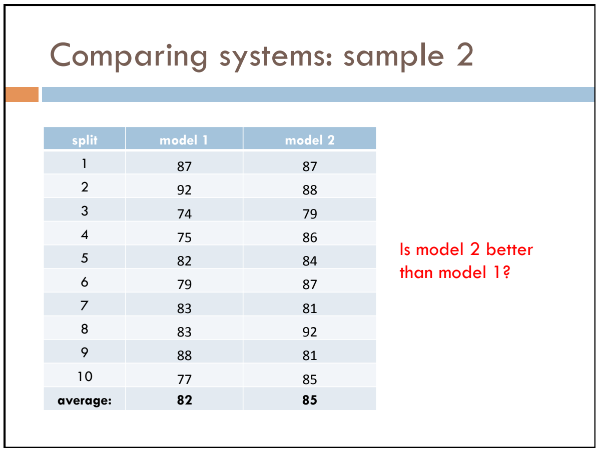
Useful in domains with limited training data: *maximizes the data we can use for training*

Some classifiers are very amenable to this approach (e.g.?)

Comparing systems: sample 1

split	model 1	model 2
1	87	88
2	85	84
3	83	84
4	80	79
5	88	89
6	85	85
7	83	81
8	87	86
9	88	89
10	84	85
average:	**85**	**85**

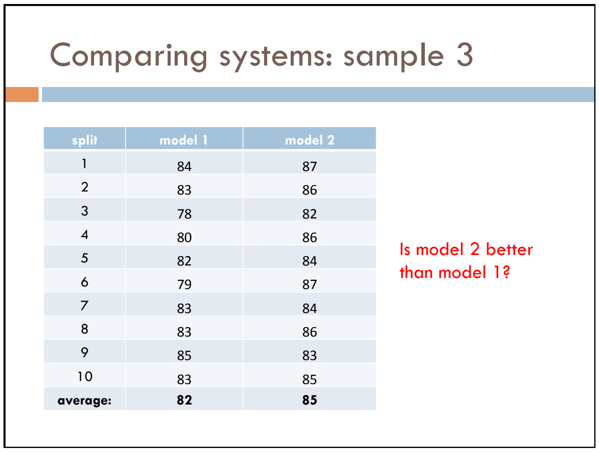
Is model 2 better than model 1?



Comparing systems: sample 2

split	model 1	model 2
1	87	87
2	92	88
3	74	79
4	75	86
5	82	84
6	79	87
7	83	81
8	83	92
9	88	81
10	77	85
average:	**82**	**85**

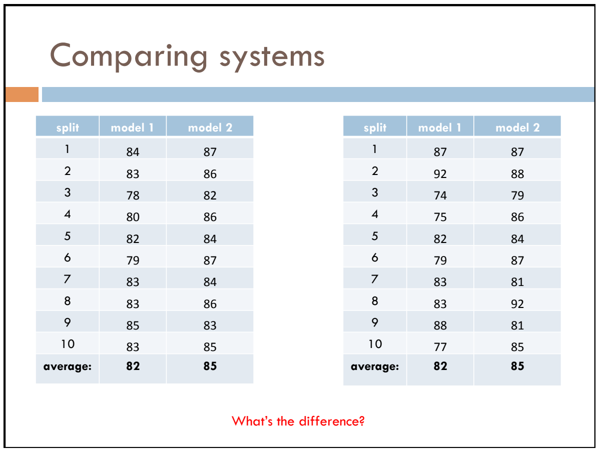
Is model 2 better than model 1?



Comparing systems: sample 3

split	model 1	model 2
1	84	87
2	83	86
3	78	82
4	80	86
5	82	84
6	79	87
7	83	84
8	83	86
9	85	83
10	83	85
average:	**82**	**85**

Is model 2 better than model 1?

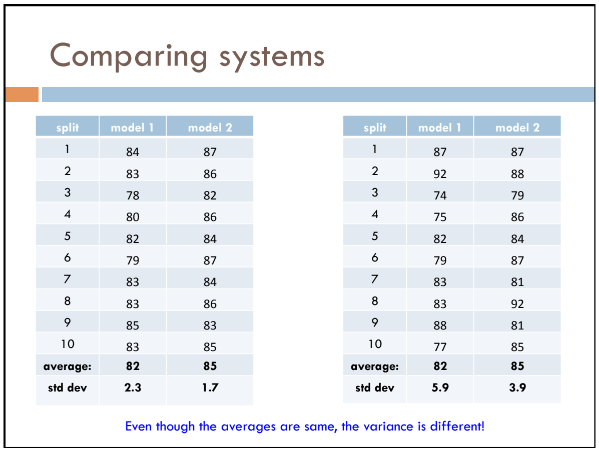


Comparing systems

split	model 1	model 2
1	84	87
2	83	86
3	78	82
4	80	86
5	82	84
6	79	87
7	83	84
8	83	86
9	85	83
10	83	85
average:	**82**	**85**

split	model 1	model 2
1	87	87
2	92	88
3	74	79
4	75	86
5	82	84
6	79	87
7	83	81
8	83	92
9	88	81
10	77	85
average:	**82**	**85**

What's the difference?

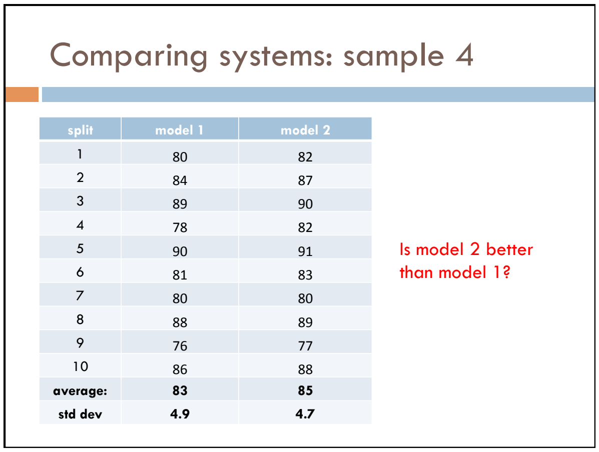


Comparing systems

split	model 1	model 2
1	84	87
2	83	86
3	78	82
4	80	86
5	82	84
6	79	87
7	83	84
8	83	86
9	85	83
10	83	85
average:	**82**	**85**
std dev	**2.3**	**1.7**

split	model 1	model 2
1	87	87
2	92	88
3	74	79
4	75	86
5	82	84
6	79	87
7	83	81
8	83	92
9	88	81
10	77	85
average:	**82**	**85**
std dev	**5.9**	**3.9**

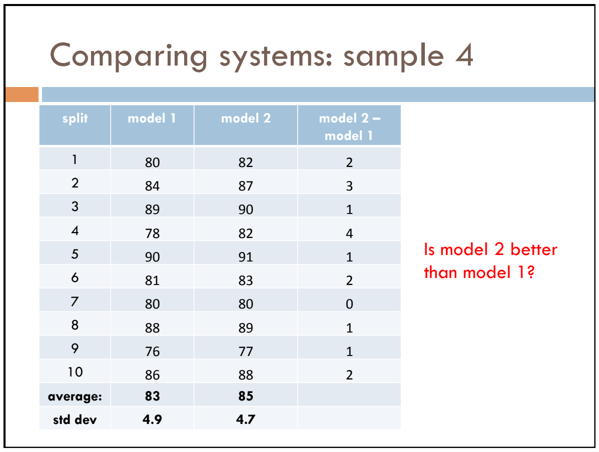
Even though the averages are same, the variance is different!



Comparing systems: sample 4

split	model 1	model 2
1	80	82
2	84	87
3	89	90
4	78	82
5	90	91
6	81	83
7	80	80
8	88	89
9	76	77
10	86	88
average:	**83**	**85**
std dev	**4.9**	**4.7**

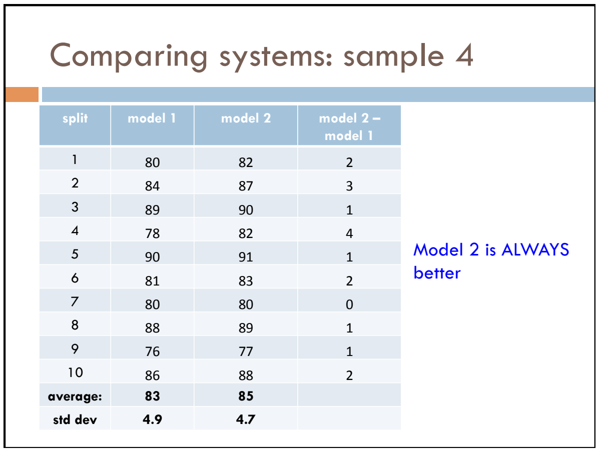
Is model 2 better than model 1?



Comparing systems: sample 4

split	model 1	model 2	model 2 – model 1
1	80	82	2
2	84	87	3
3	89	90	1
4	78	82	4
5	90	91	1
6	81	83	2
7	80	80	0
8	88	89	1
9	76	77	1
10	86	88	2
average:	**83**	**85**	
std dev	**4.9**	**4.7**	

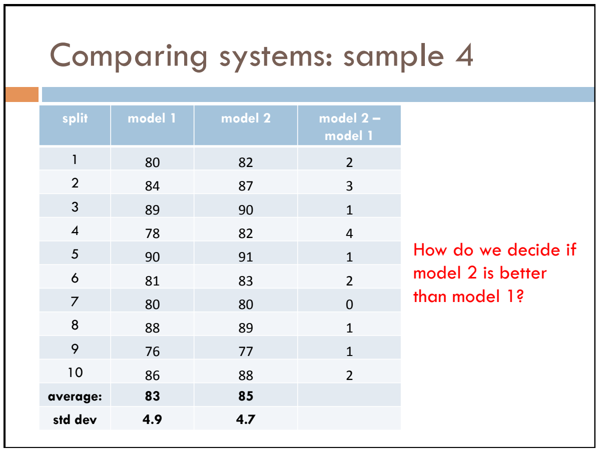
Is model 2 better than model 1?



Comparing systems: sample 4

split	model 1	model 2	model 2 – model 1
1	80	82	2
2	84	87	3
3	89	90	1
4	78	82	4
5	90	91	1
6	81	83	2
7	80	80	0
8	88	89	1
9	76	77	1
10	86	88	2
average:	**83**	**85**	
std dev	**4.9**	**4.7**	

Model 2 is ALWAYS better



Comparing systems: sample 4

split	model 1	model 2	model 2 – model 1
1	80	82	2
2	84	87	3
3	89	90	1
4	78	82	4
5	90	91	1
6	81	83	2
7	80	80	0
8	88	89	1
9	76	77	1
10	86	88	2
average:	**83**	**85**	
std dev	**4.9**	**4.7**	

How do we decide if model 2 is better than model 1?

Statistical tests

Setup:

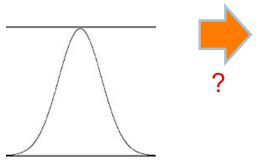
- Assume some default hypothesis about the data that you'd like to *disprove*, called the null hypothesis
- e.g. model 1 and model 2 are not statistically different in performance

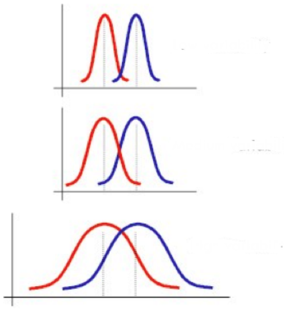
Test:

- Calculate a test statistic from the data (often assuming something about the data)
- Based on this statistic, with *some probability* we can **reject the null hypothesis**, that is, show that it does not hold

t-test

Determines whether two samples come from the same underlying distribution or not





t-test

Null hypothesis: model 1 and model 2 accuracies are no different, i.e. come from **the same** distribution

Assumptions: there are a number that often aren't completely true, but we're often not too far off

Result: probability that the difference in accuracies is due to random chance (low values are better)

Calculating t-test

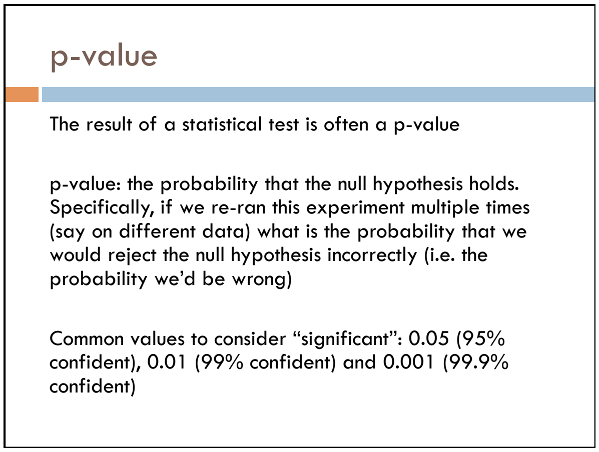
For our setup, we'll do what's called a "pair t-test"

- The values can be thought of as pairs, where they were calculated under the same conditions
- In our case, the same train/test split
- Gives more power than the unpaired t-test (we have more information)

For almost all experiments, we'll do a "two-tailed" version of the t-test

Can calculate by hand or in code, but why reinvent the wheel: use excel or a statistical package

http://en.wikipedia.org/wiki/Student's_t-test

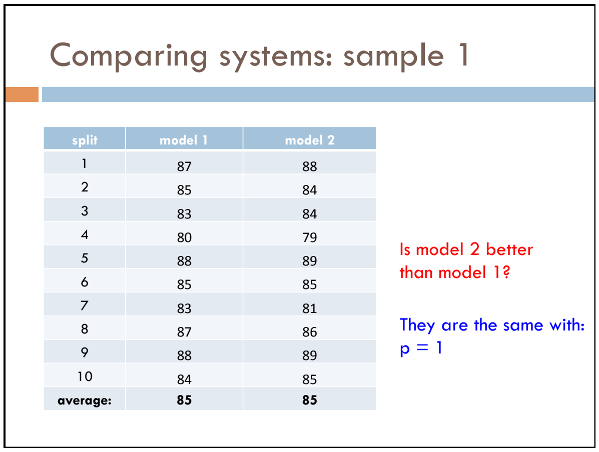


p-value

The result of a statistical test is often a p-value

p-value: the probability that the null hypothesis holds. Specifically, if we re-ran this experiment multiple times (say on different data) what is the probability that we would reject the null hypothesis incorrectly (i.e. the probability we'd be wrong)

Common values to consider "significant": 0.05 (95% confident), 0.01 (99% confident) and 0.001 (99.9% confident)

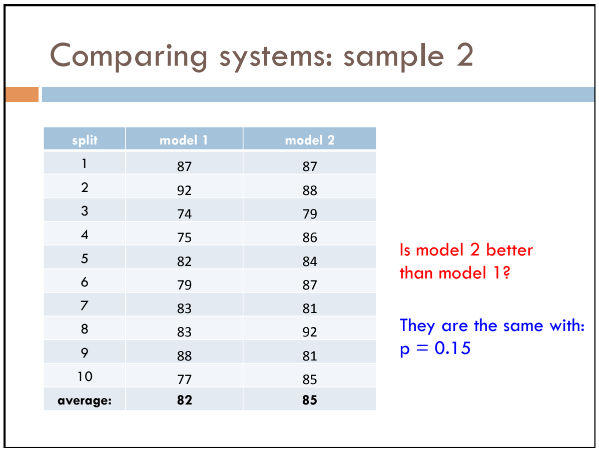


Comparing systems: sample 1

split	model 1	model 2
1	87	88
2	85	84
3	83	84
4	80	79
5	88	89
6	85	85
7	83	81
8	87	86
9	88	89
10	84	85
average:	**85**	**85**

Is model 2 better than model 1?

They are the same with: p = 1

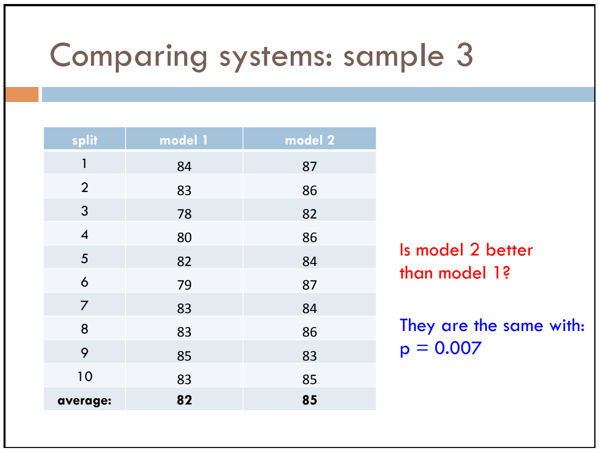


Comparing systems: sample 2

split	model 1	model 2
1	87	87
2	92	88
3	74	79
4	75	86
5	82	84
6	79	87
7	83	81
8	83	92
9	88	81
10	77	85
average:	**82**	**85**

Is model 2 better than model 1?

They are the same with: p = 0.15

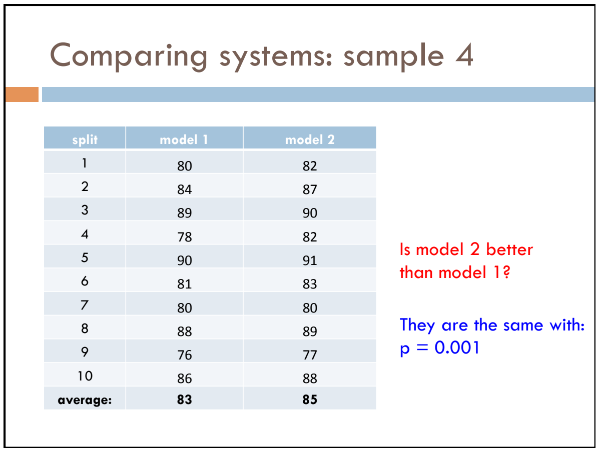


Comparing systems: sample 3

split	model 1	model 2
1	84	87
2	83	86
3	78	82
4	80	86
5	82	84
6	79	87
7	83	84
8	83	86
9	85	83
10	83	85
average:	**82**	**85**

Is model 2 better than model 1?

They are the same with: p = 0.007

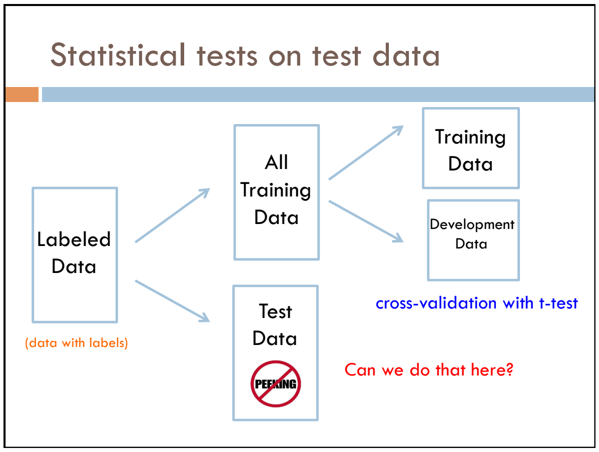


Comparing systems: sample 4

split	model 1	model 2
1	80	82
2	84	87
3	89	90
4	78	82
5	90	91
6	81	83
7	80	80
8	88	89
9	76	77
10	86	88
average:	**83**	**85**

Is model 2 better than model 1?

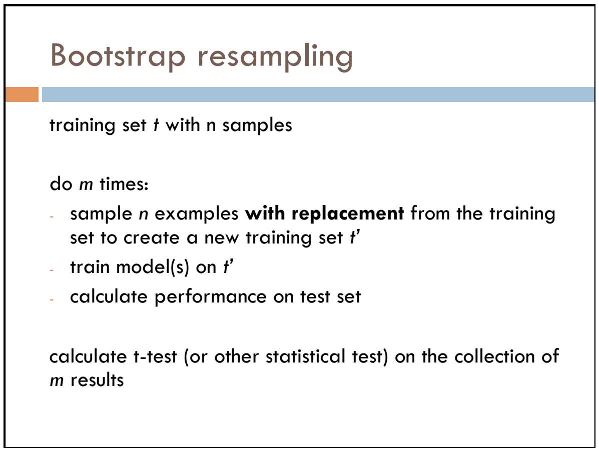
They are the same with: p = 0.001



Statistical tests on test data

cross-validation with t-test

Can we do that here?



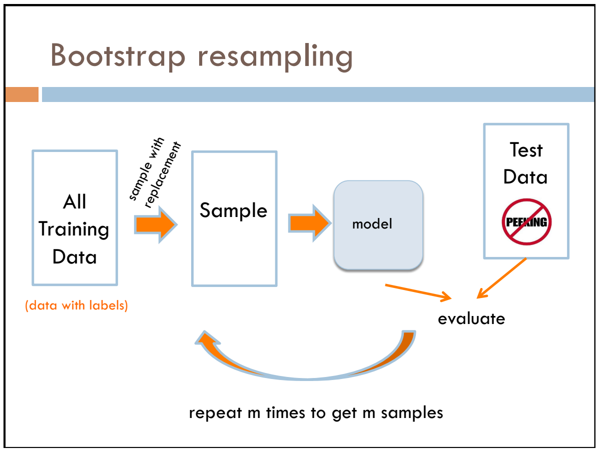
Bootstrap resampling

training set *t* with n samples

do *m* times:

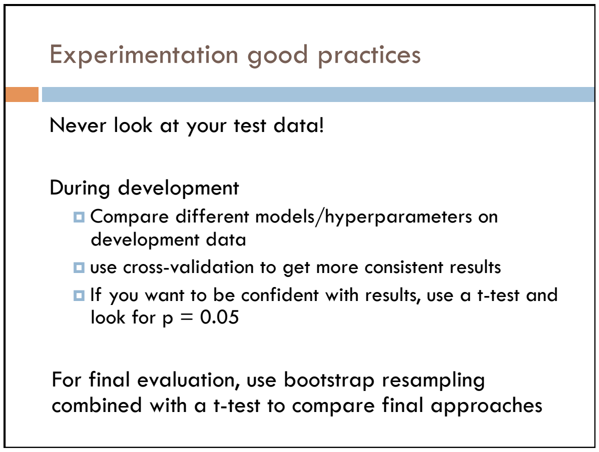
- sample *n* examples **with replacement** from the training set to create a new training set *t'*
- train model(s) on *t'*
- calculate performance on test set

calculate t-test (or other statistical test) on the collection of *m* results



Bootstrap resampling

repeat m times to get m samples



Experimentation good practices

Never look at your test data!

During development

- Compare different models/hyperparameters on development data
- use cross-validation to get more consistent results
- If you want to be confident with results, use a t-test and look for p = 0.05

For final evaluation, use bootstrap resampling combined with a t-test to compare final approaches