

# EVALUATION

David Kauchak  
CS 158 – Fall 2025

# Admin



## Assignment 3

## Reading

Extra mentor hours this week: Sat, 9-11 am

# So far...


1. Throw out outlier examples
2. Remove noisy features
3. Pick “good” features
4. Normalize feature values
  1. center data
  2. scale data (either variance or absolute)
5. Normalize example length
6. **Finally, train your model!**

# What about testing?

training data  
(labeled examples)

| Terrain | Unicycle-type | Weather | Go-For-Bride? |
|---------|---------------|---------|---------------|
| Trail   | Normal        | Rainy   | NO            |
| Road    | Normal        | Sunny   | YES           |
| Trail   | Mountain      | Sunny   | YES           |
| Road    | Mountain      | Rainy   | YES           |
| Trail   | Normal        | Snowy   | NO            |
| Road    | Normal        | Rainy   | YES           |
| Road    | Mountain      | Snowy   | YES           |
| Trail   | Normal        | Sunny   | NO            |
| Road    | Normal        | Snowy   | NO            |
| Trail   | Mountain      | Snowy   | YES           |

pre-process data



| Terrain | Unicycle-type | Weather | Go-For-Bride? |
|---------|---------------|---------|---------------|
| Trail   | Normal        | Rainy   | NO            |
| Road    | Normal        | Sunny   | YES           |
| Trail   | Mountain      | Sunny   | YES           |
| Road    | Mountain      | Rainy   | YES           |
| Trail   | Normal        | Snowy   | NO            |
| Road    | Normal        | Rainy   | YES           |
| Road    | Mountain      | Snowy   | YES           |
| Trail   | Normal        | Sunny   | NO            |
| Road    | Normal        | Snowy   | NO            |
| Trail   | Mountain      | Snowy   | YES           |

learn



model/  
classifier

“better” training data

# What about testing?

test data

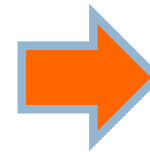
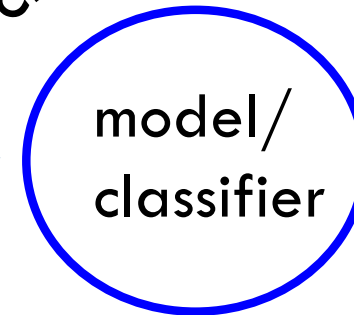
| Terrain | Unicycle-type | Weather | Go-Far-Ride? |
|---------|---------------|---------|--------------|
| Trail   | Normal        | Rainy   | NO           |
| Road    | Normal        | Sunny   | YES          |
| Trail   | Mountain      | Sunny   | YES          |
| Road    | Mountain      | Rainy   | YES          |
| Trail   | Normal        | Snowy   | NO           |
| Road    | Normal        | Rainy   | YES          |
| Road    | Mountain      | Snowy   | YES          |
| Trail   | Normal        | Sunny   | NO           |
| Road    | Normal        | Snowy   | NO           |
| Trail   | Mountain      | Snowy   | YES          |

pre-process data



| Terrain | Unicycle-type | Weather | Go-Far-Ride? |
|---------|---------------|---------|--------------|
| Trail   | Normal        | Rainy   | NO           |
| Road    | Normal        | Sunny   | YES          |
| Trail   | Mountain      | Sunny   | YES          |
| Road    | Mountain      | Rainy   | YES          |
| Trail   | Normal        | Snowy   | NO           |
| Road    | Normal        | Rainy   | YES          |
| Road    | Mountain      | Snowy   | YES          |
| Trail   | Normal        | Sunny   | NO           |
| Road    | Normal        | Snowy   | NO           |
| Trail   | Mountain      | Snowy   | YES          |

classify



prediction

How do we preprocess the test data?

# Test data preprocessing

1. Throw out outlier examples
2. Remove noisy features
3. Pick “good” features
4. Normalize feature values
  1. center data
  2. scale data (either variance or absolute)
5. Normalize example length

Which of these do we need to do on test data?  
Any issues?

# Test data preprocessing

- ~~1. Throw out outlier examples~~
2. Remove irrelevant/noisy features Remove/pick same features
3. Pick “good” features
4. Normalize feature values Do these
  1. center data
  2. scale data (either variance or absolute) Do this
5. Normalize example length

Whatever you do on training, you have to do the EXACT same on testing!

# Normalizing test data



For each feature (over all examples):

**Center:** adjust the values so that the mean of that feature is 0: subtract the **mean** from all values

Rescale/adjust feature values to avoid magnitude bias:

- ▣ **Variance scaling:** divide each value by the **std dev**
- ▣ **Absolute scaling:** divide each value by the **largest value**

What values do we use when normalizing testing data?



# Normalizing test data

For each feature (over all examples):

**Center:** adjust the values so that the mean of that feature is 0: subtract the **mean** from all values

Rescale/adjust feature values to avoid magnitude bias:

- ▣ **Variance scaling:** divide each value by the **std dev**
- ▣ **Absolute scaling:** divide each value by the **largest value**

Save these from training normalization!

# Normalizing test data

training data  
(labeled examples)

| Terrain | Unicycle-type | Weather | Go-Far-Ride? |
|---------|---------------|---------|--------------|
| Trail   | Normal        | Rainy   | NO           |
| Road    | Normal        | Sunny   | YES          |
| Trail   | Mountain      | Sunny   | YES          |
| Road    | Mountain      | Rainy   | YES          |
| Trail   | Normal        | Snowy   | NO           |
| Road    | Normal        | Rainy   | YES          |
| Road    | Mountain      | Snowy   | YES          |
| Trail   | Normal        | Sunny   | NO           |
| Road    | Normal        | Snowy   | NO           |
| Trail   | Mountain      | Snowy   | YES          |

test data

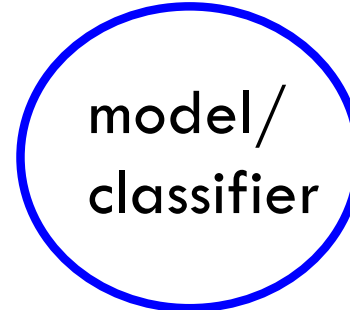
| Terrain | Unicycle-type | Weather | Go-Far-Ride? |
|---------|---------------|---------|--------------|
| Trail   | Normal        | Rainy   | NO           |
| Road    | Normal        | Sunny   | YES          |
| Trail   | Mountain      | Sunny   | YES          |
| Road    | Mountain      | Rainy   | YES          |
| Trail   | Normal        | Snowy   | NO           |
| Road    | Normal        | Rainy   | YES          |
| Road    | Mountain      | Snowy   | YES          |
| Trail   | Normal        | Sunny   | NO           |
| Road    | Normal        | Snowy   | NO           |
| Trail   | Mountain      | Snowy   | YES          |

| Terrain | Unicycle-type | Weather | Go-Far-Ride? |
|---------|---------------|---------|--------------|
| Trail   | Normal        | Rainy   | NO           |
| Road    | Normal        | Sunny   | YES          |
| Trail   | Mountain      | Sunny   | YES          |
| Road    | Mountain      | Rainy   | YES          |
| Trail   | Normal        | Snowy   | NO           |
| Road    | Normal        | Rainy   | YES          |
| Road    | Mountain      | Snowy   | YES          |
| Trail   | Normal        | Sunny   | NO           |
| Road    | Normal        | Snowy   | NO           |
| Trail   | Mountain      | Snowy   | YES          |

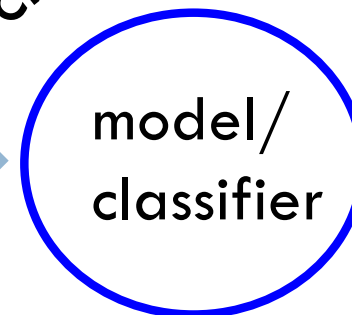
mean, std dev, max,...

| Terrain | Unicycle-type | Weather | Go-Far-Ride? |
|---------|---------------|---------|--------------|
| Trail   | Normal        | Rainy   | NO           |
| Road    | Normal        | Sunny   | YES          |
| Trail   | Mountain      | Sunny   | YES          |
| Road    | Mountain      | Rainy   | YES          |
| Trail   | Normal        | Snowy   | NO           |
| Road    | Normal        | Rainy   | YES          |
| Road    | Mountain      | Snowy   | YES          |
| Trail   | Normal        | Sunny   | NO           |
| Road    | Normal        | Snowy   | NO           |
| Trail   | Mountain      | Snowy   | YES          |

learn



classify



prediction

pre-process data

# Features pre-processing summary

Many techniques for preprocessing data








Which ones will work well will depend on the data and the classifier

Try them out and evaluate how they affect performance on dev data

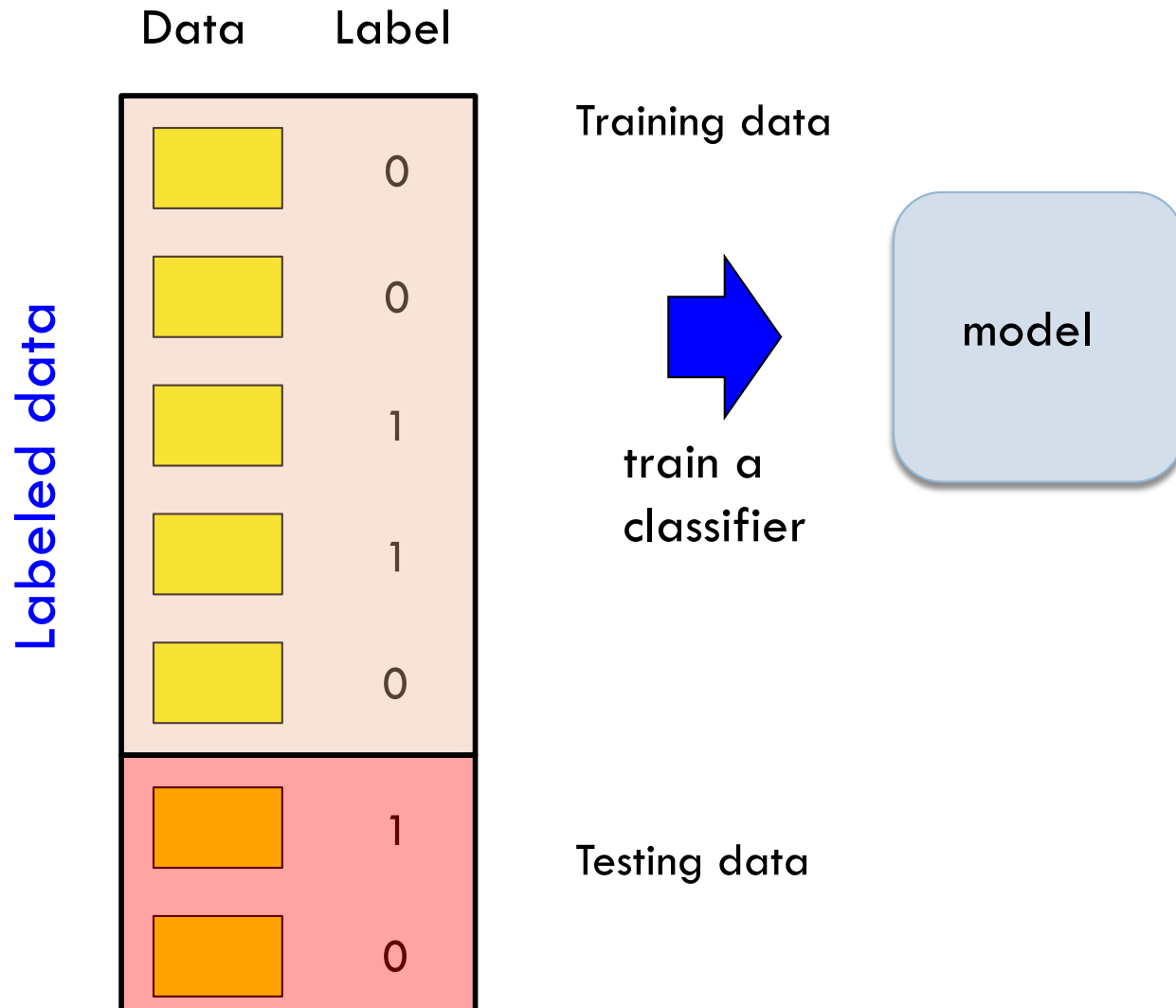
Make sure to do the **exact same** pre-processing on train and test

1. Throw out outlier examples
2. Remove noisy features
3. Pick “good” features
4. Normalize feature values
  1. center data
  2. scale data (either variance or absolute)
5. Normalize example length

# Supervised evaluation

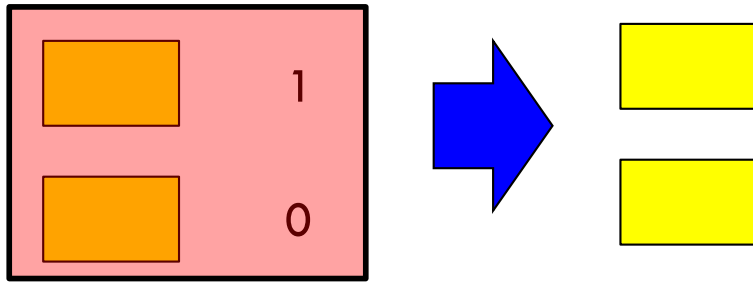
|              | Data  | Label |               |
|--------------|---|-------|---------------|
| Labeled data |    | 0     | Training data |
|              |    | 0     |               |
|              |    | 1     |               |
|              |   | 1     |               |
|              |  | 0     |               |
|              |  | 1     | Testing data  |
|              |  | 0     |               |

# Supervised evaluation



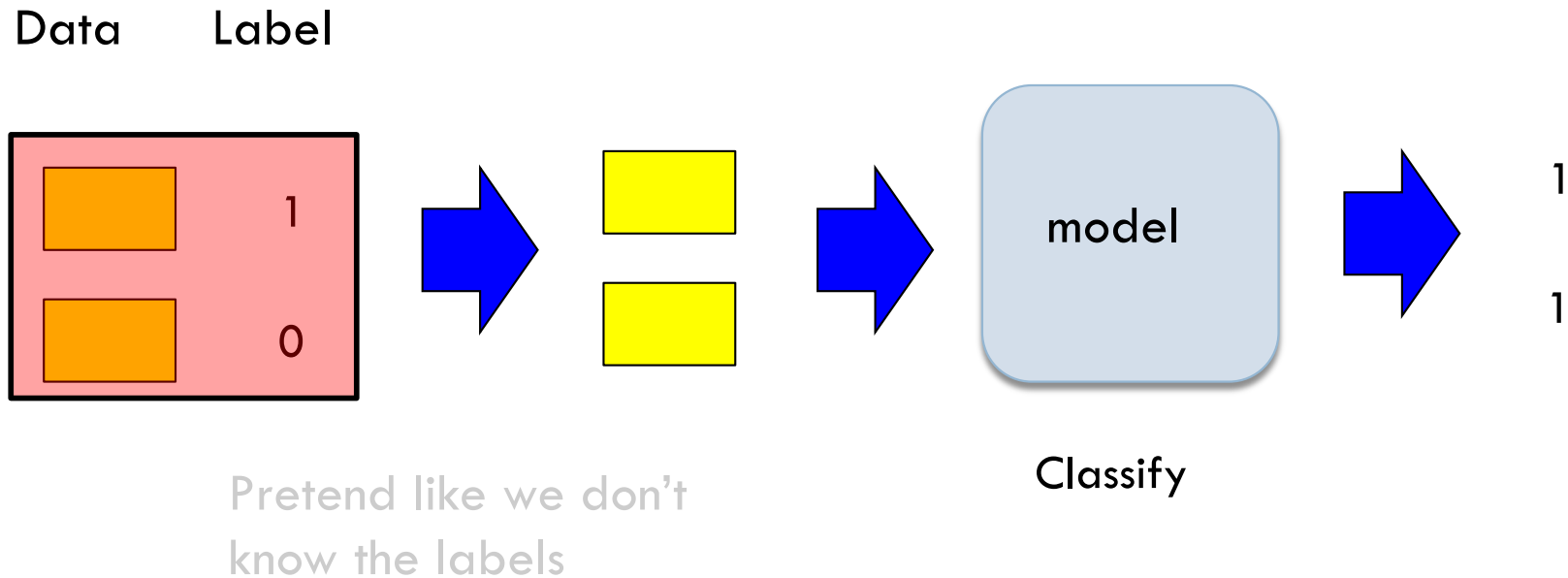
# Supervised evaluation

Data      Label

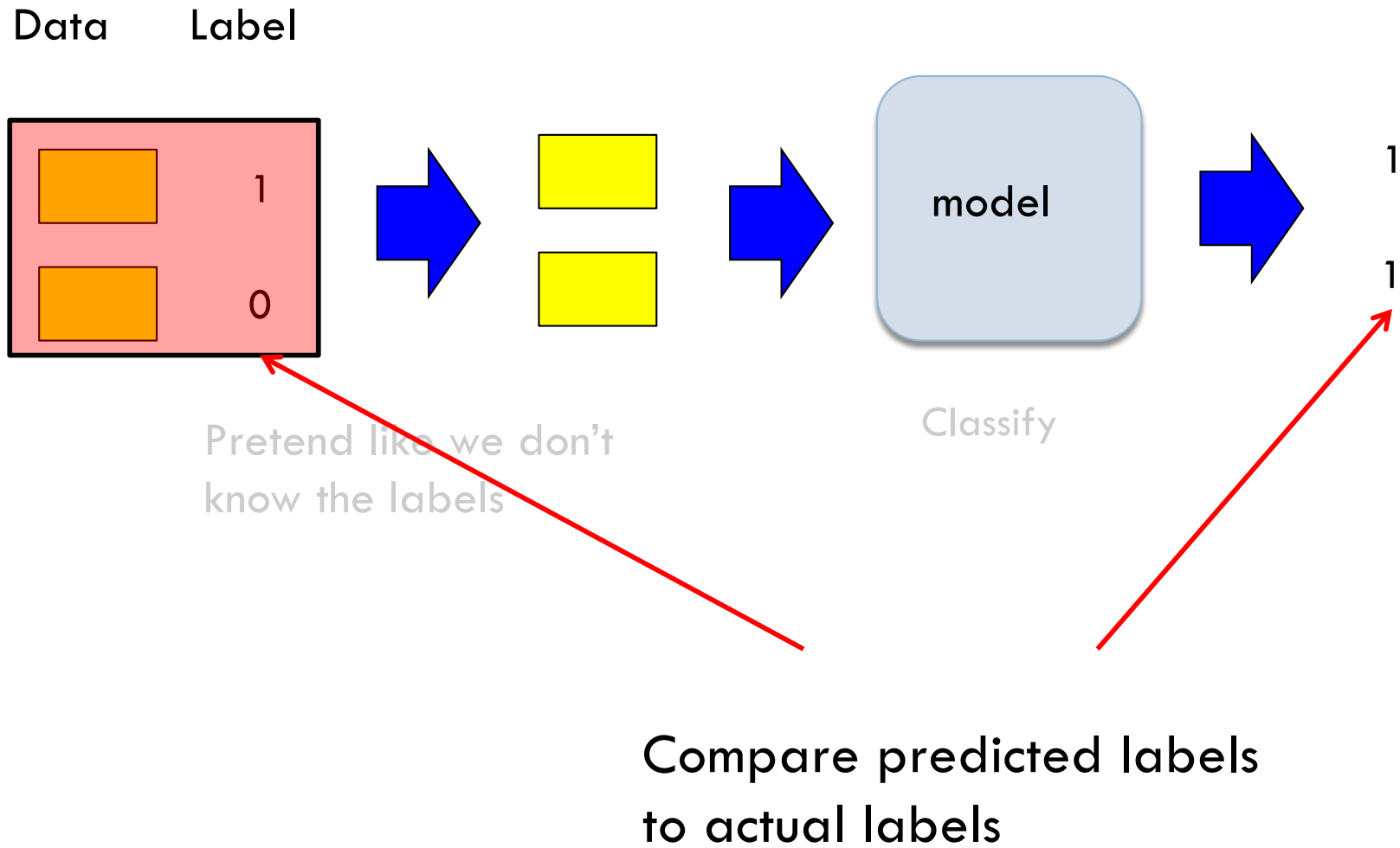


Pretend like we don't  
know the labels

# Supervised evaluation

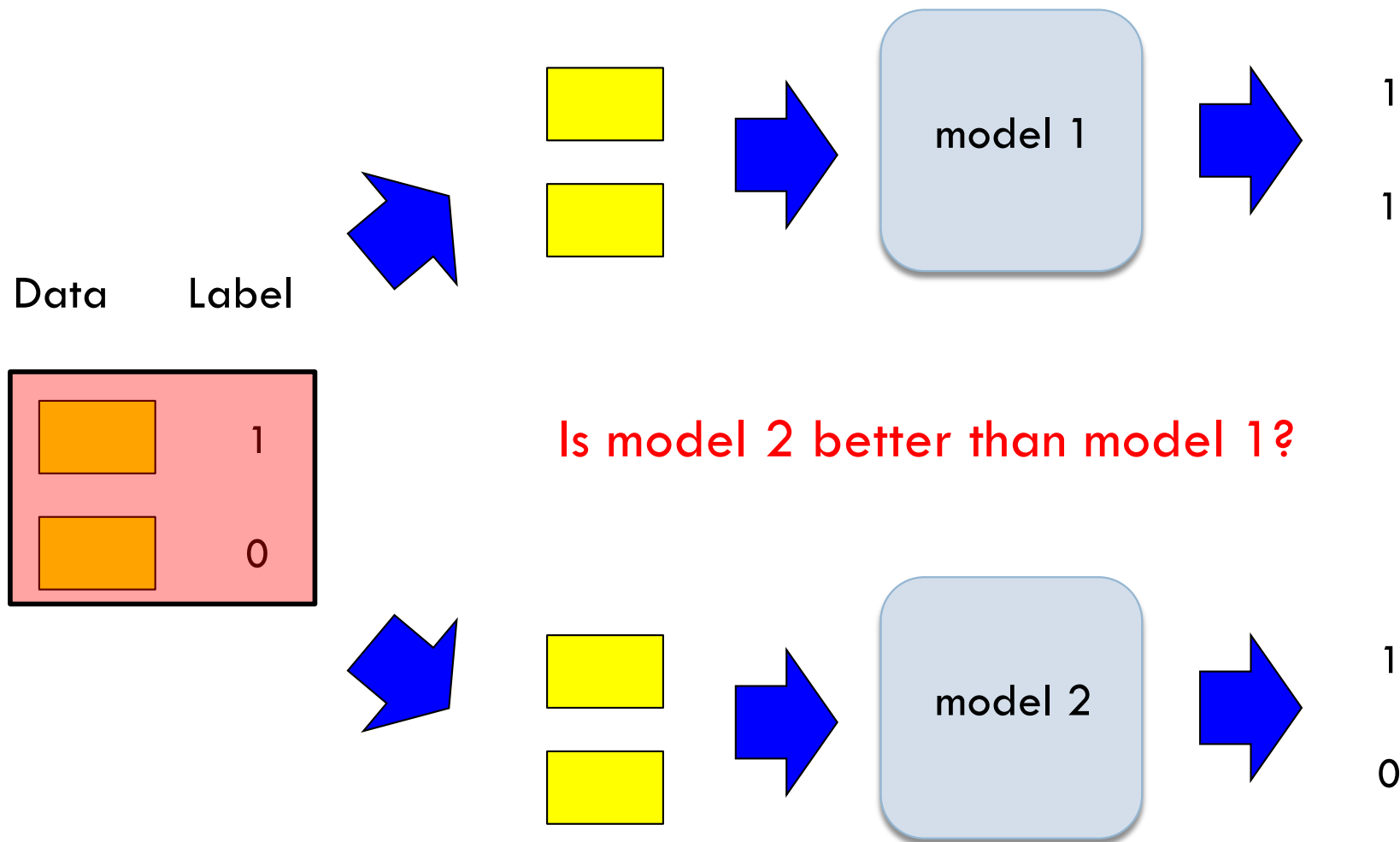


# Supervised evaluation

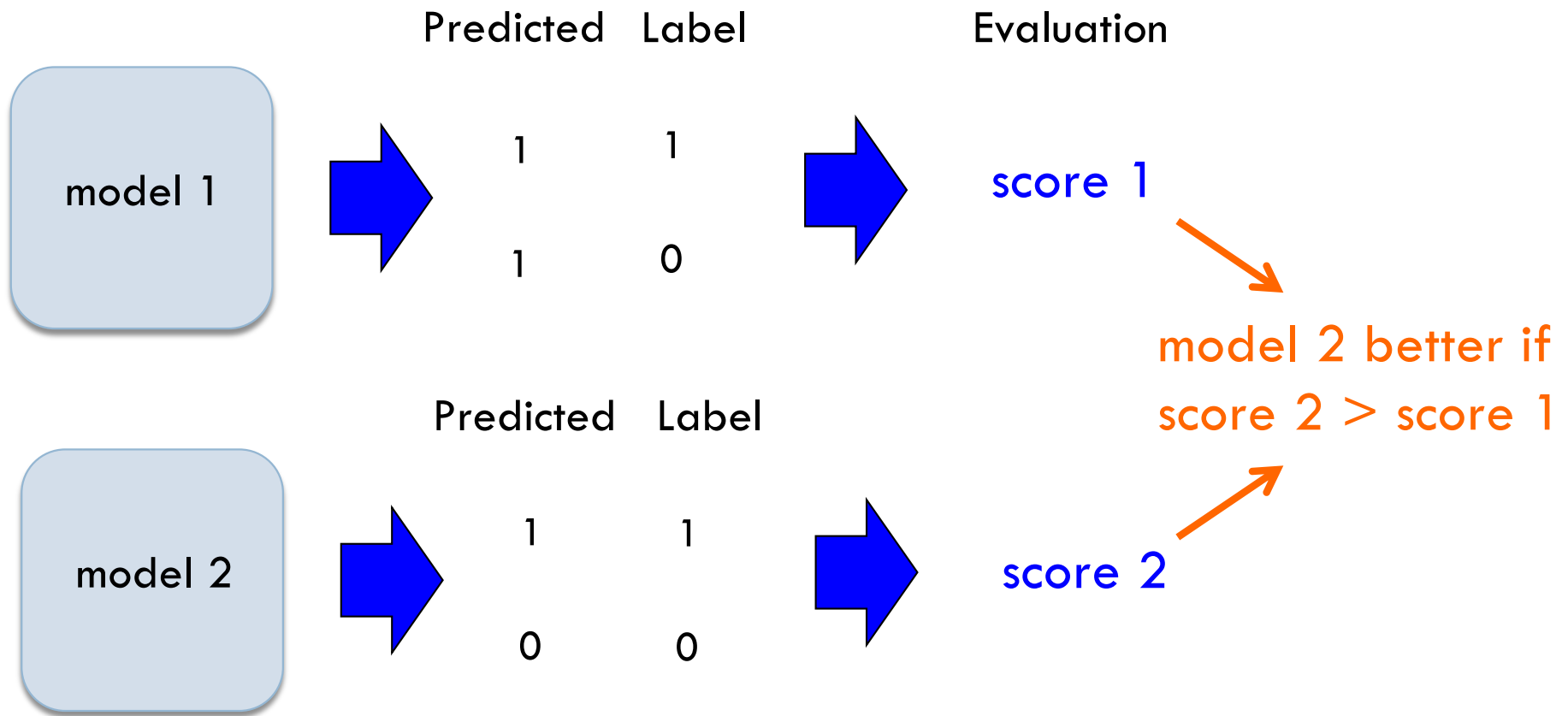




# Comparing algorithms

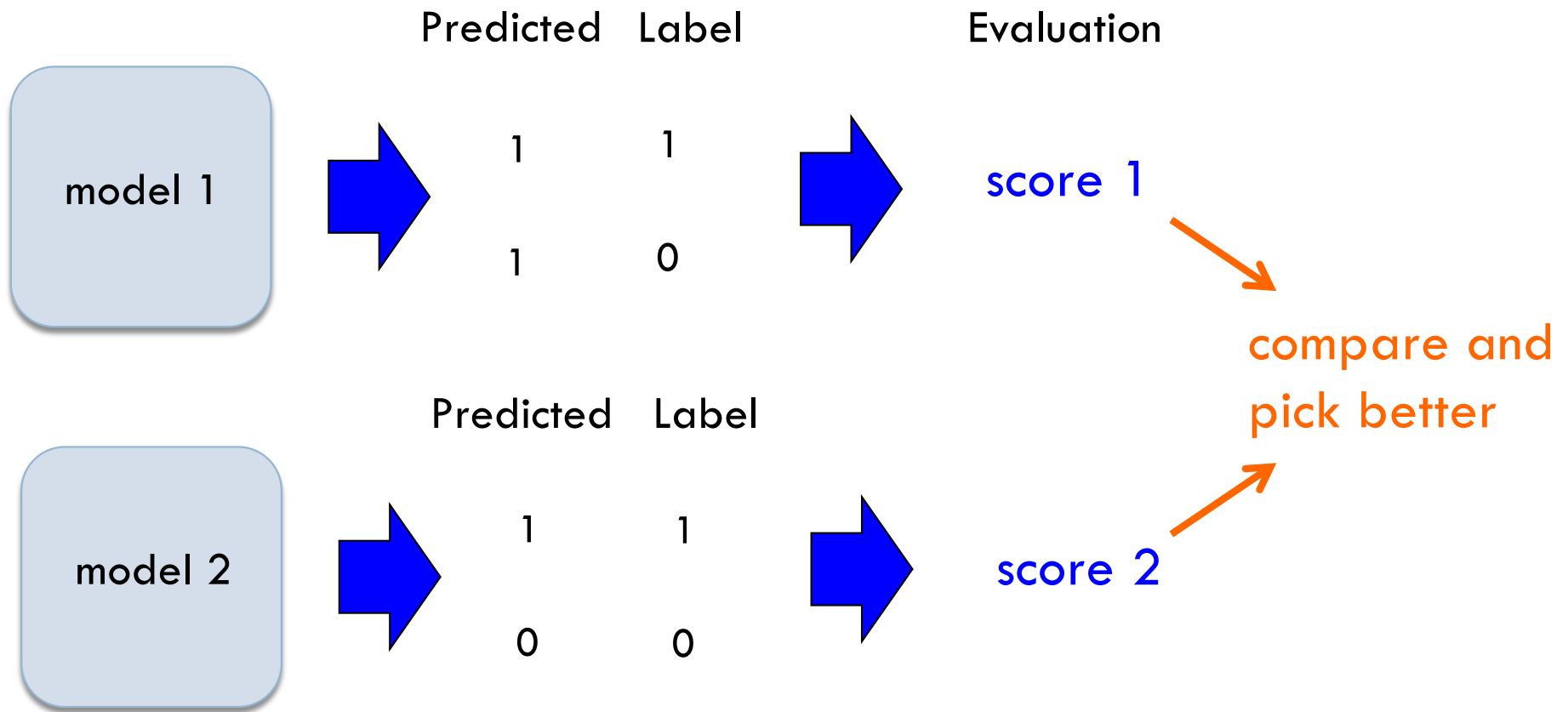


# Idea 1



When would we want to do this type of comparison?

# Idea 1



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

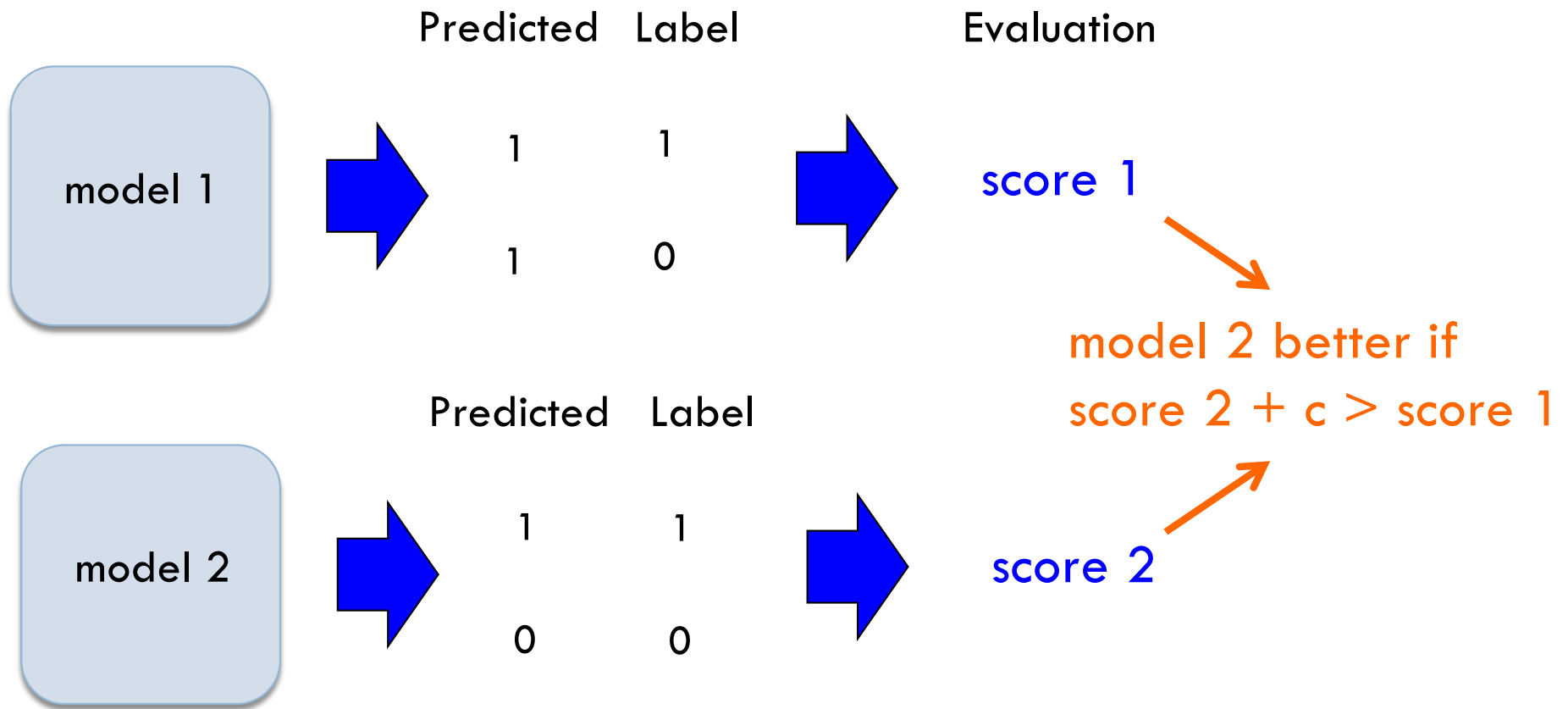


Just comparing scores on one data set isn't enough!

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

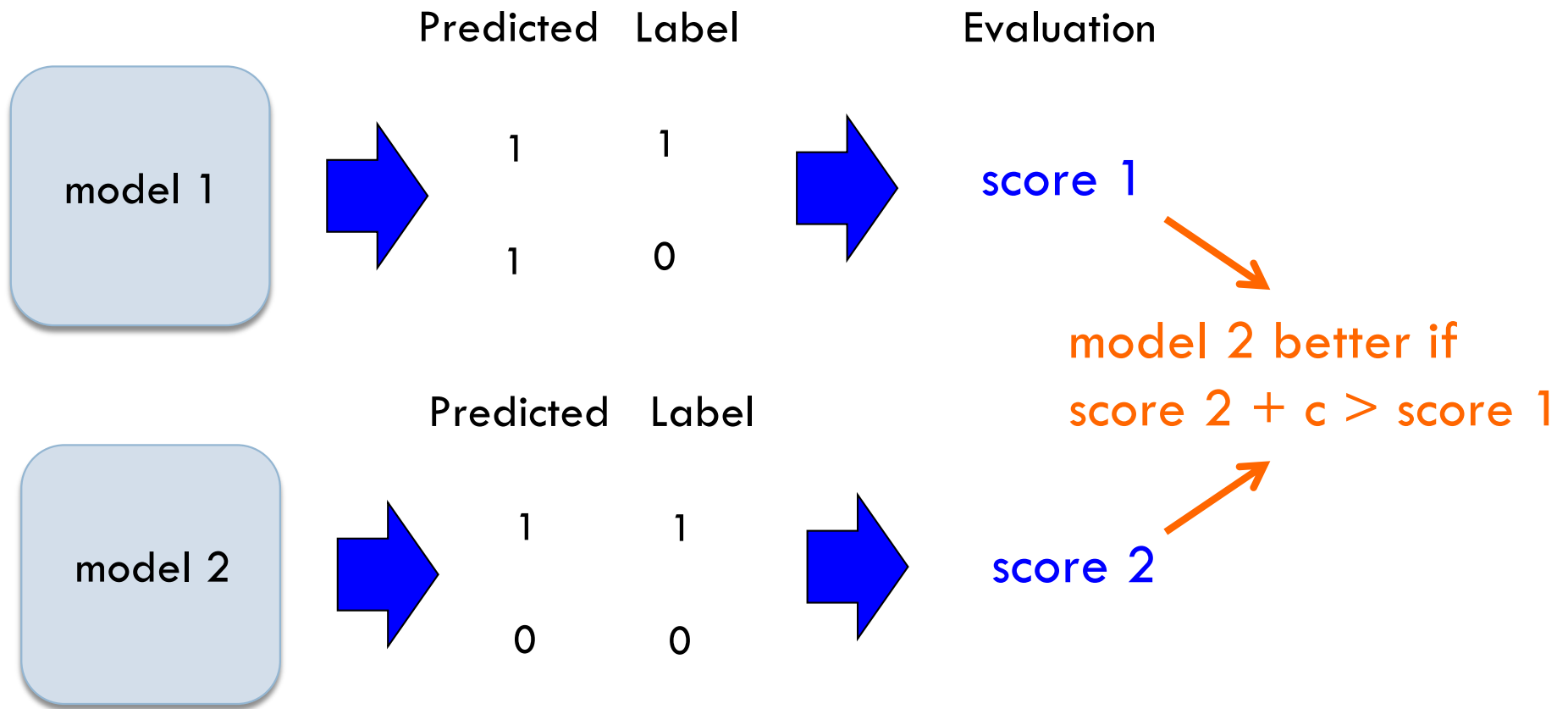
Put another way, we want to be confident that the difference is real and not just due to chance or eccentricities about the particular dataset

# Idea 2



Is this any better?

# Idea 2

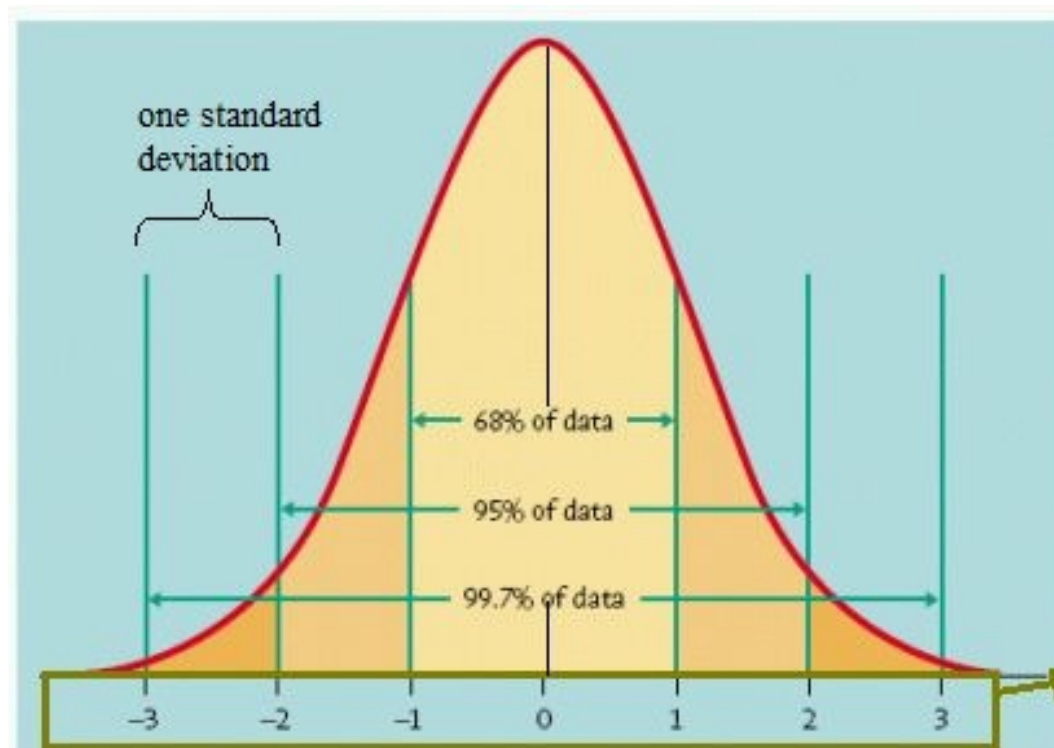


**NO!**

**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:

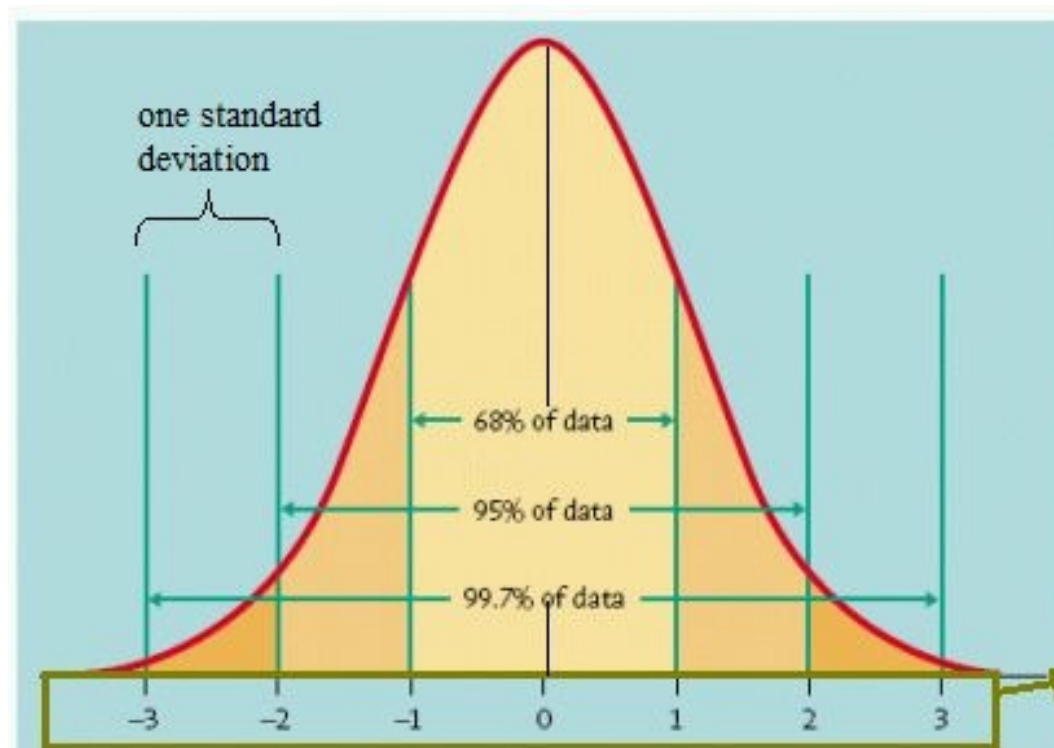


How do we know how variable a model's accuracy is?










# Variance

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



We need multiple accuracy scores! Ideas?

# Repeated experimentation







|              | Data  | Label |               |
|--------------|---|-------|---------------|
| Labeled data |    | 0     | Training data |
|              |    | 0     |               |
|              |    | 1     |               |
|              |   | 1     |               |
|              |  | 0     |               |
|              |  | 1     | Testing data  |
|              |  | 0     |               |

Rather than just splitting  
once, split multiple times







# Repeated experimentation

Training data







Data Label

|   |   |
|---|---|
|    | 0 |
|    | 0 |
|    | 1 |
|   | 1 |
|  | 0 |
|  | 1 |

Data Label

|   |   |
|---|---|
|    | 0 |
|    | 0 |
|    | 1 |
|   | 1 |
|  | 0 |
|  | 1 |

Data Label

|   |   |
|---|---|
|    | 0 |
|    | 0 |
|    | 1 |
|   | 1 |
|  | 0 |
|  | 1 |

...

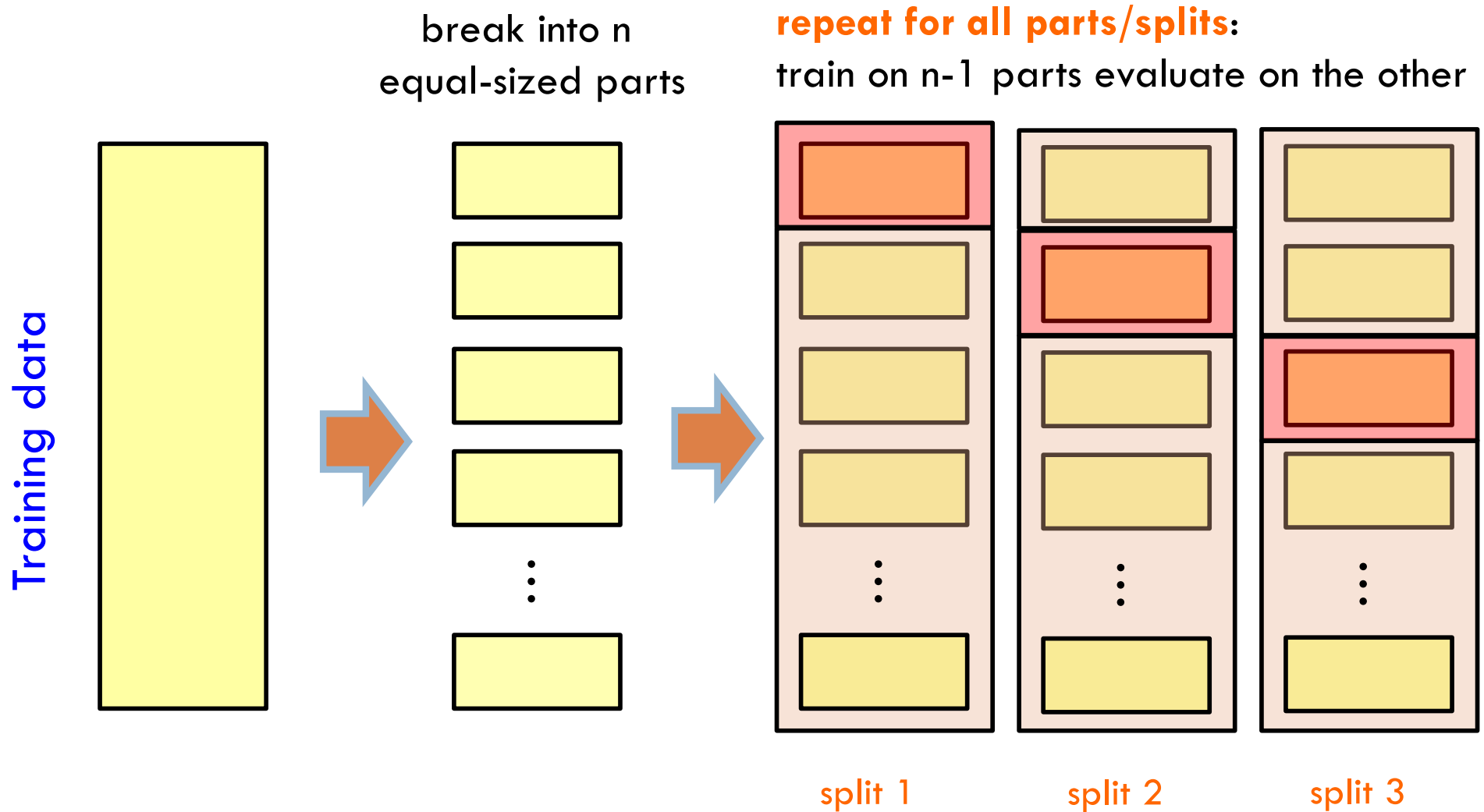


= train

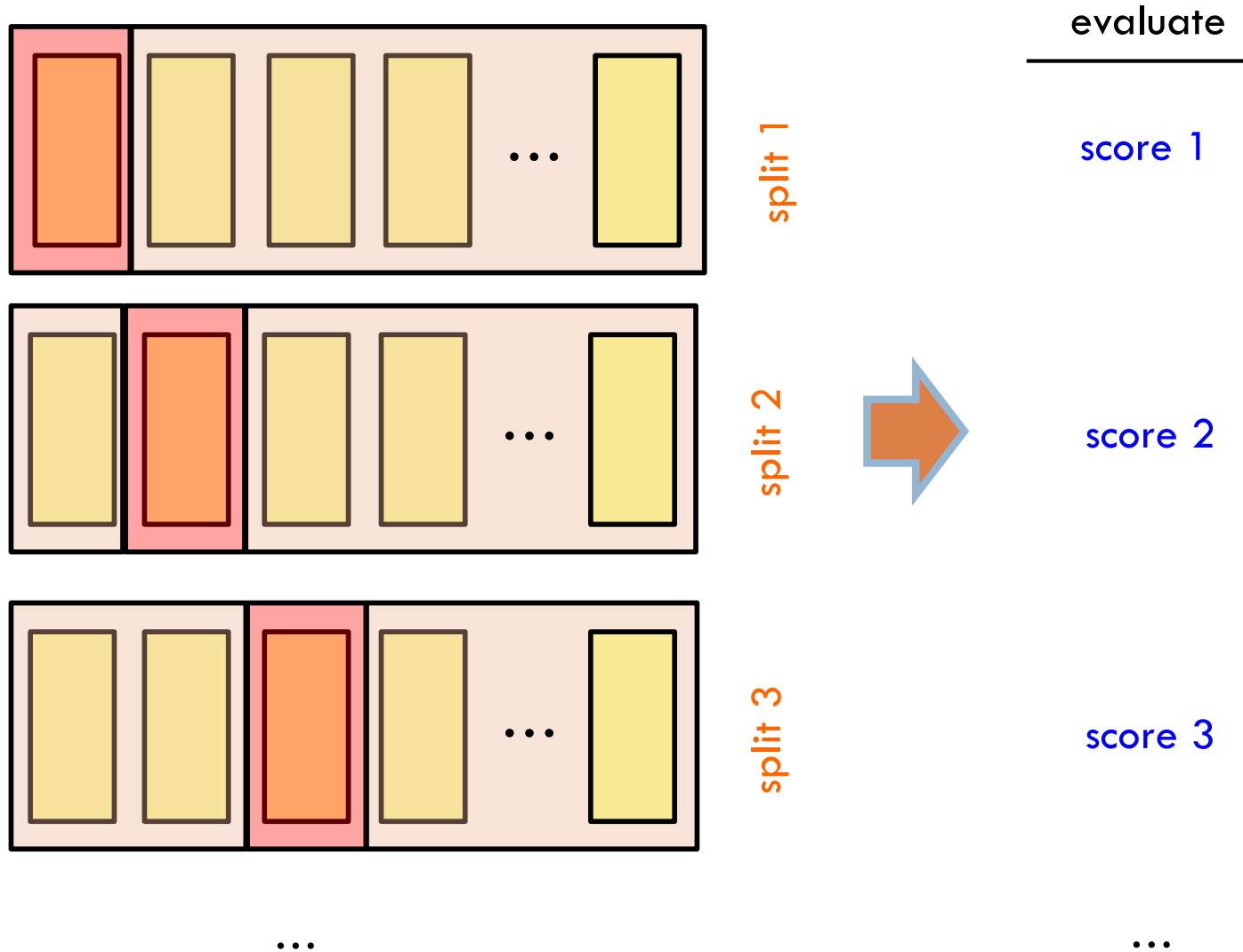


= development

# n-fold cross validation



# n-fold cross validation



# n-fold cross validation



better utilization of labeled data

more robust: doesn'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  $n = \text{number of examples}$

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.?)



# Random splits



$n$ -fold cross-validation explicitly divides the data into  $n$  partitions and each data point gets used exactly once for testing

Another approach is do some number of random X/Y splits (like we did on Assignment 2)

Pros/cons?

# Random splits



Lose the guarantee that all examples are used once for testing

Can be slower (if we do more splits)

Can allow for more samples

Both are fine approaches!

# Comparing systems



| 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      |

| 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      |

| 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: 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        |
| <b>average:</b> | <b>85</b> | <b>85</b> |

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        |
| <b>average:</b> | <b>82</b> | <b>85</b> |

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        |
| <b>average:</b> | <b>82</b> | <b>85</b> |

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        |
| <b>average:</b> | <b>82</b> | <b>85</b> |

| 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        |
| <b>average:</b> | <b>82</b> | <b>85</b> |

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         |
| <b>average:</b> | <b>82</b>  | <b>85</b>  |
| <b>std dev</b>  | <b>2.3</b> | <b>1.7</b> |

| 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         |
| <b>average:</b> | <b>82</b>  | <b>85</b>  |
| <b>std dev</b>  | <b>5.9</b> | <b>3.9</b> |

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         |
| <b>average:</b> | <b>83</b>  | <b>85</b>  |
| <b>std dev</b>  | <b>4.9</b> | <b>4.7</b> |

Is model 2 better  
than model 1?

# Comparing systems: sample 4

| split           | model 1    | model 2    | model 2 –<br>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                    |
| <b>average:</b> | <b>83</b>  | <b>85</b>  |                      |
| <b>std dev</b>  | <b>4.9</b> | <b>4.7</b> |                      |

Is model 2 better  
than model 1?

# Comparing systems: sample 4

| split           | model 1    | model 2    | model 2 –<br>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                    |
| <b>average:</b> | <b>83</b>  | <b>85</b>  |                      |
| <b>std dev</b>  | <b>4.9</b> | <b>4.7</b> |                      |

Model 2 is ALWAYS  
better

# Comparing systems: sample 4

| split           | model 1    | model 2    | model 2 –<br>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                    |
| <b>average:</b> | <b>83</b>  | <b>85</b>  |                      |
| <b>std dev</b>  | <b>4.9</b> | <b>4.7</b> |                      |

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

# Statistical tests



## Setup:

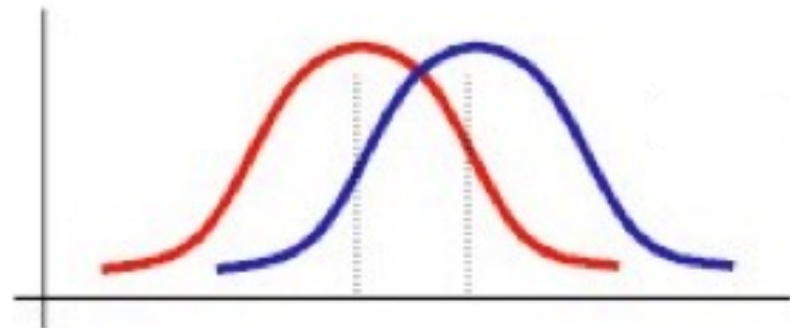
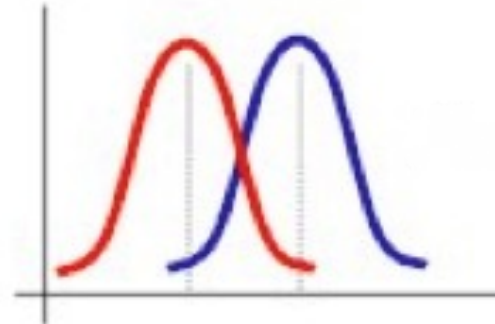
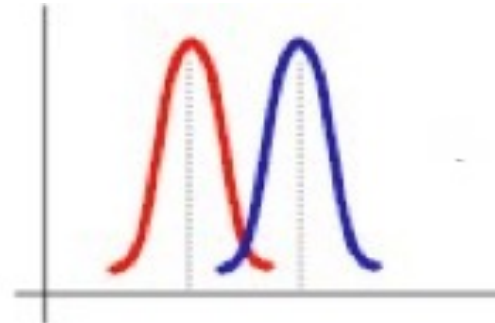
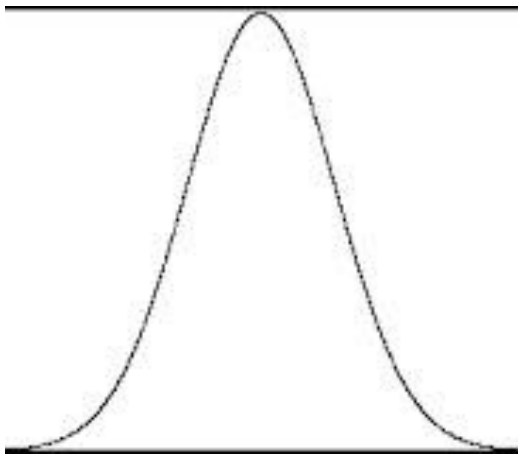
- ▣ Assume some default hypothesis about the data that you'd like to *disprove*, called the **null hypothesis**
- ▣ e.g. model 1 and model 2 are not statistically different in performance

## 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 "paired 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](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        |
| <b>average:</b> | <b>85</b> | <b>85</b> |

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        |
| <b>average:</b> | <b>82</b> | <b>85</b> |

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        |
| <b>average:</b> | <b>82</b> | <b>85</b> |

Is model 2 better  
than model 1?

They are the same with:  
 $p = 0.007$

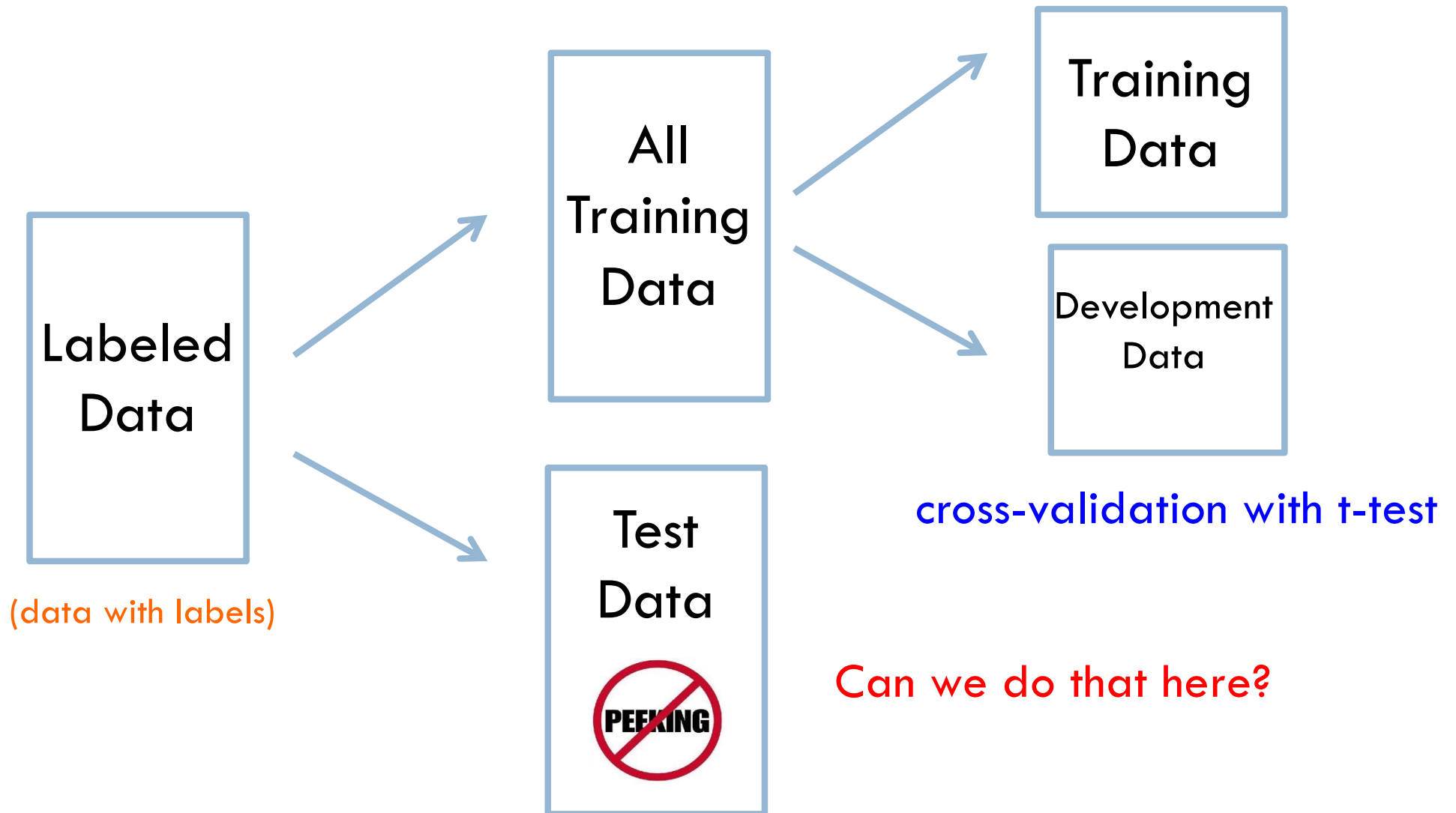
# Comparing systems: sample 4

| 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        |
| <b>average:</b> | <b>83</b> | <b>85</b> |

Is model 2 better  
than model 1?

They are the same with:  
 $p = 0.001$

# Statistical tests on test data



# Bootstrap resampling



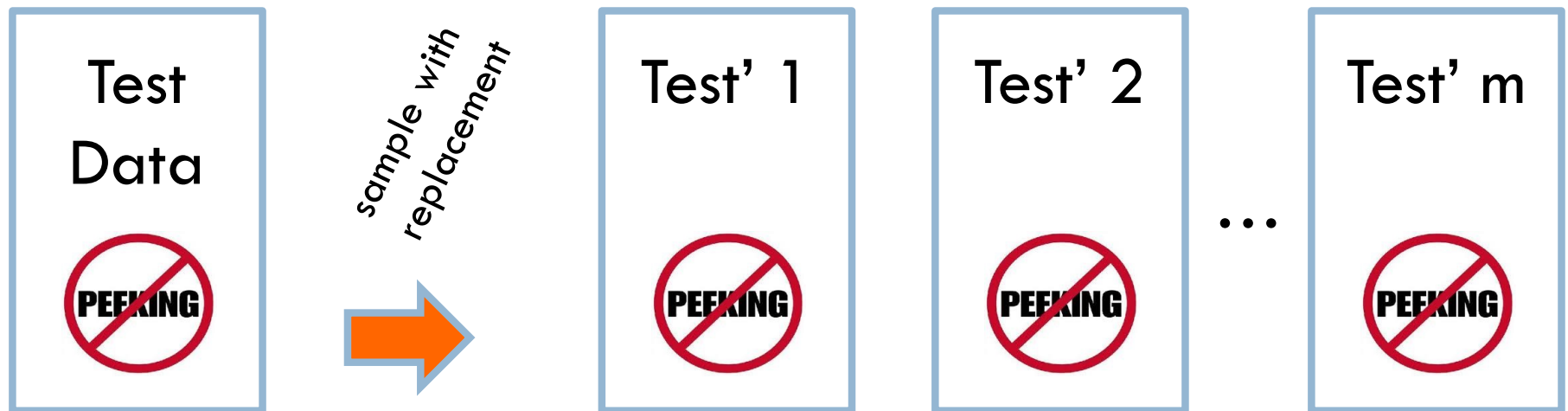
test set  $t$  with  $n$  samples

do  $m$  times:

- sample  $n$  examples **with replacement** from the test set to create a new test set  $t'$
- evaluate model(s) on  $t'$

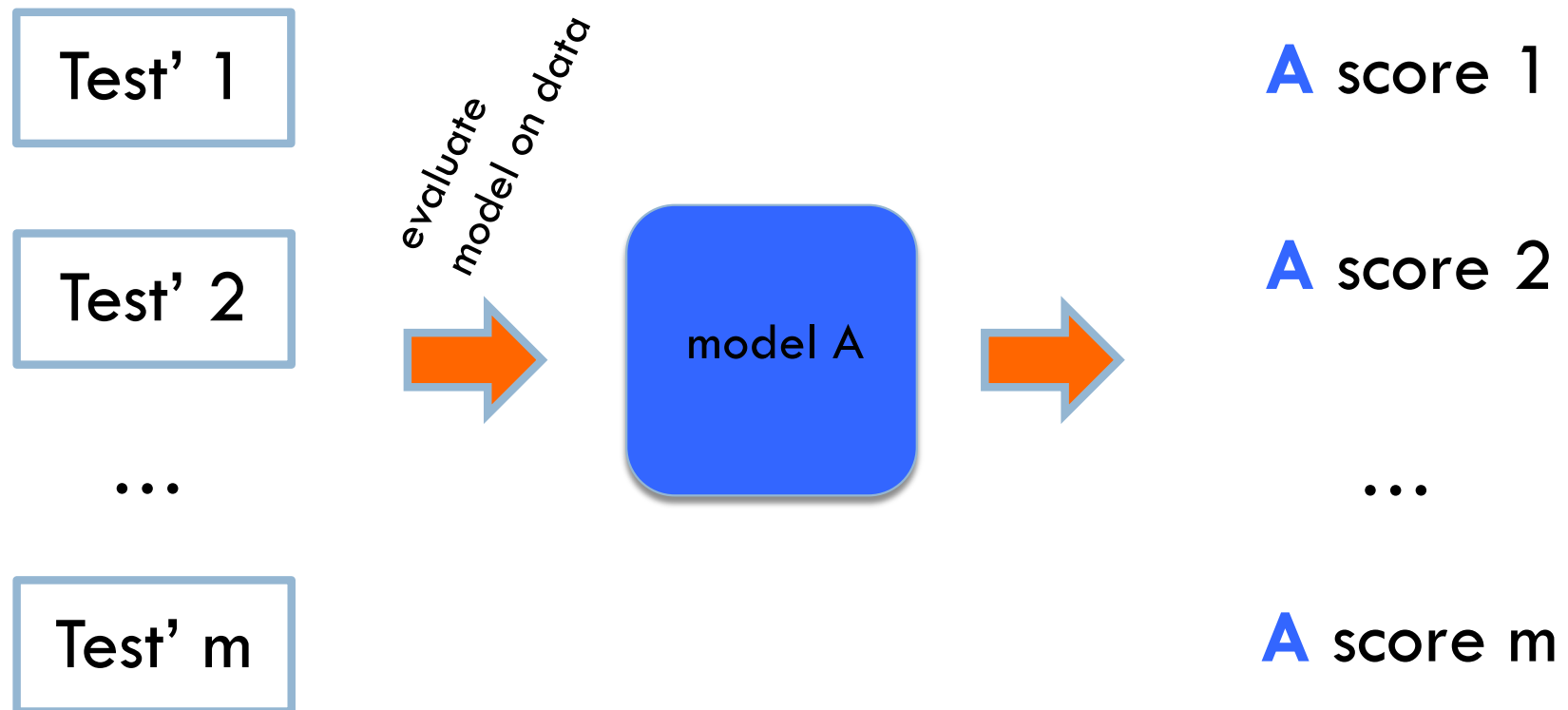
calculate t-test (or other statistical test) on the collection of  $m$  results

# Bootstrap resampling

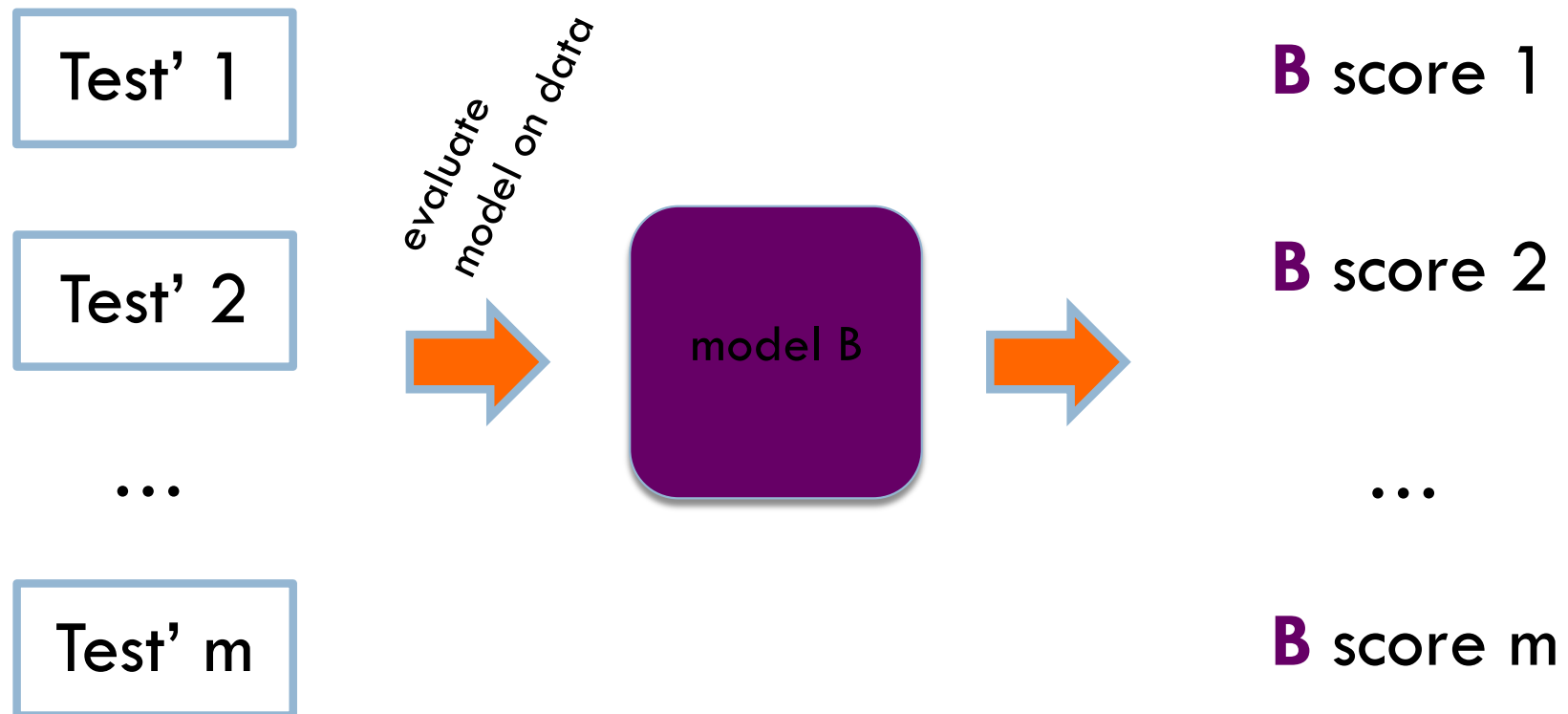




# Bootstrap resampling



# Bootstrap resampling



# Bootstrap resampling



**A** score 1

**B** score 1

**A** score 2

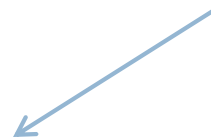
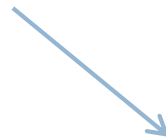
**B** score 2

...

...

**A** score m

**B** score m



paired t-test (or other analysis)

# 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  $p = 0.05$  (or even lower)

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