Regularization
Cross Validation

Regularization – Cross Validation

- Cross validation is a more advanced set of methods for splitting data into training and testing sets.
- Cross Validation Relevant Reading:
  - Section 5.1 of ISLR

Regularization – Cross Validation

- We understand the intuition behind performing a train test split, we want to fairly evaluate our model's performance on unseen data.
- Unfortunately this means we are not able to tune hyperparameters to the **entire** dataset.

Regularization – Cross Validation

- Is there a way we can achieve the following:
  - Train on all the data
  - Evaluate on all the data
- While it sounds impossible, we can achieve this with cross validation!
- Let's have an overview of the concept...

Regularization – Cross Validation

- Imagine our data set:

	X		y
Area m^2	Bedrooms	Bathrooms	Price
200	3	2	$500,000
190	2	1	$450,000
230	3	3	$650,000
180	1	1	$400,000
210	2	2	$550,000

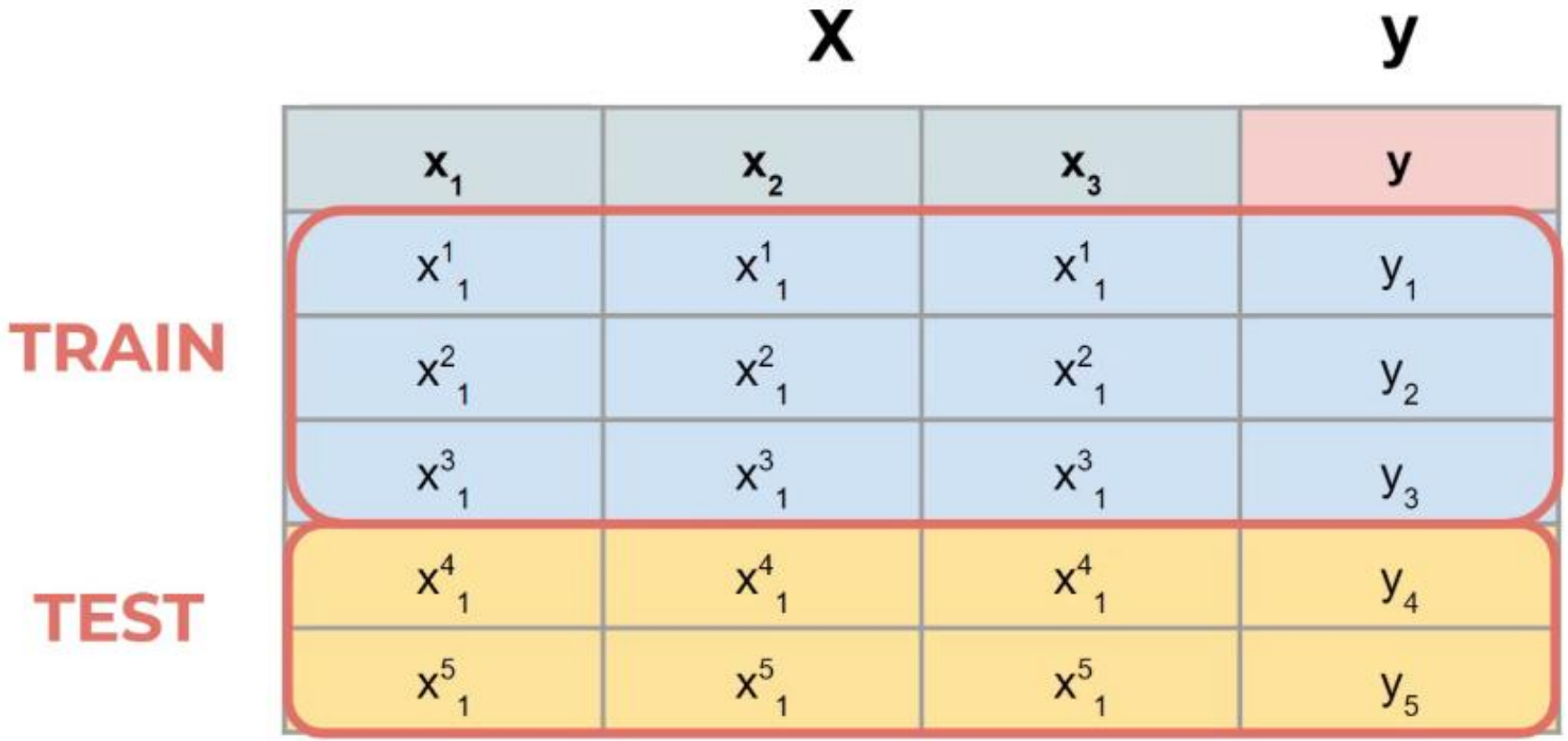
Regularization – Cross Validation

- Let's convert this data into colored blocks for cross-validation

X			y
Area m^2	Bedrooms	Bathrooms	Price
200	3	2	$500,000
190	2	1	$450,000
230	3	3	$650,000
180	1	1	$400,000
210	2	2	$550,000

Regularization – Cross Validation

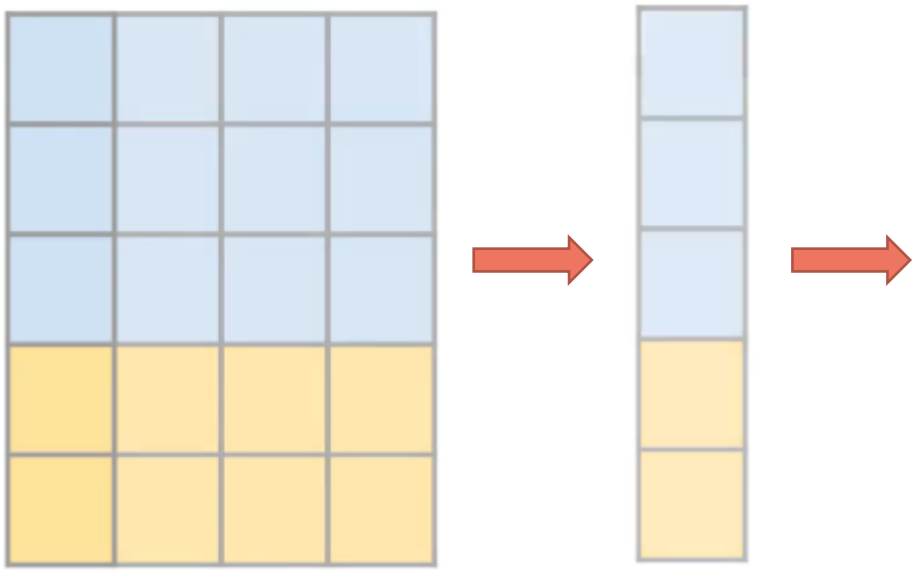
- Color based off train vs. test set.



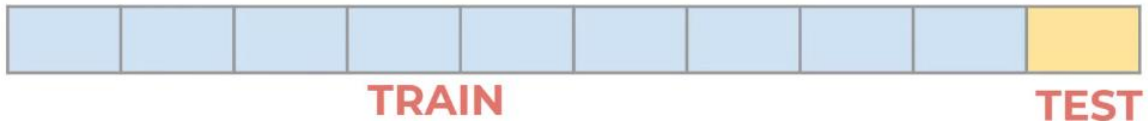
x_1	x_2	x_3	y
x^1_1	x^1_1	x^1_1	y_1
x^2_1	x^2_1	x^2_1	y_2
x^3_1	x^3_1	x^3_1	y_3
x^4_1	x^4_1	x^4_1	y_4
x^5_1	x^5_1	x^5_1	y_5

Regularization – Cross Validation

- Rotate and resize:

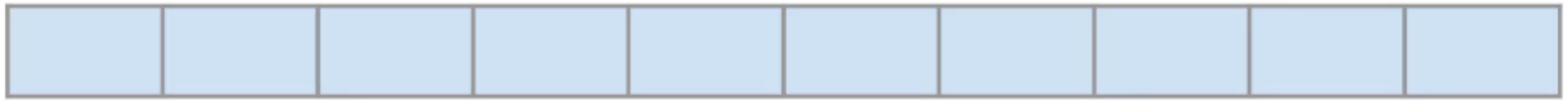


- Now we can represent full data and splits:



Regularization – Cross Validation

- How does cross validation work?

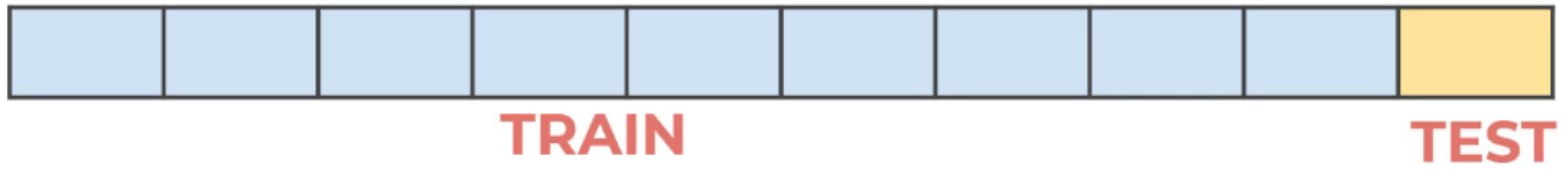


- Split data into K equal parts:

- 1/K left as test set

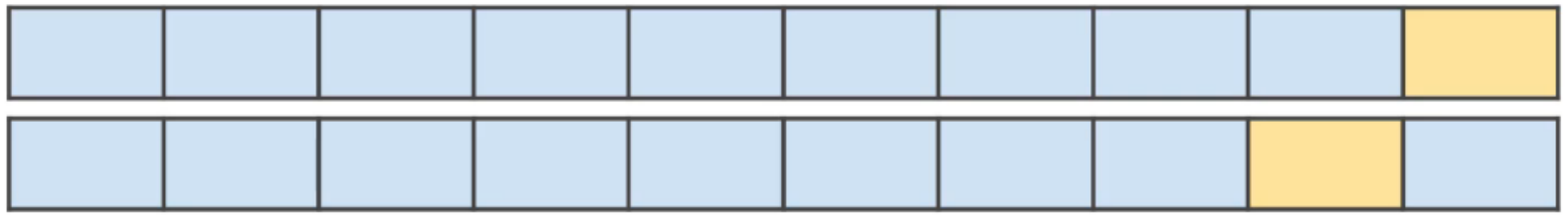
Regularization – Cross Validation

- Train model and get error metric for split:



ERROR 1

- Repeat for another 1/K split

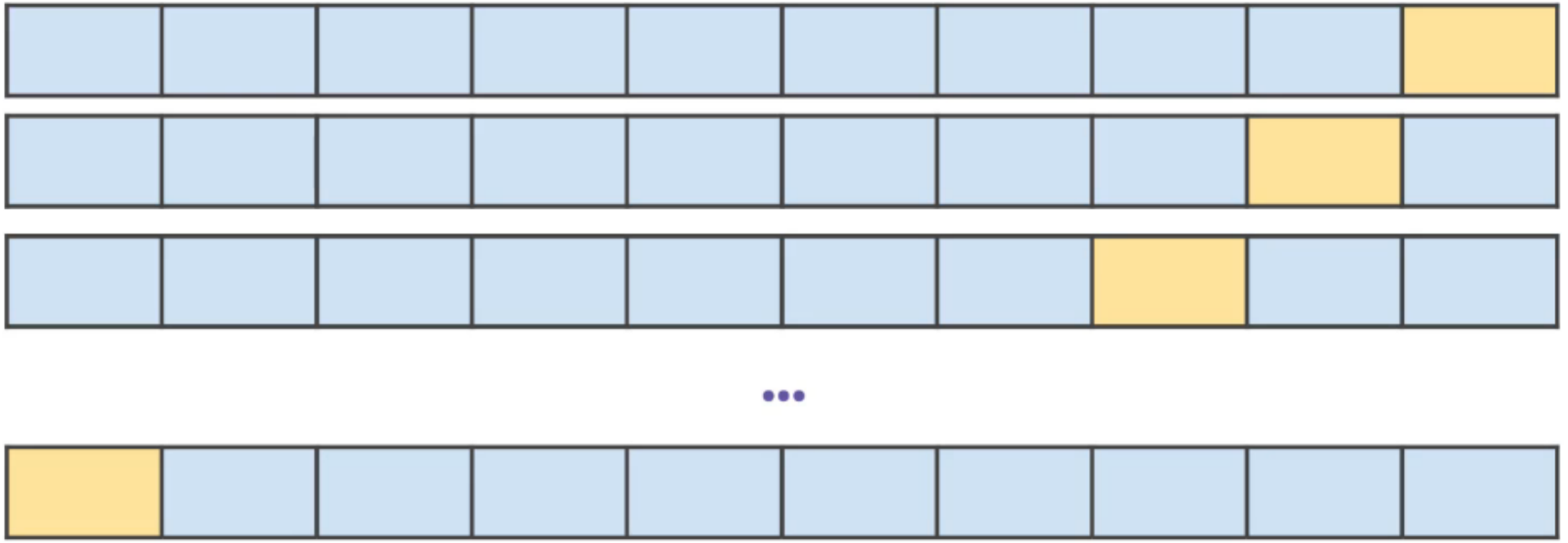


ERROR 1

ERROR 2

Regularization – Cross Validation

- Keep repeating for all possible splits



ERROR 1

ERROR 2

ERROR 3

...

ERROR K

MEAN ERROR

Regularization – Cross Validation

- We were able to train on all data **and** evaluate on all data!
- We get a better sense of true performance across multiple potential splits.
- What is the cost of this?
  - We have to repeat computations K number of times!

Regularization – Cross Validation

- This is known as K-fold cross-validation.
- Common choice for K is 10 so each test set is 10% of your total data.
- Largest K possible would be K equal to the number of number of rows.
  - This is known as **leave one out** cross validation.
  - Computationally expensive!

Regularization – Cross Validation

- One consideration to note with K-fold cross validation and a standard train test split is fairly tuning hyperparameters.
- If we tune hyperparameters to test data performance, are we ever fairly getting performance metrics?

Regularization – Cross Validation

- How can we understand how the model behaves for data that is has not seen **and** not been influenced by for hyperparameter tuning?
- For this we can use a **hold out** test set.
- Let's explore what this looks like...

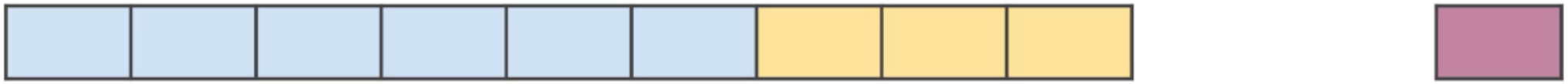
Regularization – Cross Validation- hold out Test Set

- Start with entire data set:

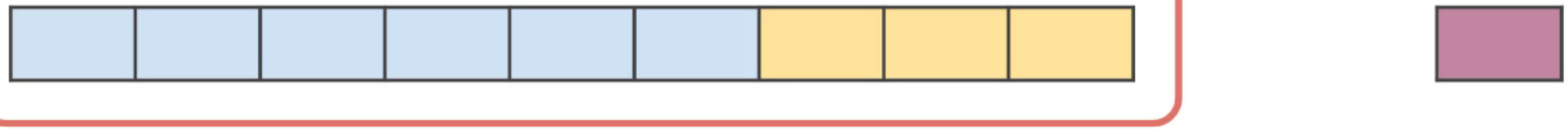
- Remove a hold out test set

Regularization – Cross Validation- hold out Test Set

- Perform “classic” train test split:

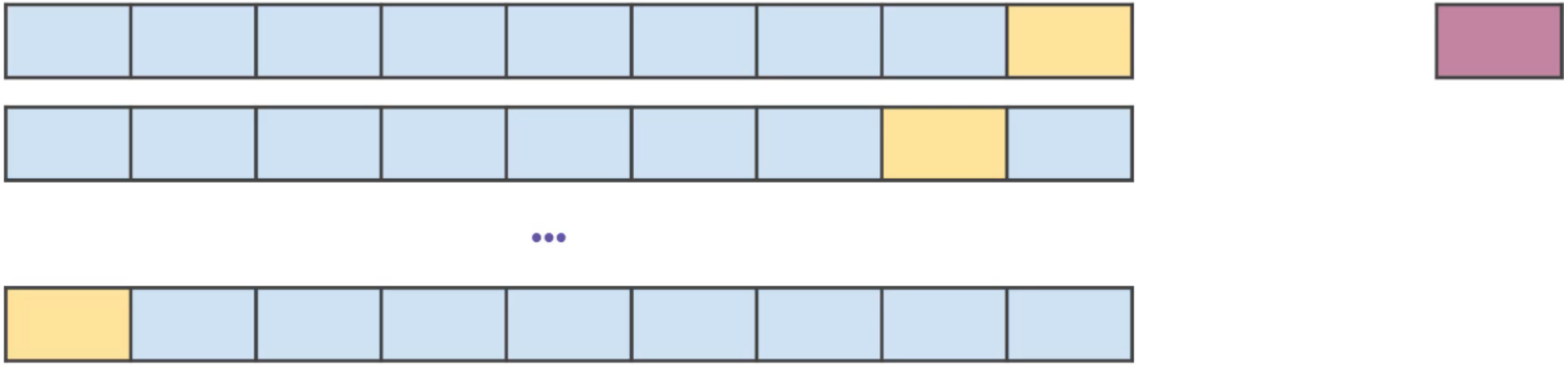


- Train and tune on this data:



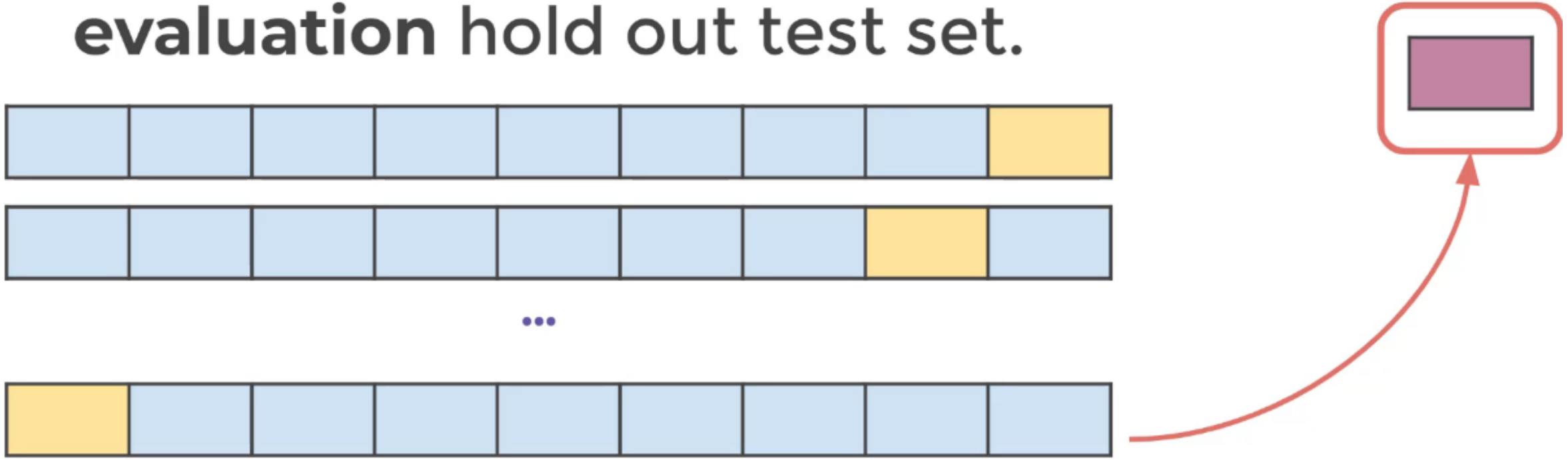
Regularization – Cross Validation- hold out Test Set

- Or K-Fold cross validation



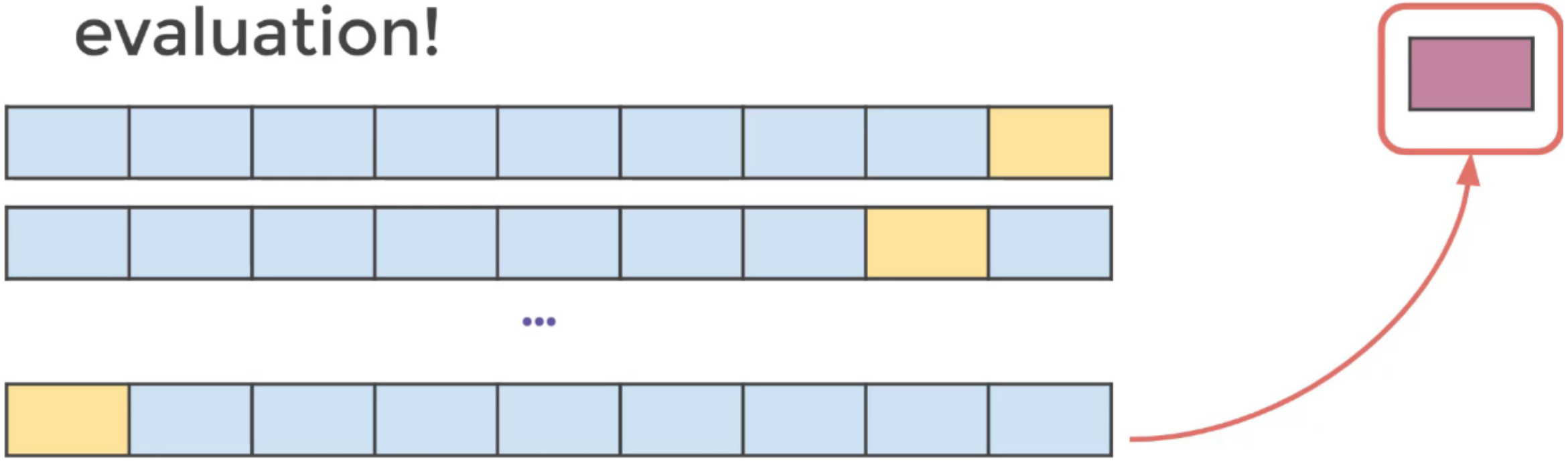
Regularization – Cross Validation- hold out Test Set

- **After** training and tuning perform **final evaluation** hold out test set.



Regularization – Cross Validation- hold out Test Set

- Can **not** tune after this **final** test evaluation!



Regularization – Cross Validation- hold out Test Set

- Train | Validation | Test Split

- Allows us to get a true final performance metric to report.
- No editing model after this!

Regularization – Cross Validation- hold out Test Set

- Train | Validation | Test Split

- Allows us to get a true final performance metric to report.
- No editing model after this!

Regularization – Cross Validation- hold out Test Set

- All these approaches are valid, each situation is unique!
- Keep in mind:
  - Previous modeling work
  - Reporting requirements
  - Fairness of evaluation
  - Context of data and model