

[544] Spark MLlib I

Tyler Caraza-Harter

Outline

ML Review

Training/Predicting APIs

Demos

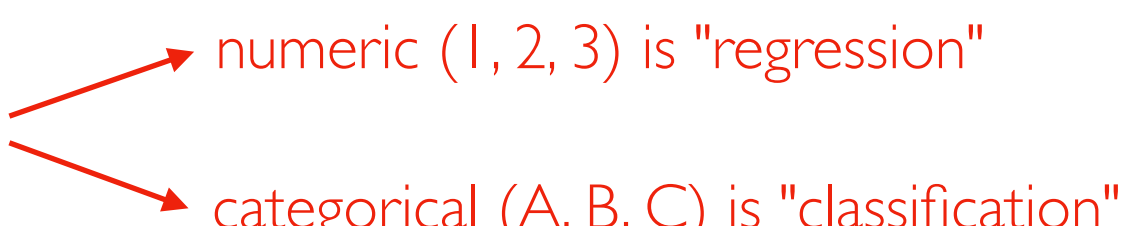
Machine Learning, Major Ideas

Categories of Machine Learning:

- **Reinforcement learning:** agent makes series of actions to maximize reward
- **Unsupervised learning:** looking for generate patterns
- **Supervised learning:** train models to predict unknowns

Models are functions that return predictions:

```
def my_model(some_info) :  
    ...  
    return some_prediction
```



numeric (1, 2, 3) is "regression"

categorical (A, B, C) is "classification"

Example:

```
def weather_forecast(temp_today, temp_yesterday) :  
    ...  
    return temp_tomorrow
```

Learning from Data

	x1	x2	y
0	2	8	5
1	9	2	6
2	4	1	0
3	7	9	7
4	2	2	3
5	3	4	3
6	3	5	9
7	7	1	4
8	6	6	3
9	4	3	?
10	1	2	?
11	2	9	?

- **feature** columns: x1 and x2
- **label** column: y

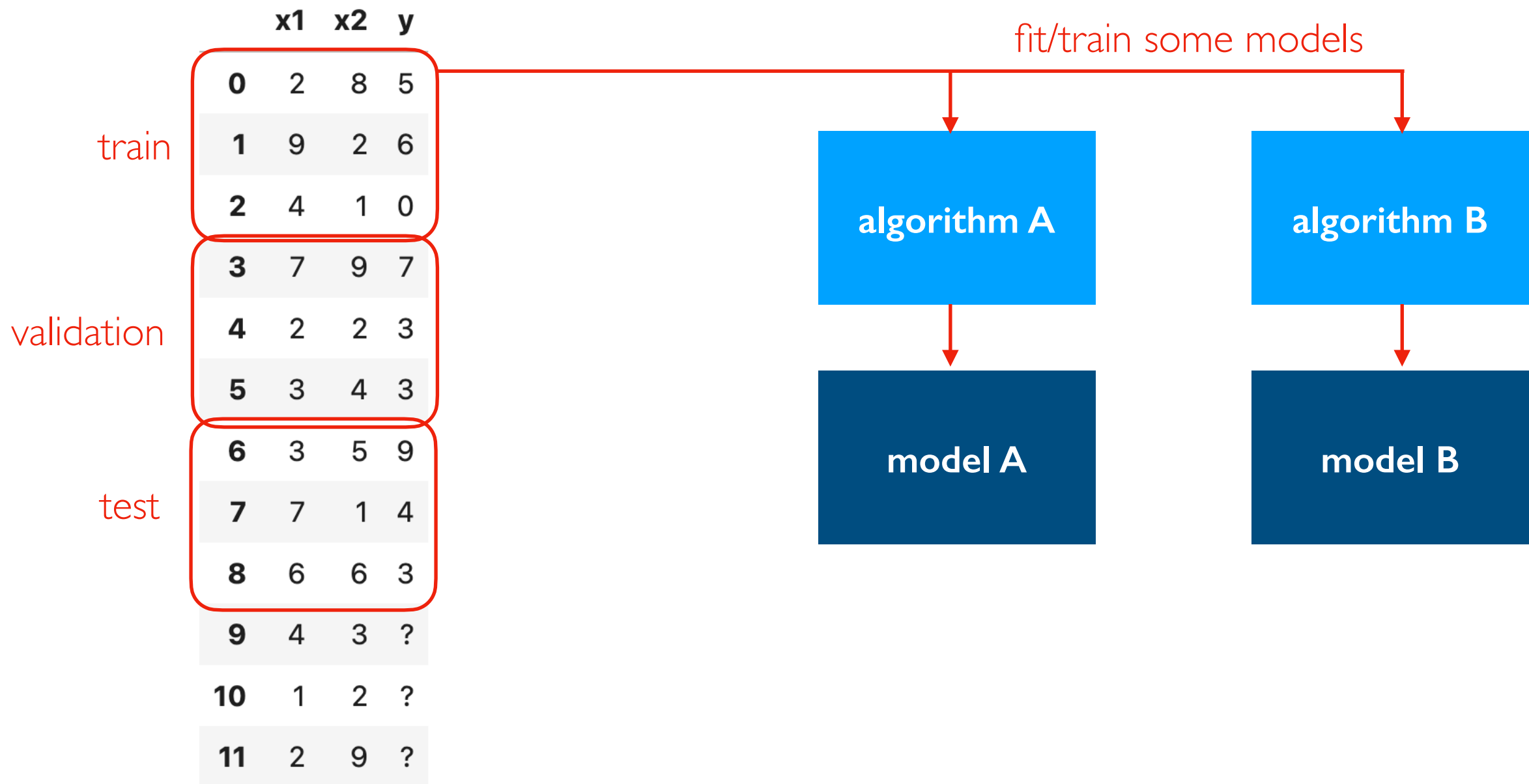
how can the cases where we DO know y help us predict the cases where we do not?

Learning from Data

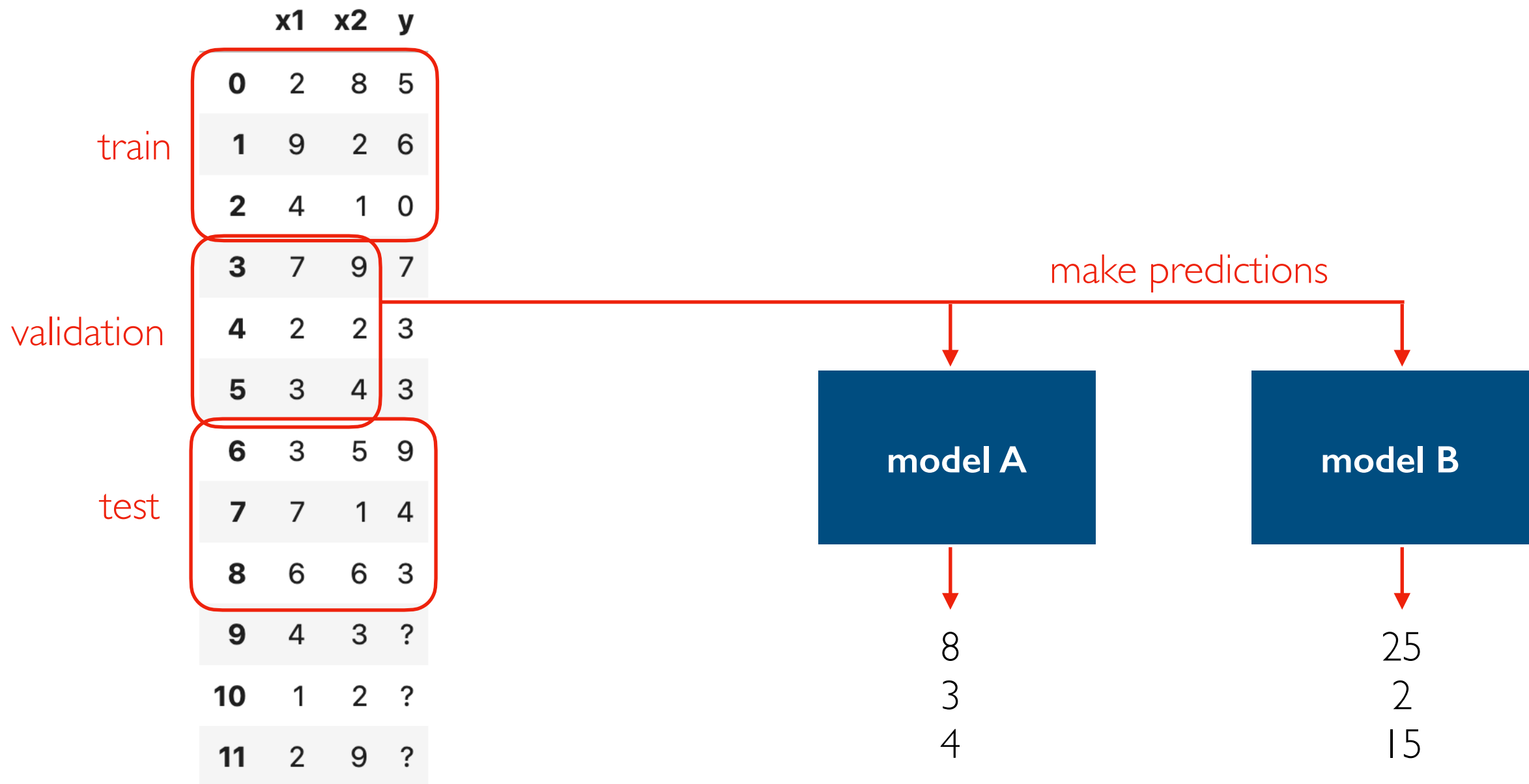
		x1	x2	y
train	0	2	8	5
	1	9	2	6
	2	4	1	0
validation	3	7	9	7
	4	2	2	3
	5	3	4	3
test	6	3	5	9
	7	7	1	4
	8	6	6	3
	9	4	3	?
	10	1	2	?
	11	2	9	?

random split

Learning from Data



Learning from Data



Learning from Data

		x1	x2	y
train	0	2	8	5
	1	9	2	6
	2	4	1	0
validation	3	7	9	7
	4	2	2	3
	5	3	4	3
test	6	3	5	9
	7	7	1	4
	8	6	6	3
	9	4	3	?
	10	1	2	?
	11	2	9	?

which model
predicts better?

model A

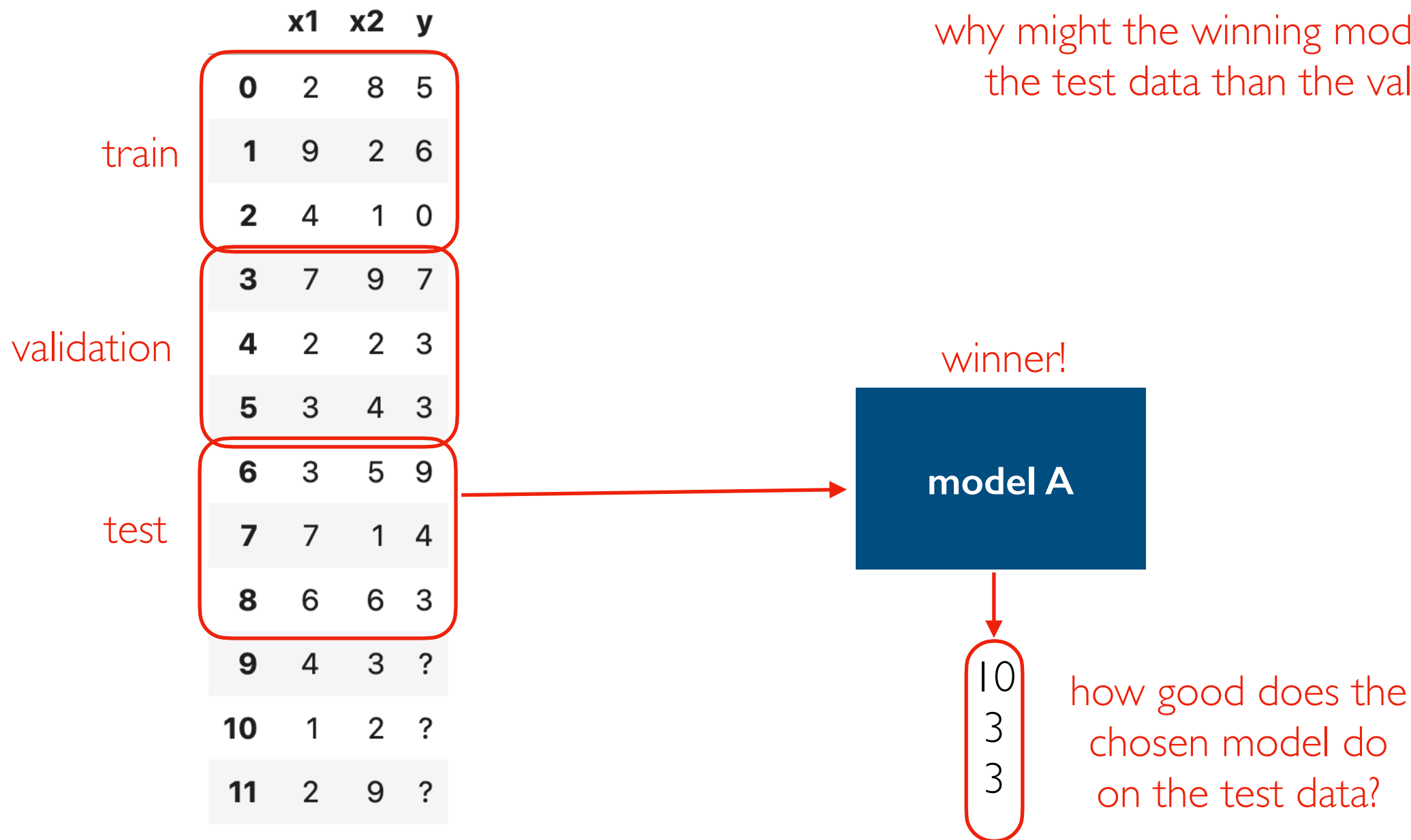
8
3
4

model B

25
2
15

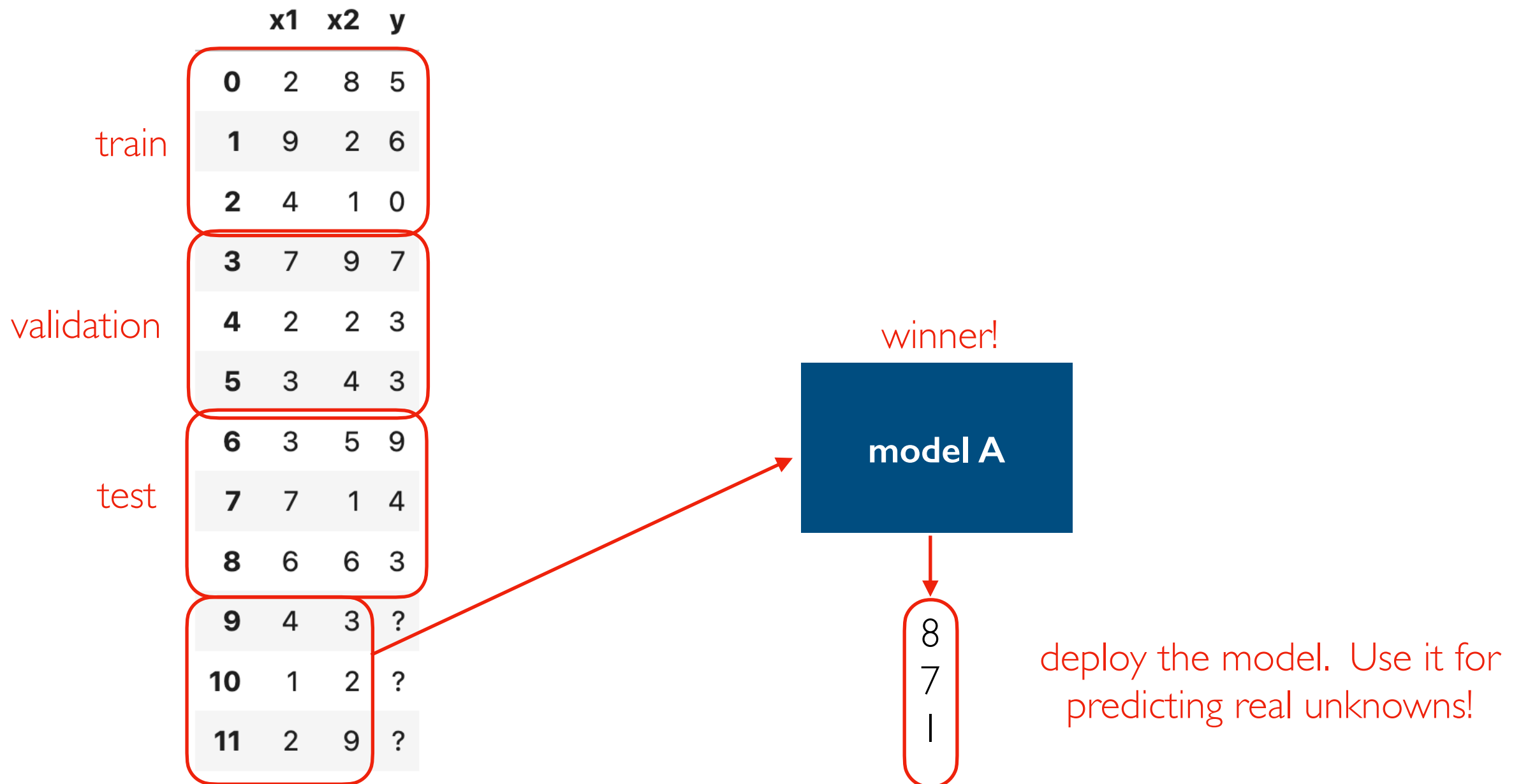
winner!

Learning from Data



models that do good on train data but bad on validation/test data have "overfitted"

Learning from Data



Outline

ML Review

Training/Predicting APIs

- sklearn
- PyTorch
- Spark MLlib

Demos


Training

scikit-learn

```
model = ????  
model.fit(X, y)  
# model parameters can relate X to y
```

pytorch

```
model = ????  
# TODO: optimizer, loss function  
# training loop  
for epoch in range(???):  
    for X, y in ????:  
        ...  
# model parameters can relate X to y
```

- 
- models are **mutable**
 - fitting sets/improves parameters

Spark MLlib

```
unfit_model = ????  
fit_model = unfit_model.fit(df)  
# fit_model params can relate x to y
```

- models are **immutable**
- fitting returns new model object

Predicting

scikit-learn

```
y = model.predict(X)
```

pytorch

```
y = model(X)
```

Spark MLlib

```
df2 = fit_model.transform(df)
```

Data

scikit-learn

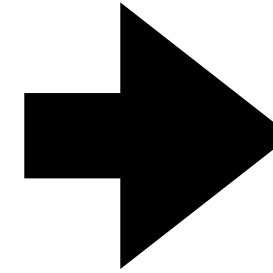
```
y = model.predict(X)
```

pytorch

```
y = model(X)
```

X (features)

y (label)



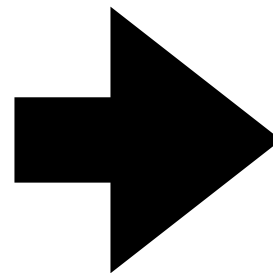
Spark MLlib

```
df2 = fit_model.transform(df)
```

X (features)

df

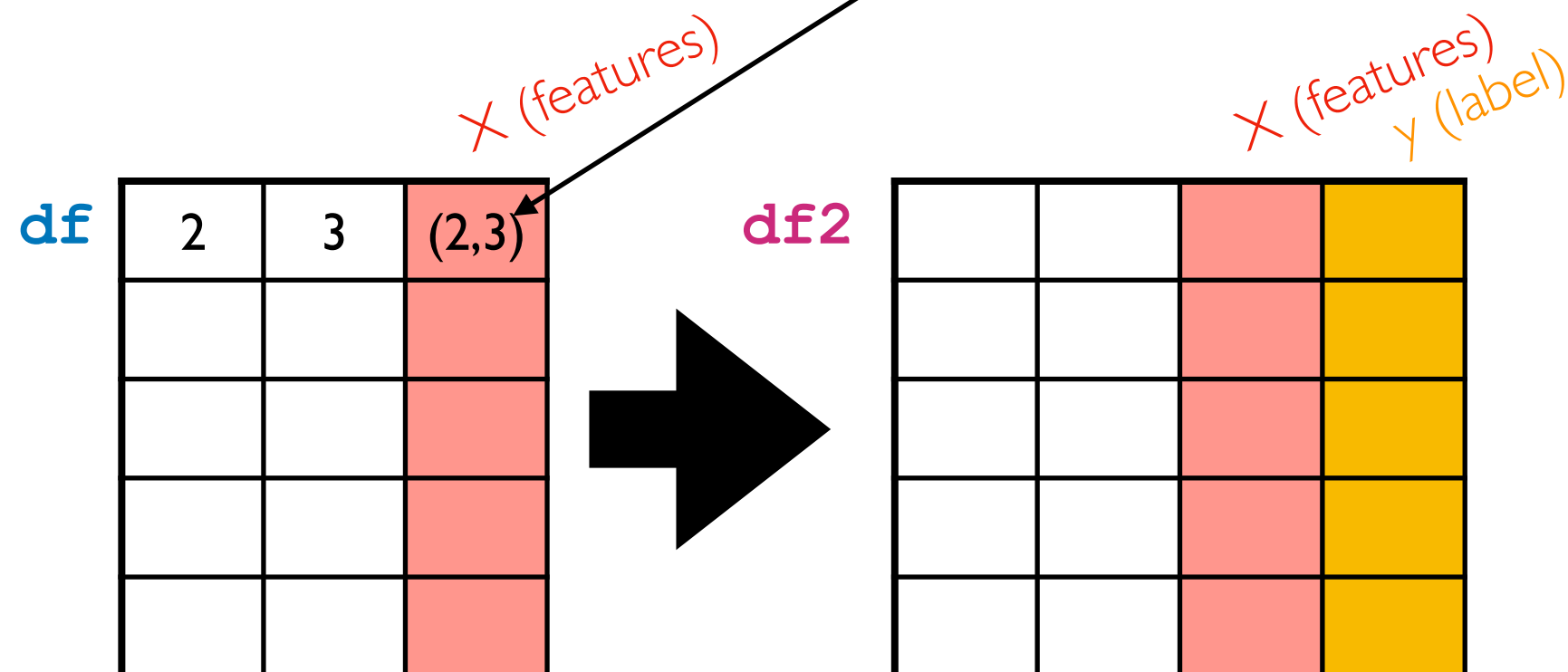
df2



X (features)
y (label)

Features Column

- we only get one features column
- it usually contains vectors
- those vectors typically contain values from other columns
- example:
(2, 3)



Terminology

Spark and scikit-learn use many of the same terms, with very different meaning.

Transformer (scikit-learn)

- has .transform method
- takes a DataFrame, returns a different DataFrame
- used as **preprocessing step** for a model

Transformer (Spark)

- has .transform method
- takes a DataFrame, returns original **with 1 or more additional columns**
- **a fitted model is a transformer** that adds a prediction column

Estimator (scikit-learn)

- has .fit and .predict methods
- .fit **modifies** the object
- makes predictions after learning params

Estimator (Spark)

- has .fit method that **returns new object**
- an **unfitted model** is an estimator; calling .fit returns a **fitted model** (a transformer)

Pipeline

Both scikit-learn and Spark: a pipeline is a series of stages (transformers/estimators). fit/transform/etc. are called as appropriate on each stage.

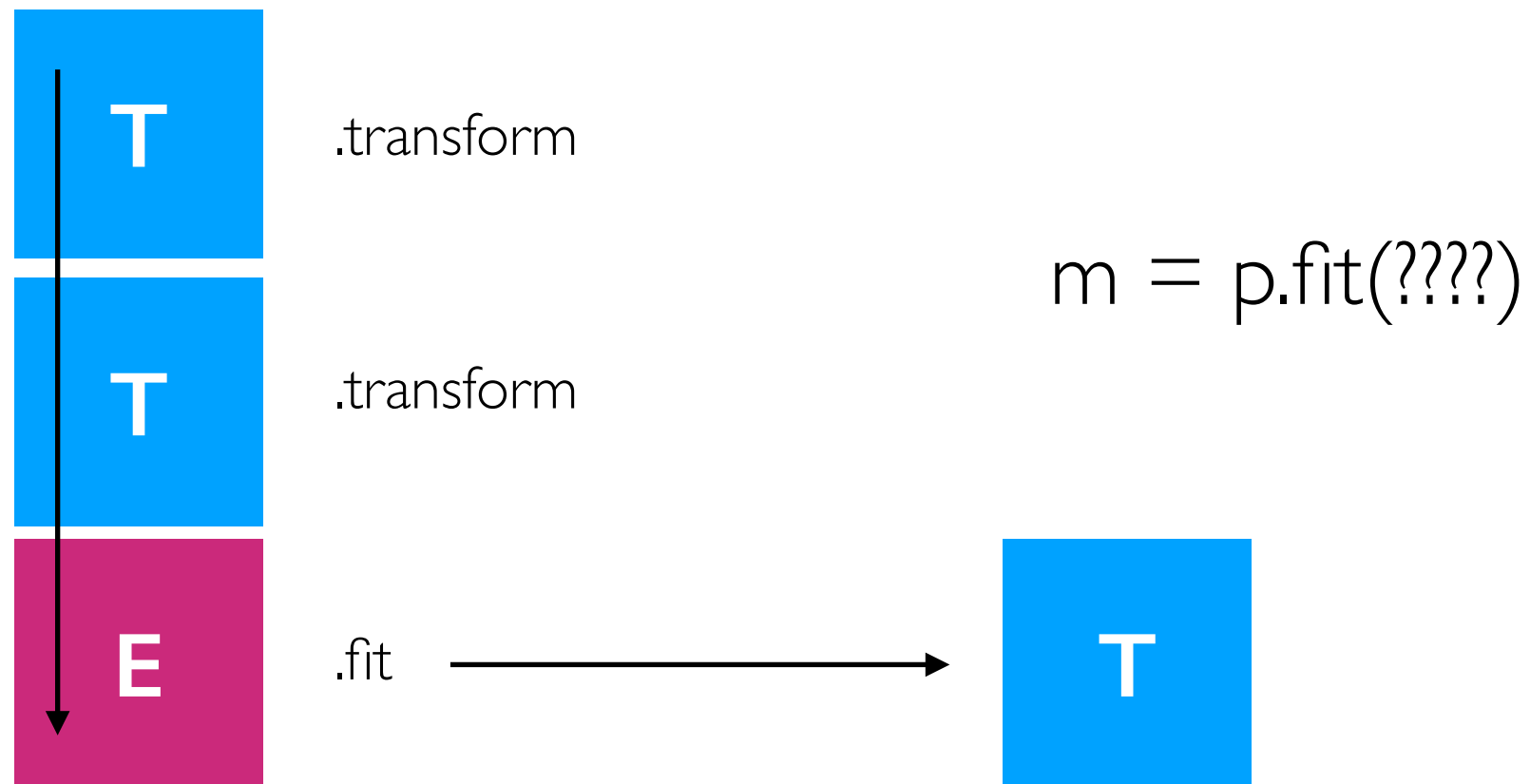
Pipeline (p)



Pipeline (Spark Example)

Both scikit-learn and Spark: a pipeline is a series of stages (transformers/estimators). fit/transform/etc. are called as appropriate on each stage.

Pipeline (p)



Pipeline (Spark Example)

Both scikit-learn and Spark: a pipeline is a series of stages (transformers/estimators). fit/transform/etc. are called as appropriate on each stage.

Pipeline (p)



PipelineModel (m)



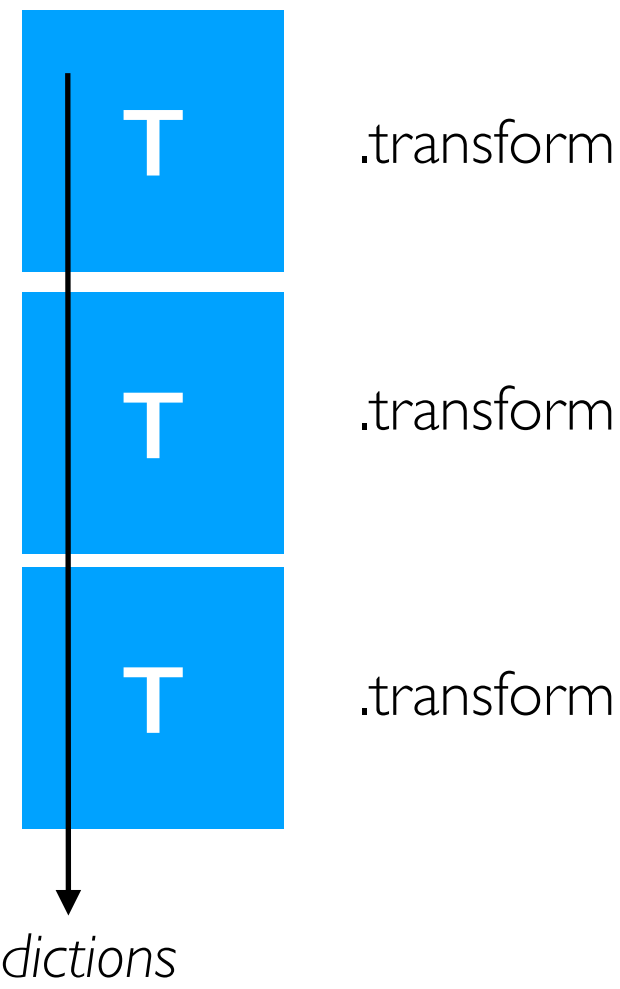
Pipeline (Spark Example)

Both scikit-learn and Spark: a pipeline is a series of stages (transformers/estimators). `fit/transform/etc.` are called as appropriate on each stage.

Pipeline (p)



PipelineModel (m)



`m.transform(????)`

TopHat

Outline

ML Review

Training/Predicting APIs

Demos

Spark mllib packages

- pyspark.mllib -- based on RDDs
- **pyspark.ml** -- based on DataFrames