Exceptions to the Rule

A gentle introduction to ML

CHIDATA

Recap: Simulation v.s. Model Fitting

$$y_i = f_\theta(x_i, \epsilon_i)$$

Simulation: Observe xi, ei, and theta, **generate yi**

Fitting: Observe xi, yi, **infer theta**

Why not observe epsilon?

"Generic" term that captures process/observation/modeling error

Recap: Bias-Variance Tradeoff

Bias: How closely does your selected model align with the data

Variance: How different predictions are (are nearby predictions similar to each other).

Recap: Bias-Variance Tradeoff

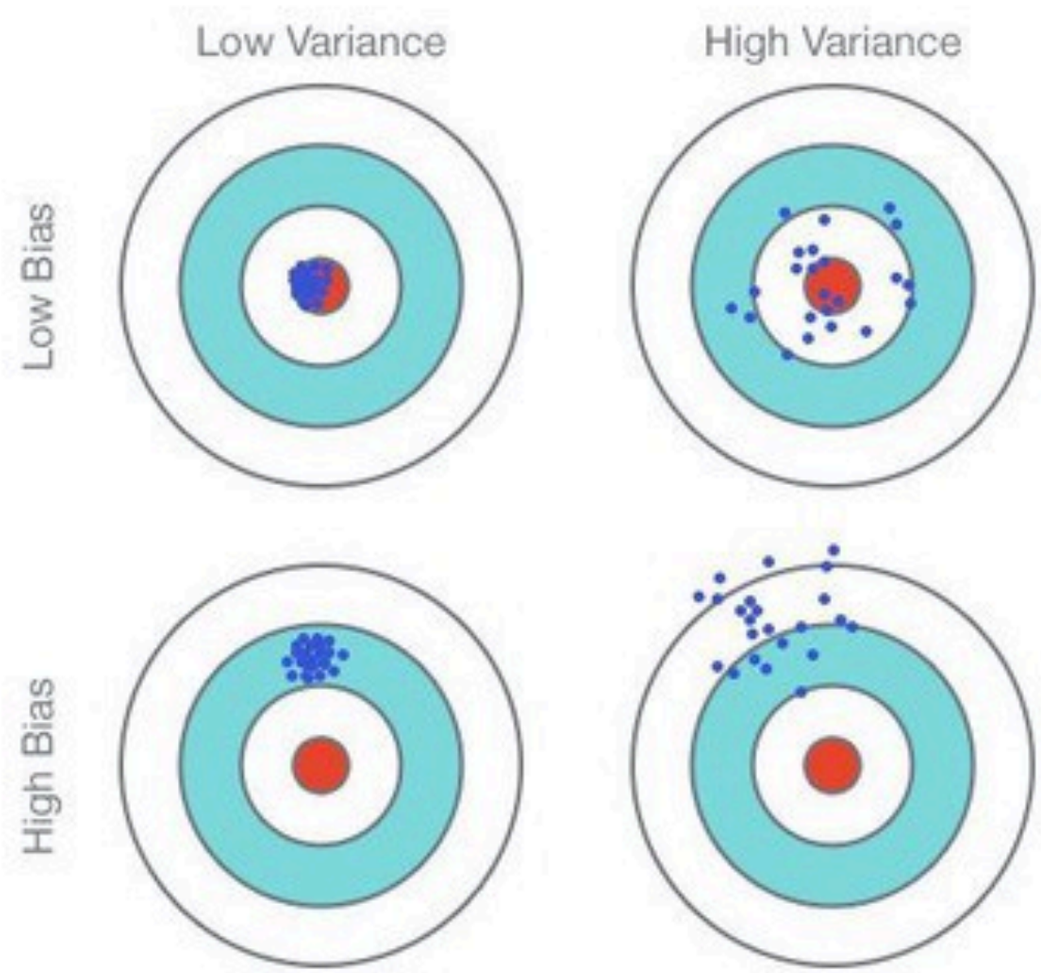
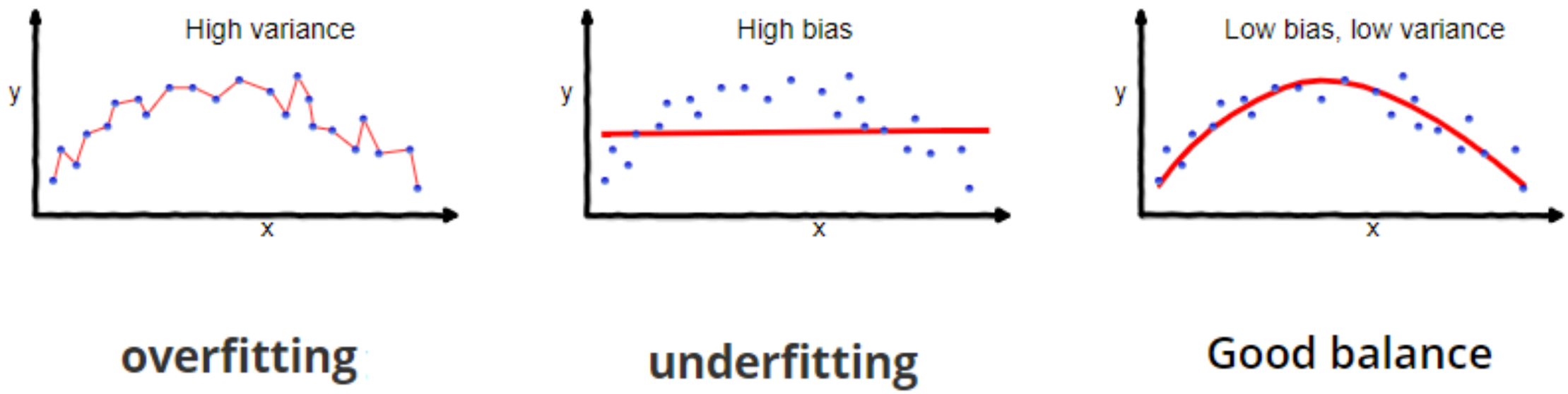


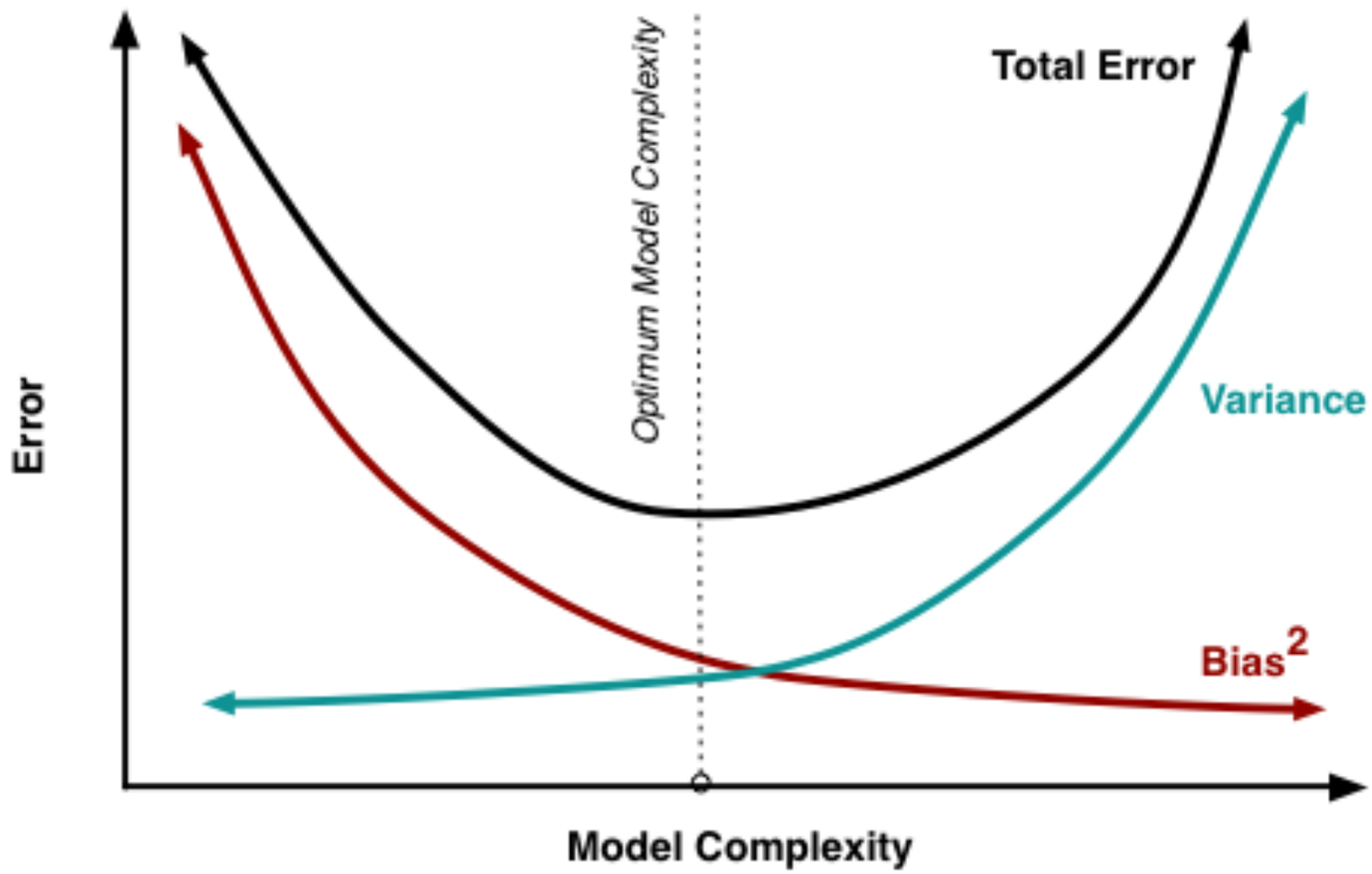
Fig. 1: Graphical Illustration of bias-variance trade-off , Source: Scott Fortmann-Roe., Understanding Bias-Variance Trade-off

Recap: Bias-Variance Tradeoff



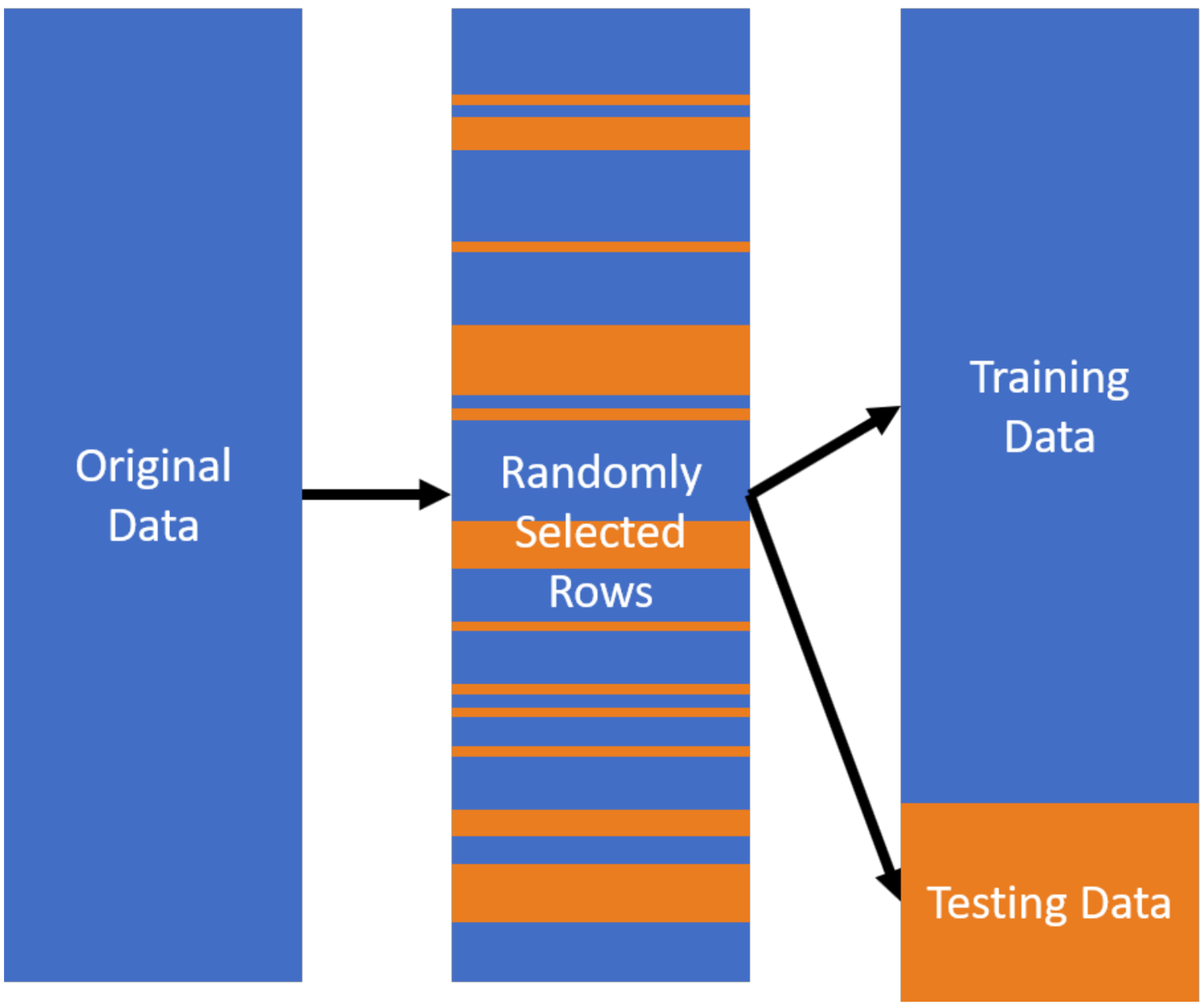
Large classes (more parameters) have **lower bias** but **higher variance**.

Recap: Bias-Variance Tradeoff



Degree of polynomial of the fit

Recap: Training and Testing



Use this data for fitting

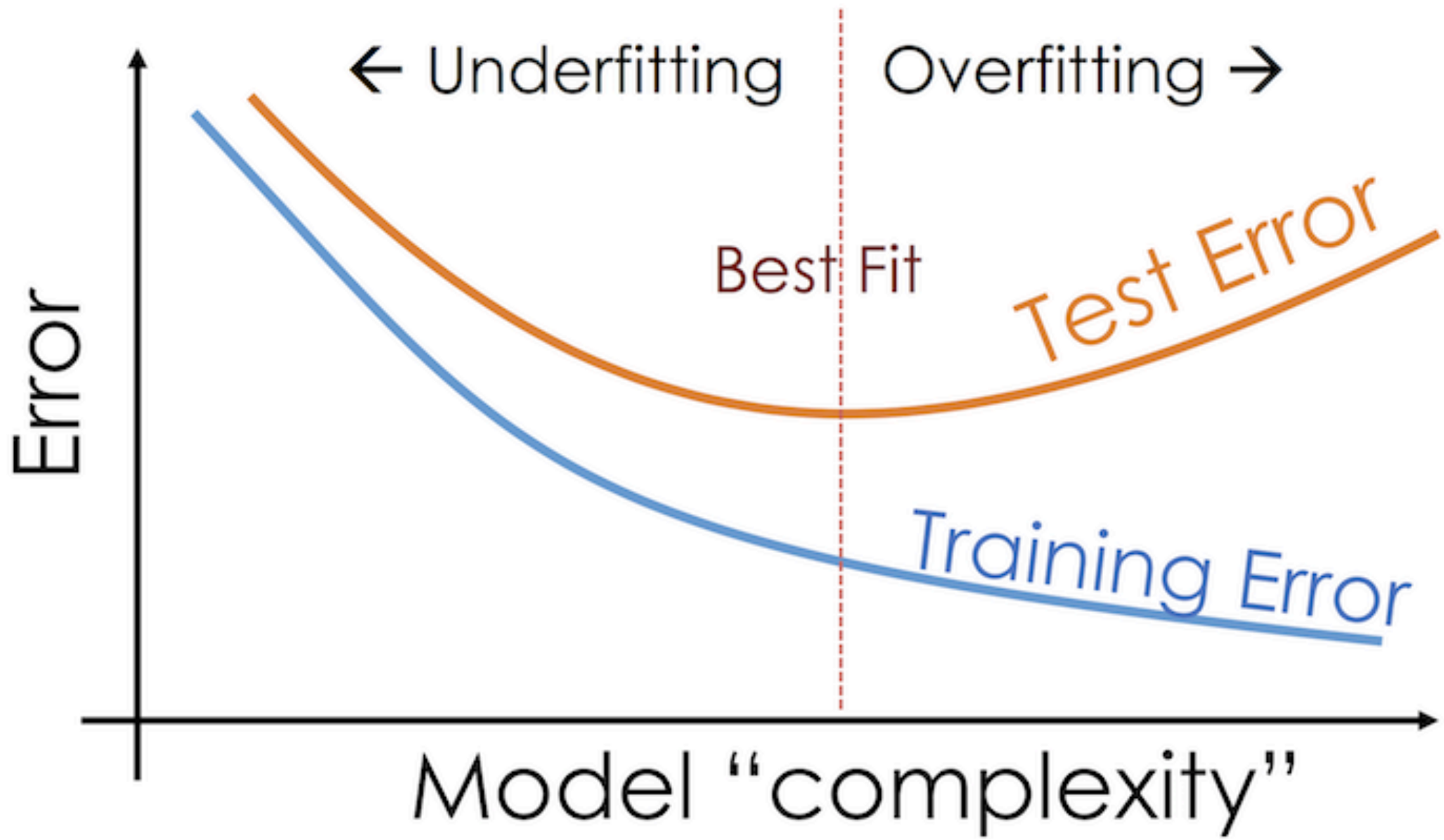
Use this data for evaluation

Recap: Training and Testing

Training error: average loss over the training set

Testing error: average loss over the testing set

Recap: Training and Testing



Another Perspective

Another Perspective

Got Sick?	Will I Get Sick?
 Y	
 N	
 Y	
 N	

What patterns do we see in this data?

Another Perspective

Got Sick?	Will I Get Sick?
 Y	
 N	
 Y	
 N	

Both “sick” fruits are round

Another Perspective

Got Sick?	Will I Get Sick?
 Y	
 N	
 Y	
 N	

Both “not sick” fruits are yellow

Another Perspective

Got Sick?	Will I Get Sick?
 Y	
 N	
 Y	
 N	

What about (is_oval AND not yellow)?

Overfitting

Rules are too precise

	Got Sick?
	Y
	N
	Y
	N

(Purple And Spots)
OR (Orange With Leaves)

Unlikely to be relevant on new fruits

Underfitting

Rules are not expressive enough

	Got Sick?
	Y
	N
	Y
	N

Has Stem?

Unlikely to be relevant on new fruits

Overfitting/Underfitting

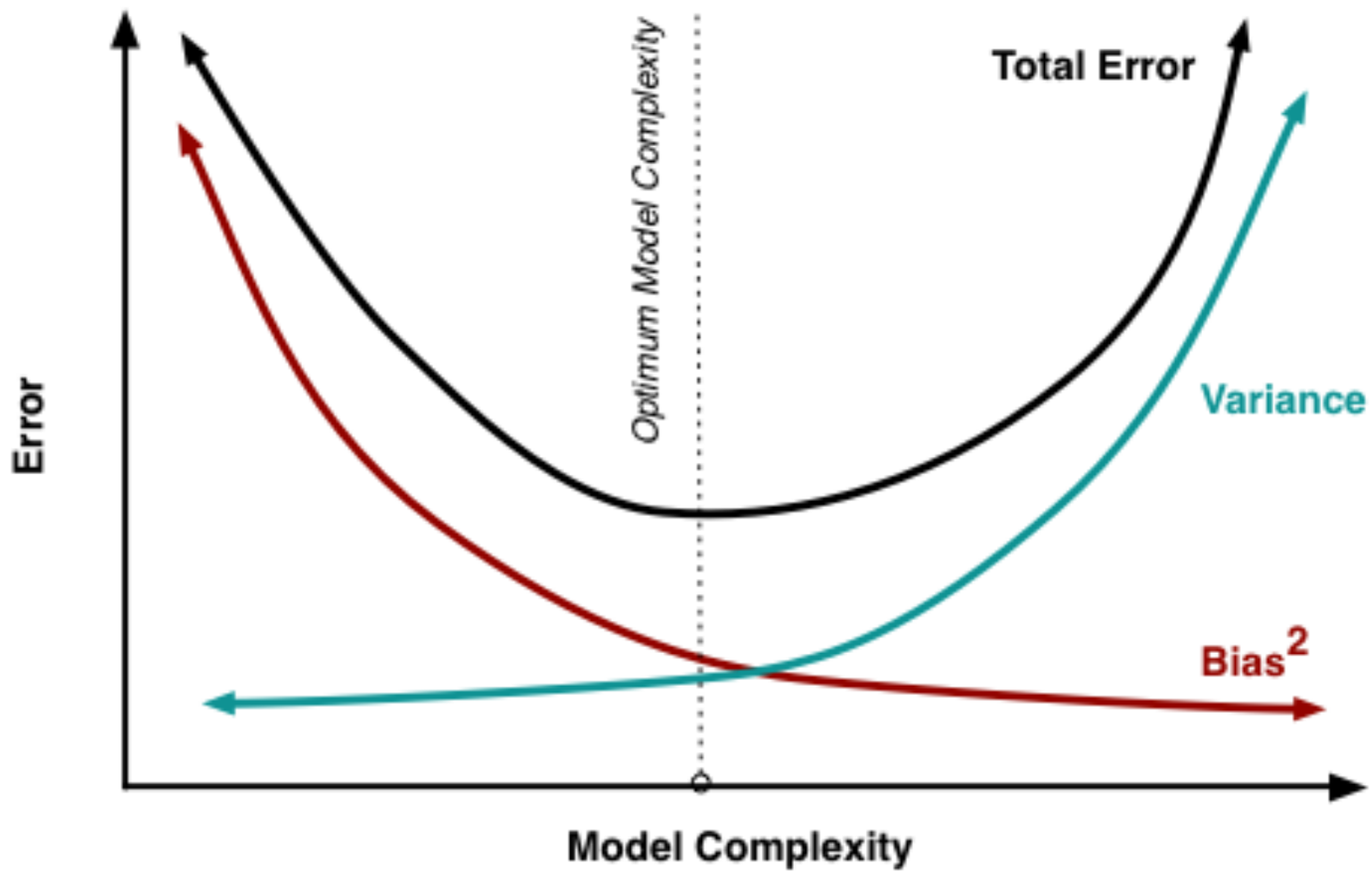
Model Complexity

How many ways can the rules partition the data

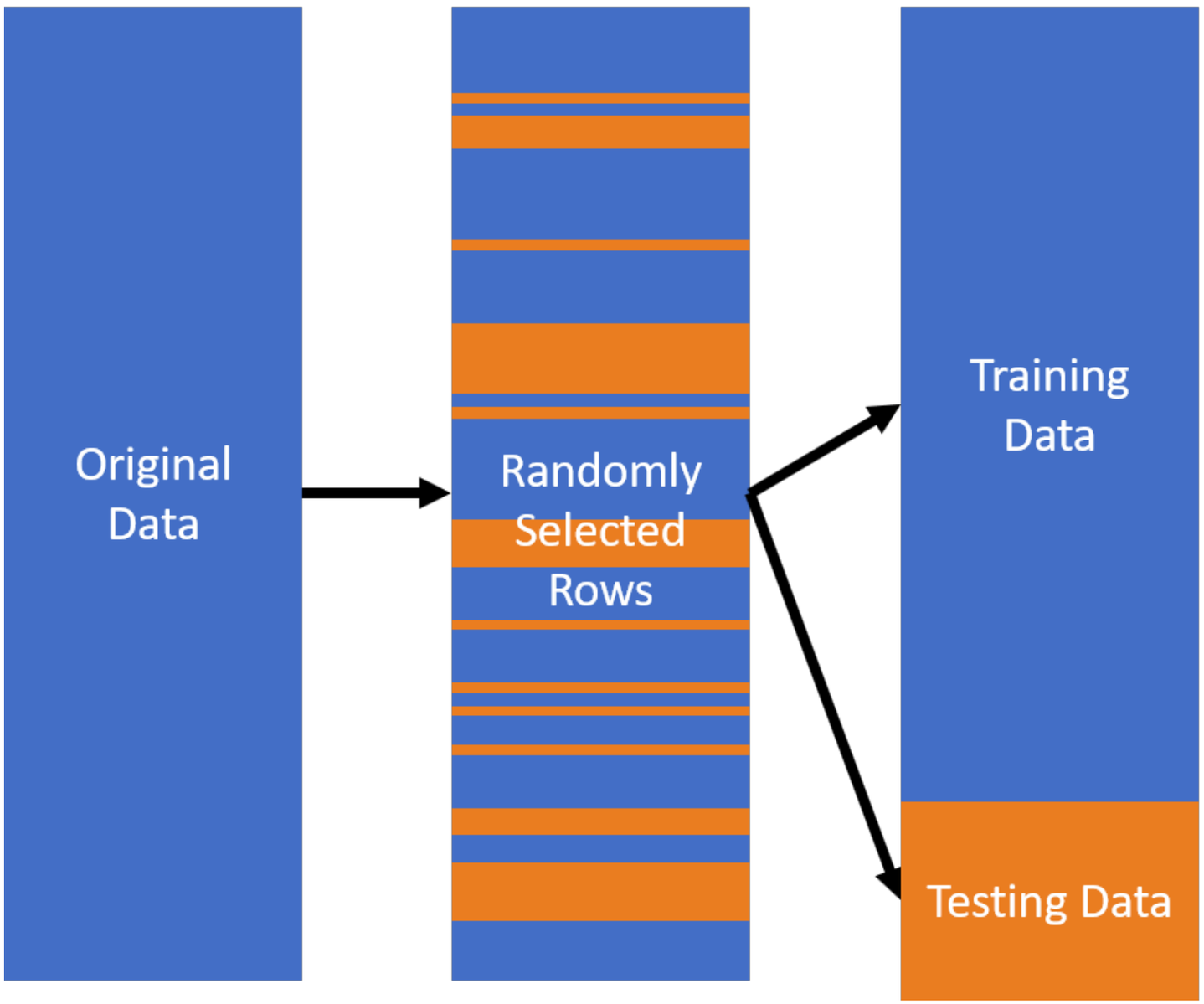
Not expressive enough

Does not apply to future data

Same Tradeoff!



Training and Testing



Use this data for rule selection

Use this data for rule evaluation

More Formally

Features	X	Y (got sick?)
	[redness, how oval]	
	[0.95, 0.25]	1
	[0.25, 0.85]	0
	[0.45, 0.05]	1
	[0.05, 0.75]	0

Another Perspective

	X [redness, how oval]	Y (got sick?)
	[0.95, 0.25]	1
	[0.25, 0.85]	0
	[0.45, 0.05]	1
	[0.05, 0.75]	0

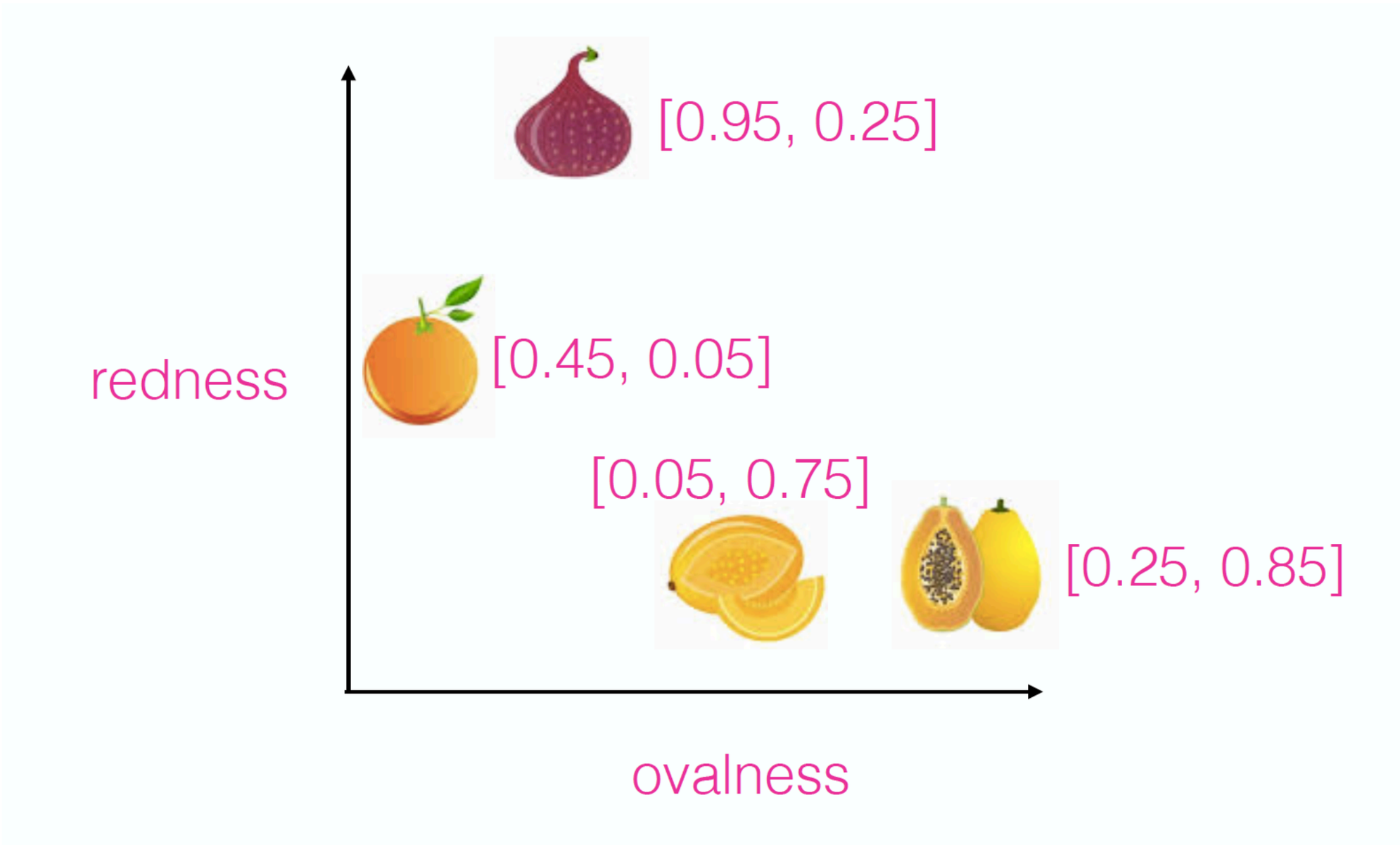
Find a function f(x) -> y

Exactly the same

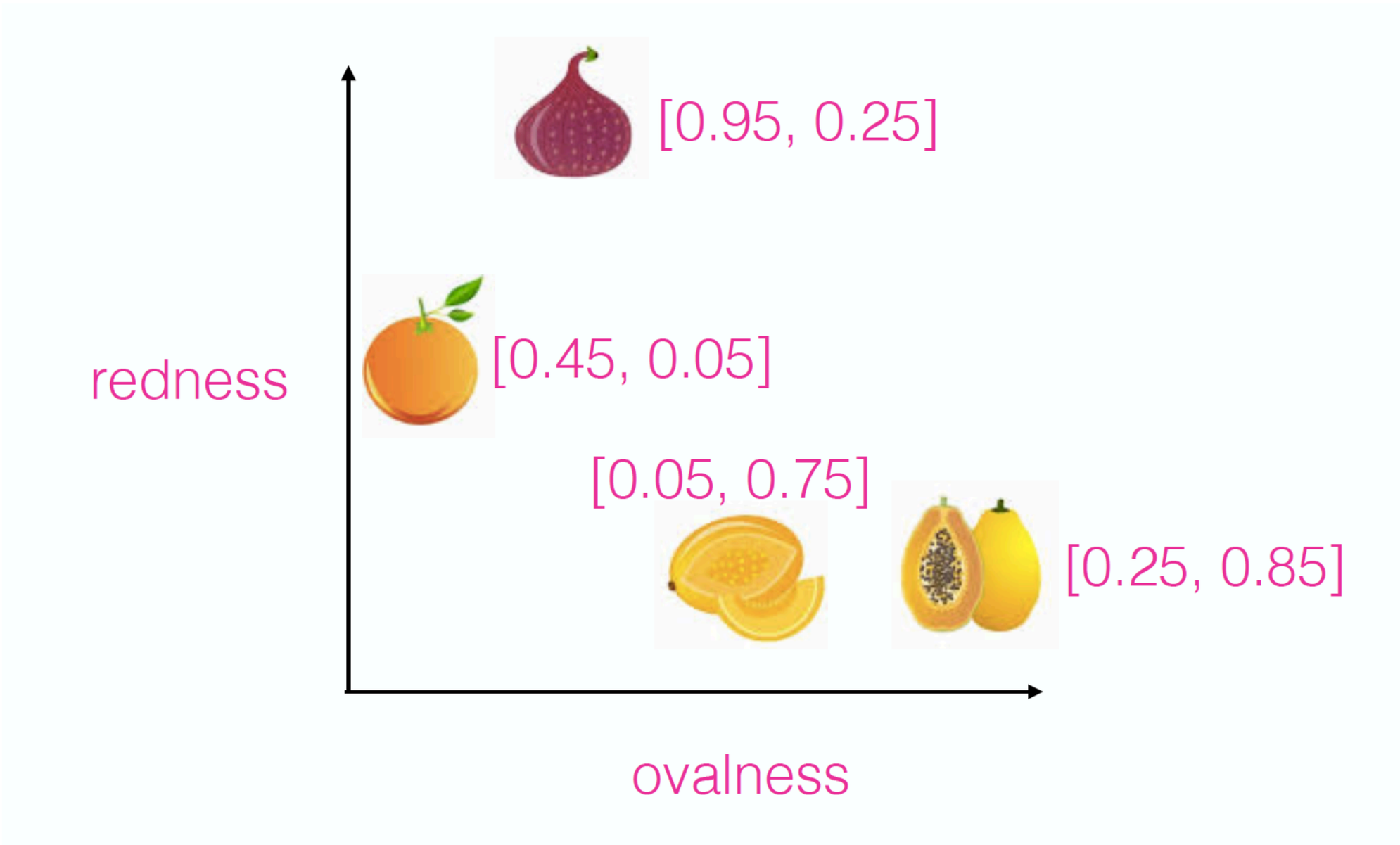
$$y_i = f_\theta(x_i, \epsilon_i)$$

Fitting: Observe xi, yi, **infer theta**

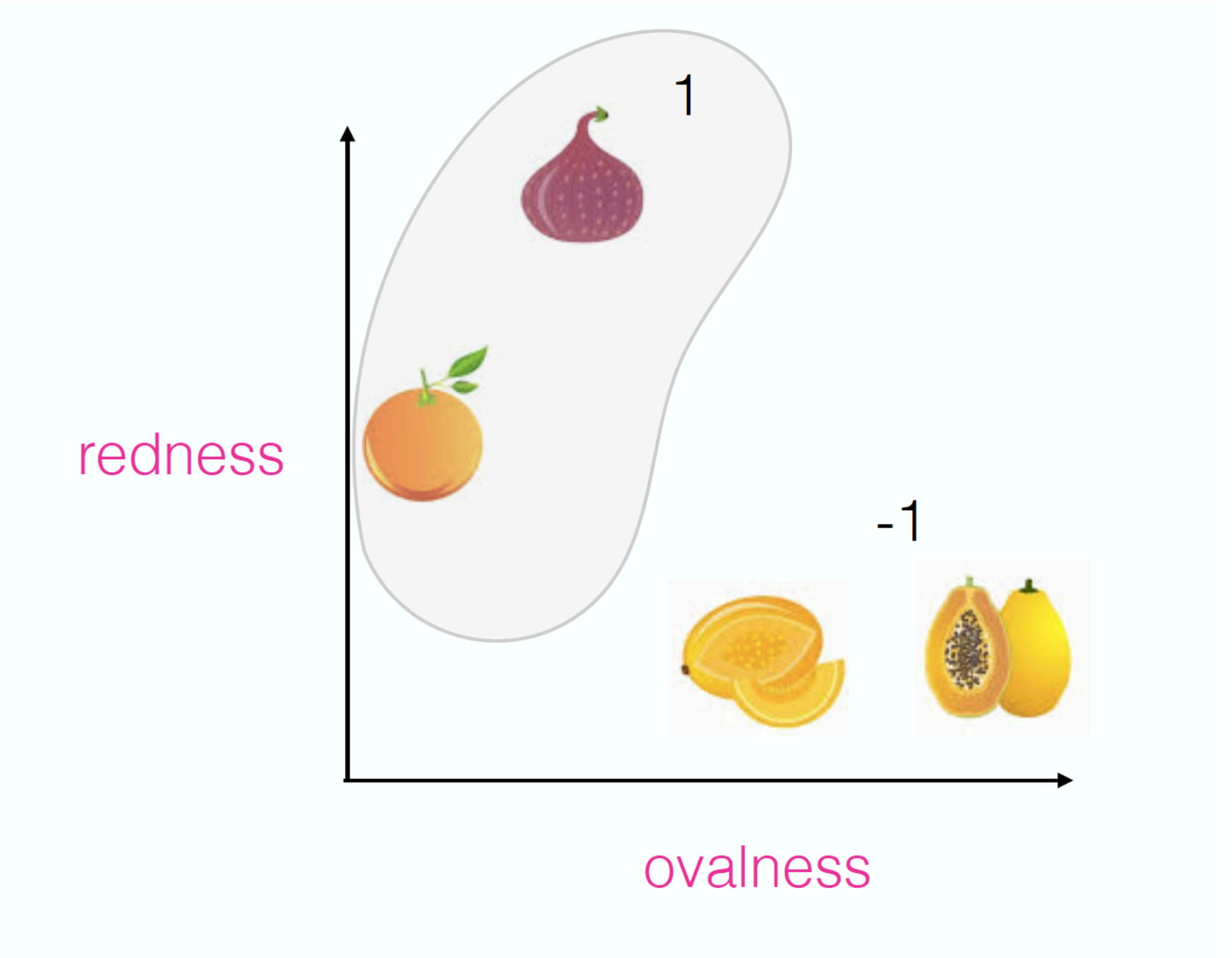
Another Perspective



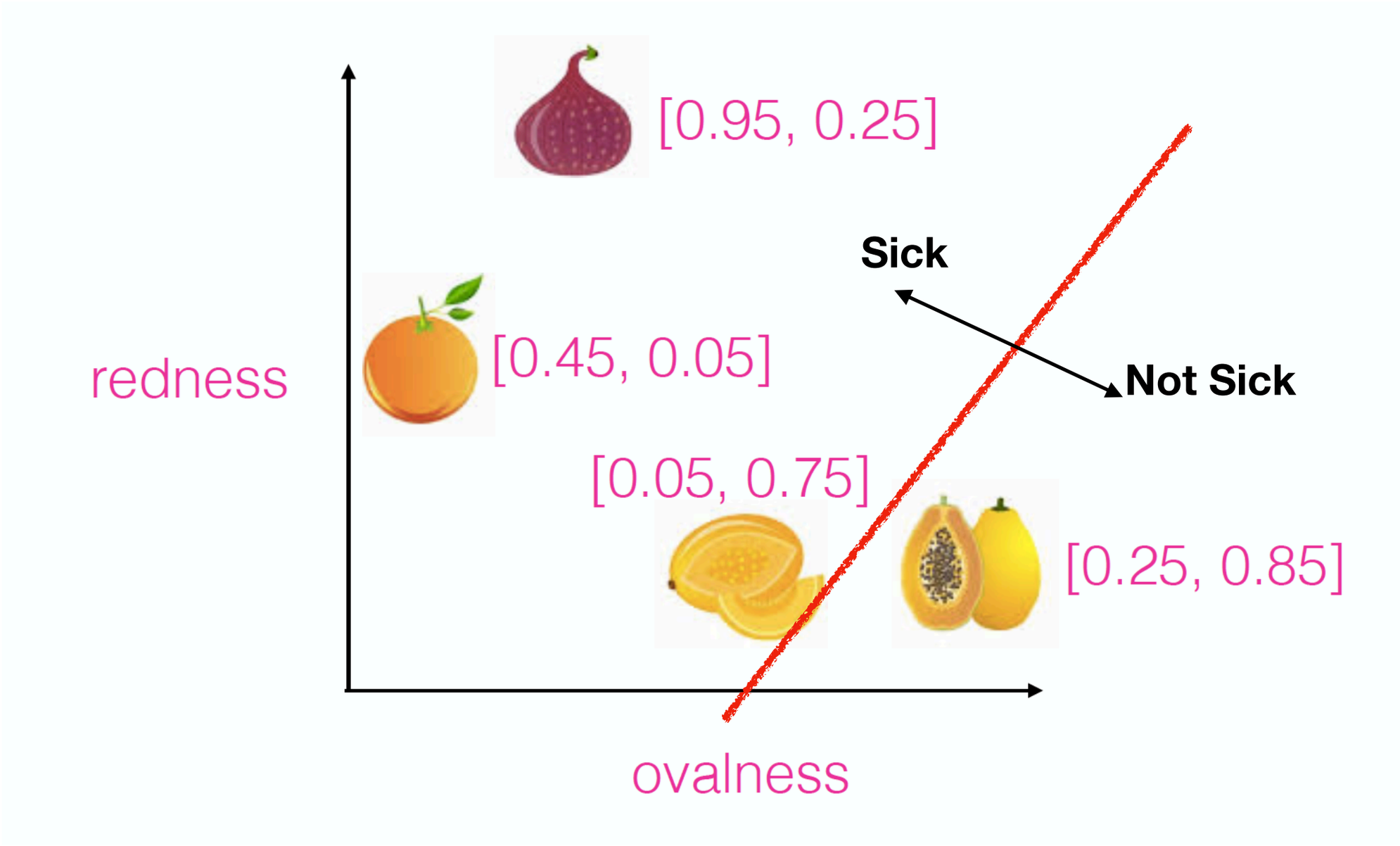
Another Perspective



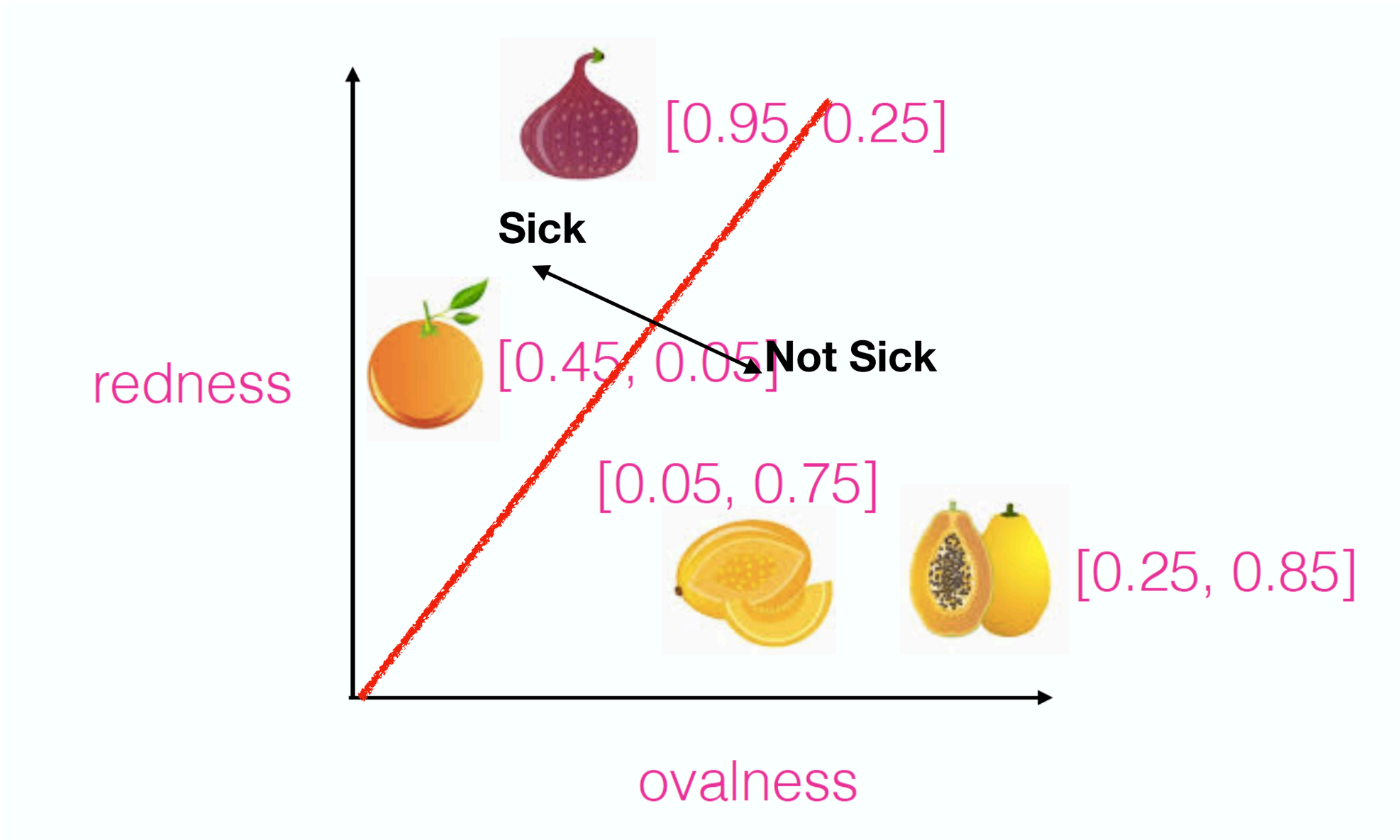
Another Perspective



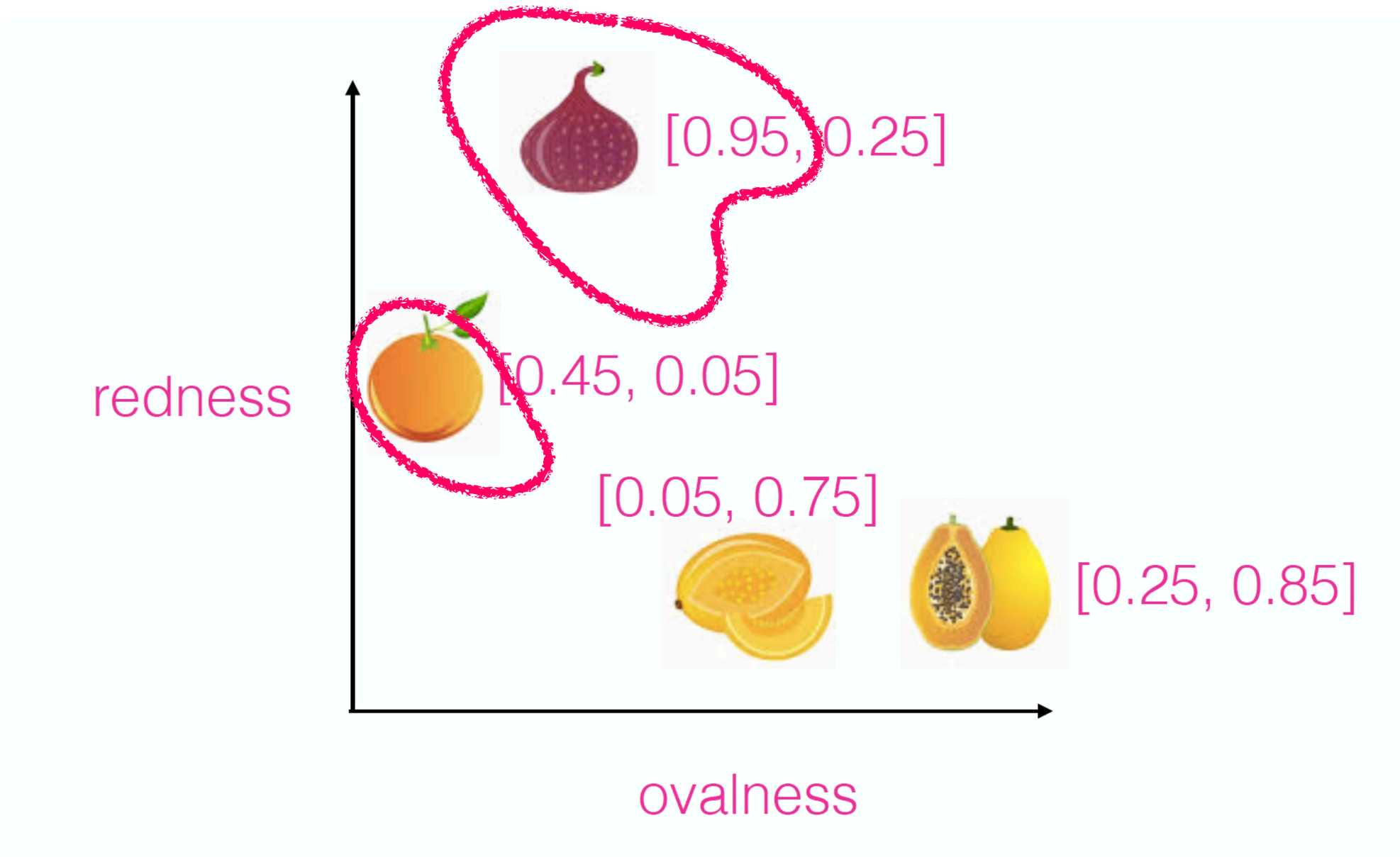
High Bias


[0.95, 0.25]
Sick
Not Sick
redness
[0.45, 0.05]
[0.05, 0.75]
[0.25, 0.85]
ovalness

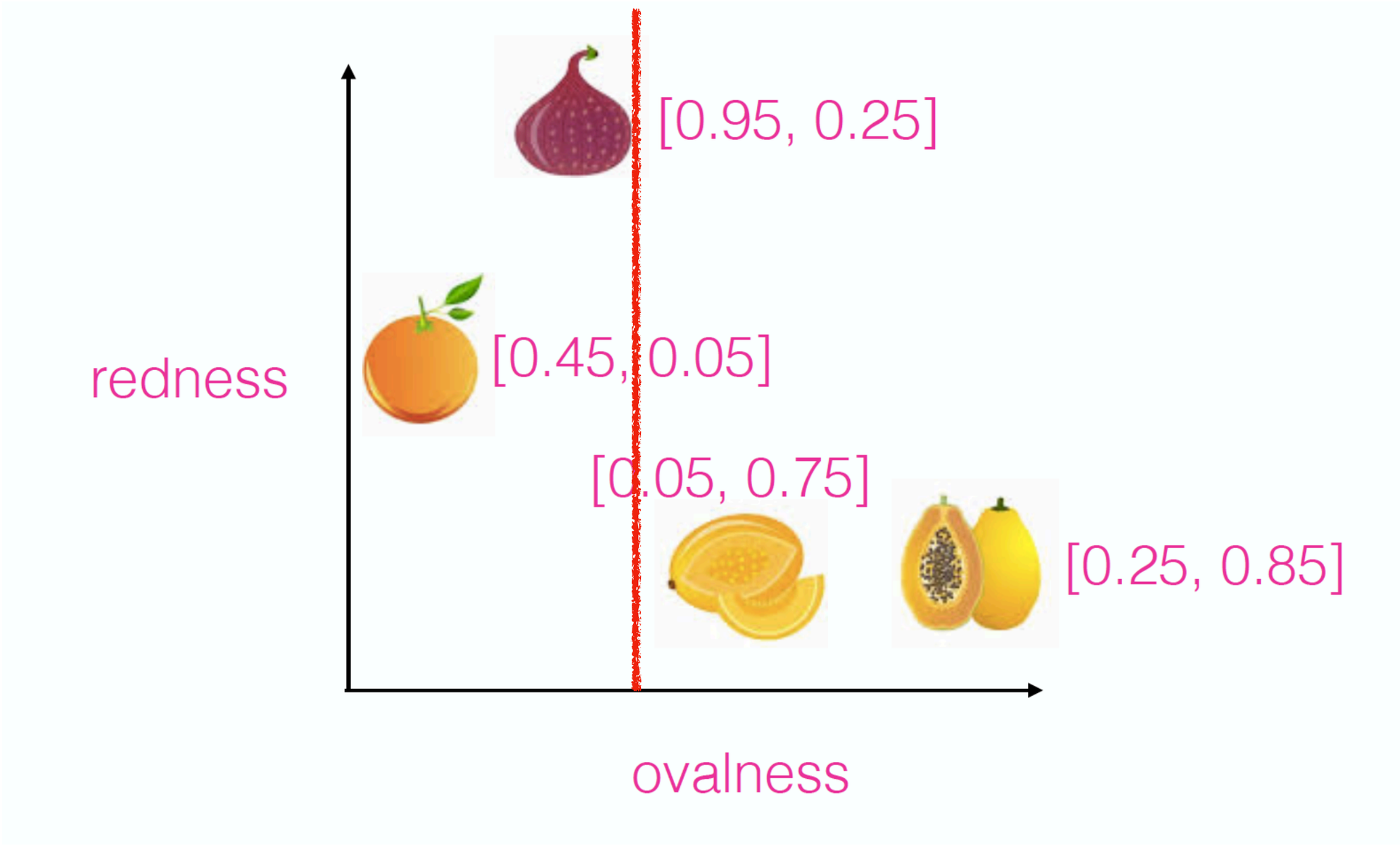
Low Bias


[0.95, 0.25]
Sick
Not Sick
[0.45, 0.05]
redness
[0.05, 0.75]
[0.25, 0.85]
ovalness

High Variance



Low Variance



What Does the Model Look Like?

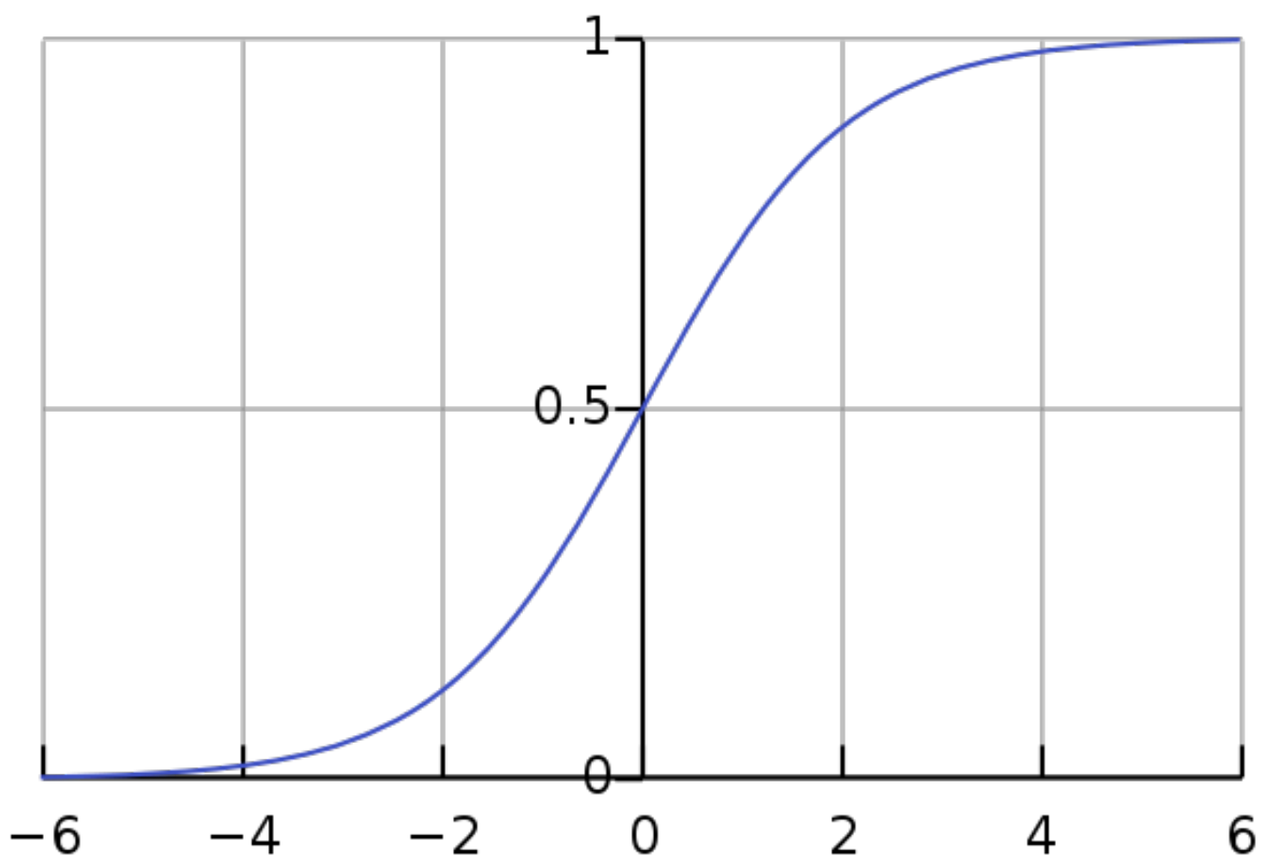
$$y_i = f_\theta(x_i, \epsilon_i)$$

Fitting: Observe xi, yi, **infer theta**

“Logistic” Regression

Function with values between 0 and 1

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}.$$



"Logistic Regression"

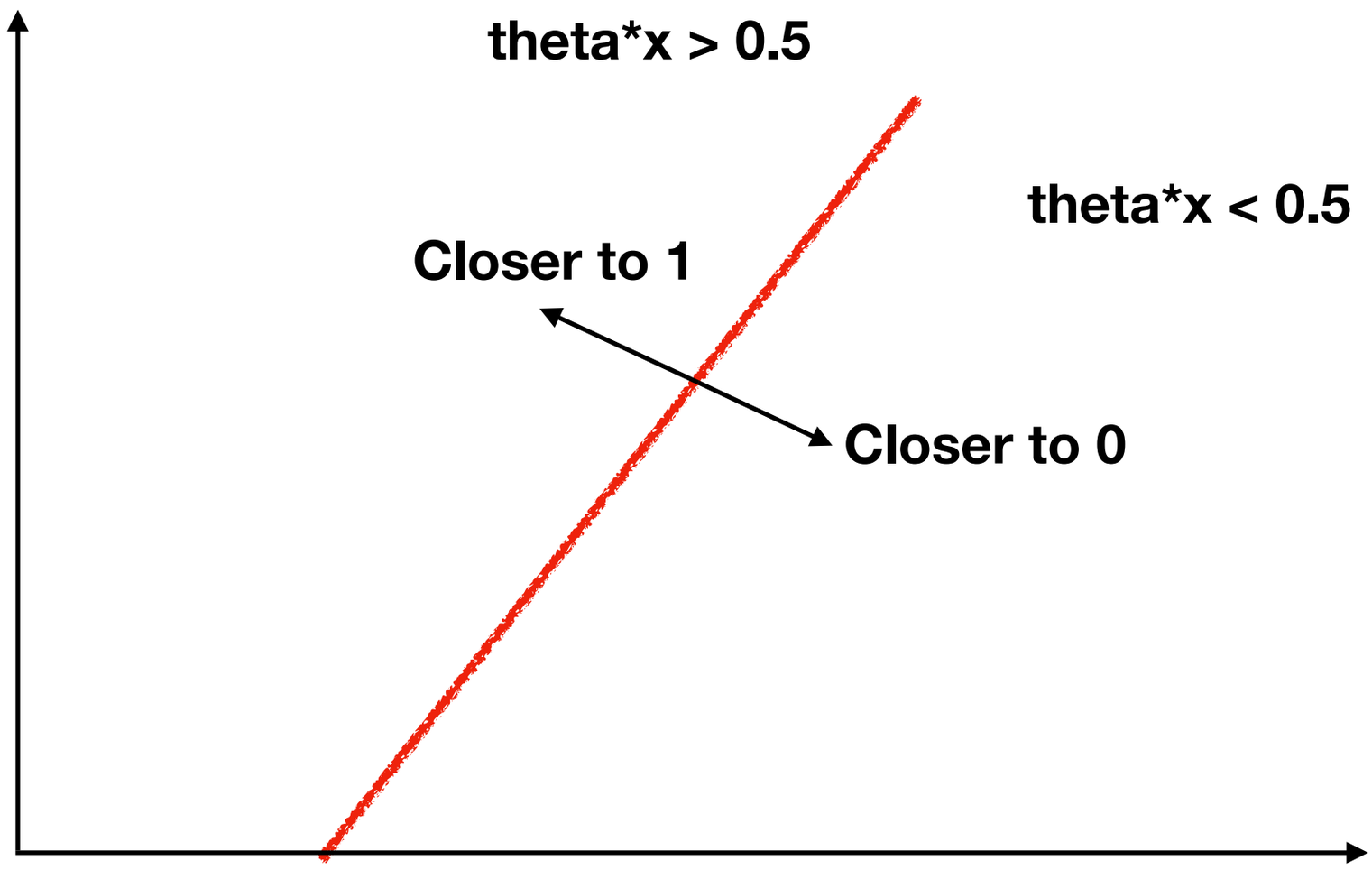
$$f(x) = \theta^T x$$

- Linear function of the features.

$$g(x) = \frac{1}{1 + e^{-f(x)}}$$

- When f(x) is high (greater than 0), g(x) tends towards 1
- When f(x) is low (less than 0), g(x) tends towards 0

A Linear Partioning



Objective

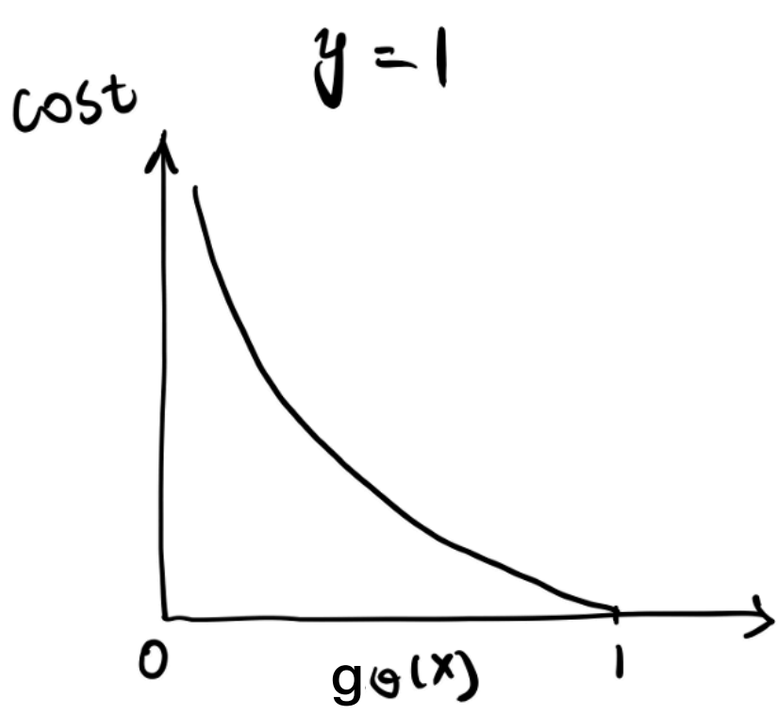
	X [redness, how oval]	Prediction
	[0.95, 0.25]	0.96 ~~ 1
	[0.25, 0.85]	0.1 ~~ 0
	[0.45, 0.05]	0.87 ~~ 1
	[0.05, 0.75]	0.14 ~~ 1

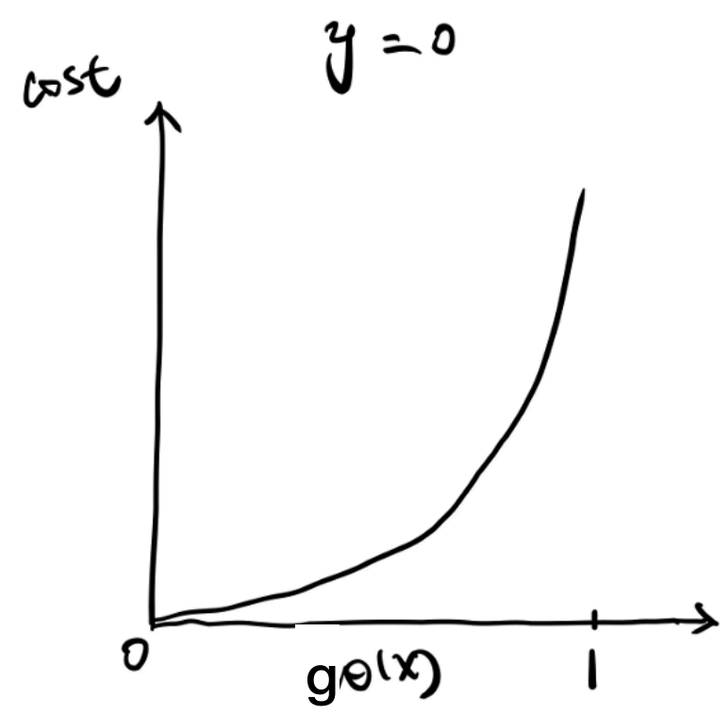
Get the predictions as close to true 0 or 1

"Logistic Regression"

Loss function

$$L(y_i, g_\theta(x_i))) = \begin{cases} -log(g_\theta(x_i)) & \text{if } y_i = 1 \\ -log(1 - g_\theta(x_i)) & \text{if } y_i = 0 \end{cases}$$

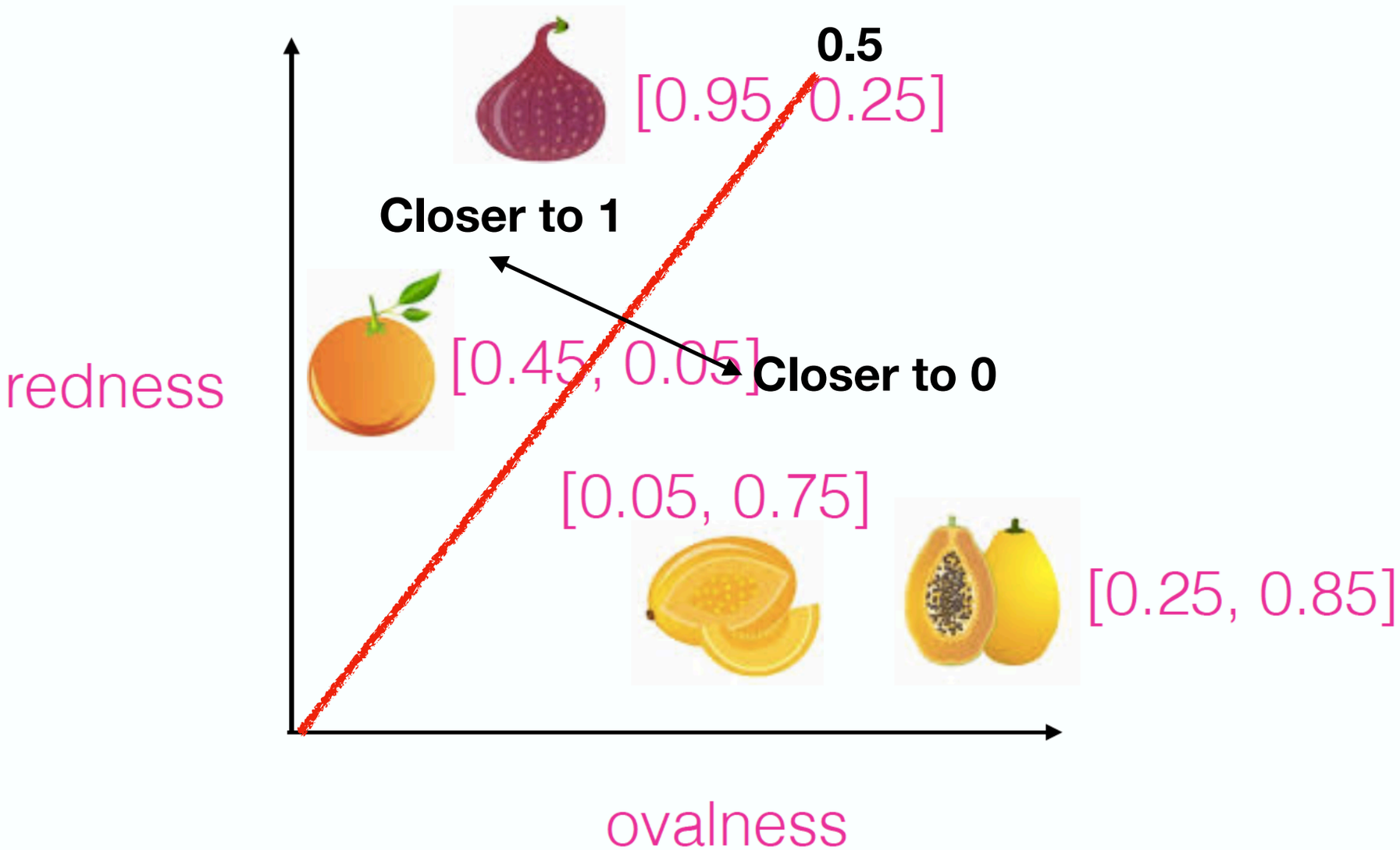




“Logistic Regression”

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.linear_model import LogisticRegression
>>> X, y = load_iris(return_X_y=True)
>>> clf = LogisticRegression(random_state=0).fit(X, y)
>>> clf.predict(X[:2, :])
array([0, 0])
>>> clf.predict_proba(X[:2, :])
array([[9.8...e-01, 1.8...e-02, 1.4...e-08],
       [9.7...e-01, 2.8...e-02, ...e-08]])
>>> clf.score(X, y)
0.97...
```

A Linear Partioning



“Linear” Regression

Least squares principle

$$\min_{a,c} \sum_{i=1}^{N} (y_i - (ax_i + c))^2$$

Actual **Prediction**

Find parameters that minimize the squared error between prediction and actual.

"Linear" Regression

Least squares principle

Features

$$\min_{\theta} \sum_{i=1}^{N} (y_i - \theta^T x)^2$$

Actual Prediction

Find parameters that minimize the squared error between prediction and actual.

“Logistic” Regression

Least squares principle (kind of…)

$$\min_{\theta} \sum_{i=1}^{N} -y_i \log(g_\theta(x_i)) + (1 - y_i) \log(1 - g_\theta(x_i))$$

Find parameters that linearly partition the feature space to best match the labels.

Recap: Simulation v.s. Model Fitting

$$y_i = f_\theta(x_i, \epsilon_i)$$

Simulation: Observe xi, ei, and theta, **generate yi**

Fitting: Observe xi, yi, **infer theta**

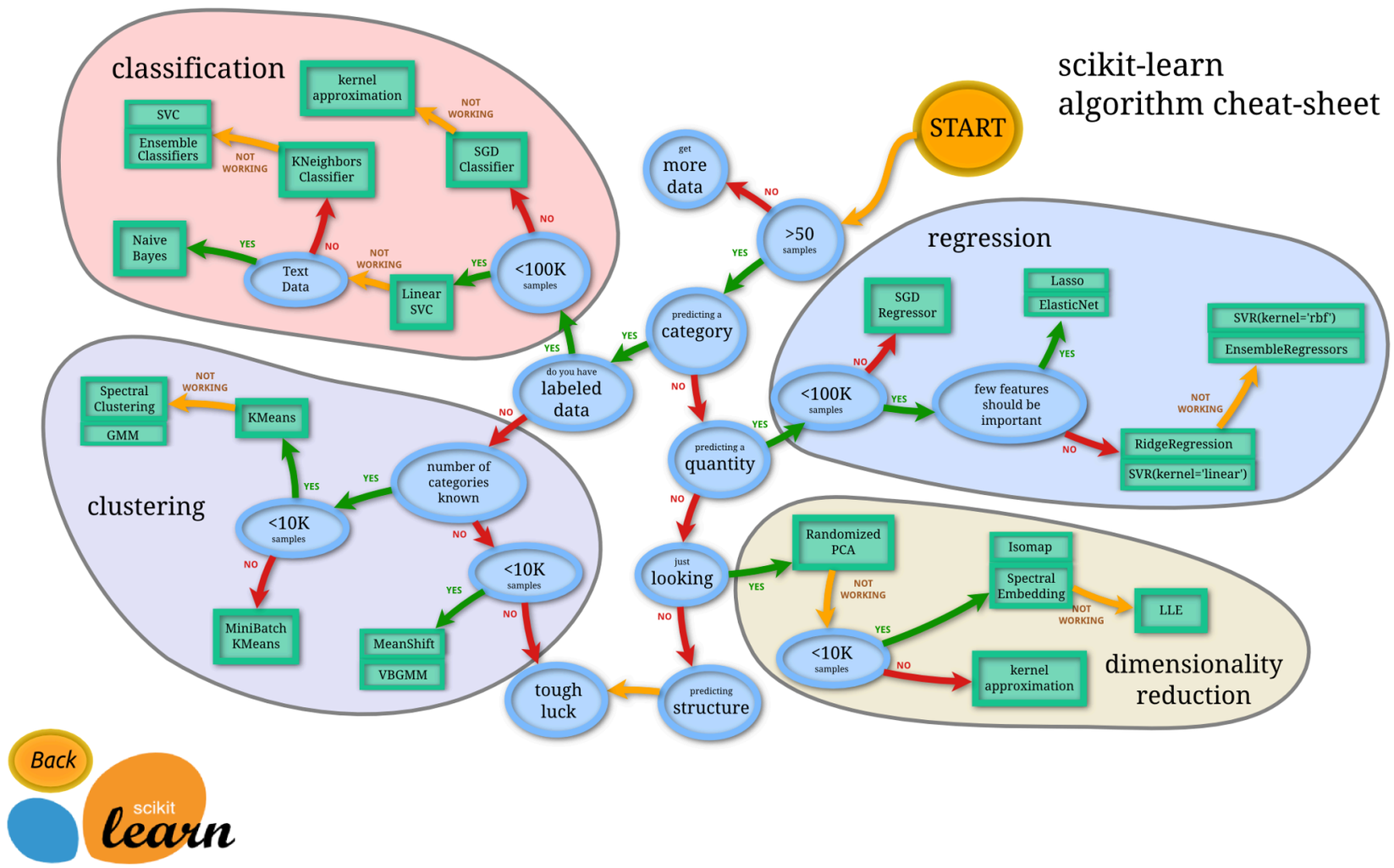
Regression Problems -> Continuous Y

Classification Problems -> Discrete Y

Machine Learning

Given features (X) and labels (Y)

Find a function to predict Y from X



Features Are Up-to You

Features

	X	Y (got sick?)
	[redness, how oval]	
	[0.95, 0.25]	1
	[0.25, 0.85]	0
	[0.45, 0.05]	1
	[0.05, 0.75]	0