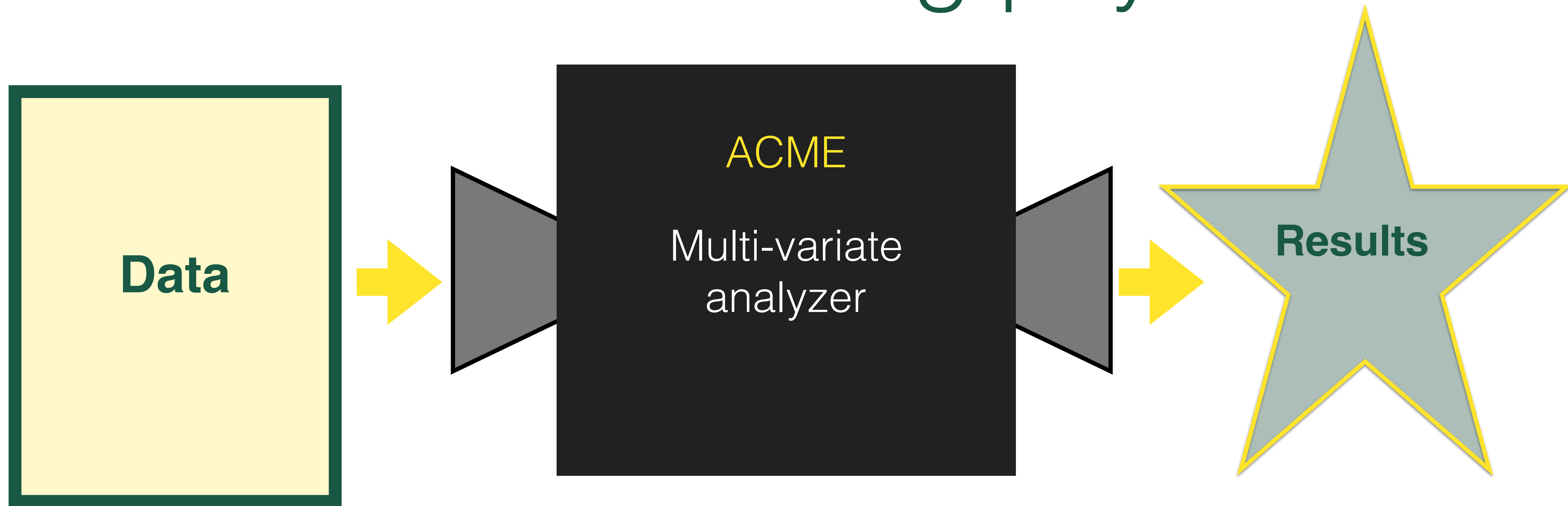


Machine learning physics



Bryan Ost diek

Collider Physics and the Cosmos Workshop
Galileo Galilei Institute for Theoretical Physics, Arcetri Florence
October 6, 2017

What is machine learning (for)?

Labeled data

Unlabeled data

Getting information from data

Supervised Learning

- Classification
- Numerical Predictions
- etc

Unsupervised Learning

- Clustering
- Anomaly Detection
- Generative adversarial networks
- etc

Hybrid?

- Learning from label proportions
- Classification without labels

Opening the box

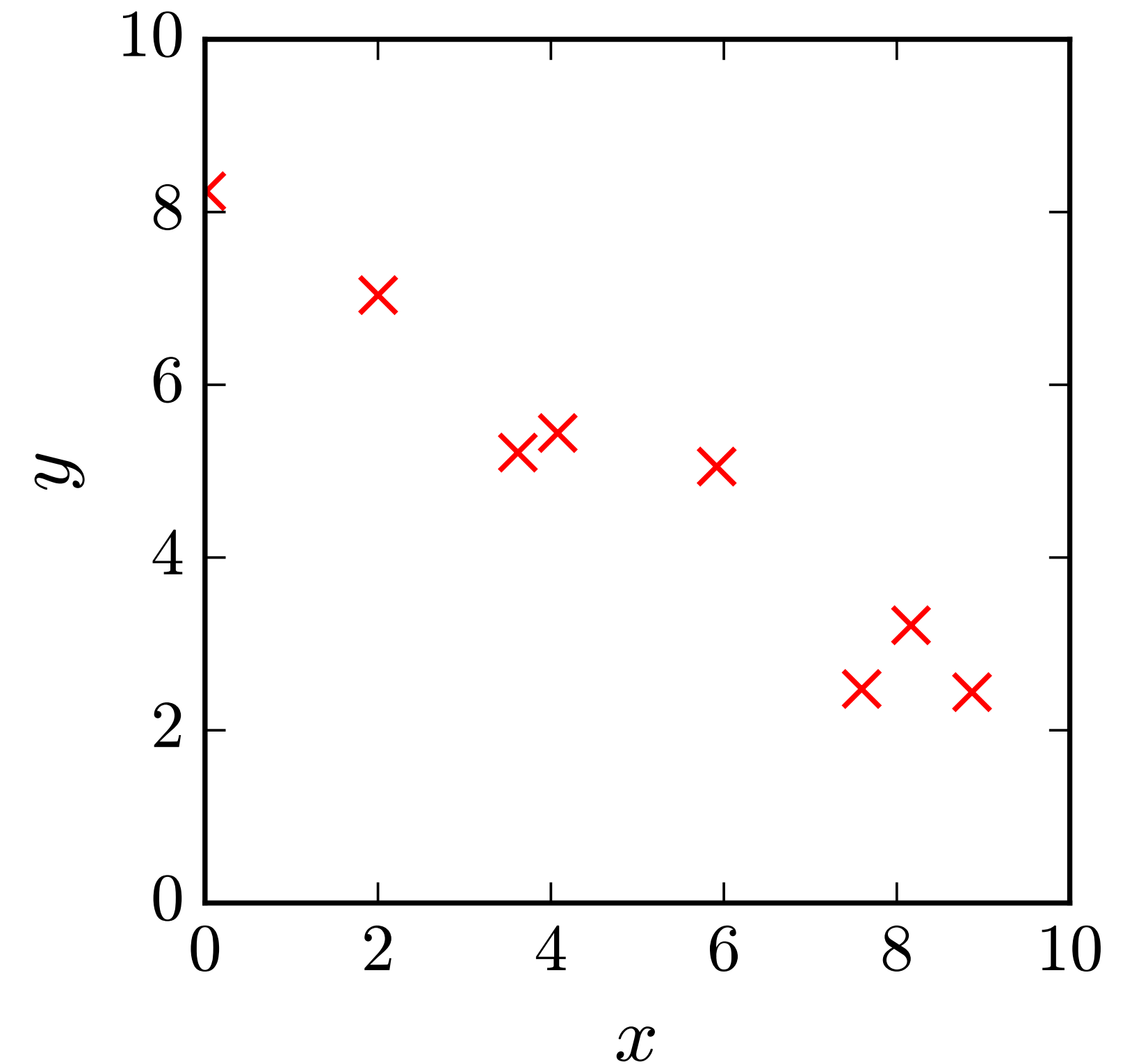


1. Review supervised learning
 - a. Linear Regression
 - b. Logistic Regression
2. Motivate use of Neural Networks
3. Weak supervision
(1706.09451)
4. What is the machine learning?
(1709.10106)

Review: Linear Regression

How to fit data

1. Plot the data
2. Define the function
 - $f(x, \vec{a}) = a_0 + a_1x$
3. Choose how to know what fits best
 - a.k.a. *Loss Function*
 - MSE: $L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^N (f(x_i, \vec{a}) - y_i)^2$
5. Find the minimum error (loss) (cost)
 - $a_{\text{best}} = a$ when $\left(\frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \Big|_{x, y} = 0 \right)$



Review: Linear Regression

How to fit data

1. Plot the data
2. Define the function
3. Choose how to know what fits best

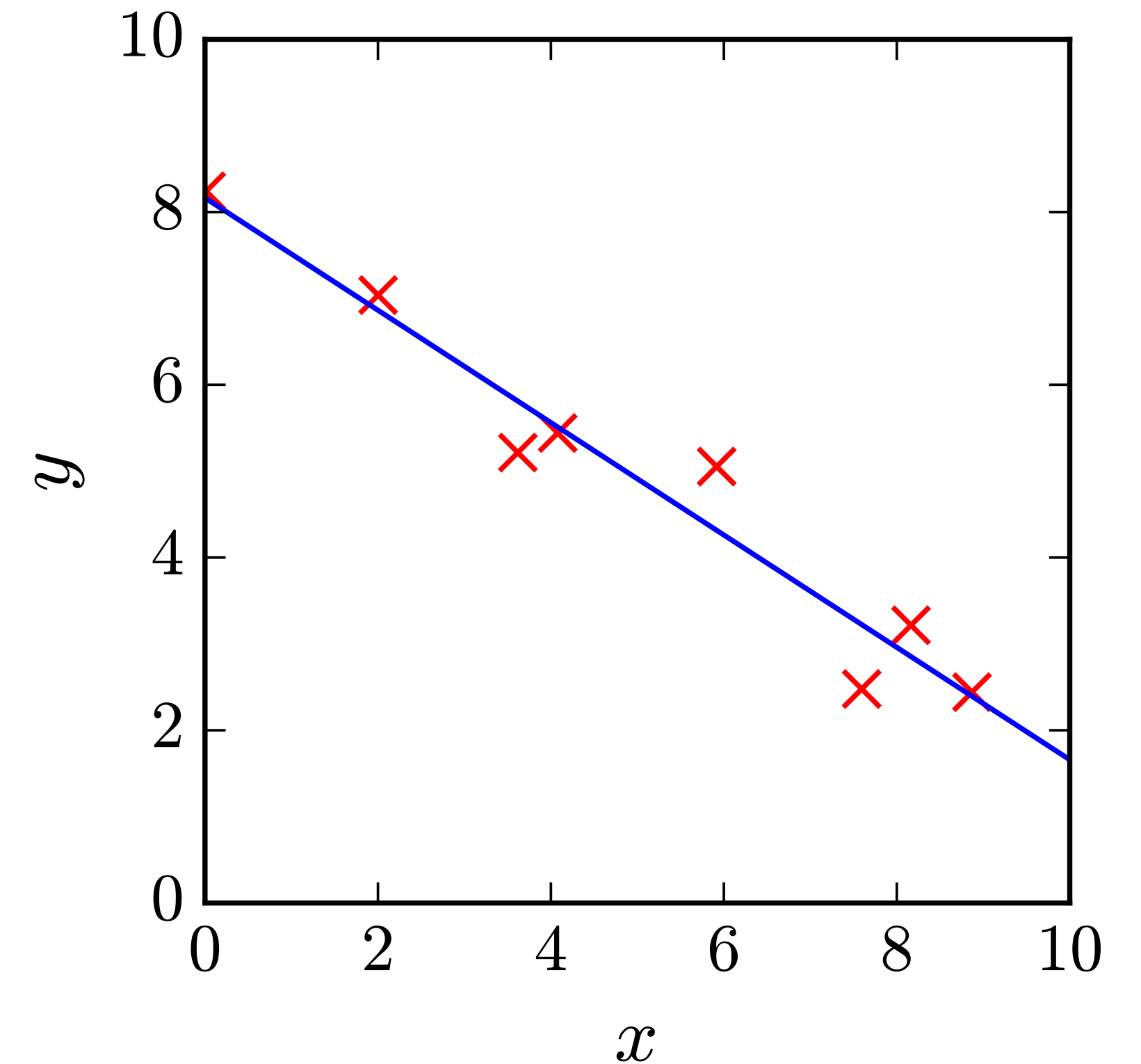
- $f(x, \vec{a}) = a_0 + a_1x$

- a.k.a. *Loss Function*

- MSE: $L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^N (f(x_i, \vec{a}) - y_i)^2$

5. Find the minimum error (loss) (cost)

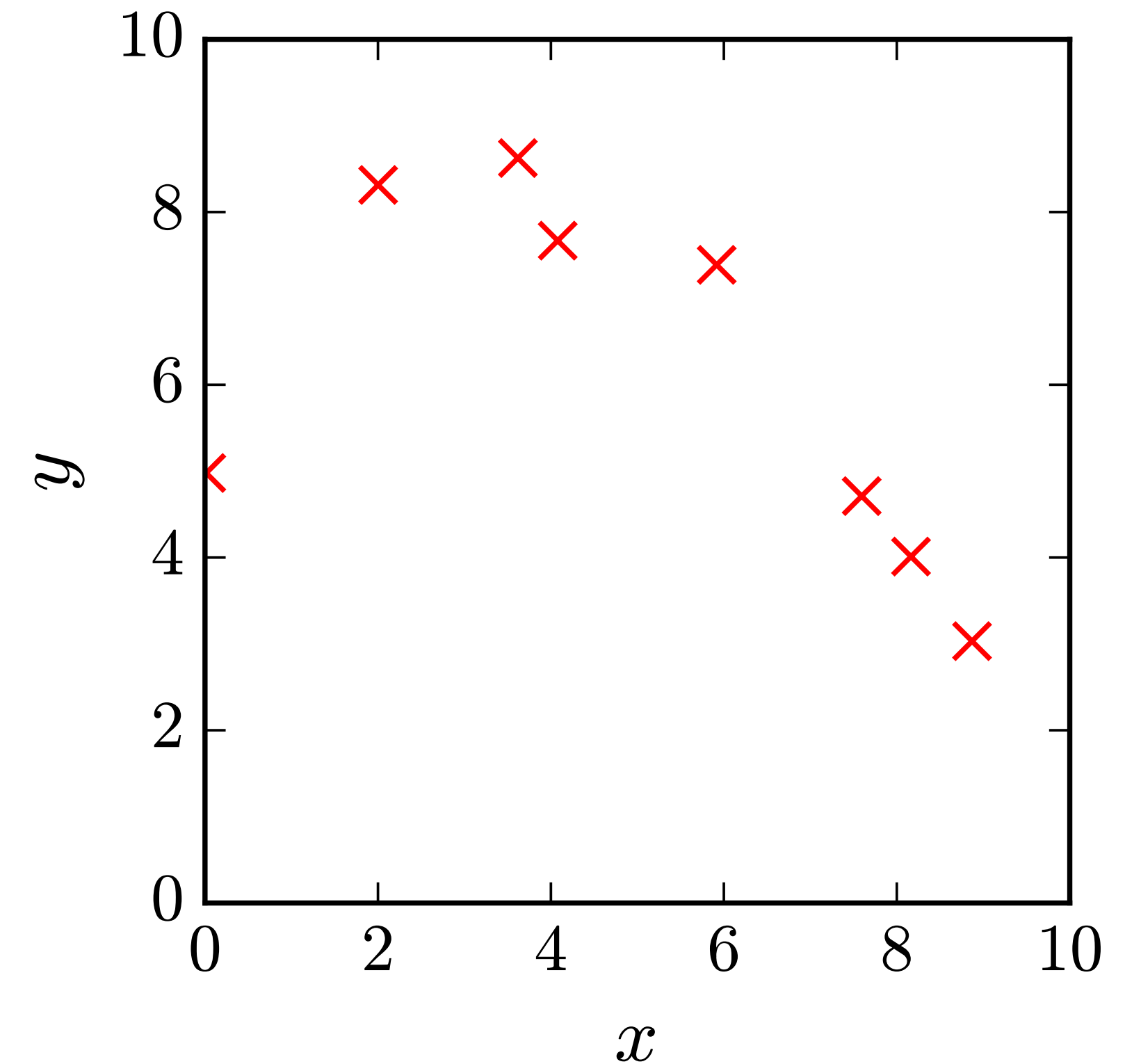
- $a_{\text{best}} = a$ when $\left(\frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \Big|_{x, y} = 0 \right)$



Review: Linear Regression

How to fit data

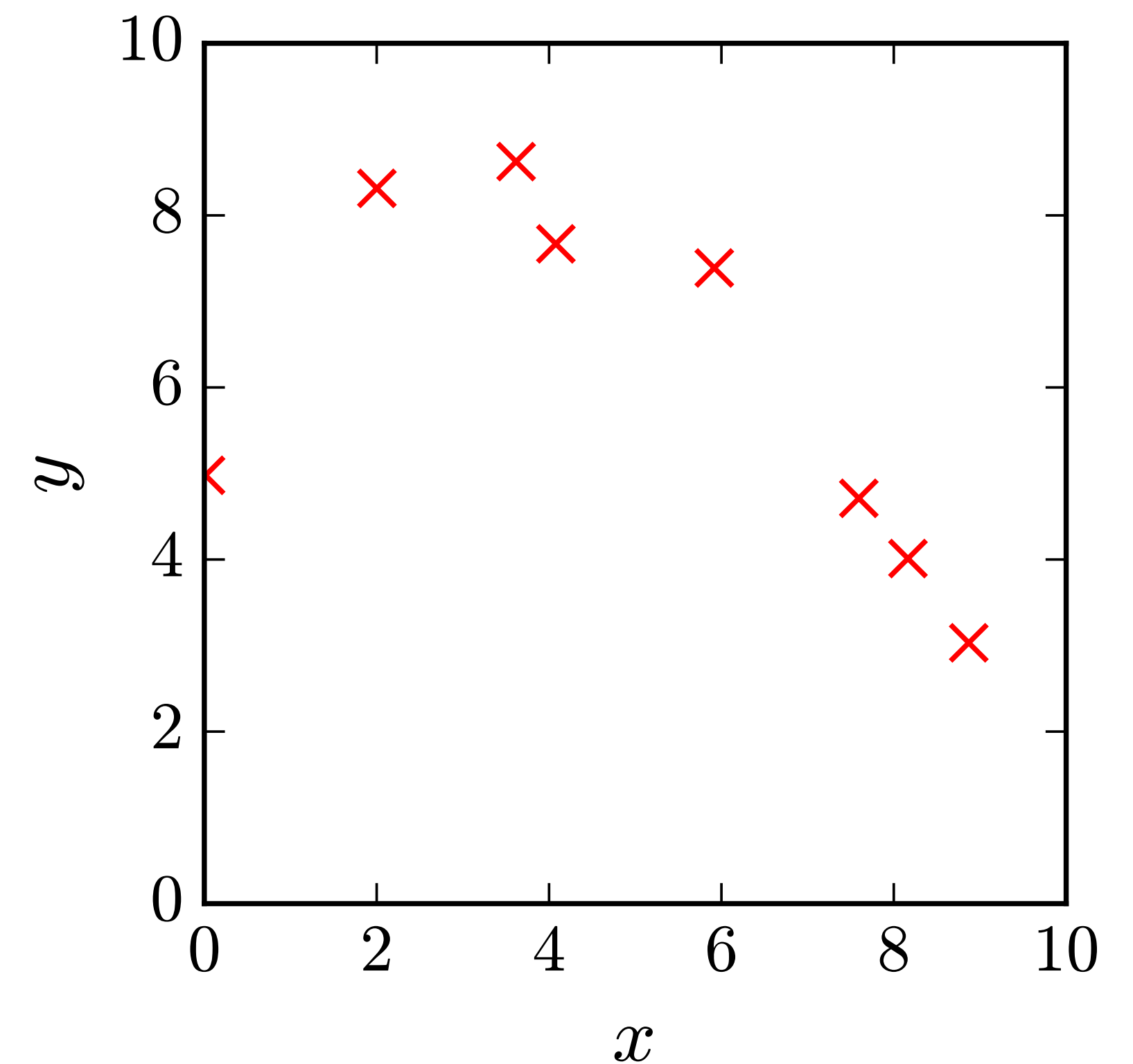
1. Plot the data
2. Define the function
 - $f(x, \vec{a}) = a_0 + a_1x + a_2x^2$
3. Choose how to know what fits best
 - a.k.a. *Loss Function*
 - MSE: $L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^N (f(x_i, \vec{a}) - y_i)^2$
5. Find the minimum error (loss) (cost)
 - $a_{\text{best}} = a$ when $\left(\frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \Big|_{x, y} = 0 \right)$



Review: ~~Linear~~ Quadratic? Regression

How to fit data

1. Plot the data
2. Define the function
 - $f(x, \vec{a}) = a_0 + a_1x + a_2x^2$
3. Choose how to know what fits best
 - a.k.a. *Loss Function*
 - MSE: $L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^N (f(x_i, \vec{a}) - y_i)^2$
5. Find the minimum error (loss) (cost)
 - $a_{\text{best}} = a$ when $\left(\frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \Big|_{x, y} = 0 \right)$



Review: ~~Linear~~ Quadratic? Regression

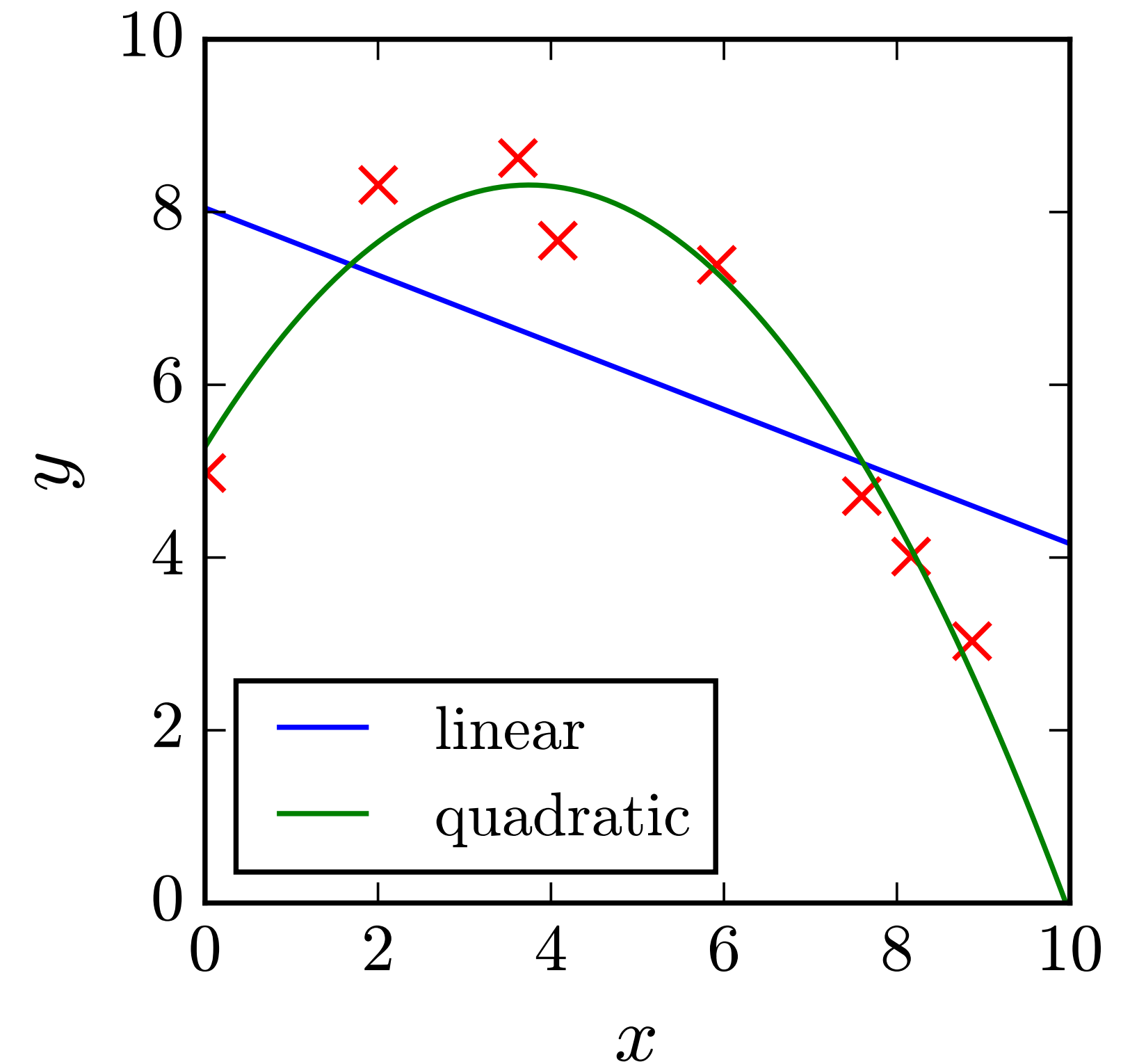
How to fit data

1. Plot the data
2. Define the function
 - $f(x, \vec{a}) = a_0 + a_1x + a_2x^2$
3. Choose how to know what fits best
 - a.k.a. *Loss Function*

- MSE:
$$L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^N (f(x_i, \vec{a}) - y_i)^2$$

5. Find the minimum error (loss) (cost)

- $a_{\text{best}} = a$ when
$$\left(\frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \Big|_{x, y} = 0 \right)$$



Is that good enough?
How many parameters can we add?

Review: ~~Linear~~ Quadratic? Regression

How to fit data

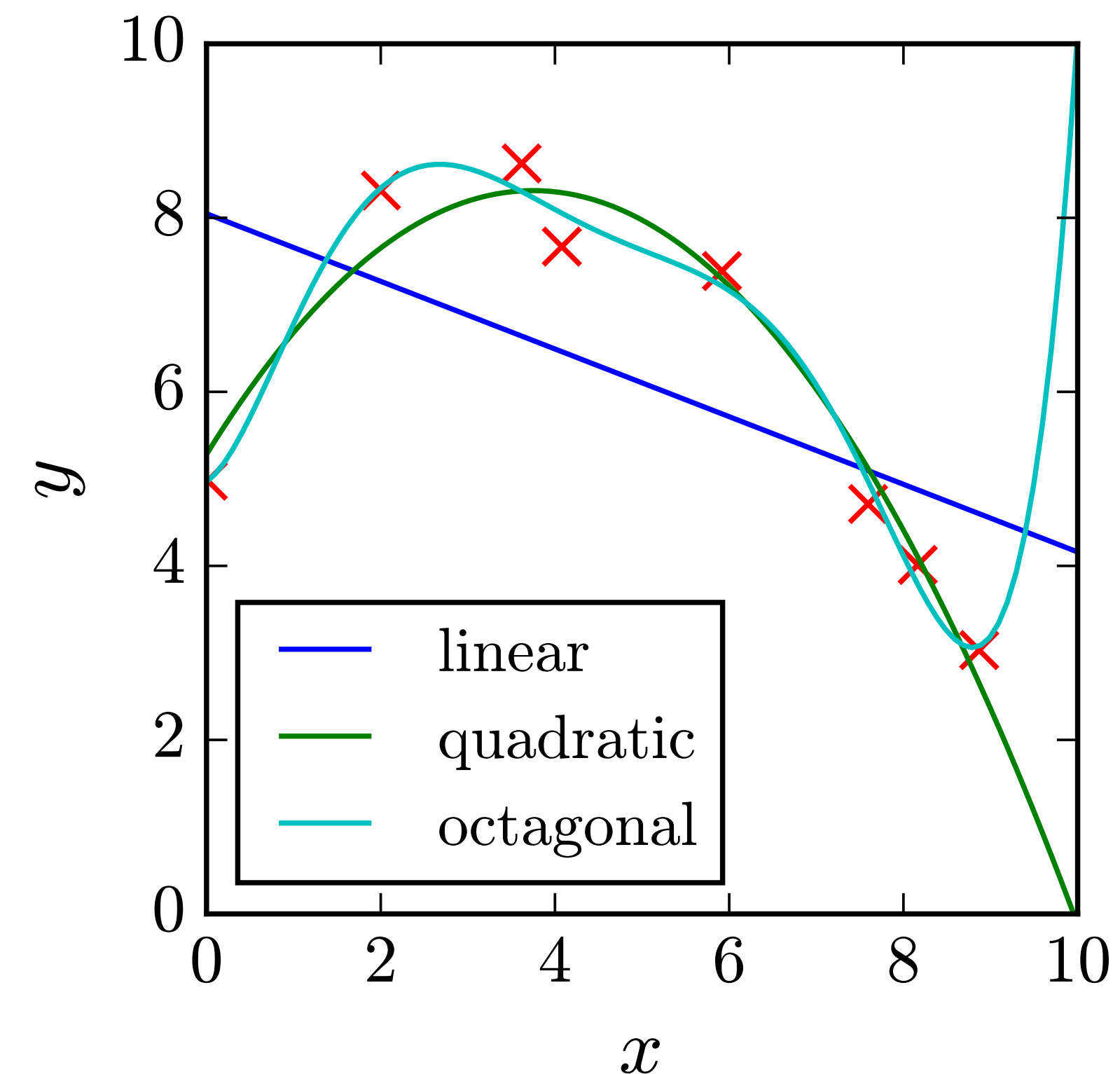
1. Plot the data
2. Define the function
 - $f(x, \vec{a}) = a_0 + a_1x + a_2x^2$
3. Choose how to know what fits best

- a.k.a. *Loss Function*

- MSE: $L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^N (f(x_i, \vec{a}) - y_i)^2$

5. Find the minimum error (loss) (cost)

- $a_{\text{best}} = a$ when $\left(\frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \Big|_{x, y} = 0 \right)$



Is that good enough?
How many parameters can we
add?

Review: ~~Linear~~ Quadratic? Regression

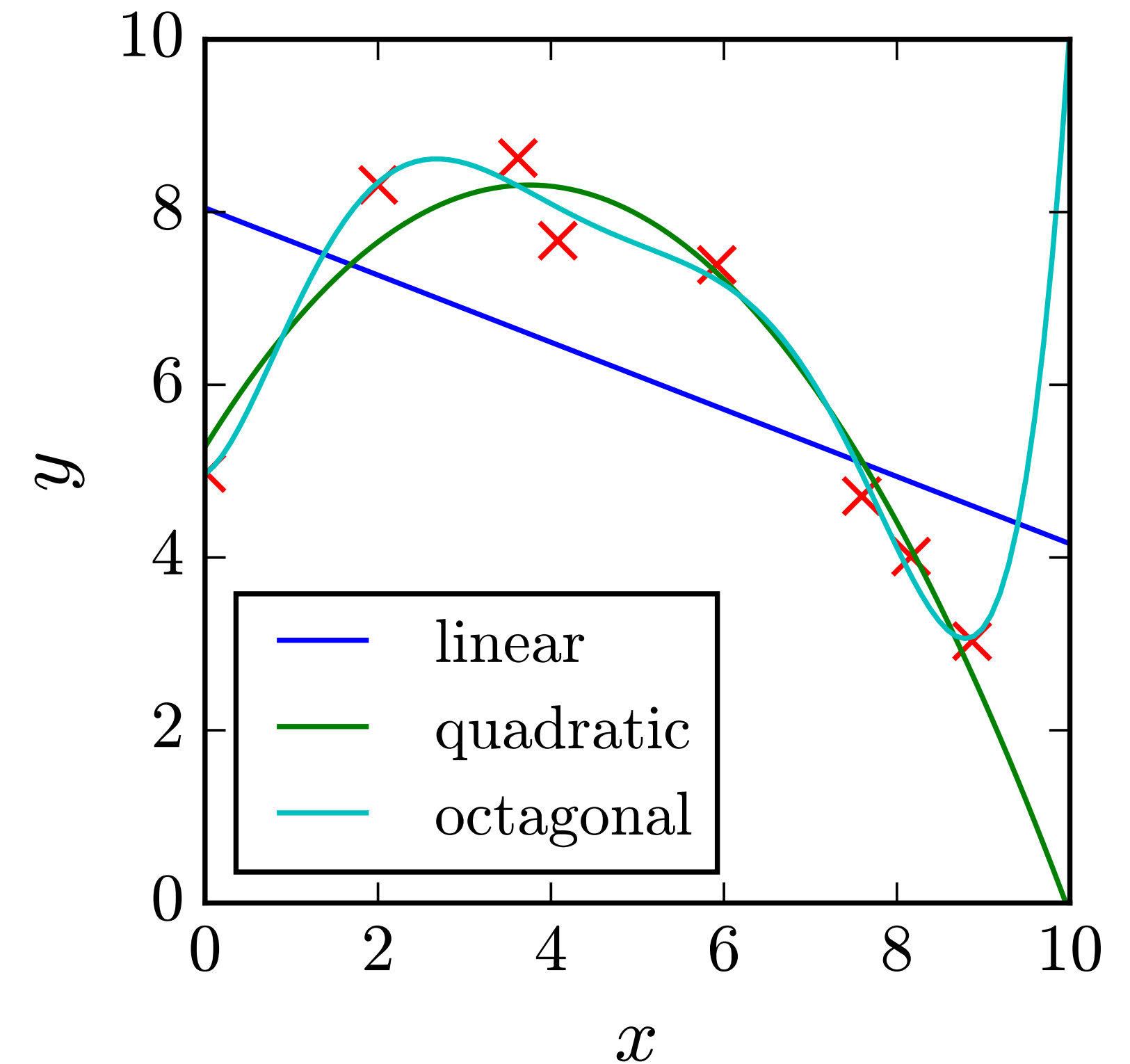
How to fit data

1. Plot the data
2. Define the function
 - $f(x, \vec{a}) = a_0 + a_1x + a_2x^2$
3. Choose how to know what fits best
 - a.k.a. *Loss Function*

- MSE: $L(x, y, \vec{a}) = \frac{1}{N} \sum_{i=1}^N (f(x_i, \vec{a}) - y_i)^2$

5. Find the minimum error (loss) (cost)

- $a_{\text{best}} = a$ when $\left(\frac{\partial L(x, y, \vec{a})}{\partial \vec{a}} \Big|_{x, y} = 0 \right)$



How to avoid overfitting? we can't use a model that is too complex. Avoid over fitting to be able to use for predictions

Logistic Regression

What if we are trying to predict a class, not a number?



quark(s)



gluon

Logistic Regression

What if we are trying to predict a class, not a number?

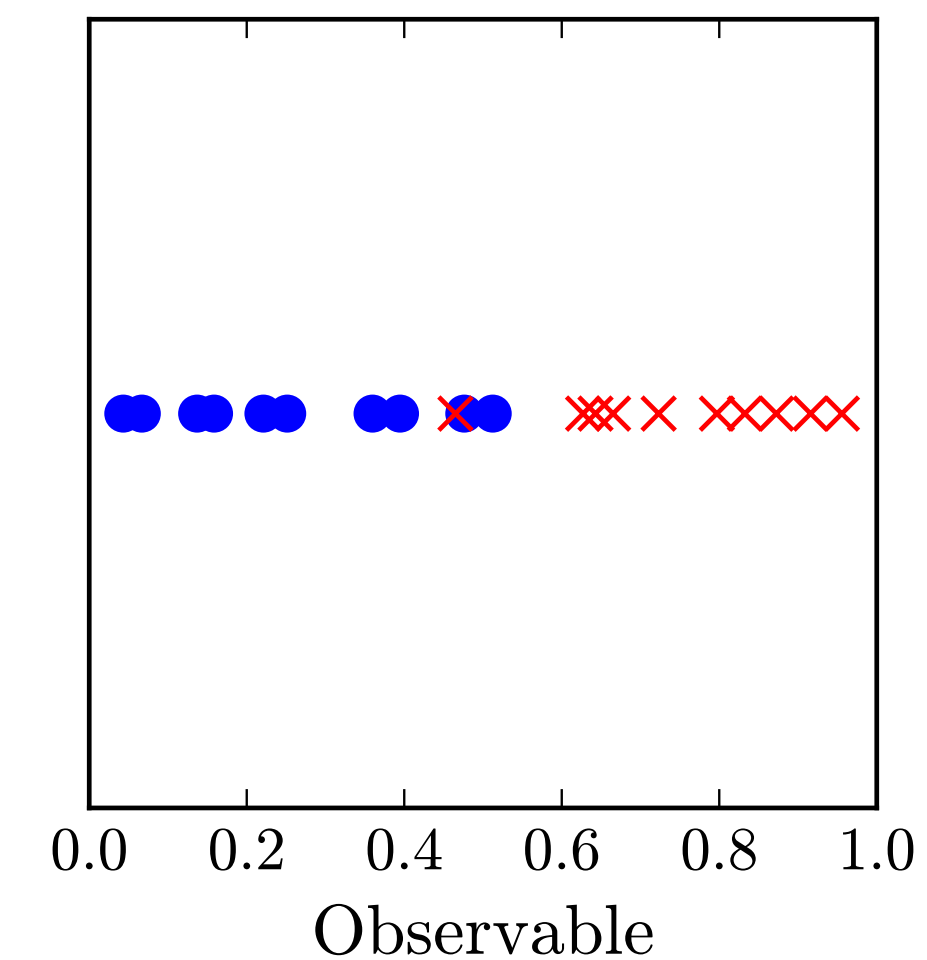


quark(s)



gluon

What is the y-value we are trying to fit/predict?



Logistic Regression

What if we are trying to predict a class, not a number?



quark(s)

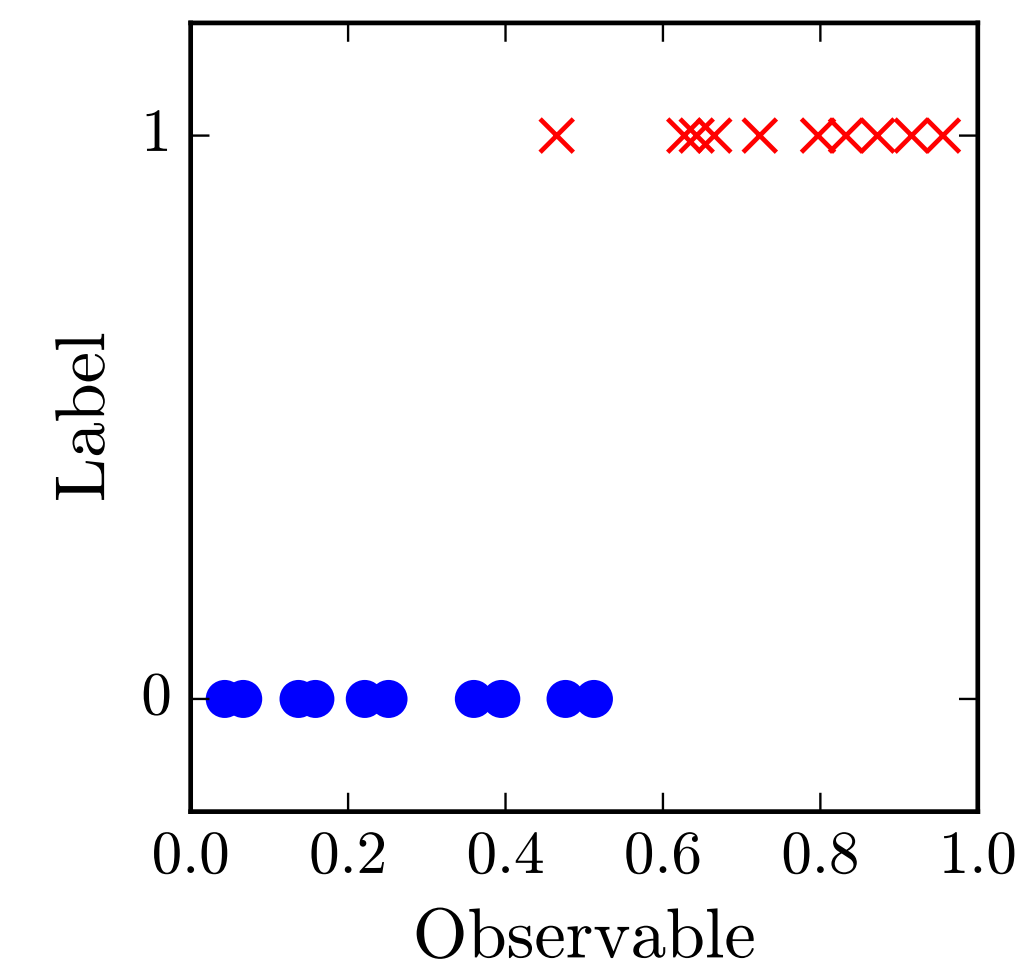


gluon

What is the y-value we are trying to fit/predict?

Define one class as 1 (Signal)

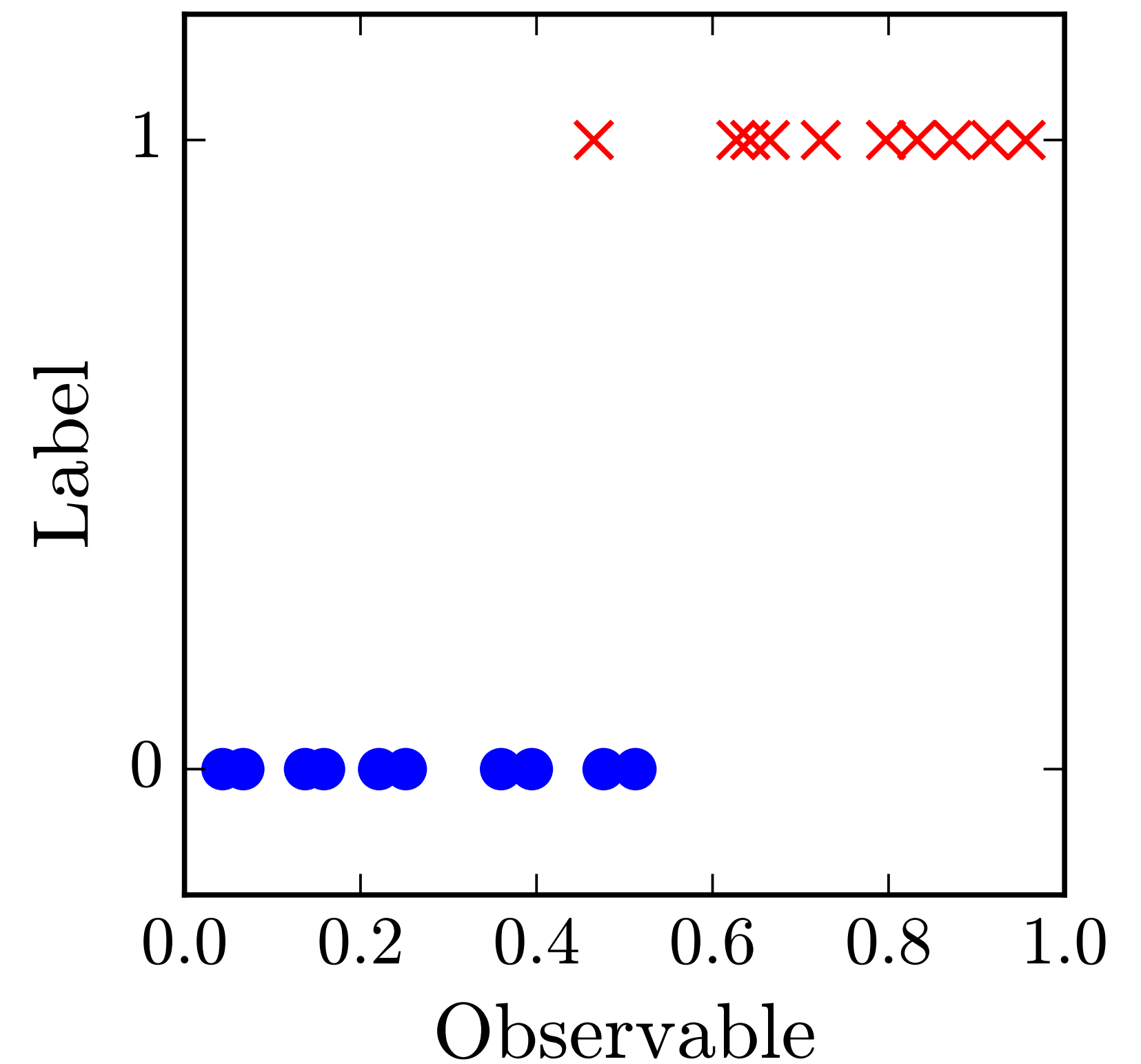
Other class as 0 (Background)



Logistic Regression

What if we are trying to predict a class, not a number?

Define one class as 1 (Signal)
Other class as 0 (Background)

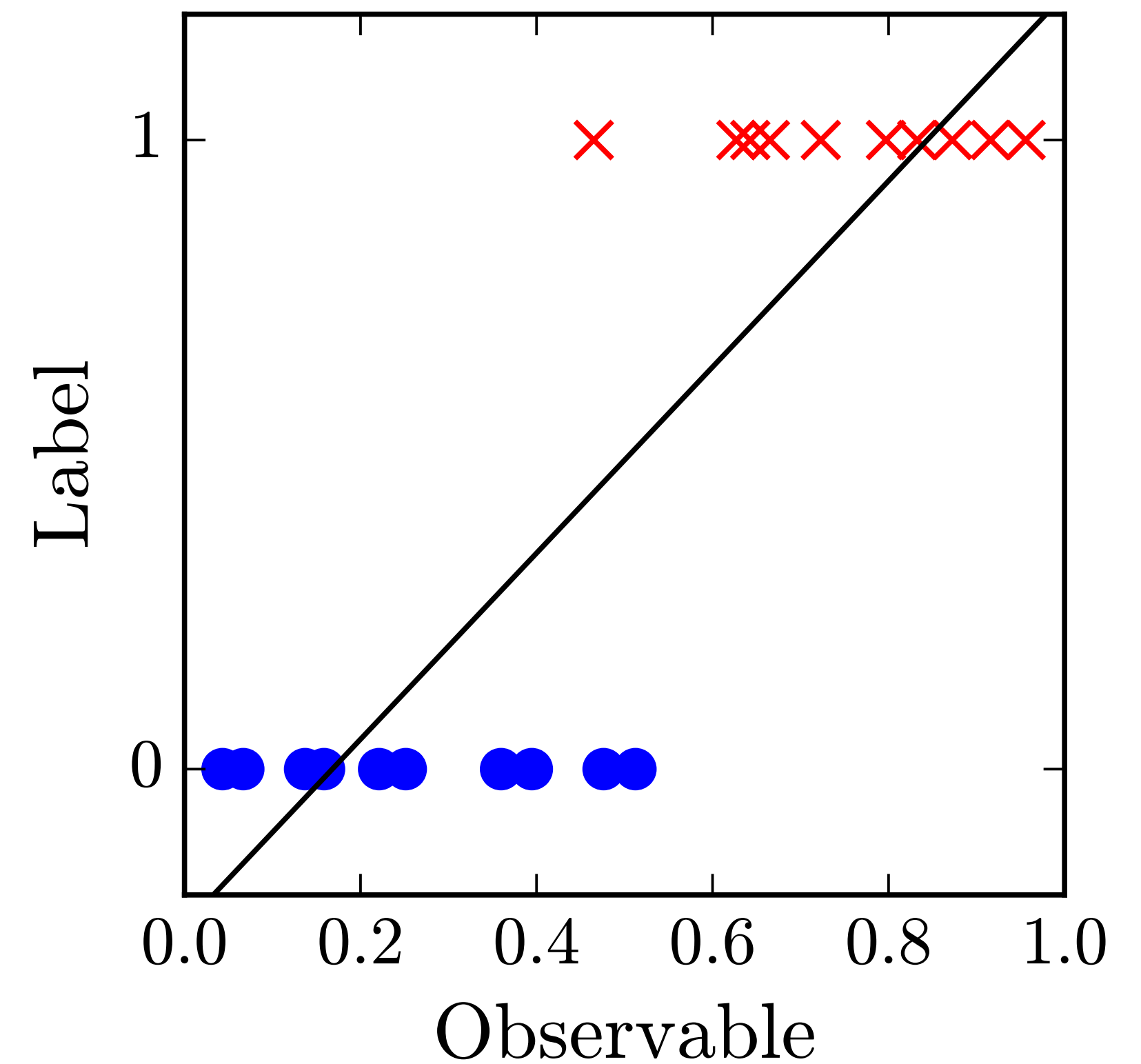


Logistic Regression

What if we are trying to predict a class, not a number?

Define one class as 1 (Signal)
Other class as 0 (Background)

- Linear Fit?

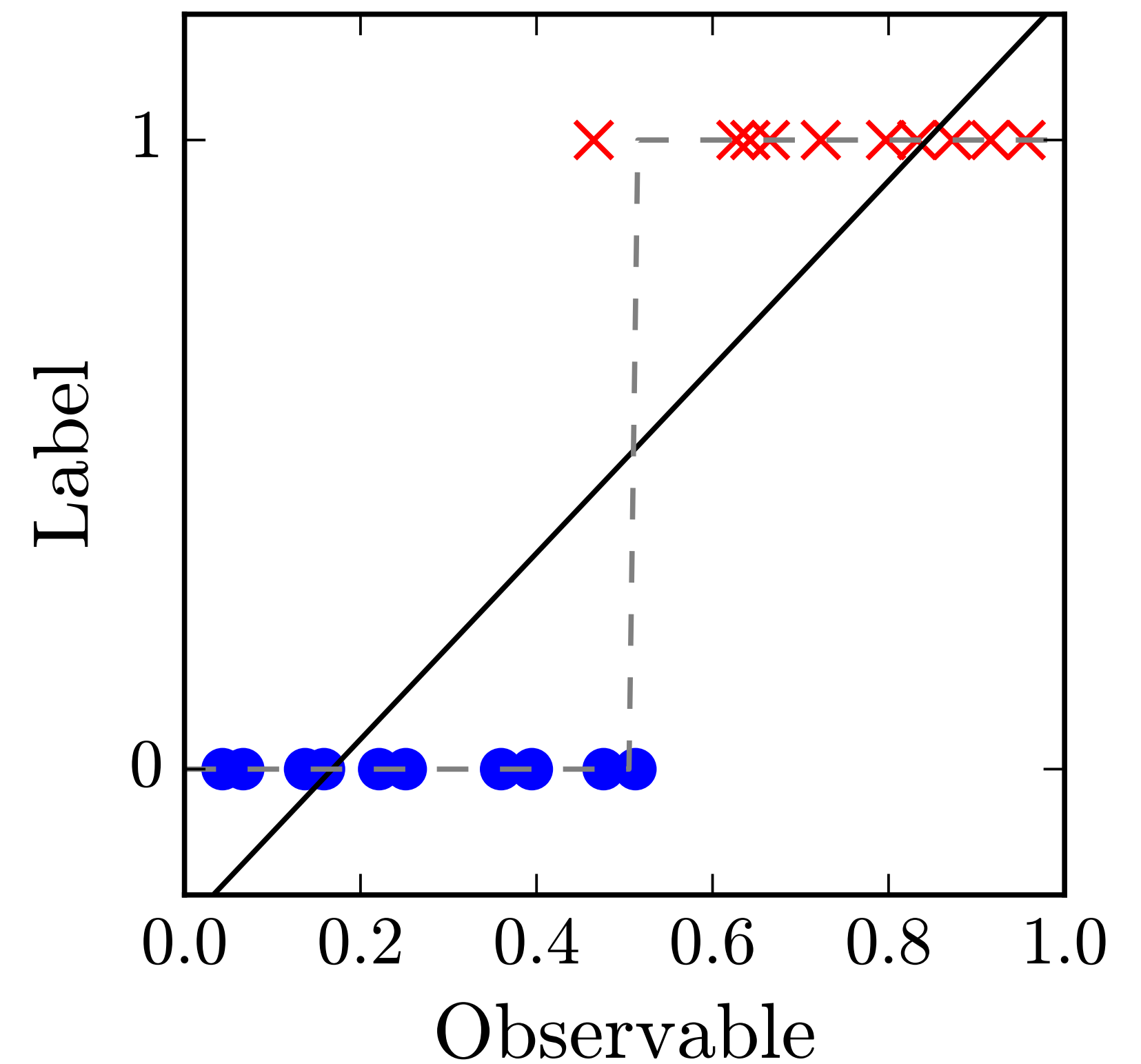


Logistic Regression

What if we are trying to predict a class, not a number?

Define one class as 1 (Signal)
Other class as 0 (Background)

- Linear Fit?
- Round to nearest number?

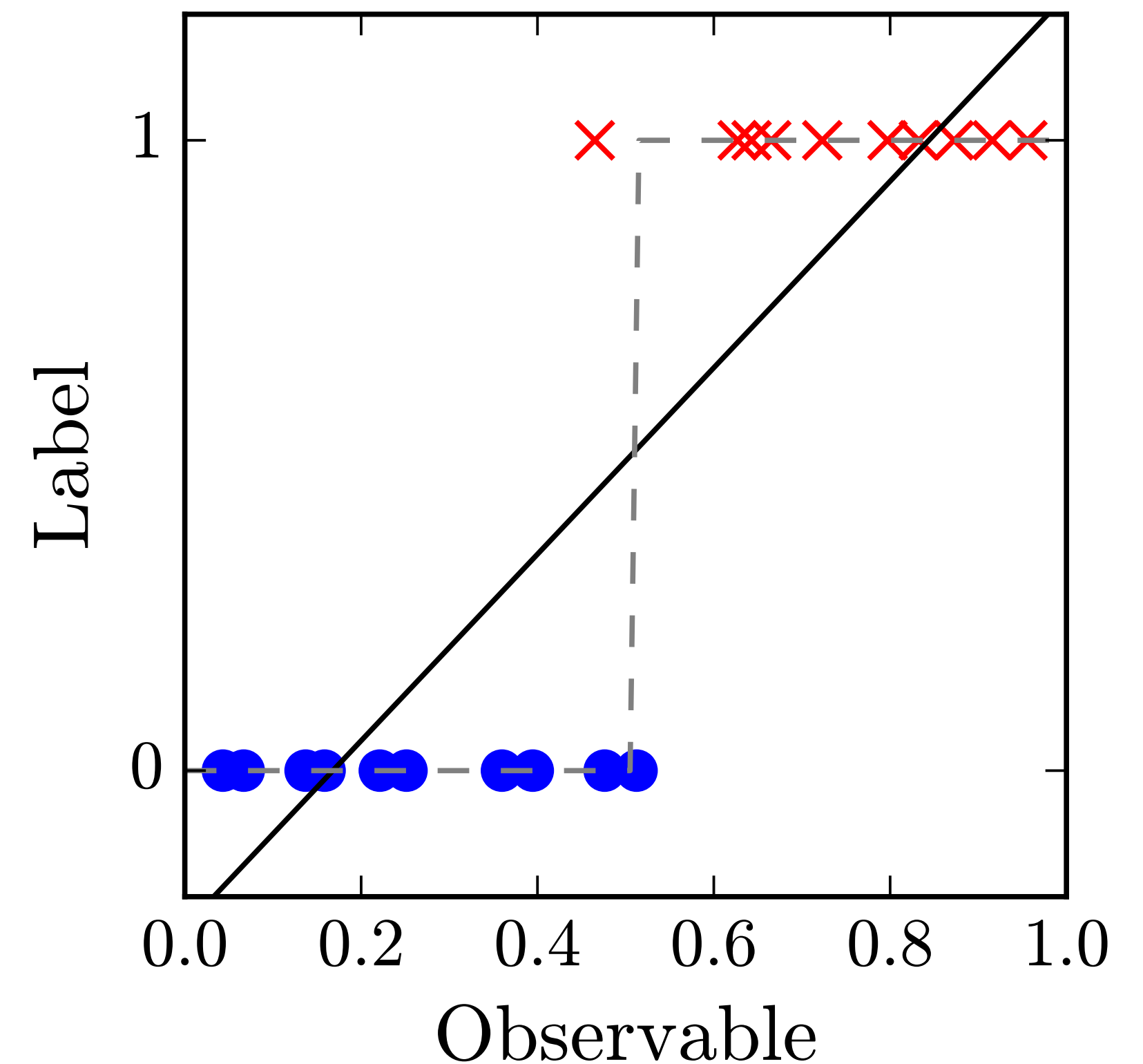


Logistic Regression

What if we are trying to predict a class, not a number?

Define one class as 1 (Signal)
Other class as 0 (Background)

- Linear Fit?
- Round to nearest number?
- How does it generalize to more data?

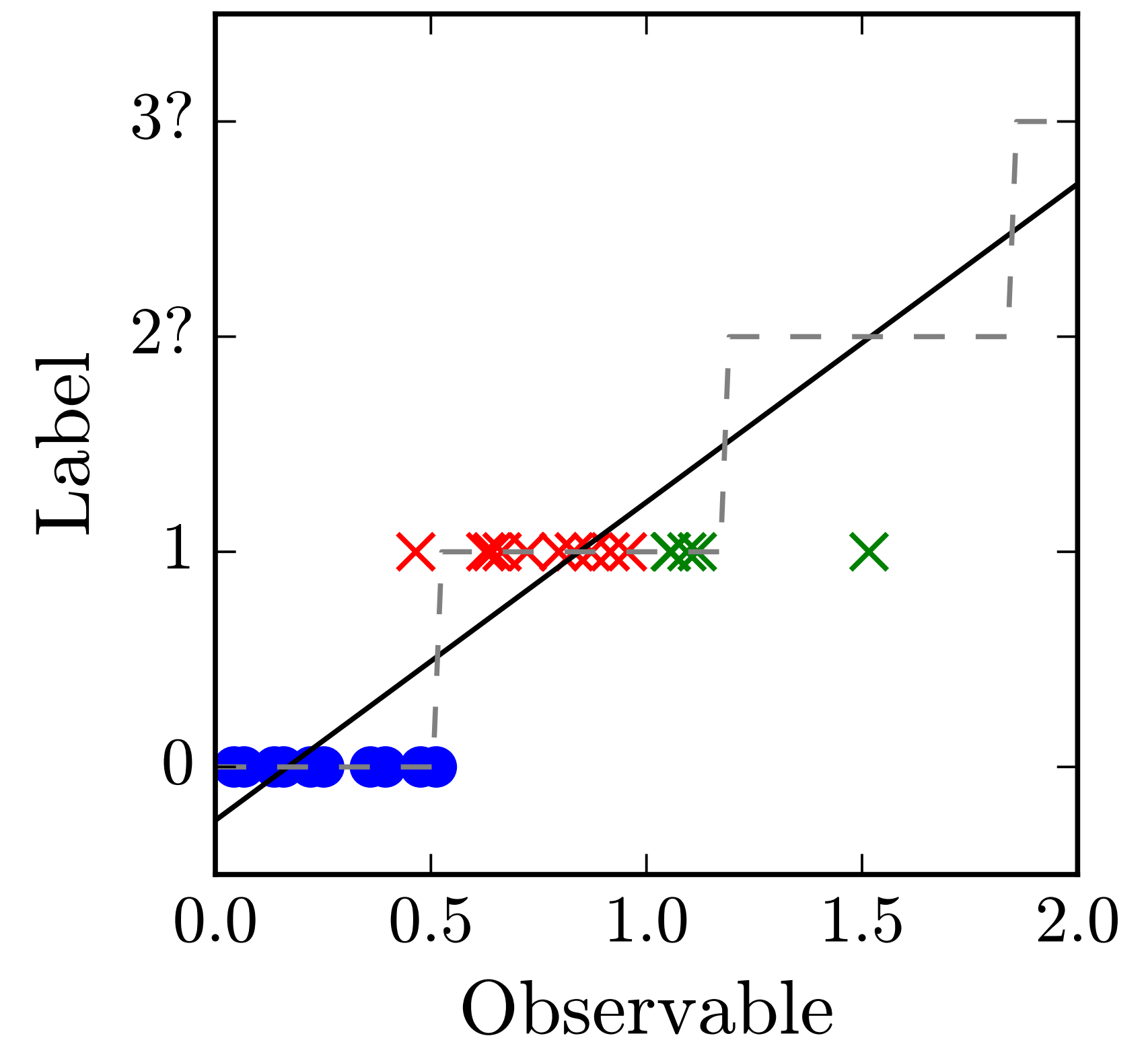


Logistic Regression

What if we are trying to predict a class, not a number?

Define one class as 1 (Signal)
Other class as 0 (Background)

- Linear Fit?
- Round to nearest number?
- How does it generalize to more data?

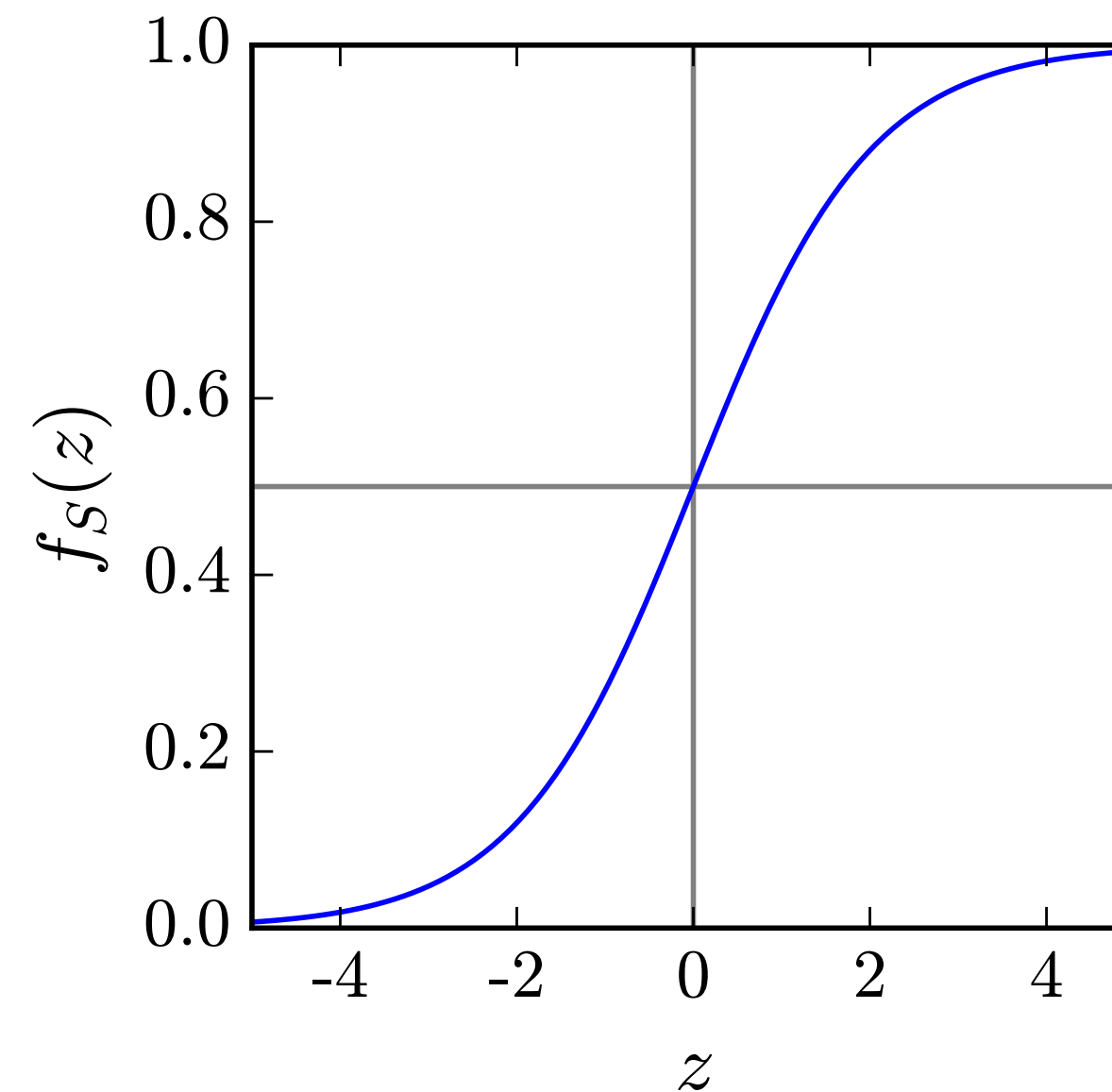


Logistic Regression

What if we are trying to predict a class, not a number?

- Change the shape of function: Logistic/Sigmoid function

$$f_S(z) = \frac{1}{1 + e^{-z}}$$



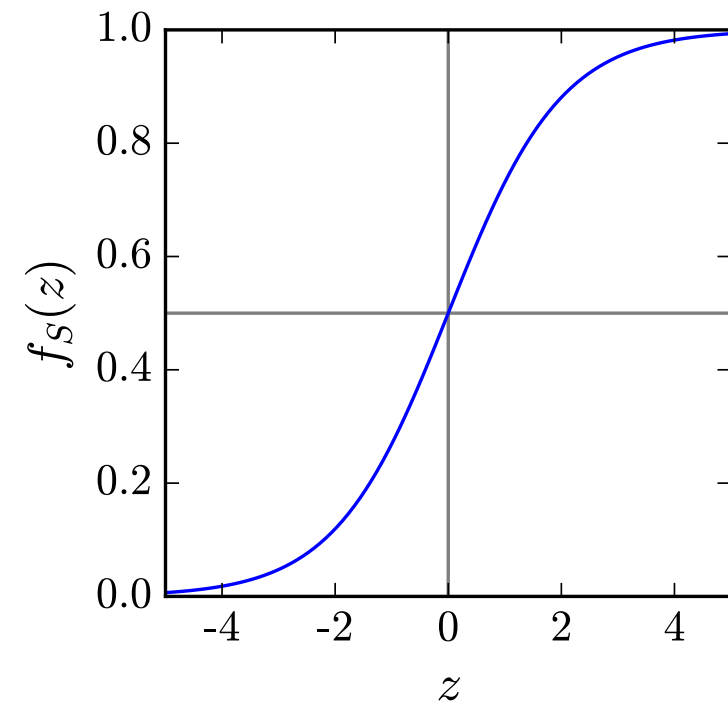
Does not add parameters

- Change the loss function: BCE

$$L(\vec{x}, \vec{y}, \vec{a}) = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log \left(f_S(p(x, a)) \right) + (1 - y_i) \log \left(1 - f_S(p(x, a)) \right) \right)$$

Logistic Regression

What if we are trying to predict a class, not a number?

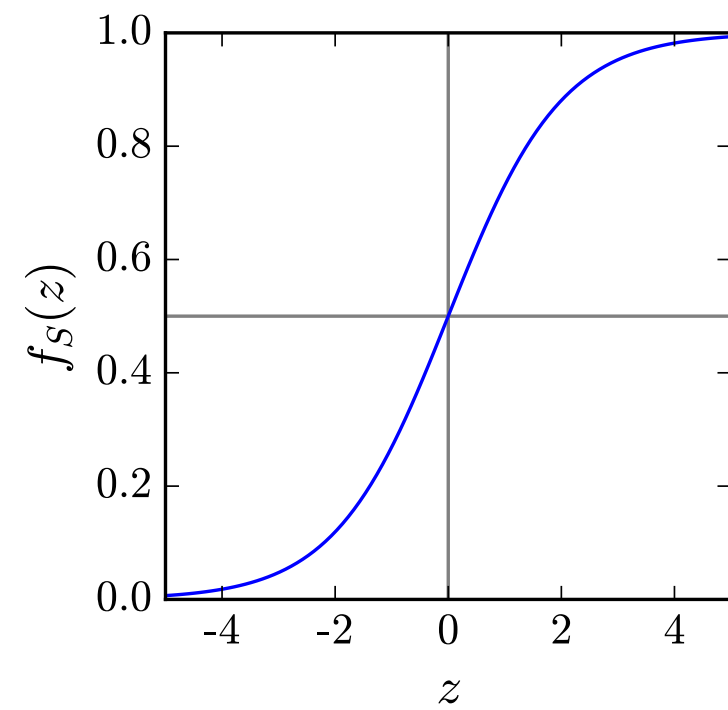


$$L(\vec{x}, \vec{y}, \vec{a}) = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log \left(f_S(p(x, a)) \right) + (1 - y_i) \log \left(1 - f_S(p(x, a)) \right) \right)$$

$$f_S(z) = \frac{1}{1 + e^{-z}} \quad z = p(x, a)$$

Logistic Regression

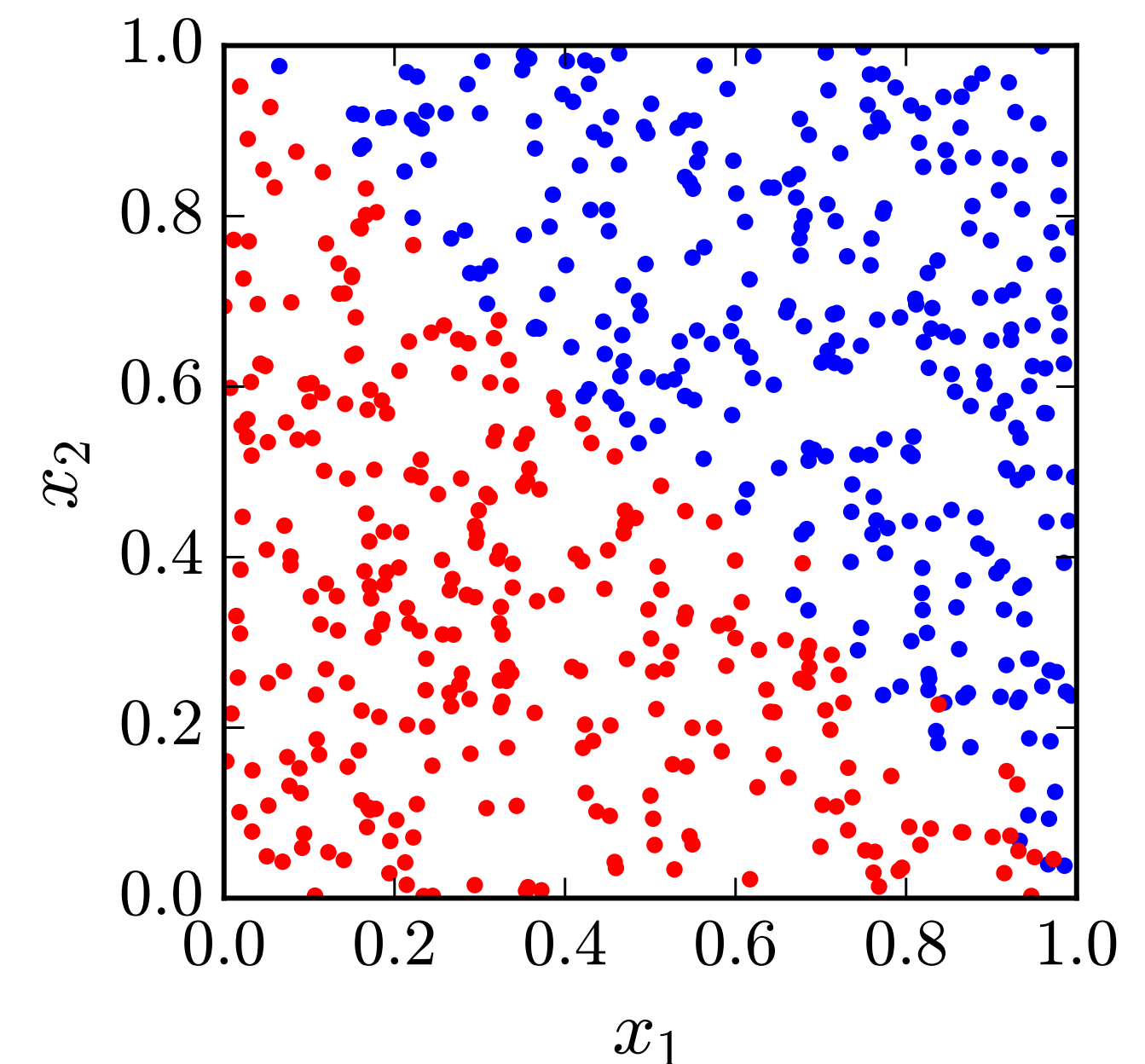
What if we are trying to predict a class, not a number?



$$L(\vec{x}, \vec{y}, \vec{a}) = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log \left(f_S(p(x, a)) \right) + (1 - y_i) \log \left(1 - f_S(p(x, a)) \right) \right)$$

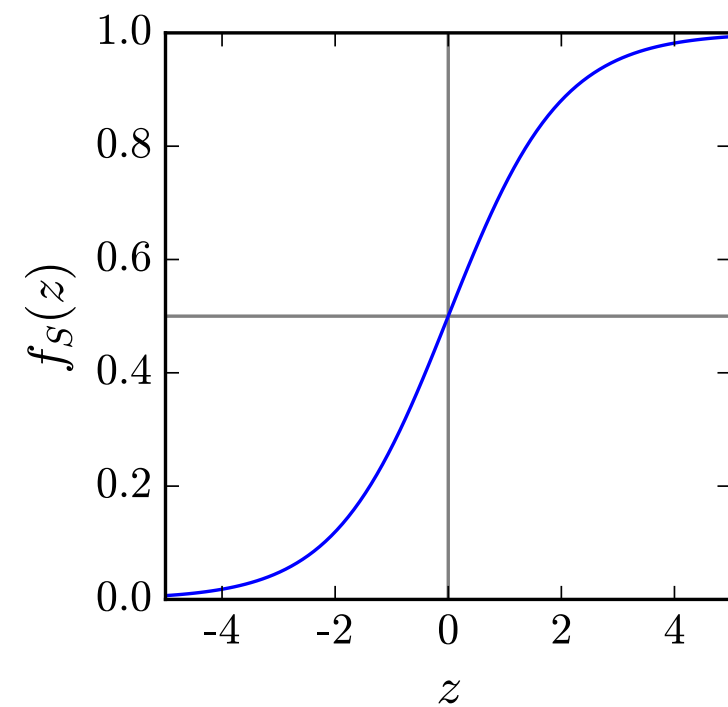
$$f_S(z) = \frac{1}{1 + e^{-z}} \quad z = p(x, a)$$

What is $p(x, a)$?



Logistic Regression

What if we are trying to predict a class, not a number?

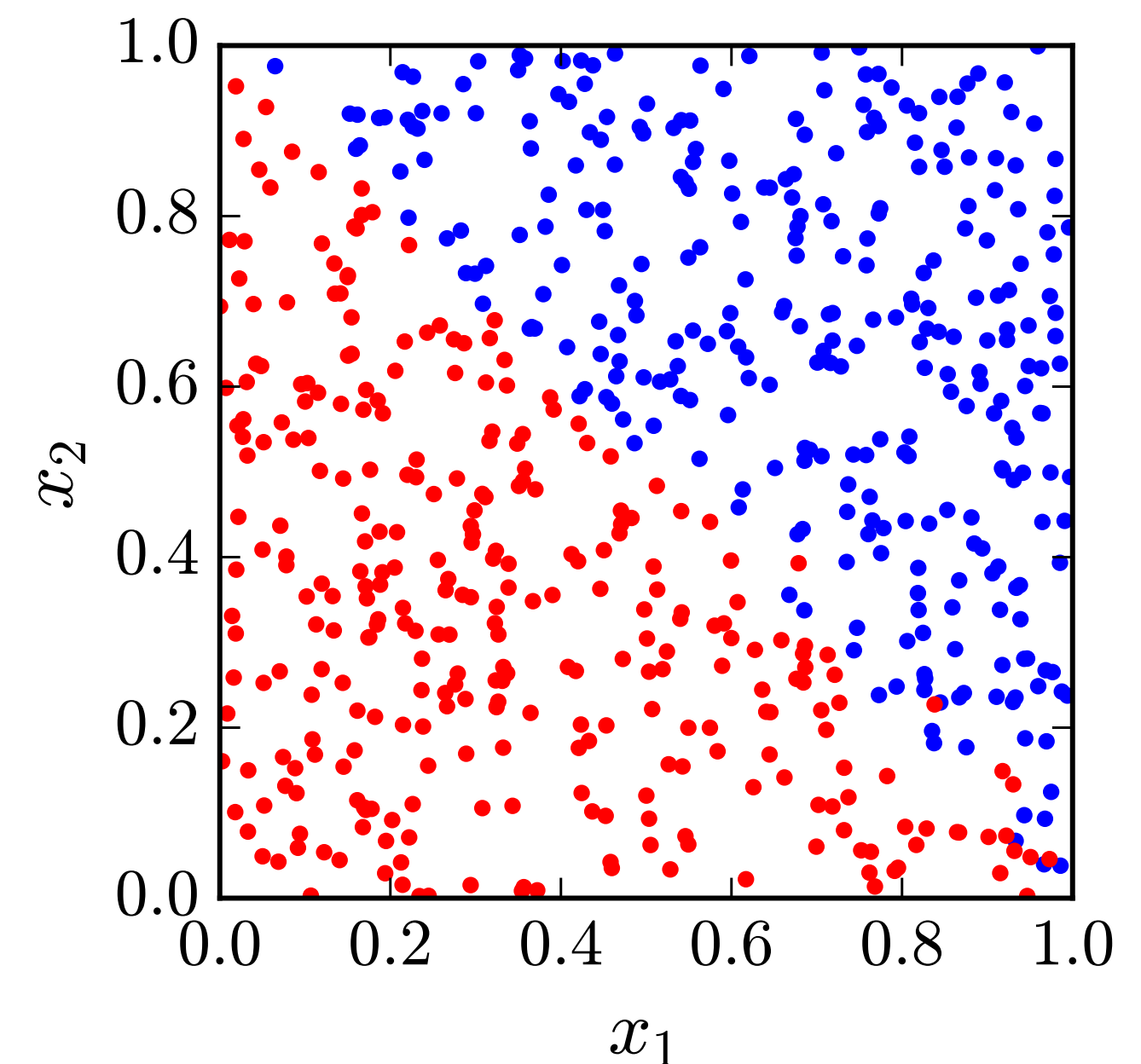
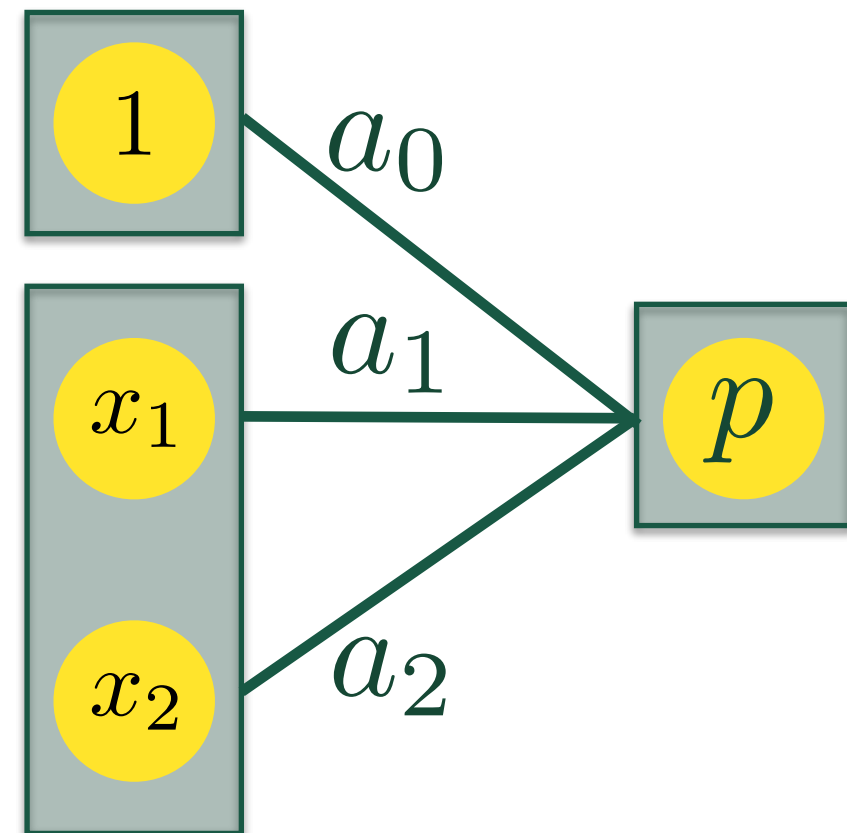


$$L(\vec{x}, \vec{y}, \vec{a}) = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log \left(f_S(p(x, a)) \right) + (1 - y_i) \log \left(1 - f_S(p(x, a)) \right) \right)$$

$$f_S(z) = \frac{1}{1 + e^{-z}} \quad z = p(x, a)$$

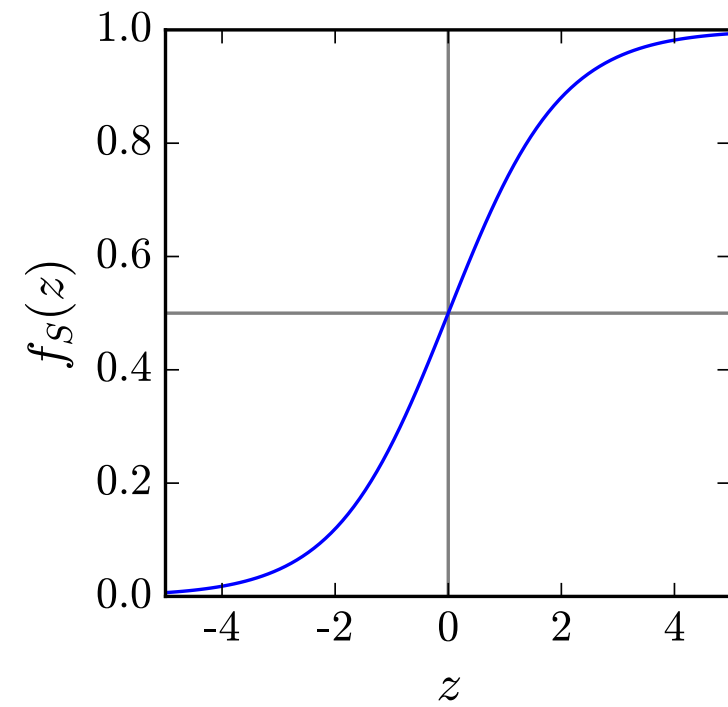
What is $p(x, a)$?

$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$



Logistic Regression

What if we are trying to predict a class, not a number?

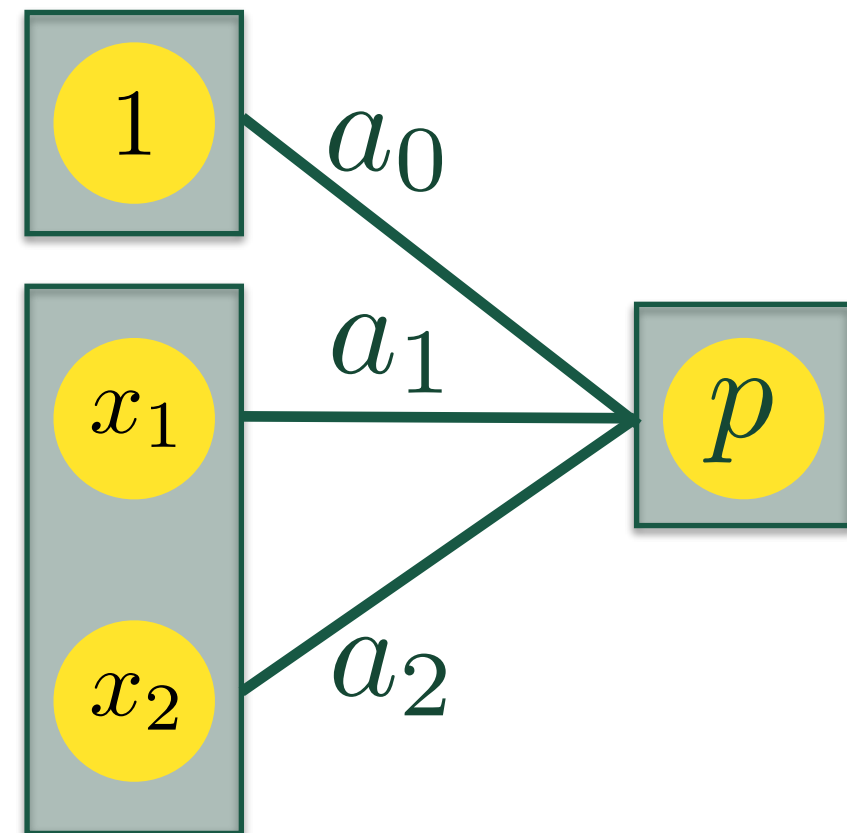


$$L(\vec{x}, \vec{y}, \vec{a}) = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log \left(f_S(p(x, a)) \right) + (1 - y_i) \log \left(1 - f_S(p(x, a)) \right) \right)$$

$$f_S(z) = \frac{1}{1 + e^{-z}} \quad z = p(x, a)$$

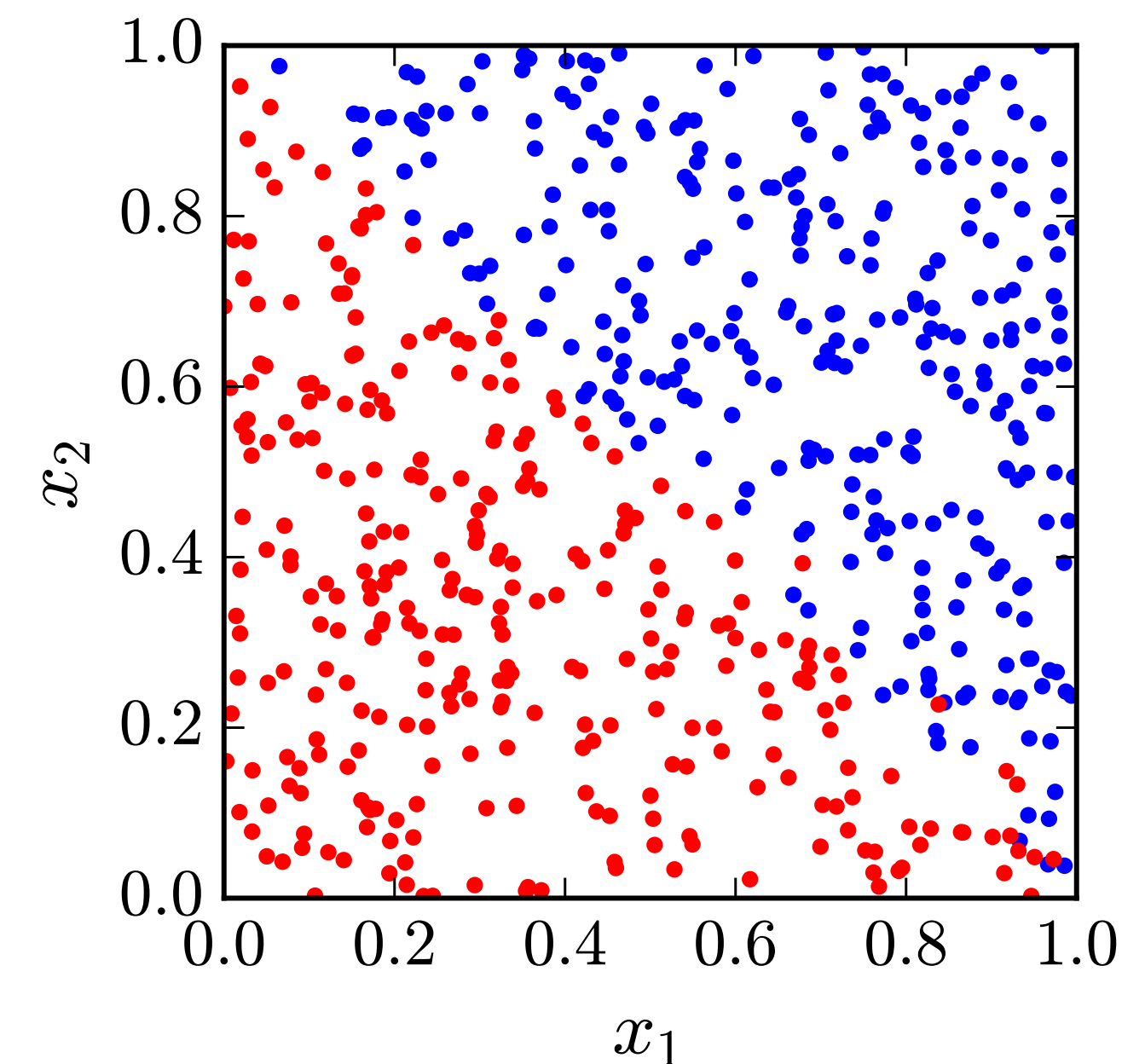
What is $p(x, a)$?

$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$



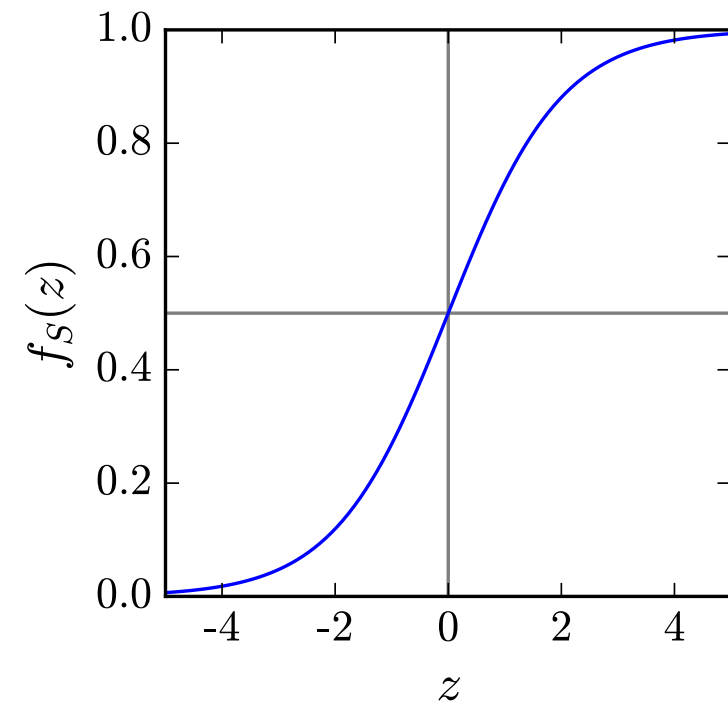
Minimize the loss with respect to \vec{a}

Boundary at $p(x, a) = 0$



Logistic Regression

What if we are trying to predict a class, not a number?

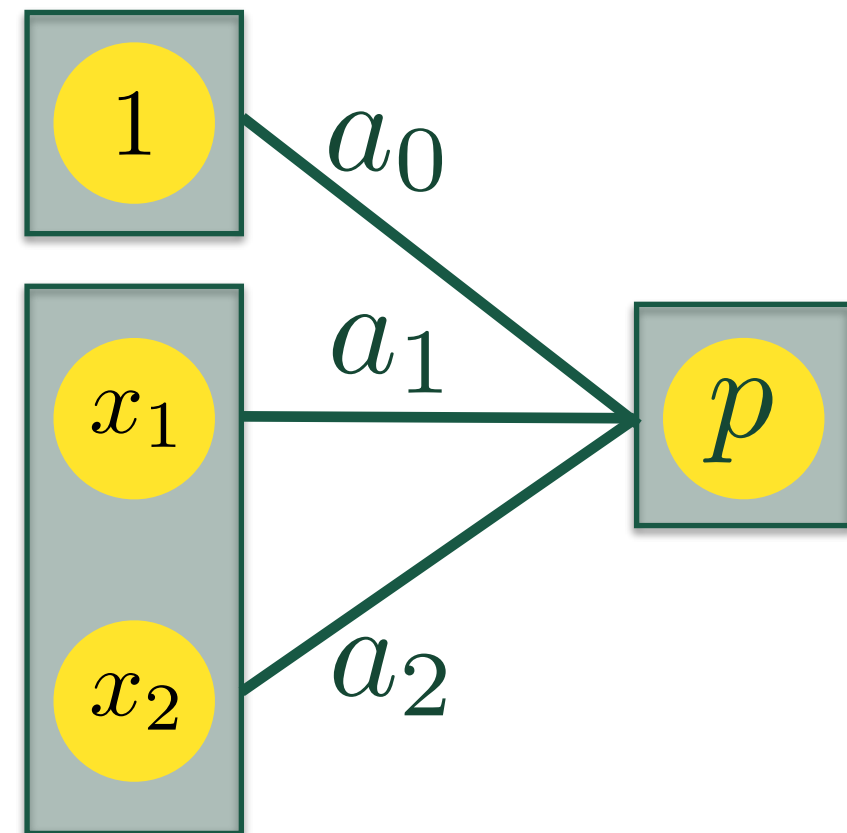


$$L(\vec{x}, \vec{y}, \vec{a}) = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log \left(f_S(p(x, a)) \right) + (1 - y_i) \log \left(1 - f_S(p(x, a)) \right) \right)$$

$$f_S(z) = \frac{1}{1 + e^{-z}} \quad z = p(x, a)$$

What is $p(x, a)$?

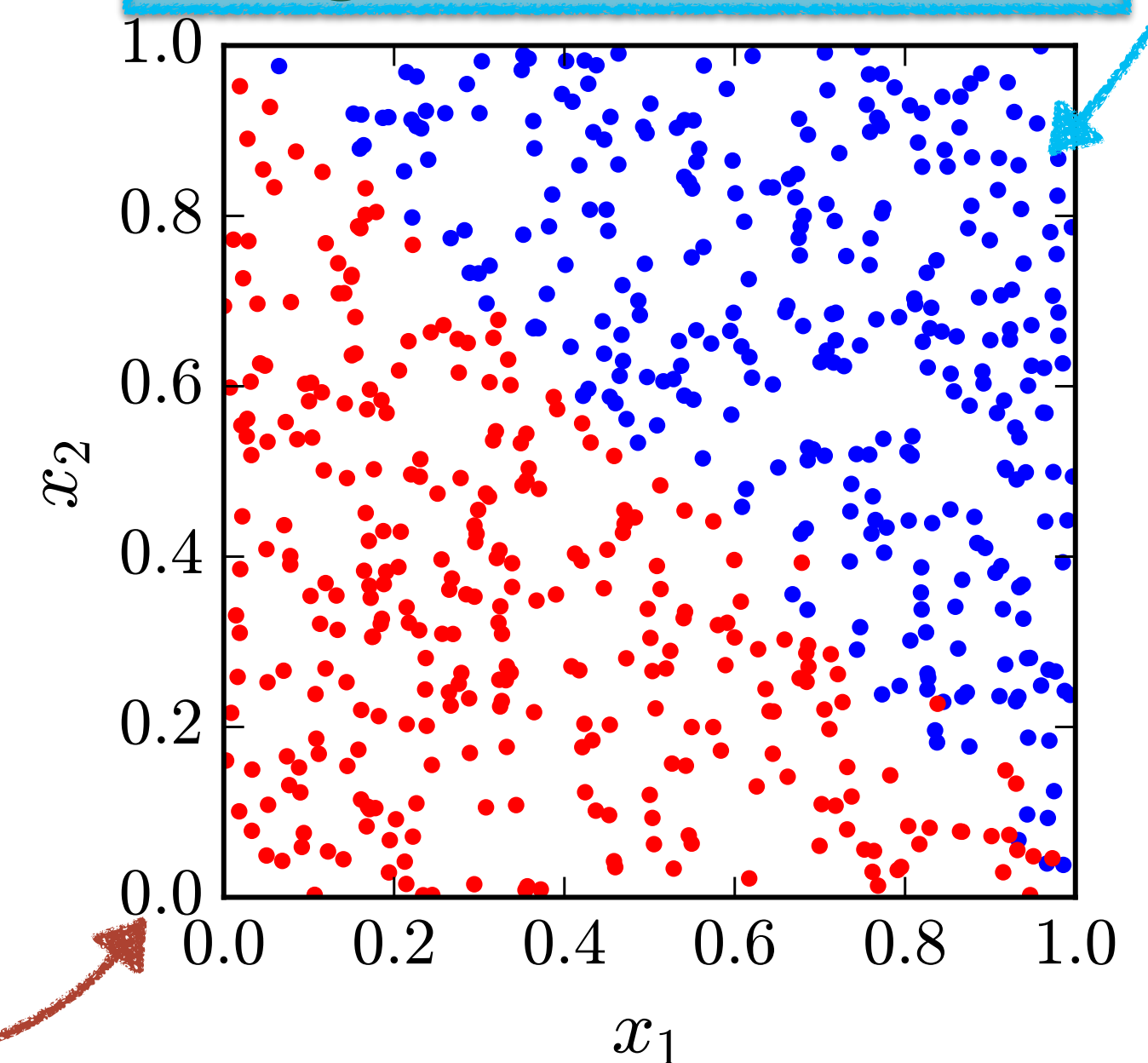
$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$



Minimize the loss with respect to \vec{a}

Boundary at $p(x, a) = 0$

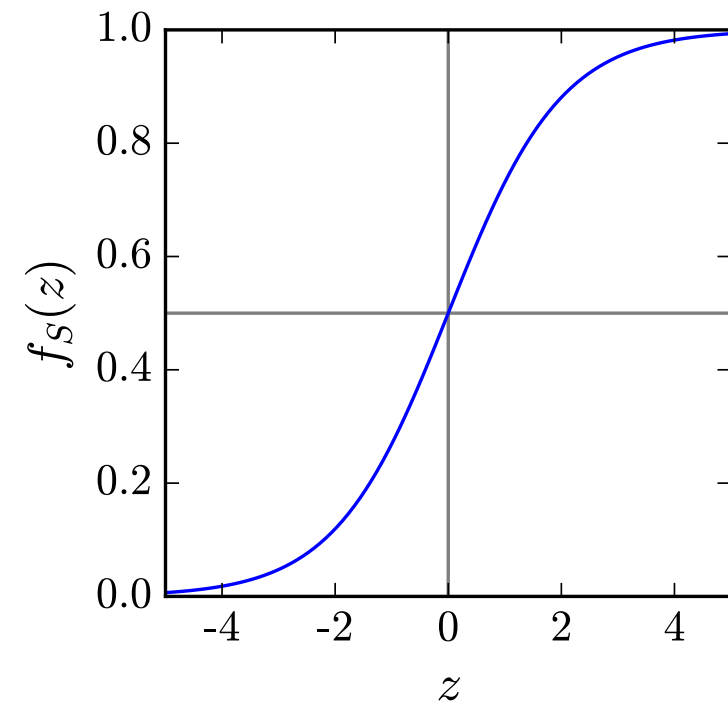
Large + values of p



Large - values of p

Logistic Regression

What if we are trying to predict a class, not a number?

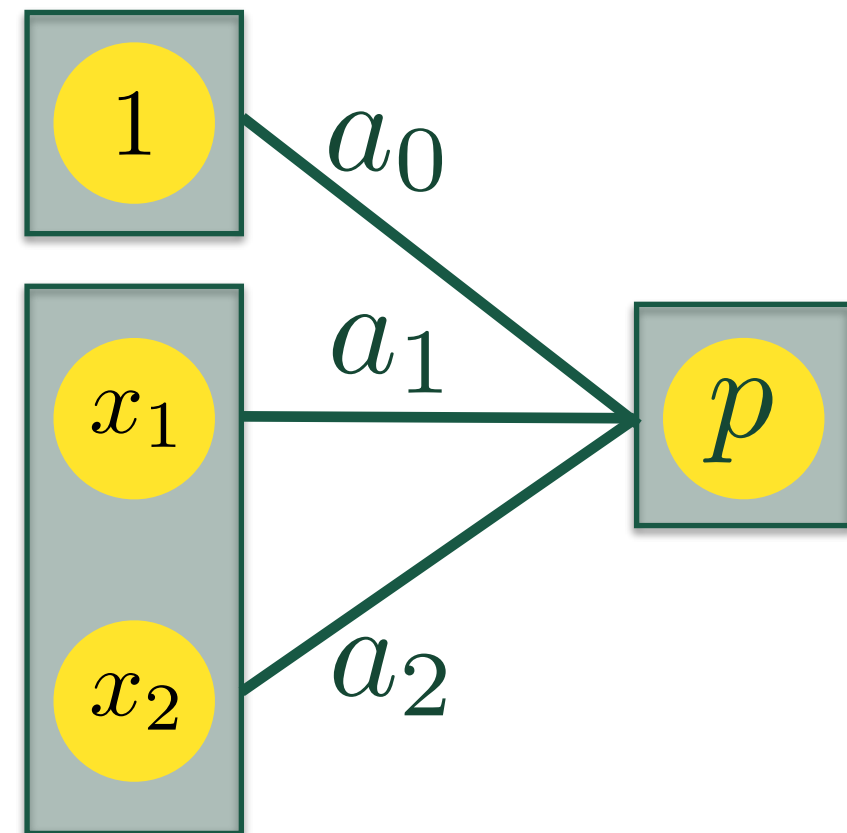


$$L(\vec{x}, \vec{y}, \vec{a}) = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log \left(f_S(p(x, a)) \right) + (1 - y_i) \log \left(1 - f_S(p(x, a)) \right) \right)$$

$$f_S(z) = \frac{1}{1 + e^{-z}} \quad z = p(x, a)$$

What is $p(x, a)$?

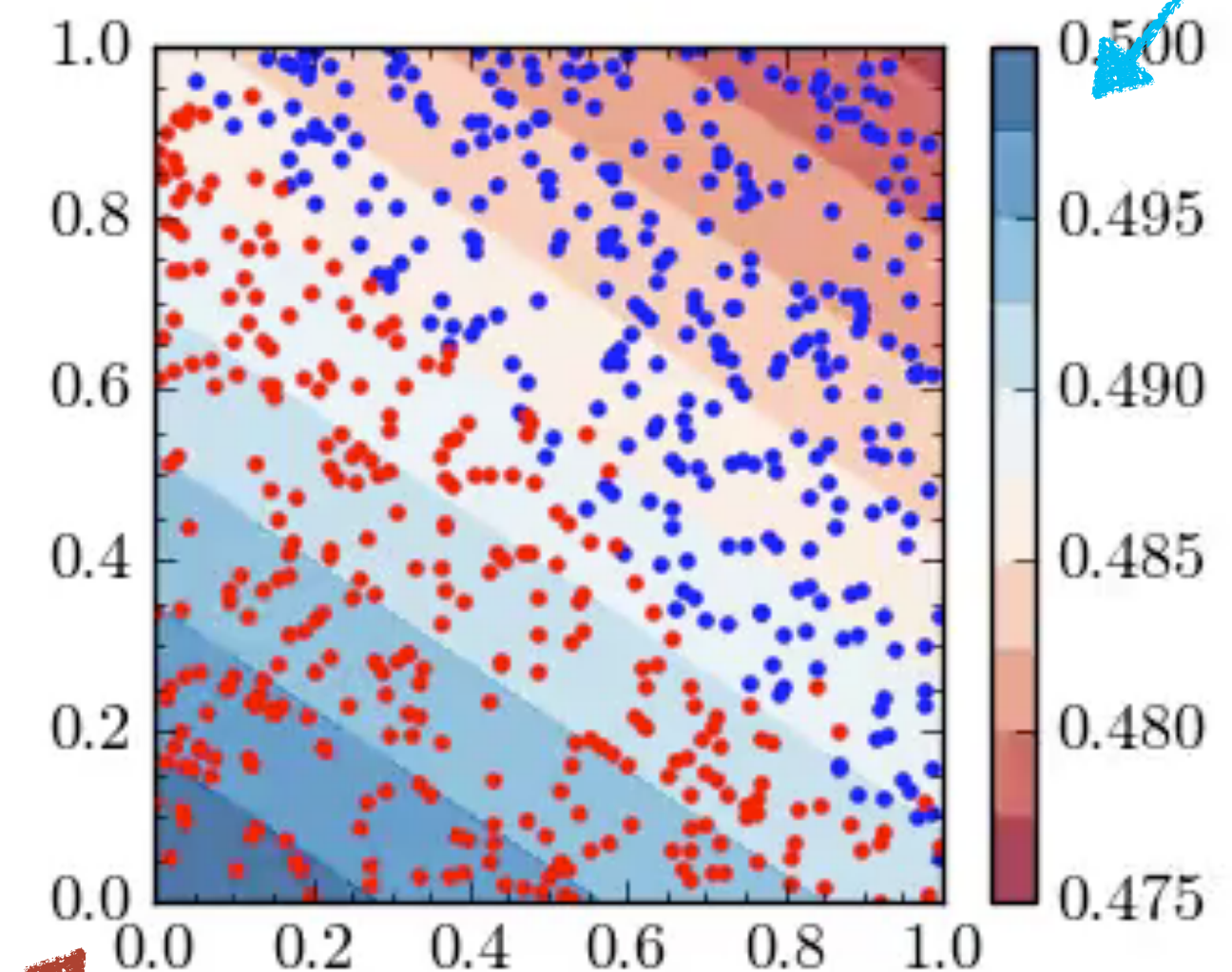
$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$



Minimize the loss with respect to \vec{a}

Boundary at $p(x, a) = 0$

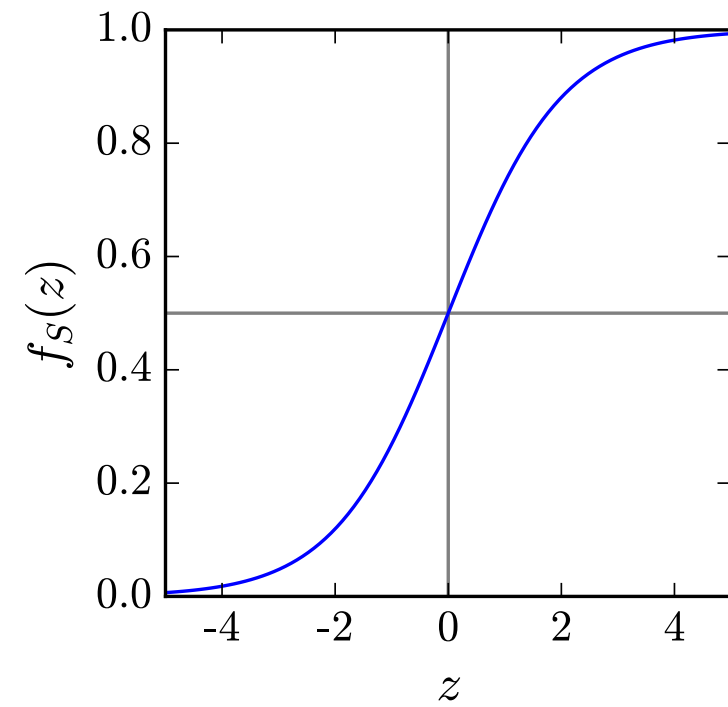
Large + values of p



Large - values of p

Logistic Regression

What if we are trying to predict a class, not a number?

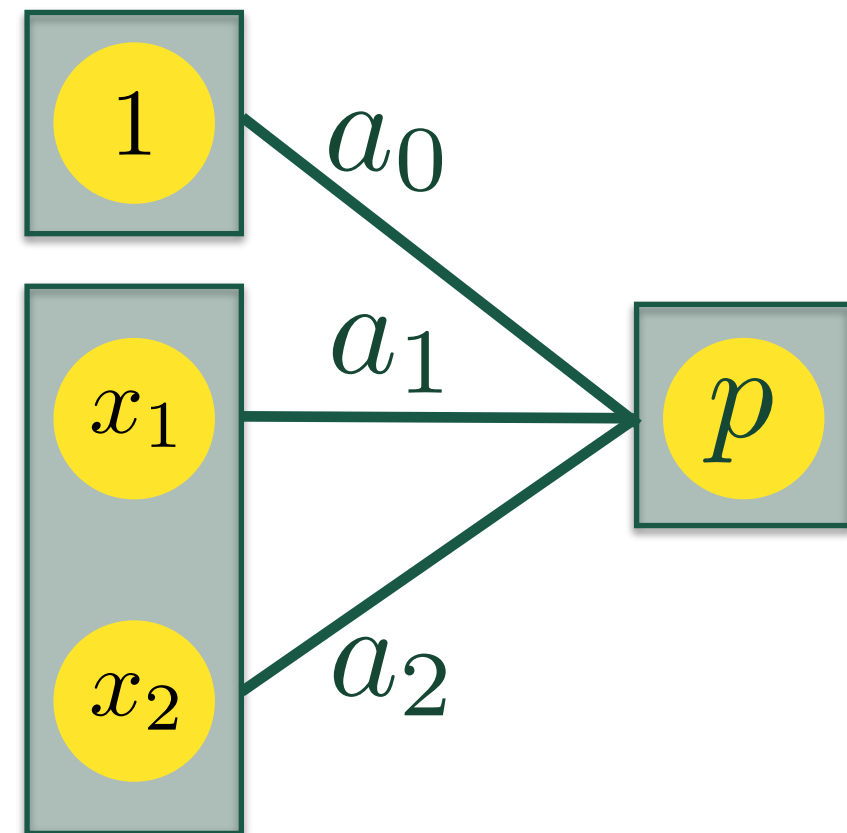


$$L(\vec{x}, \vec{y}, \vec{a}) = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log \left(f_S(p(x, a)) \right) + (1 - y_i) \log \left(1 - f_S(p(x, a)) \right) \right)$$

$$f_S(z) = \frac{1}{1 + e^{-z}} \quad z = p(x, a)$$

What is $p(x, a)$?

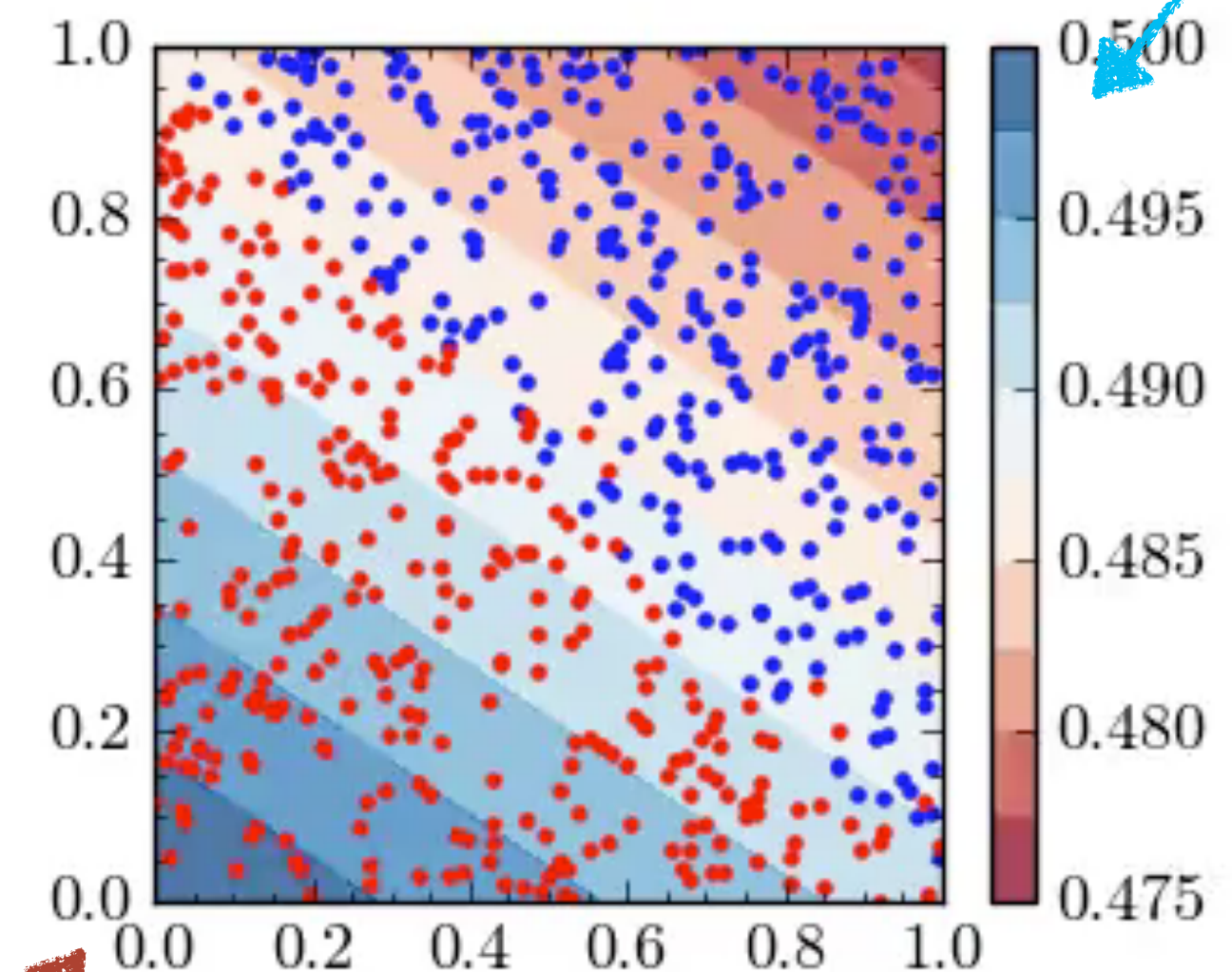
$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$



Minimize the loss with respect to \vec{a}

Boundary at $p(x, a) = 0$

Large + values of p

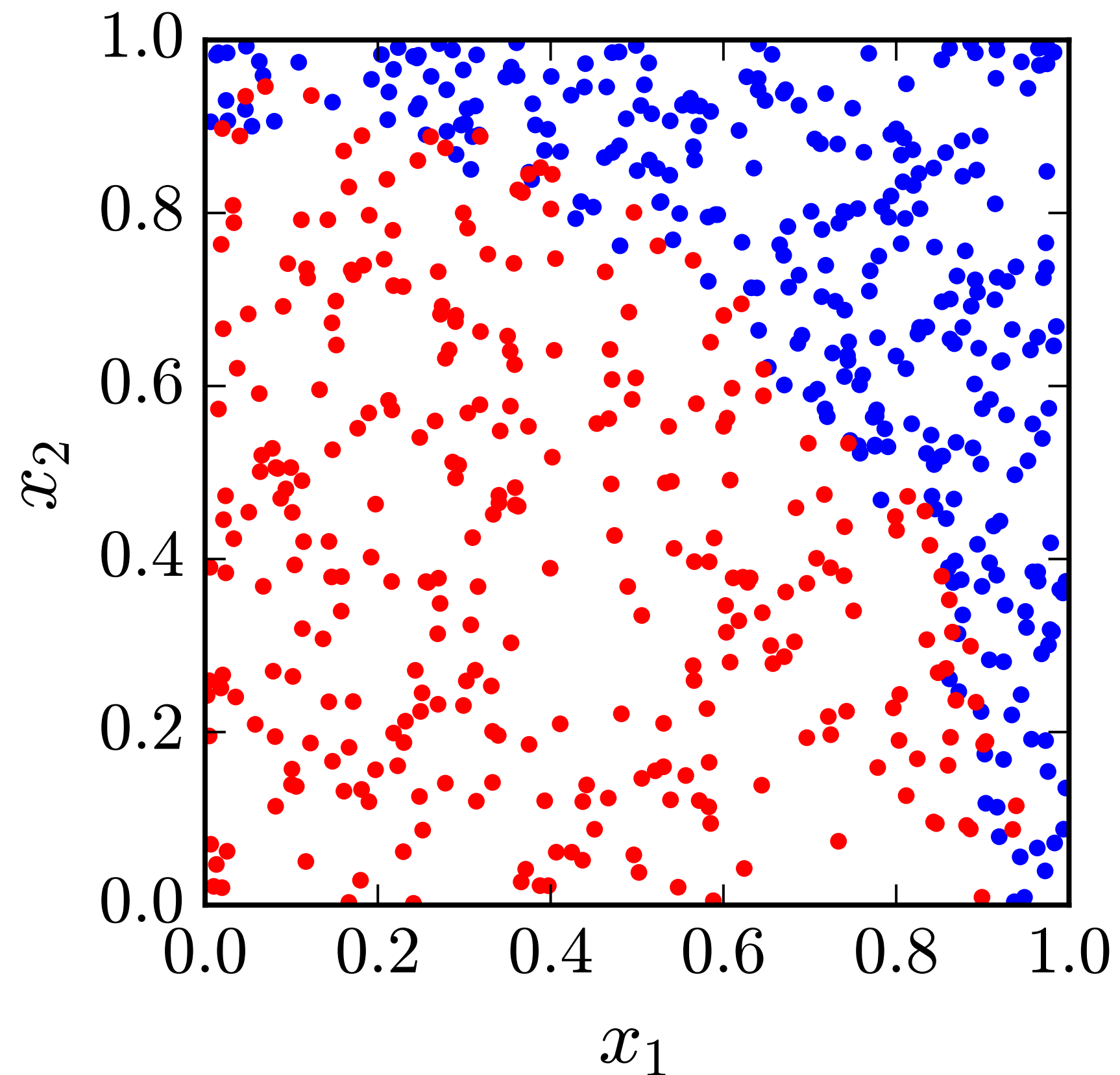


Large - values of p

Logistic Regression

What if there is a shape in the data?

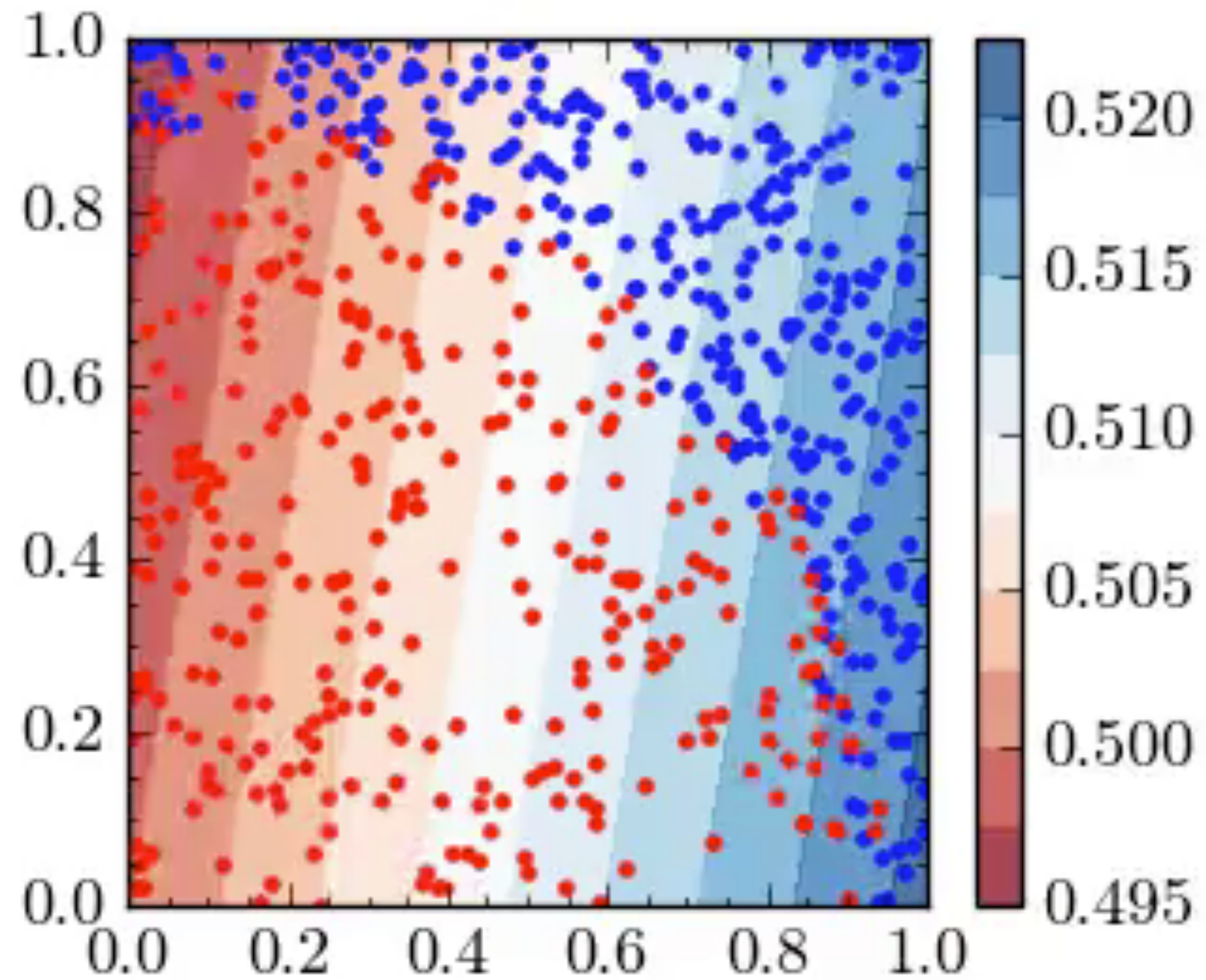
$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$



Logistic Regression

What if there is a shape in the data?

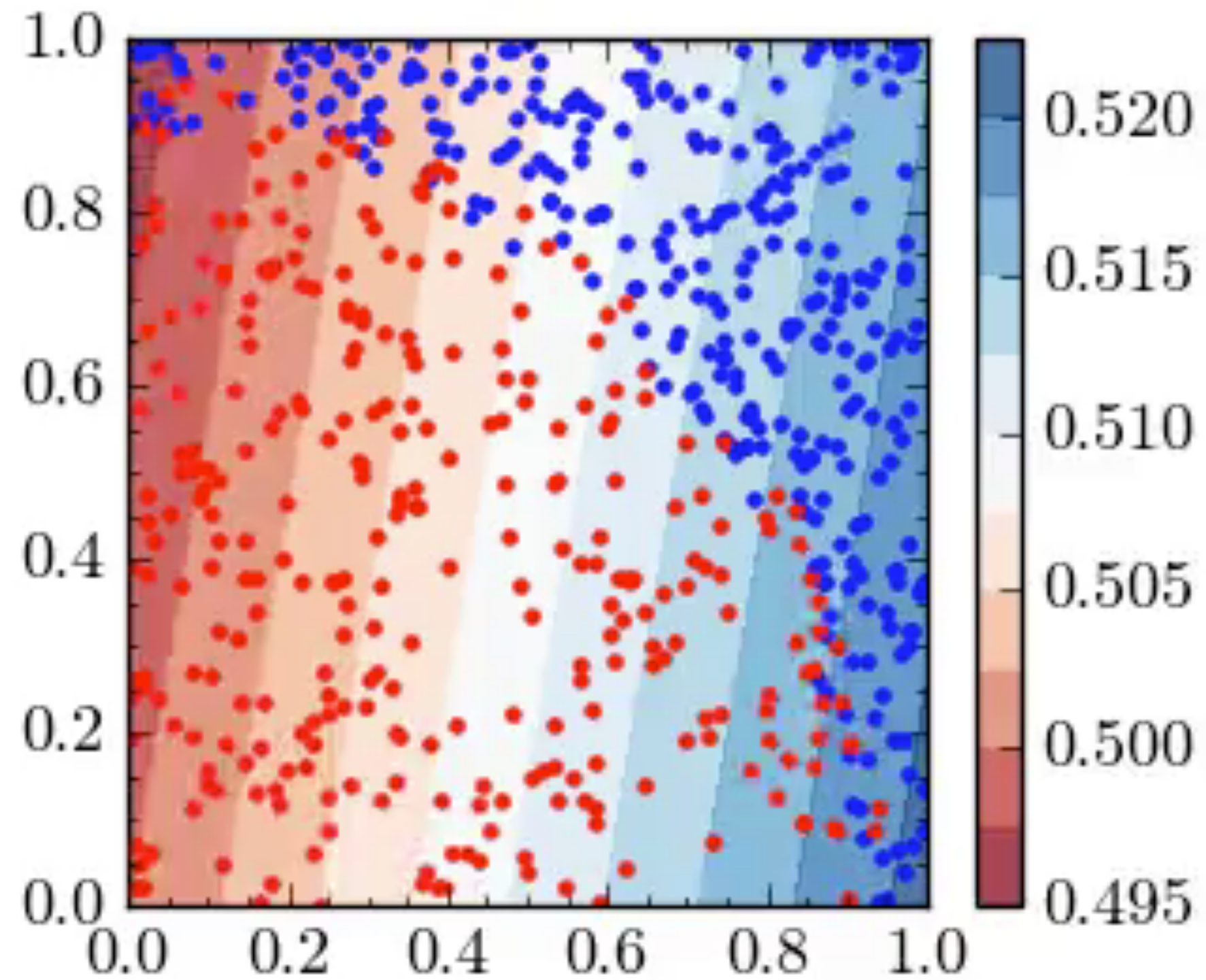
$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$



Logistic Regression

What if there is a shape in the data?

$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$



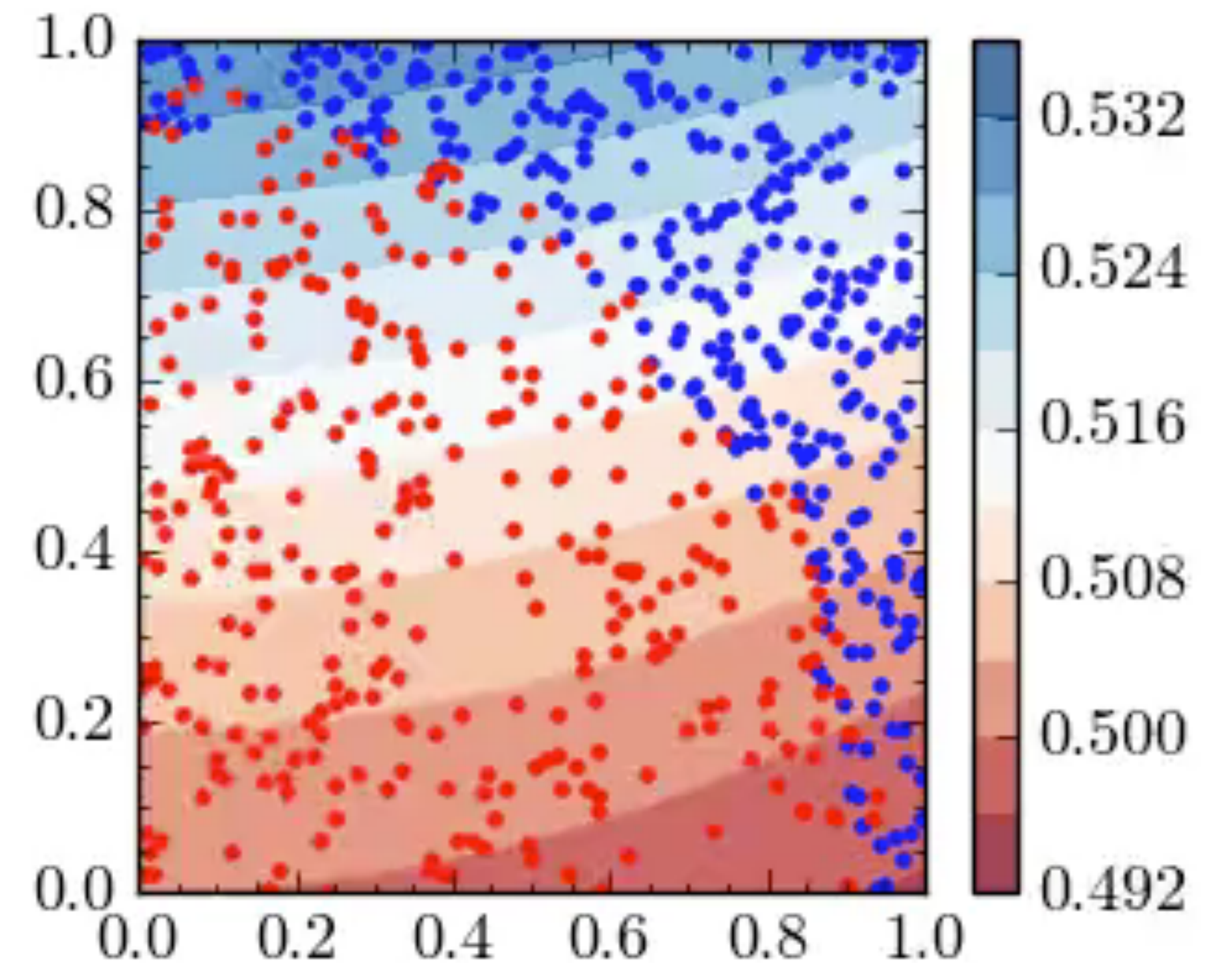
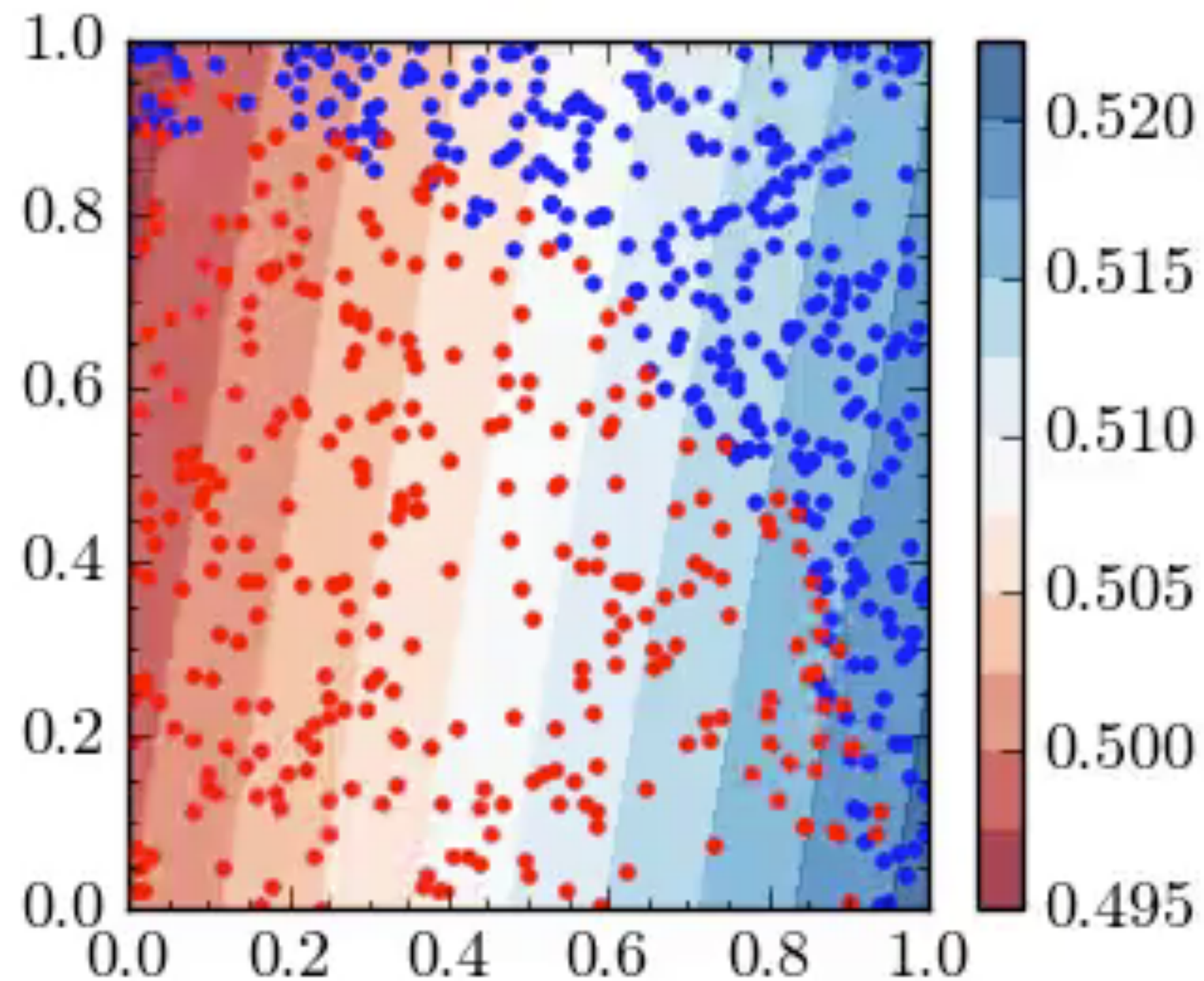
Logistic Regression

What if there is a shape in the data?

$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$

$$p(x, a) = a_0 + a_1 x_1 + a_2 x_2$$

$$+ a_3 x_1^2 + a_4 x_2^2 + a_5 x_1 x_2$$



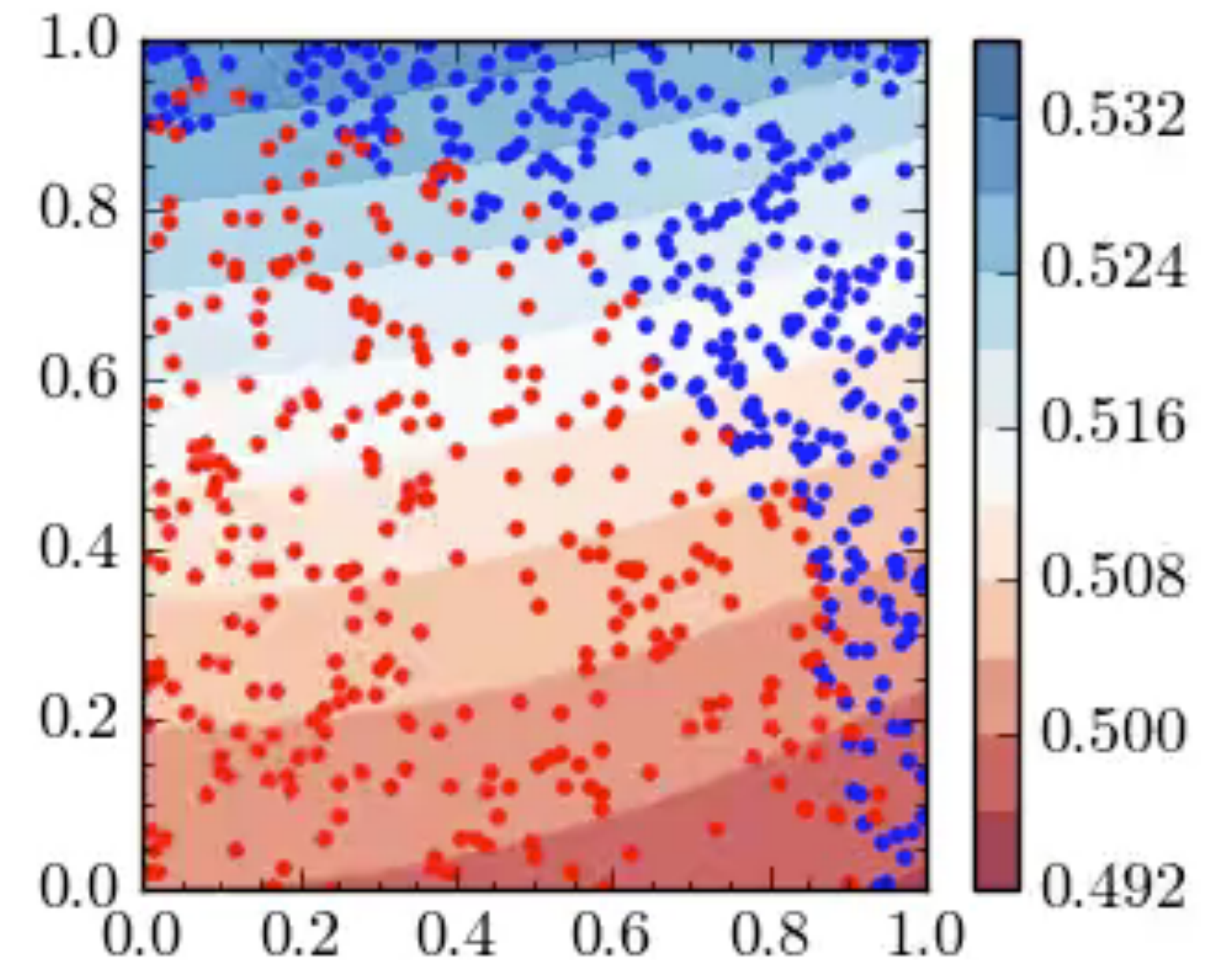
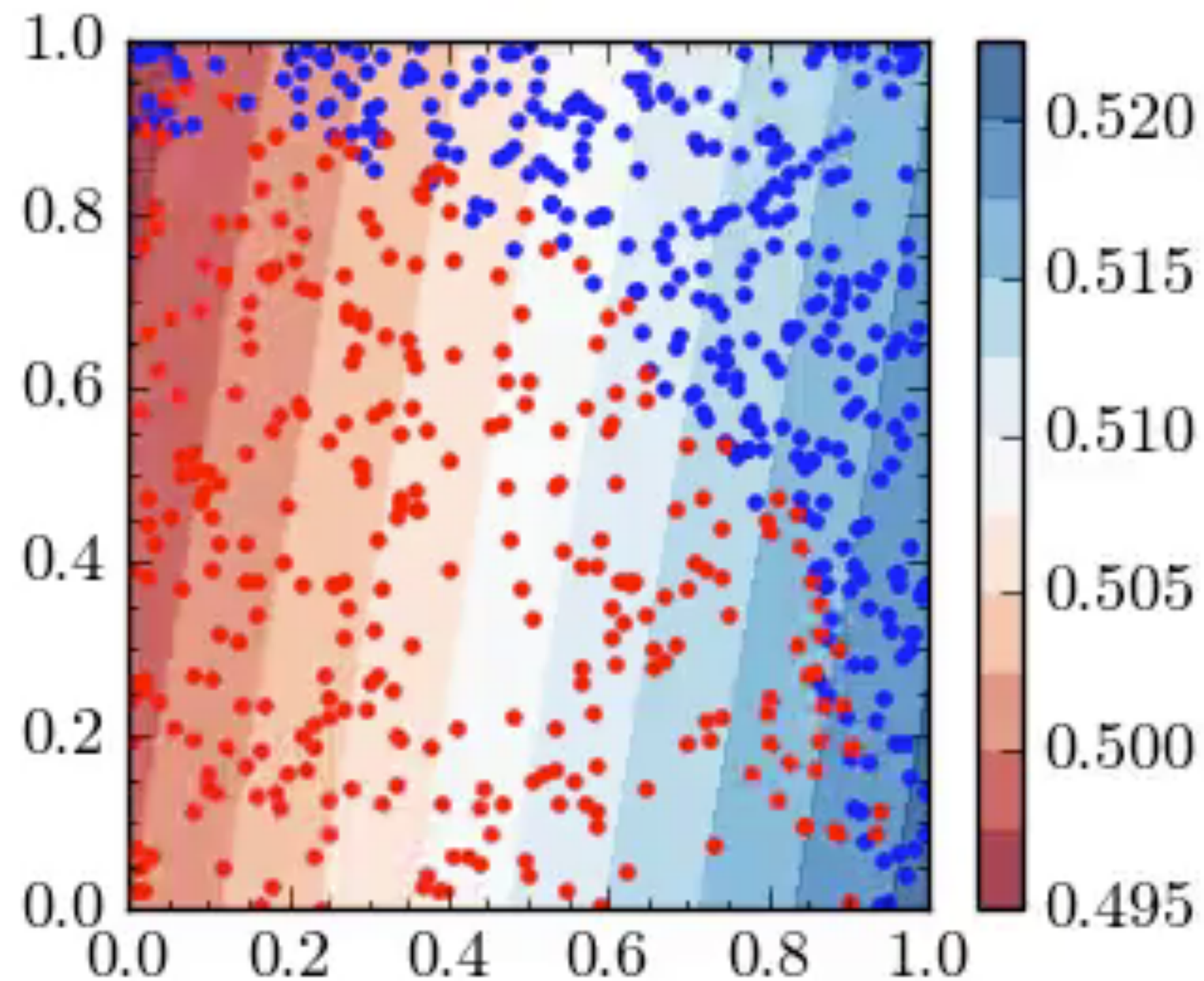
Logistic Regression

What if there is a shape in the data?

$$p(x, a) = a_0 + x_1 a_1 + x_2 a_2$$

$$p(x, a) = a_0 + a_1 x_1 + a_2 x_2$$

$$+ a_3 x_1^2 + a_4 x_2^2 + a_5 x_1 x_2$$



Regression Review

1. Can use nearly the same process for fitting a curve (predicting a number) or classification
2. Minimize a defined cost function
3. Easy to add parameters if shape is unknown — worry about over-fitting
4. If many inputs and complicated shapes, number of parameters necessary grows very quickly

Why use more complicated algorithms?

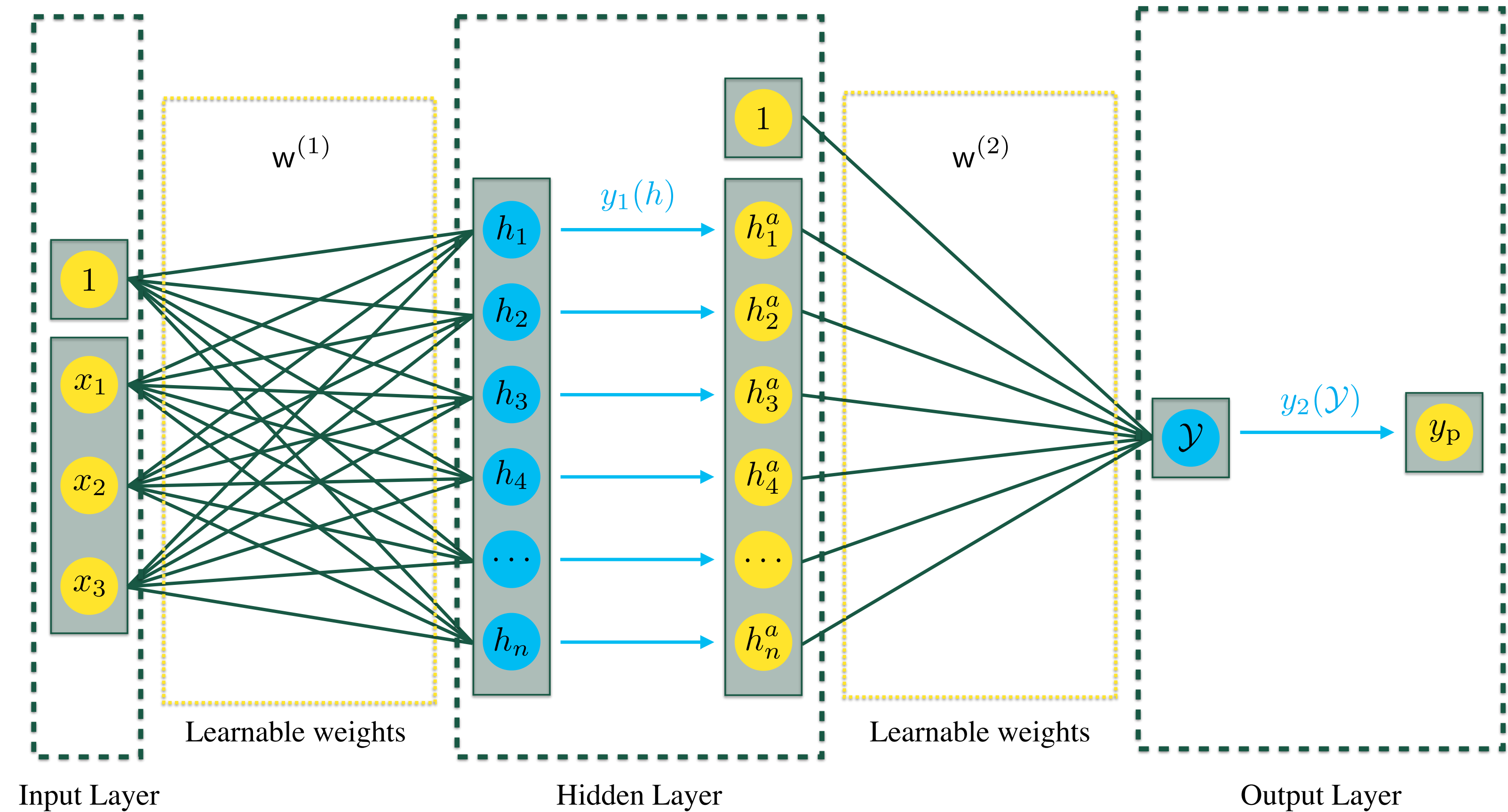
Opening the box



1. Review supervised learning
 - a. Linear Regression
 - b. Logistic Regression
2. Motivate use of Neural Networks
3. Weak supervision
(1706.09451)
4. What is the machine learning?
(1709.10106)

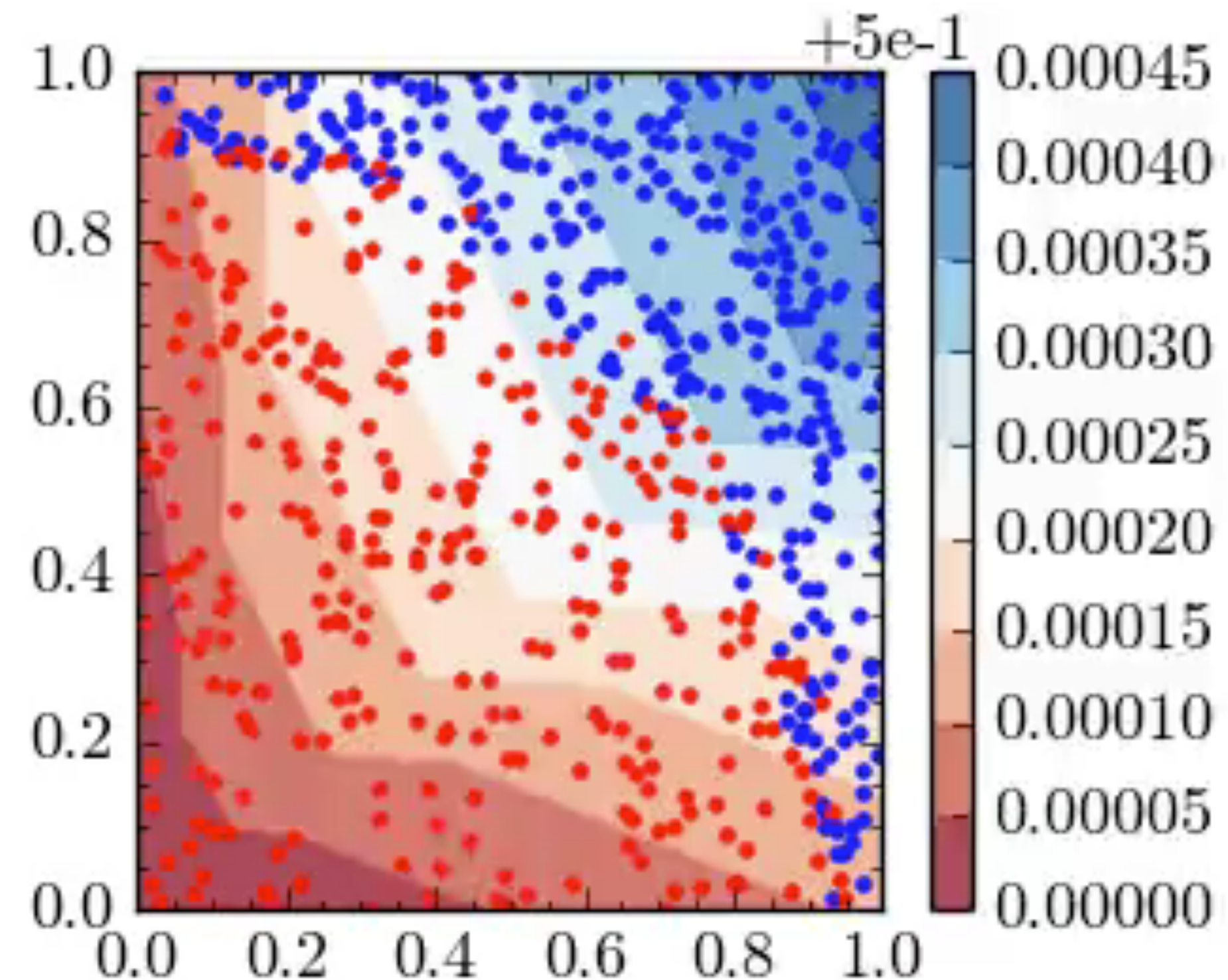
Neural Networks

- Can be used in classifications and numerical predictions
- Don't add more inputs, let machine find own shape
- Ability to learn 'any' function
- More nodes/hidden layers allows for more complex features



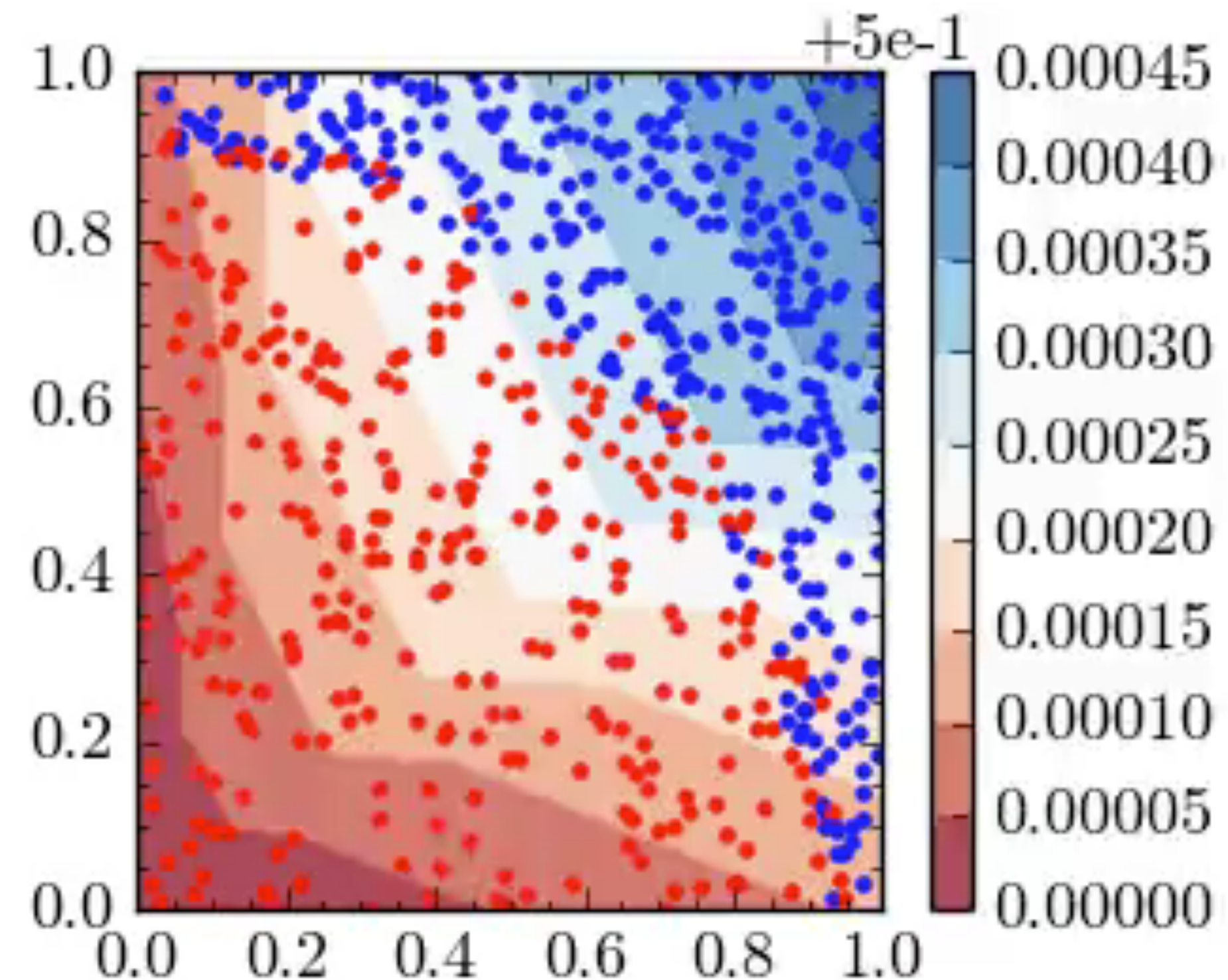
Neural Networks

- Can be used in classifications and numerical predictions
- Don't add more inputs, let machine find own shape
- Ability to learn 'any' function
- More nodes/hidden layers allows for more complex features

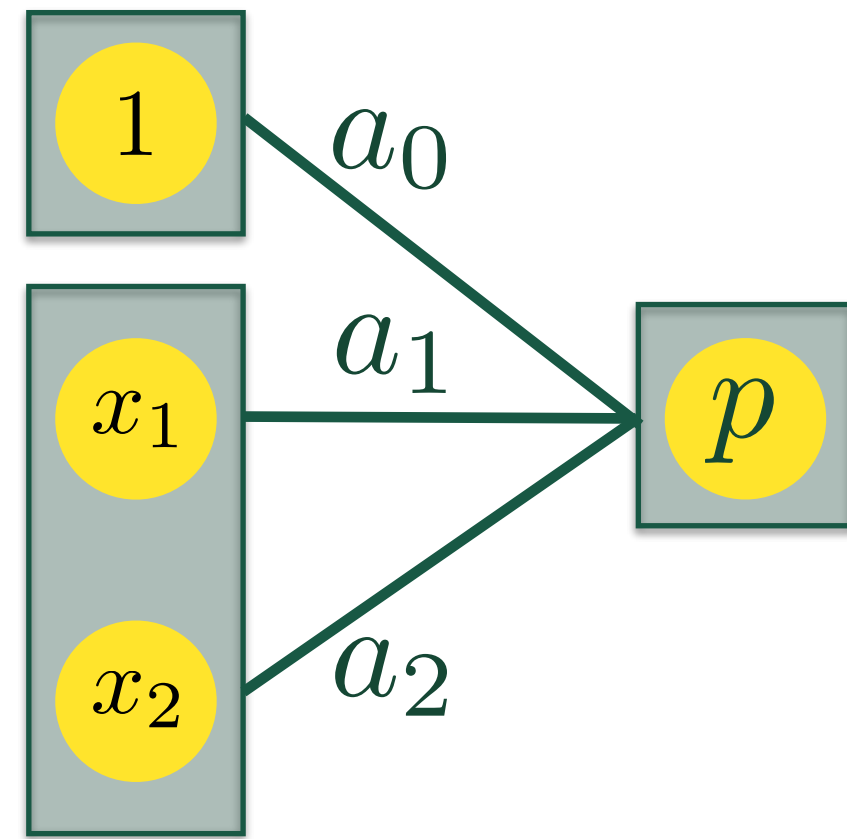


Neural Networks

- Can be used in classifications and numerical predictions
- Don't add more inputs, let machine find own shape
- Ability to learn 'any' function
- More nodes/hidden layers allows for more complex features



Neural Networks



OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

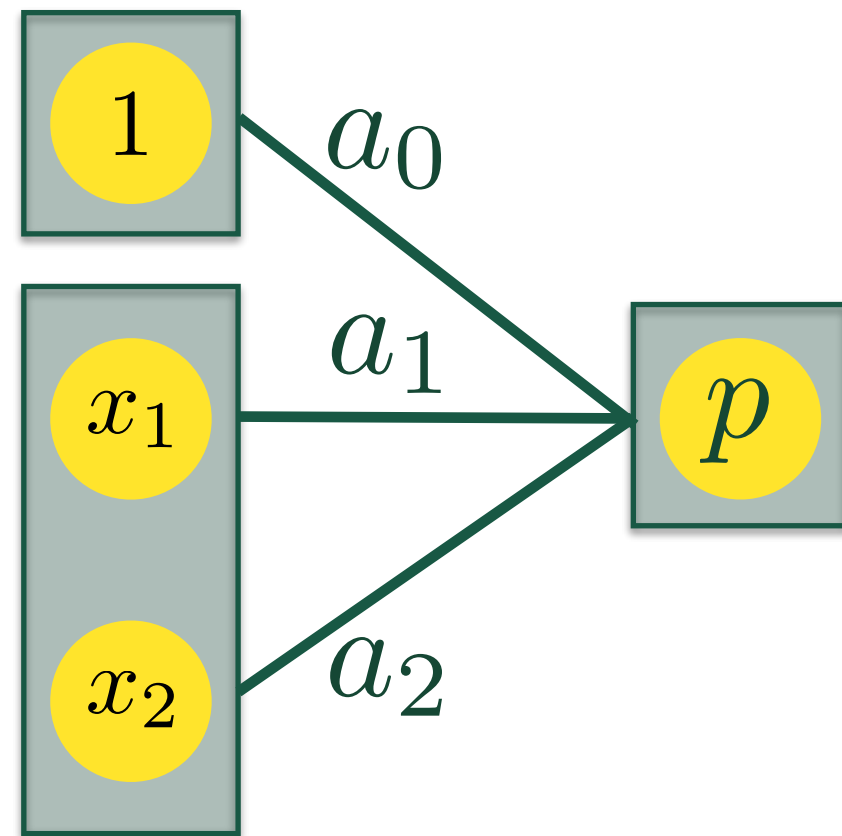
$$a_0 = -10, \quad a_1 = 15, \quad a_2 = 15$$

AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

$$a_0 = -20, \quad a_1 = 15, \quad a_2 = 15$$

Neural Networks



XOR		
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

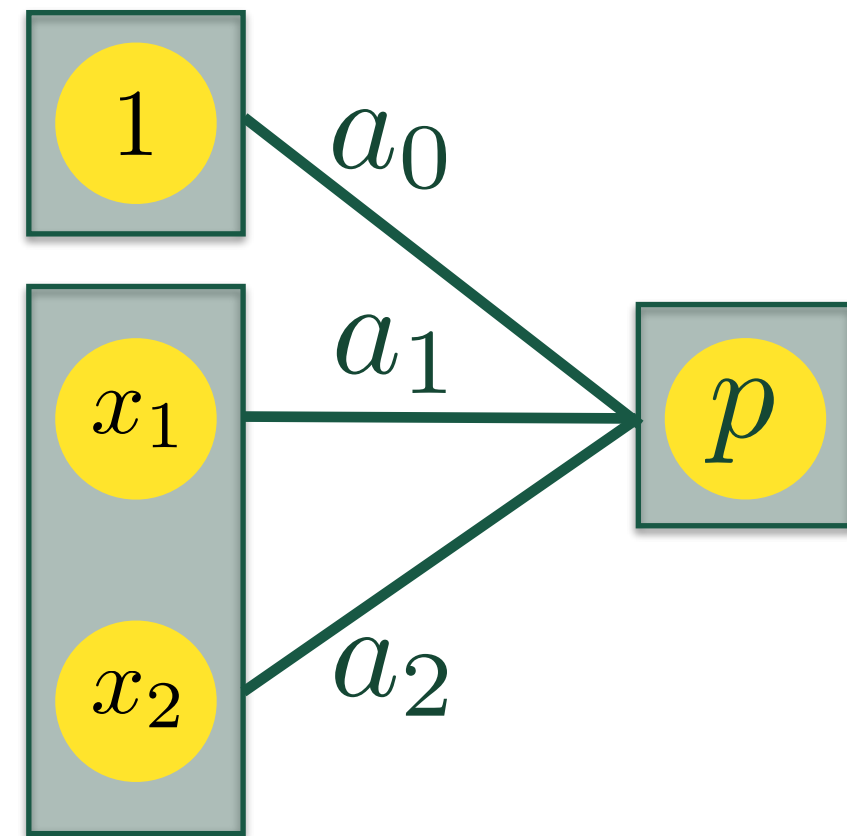
OR		
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

$a_0 = -10, a_1 = 15, a_2 = 15$

AND		
x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

$a_0 = -20, a_1 = 15, a_2 = 15$

Neural Networks



OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

$$a_0 = -10, \quad a_1 = 15, \quad a_2 = 15$$

AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

$$a_0 = -20, \quad a_1 = 15, \quad a_2 = 15$$

XOR

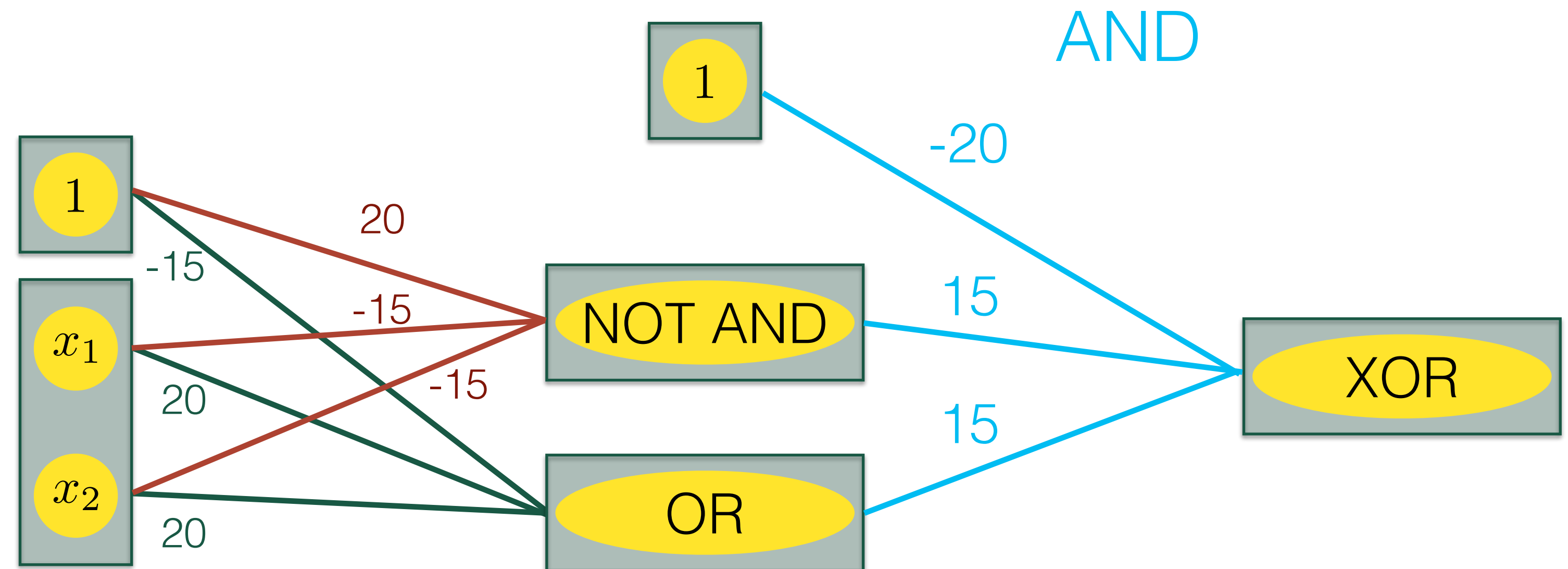
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

This system cannot produce XOR

(cannot make a two sided cut)

Neural Networks

XOR		
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0



Simple example showing that neural network can access 'high-level' functions

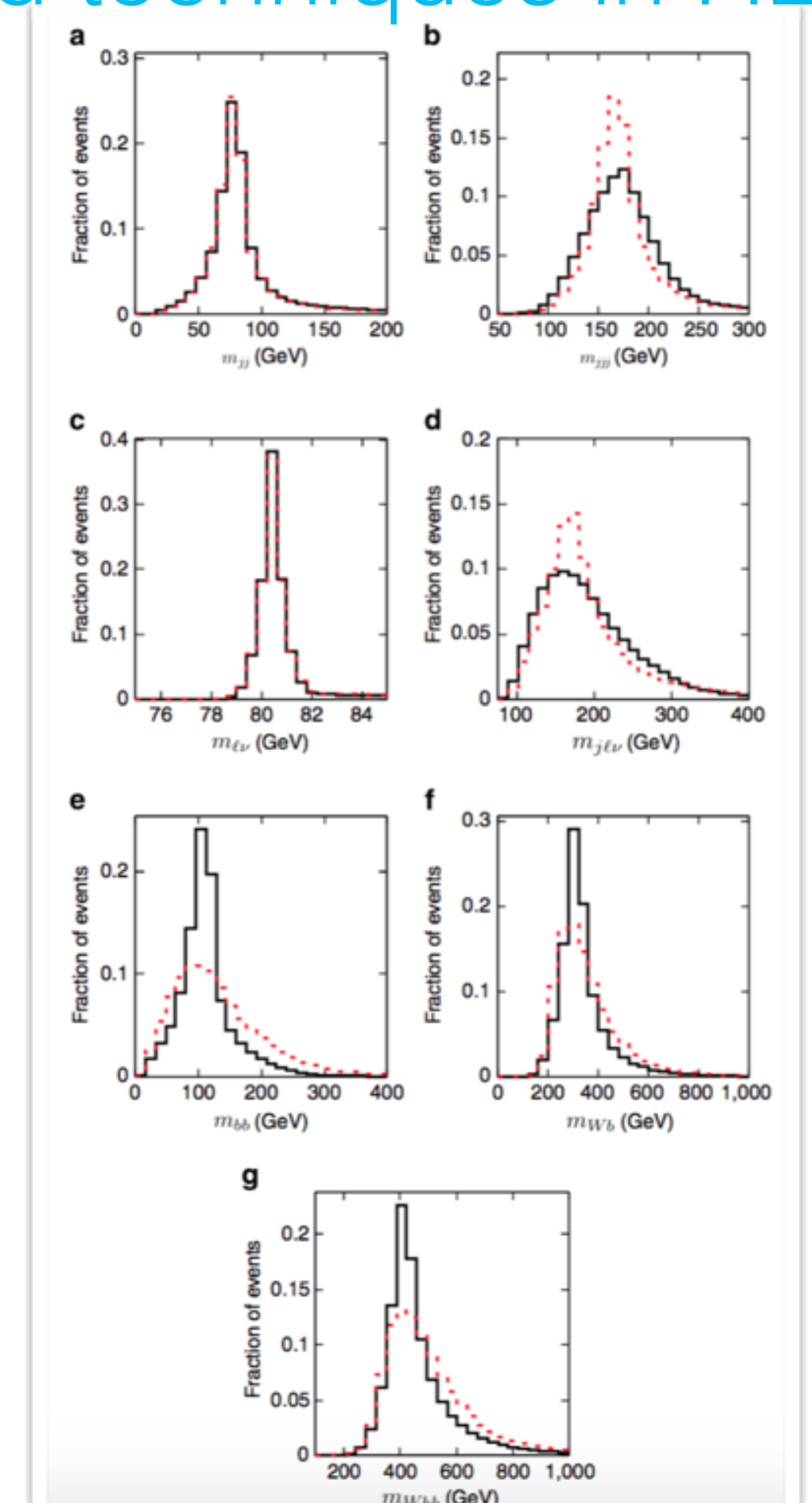
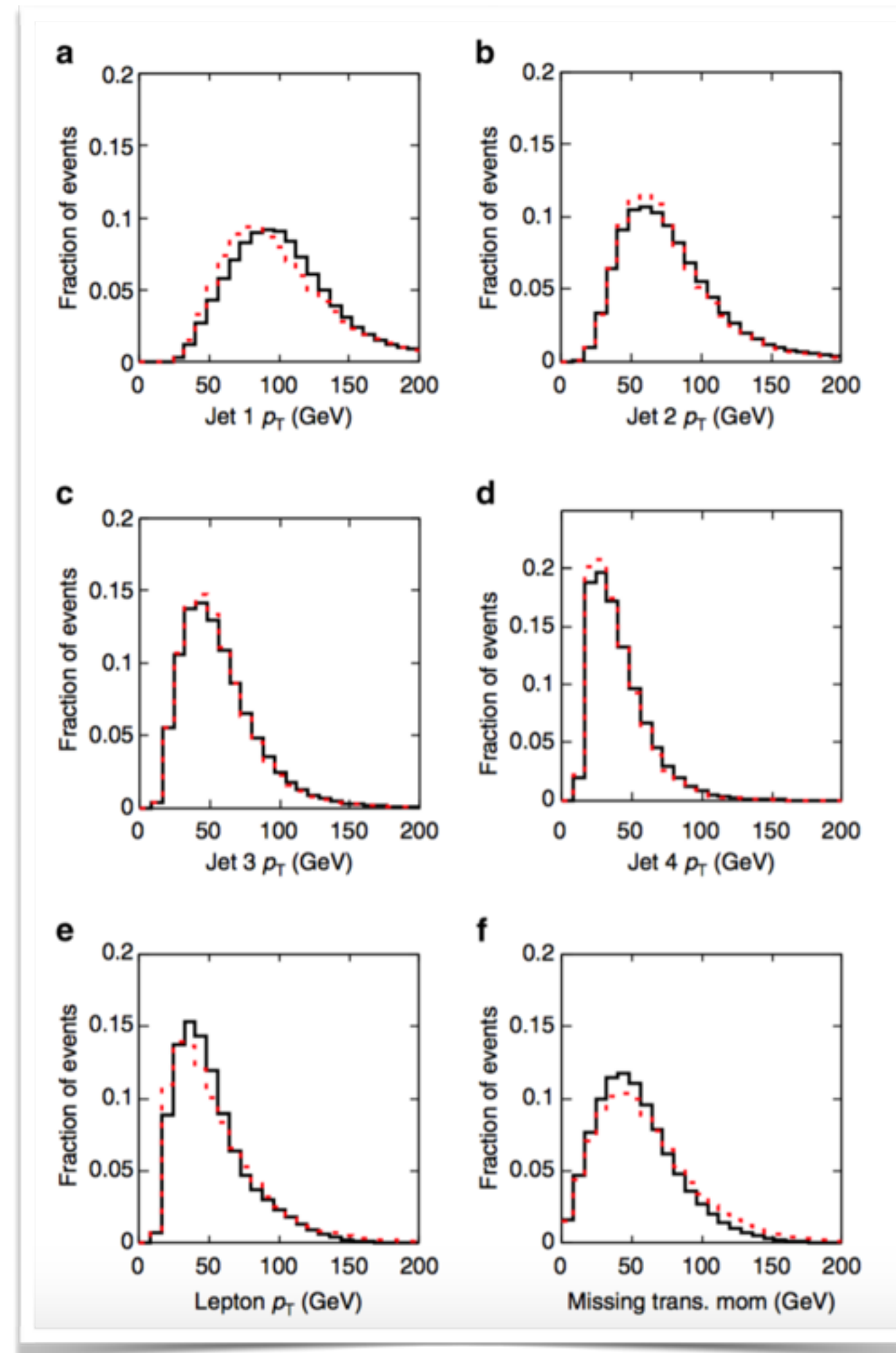
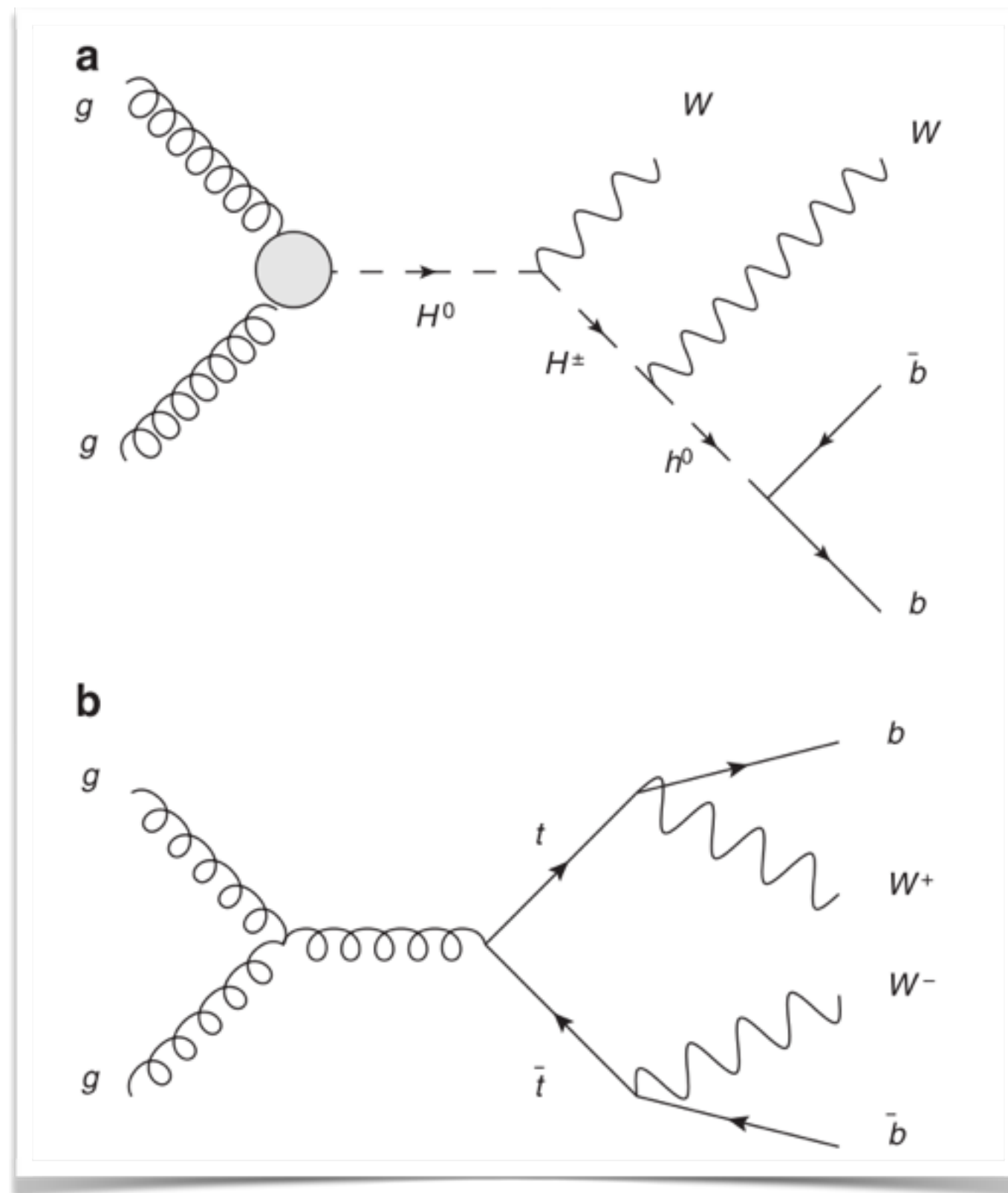
To learn weights, need LARGE training set and CPU time

Physics example next

Neural Networks (Deep Learning)

Baldi, Sadowski, and Whiteson [arXiv:1402.4735]

First paper to show deep learning outperforming standard techniques in HEP



Neural Networks (Deep Learning)

Baldi, Sadowski, and Whiteson [arXiv:1402.4735]

11 million training examples
5 layer deep network

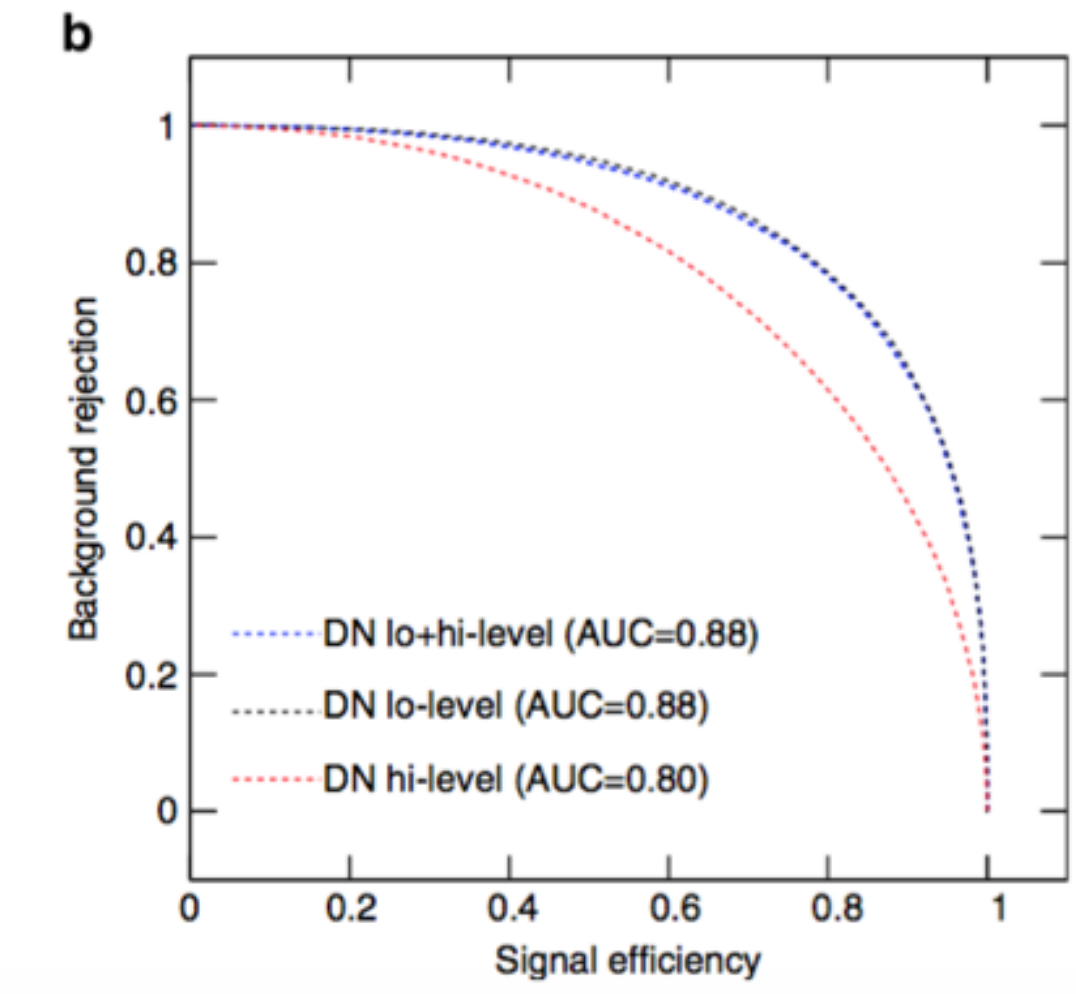
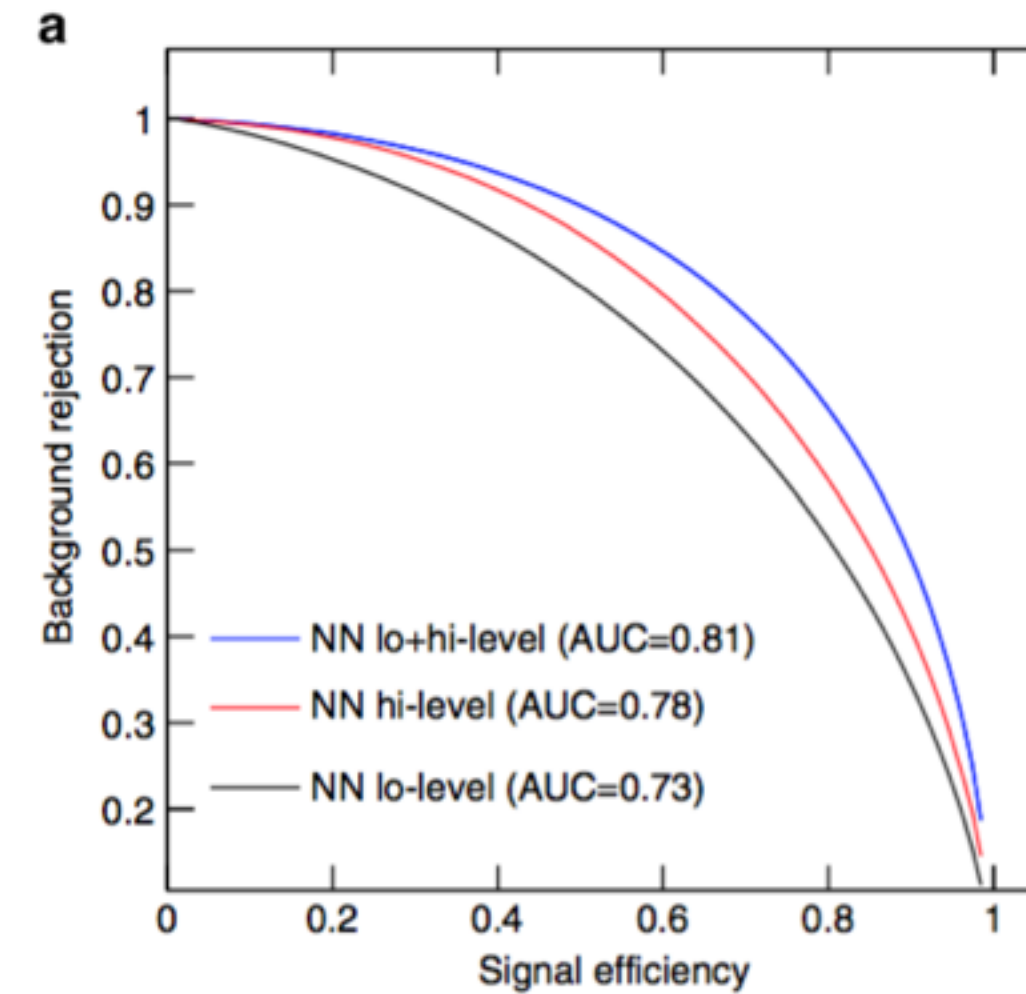


Table 1 | Performance for Higgs benchmark.

Technique	Low-level	High-level	Complete
<i>AUC</i>			
BDT	0.73 (0.01)	0.78 (0.01)	0.81 (0.01)
NN	0.733 (0.007)	0.777 (0.001)	0.816 (0.004)
DN	0.880 (0.001)	0.800 (<0.001)	0.885 (0.002)
<i>Discovery significance</i>			
NN	2.5 σ	3.1 σ	3.7 σ
DN	4.9 σ	3.6 σ	5.0 σ

Comparison of the performance of several learning techniques: boosted decision trees (BDT), shallow neural networks (NN), and deep neural networks (DN) for three sets of input features: low-level features, high-level features and the complete set of features. Each neural network was trained five times with different random initializations. The table displays the mean area under the curve (AUC) of the signal-rejection curve in Fig. 7, with s.d. in parentheses as well as the expected significance of a discovery (in units of Gaussian σ) for 100 signal events and $1,000 \pm 50$ background events.

Methods trained with only the high-level features, however, have a weaker performance than those trained with the full suite of features, which suggests that despite the insight represented by the high-level features, they do not capture all the information contained in the low-level features. The deep-learning techniques show nearly equivalent performance using the low-level features and the complete features, suggesting that they are automatically discovering the insight contained in the high-level features.

Neural Networks (Deep Learning)

Baldi, Sadowski, and Whiteson [arXiv:1402.4735]

11 million training examples
5 layer deep network

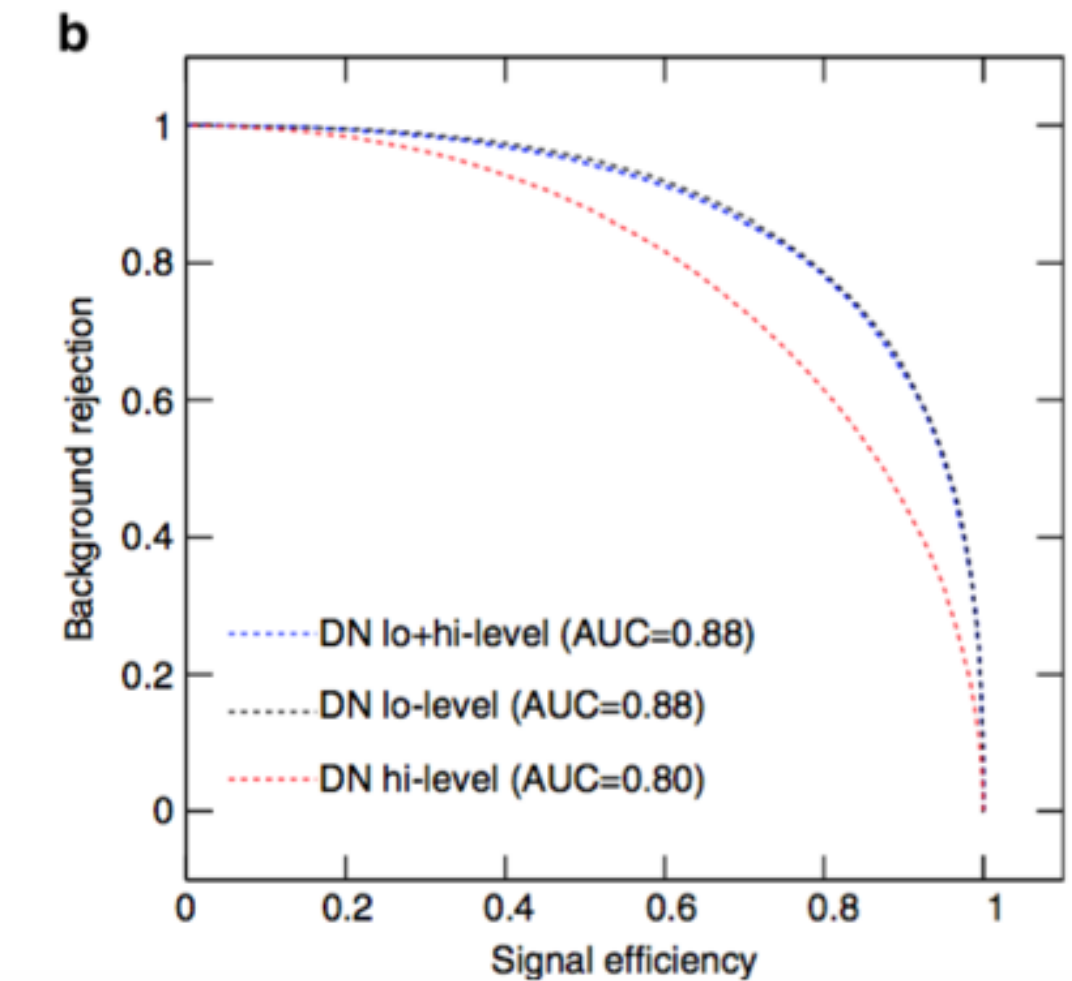
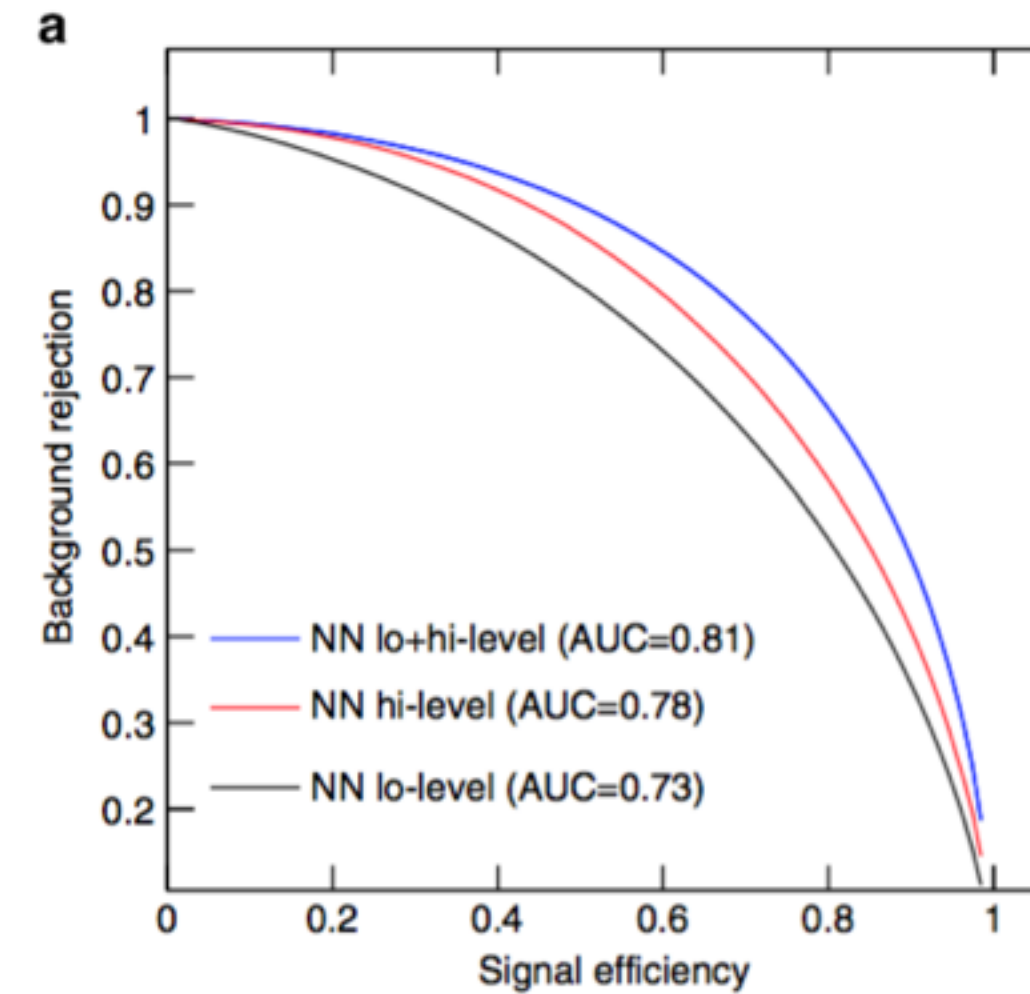
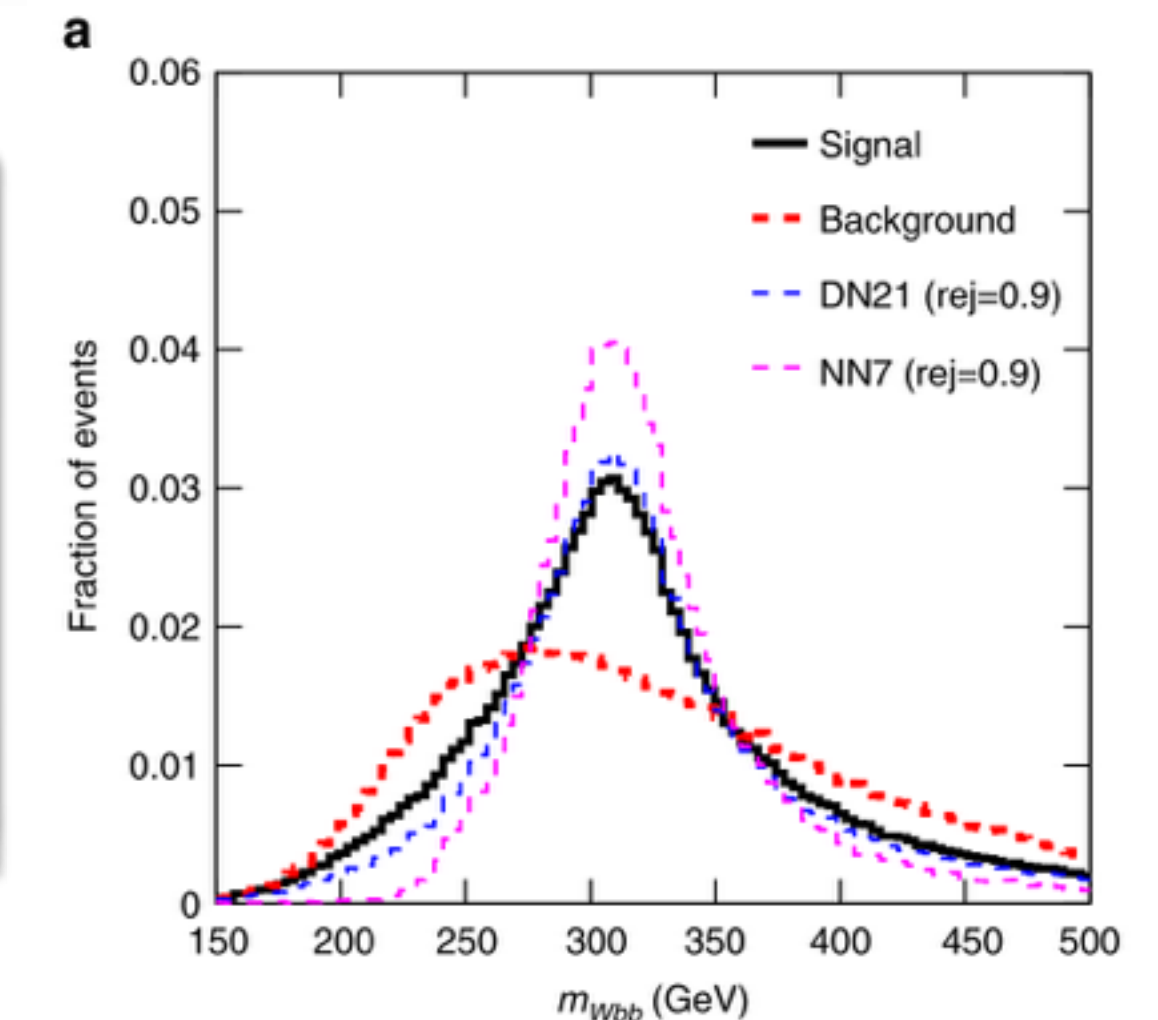


Table 1 | Performance for Higgs benchmark.

Technique	Low-level	High-level	Complete
<i>AUC</i>			
BDT	0.73 (0.01)	0.78 (0.01)	0.81 (0.01)
NN	0.733 (0.007)	0.777 (0.001)	0.816 (0.004)
DN	0.880 (0.001)	0.800 (<0.001)	0.885 (0.002)
<i>Discovery significance</i>			
NN	2.5 σ	3.1 σ	3.7 σ
DN	4.9 σ	3.6 σ	5.0 σ

Comparison of the performance of several learning techniques: boosted decision trees (BDT), shallow neural networks (NN), and deep neural networks (DN) for three sets of input features: low-level features, high-level features and the complete set of features. Each neural network was trained five times with different random initializations. The table displays the mean area under the curve (AUC) of the signal-rejection curve in Fig. 7, with s.d. in parentheses as well as the expected significance of a discovery (in units of Gaussian σ) for 100 signal events and $1,000 \pm 50$ background events.

Methods trained with only the high-level features, however, have a weaker performance than those trained with the full suite of features, which suggests that despite the insight represented by the high-level features, they do not capture all the information contained in the low-level features. The deep-learning techniques show nearly equivalent performance using the low-level features and the complete features, suggesting that they are automatically discovering the insight contained in the high-level features.



Opening the box



1. Review supervised learning
 - a. Linear Regression
 - b. Logistic Regression
2. Motivate use of Neural Networks
3. Weak supervision
(1706.09451)
4. What is the machine learning?
(1709.10106)

Weak Supervision

Started based on Learning with Label Proportions (LLP) from CS literature

Weak Supervision

Started based on Learning with Label Proportions (LLP) from CS literature

Dery, Nachman, Rubbo, and Schwarzman, “Weakly Supervised Classification in High Energy Physics,” [arXiv:1702.00414]

Don't need to know if individual event is signal or background. Need multiple sets of data where we know how the fraction of signal and background in each. Can train straight on LHC data instead of MC.

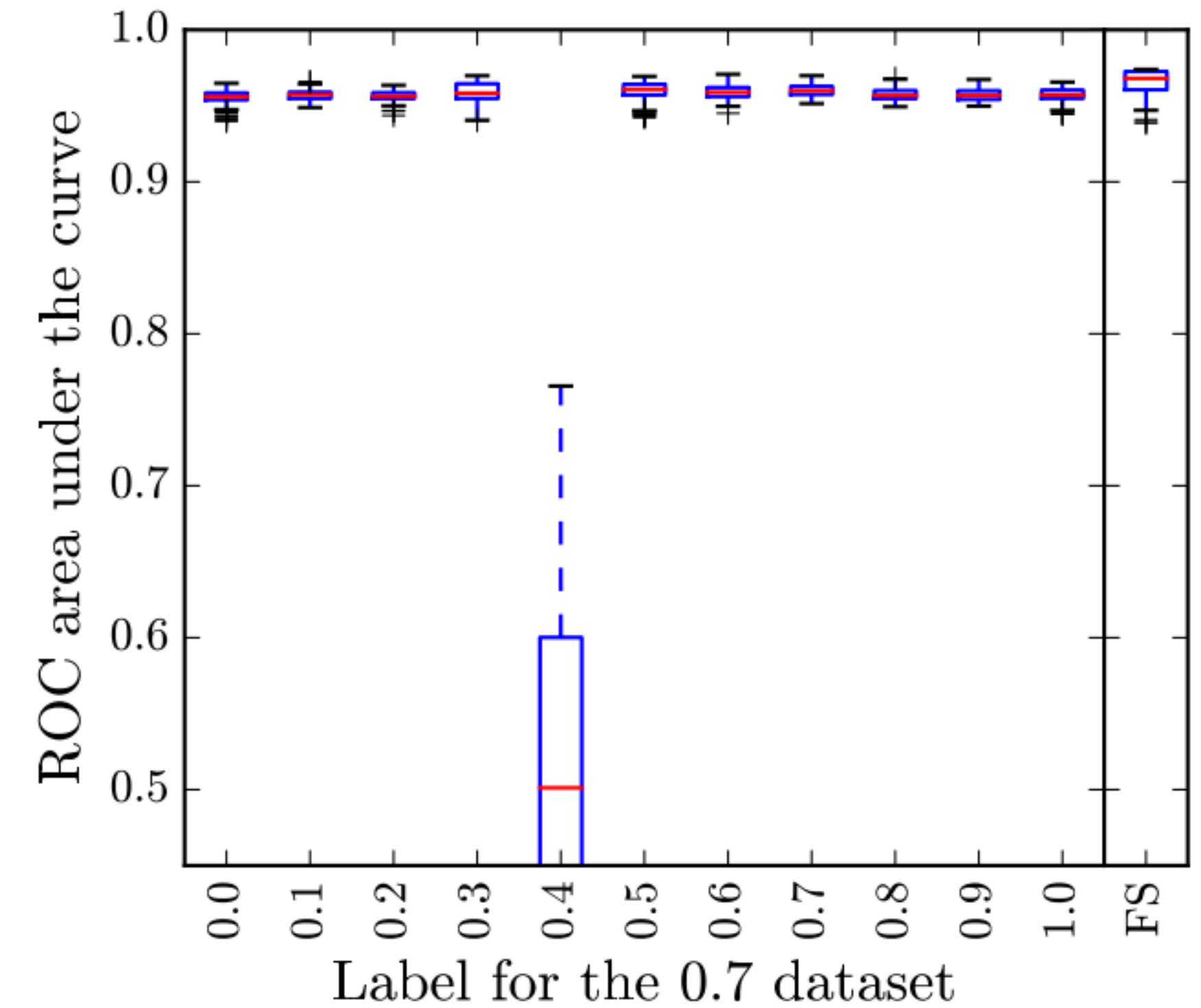
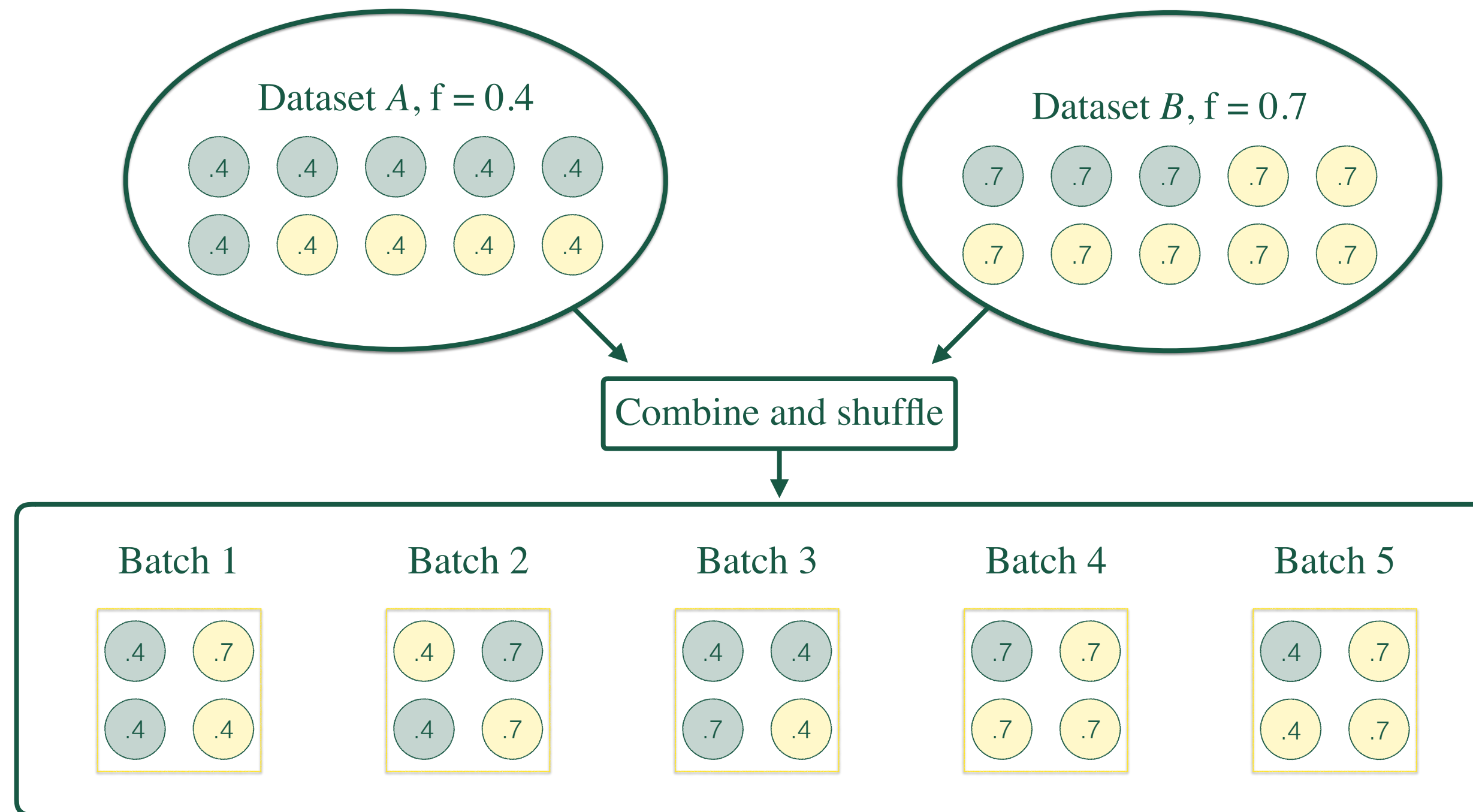
Cohen, Freytsis, and BO, “(Machine) Learning to Do More with Less”, [arXiv:1706.09451]

Studied how the performance changes if there are errors in the fractions. The process is found to be robust against various types of systematic errors.

Metodiev, Nachman, and Thaler, “Classification without labels: Learning from mixed samples in high energy physics,” [arXiv:1708.02949]

Found that the label can actually be arbitrary. Neither individual label nor class portion is actually necessary.

Weak Supervision



This method allows for training on data instead of MC. This alleviates worry of mismodeling of the detector, QCD, etc. Still much to do to understand other systematics and how to realistically implement at colliders.

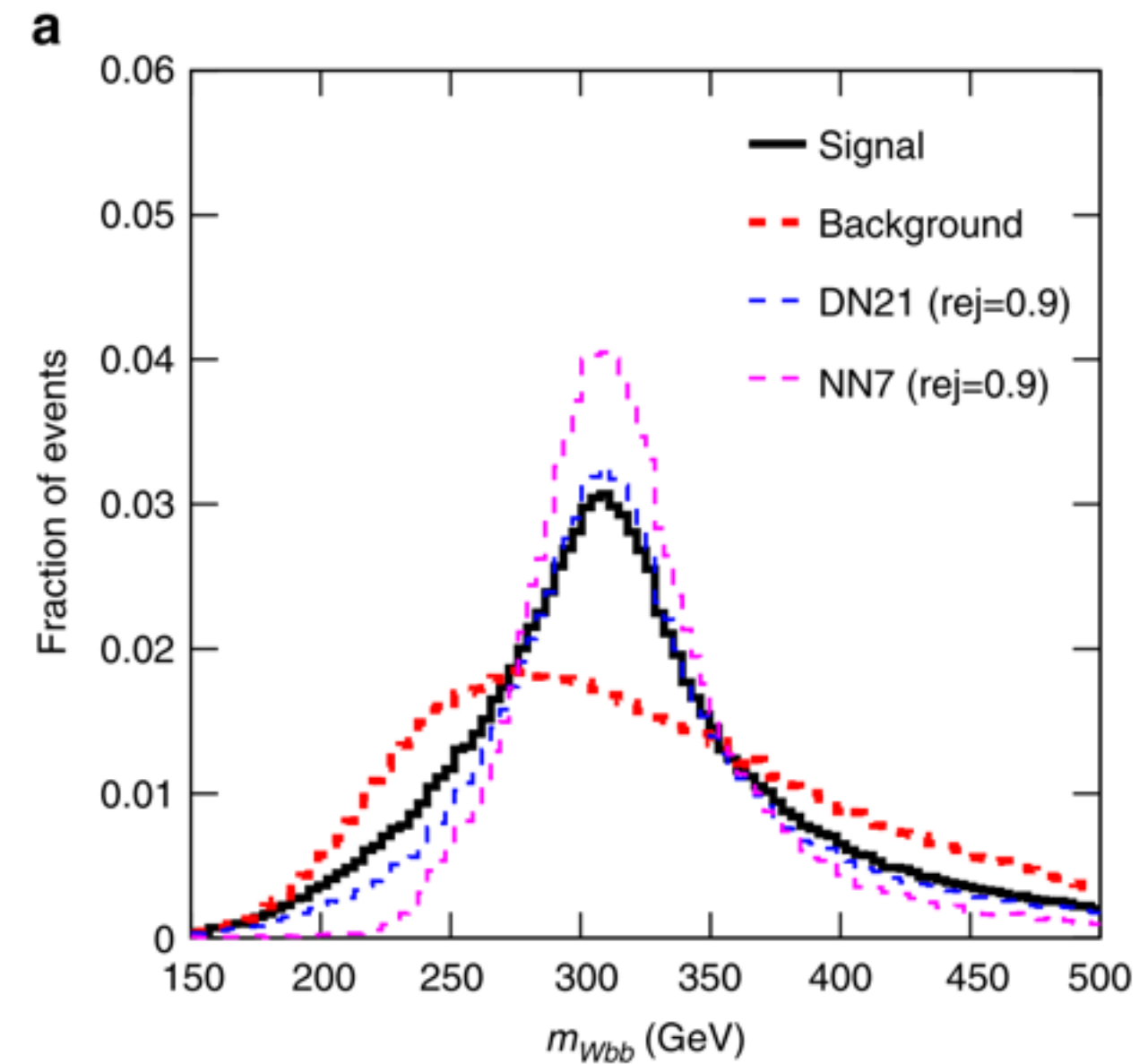
Opening the box



1. Review supervised learning
 - a. Linear Regression
 - b. Logistic Regression
2. Motivate use of Neural Networks
3. Weak supervision
(1706.09451)
4. What is the machine learning?
(1709.10106)

What is the machine learning?

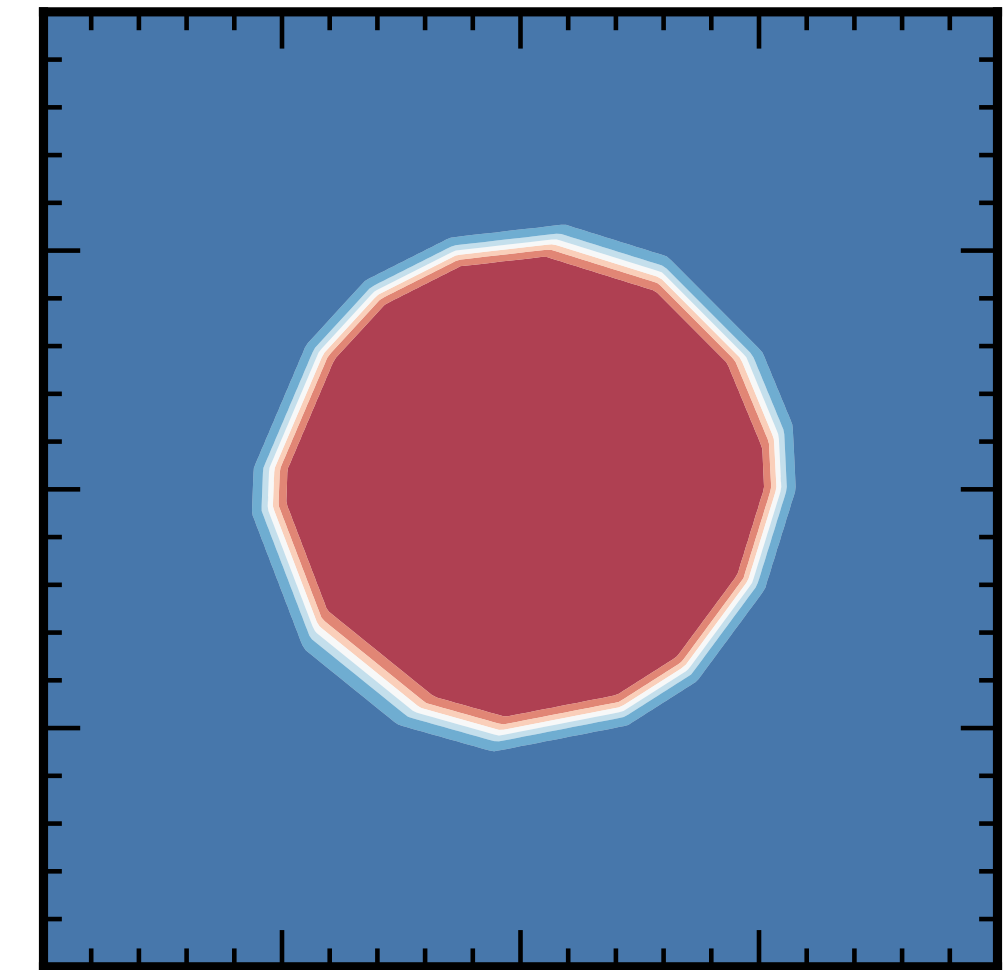
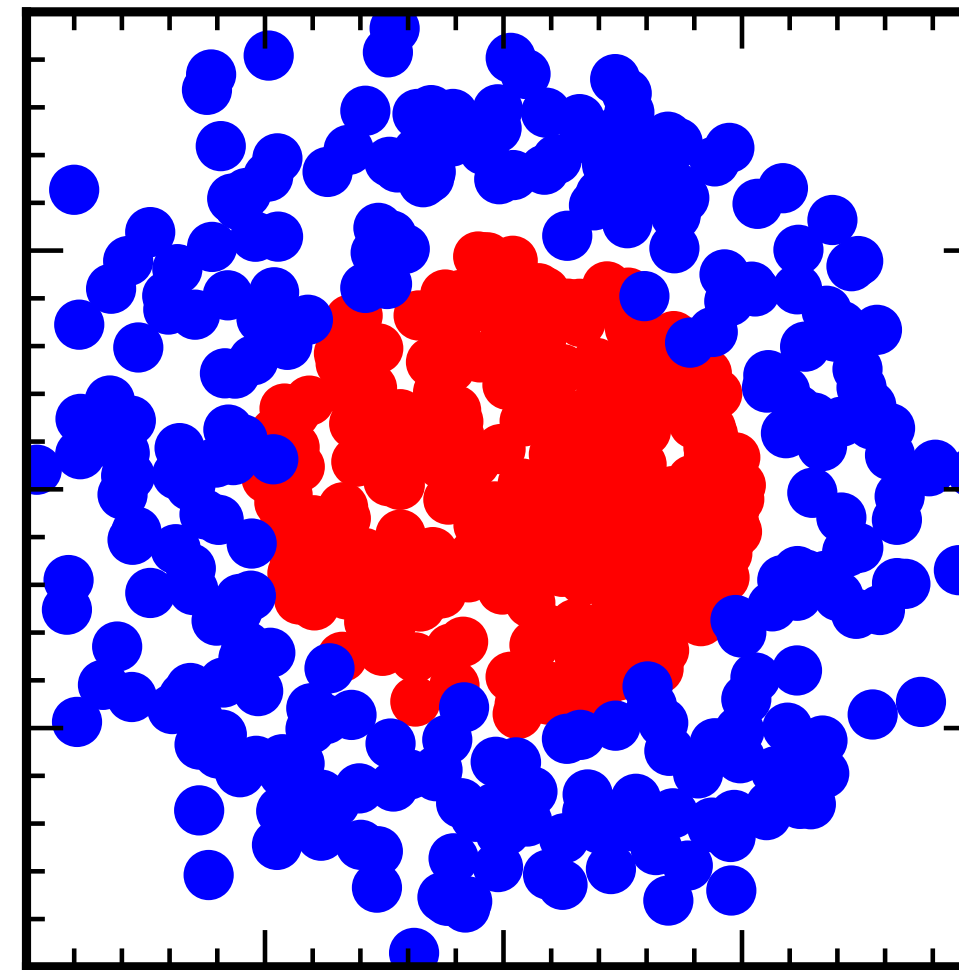
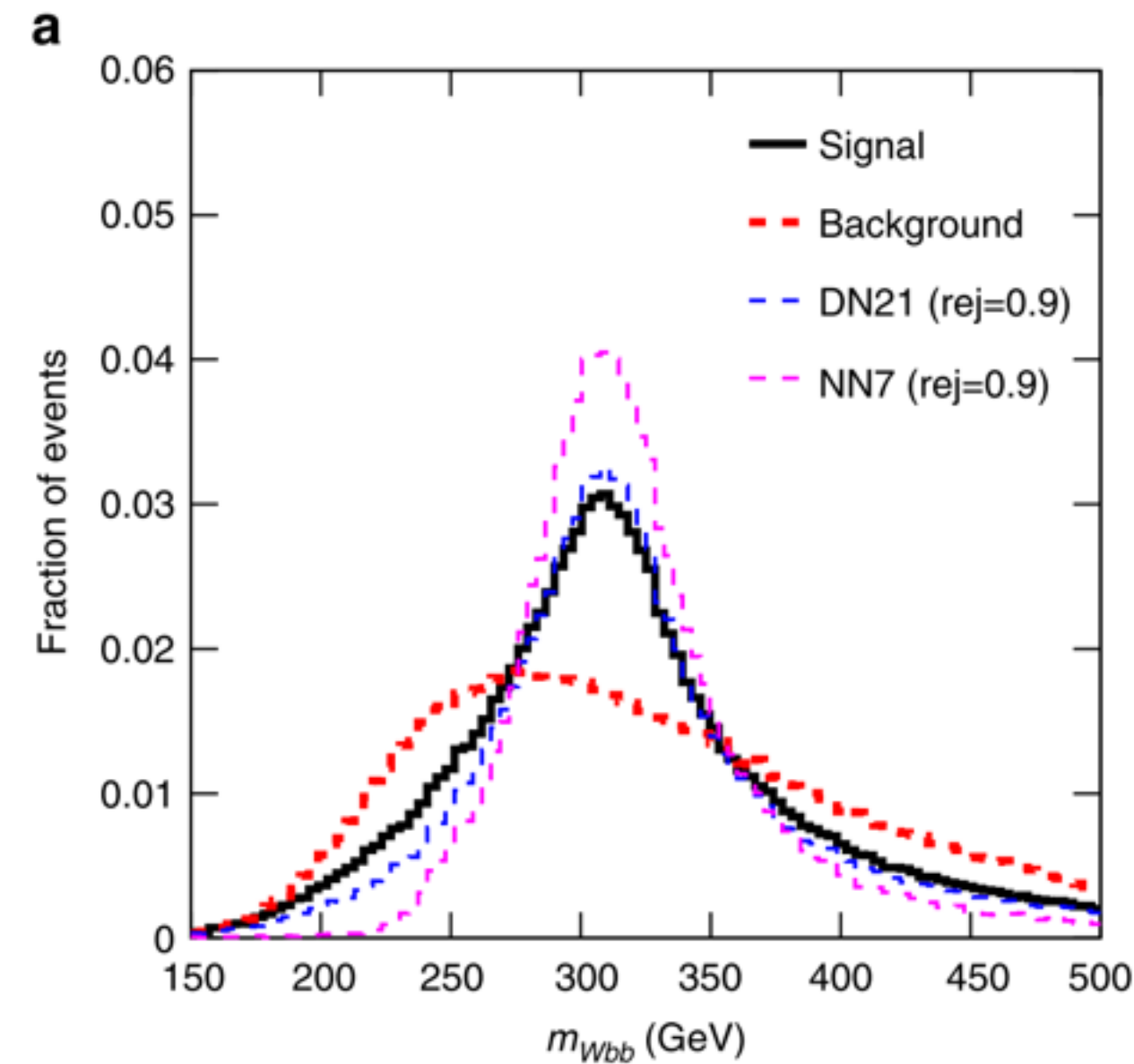
Did the machine learn the invariant mass of the wbb system?



Chang, Cohen, and BO [arXiv:1709.10106]

What is the machine learning?

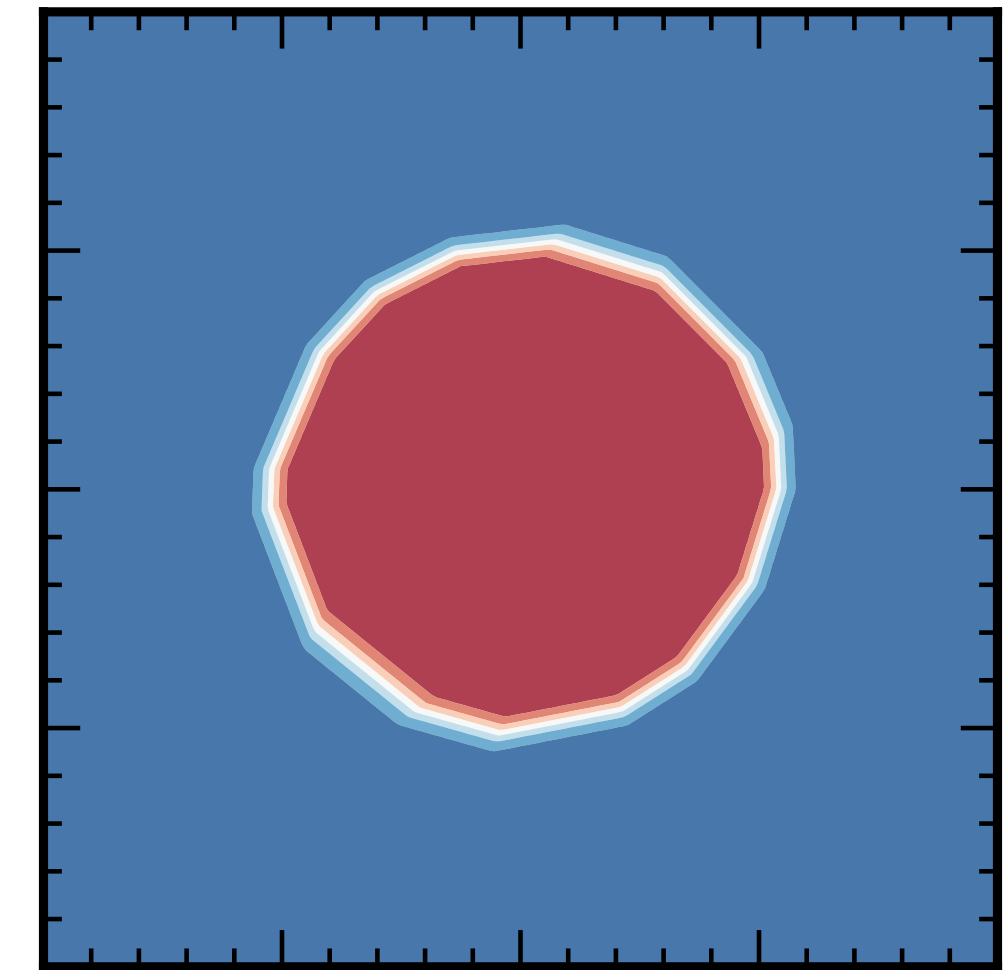
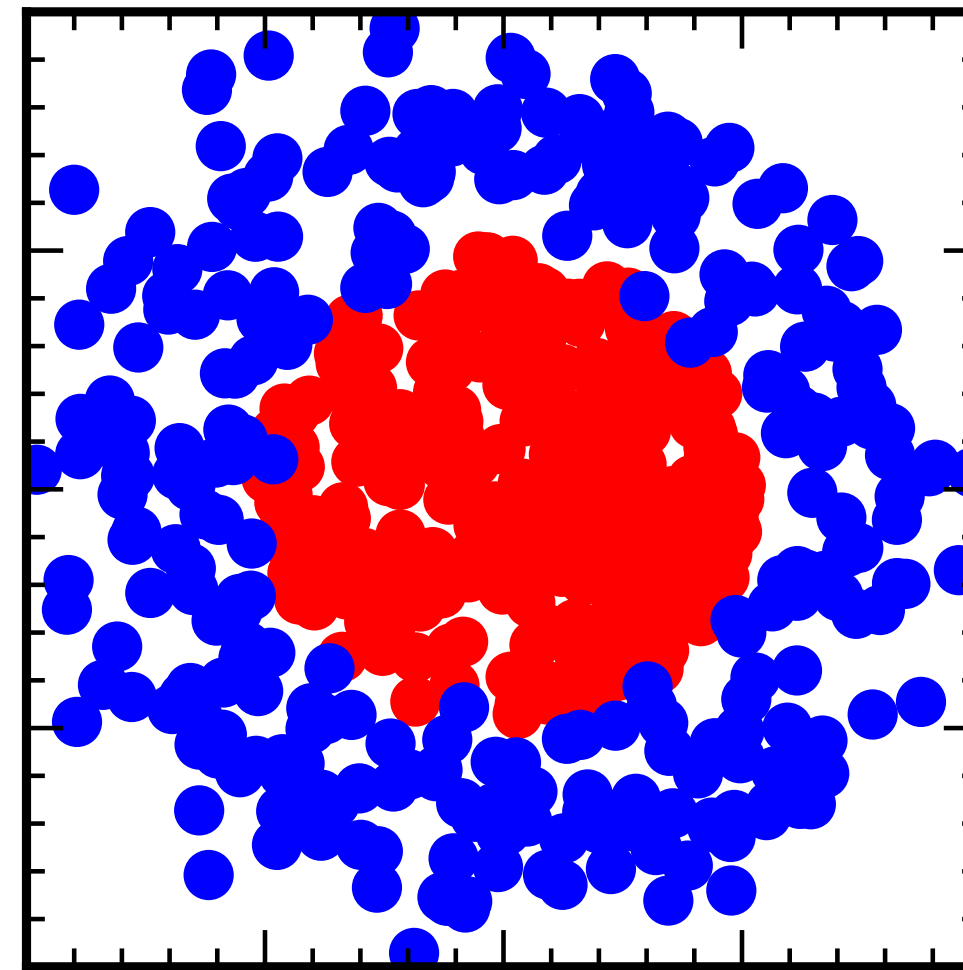
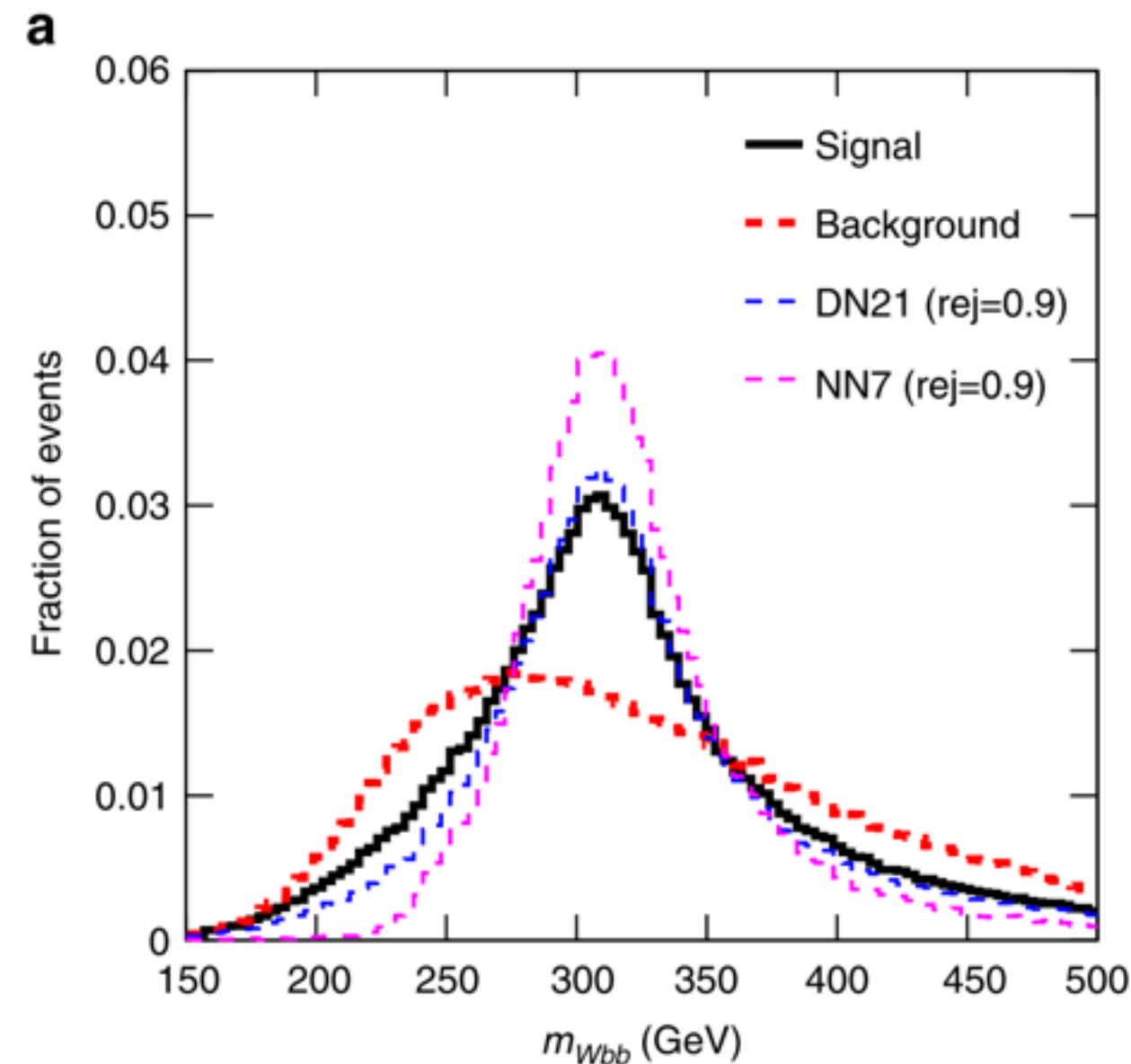
Did the machine learn the invariant mass of the wbb system?



Chang, Cohen, and BO [arXiv:1709.10106]

What is the machine learning?

Did the machine learn the invariant mass of the wbb system?

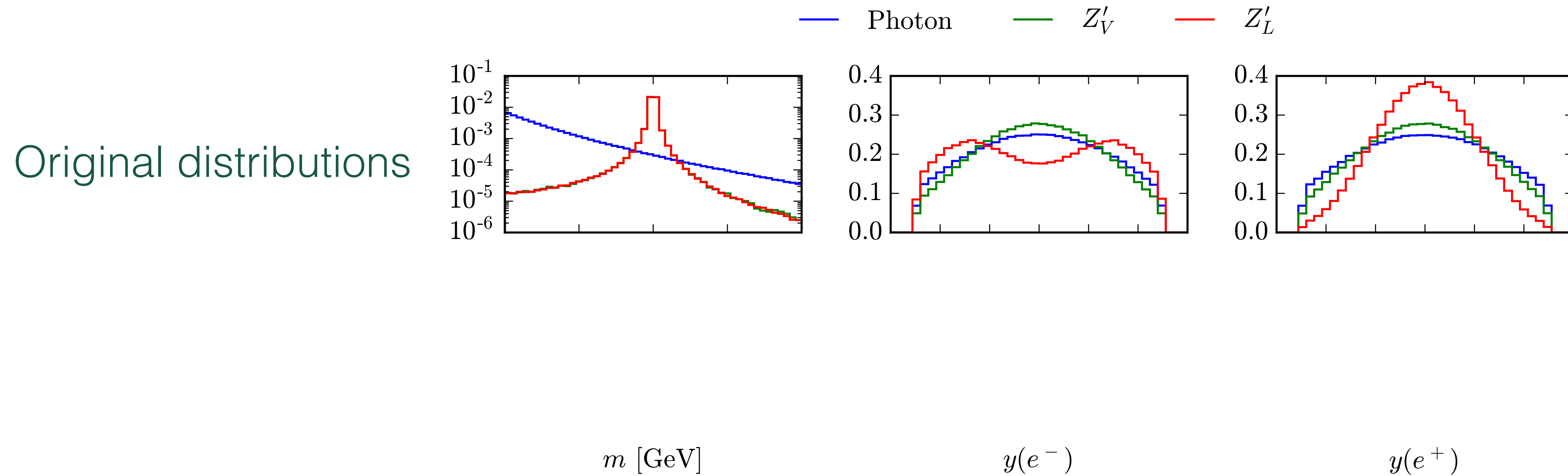


It has learned generically where events are in “any” parameter space. Wrong question to ask.

Chang, Cohen, and BO [arXiv:1709.10106]

What is the machine learning?

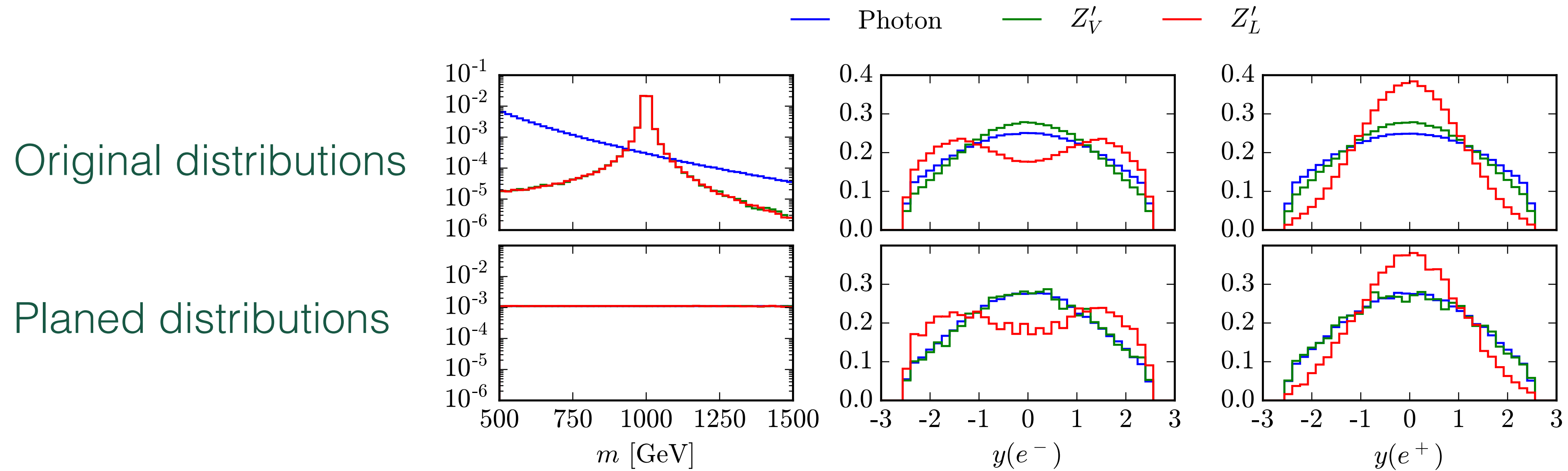
How much information is there to learn in a given distribution?



(E, \vec{p})	m	PLANED	LINEAR AUC	DEEP AUC
✓	✗	✗	0.746221(01)	0.988510(98)
✓	✓	✗	0.938967(01)	0.989007(03)
✓	✗	m	0.50550(29)	0.4942(48)

What is the machine learning?

How much information is there to learn in a given distribution?



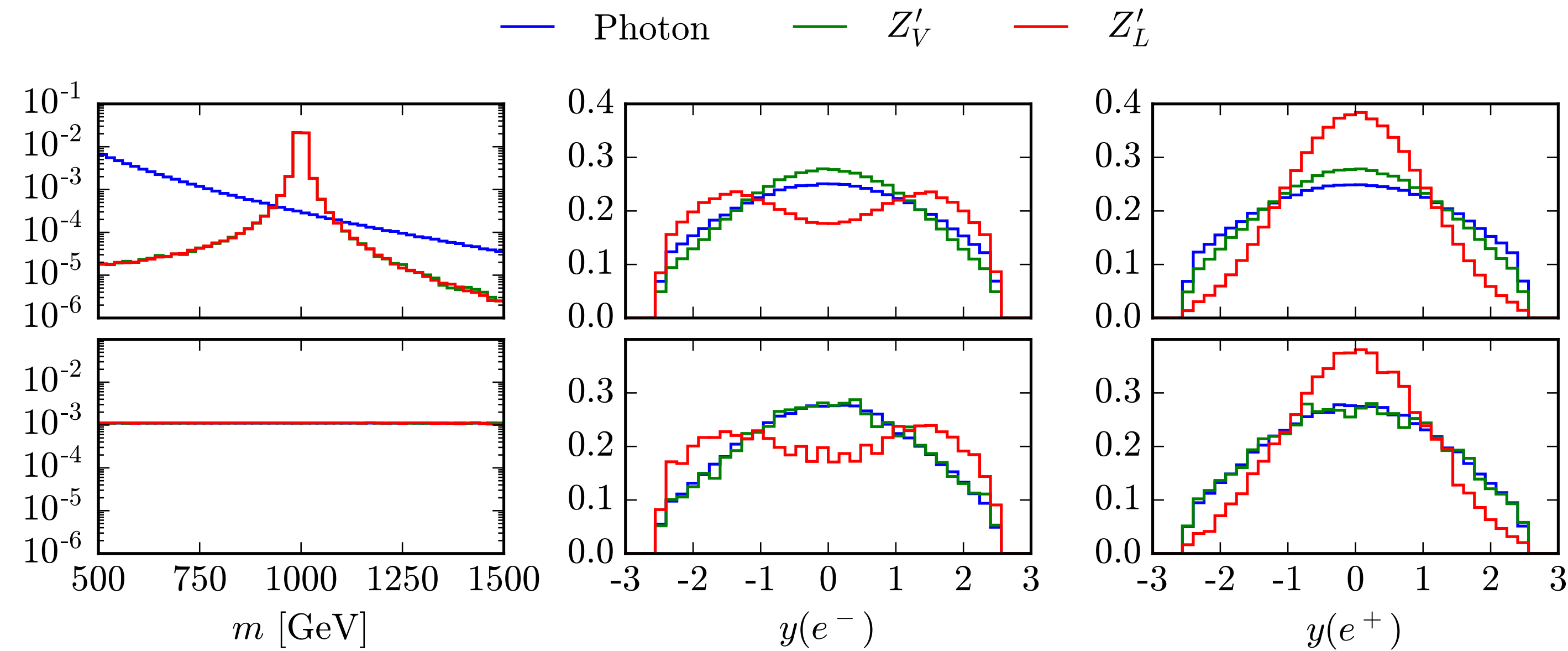
(E, \vec{p})	m	PLANED	LINEAR AUC	DEEP AUC
✓	✗	✗	0.746221(01)	0.988510(98)
✓	✓	✗	0.938967(01)	0.989007(03)
✓	✗	m	0.50550(29)	0.4942(48)

What is the machine learning?

How much information is there to learn in a given distribution?

Original distributions

Planned distributions

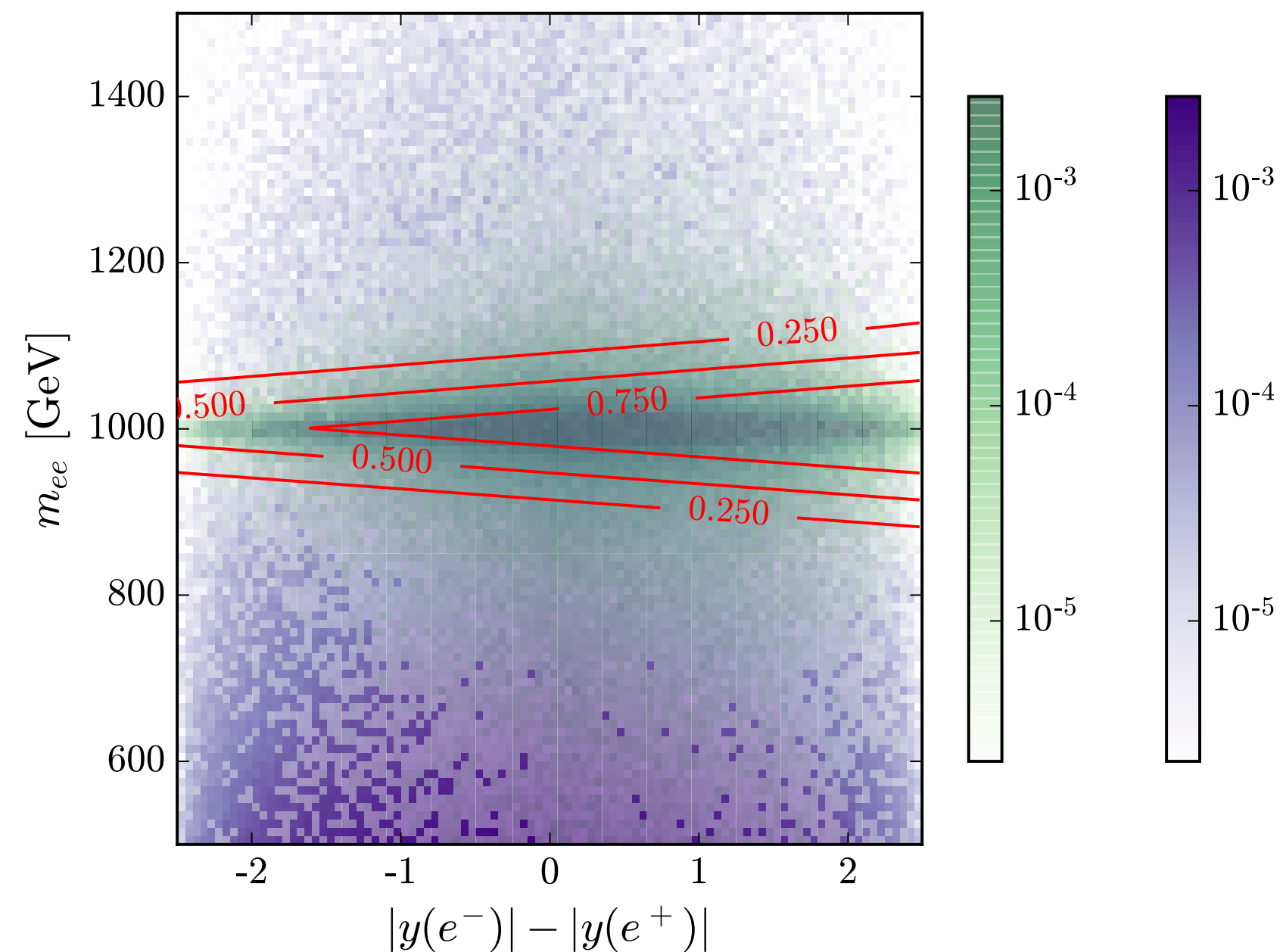
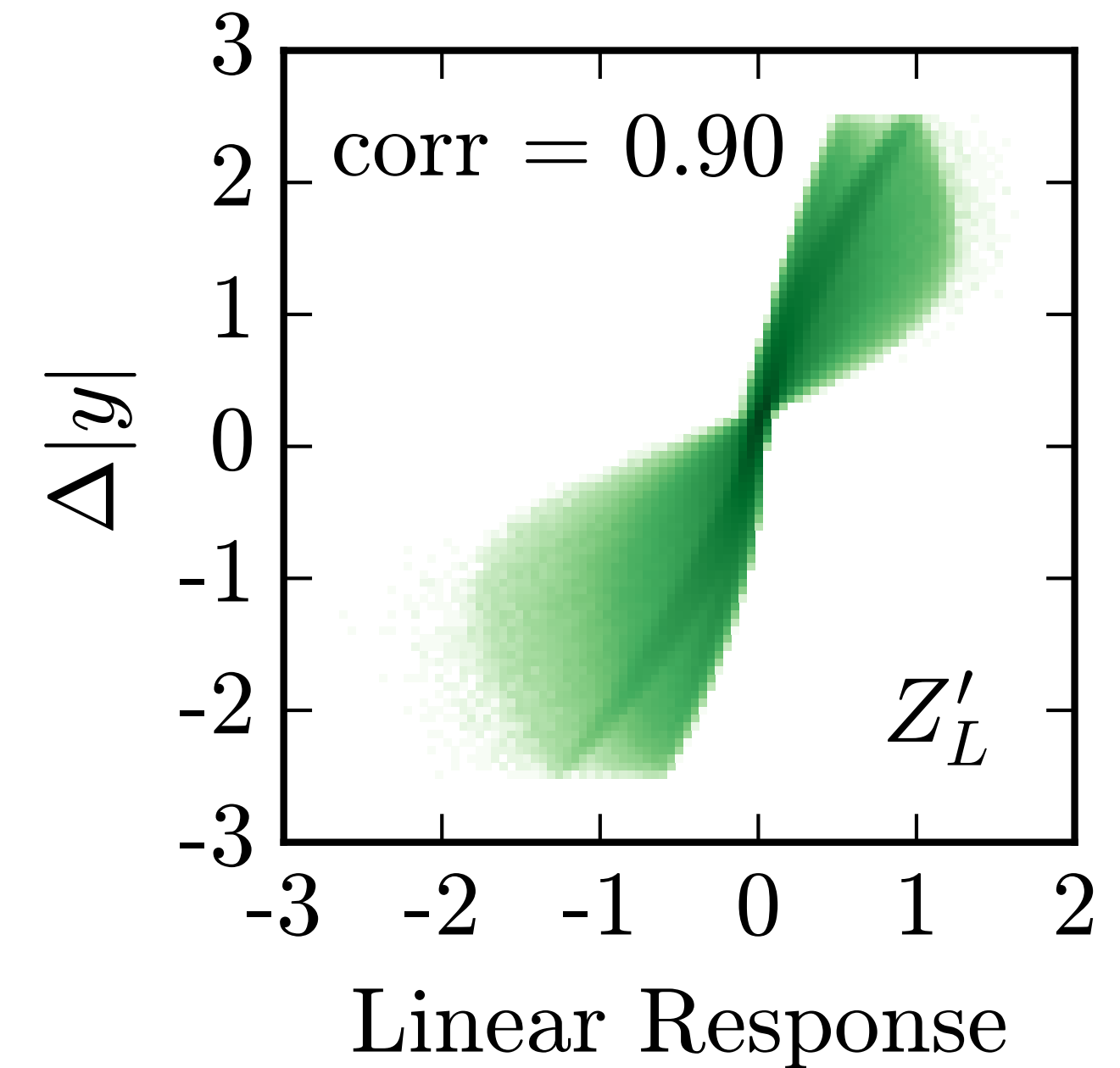


(E, \vec{p})	m	PLANED	LINEAR AUC	DEEP AUC
✓	✗	✗	0.746221(01)	0.988510(98)
✓	✓	✗	0.938967(01)	0.989007(03)
✓	✗	m	0.50550(29)	0.4942(48)

(E, \vec{p})	m	PLANED	LINEAR AUC	DEEP AUC
✓	✗	✗	0.763280(05)	0.989353(59)
✓	✓	✗	0.942004(02)	0.989826(10)
✓	✗	m	0.626648(28)	0.6258(24)
✓	✗	$(m, \Delta y)$	0.52421(15)	0.5320(25)

What is the machine learning?

(E, \vec{p})	m	PLANED	LINEAR AUC	DEEP AUC
✓	✗	✗	0.763280(05)	0.989353(59)
✓	✓	✗	0.942004(02)	0.989826(10)
✓	✗	m	0.626648(28)	0.6258(24)
✓	✗	$(m, \Delta y)$	0.52421(15)	0.5320(25)



Using only mass: AUC = 0.939, ACC = 0.937
 Using both: AUC = 0.989, ACC = 0.958

Conclusion

- Machine learning is essentially a large scale minimization problem
- Deep learning can produce “any” function with large enough sample and training time
- Novel ideas on how to use to mitigate systematics and mismodeling
- With physical insight, becomes a powerful tool for discovering what aspects of input distributions provide separating power