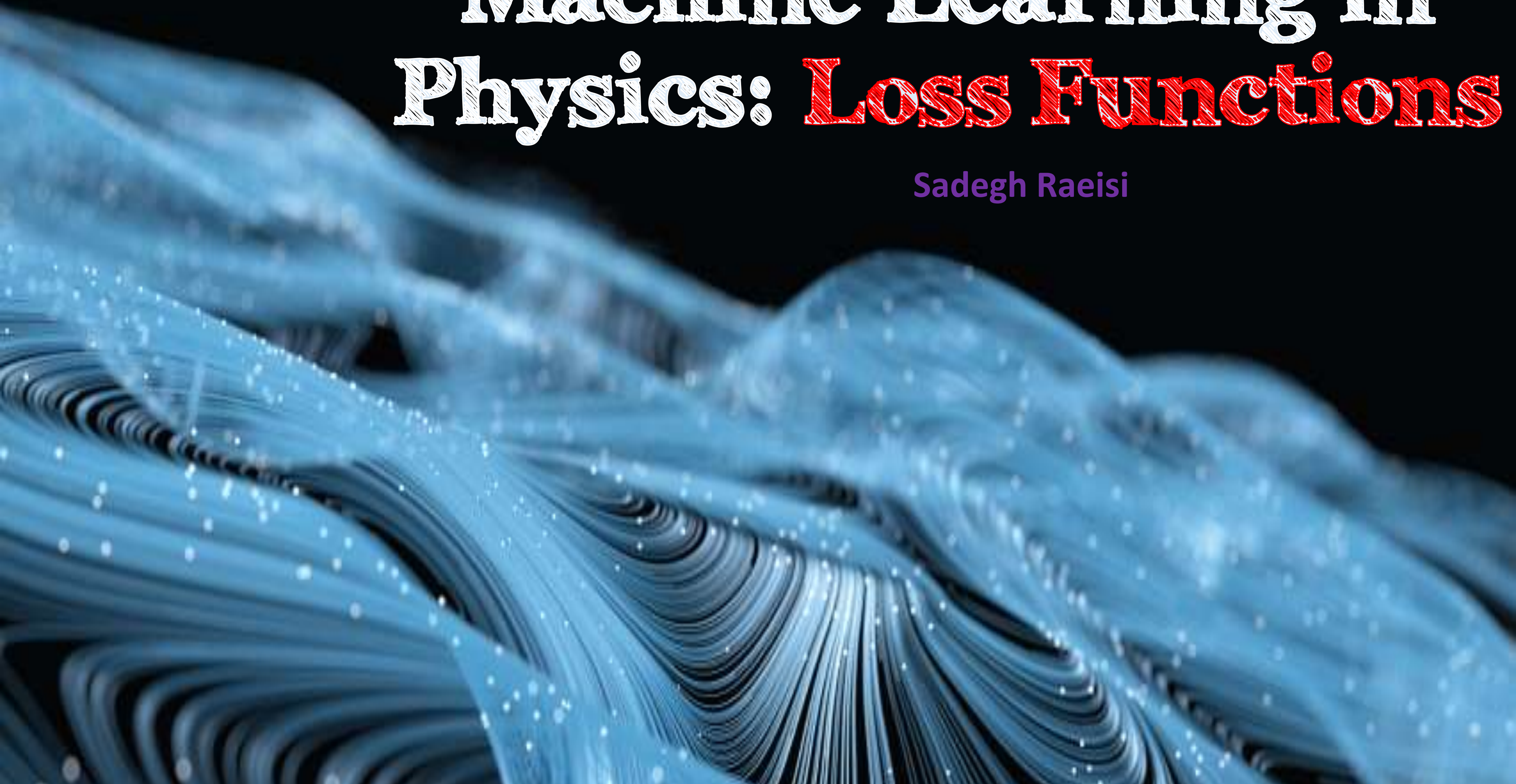
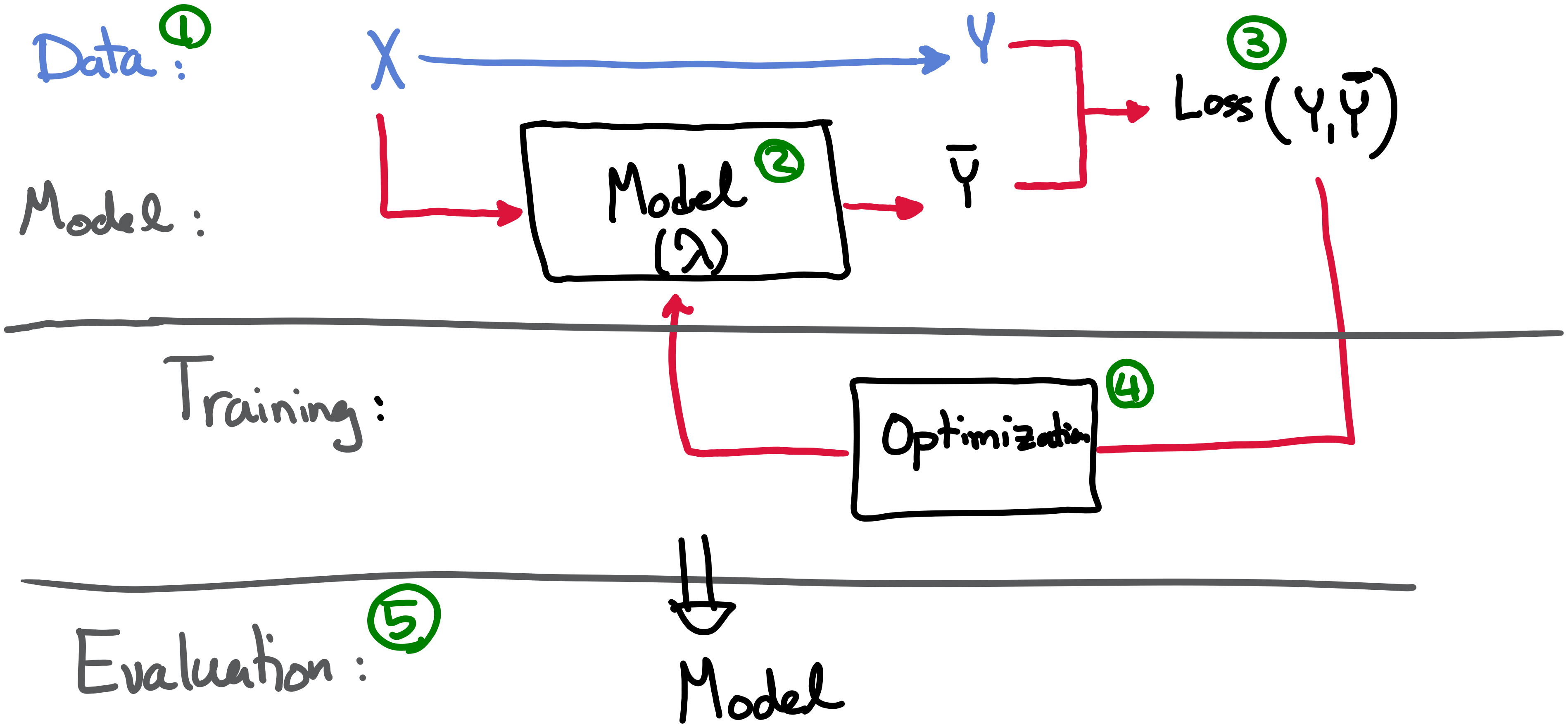
Machine Learning in Physics: Loss Functions

Sadegh Raeisi



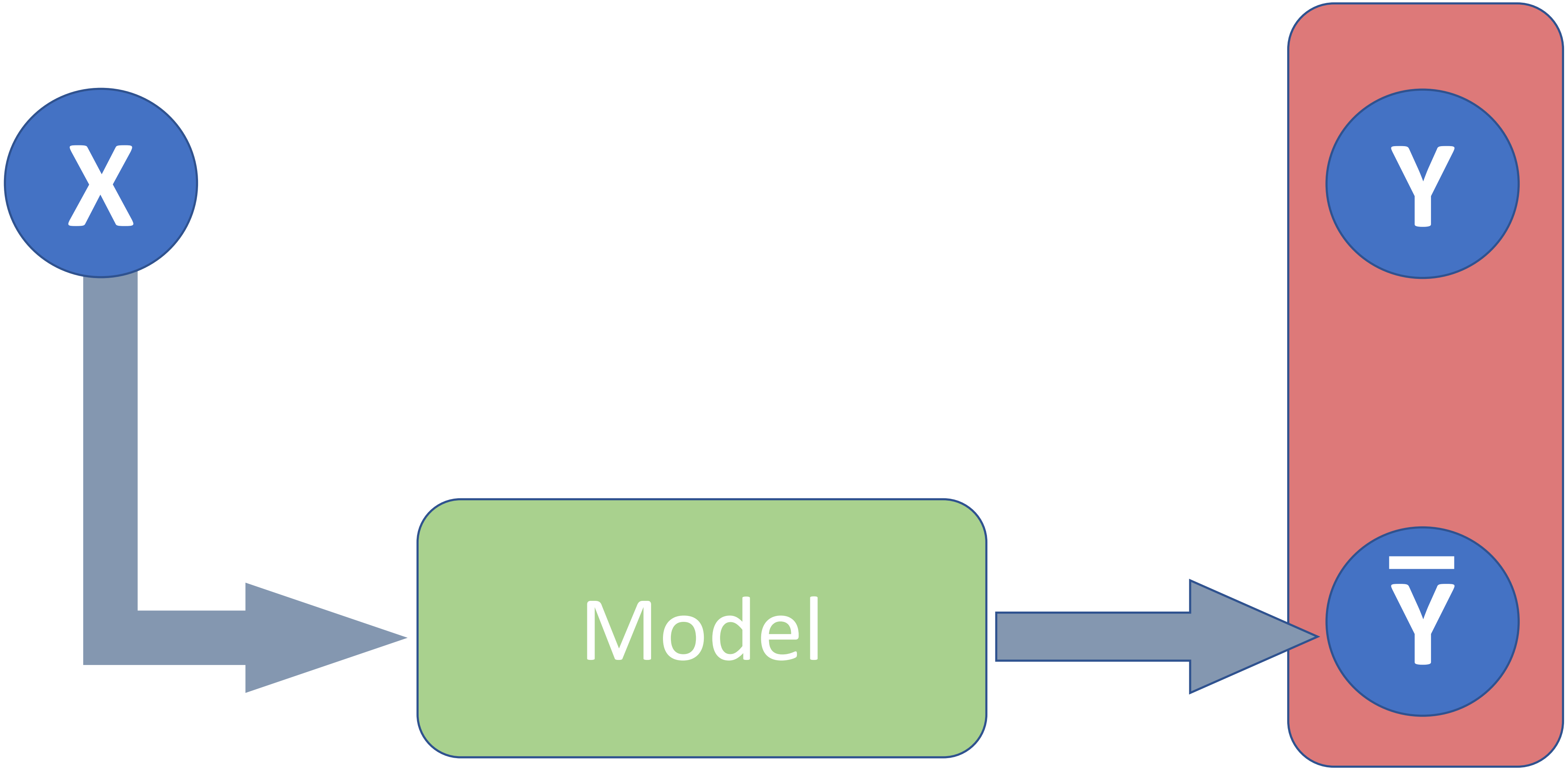
Supervised: Ingredients



Outline

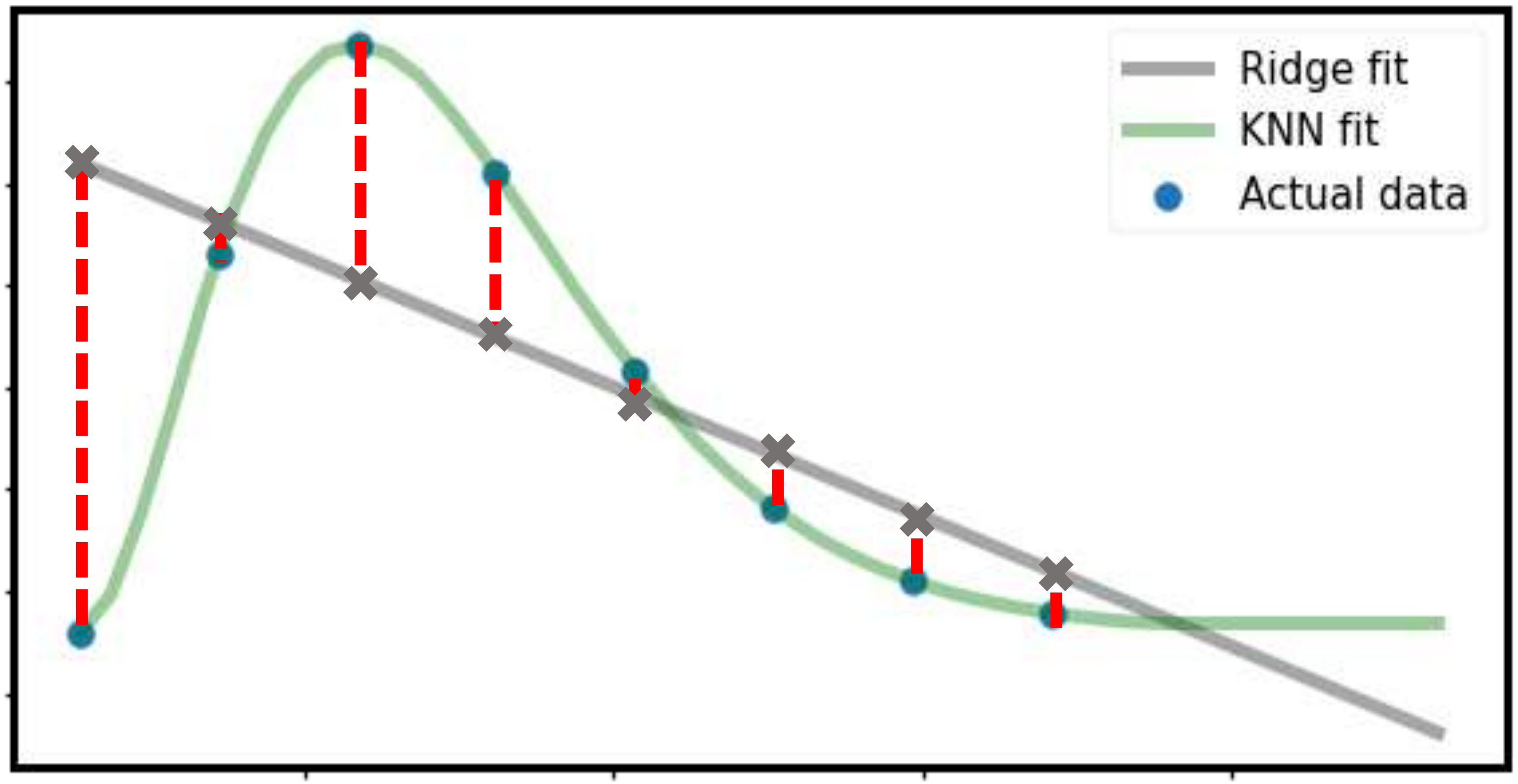
- Concept
- Regression
- Classification
- Other Loss functions

Concept



X
Model
Y
Ȳ

How close are the predictions?

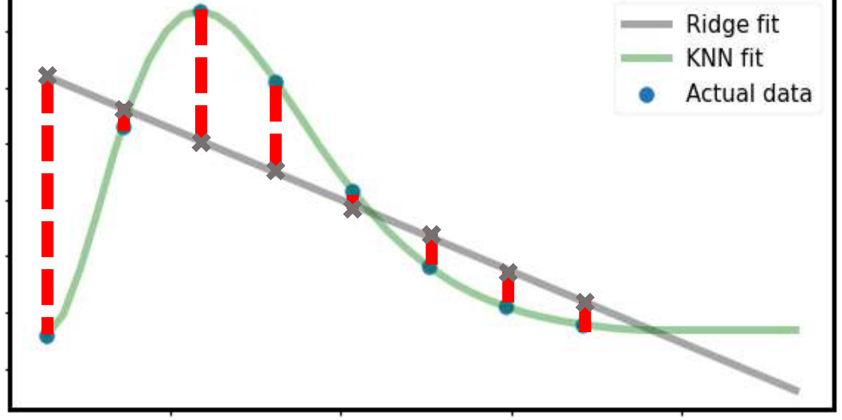


How can we quantify the difference?

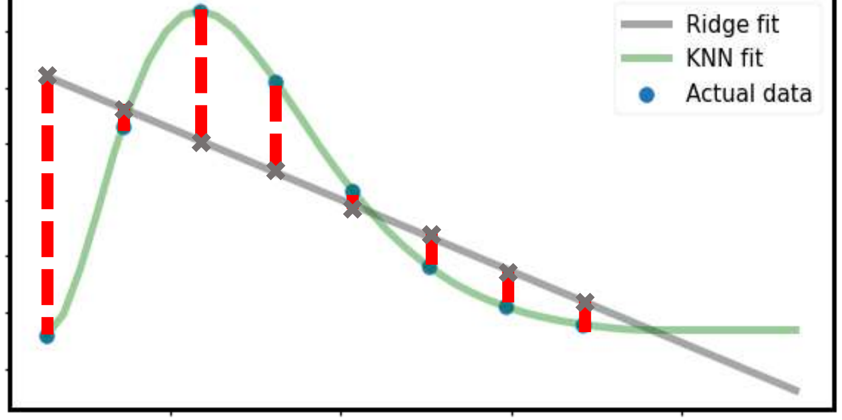
$$\text{Dist}(Y, \bar{Y})$$

Regression

Minkowski distance



$$D(Y,\bar{Y}) = \left(\sum_i \left| Y^i - \overline{Y^i} \right|^p \right)^{\frac{1}{p}}$$



Minkowski distance

$$D_1(Y,\bar{Y}) = \sum_i \left|Y^i - \bar{Y^i}\right|$$

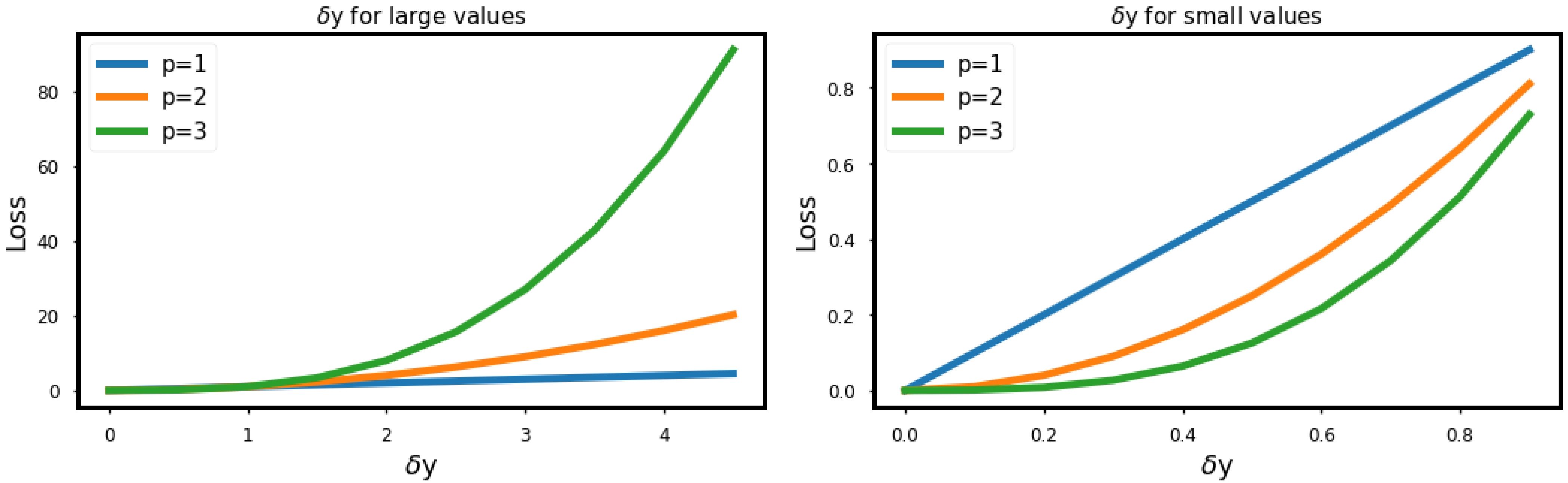
$$D_2(Y,\bar{Y}) = \sqrt{\sum_i \left|Y^i - \bar{Y^i}\right|^2}$$

➡ Penalizes larger deviations more

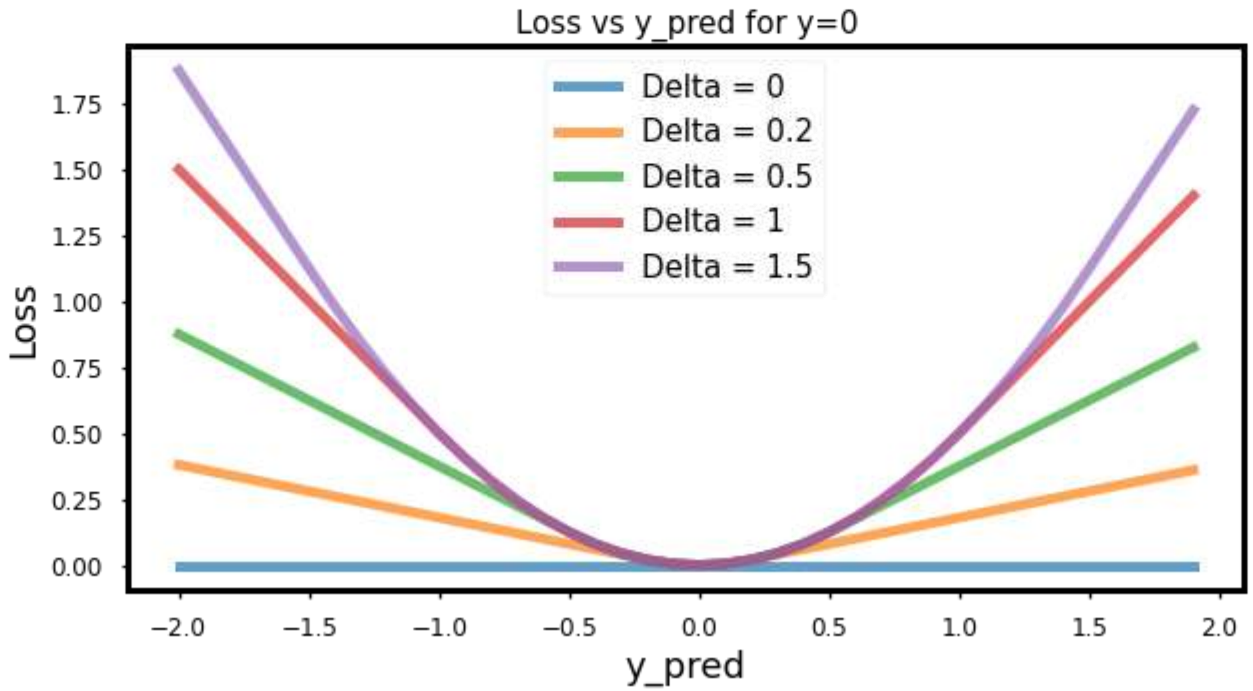
$$D_\infty(Y,\bar{Y}) = \max_i \left|Y^i - \overline{Y^i}\right|$$

➡ Worst case

Minkowski distance



Huber: the best of both worlds

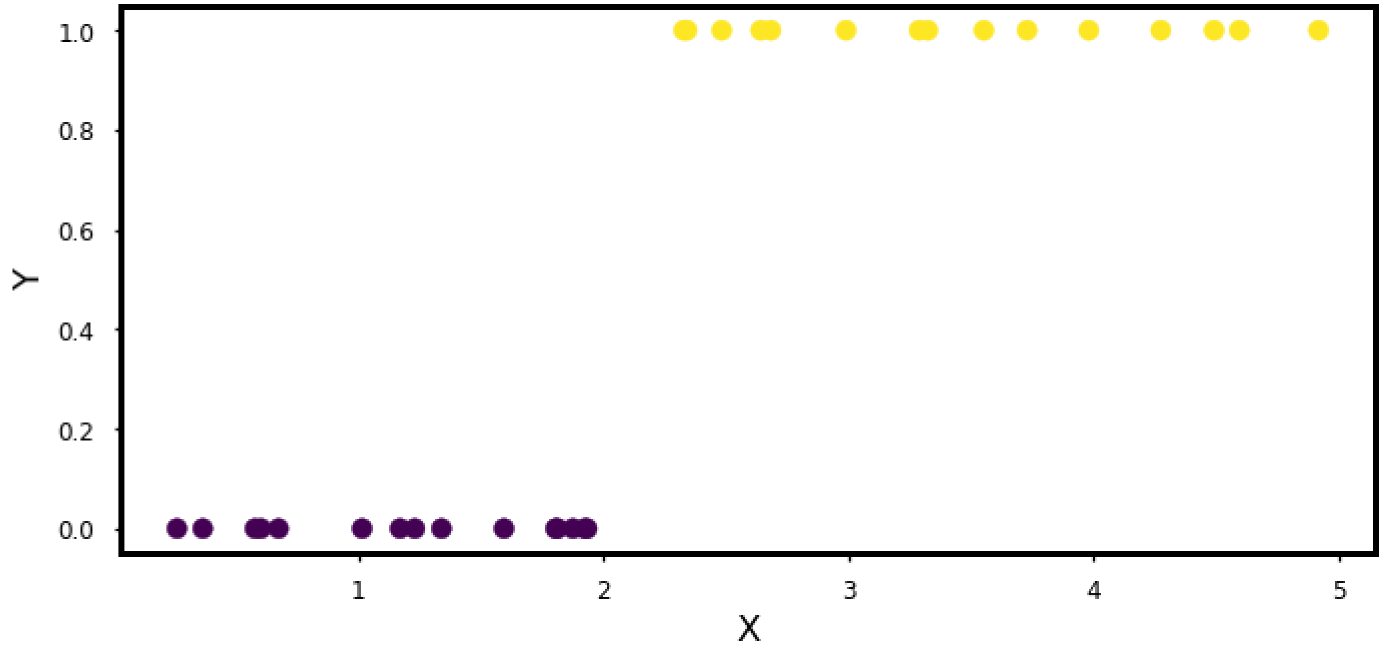


What other functions?

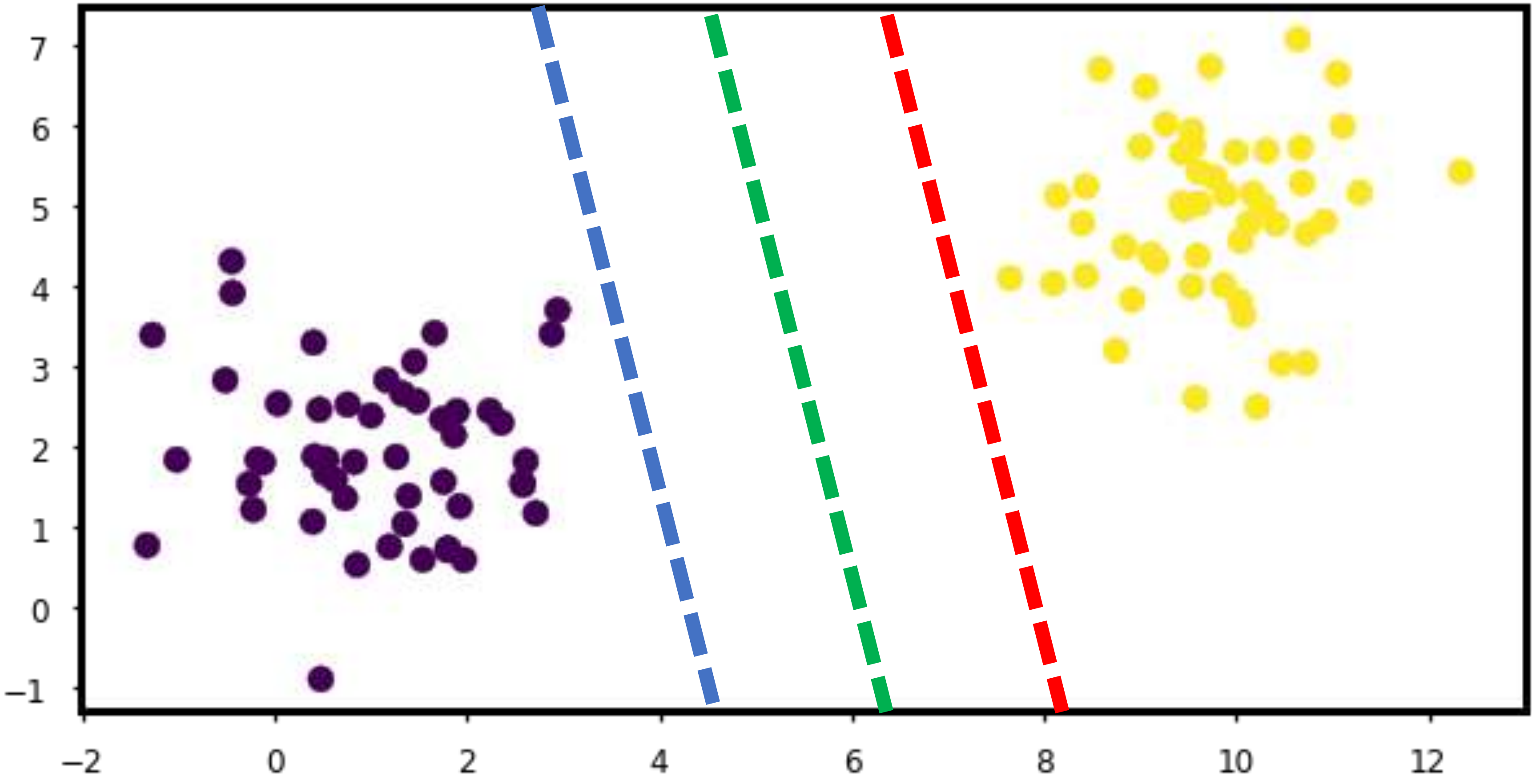
Take a look at scikit-learn!

Classification

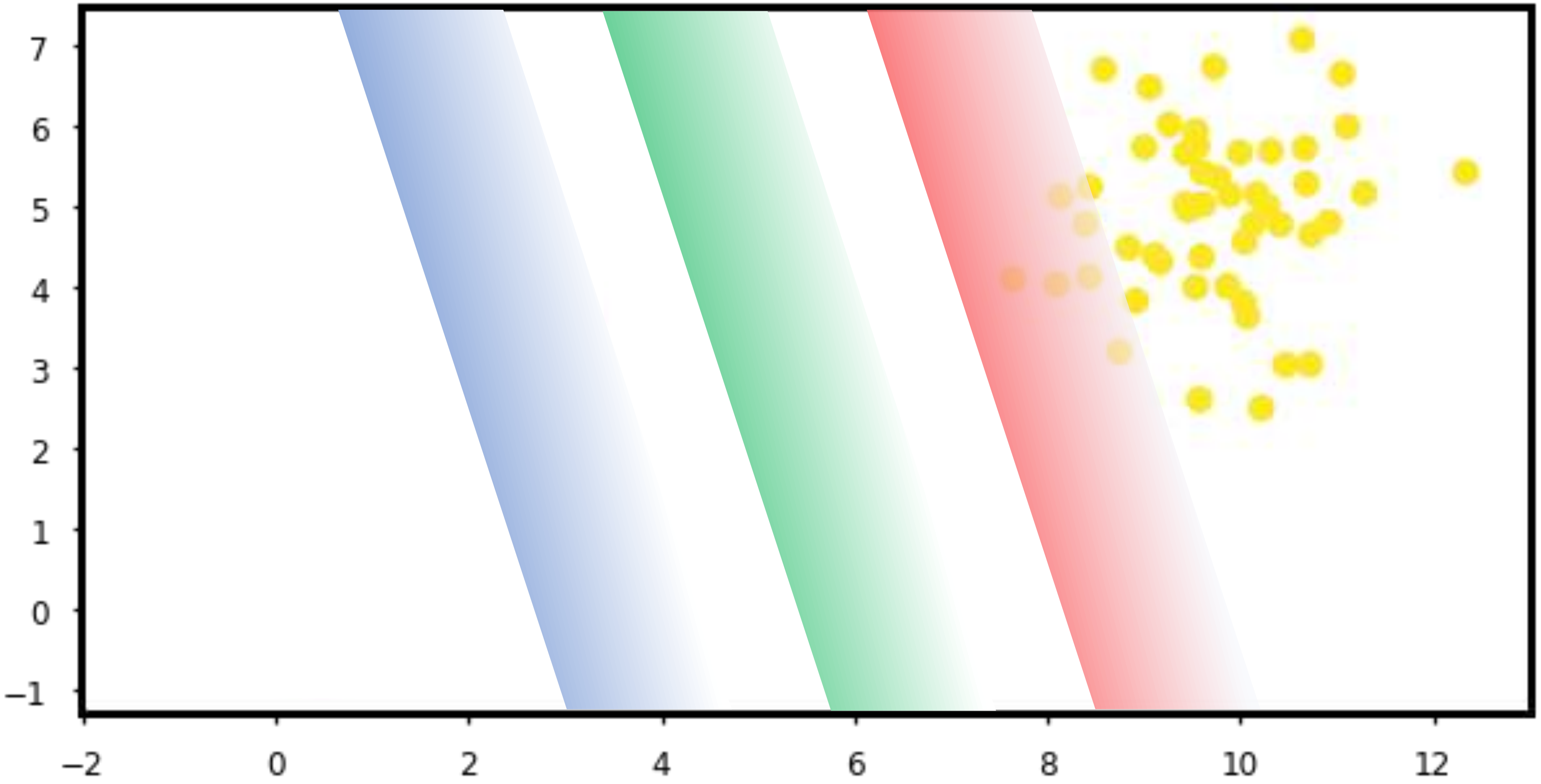
We can use regression loss functions



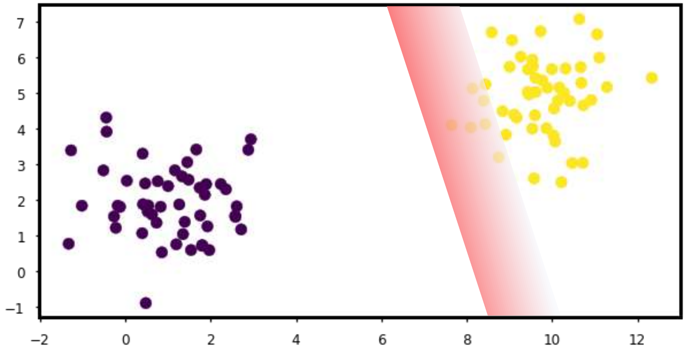
Hinge

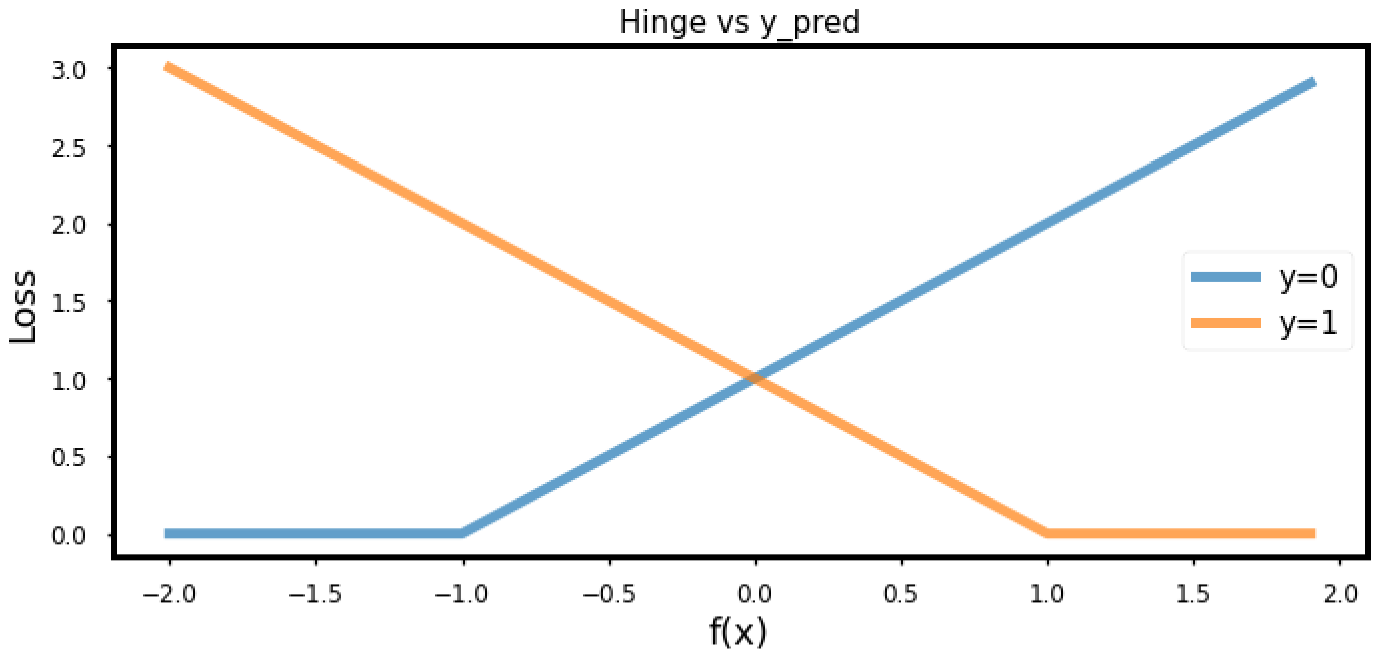


Hinge

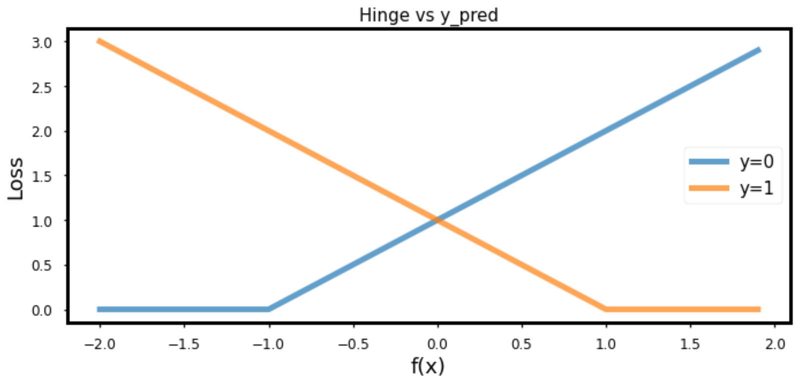


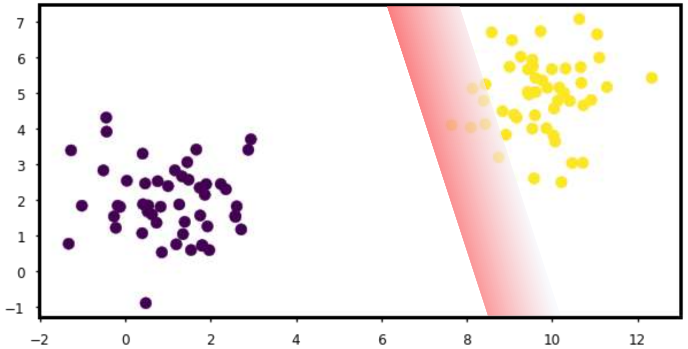
Hinge


7
6
5
4
3
2
1
0
−1
−2
0
2
4
6
8
10
12


Hinge vs y_pred
Loss
f(x)
y=0
y=1
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Hinge



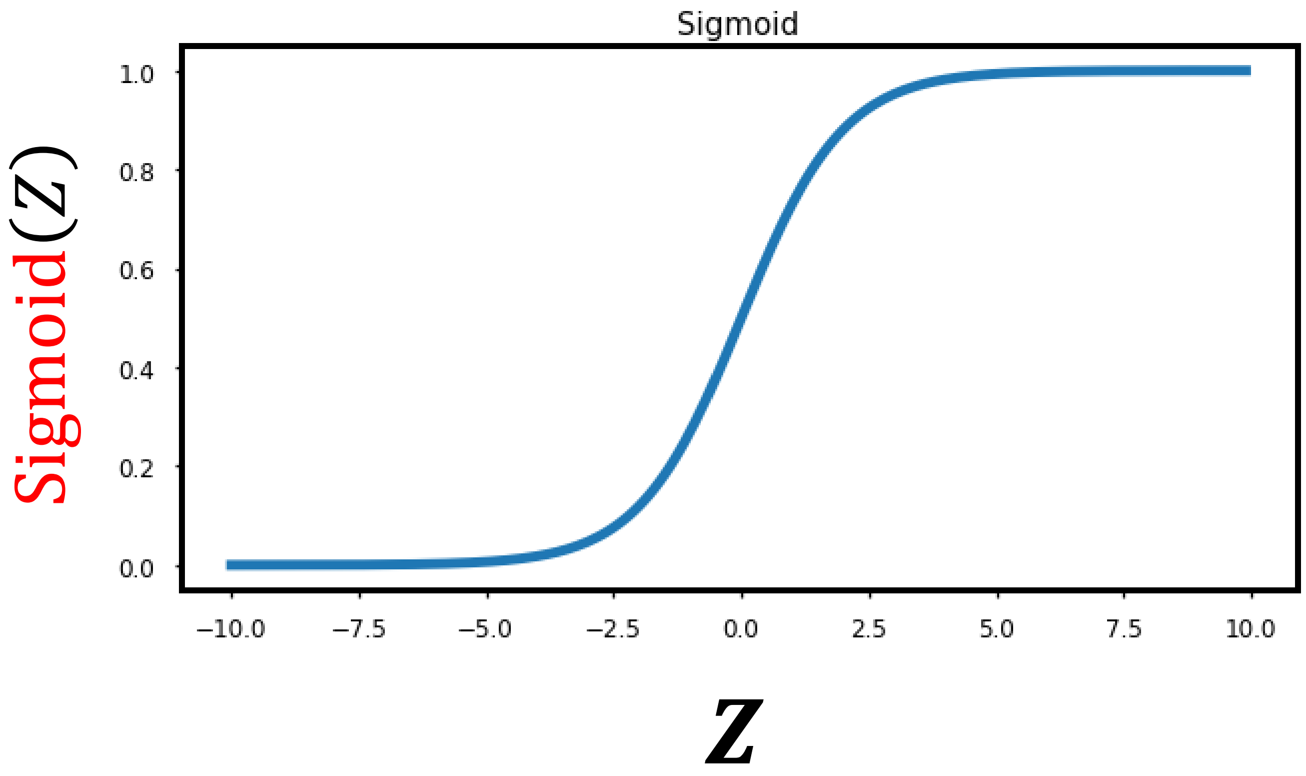


$$D_h(Y, \bar{Y}) = \max(0, 1 - Y^i * f_w(X^i))$$

Note that f_w is the decision boundary, not the class.
Also note that $Y^i = \pm 1$ and not $\{0,1\}$.

Probability:

Logistic Regression

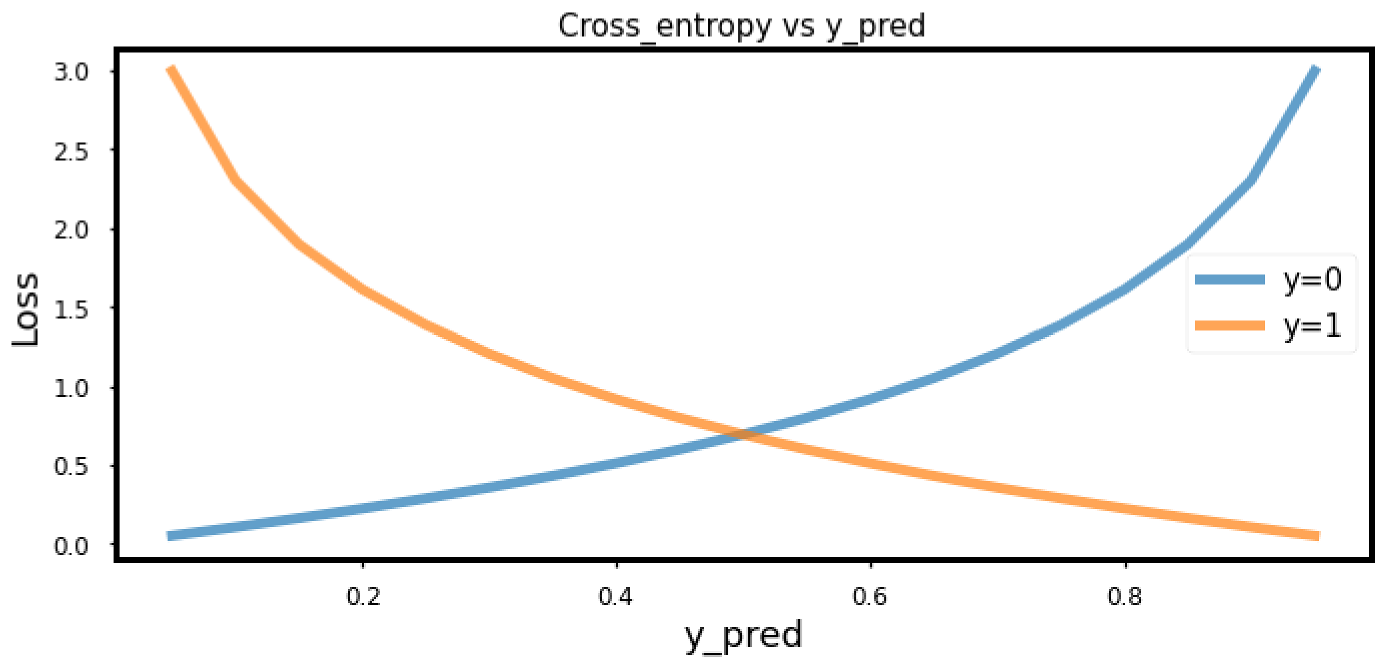


Probability: Cross-entropy

$$D_{CE}(Y,\bar{Y}) = -\sum_i Y^i \log \bar{Y}^i + (1 - Y^i)\log(1 - \bar{Y}^i)$$

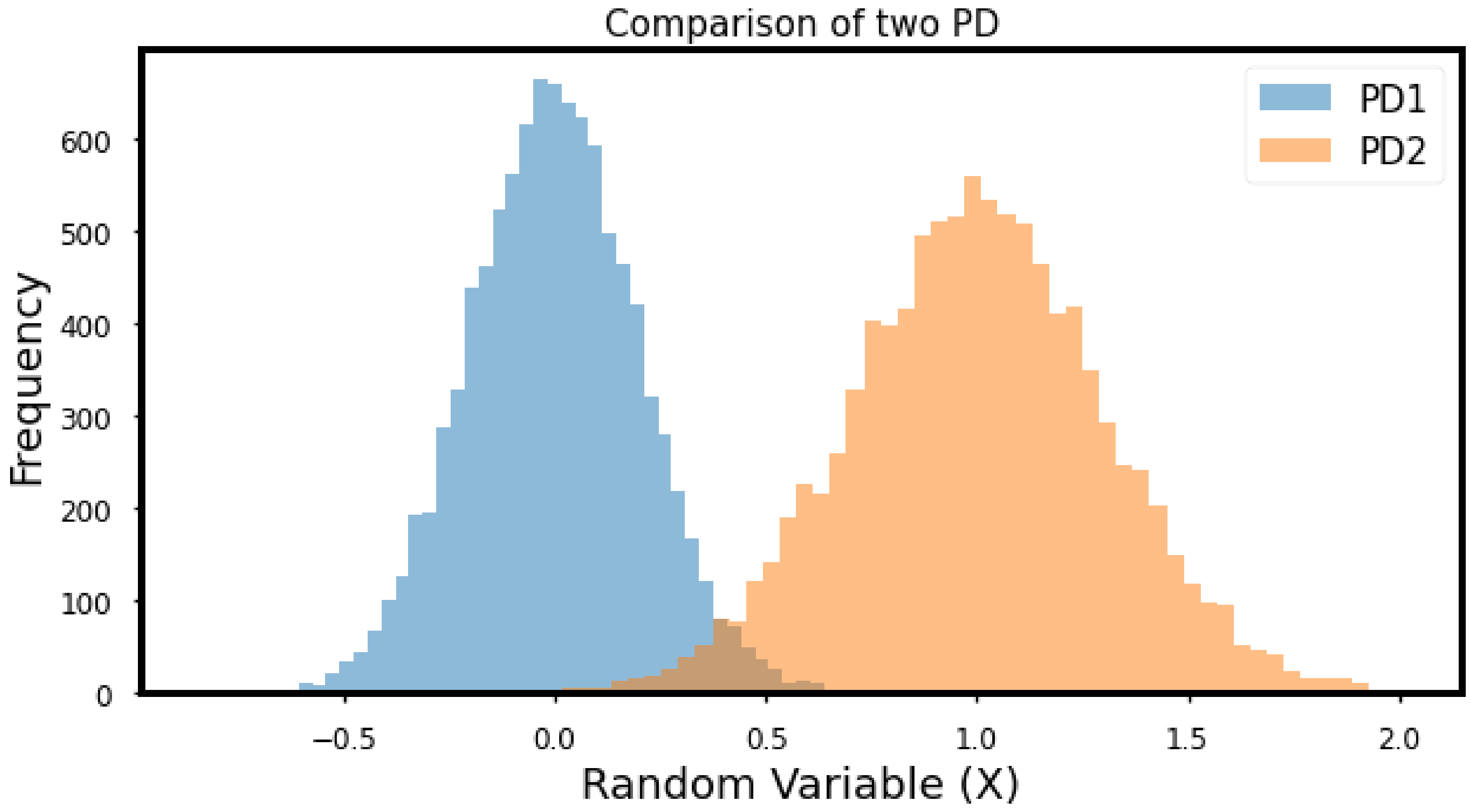
Probability: Cross-entropy

$$D_{CE}(Y, \bar{Y}) = -\sum_i Y^i \log \bar{Y}^i + (1 - Y^i) \log(1 - \bar{Y}^i)$$

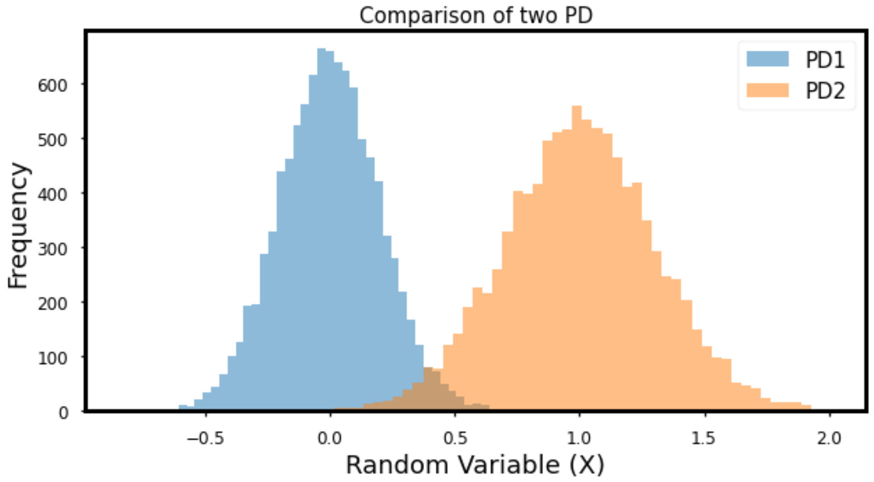


Other loss functions

Probability Distributions



Probability Distributions



- Kullback–Leibler divergence (Relative Entropy)

$$D_{KL}(PD_1||PD_2) = \sum_X PD_1(X)\log\left(\frac{PD_1(X)}{PD_2(X)}\right)$$

What would be a good loss function for a multi-class problem?

- $Y \in \{-1, 0, 1\}$

What would be a good loss function for $Y \in \{0, 1\}^{\otimes n}$?

Example:

$$Y^1 = (0, 0, 1, 0, 1)$$

Significance of the Loss function

How do we choose the right loss function?

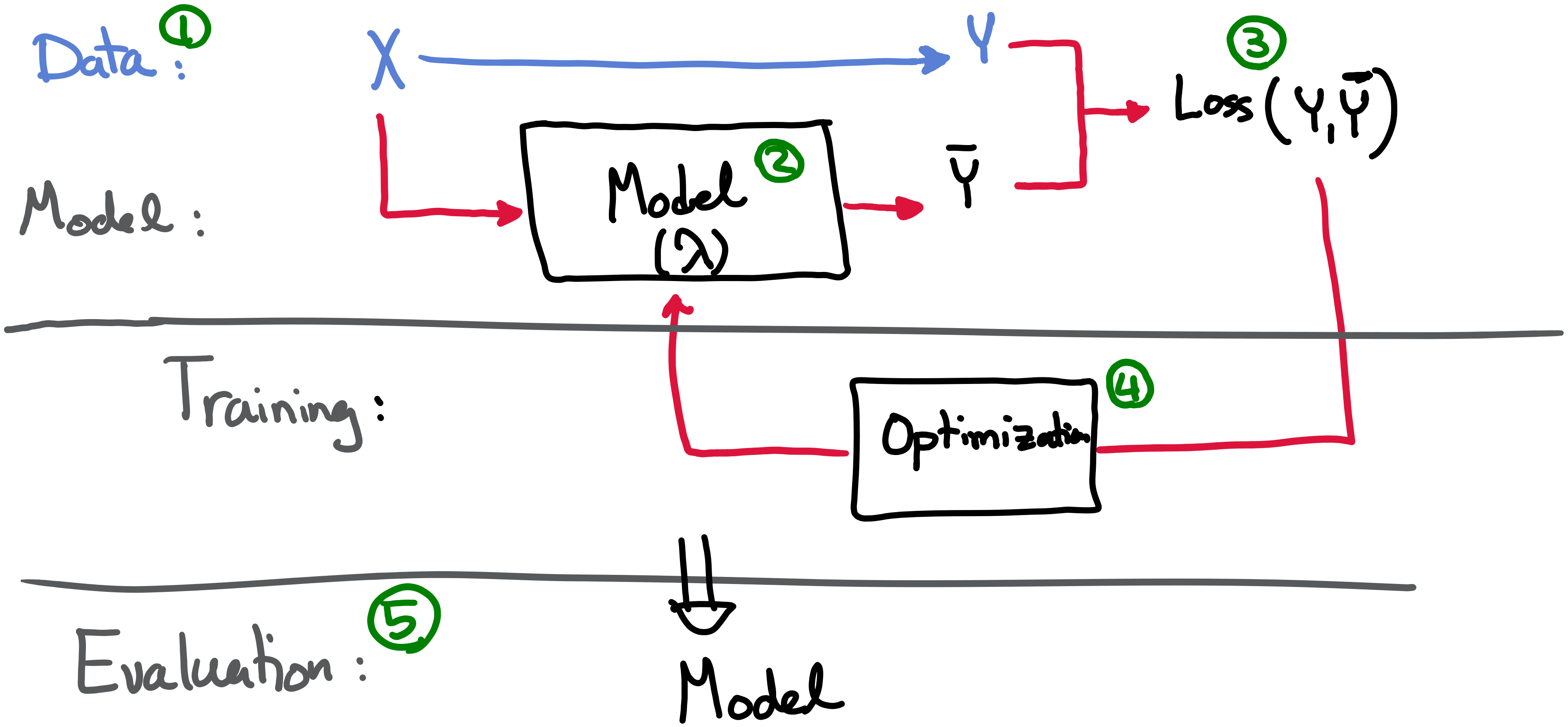
- Convexity

- Objective

How does this choice affect the model?

See the NB.

So far ...



Data: ①
X
Y
Model:
Model ②
(λ)
Ȳ
Loss(Y, Ȳ) ③
Training:
Optimization ④
Evaluation: ⑤
Model