January 2025

Module 4 – Regularization

Data Science For Business

NYU | STERN

Quiz time!

Quiz discussion!

Q1

What does increasing the degree of a polynomial model typically do to the model's fit?

- Improves the fit on both training and test data indefinitely
- Reduces complexity and risk of overfitting
- Increases complexity, potentially leading to overfitting
- Has no significant impact on the fit

Q1

What does increasing the degree of a polynomial model typically do to the model's fit?

- Improves the fit on both training and test data indefinitely
- Reduces complexity and risk of overfitting
- [✓ Increases complexity, potentially leading to overfitting]
- Has no significant impact on the fit

Q2

Why is cross-validation a better approach than a simple train-test split?

- It uses more training data for model fitting
- It reduces the variance in evaluation metrics for test sets
- It tests every instance at least once
- All of the above

Q2

Why is cross-validation a better approach than a simple train-test split?

- It uses more training data for model fitting
- It reduces the variance in evaluation metrics for test sets
- It tests every instance at least once
- All of the above

Q3

Which of the following indicates overfitting in a model?

- Low accuracy on training data but high on test data
- High accuracy on training data but low on test data
- High accuracy on both training and test datasets
- Equal accuracy on training and test datasets

Q3

Which of the following indicates overfitting in a model?

- Low accuracy on training data but high on test data
- [✓ High accuracy on training data but low on test data]
- High accuracy on both training and test datasets
- Equal accuracy on training and test datasets

Q4

We've now looked at two types of supervised learning problems. What are they and what is one model for each type of problem?

Q4

We've now looked at two types of supervised learning problems. What are they and what is one model for each type of problem?

Supervised Learning

Classification / Probability Estimation

- Decision tree
- Linear/ polynomial logistic regression

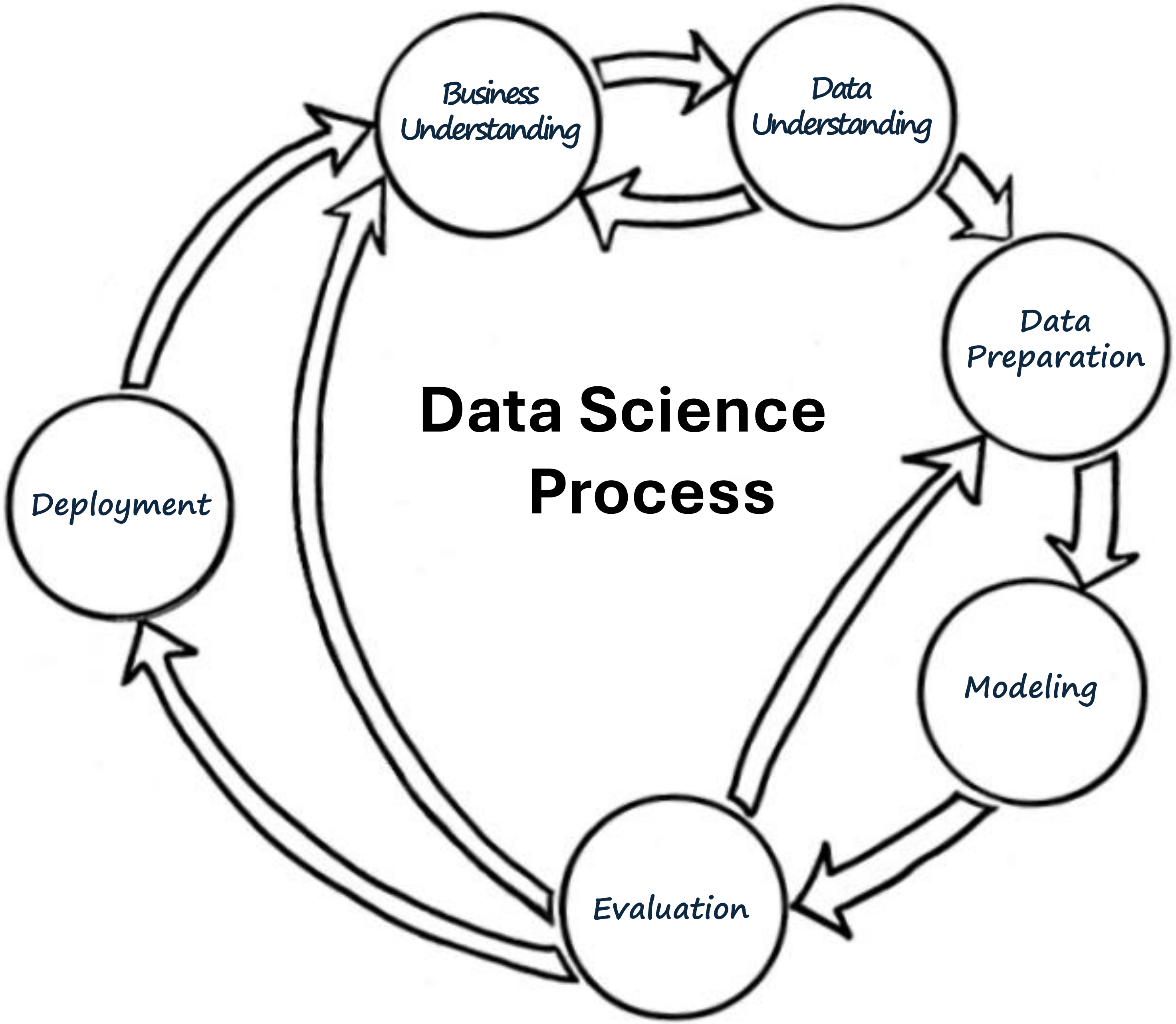
Regression

- Linear/polynomial regression
- Regression tree

Agenda

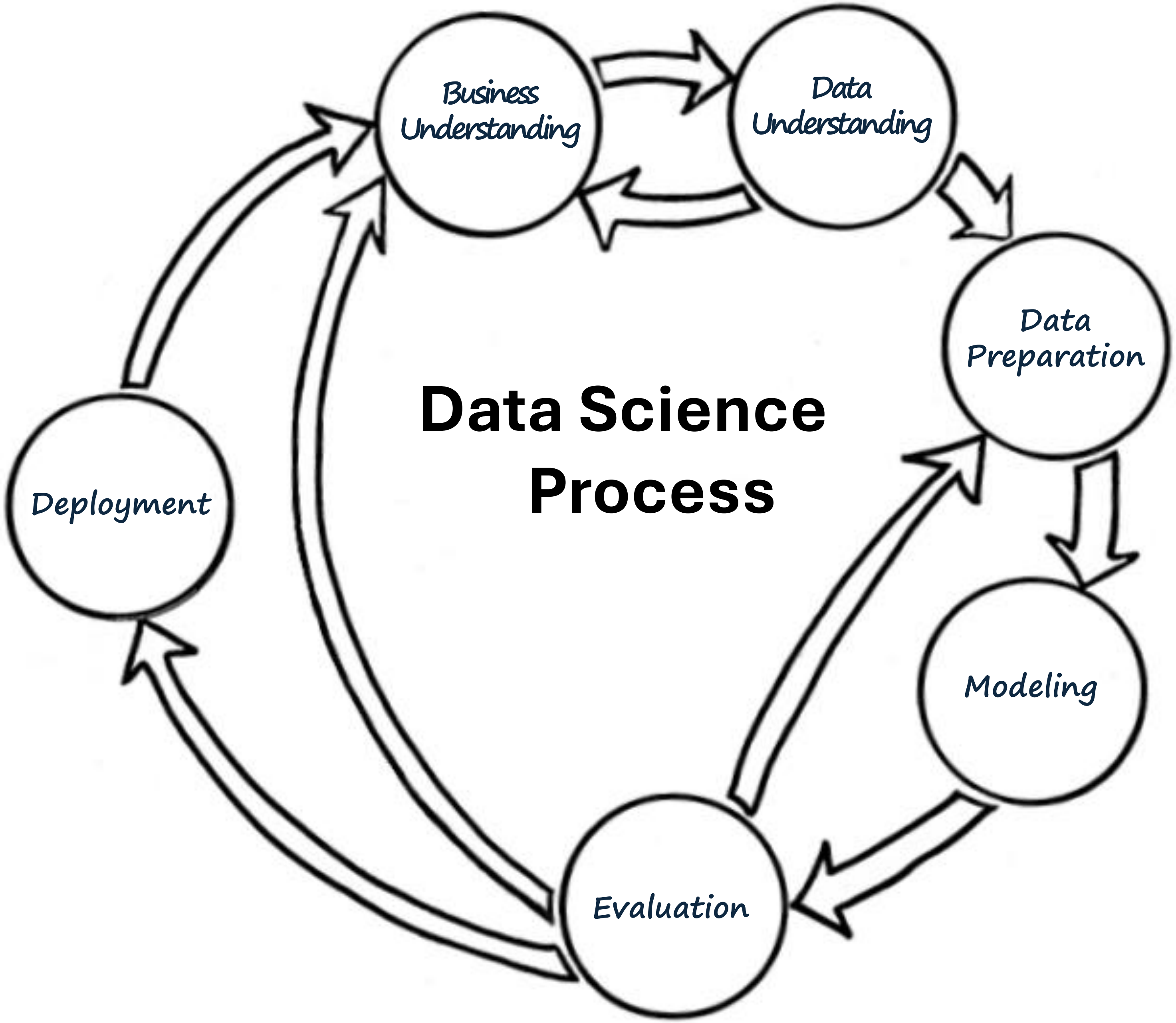
- **Week 1**
  - ~~**Module 1 (Thursday):** Intro to data science + Python for DS~~
  - ~~**Module 2 (Friday):** Intro to supervised learning~~
- **Week 2**
  - ~~**Module 3 (Monday):** Fitting models, generalization~~
  - **Module 4 (Tuesday):** Regularization
  - **Module 5 (Wednesday):** Evaluation (ROC, cost visualization)
  - **Module 6 (Thursday):** Modeling text data
- **Week 3**
  - **Module 7 (Monday):** Neural networks, GenAI
  - **Module 8 (Tuesday):** Guest lecture(s)
  - **Module 9 (Wednesday):** Causal inference, AB testing, wrap up
  - **Final Exam (Thursday)**

Where we are


Business Understanding
Data Understanding
Data Preparation
Data Science Process
Deployment
Modeling
Evaluation

Where we are

Last class



Building Your Toolbox
Last class

Types of Tasks and Models

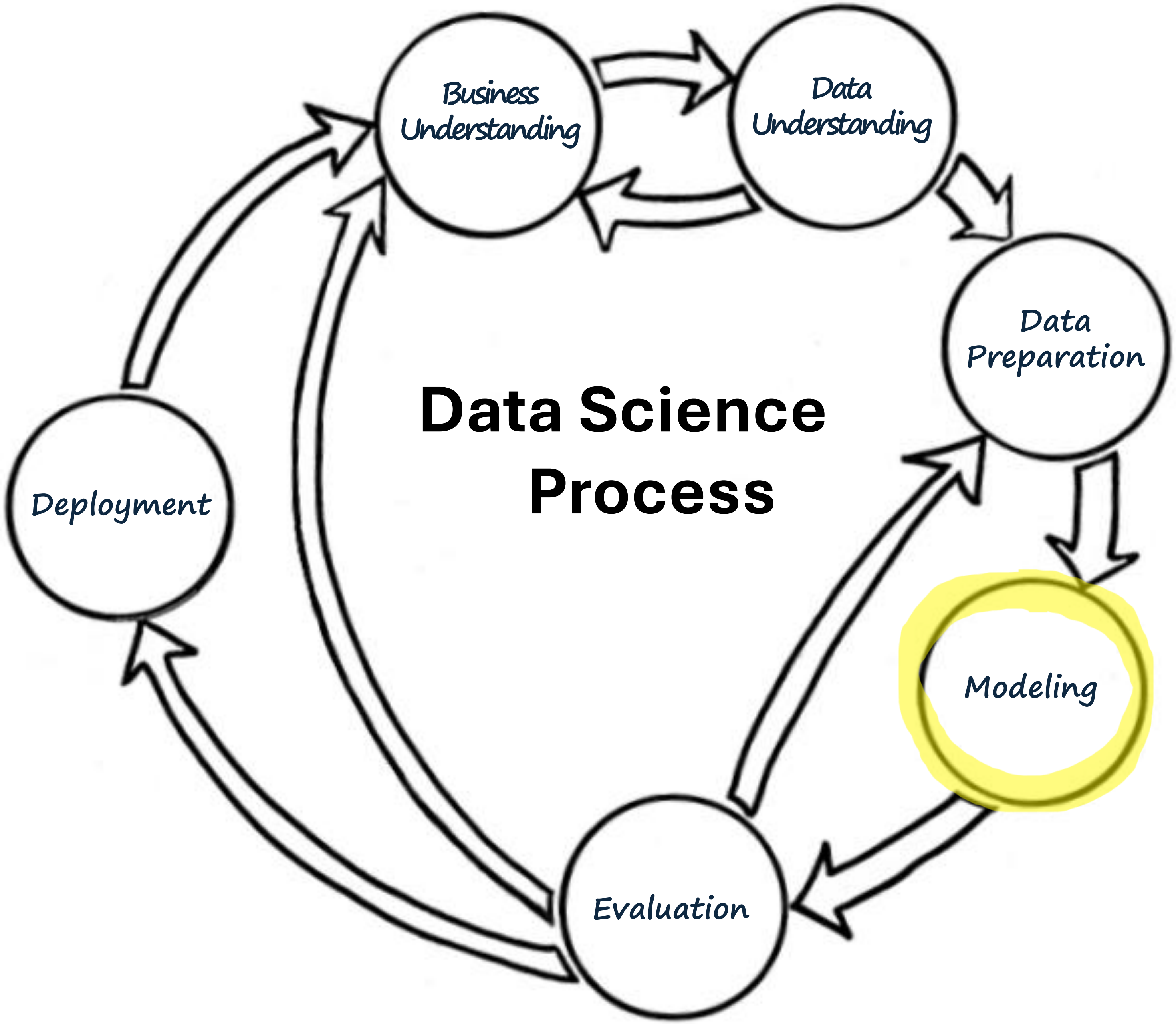
Supervised Learning

Classification / Probability Estimation

- Decision tree
- Linear/polynomial logistic regression

Regression

- Linear/polynomial regression
- Regression tree



Recap

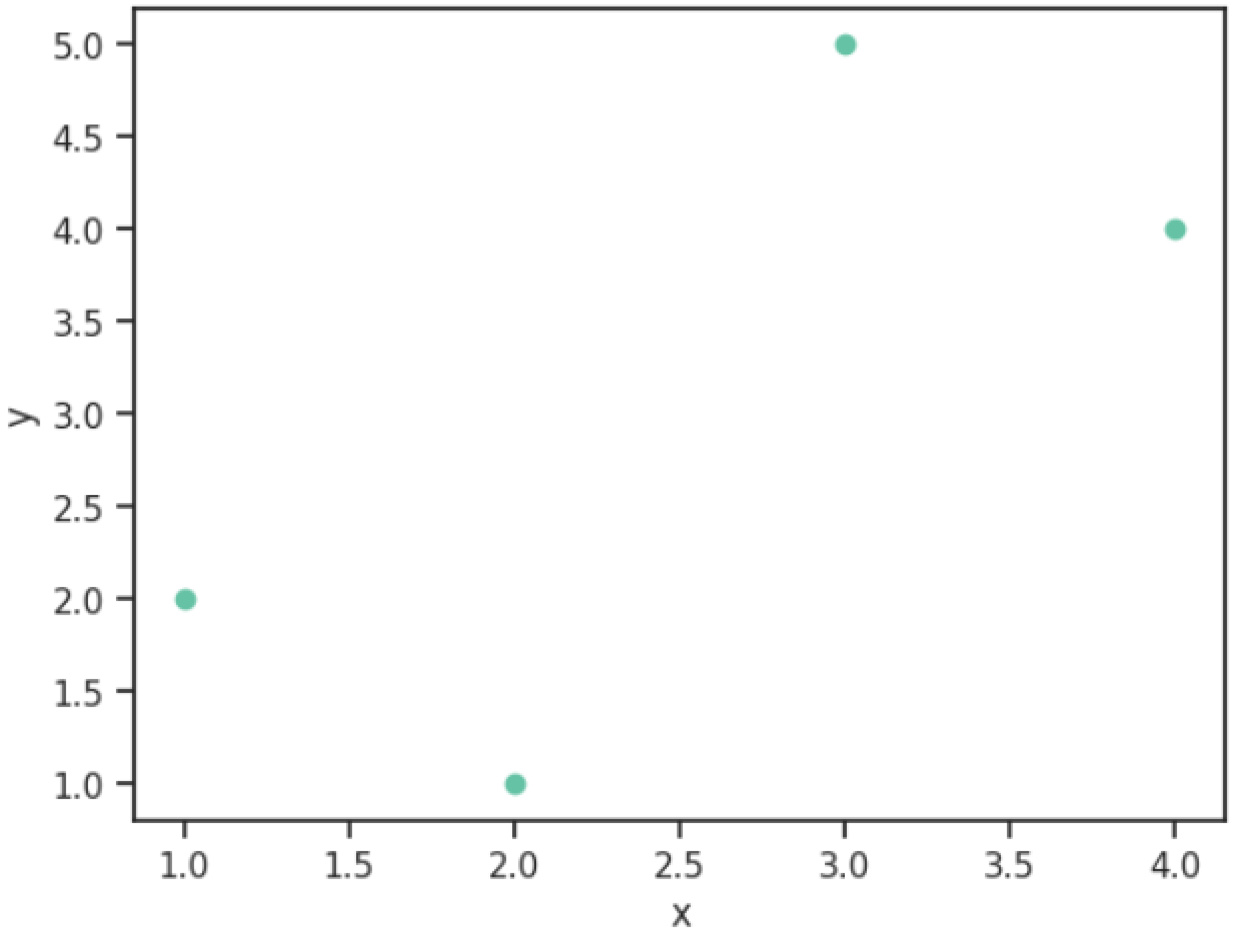
Last class

Linear/polynomial regression

Recap
Last class

Linear/polynomial regression

We talked briefly about how we use a **loss function** to pick the line of best fit



Recap

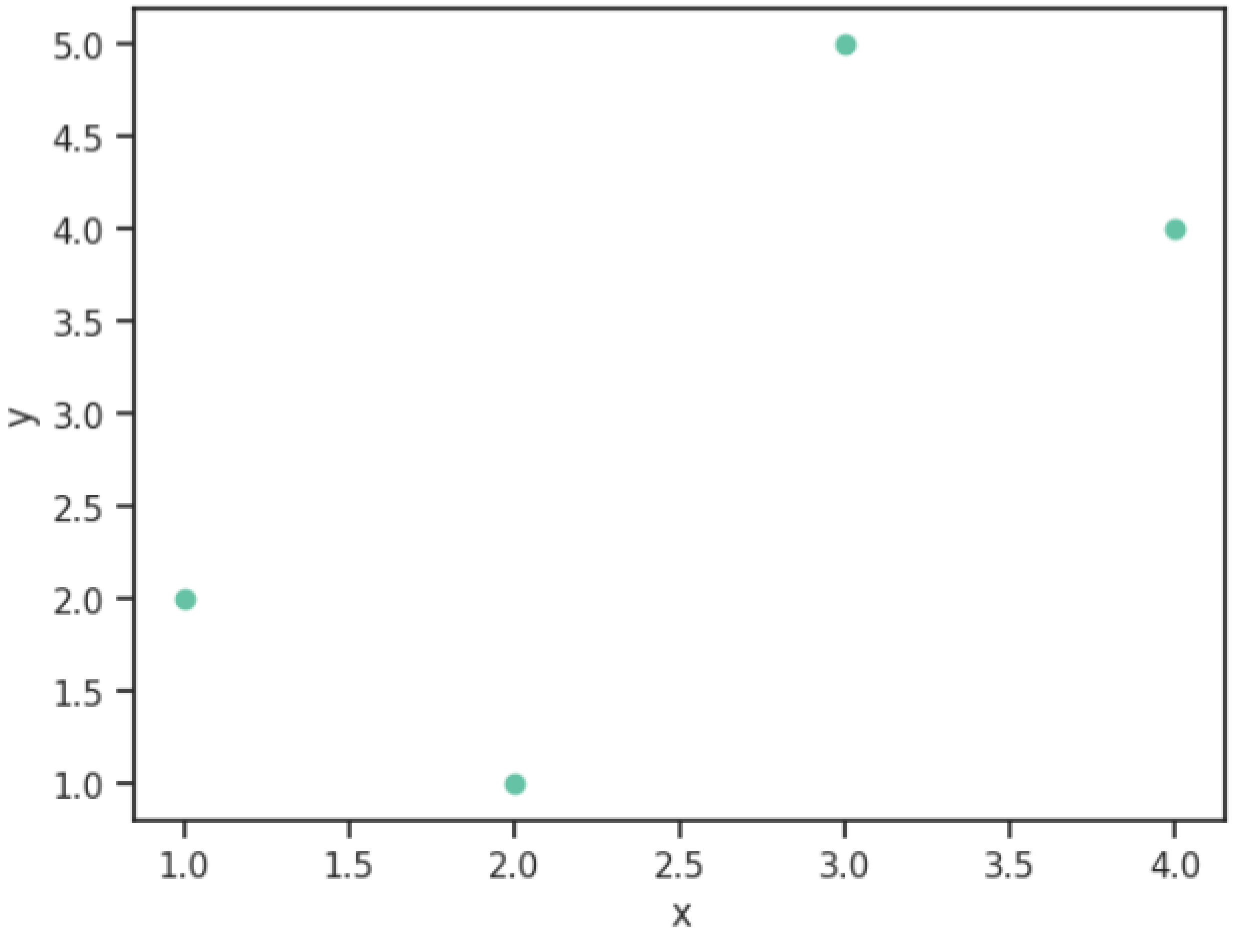
Last class

Linear/polynomial regression

We talked briefly about how we use a **loss function** to pick the line of best fit

This is very similar to the **objective function**

(Often this is the case: **objective function = - loss function**)



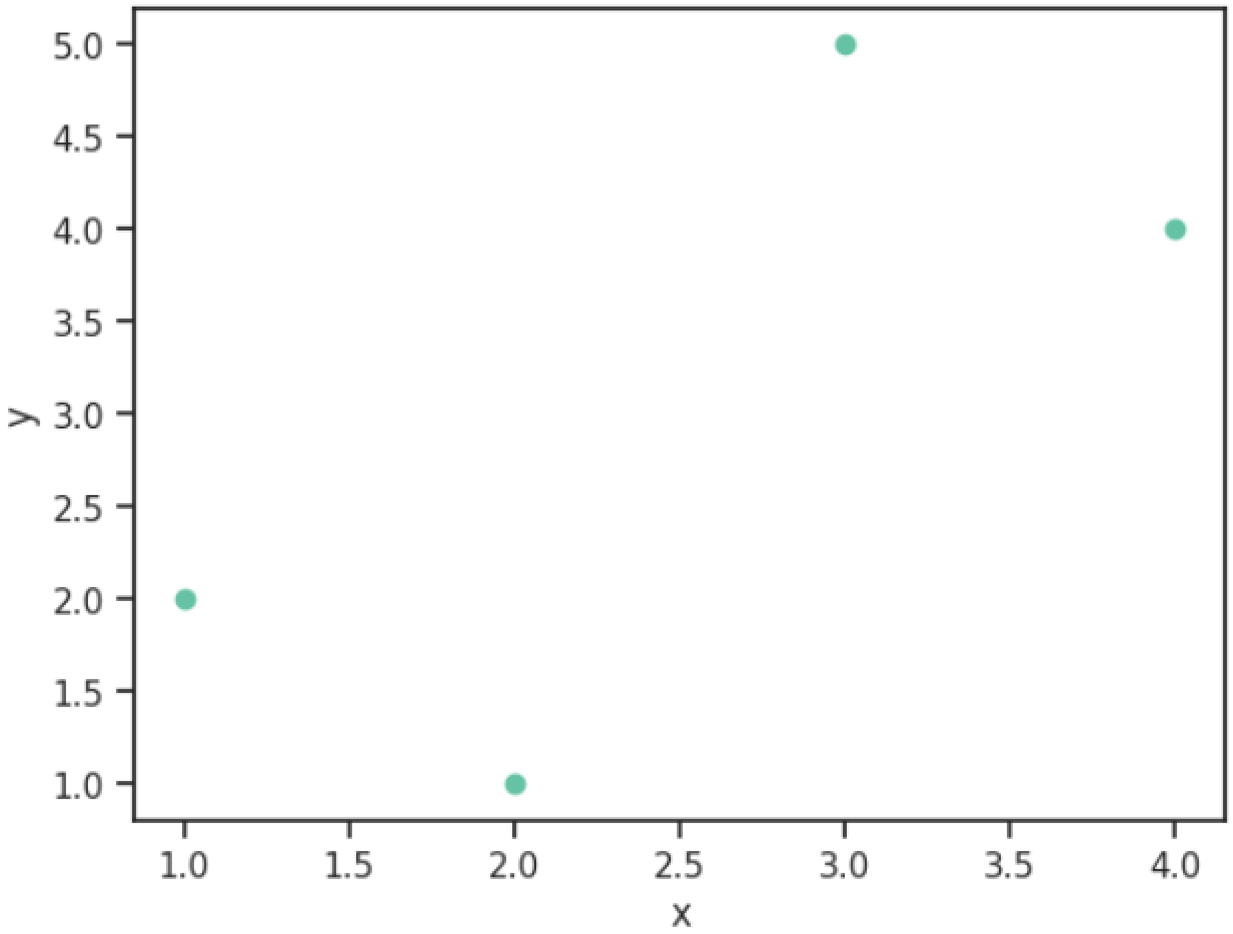
Recap

Last class

Linear/polynomial regression

In **regression** the **loss function** is typically: **Mean Squared Error**

What does this mean?



Recap

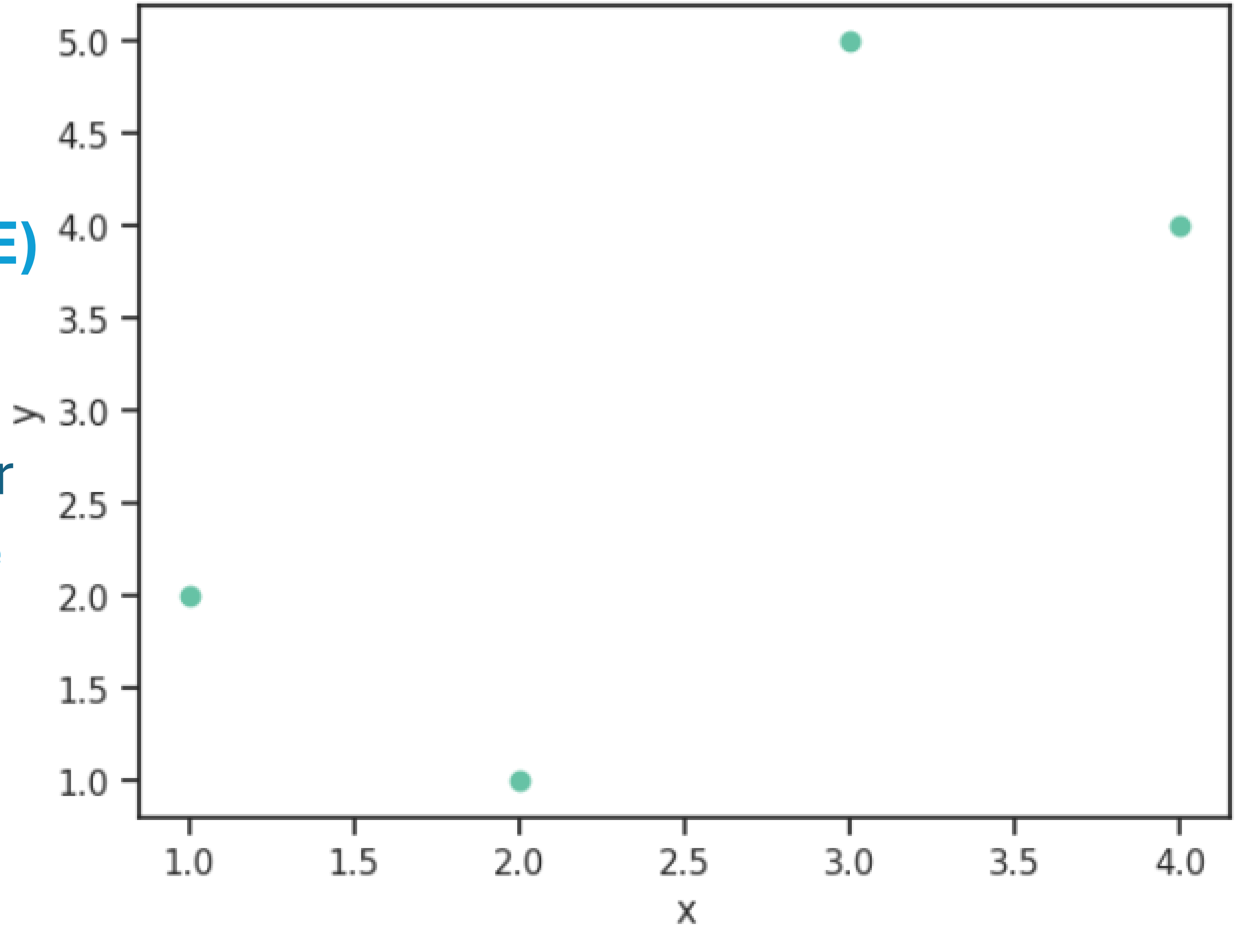
Last class

Linear/polynomial regression

In **regression** the **loss function** is typically: **Mean Squared Error (MSE)**

MSE = **average squared error** between our **model's prediction** for the target **and the true target value**

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$



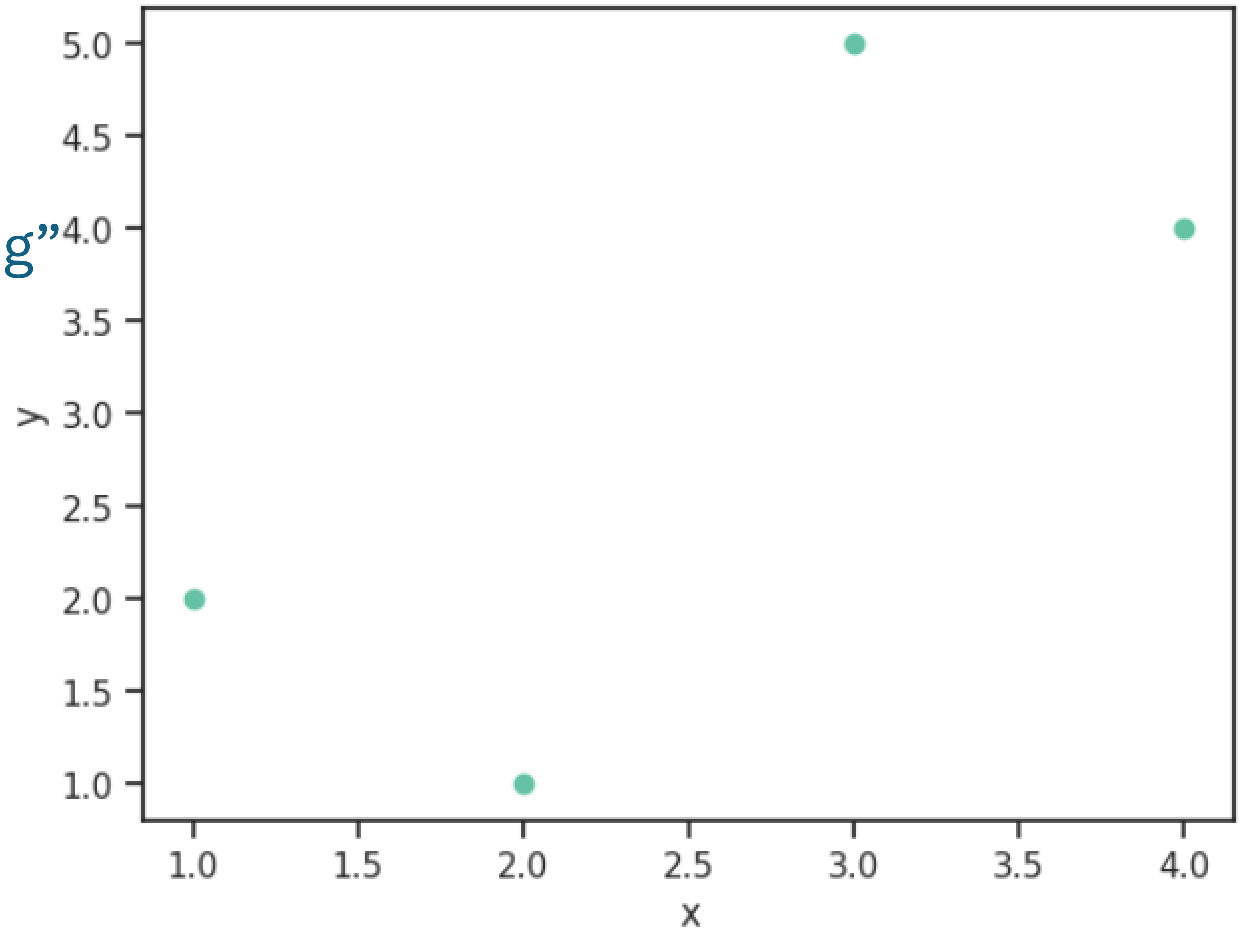
Recap

Last class

Linear/polynomial regression

If we find the line parameters that **minimize MSE**, we get a “good fitting” line

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$



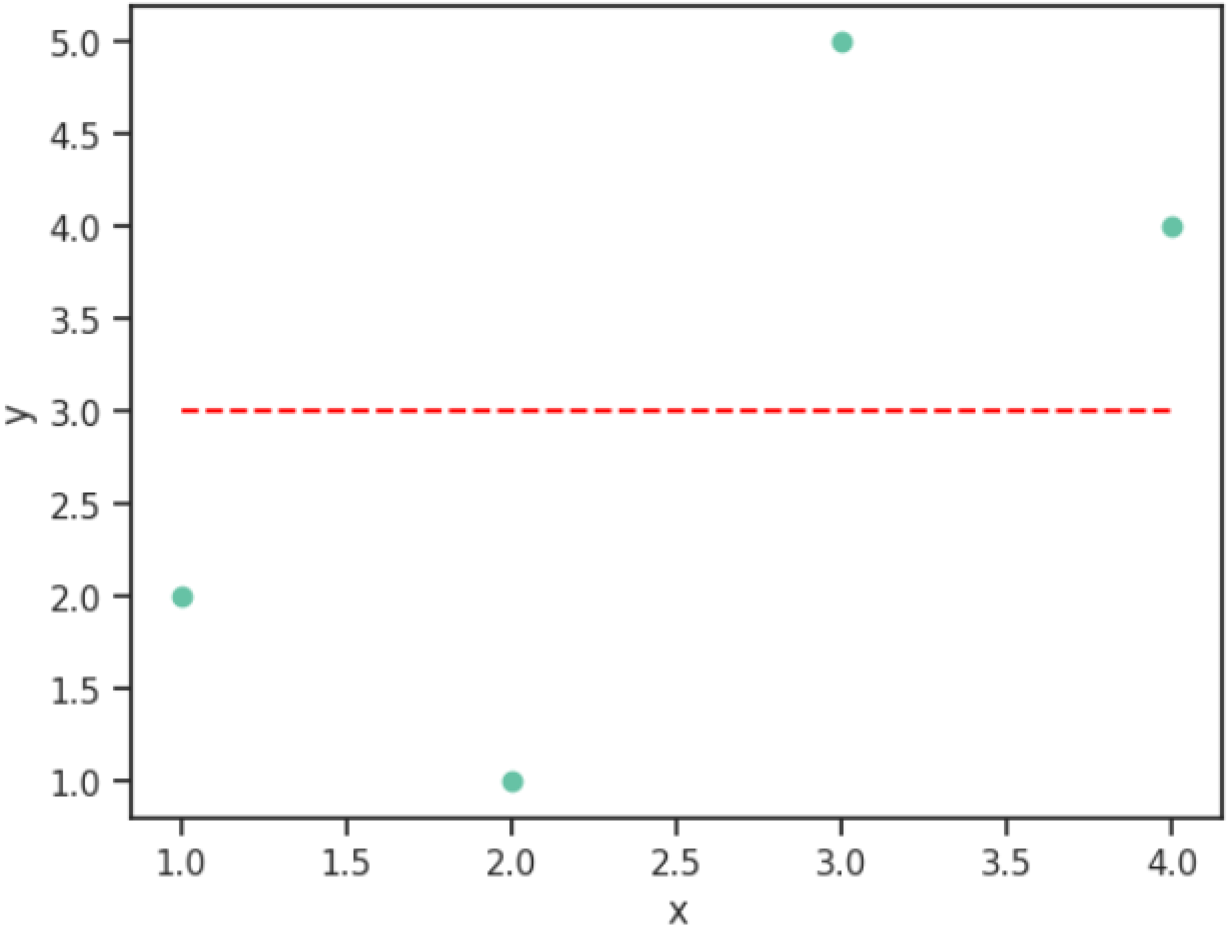
Recap

Last class

Linear/polynomial regression

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

y_hat = mx + b = w_1x + w_0



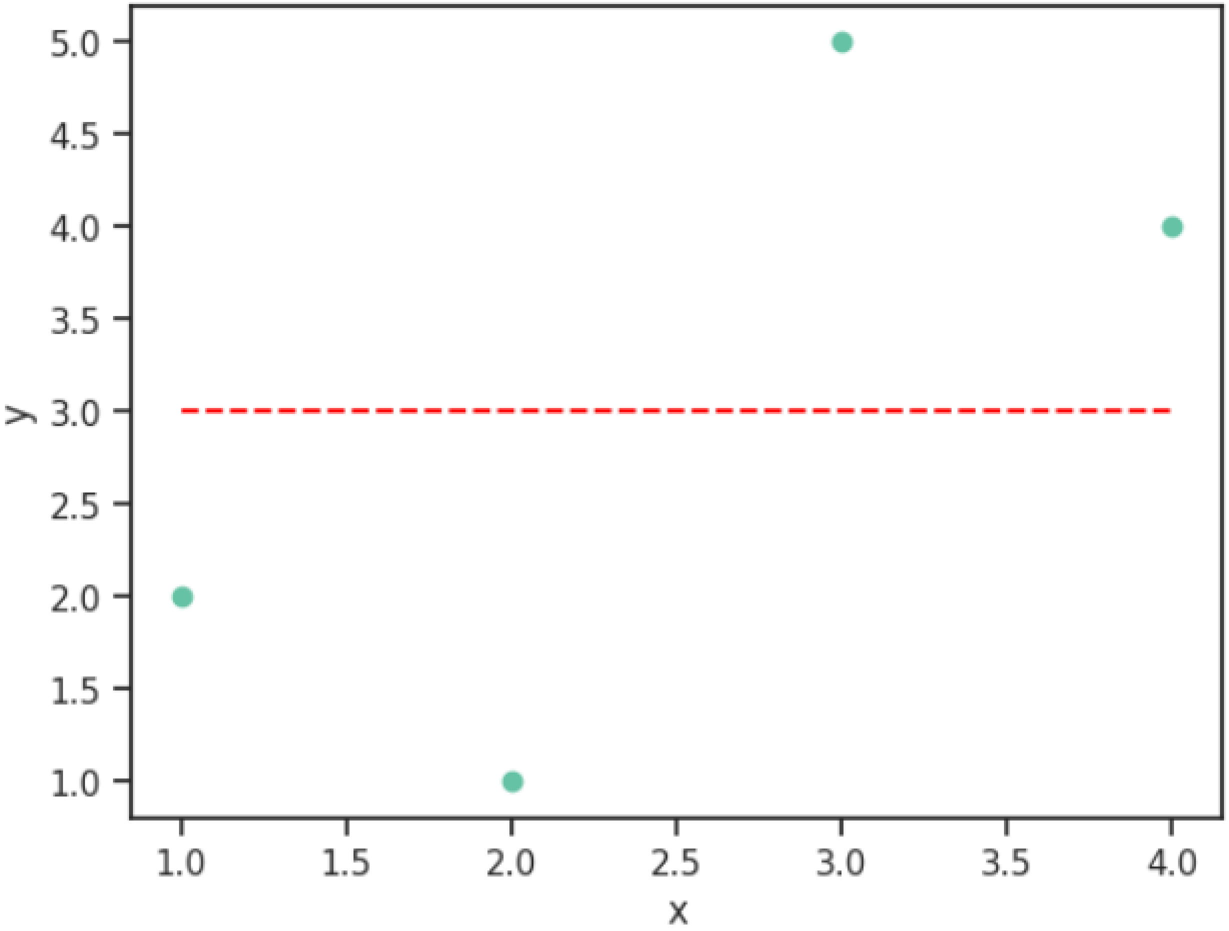
Recap
Last class

Linear/polynomial regression

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

y_hat = mx + b = w_1x + w_0

$w_1=0$
$w_0=3$



Recap

Last class

Linear/polynomial regression

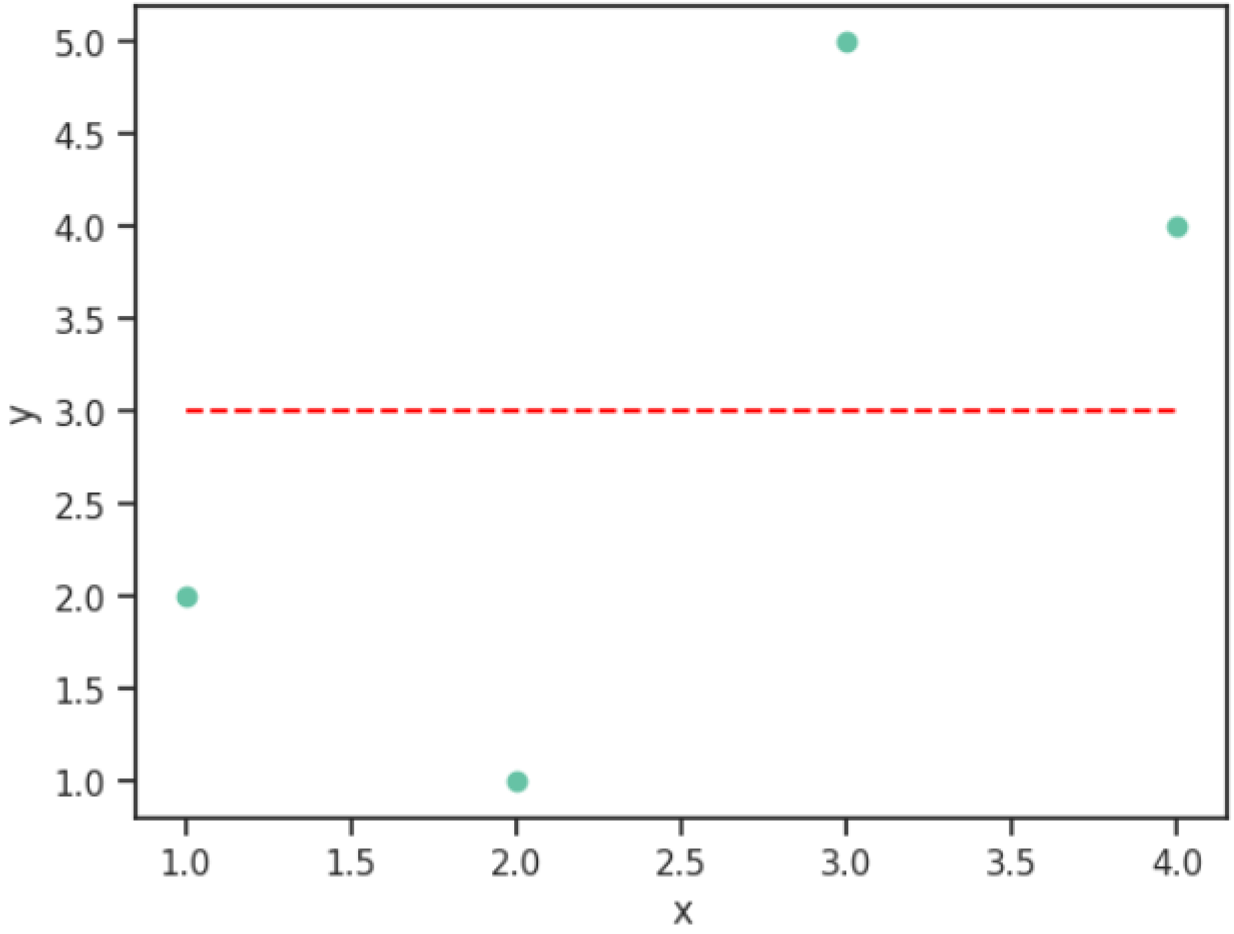
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$y\_hat = mx + b = w_1x + w_0$

$w_1=0$
$w_0=3$

```
y_hat = [3, 3, 3, 3]
```

```
MSE = 2.5
```



Recap

Last class

Linear/polynomial regression

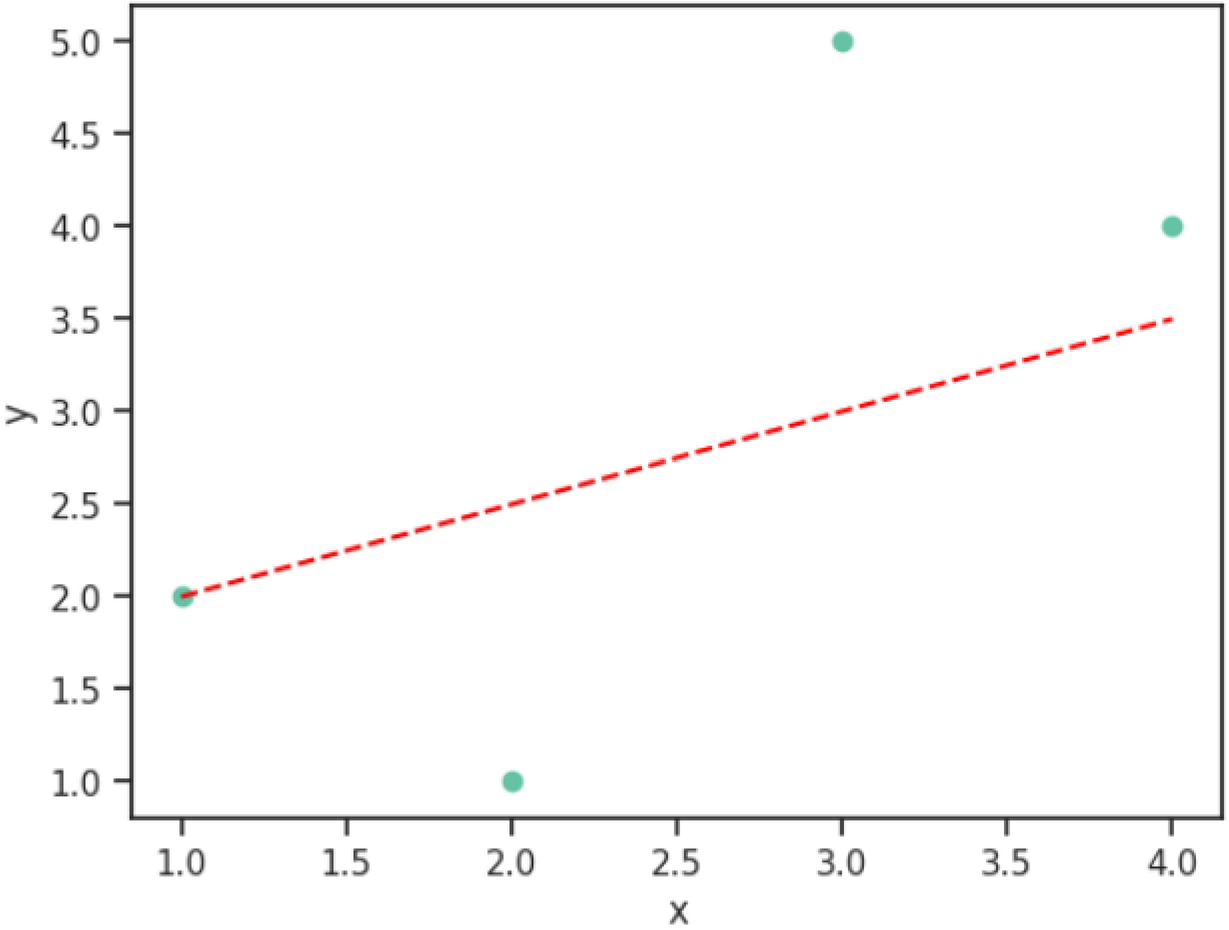
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

y_hat = mx + b = w_1x + w_0

w_1=.5
w_0=1.5

y_hat = [2, 2.5, 3, 3.5]

MSE = 1.625



Recap

Last class

Linear/polynomial regression

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

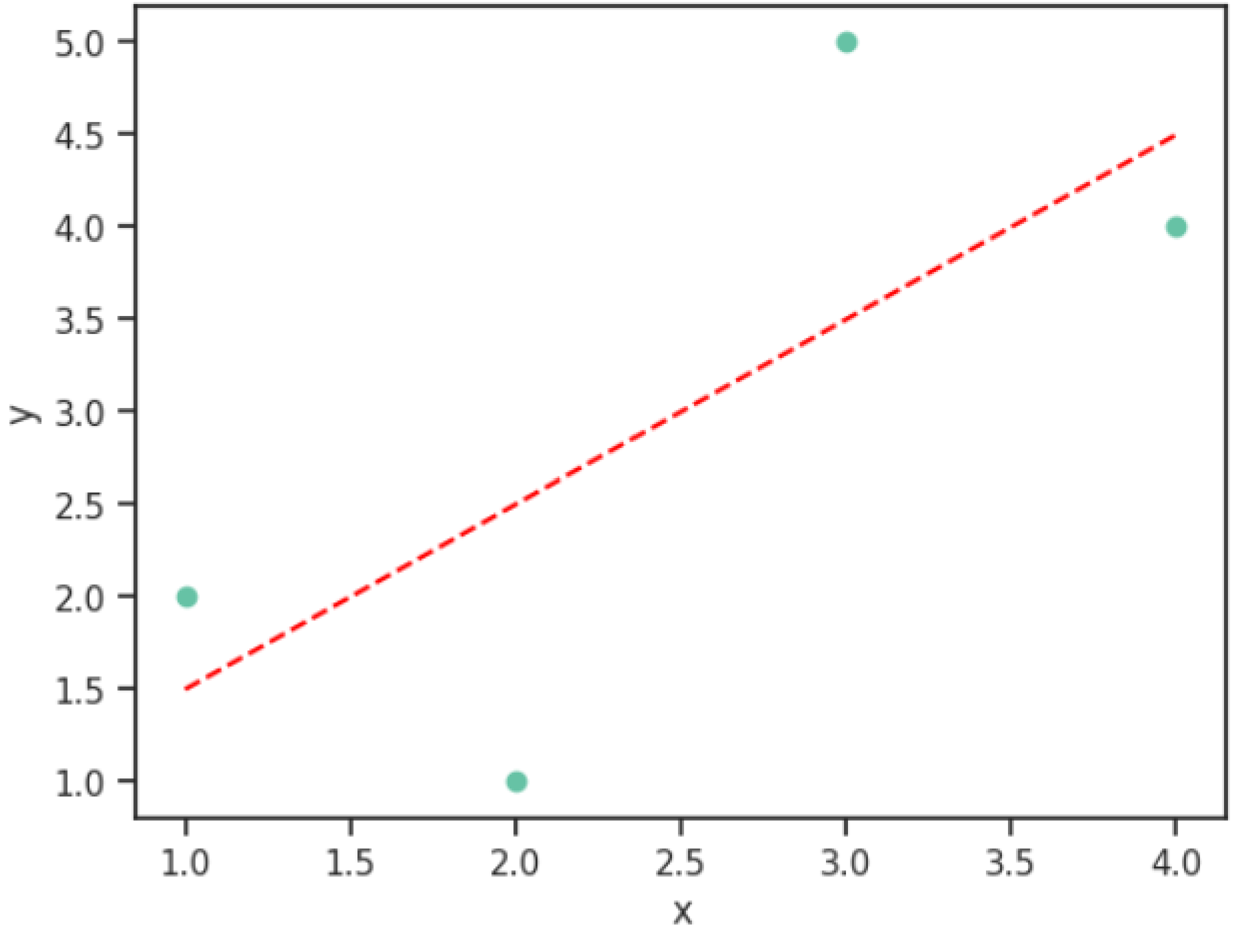
y_hat = mx + b = w_1x + w_0

w_1=1

w_0=.5

```
y_hat = [1.5, 2.5, 3.5, 4.5]
```

```
MSE = 1.25
```



Recap

Last class

Linear/polynomial regression

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

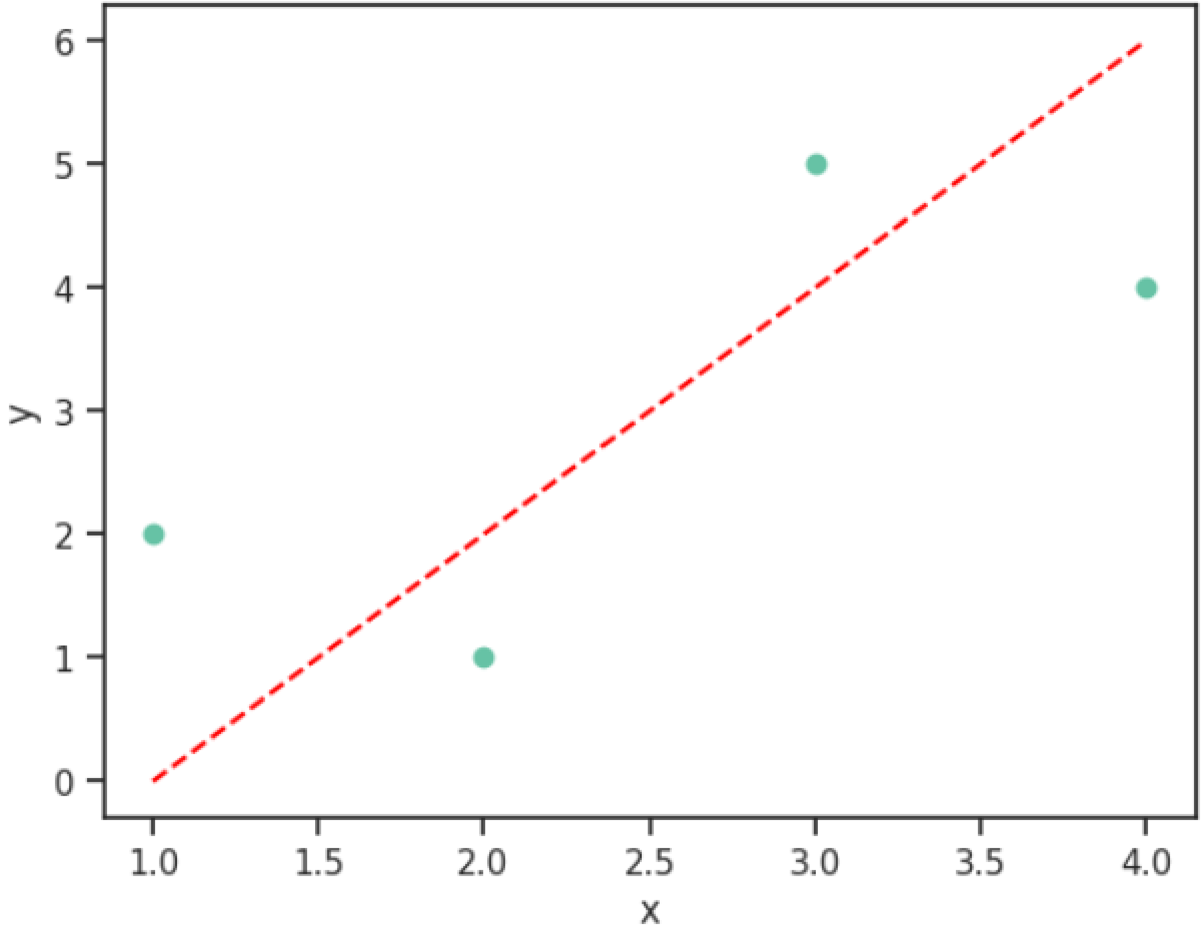
y_hat = mx + b = w_1x + w_0

w_1=2

w_0= -2

```
y_hat = [0, 2, 4, 6]
```

```
MSE = 2.5
```



Recap
Last class

Linear/polynomial regression

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

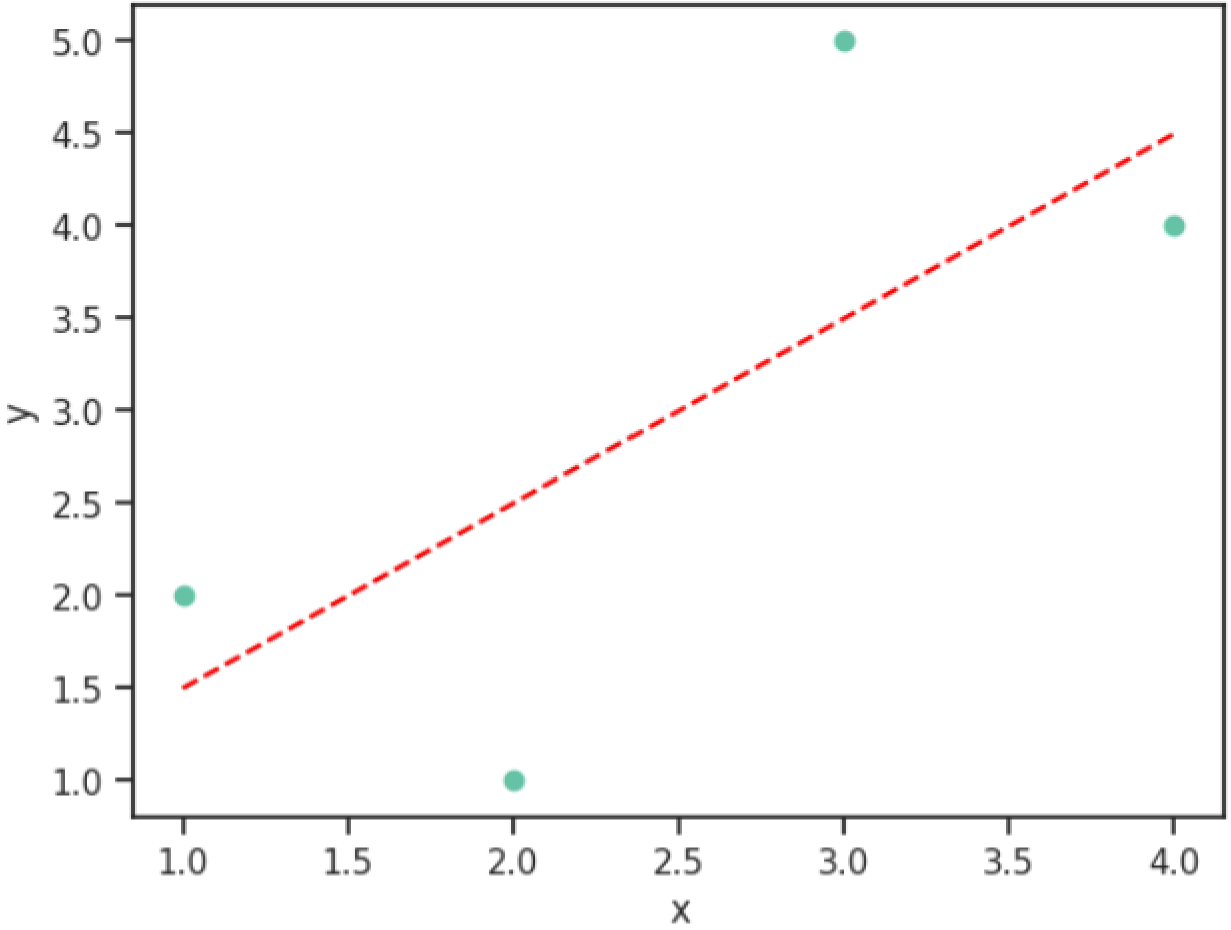
y_hat = mx + b = w_1x + w_0

w_1=.5
w_0=1.5

This one's the best!

```
y_hat = [1.5, 2.5, 3.5, 4.5]
```

```
MSE = 1.25
```



Recap

Last class

Linear/polynomial regression

If we have more than 1 feature (x), our line equation looks like:

y_hat = $w_{10}x_{10}$ + … + w_2x_2 +w_1x_1 + w_0

Building Your Toolbox

Last class

Question!

- We actually do polynomial regression with a linear regression. **How do we do this? (Hint: it involves manipulating the features themselves)**

Building Your Toolbox

Last class

Question!

- We actually do polynomial regression with a linear regression. **How do we do this? (Hint: it involves manipulating the features themselves)**

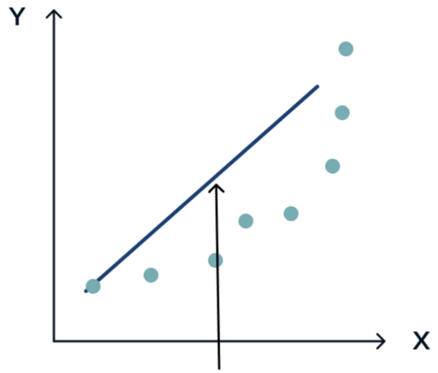
$\text{y\_hat} = w_2x^2 + w_1x + w_0$

or $\text{y\_hat} = w_3x^3 + w_2x^2 + w_1x + w_0$

or $\text{y\_hat} = w_4x_2^2 + w_3x_2 + w_2x_1^2 + w_1x_1 + w_0$

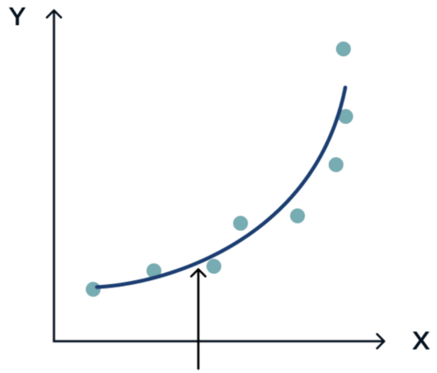
We construct **polynomial/non-linear features!**

Simple linear model



$y=b_0+b_1x$

Polynomial model

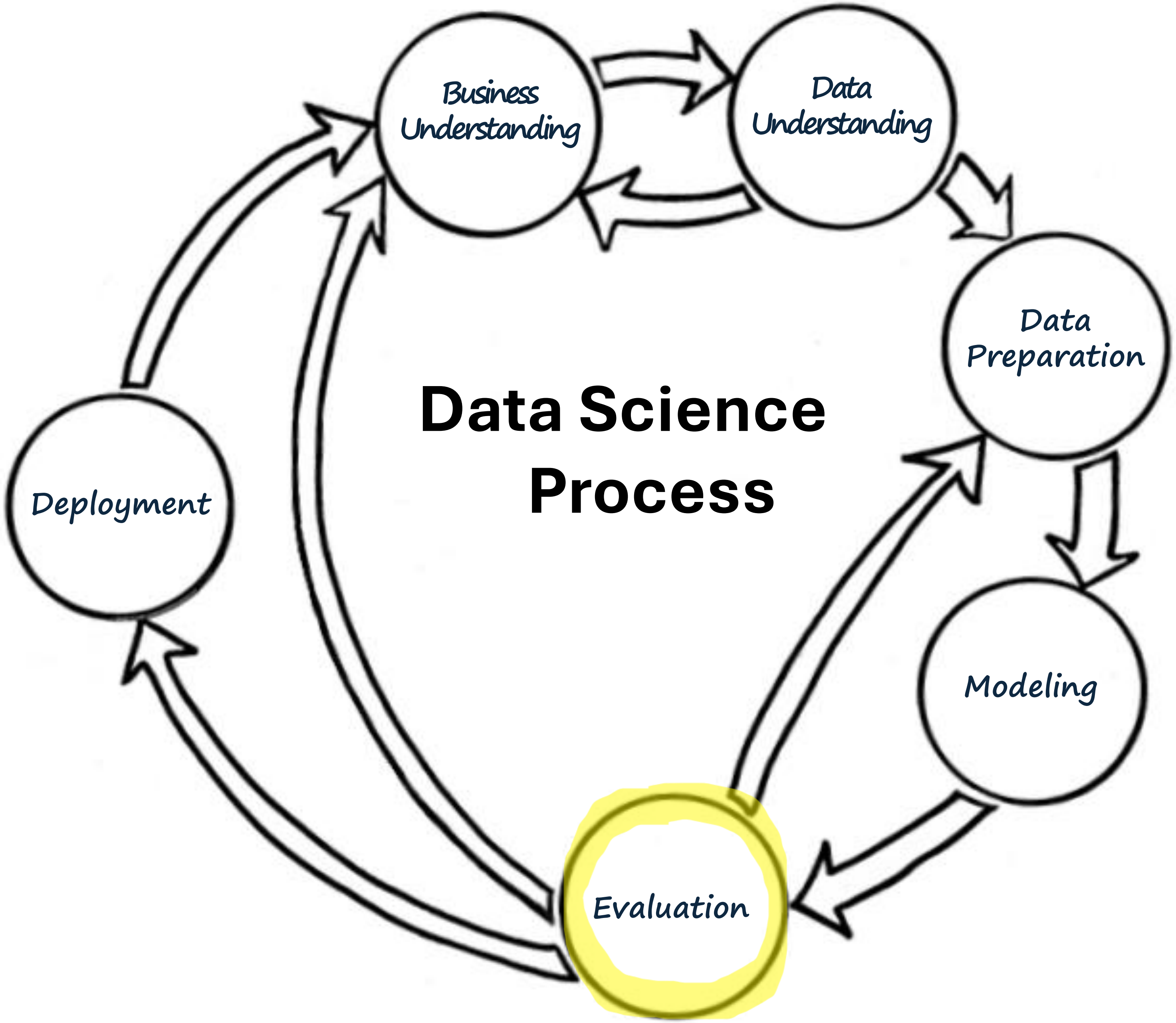


$y=b_0+b_1x+b_2x_1^2$

Building Your Toolbox
Last class

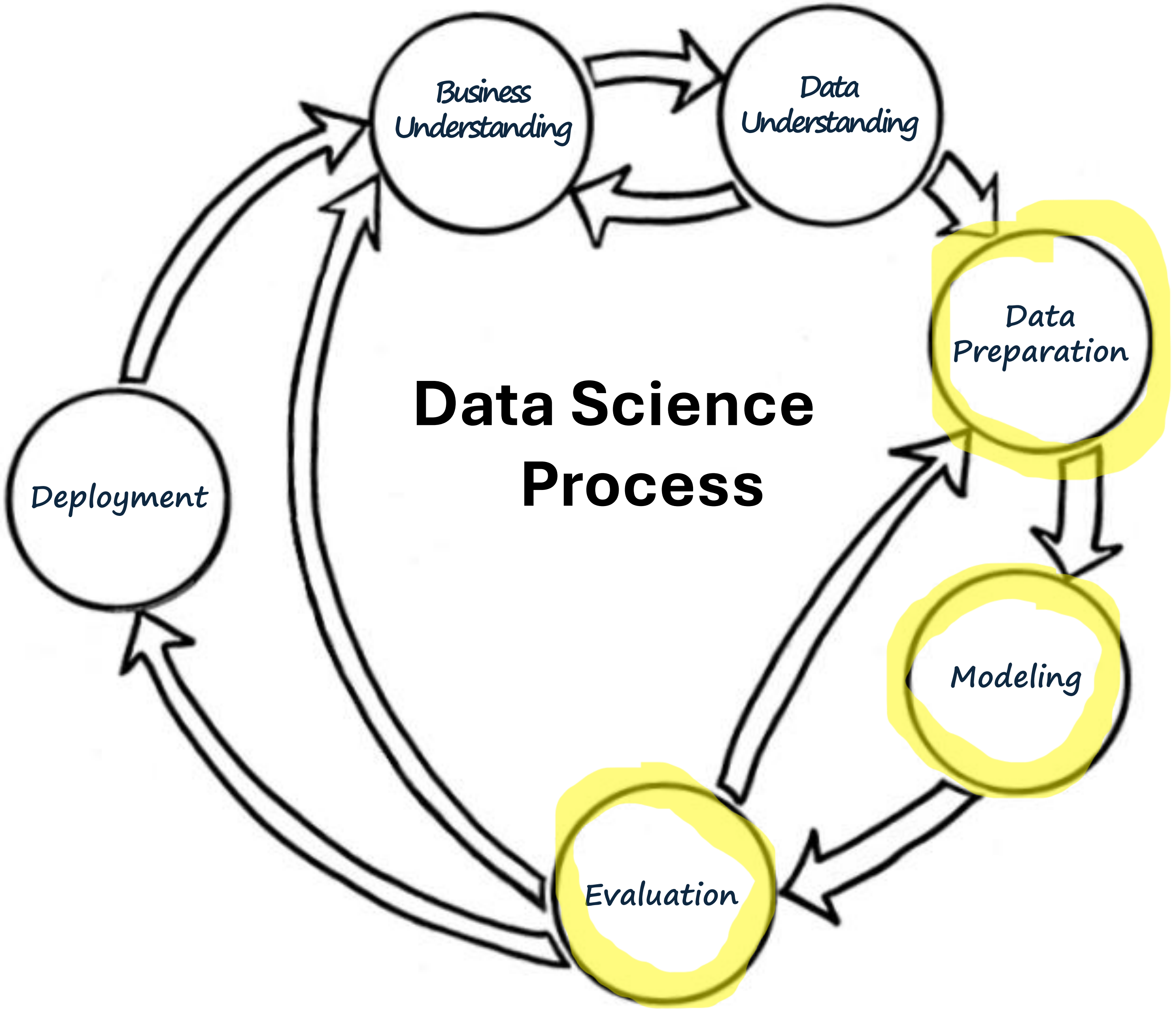
Evaluation Techniques/Considerations

- **Overfitting** (the training data) → **worse generalization** (on unseen data)
- **K-fold cross-validation** is a way to evaluate generalization



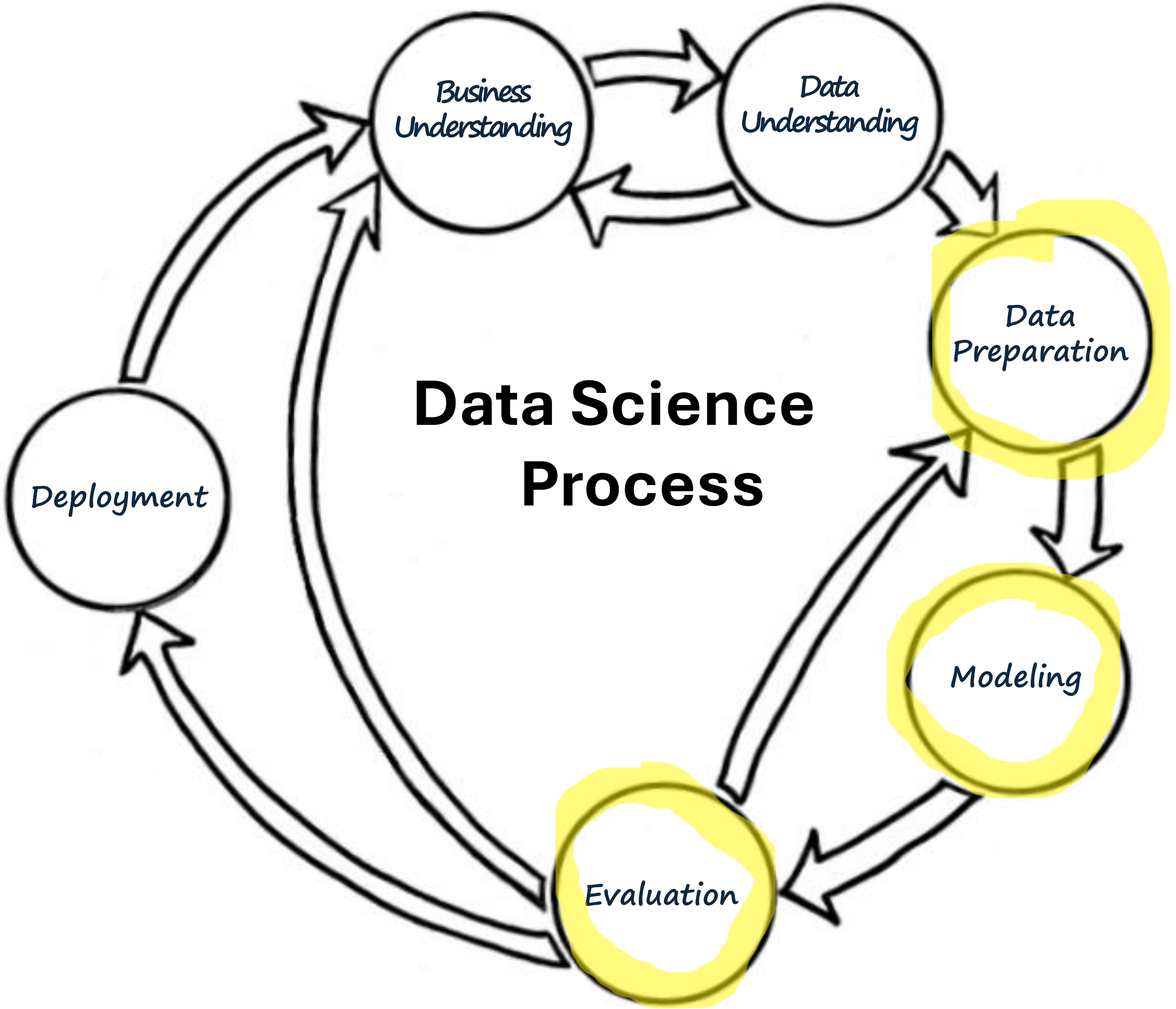
Building Your Toolbox

Today



Building Your Toolbox
Today

Show more ways to prevent **overfitting = ensure generalization**



Building Your Toolbox

Today

- How to avoid overfitting? → **Regularization**
  - **L1, L2**
- Data preparation to perform regularization well
  - **Normalization**
  - **Standardization**
- How to avoid overfitting? → **Hyper-parameter tuning**
- How to assess degree of hyper-parameter tuning? → **Cross validation on training set**
- How to assess amount of data on generalization performance? → **Learning curves**

Notebook time!