January 2025

Module 5 – Model Assessment

Data Science For Business

Quiz time!

Quiz discussion!

Q1

What is the primary difference between L1 and L2 regularization?

- L1 regularization penalizes the sum of weights, L2 penalizes the sum of squared weights.
- L2 regularization penalizes the sum of weights, L1 penalizes the sum of squared weights.
- L1 regularization penalizes the sum of absolute value of the weights, L2 penalizes the sum of squared weights.
- L2 regularization penalizes the sum of absolute value of the weights, L1 penalizes the sum of squared weights.

Q1

What is the primary difference between L1 and L2 regularization?

- L1 regularization penalizes the sum of weights, L2 penalizes the sum of squared weights.
- L2 regularization penalizes the sum of weights, L1 penalizes the sum of squared weights.
- L1 regularization penalizes the sum of absolute value of the weights, L2 penalizes the sum of squared weights.
- L2 regularization penalizes the sum of absolute value of the weights, L1 penalizes the sum of squared weights.

Q2

What is the purpose of regularization in logistic regression?

- To increase the complexity of the model.
- To ensure the model fits the training data perfectly.
- To penalize complex models and reduce overfitting.
- To avoid using nonlinear features.

Q2

What is the purpose of regularization in logistic regression?

- To increase the complexity of the model.
- To ensure the model fits the training data perfectly.
- To penalize complex models and reduce overfitting.
- To avoid using nonlinear features.

Q3

In linear/polynomial regression, we can increase the degree of the model to increase the complexity. We can also increase complexity by

- increasing regularization on the model in the training process
- decreasing regularization on the model in the training process

Q3

In linear/polynomial regression, we can increase the degree of the model to increase the complexity. We can also increase complexity by

- increasing regularization on the model in the training process
- decreasing regularization on the model in the training process

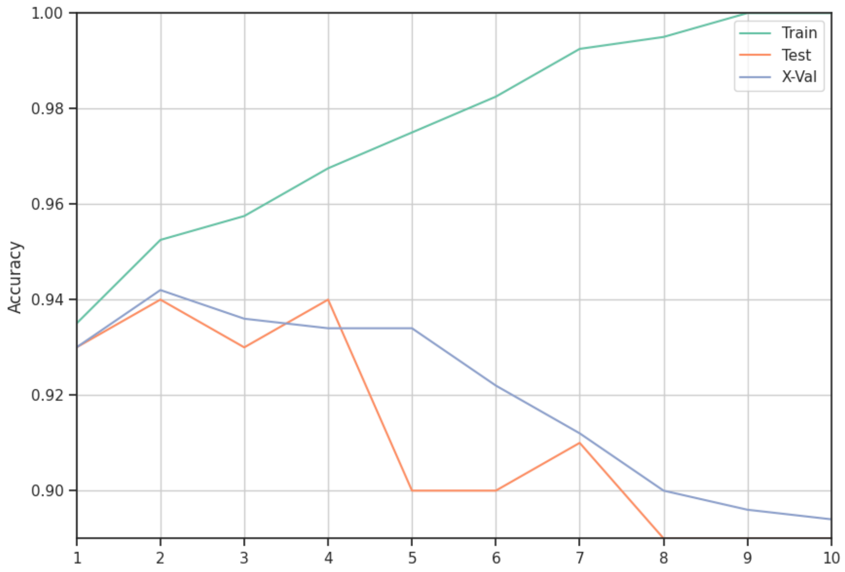
Q4

Describe the difference between a fitting curve and a learning curve. What is on the x and y axis for each? You can write this in bullet point format.

Q4

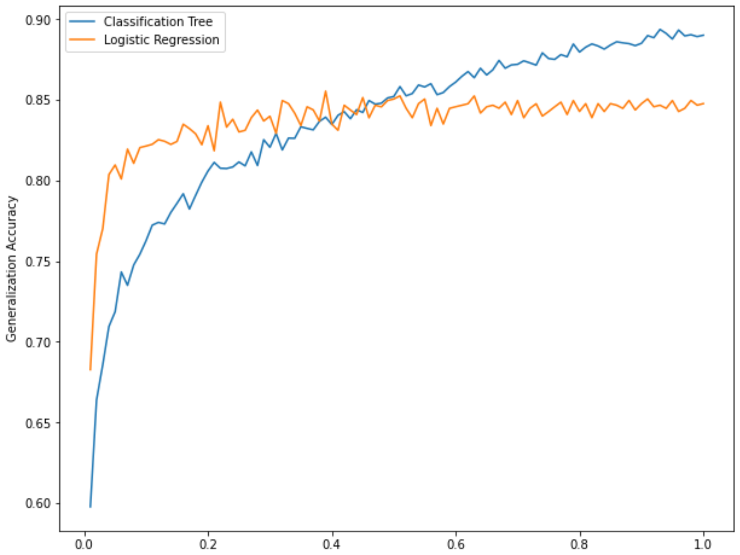
Describe the difference between a fitting curve and a learning curve. What is on the x and y axis for each? You can write this in bullet point format.

Fitting Curve



Model Complexity

Learning Curve



Percent of Data

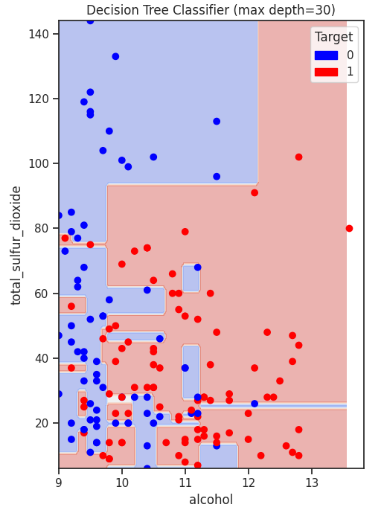
Q5

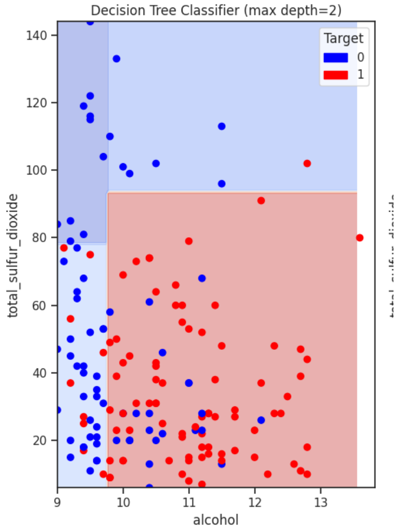
You are tasked with building a model to predict whether a new type of concrete mixture will achieve a compressive strength of 35 MPa or more. Your dataset contains multiple features, such as the quantities of cement, water, and aggregate, and you observe that some features have values ranging from 0 to 1, while others range from 0 to 1000. You have information on the target variable. Your boss wants you to create a decision tree to model concrete strength. Unfortunately, the standard regularization approach of penalizing weights cannot be applied to decision tress because they are not directly minimizing a loss function/maximizing an objective function. How might you ensure that this decision tree does not overfit the training data?

Q5

You are tasked with building a model to predict whether a new type of concrete mixture will achieve a compressive strength of 35 MPa or more. Your dataset contains multiple features, such as the quantities of cement, water, and aggregate, and you observe that some features have values ranging from 0 to 1, while others range from 0 to 1000. You have information on the target variable. Your boss wants you to create a decision tree to model concrete strength. Unfortunately, the standard regularization approach of penalizing weights cannot be applied to decision tress because they are not directly minimizing a loss function/maximizing an objective function. How might you ensure that this decision tree does not overfit the training data?

Limit depth of tree (potentially by enforcing a minimum number of samples per leaf)





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)**

Agenda

Core Toolkit

- **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)**

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)**

Core Toolkit

Other Models/ Applications

Grading notes

- Assignment 2 may not be graded until this weekend
- I’m trying to get a grader to speed this up!
- Otherwise, all grades should be in so far – please let me know if not!

Feedback

Thank you all for your feedback!

A summary of many of the main ideas:

- **A little more break time**
- **Live coding is good, but slow it down**
- **Polls**
- **Group work is helpful**

Group discussion!

MegaTelco

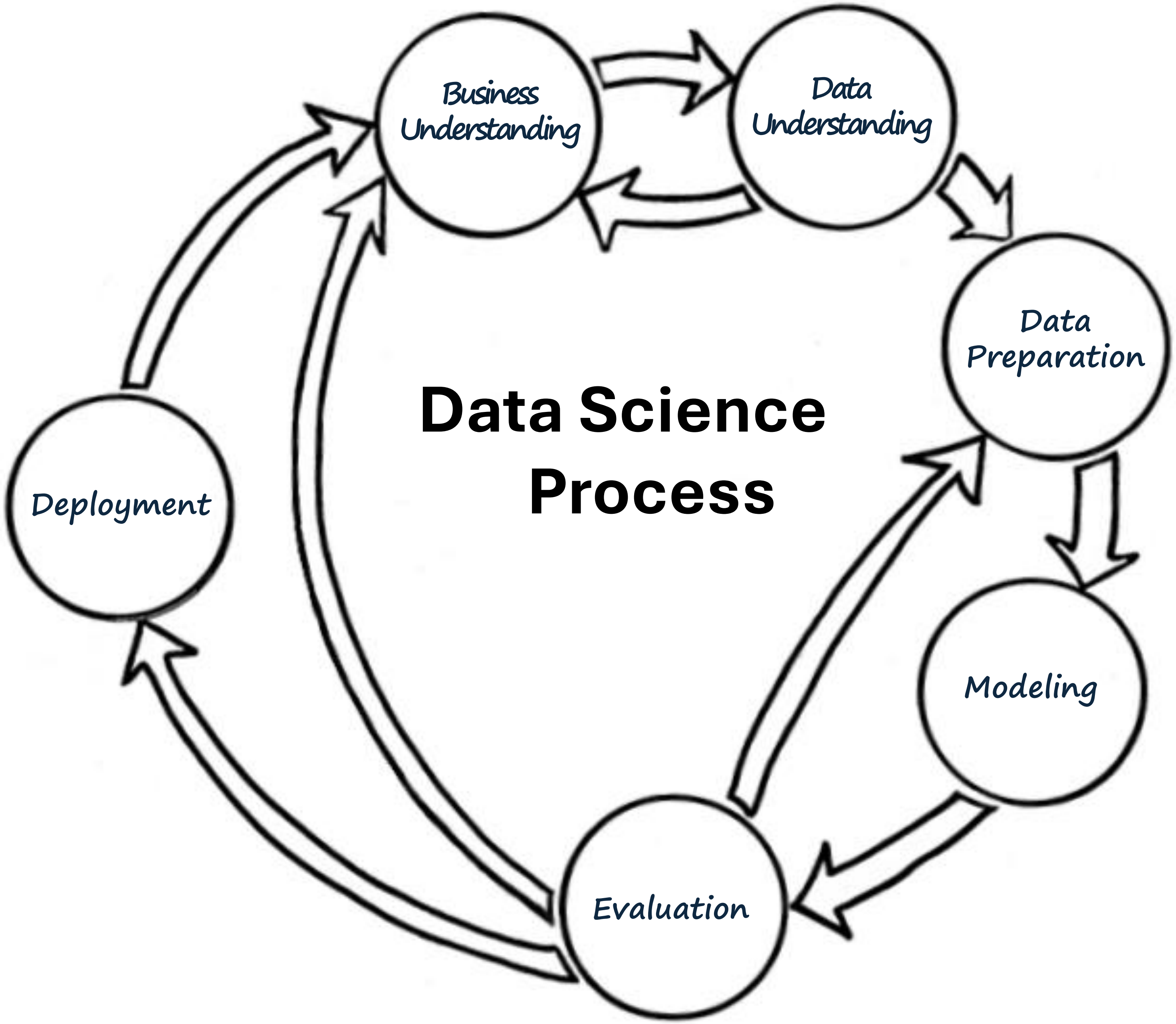
Henrietta, a Data Science Product Manager, has just joined MegaTelCo, one of the largest telecommunication firms. MegaTelco is having a major problem with churn in their wireless business. In the mid-Atlantic region, 20% of cell-phone customers leave when their phones are paid off, and it is getting increasingly difficult to acquire new customers. They call her in to help understand the problem and devise a solution. Marketing has designed a special retention offer.

Specifically, your task is to help Henrietta devise a precise, step-by-step plan for how the analyst/tech team should use MegaTelCo's vast data resource to decide which customers to target with the special retention offer prior to them paying off their phones.

Be specific as to what data to use and how to use them, and specifically how the team should decide on the set of customers to target to best reduce churn for a particular incentive budget. Use your better judgment as to what data MegaTelco would have.

So: Where should we start?

Where we are
So far



Where we are
So far

Clearly **define the problem**:

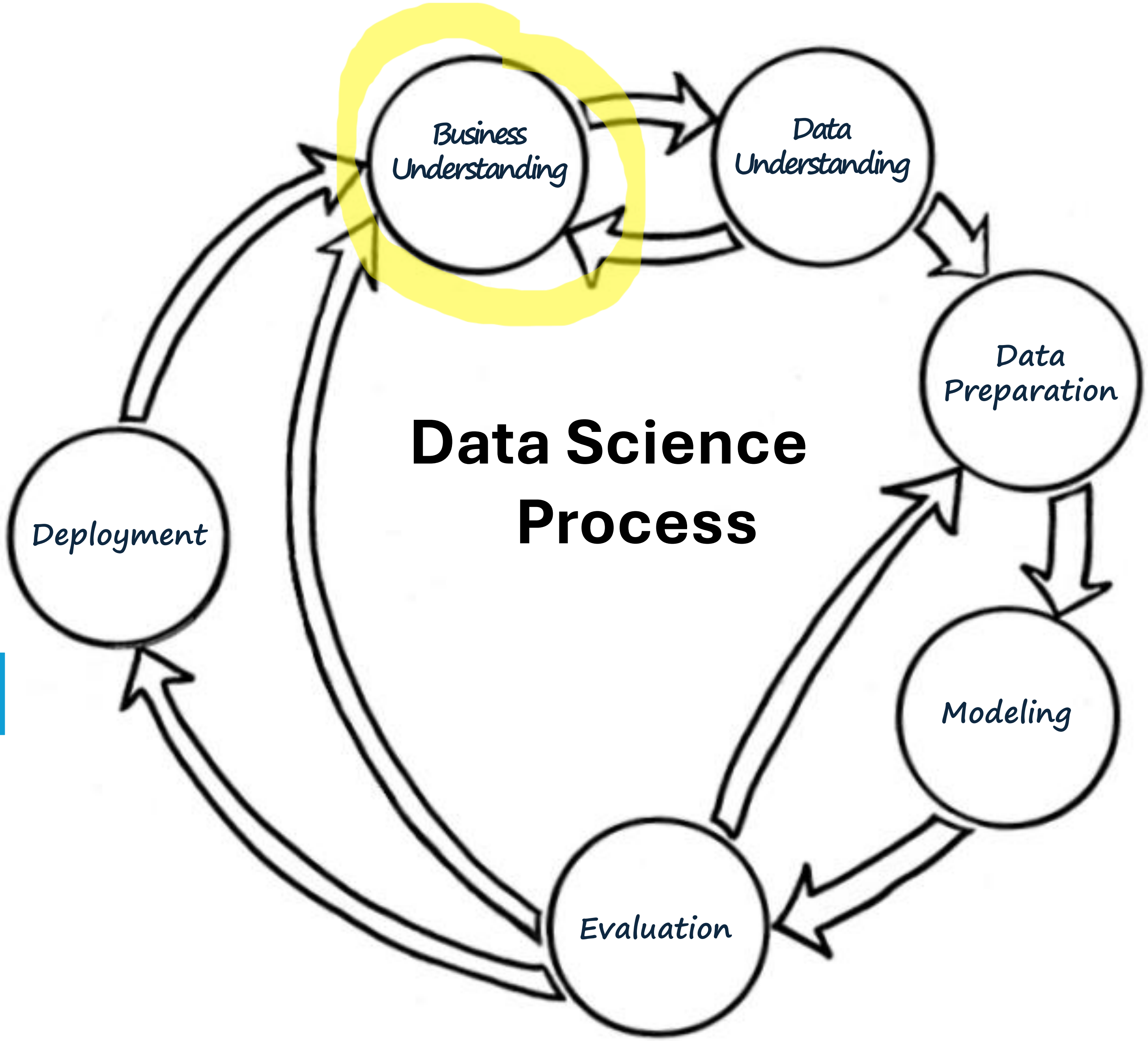
We want to identify customers who are likely to leave

Key DS problem types (so far):

Supervised Learning

Classification / Probability Estimation

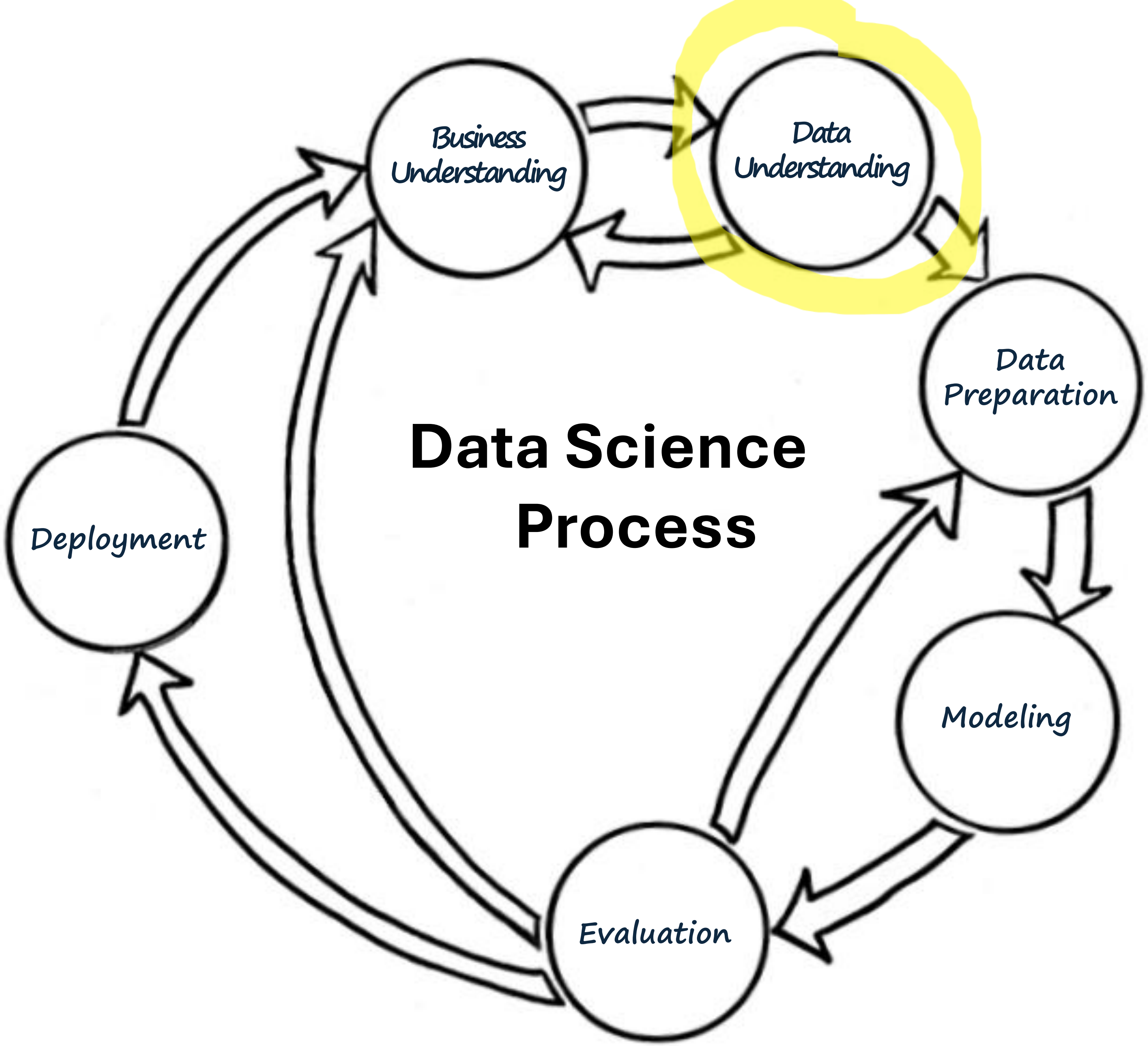
Regression



Where we are
So far

What **data do we have** – is it actually informative of the thing we care about (churn)?

Even if we don't know, we can EVALUATE and see!



Where we are
So far

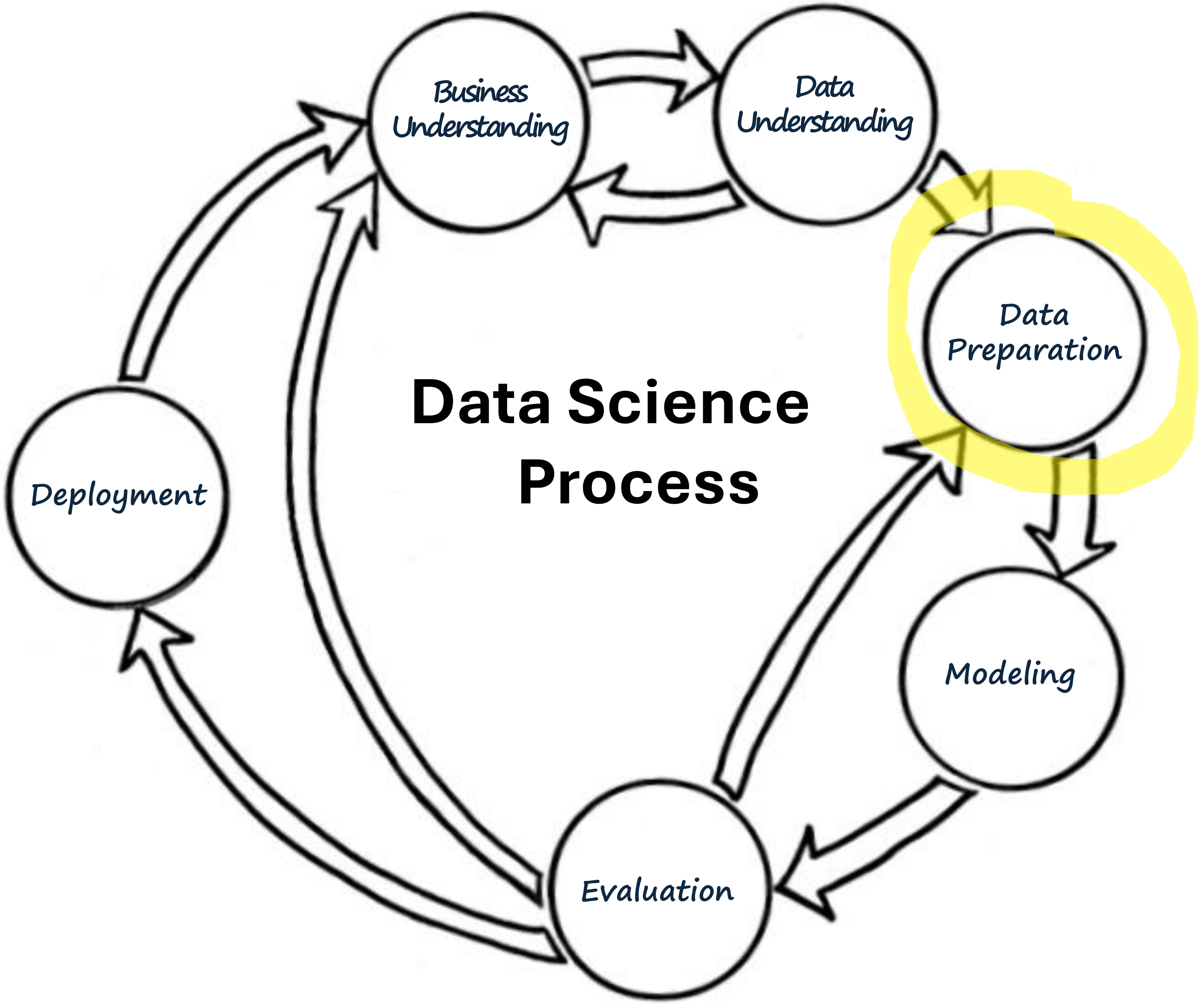
Convert **binary variables** into **1/0**

Normalize/standardize data (generally a good idea, necessary if we're doing regularization)

We need **target** and **features**

Instances in **training data** need to be the same **as instances we'll use at inference time**

Break data up into **train and test sets** (for evaluation later!)



Where we are
So far

Convert **binary variables** into **1/0**

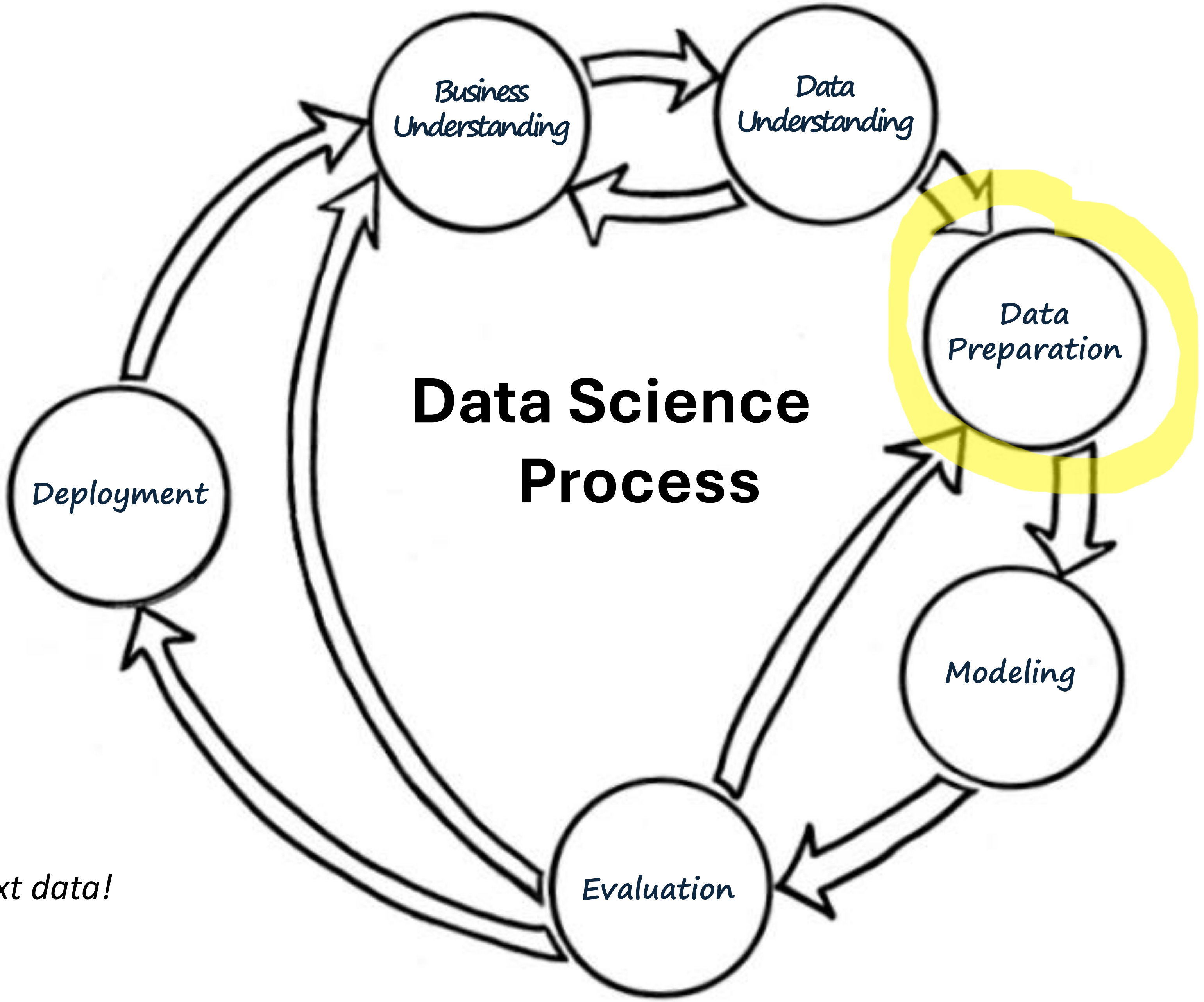
Normalize/standardize data (generally a good idea, necessary if we're doing regularization)

We need **target** and **features**

Instances in **training data** need to be the same **as instances we'll use at inference time**

Break data up into **train and test sets** (for evaluation later!)

Tomorrow: we'll talk about if the data is text data!



Where we are
So far

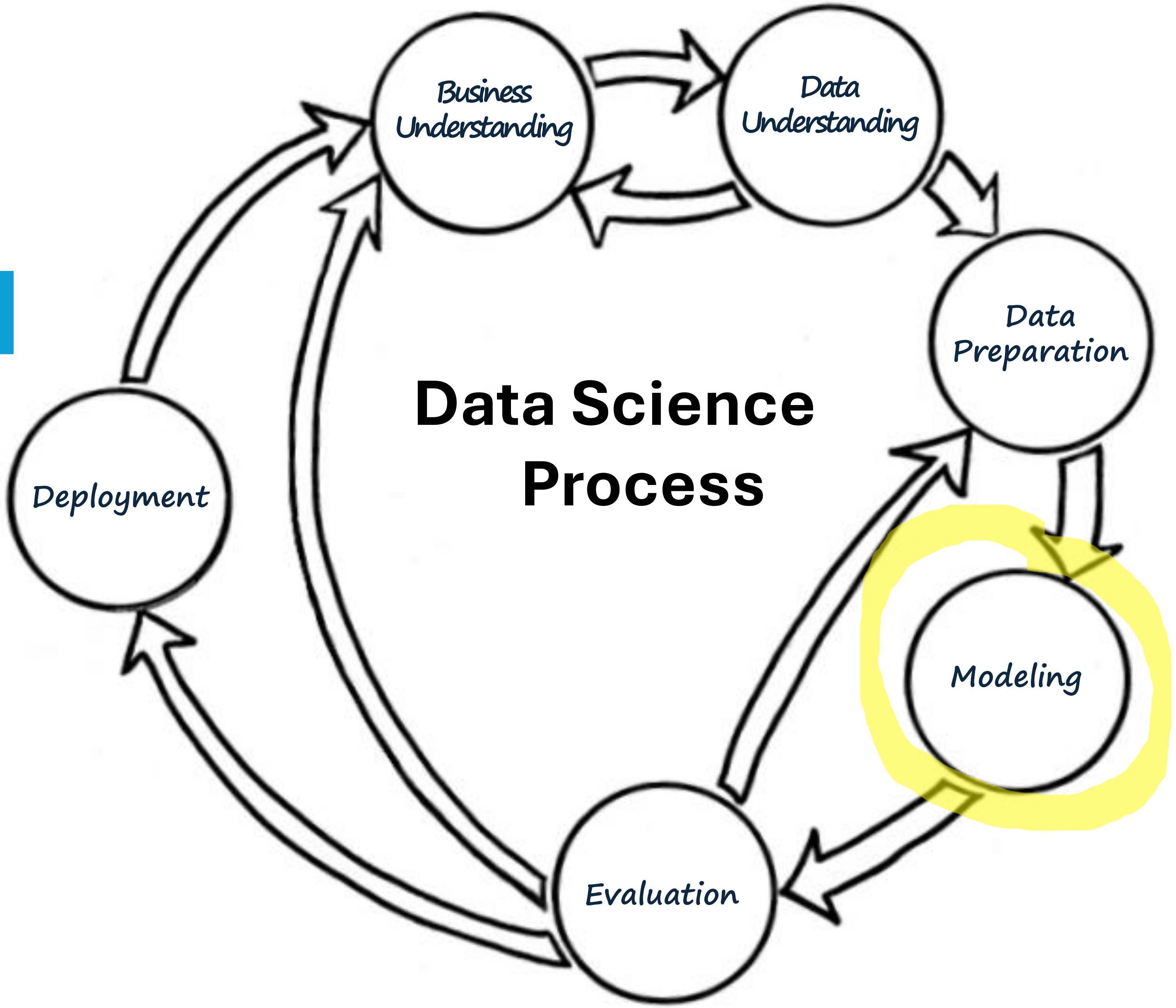
Supervised Learning

Classification / Probability Estimation

- Decision tree
- Linear/ polynomial logistic regression

Regression

- Linear/polynomial regression
- Regression tree



Where we are
So far

Supervised Learning

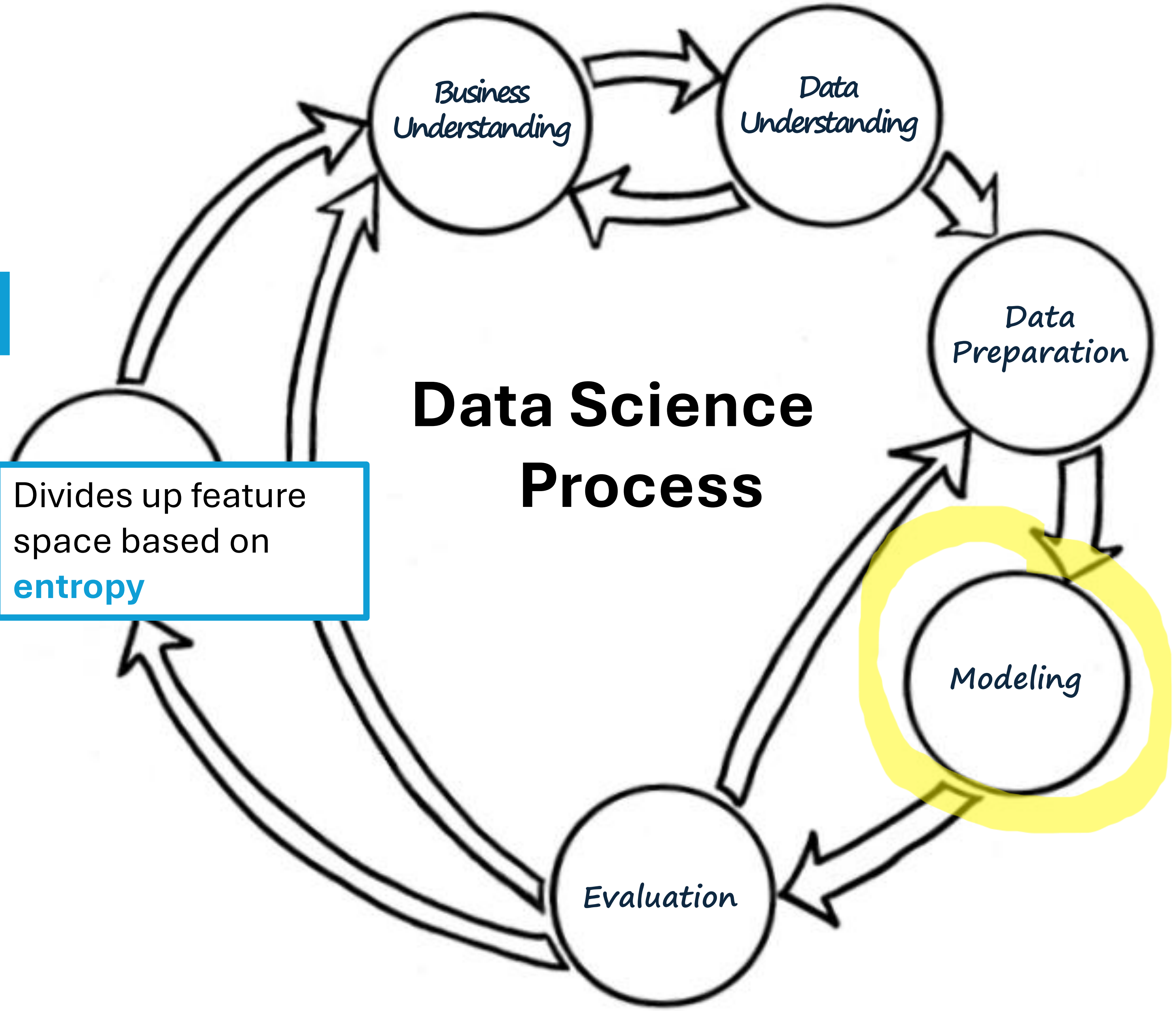
Classification / Probability Estimation

- Decision tree
- Linear/ polynomial logistic regression

Divides up feature space based on **entropy**

Regression

- Linear/polynomial regression
- Regression tree



Where we are
So far

Supervised Learning

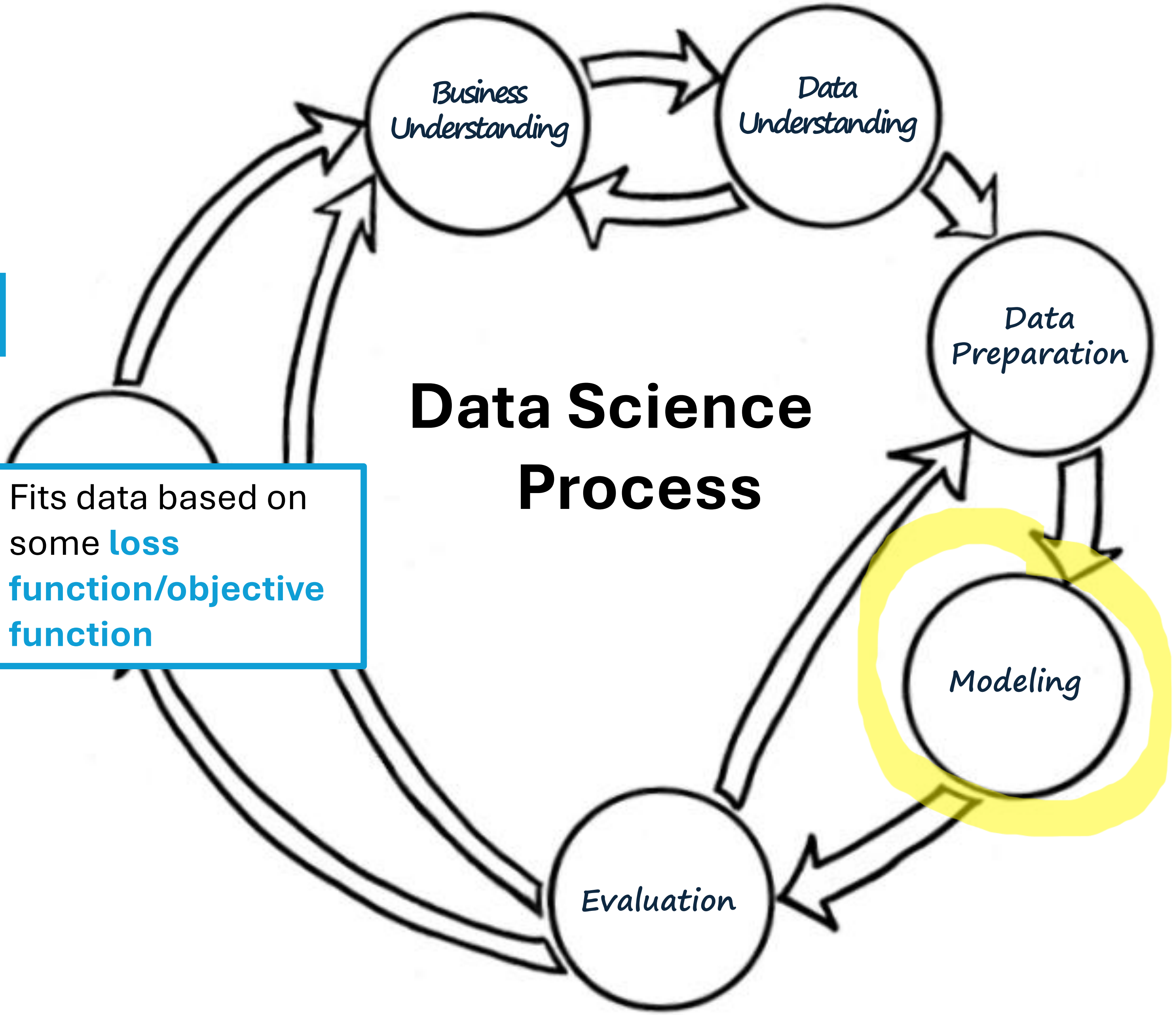
Classification / Probability Estimation

- Decision tree
- Linear/ polynomial logistic regression

Regression

- Linear/polynomial regression
- Regression tree

Fits data based on some **loss function/objective function**



Where we are
So far

Supervised Learning

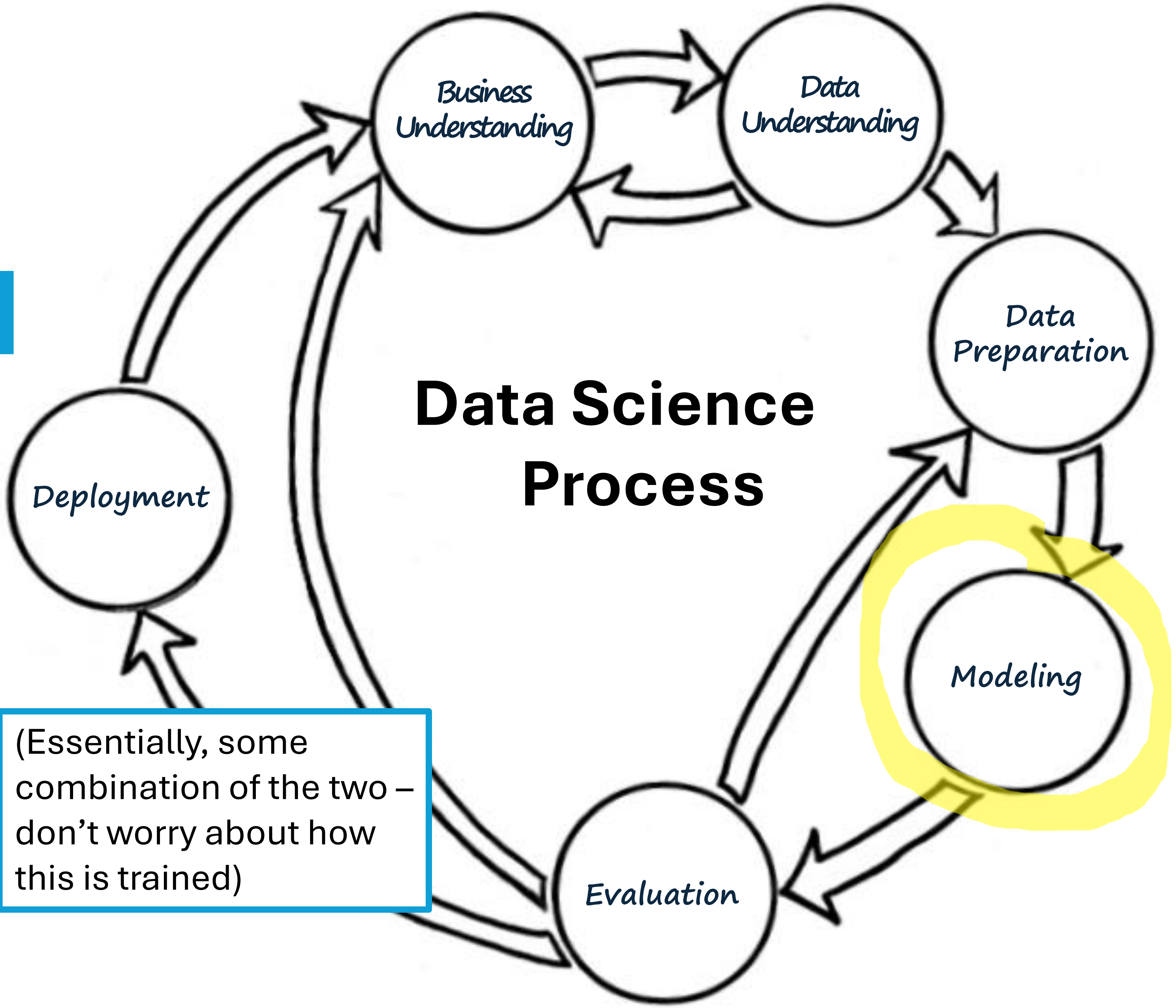
Classification / Probability Estimation

- Decision tree
- Linear/ polynomial logistic regression

Regression

- Linear/polynomial regression
- Regression tree

(Essentially, some combination of the two – don't worry about how this is trained)



Where we are
So far

How should we ensure we don't **overfit** the training data (i.e. ensure **generalizability**)?

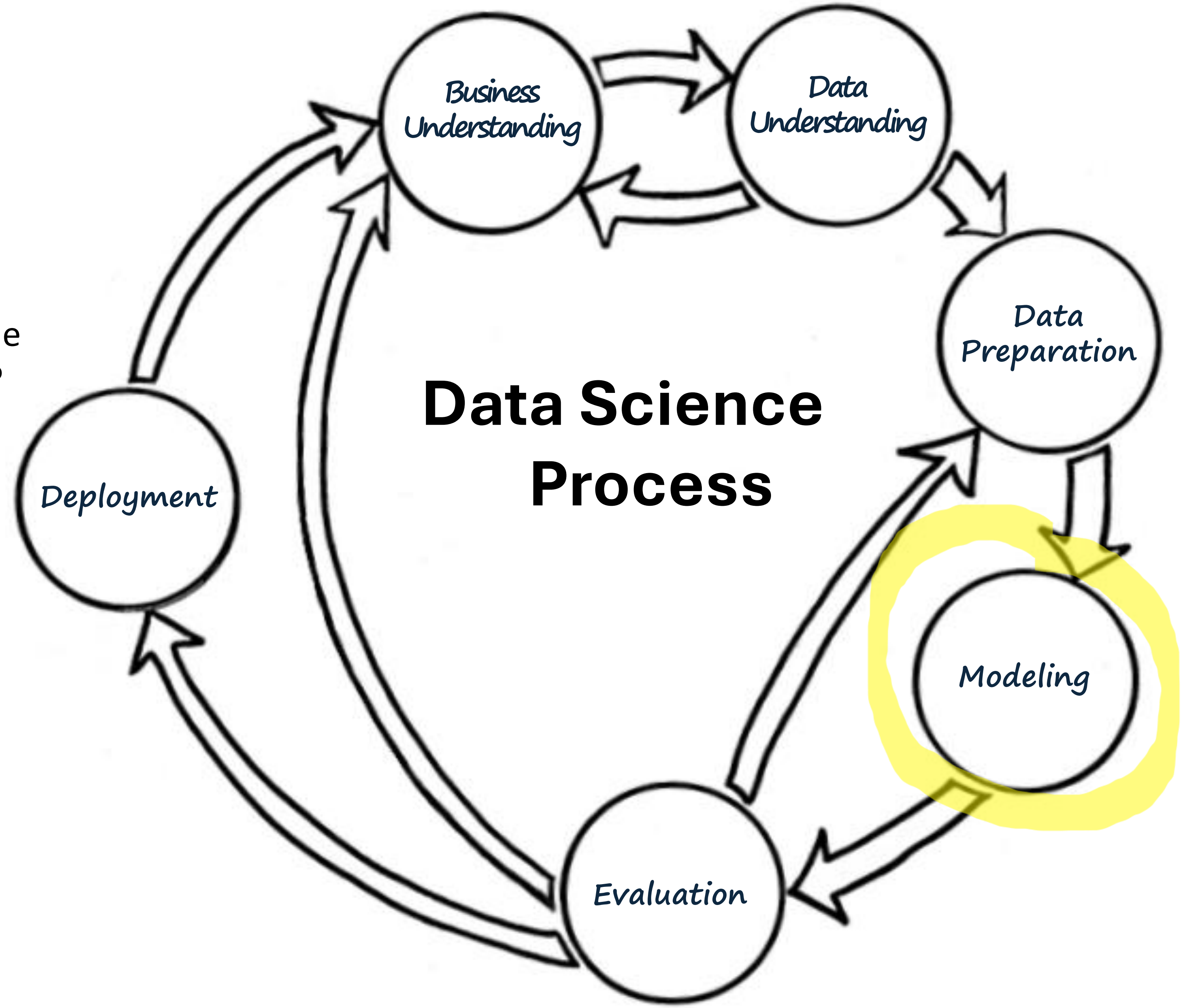
Limit tree size?

Regularize (L1, L2)?

→ How strong should the **regularization penalty (λ or C = 1/ λ)** be?

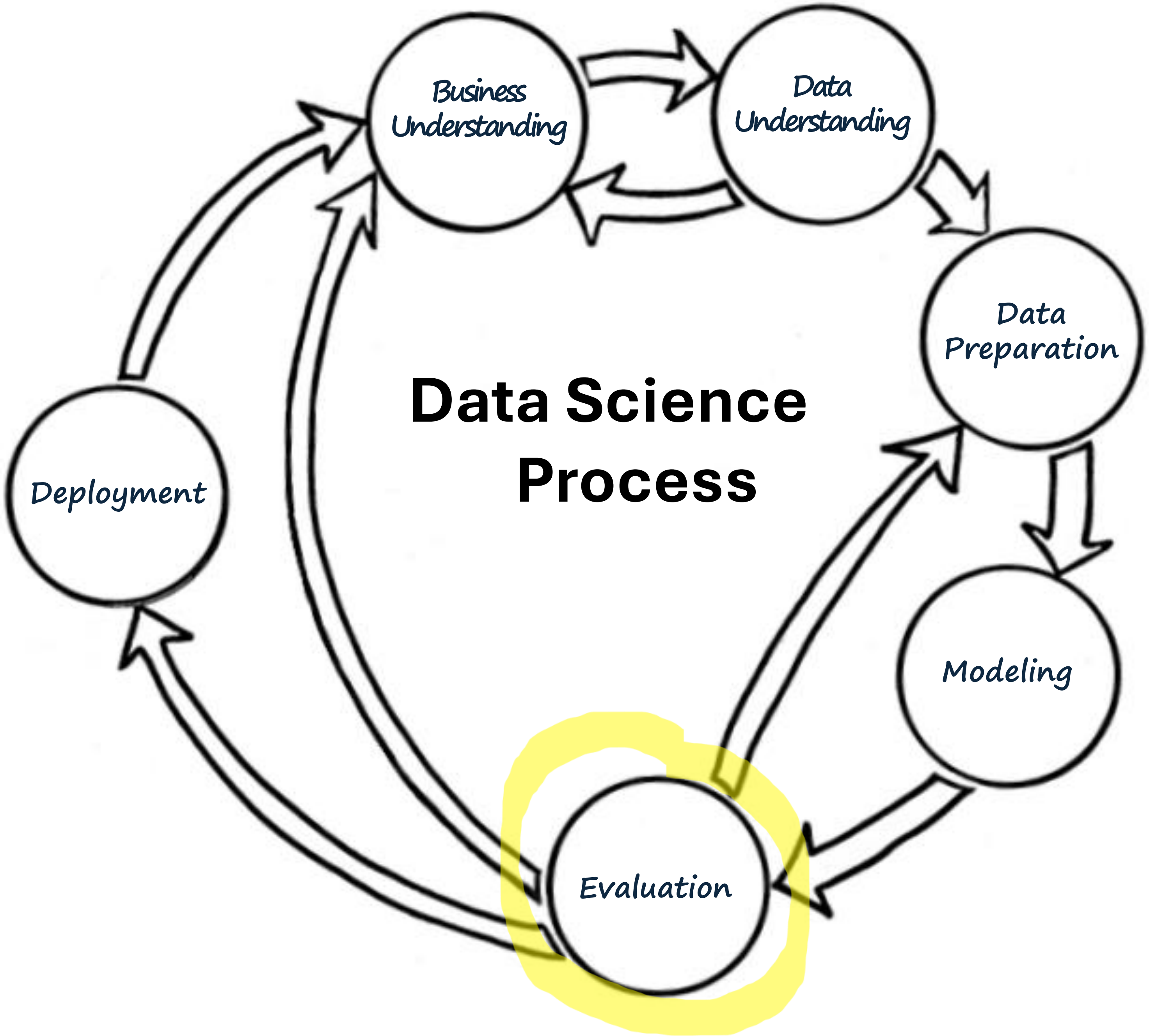
For classification problems we can:

- Visualize the **decision surface**
- Use the model to come up with **class probability estimates** instead of just class predictions



Where we are
So far

Spent a LOT of time here – more today!



Where we are
So far

Overfit/ how to find model complexity?

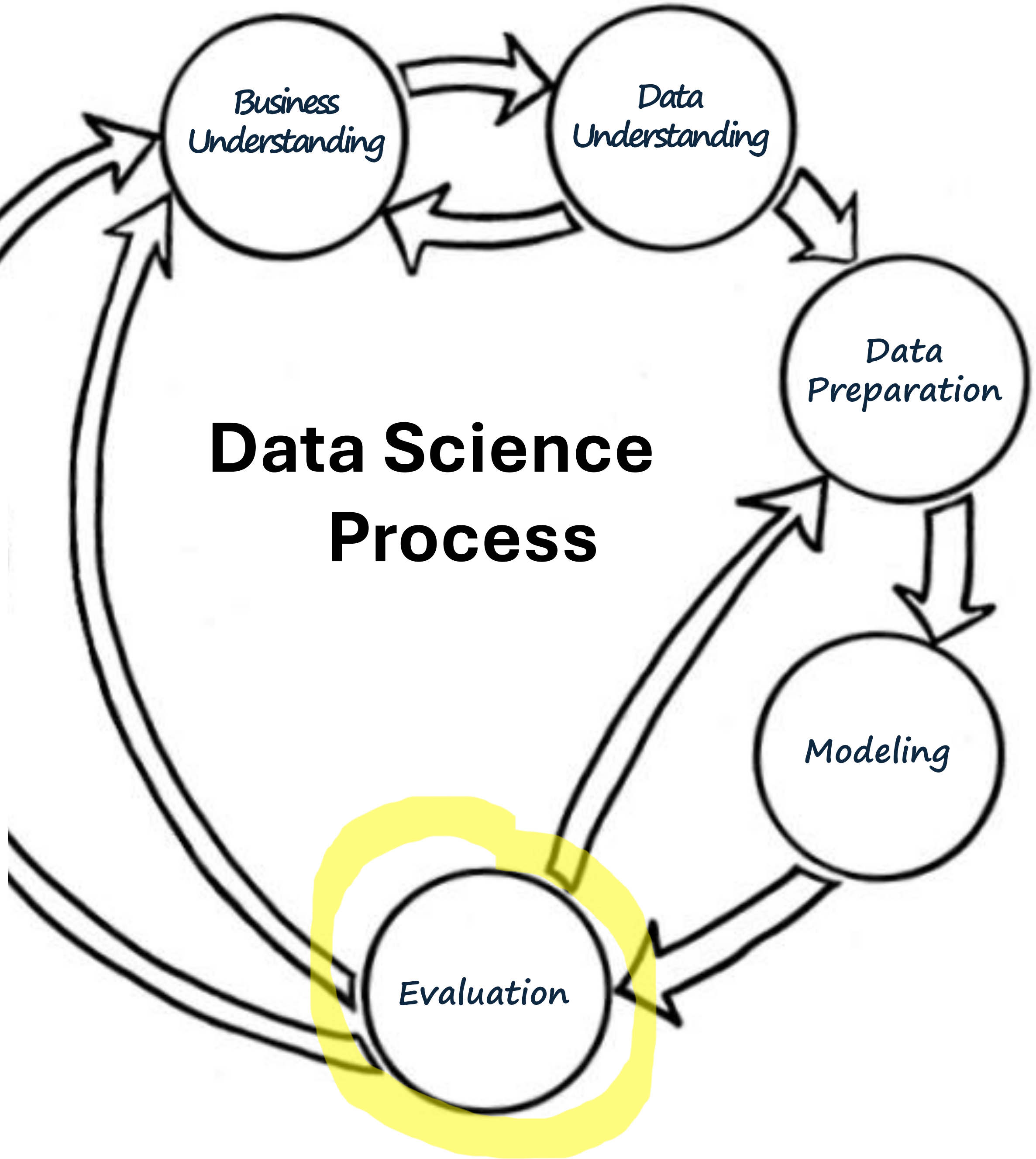
Fitting curves

How much data to use/should we get more?

Learning curves

K-fold cross validation – we can use this on our **training set** to try a bunch of different **hyper parameters:**

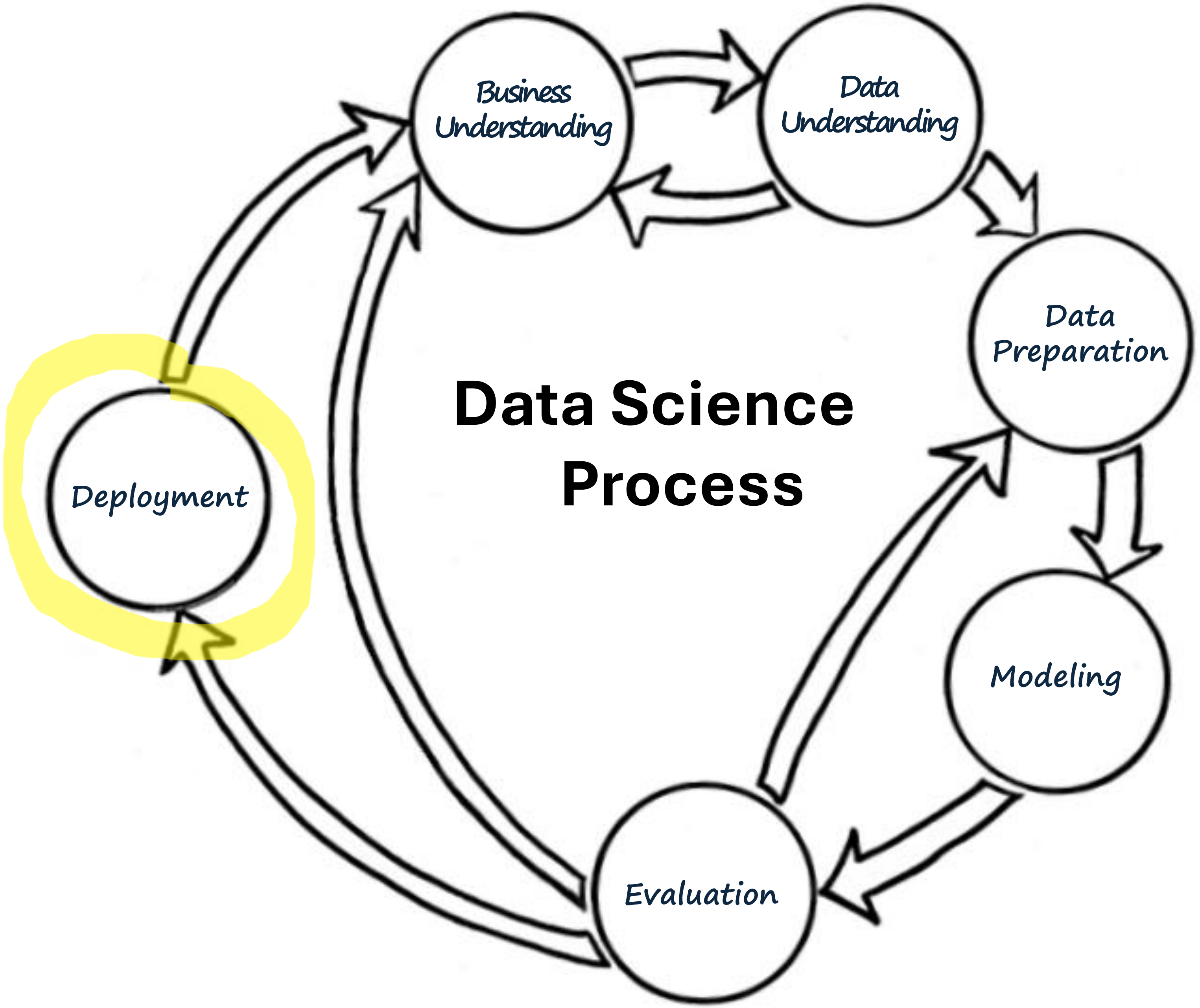
- Models
- Regularization amounts (λ) (for any model w/ a loss function, such as linear/logistic regression)
- Degrees or other constructed features (x^2, x^{10}, log x)
- Depth for decision trees
- And so on…



Where we are
So far

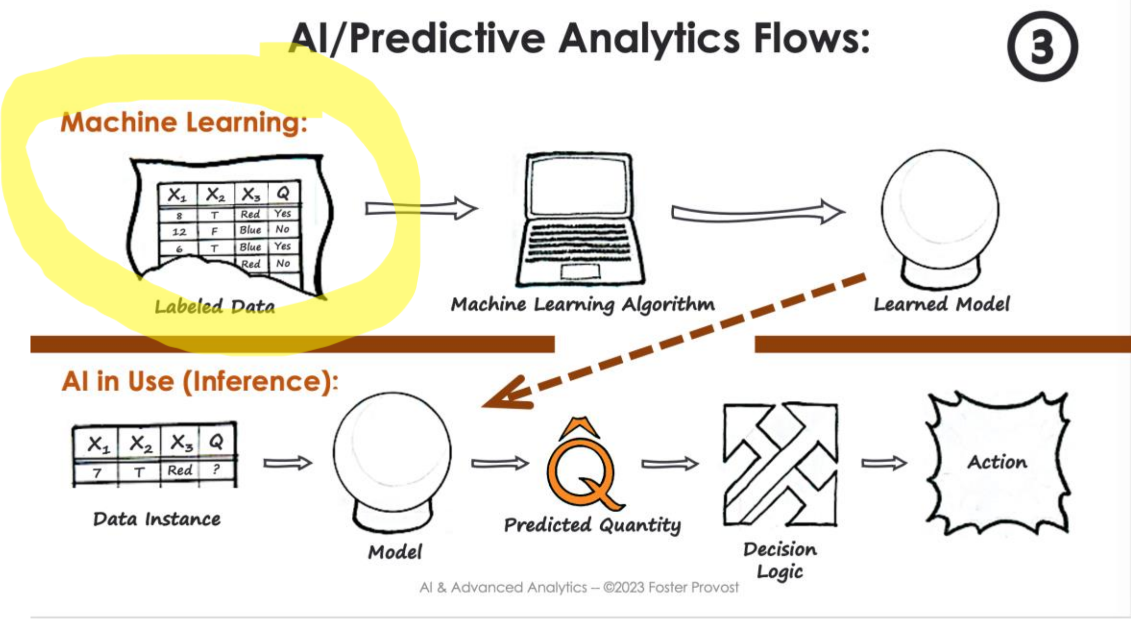
This will come up a bit today!

But a lot more for future classes (think machine learning *engineering*)



Where we are
So far

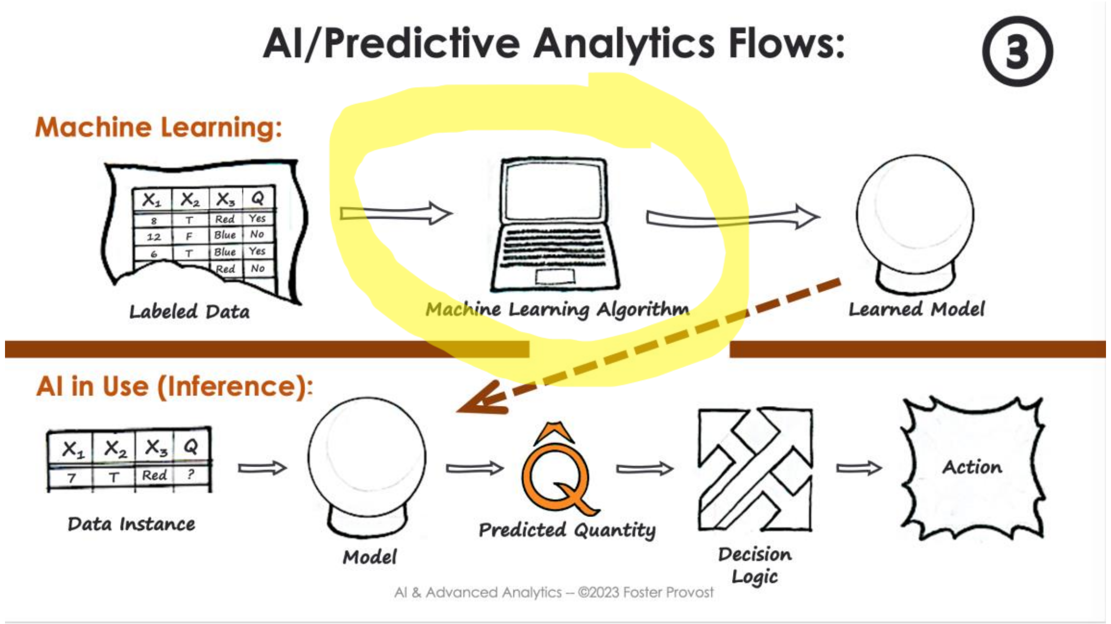
Again, make sure we have **targets** and **features** for all training instances



Where we are
So far

Train our model (sklearn.fit) to minimize **loss function** (or a regularized objective function)

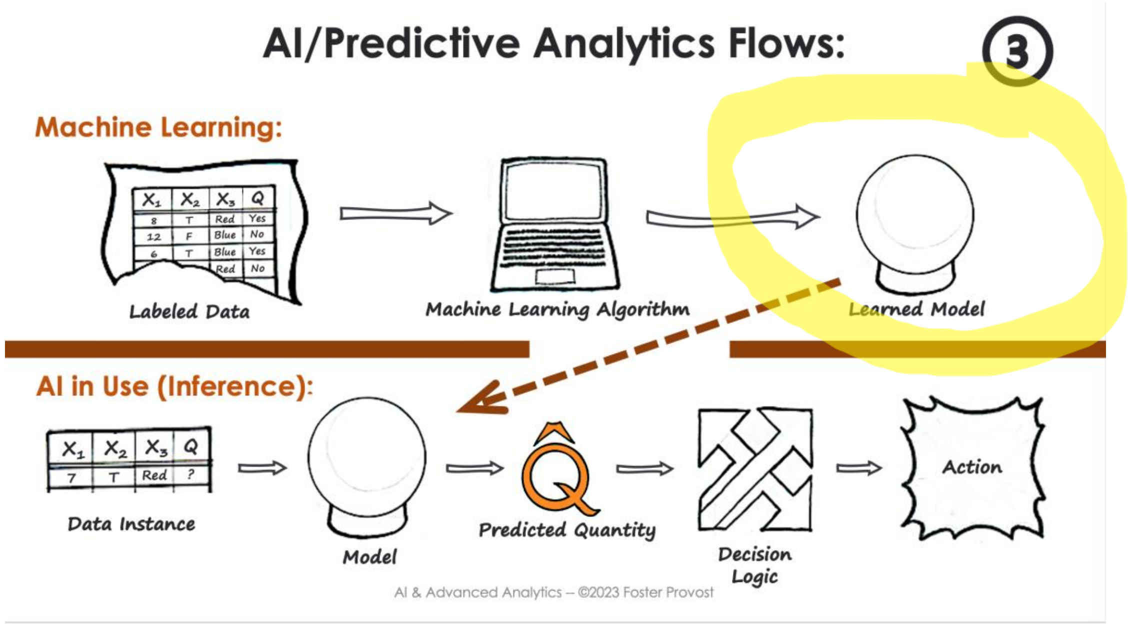
This is machine learning!



Where we are
So far

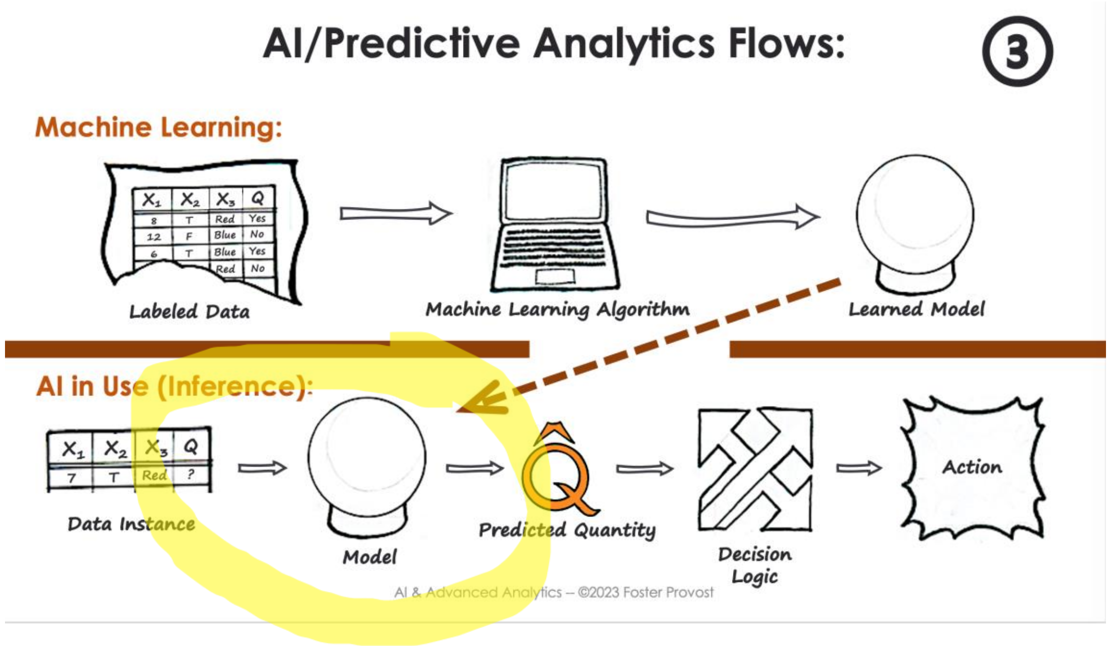
We end with some learned model

(a trained linear/logistic regression/decision tree/ regression tree)



Where we are
So far

model.predict(X)



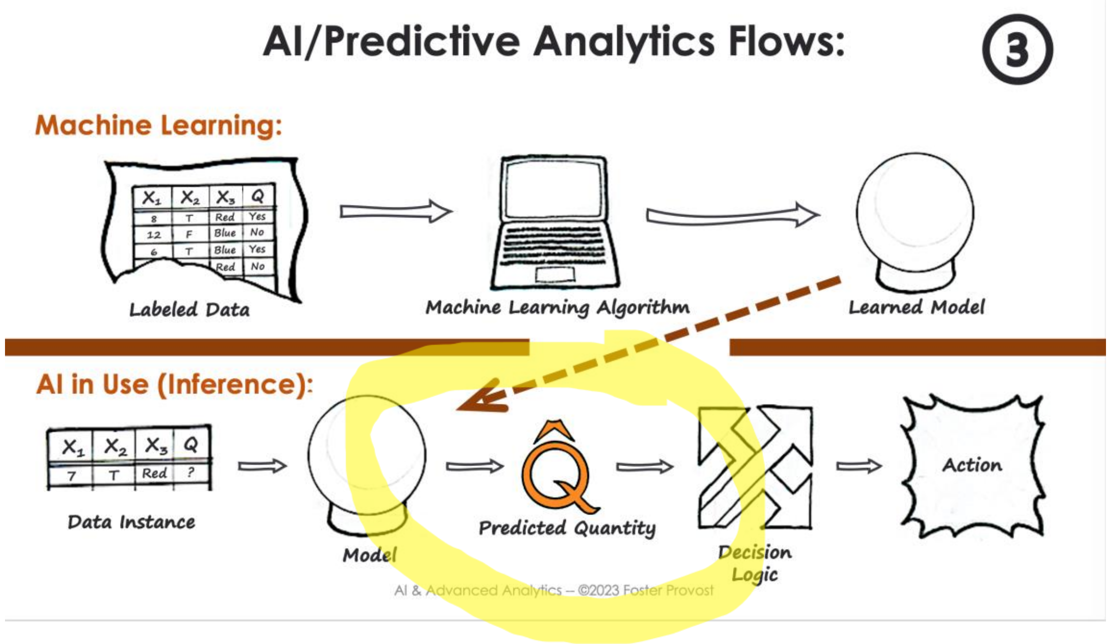
Where we are
So far

model.predict(X)

Gives us a **prediction of the target** (or for classification, a prediction of the probabilities for the target value)

For MegaTelCo – this could be:

Pr(Churn within a year | X)



Where we are
So far

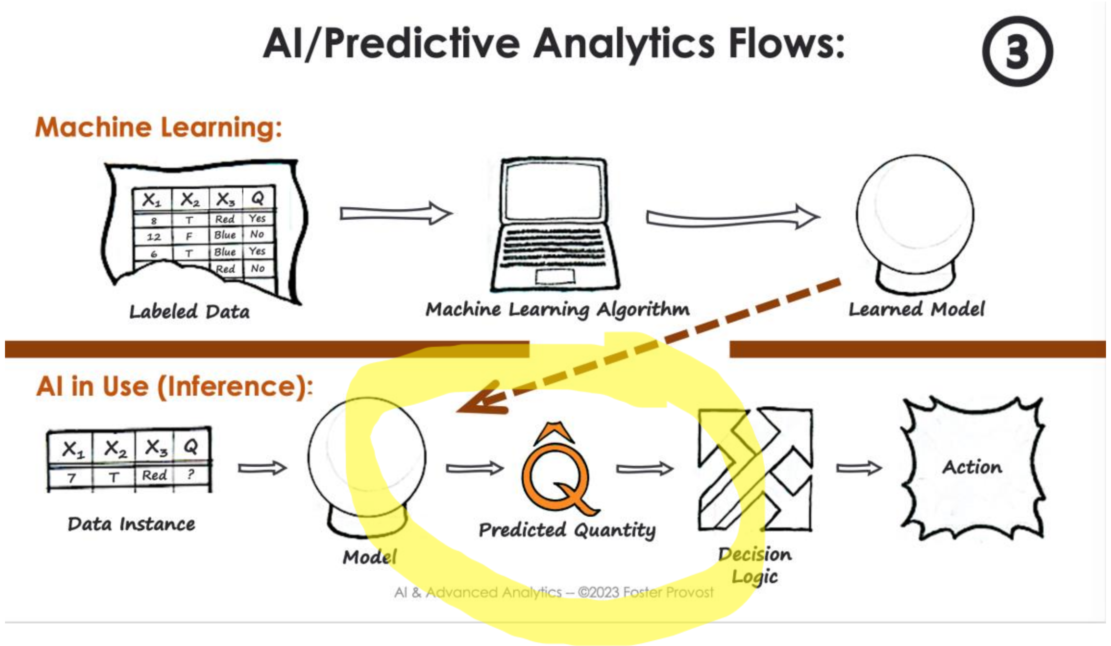
model.predict(X)

Gives us a **prediction of the target** (or for classification, a prediction of the probabilities for the target value)

For MegaTelCo – this could be:

Pr(Churn within a year | X)

"Probability of churn within a year *given* features, x"



Where we are
So far

Almost always, we want to use DS/ML/AI to **make better decisions**!

Where we are
So far

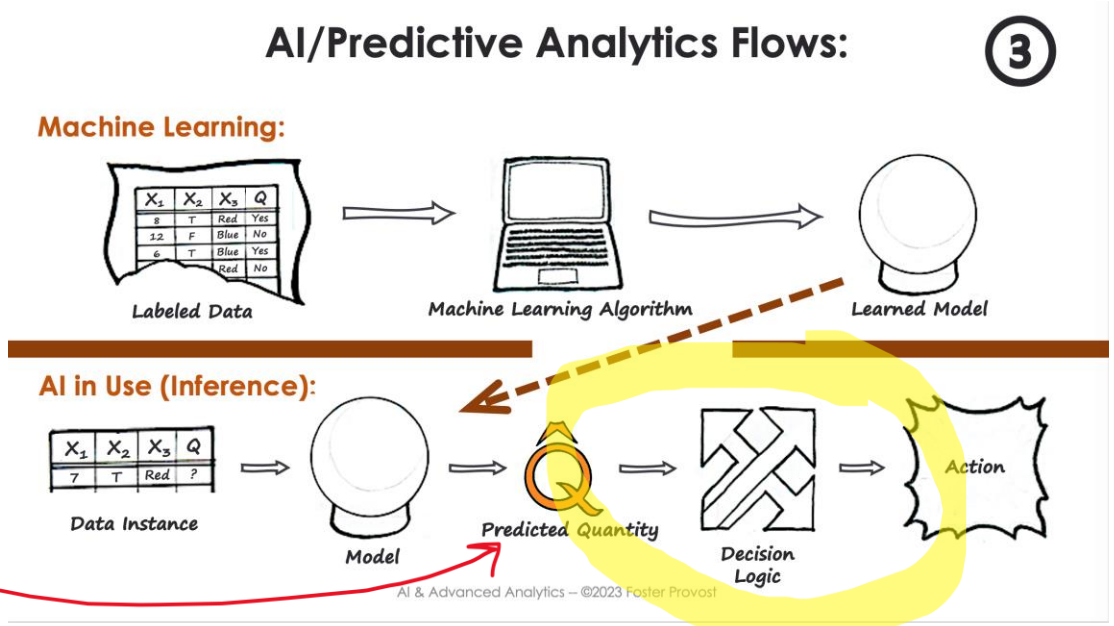
Uh? Not too sure…

With Henrietta and MegaTelCo:

What we have so far,

Pr(Churn within a year | X)

Who really cares what this probability is?

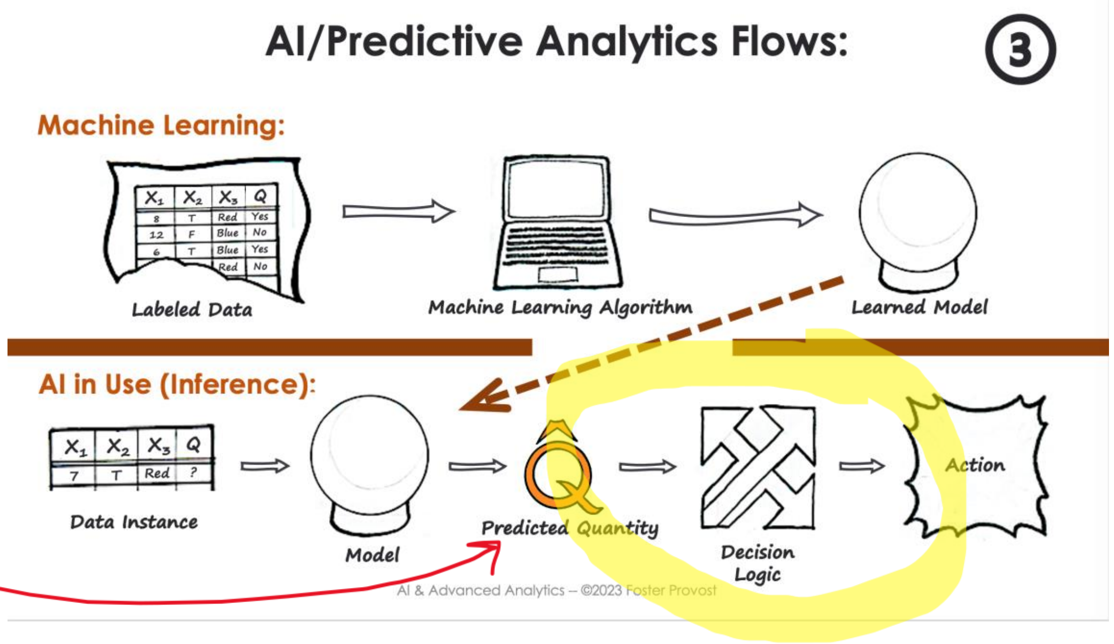


Where we are
So far

With Henrietta and MegaTelCo:

What we have so far,

Pr(Churn within a year | X)



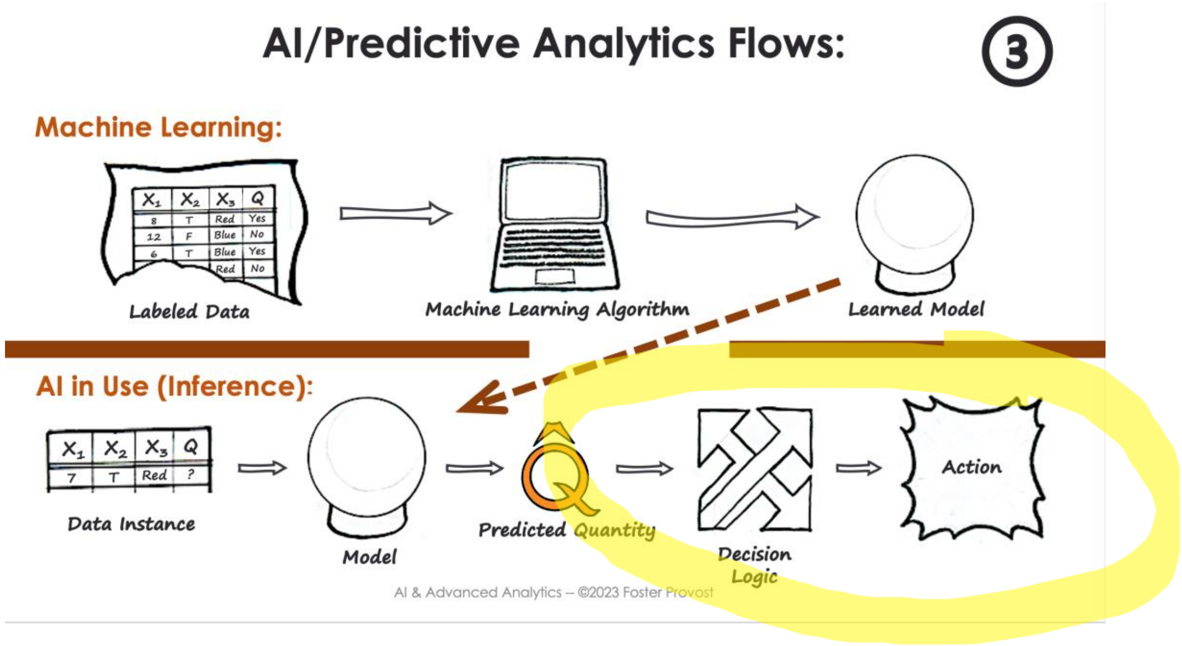
What we actually want to do?

Send out retention offers to minimize churn

How should we use this probability to make this decision? What factors about the offer and the customer (instance) are pertinent to this decision? Discuss!

Where we are

Today



Where we are **Today**

Matrices:

Confusion Matrix

Cost Matrix

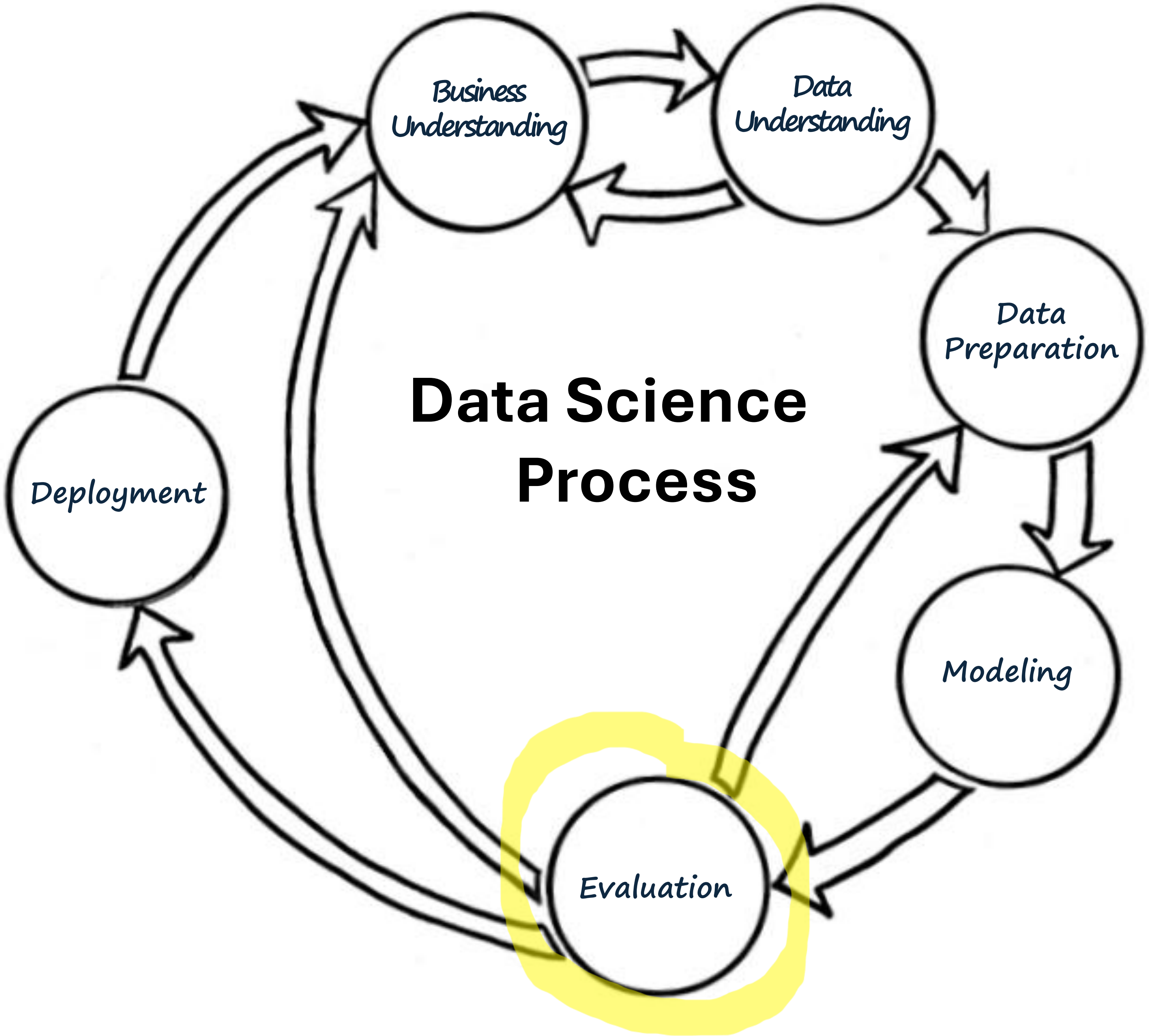
Curves:

ROC curve

Cumulative Response Curves

Lift Curve

Calibration Curve



Notebook time!