# Lecture 01:

## Introduction

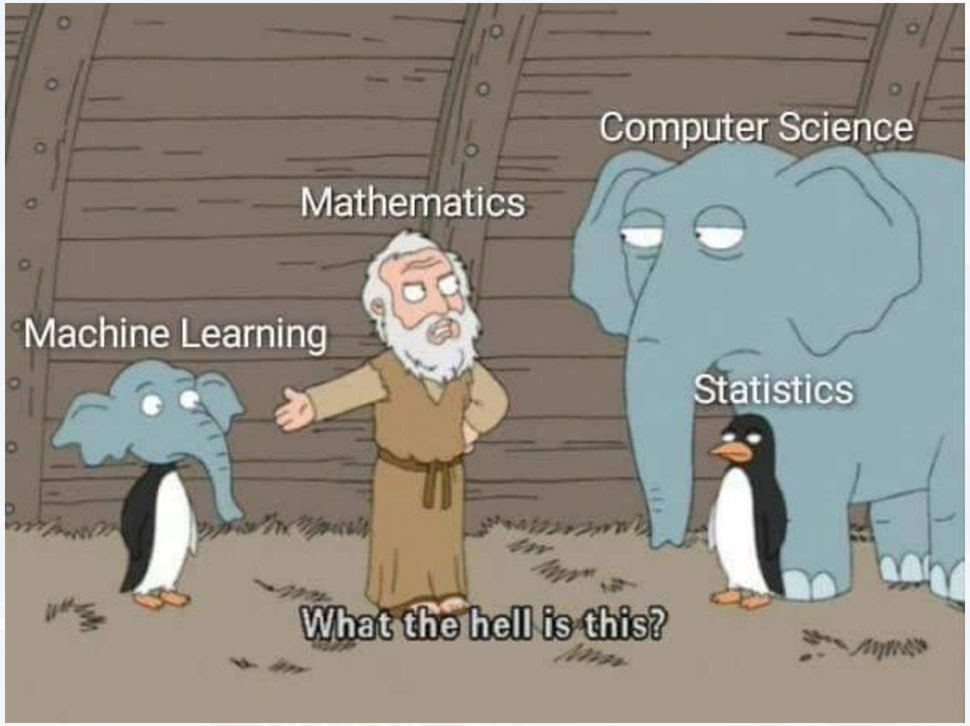

Markus Hohle

University California, Berkeley


**Machine Learning Algorithms**

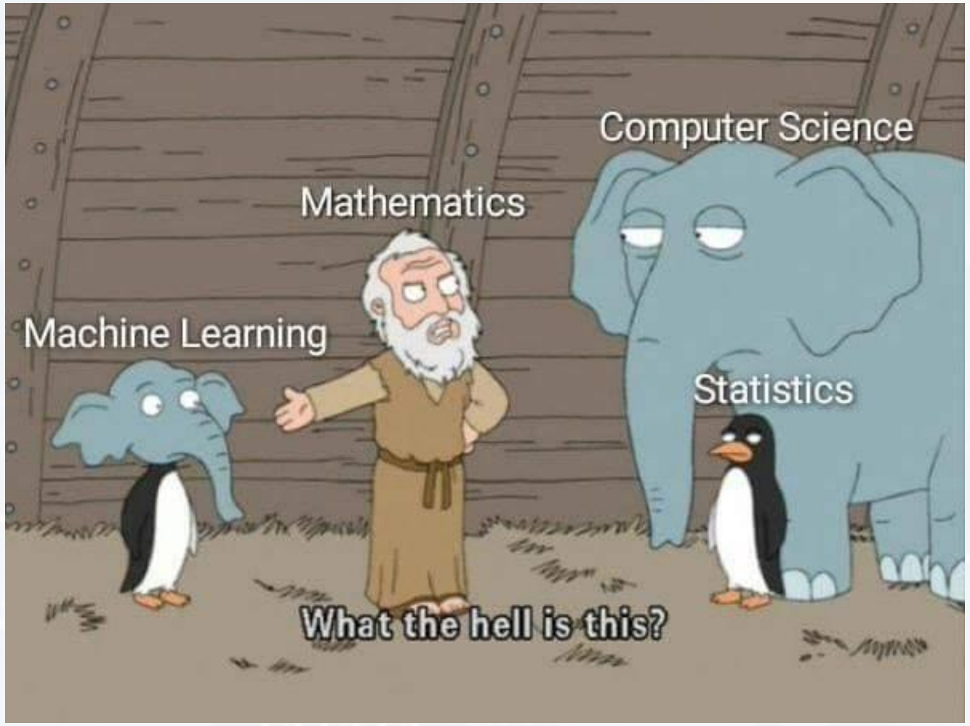
MSSE 277B, 3 Units

Berkeley MSSE
MOLECULAR SCIENCE & SOFTWARE ENGINEERING




Outline

- Motivation
- What is Machine Learning?
- The Lecture
- Course Map



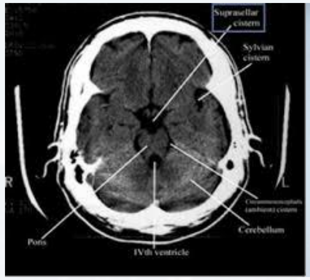

<u>Outline</u>

- Motivation
- What is Machine Learning?
- The Lecture
- Course Map

**Why Machine Learning?**

- **automating workflows**

  → face recognition, identification/authentication tools
  → **manufacturing processes, quality control**
  → **repetitive, standardized workflows**



- **handling large data sets**

  → **epigenetic data for drug development**
  → traffic management
  → **disease control**
  → stock market dynamics

- **generating new data**

  → **new materials (semi/superconductor, linker proteins etc)**
  → AI generated images/videos

**Why Machine Learning?**

LinkedIn: 1st three Job Postings, Dec 19th 2025

## Top job picks for you

Based on your profile, preferences, and activity like applies, searches, and saves

**Research Scientist - Vision Data Infrastructure**

Storm3 • San Francisco, CA (Hybrid) • $250K/yr - $600K/yr • Vision benefit

Actively reviewing applicants

Promoted · Easy Apply

**Staff AI Engineer**

Lumicity • San Francisco Bay Area (Hybrid) • $250K/yr - $280K/yr • 4 benefits

Actively reviewing applicants

Promoted · Easy Apply

**Staff AI/ML Engineer**

Harnham • Sunnyvale, CA (On-site) • $200K/yr - $225K/yr • 4 benefits

Actively reviewing applicants

Promoted · Easy Apply

## What You Bring

- 5+ years building and deploying ML/AI systems at scale.
- Strong Python/Java/C++ skills and experience with ML frameworks (TensorFlow, PyTorch).

**Requirements:**

- Strong experience with **data engineering**, **computer vision**, or **machine learning infrastructure**.
- Expertise in building and scaling **ETL/data pipelines** for large unstructured datasets.
- Proficiency with **Python**, **PyTorch**, and distributed data frameworks (e.g., **Ray**, **Spark**, **Dask**).
- Experience with **WebDataset**, **TFRecords**, **Parquet**, or similar high-throughput data formats.
- Familiarity with **GPU-accelerated preprocessing**, **NVIDIA DALI**, or equivalent systems.

Actively reviewing applicants

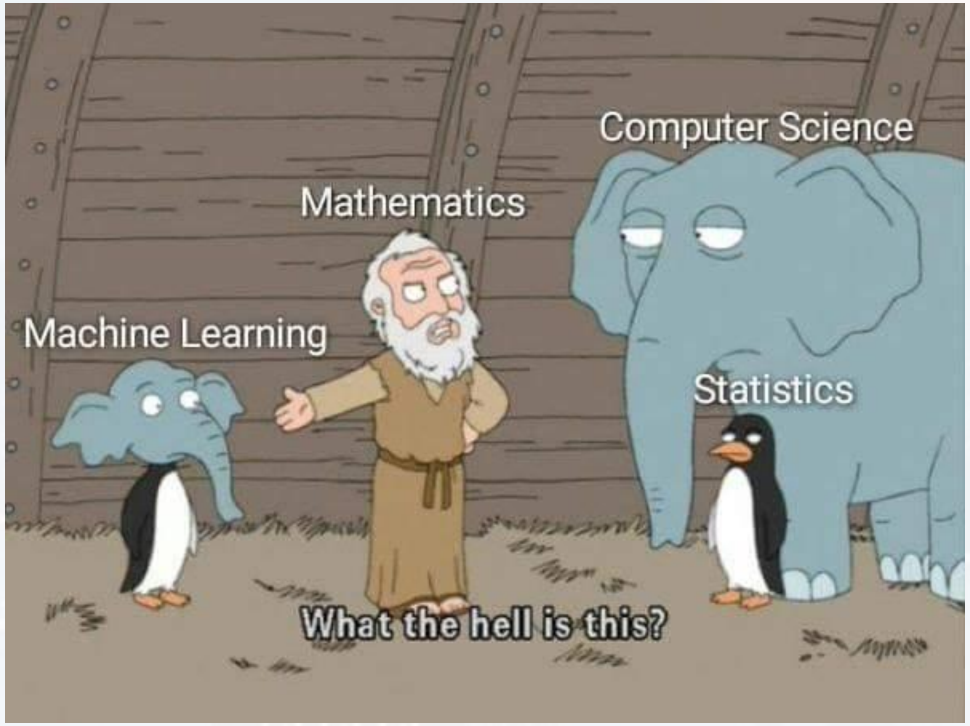
Promoted · Easy Apply

**Why Machine Learning?**

*Listen up son, there are three languages you need to learn in order to become successful:*

- *English*
- *Math*
- *Python *)*

*) or C++




Outline

- Motivation
- What is Machine Learning?
- The Lecture
- Course Map

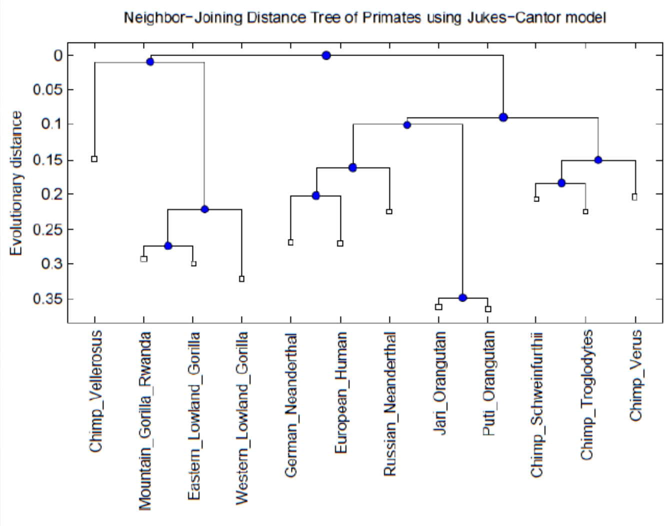
**Wiki says:**

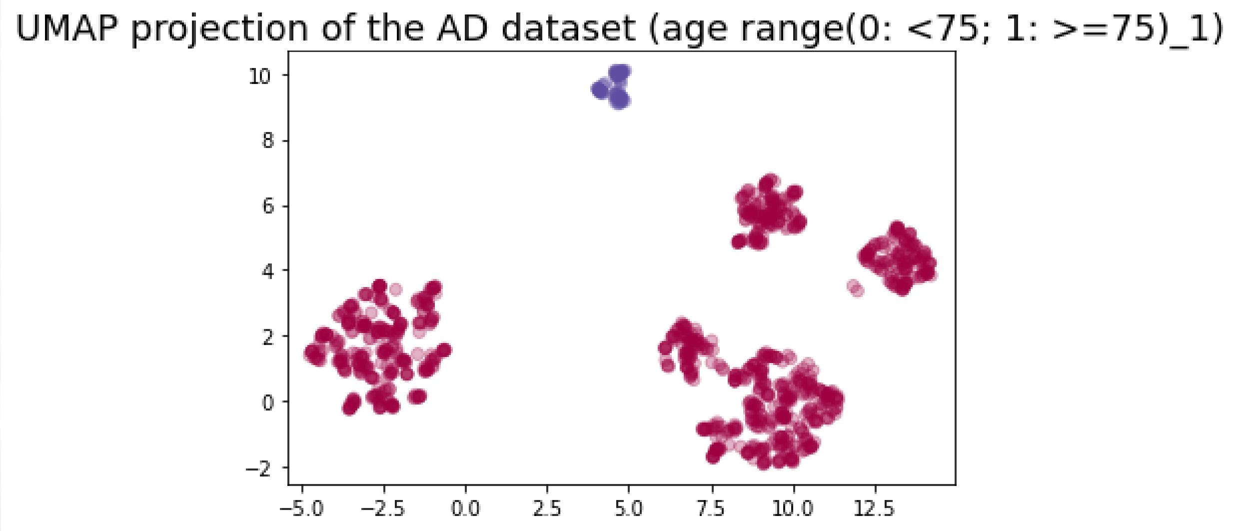
*"**Machine learning** (**ML**) is a field of study in **artificial intelligence** concerned with the development and study of **statistical algorithms** that can learn from data and generalize to unseen data, and thus perform tasks **without explicit instructions**"*

→ **more flexible than standard programs**

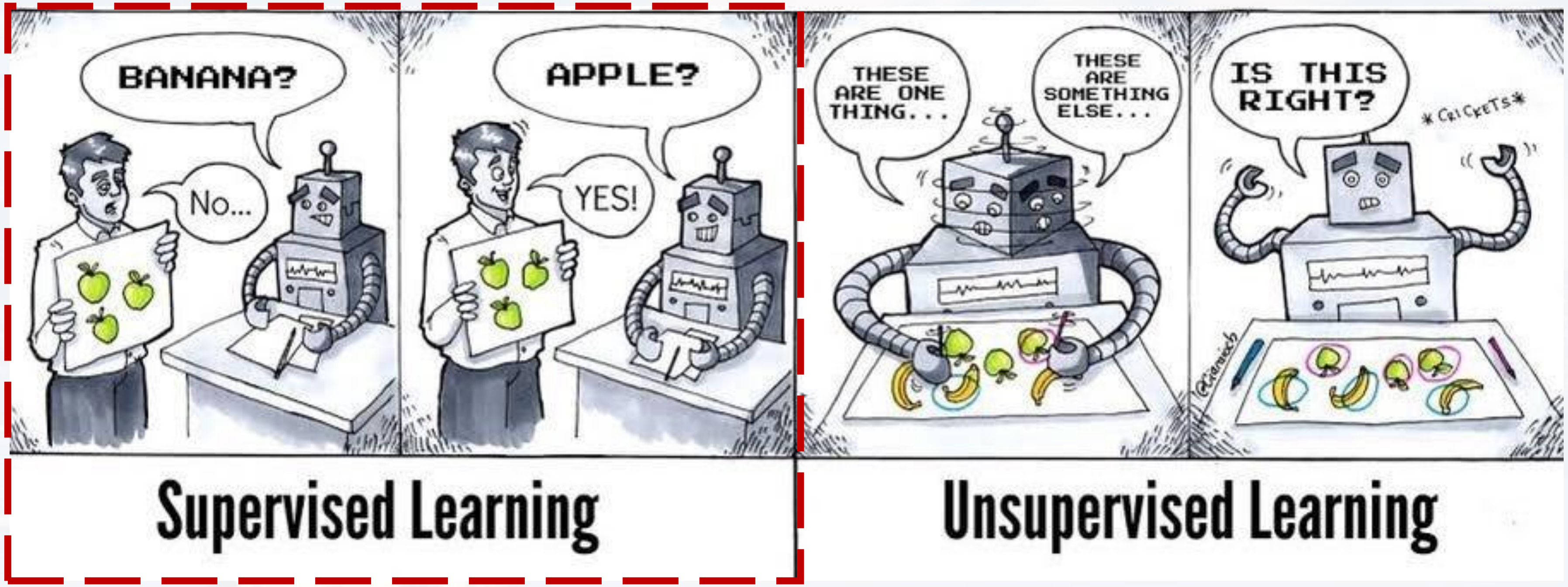
- *artificial intelligence*
- *statistics*

cluster and/or categorize data and understand relations/interactions, find pattern



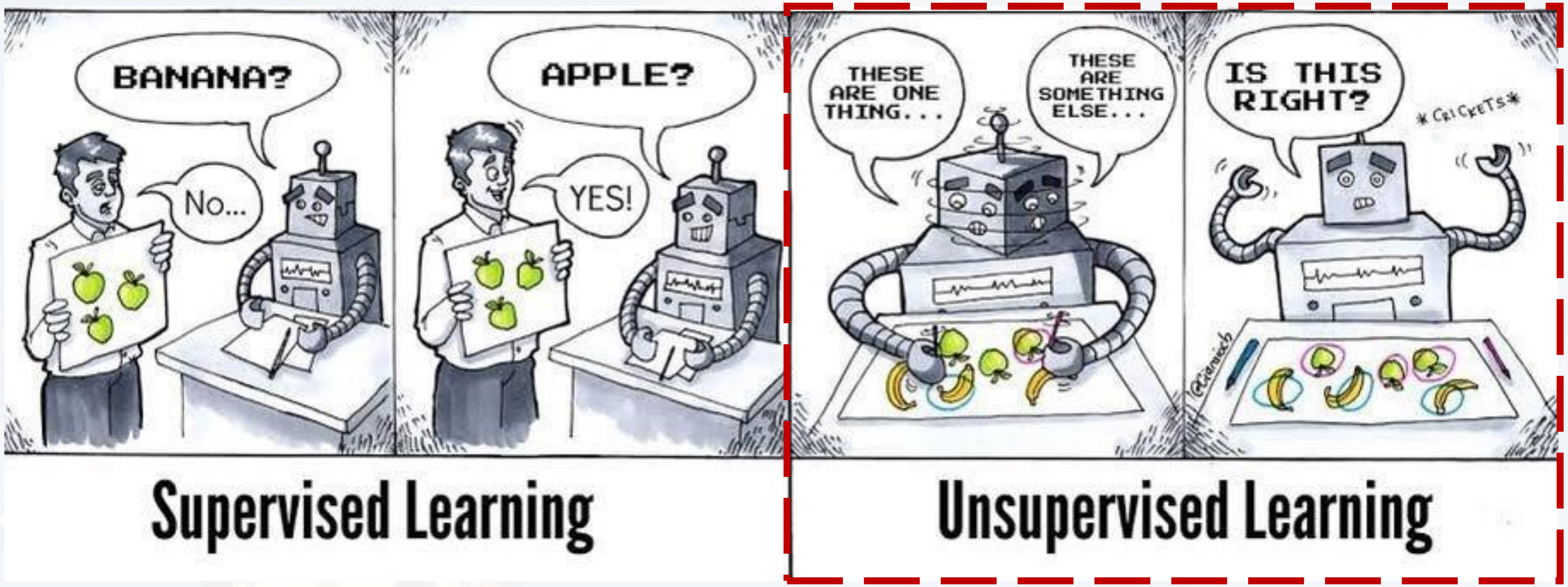
curve fitting, linear/logistic models:
**training** data set and a **test** data set…




- **S**upport **V**ector **M**achine
- K-nearest
- ANNs

curve fitting, linear/logistic models:
**training** data set and a **test** data set…

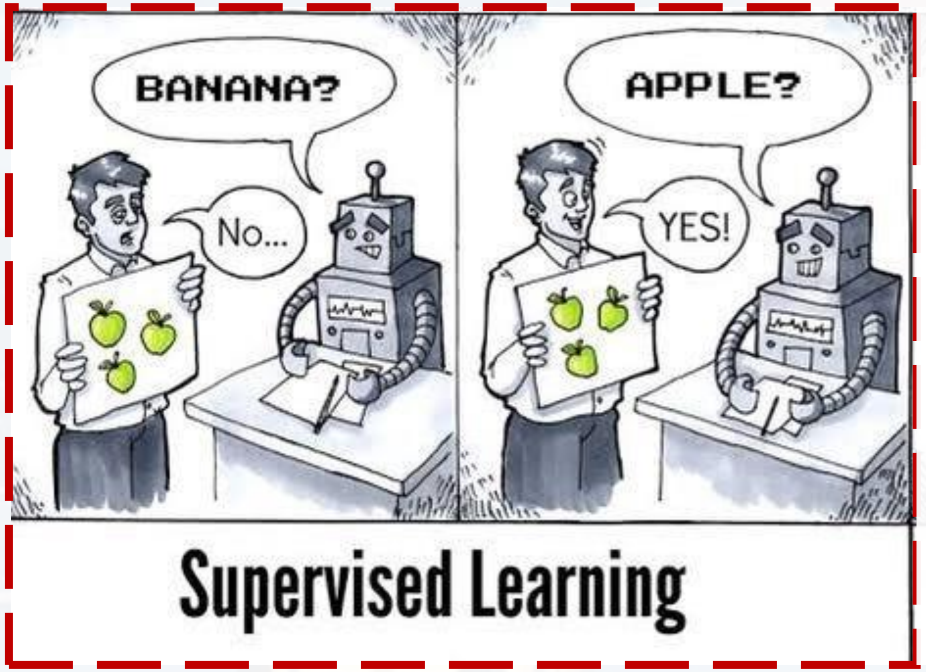
**no training** data set required – we can start right away!




- **S**upport **V**ector **M**achine
- K-nearest
- ANNs

- K - means
- GMM
- trees

curve fitting, linear/logistic models:
**training** data set and a **test** data set…




1) creating the model:

```
my_model = library.method(argument1 = 'arg1'
                                       , … )
```

2) training the model

```
out = my_model.fit(xtrain, ytrain)
```

3) evaluation

```
ypred = out.predict(xeval)
accur = (ypred == yeval).sum()/len(yeval)
```

4) prediction (actual application)

```
ypred = out.predict(xnew)
```

- **S**upport **V**ector **M**achine
- K-nearest
- ANNs

**no training** data set required – we can start right away!

1) creating the model:

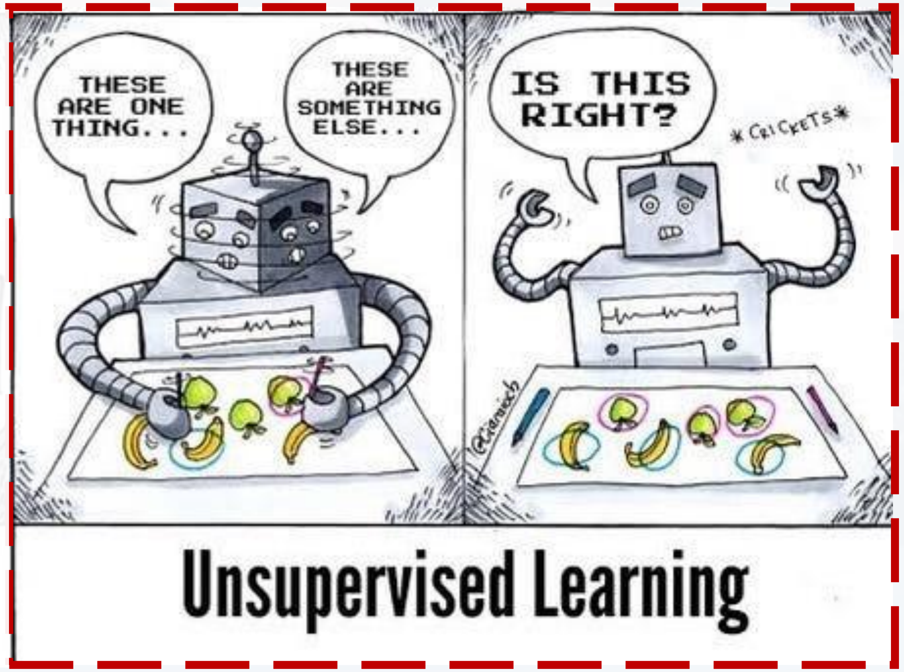
```
my_model = library.method(argument1 = 'arg1'
                                      , … )
```

3) evaluation (*if* y is known)

```
ypred = out.predict(xeval)
accur = (ypred == yeval).sum()/len(yeval)
```

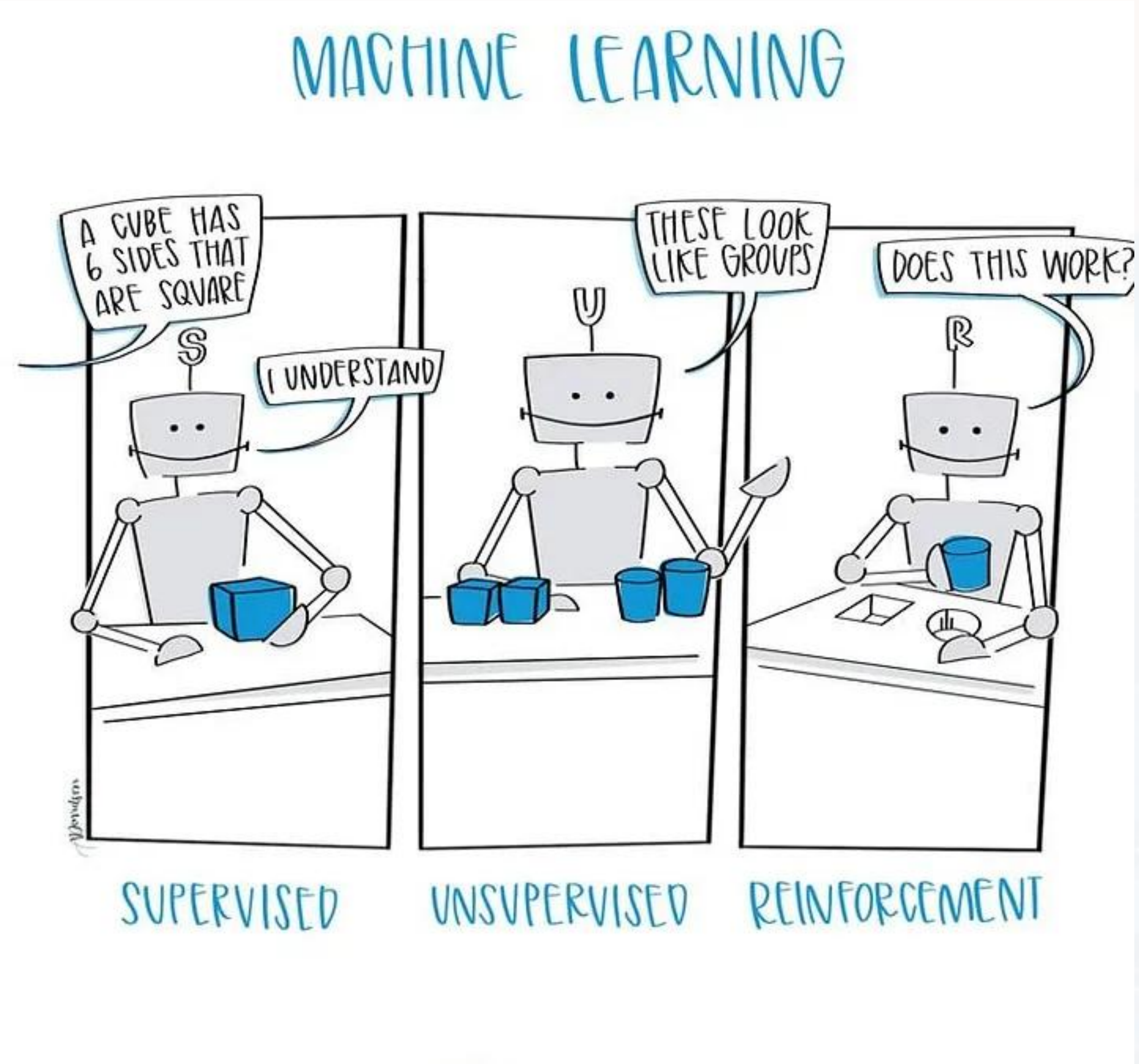
4) prediction (actual application)

```
ypred = out.predict(xnew)
```




- K - means
- GMM
- trees

# What is Machine Learning?



# Introduction

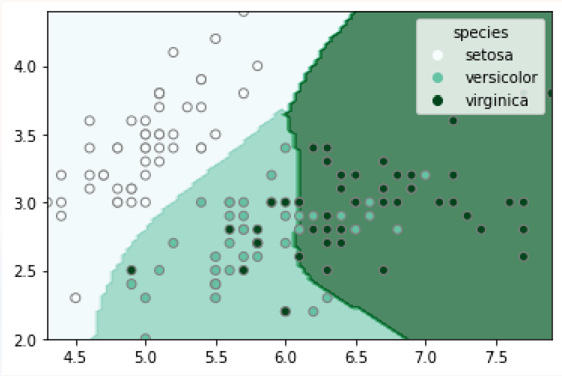


**tasks:**

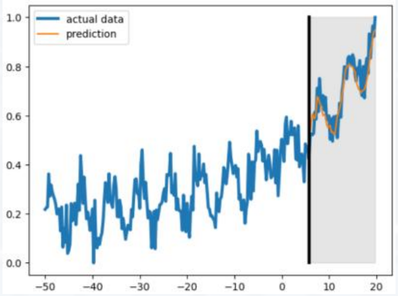
optimizing an *objective* function

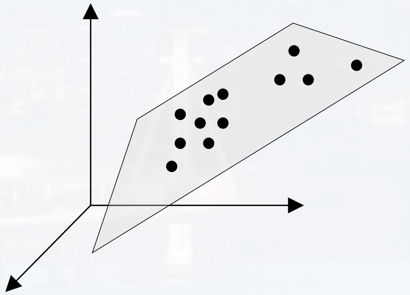
- classification



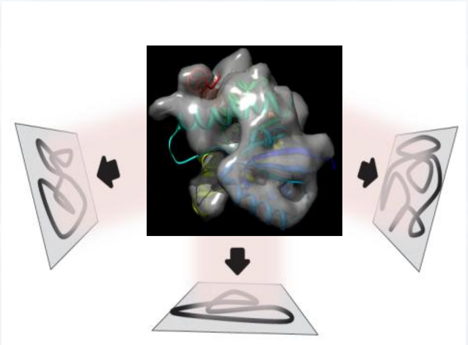



- regression



- generation



# Introduction

## What is Machine Learning?

**tasks:**

- classification

| Index | molecular_weight | electronegativity | bond_lengths | num_hydrogen_bonds | logP | label |
|---|---|---|---|---|---|---|
| 0 | 341.704 | 2.65585 | 3.09407 | 2 | 9.11147 | Toxic |
| 1 | 335.951 | 3.22262 | 2.89039 | 7 | 8.92848 | Toxic |
| 2 | 235.203 | 2.44115 | 2.48203 | 1 | 6.49731 | Non-Toxic |
| 3 | 246.505 | 2.76656 | 2.71547 | 7 | 7.45089 | Non-Toxic |
| 4 | 437.939 | 3.4801 | 3.59569 | 3 | 10.9156 | Non-Toxic |

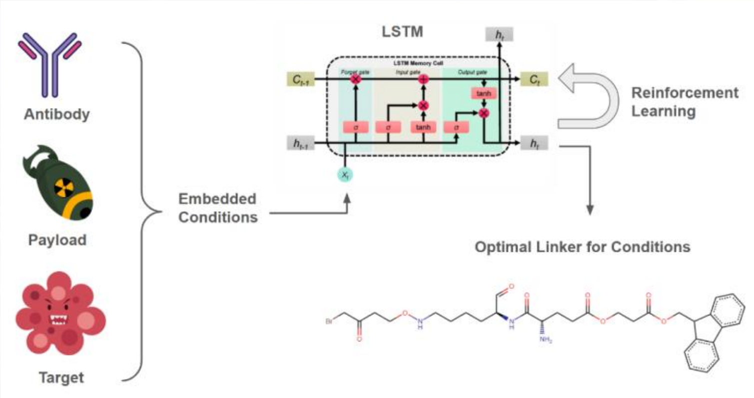
- regression

| Index | molecular_weight | electronegativity | bond_lengths | num_hydrogen_bonds | logP | toxicity_score |
|---|---|---|---|---|---|---|
| 0 | 341.704 | 2.65585 | 3.09407 | 2 | 9.11147 | 80.9281 |
| 1 | 335.951 | 3.22262 | 2.89039 | 7 | 8.92848 | 83.4911 |
| 2 | 235.203 | 2.44115 | 2.48203 | 1 | 6.49731 | 61.8406 |
| 3 | 246.505 | 2.76656 | 2.71547 | 7 | 7.45089 | 57.0538 |
| 4 | 437.939 | 3.4801 | 3.59569 | 3 | 10.9156 | 131.326 |

**Final Project 277B Fall 2025**

**Optimizing ADC Linkers via LSTM Reinforcement Learning**

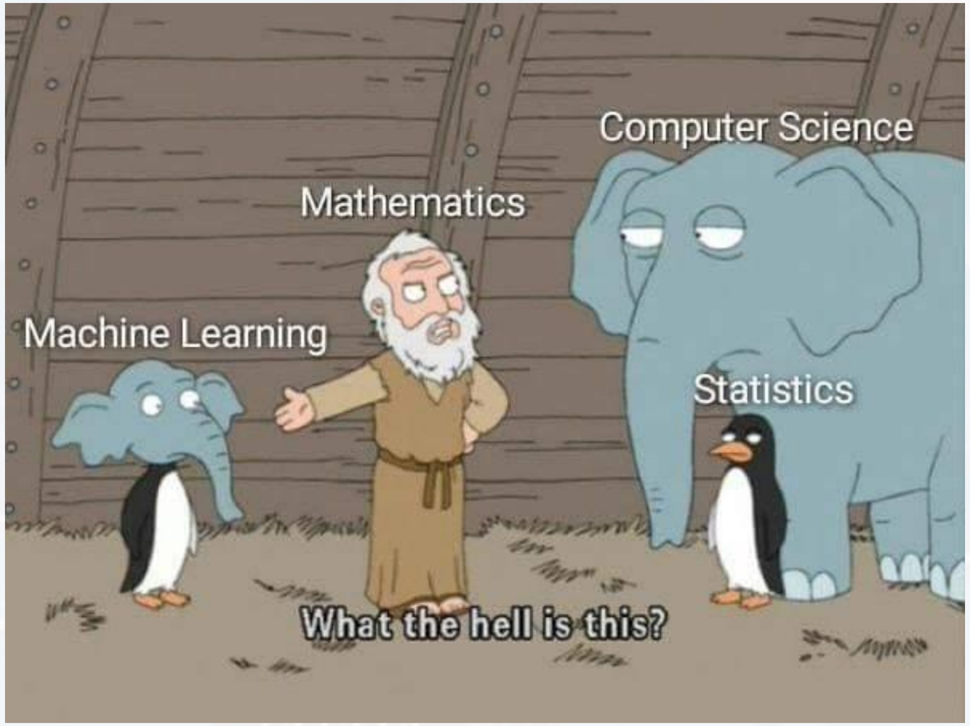
Alex Chase, Jonas Kazimli, Sana Khan, Hailey Monaco, Natalia Rivera, and Leah Sutherland




- generation

“create a molecule with the properties XYZ, such that it is non-toxic”

carrier or linker proteins (drug development)




<u>Outline</u>

- Motivation
- What is Machine Learning?
- The Lecture
- Course Map

GSI: Spencer Uyematsu

Bioinformatics Scientist at Proteios Technology

MSSE alumnus

Lecturer: Markus Hohle

Lecturer at UC Berkeley &

Data Analysis Consultant

PhD Physics

# Introduction



| | |
|---|---|
| Lecture: | Mo 5:30 – 7:30pm PT |
| Discussion (Spencer/Markus): | Th 5:30 – 7:30pm PT |
| Office Hours (Markus): | Fr 5:00 – 7:00pm PT |

- **you can pick a ML/AI project**
- **work in teams of 4/5**
- **start planning after 1/3 of the course**
- **more detailed intro after ½ of the course**
- **submit proposal**
- **checkpoint**
- **presentation**

| | | |
|---|---|---|
| Material: | pre recorded Lectures: | → bcourses |
| | codes/slides | → bcourses & GitHub |
| | HW assignments | → bcourses |
| Grades: | HW assignments: | → 50% |
| | Final Project | → 40% |
| | Course Attendance | → 10% |

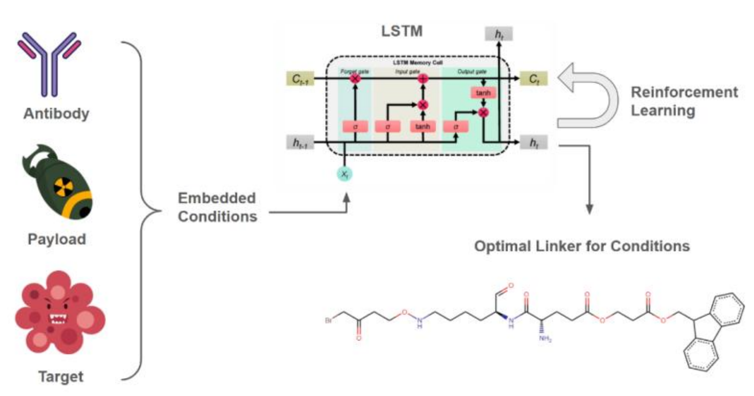
# Introduction



## Final Project examples

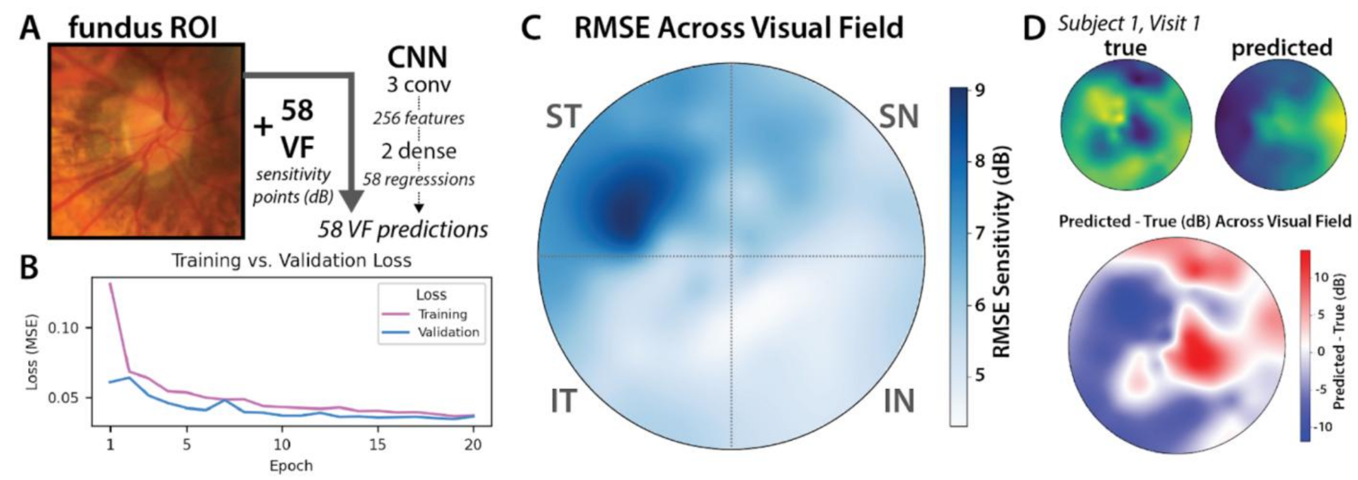
### Optimizing ADC Linkers via LSTM Reinforcement Learning

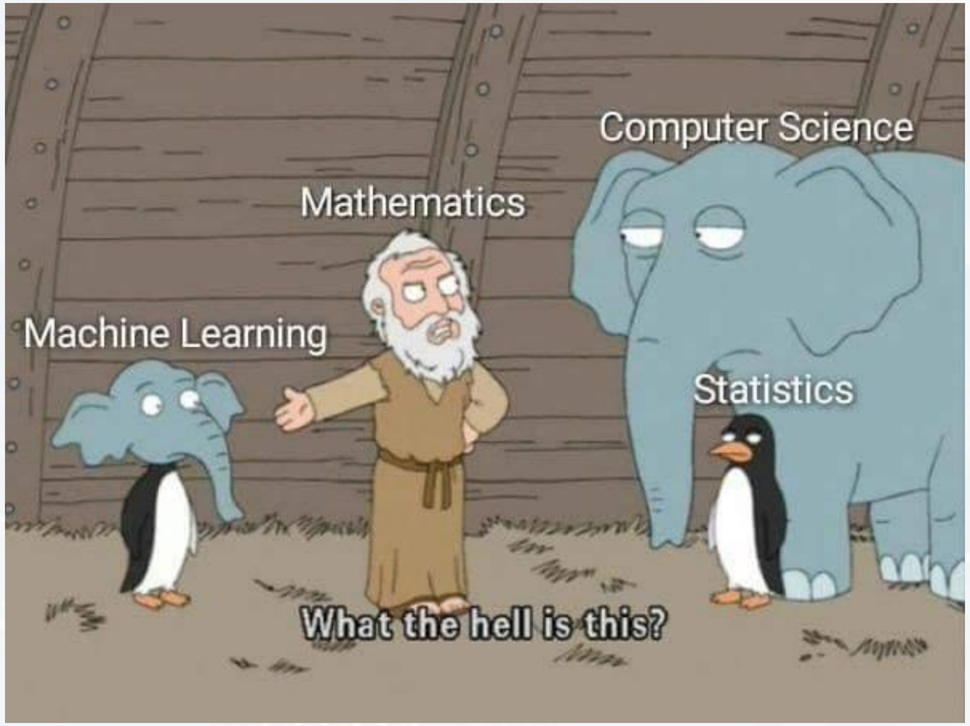
Alex Chase, Jonas Kazimli, Sana Khan, Hailey Monaco, Natalia Rivera, and Leah Sutherland




### *Machine Learning Applications for Visual Field Analysis: A Study of Vision Loss Progression*

Authors: David Houshangi, Lily Hirano, Kirk Ehmsen, Yash Maheshwaran, Christian Fernandez
**University of California at Berkeley - College of Chemistry - Team 1**






Outline

- Motivation
- What is Machine Learning?
- The Lecture
- Course Map

# Introduction

## Course Map

Lecture 1: Course Overview and Introduction to Machine Learning

classic ML tools & algorithms

Lecture 2: Bayesian Methods in Machine Learning

Lecture 3: Dimensionality Reduction: Principal Component Analysis

Lecture 4: Linear and Non-linear Regression and Classification

Lecture 5: Unsupervised Learning: Clustering and Gaussian Mixture Models

Lecture 6: Adaptive Learning and Gradient Descent Optimization Algorithms

ANNs/AI/Deep Learning

Lecture 7: Introduction to Artificial Neural Networks - The Perceptron

Lecture 8: Introduction to Artificial Neural Networks - Building Multiple Dense Layers

Lecture 9: Convolutional Neural Networks (CNNs) - Part I

Lecture 10: CNNs - Part II

Lecture 11: Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs)

Lecture 12: Combining LSTMs and CNNs

Lecture 13: Optional: Running Models on GPUs and Parallel Processing

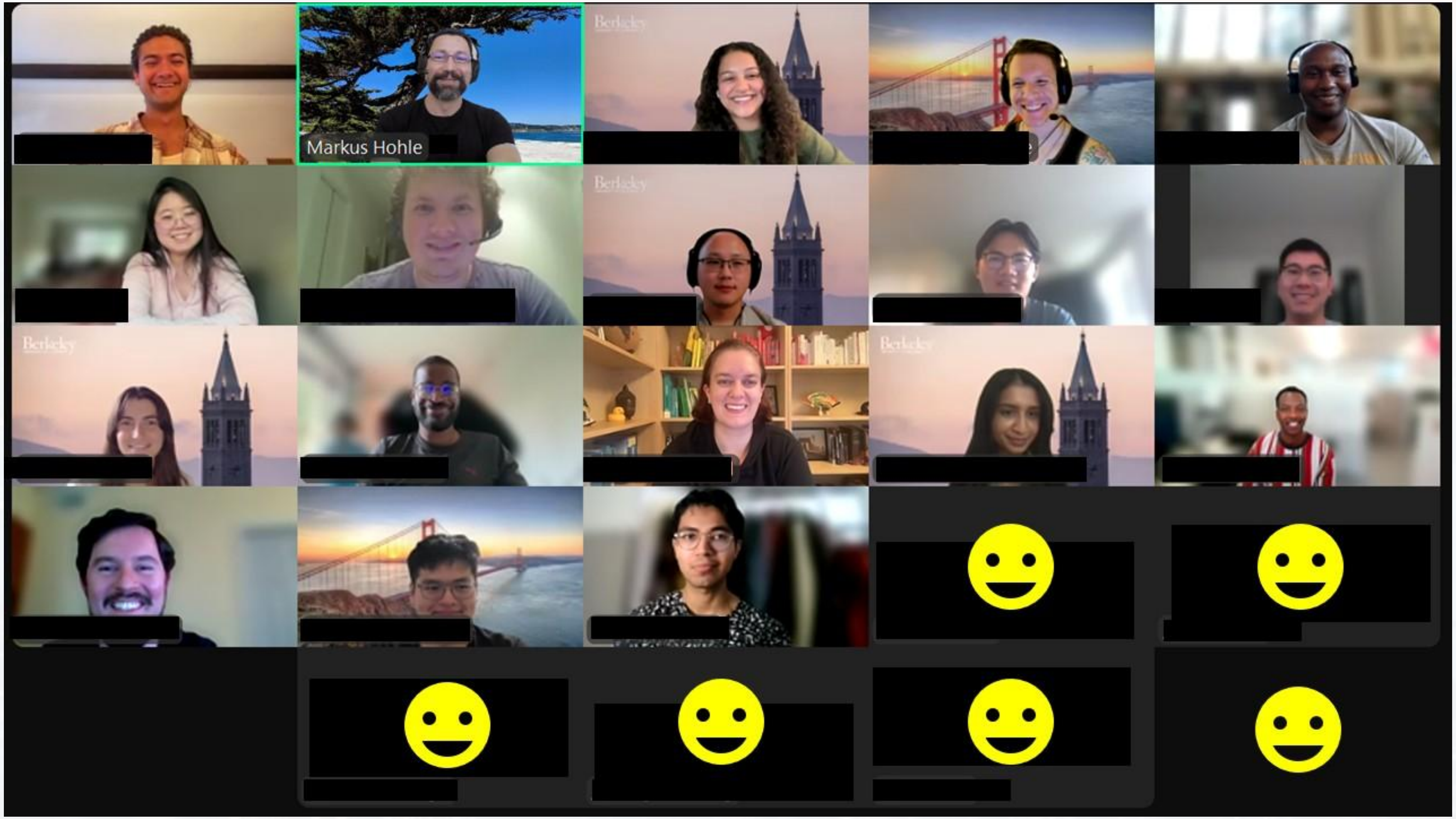
Lecture 14: Project Presentations

Lecture 15: Transformer

Lecture 16: GNN

# Introduction

May 2025



# Introduction



Dec 2025

Thank you very much for your attention!

?Questions?