Welcome to CME 250 Introduction to Machine Learning!

Spring 2020 – Online version

May 4th, 2020

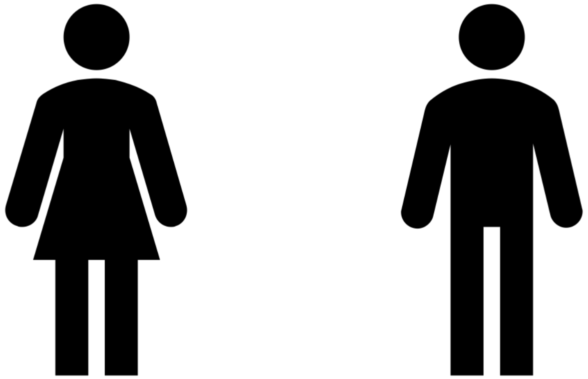
ICME
Stanford | Institute for Computational & Mathematical Engineering

Today's schedule: Wrap-up

- Practice exercise:
  - Regression
  - Classification
  - Model selection using Cross Validation
- What are neural networks?
  - Mathematical expression
  - Similarities to other ML algorithms
  - Main challenges
- What is next?
  - How to keep up with ML?

Let's get to know each other…

Breakout room



You

Another student

Name

Location

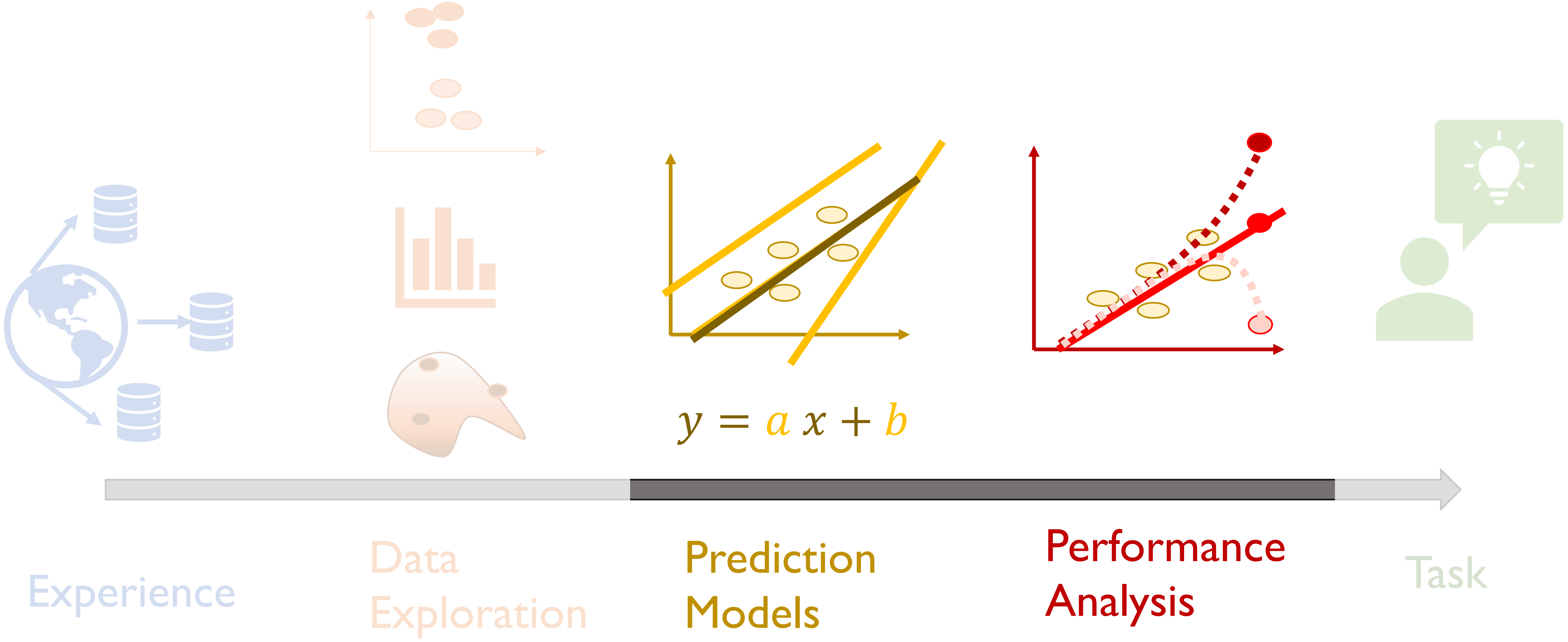
Department

Year

What has changed in the last 5 weeks?

3 mins

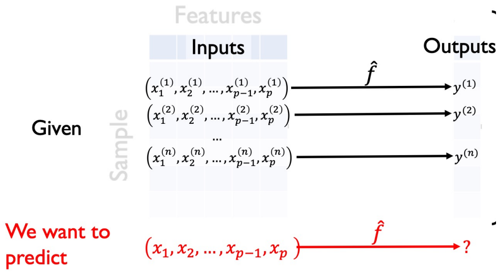
Chat/Audio/Video

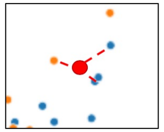
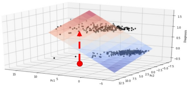
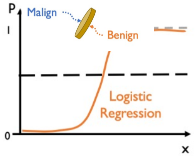
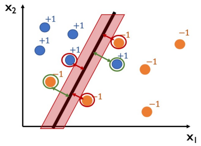
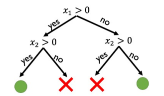
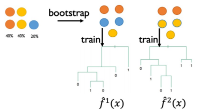
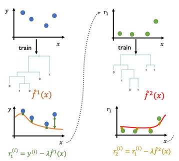

$y = a\,x + b$
Experience
Data Exploration
Prediction Models
Performance Analysis
Task

Recap

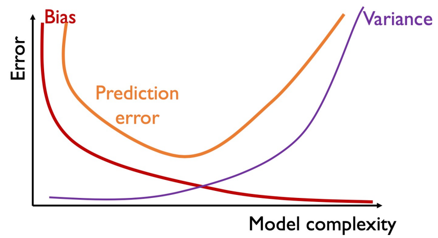
Supervised Learning

Learn from examples

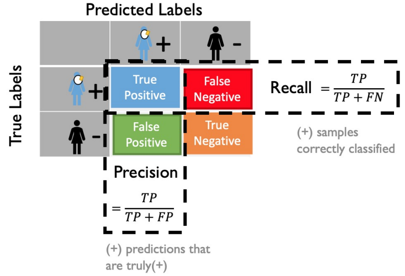


	Regression Y is quantitative	Classification Y is categorical
 KNN	✓	✓
 Linear Regression	✓	Dummy variables
 Logistic Regression	✗	✓
 SVM	✗	✓
 CART	Overfitting vs pruning	Overfitting vs pruning
 Random Forest	✓	✓
 Gradient Boosting Trees	✓	✓

Model Selection

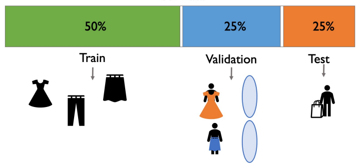


Confusion Matrix



Cross-Validation

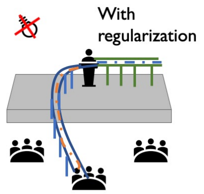
Estimate prediction error



K-fold CV, LOOCV

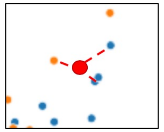
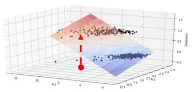
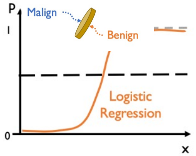
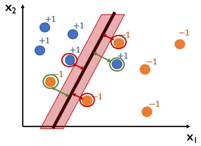
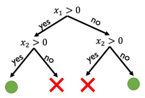
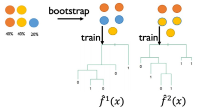
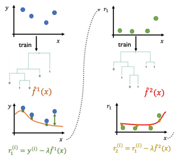
Regularization

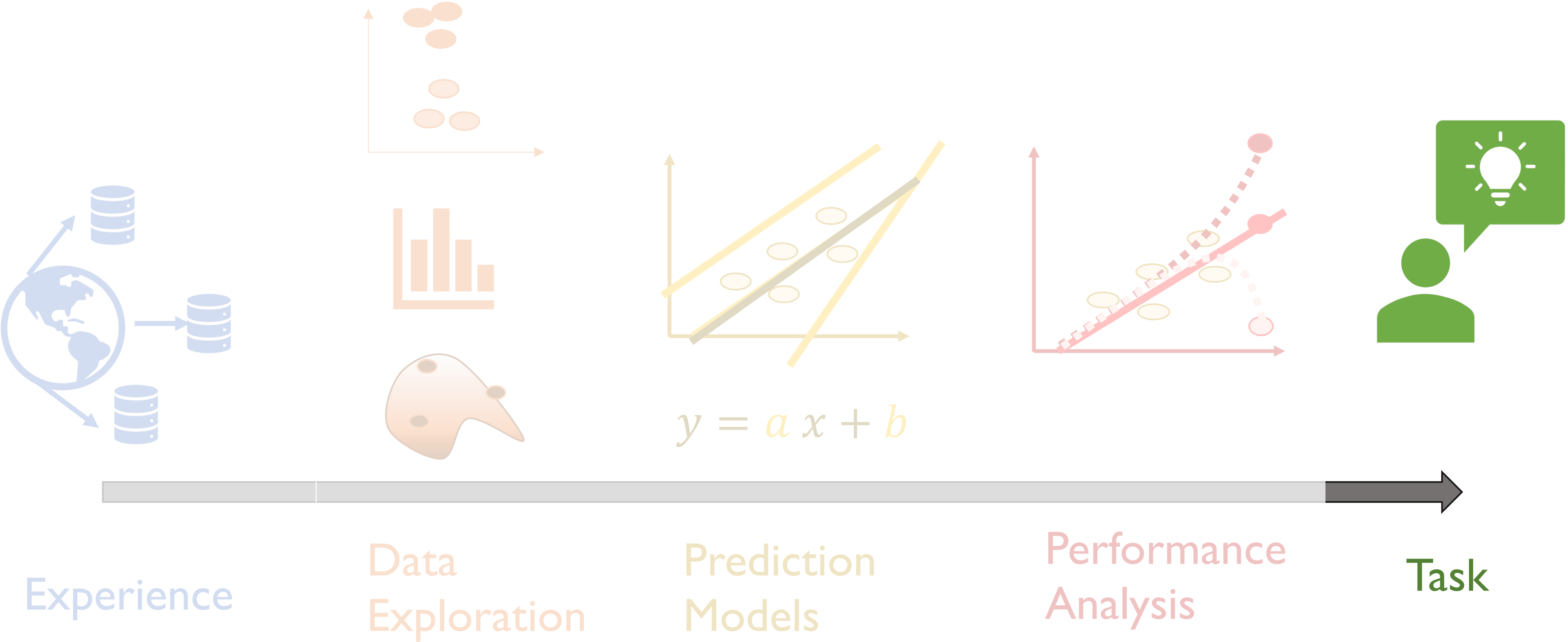
Control complexity: Hyperparameters



Ridge, Lasso

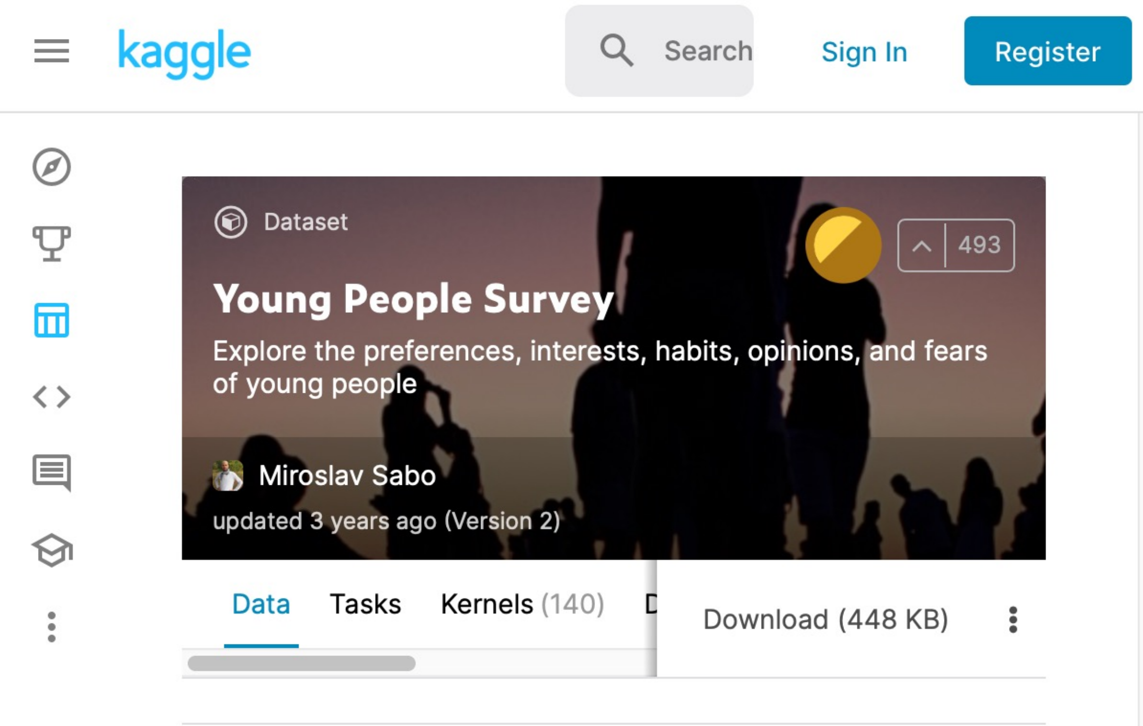
Recap

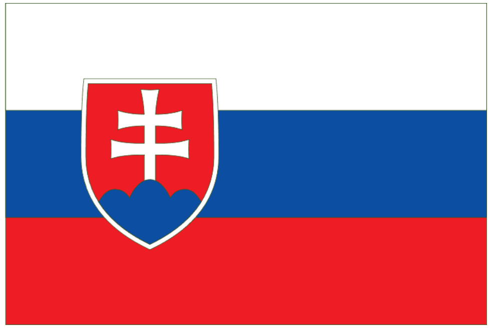
	Regression Y is quantitative	Classification Y is categorical	Interpretability	Flexibility Non-linear boundary	Tuning # Hyperparameters
 KNN	✓	✓	✗	✓	😐 #neighbors, Distance
 Linear Regression	✓	😐 Dummy variables	✓	😐 Create additional features	✓ #Features, Regularization
 Logistic Regression	✗	✓	✓	😐	✓
 SVM	✗	✓	✗	✓	😐 Kernel, Regularization
 CART	😐 Overfitting vs pruning	😐	✓	✓	✓ Tree depth
 Random Forest	✓	✓	😐	✓	😐 Tree depth, # trees, # features, learning rate
 Gradient Boosting Trees	✓	✓	😐	✓	😐



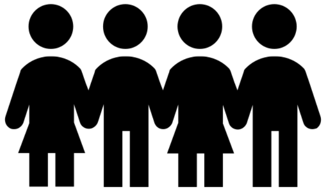
$y = a\,x + b$
Experience
Data Exploration
Prediction Models
Performance Analysis
Task

Example of Supervised Learning :
Young people Survey





Ages 15-30



Music preferences (19 items)
Movie preferences (12 items)
Hobbies & interests (32 items)
Phobias (10 items)
Health habits (3 items)
Personality traits, views on life, & opinions (57 items)
Spending habits (7 items)
Demographics (10 items)

https://www.kaggle.com/miroslavsabo/young-people-survey

Example of Supervised Learning : Goal

Ordered Categorical 1-5

- Music preferences (19 items)
- Movie preferences (12 items)
- Hobbies & interests (32 items)
- Phobias (10 items)
- Health habits (3 items)
- Personality traits, views on life, & opinions (57 items)
- Spending habits (7 items)

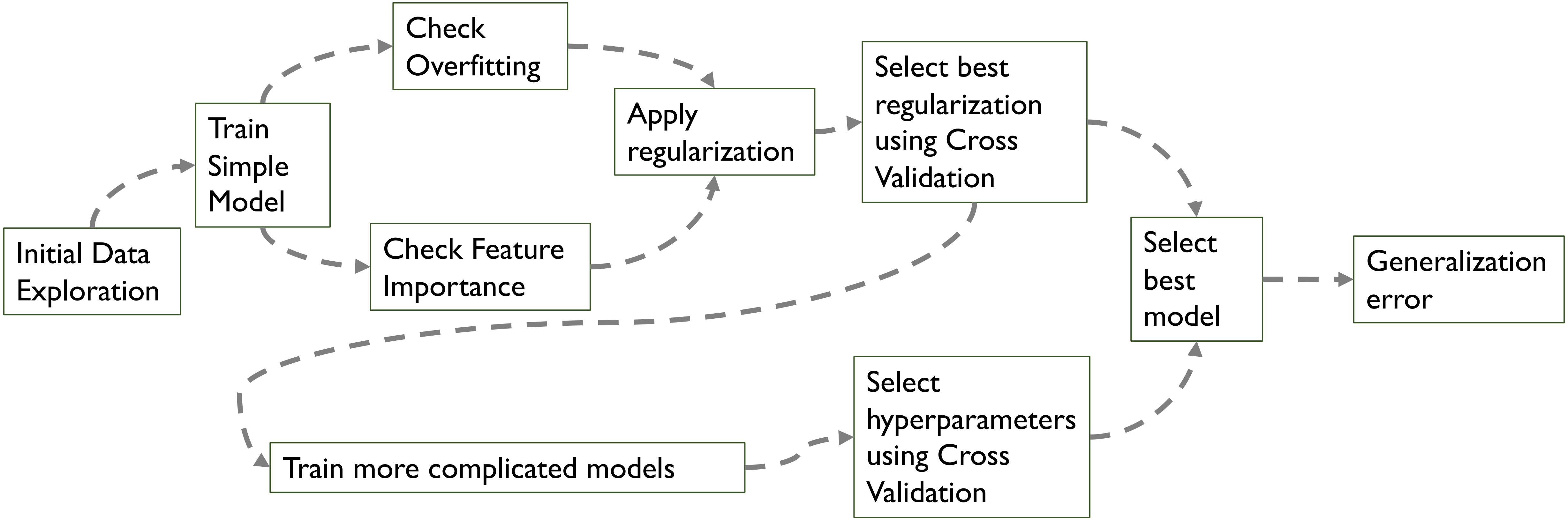
Demographics (8 items)

Predict

Age (Regression)

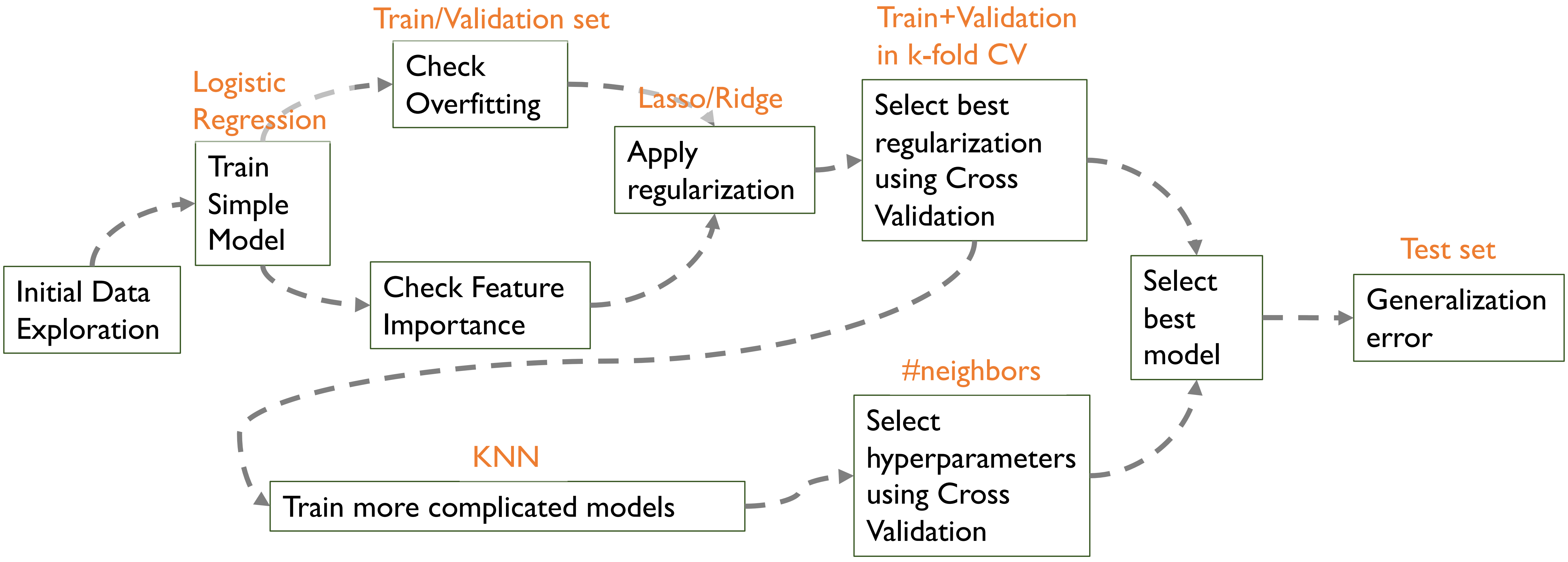
Gender (Classification)

Example of Supervised Learning : Roadmap



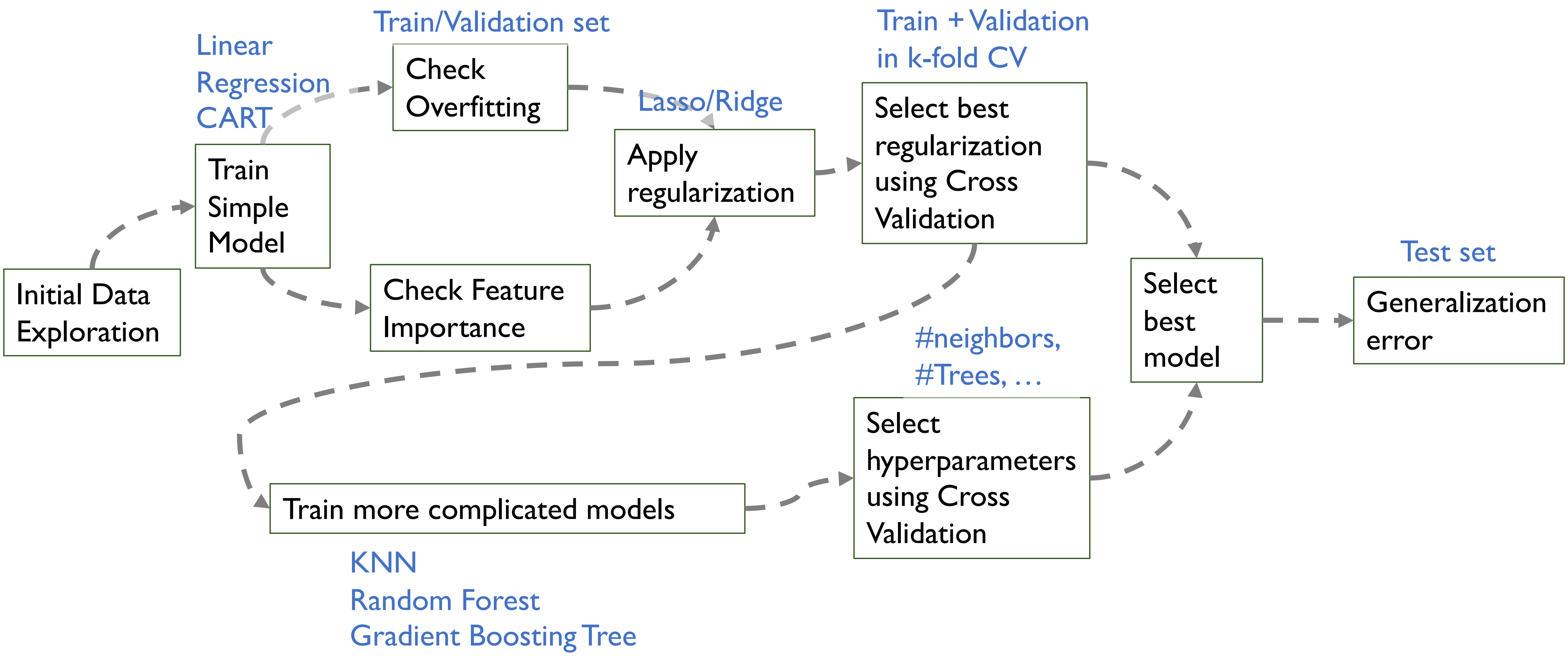
*Not a unique way to approach the problem

Example of Supervised Learning : Roadmap Classification

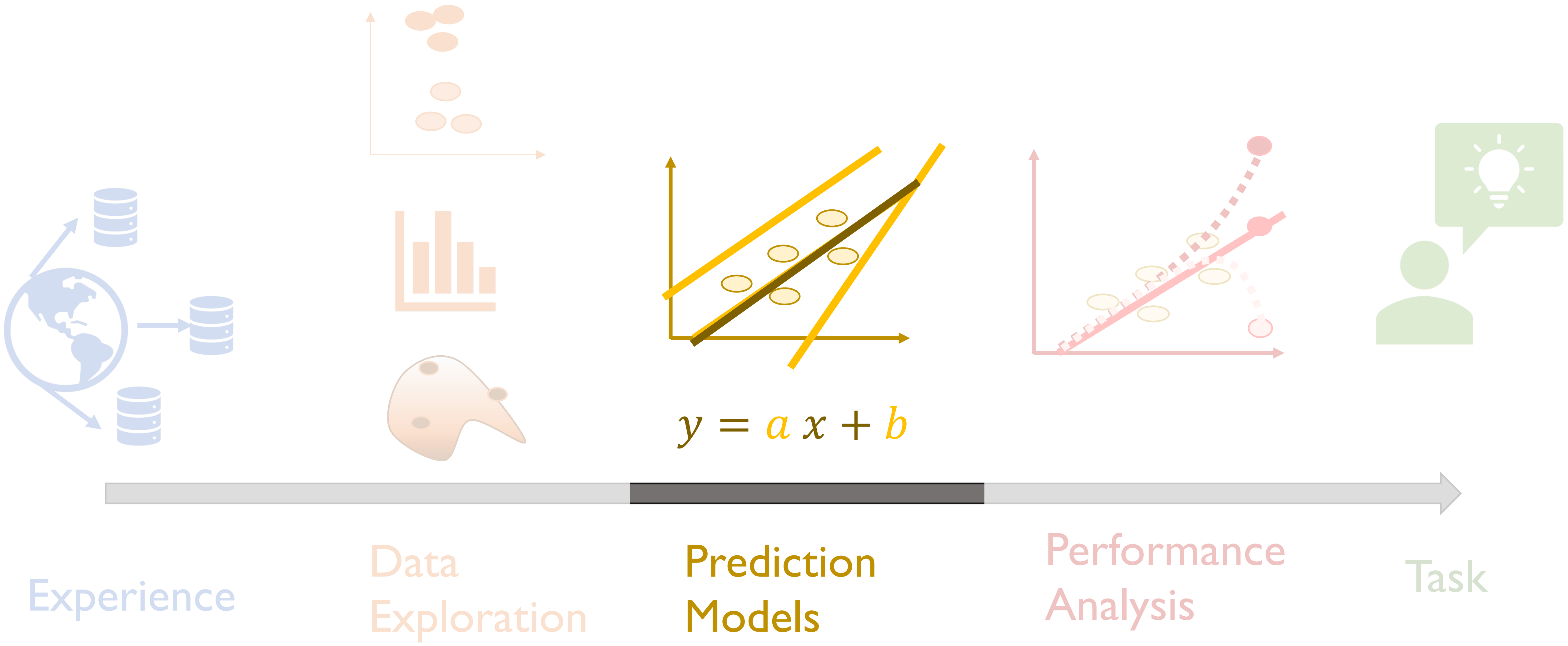


*Not a unique way to approach the problem

Example of Supervised Learning : Roadmap Regression

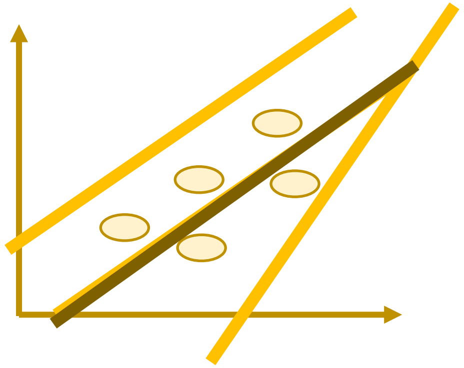


*Not a unique way to approach the problem



$y = a\,x + b$
Experience
Data Exploration
Prediction Models
Performance Analysis
Task

Supervised Learning Part IV:
Intro to Neural Networks & Deep Learning



$$y = a\,x + b$$

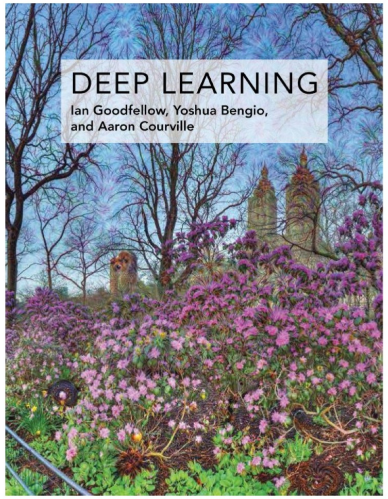
Prediction
Models

Elements Statistical Learning

Chapter 11: (Vanilla) Neural Networks

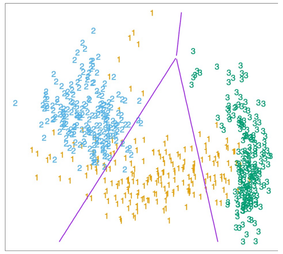
Deep Learning

Ian Goodfellow, Yoshua Bengio, and Aaron Courville

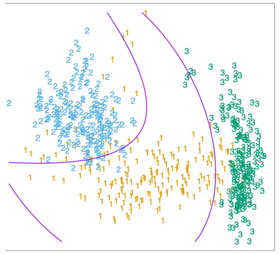


Motivation for Neural Networks

Linear/Logistic Regression



Interpretable but only **linear** decision boundaries

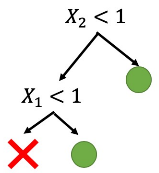


$X_1, X_2,$
$X_1X_2,$
X_1^2, X_2^2

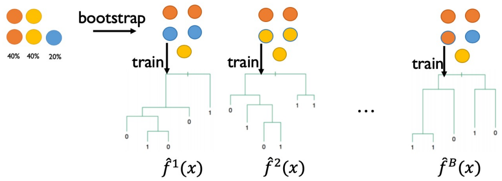
Find "perfect" **features**

Neural Networks: Combine weak learners to create "perfect" features

CART
vs.
Random Forest
(Ensemble methods)



1 weak learner does not have predicting power



Combining many weak learners enhances prediction

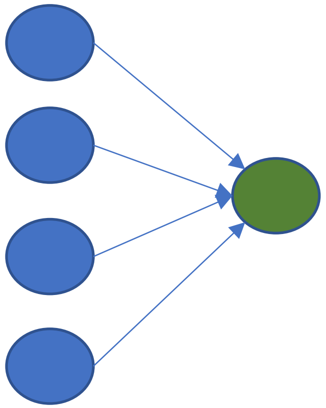
What is a Deep Neural Network?

Linear regression

$$y \approx w^T x$$

Logistic regression

$$y \approx \frac{\exp(w^T x)}{1 + \exp(w^T x)}$$
$$= g(w^T x)$$



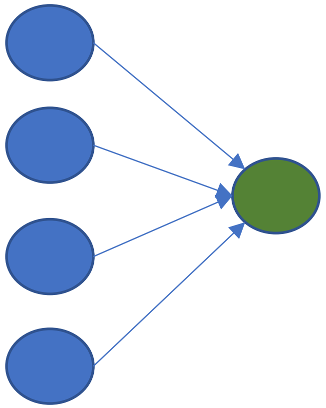
What is a Deep Neural Network?

Linear regression

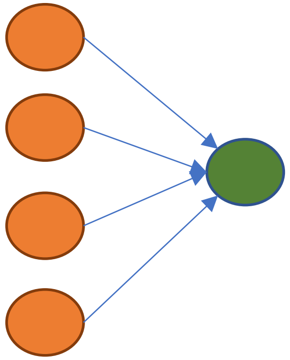
$$y \approx w^T x$$

Logistic regression

$$y \approx \frac{\exp(w^T x)}{1 + \exp(w^T x)}$$
$$= g(w^T x)$$



1-hidden layer
Neural Network



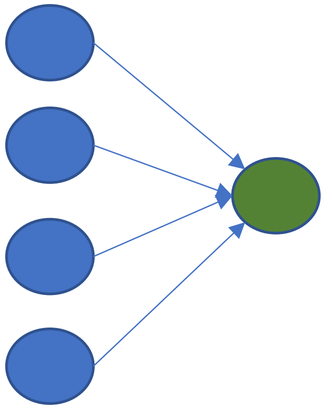
What is a Deep Neural Network?

Linear regression

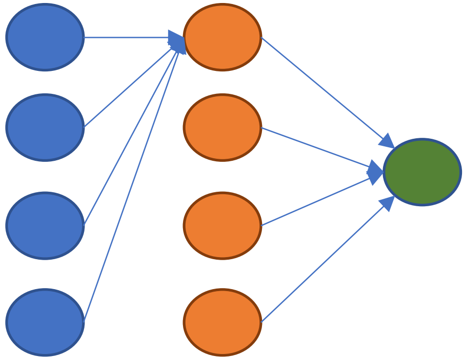
$$y \approx w^T x$$

Logistic regression

$$y \approx \frac{\exp(w^T x)}{1 + \exp(w^T x)}$$
$$= g(w^T x)$$



1-hidden layer
Neural Network



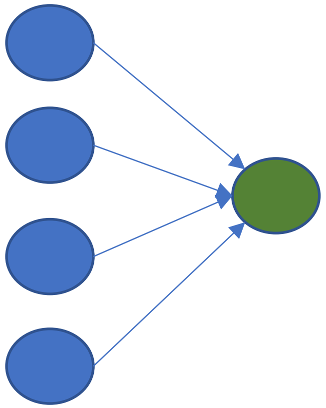
What is a Deep Neural Network?

Linear regression

$$y \approx w^T x$$

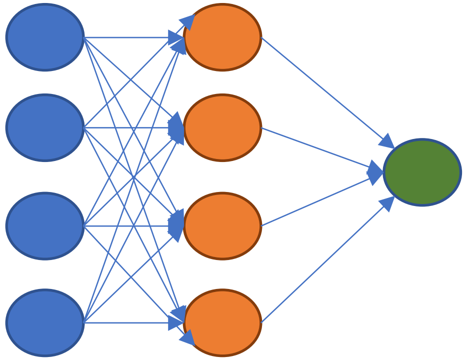
Logistic regression

$$y \approx \frac{\exp(w^T x)}{1 + \exp(w^T x)} = g(w^T x)$$



1-hidden layer Neural Network

$$y \approx h\left(\sum_{m=1}^{k} a_m g(w_m^T x)\right) = h\left(a^T g(W^T x)\right)$$



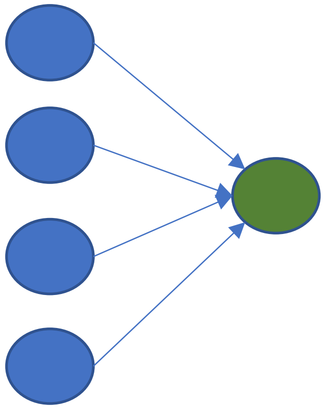
What is a Deep Neural Network?

Linear regression

$$y \approx w^T x$$

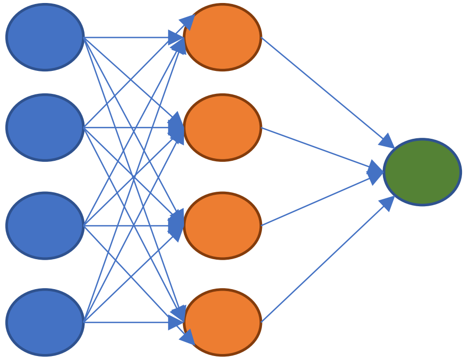
Logistic regression

$$y \approx \frac{\exp(w^T x)}{1 + \exp(w^T x)}$$
$$= g(w^T x)$$

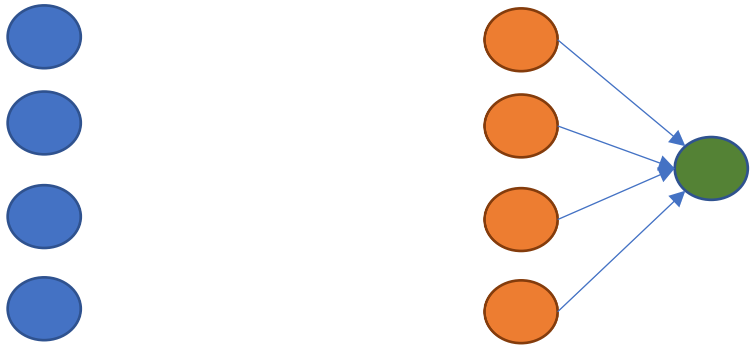


1-hidden layer Neural Network

$$y \approx h\left(\sum_{m=1}^{k} a_m g(w_m^T x)\right)$$
$$= h\left(a^T g(W^T x)\right)$$



Deep Neural Network



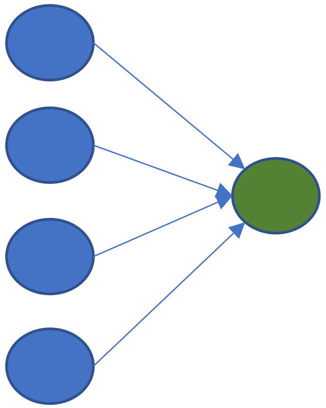
What is a Deep Neural Network?

Linear regression

$$y \approx w^T x$$

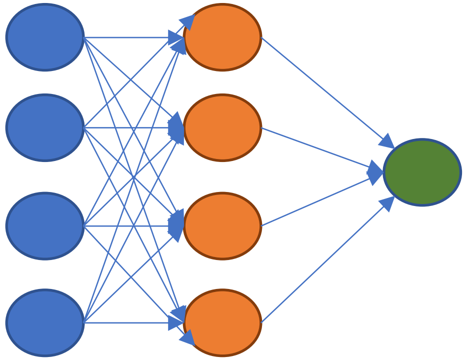
Logistic regression

$$y \approx \frac{\exp(w^T x)}{1 + \exp(w^T x)} = g(w^T x)$$

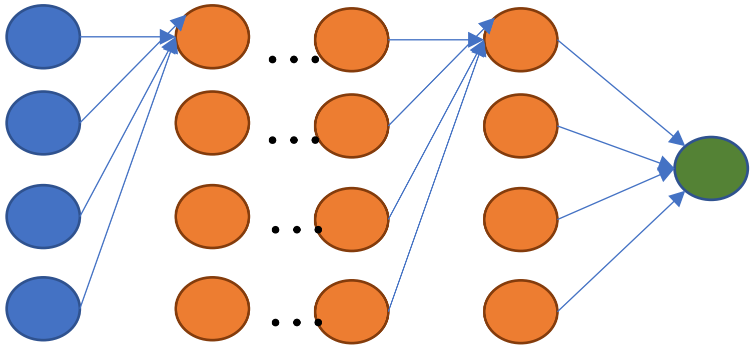


1-hidden layer Neural Network

$$y \approx h\left(\sum_{m=1}^{k} a_m g(w_m^T x)\right) = h\left(a^T g(W^T x)\right)$$



Deep Neural Network



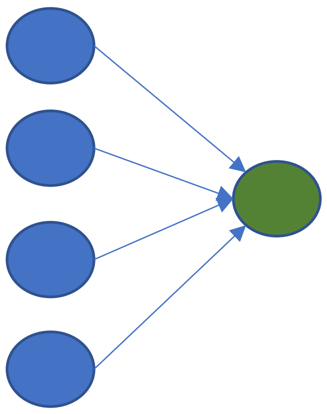
What is a Deep Neural Network?

Linear regression

$$y \approx w^T x$$

Logistic regression

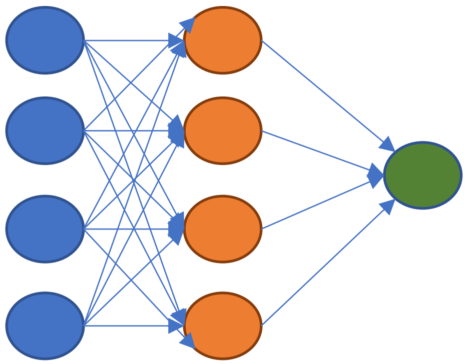
$$y \approx \frac{\exp(w^T x)}{1 + \exp(w^T x)}$$
$$= g(w^T x)$$



#params = d

1-hidden layer Neural Network

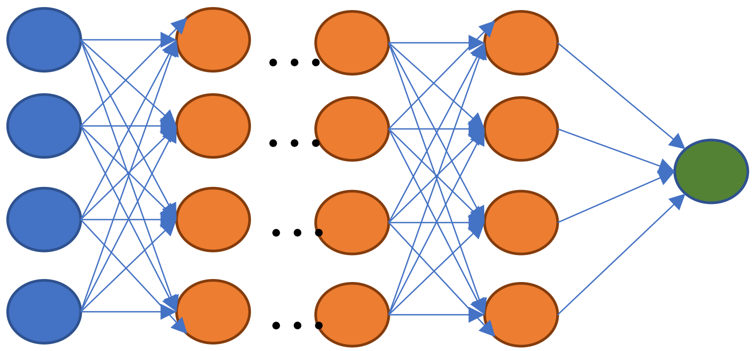
$$y \approx h\left(\sum_{m=1}^{k} a_m g(w_m^T x)\right)$$
$$= h\left(a^T g(W^T x)\right)$$



#params = d*k + k

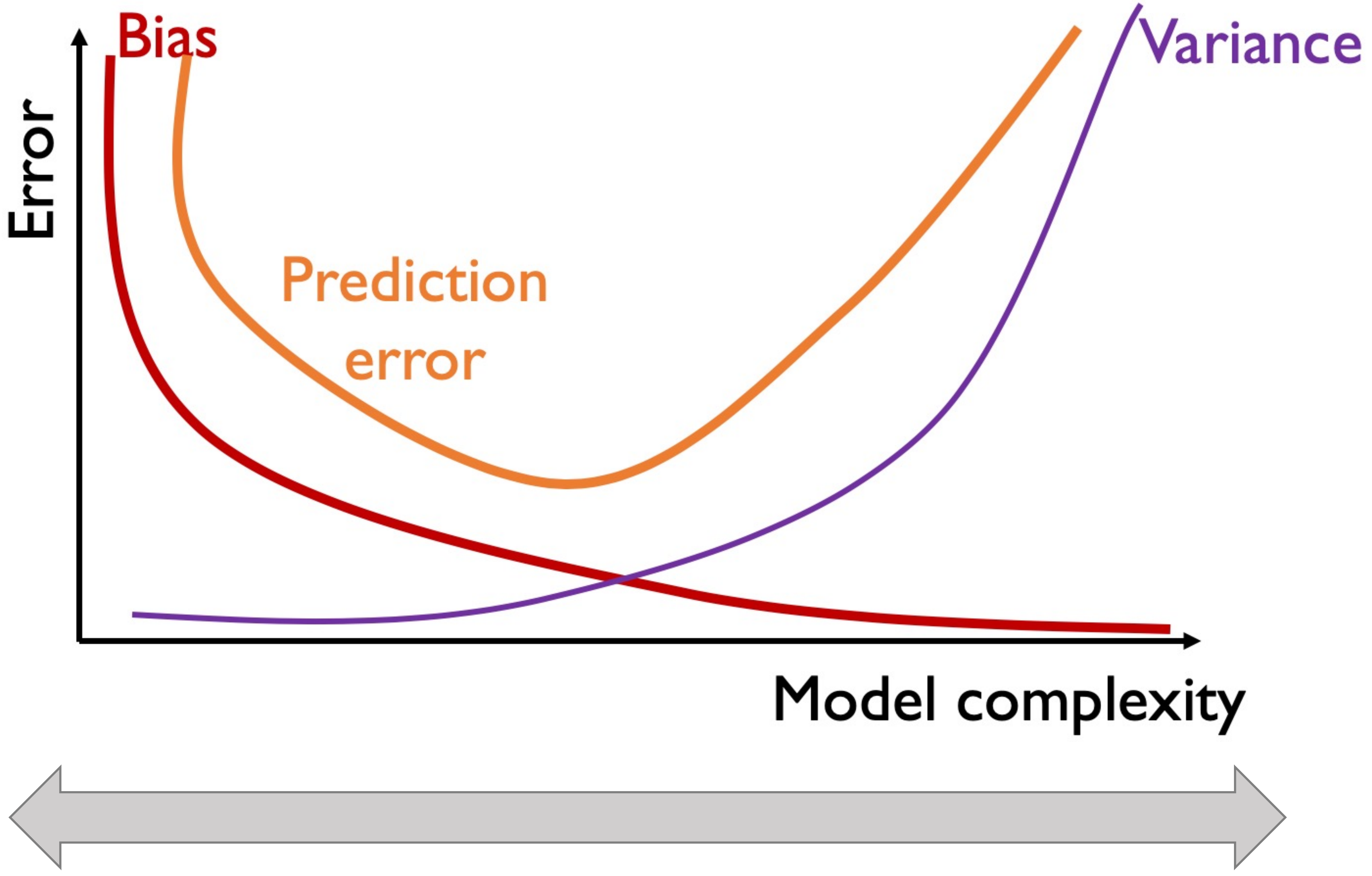
Deep Neural Network

$$z_1 = g(W_1^T x),$$
$$\dots,$$
$$z_l = g\left(W_l^T z_{l-1}\right),$$
$$y \approx h(a^T z_l),$$

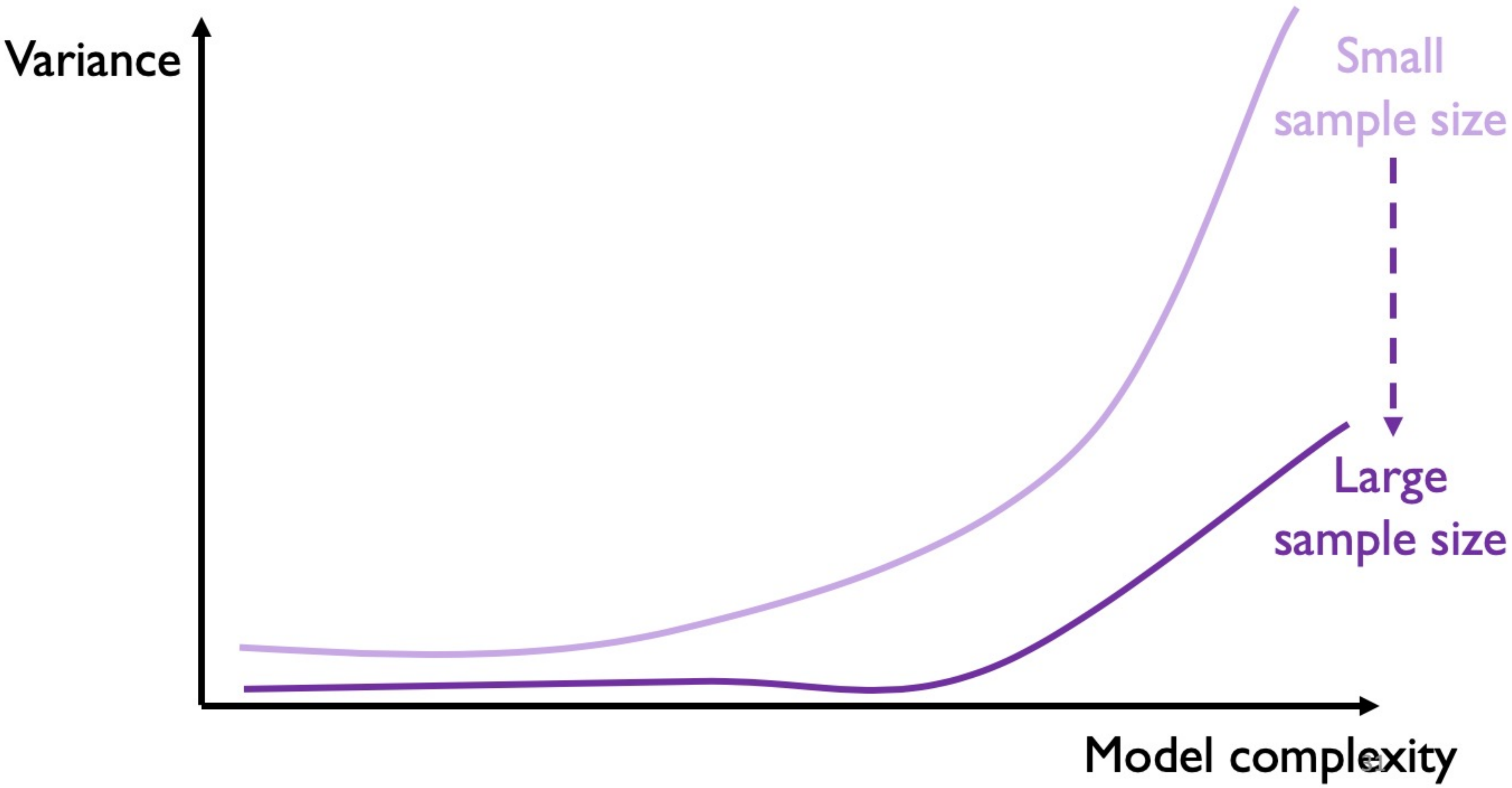


#params = l*(d*k) + k

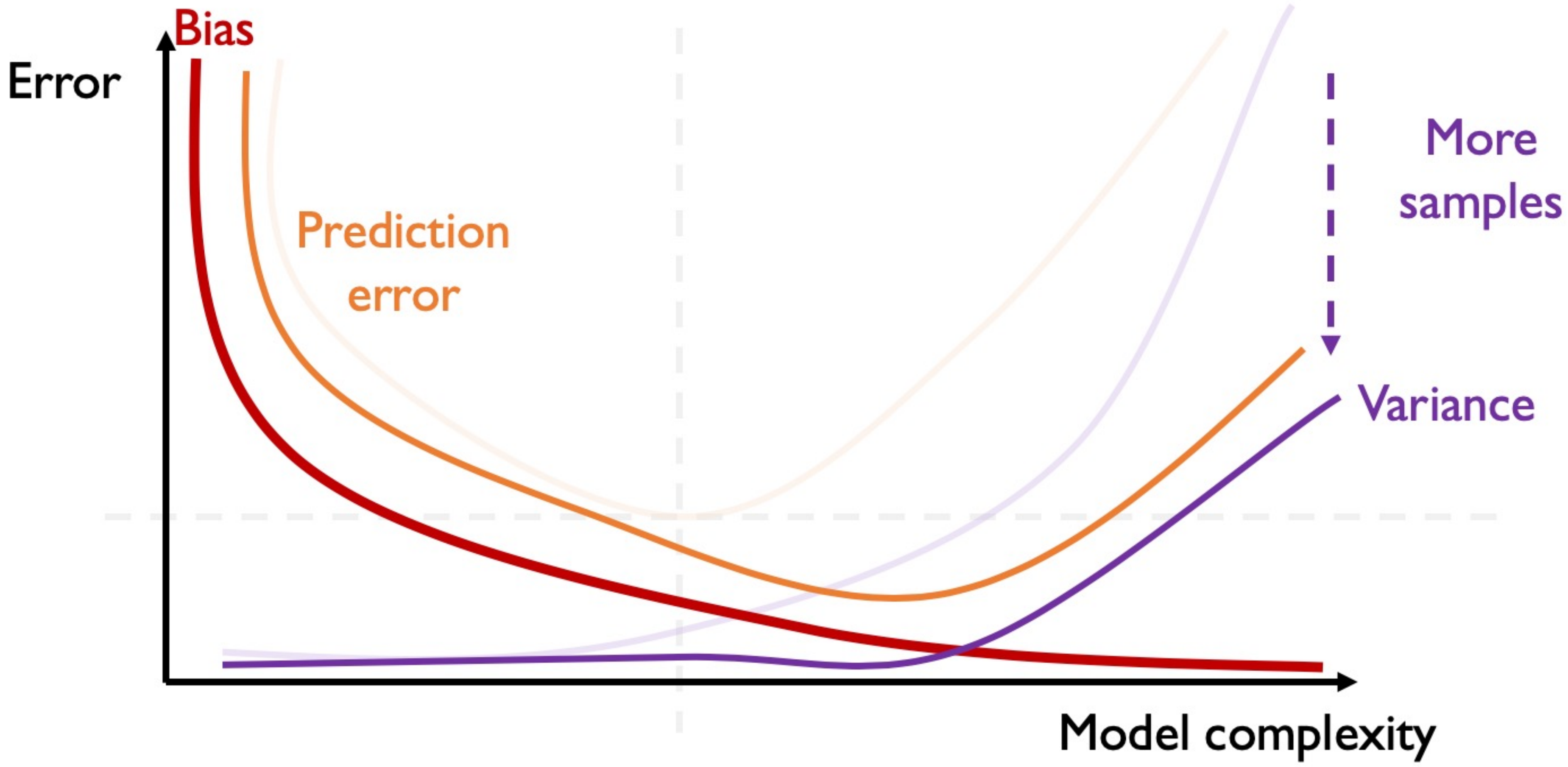
How is the bias - variance tradeoff of Deep NNs?



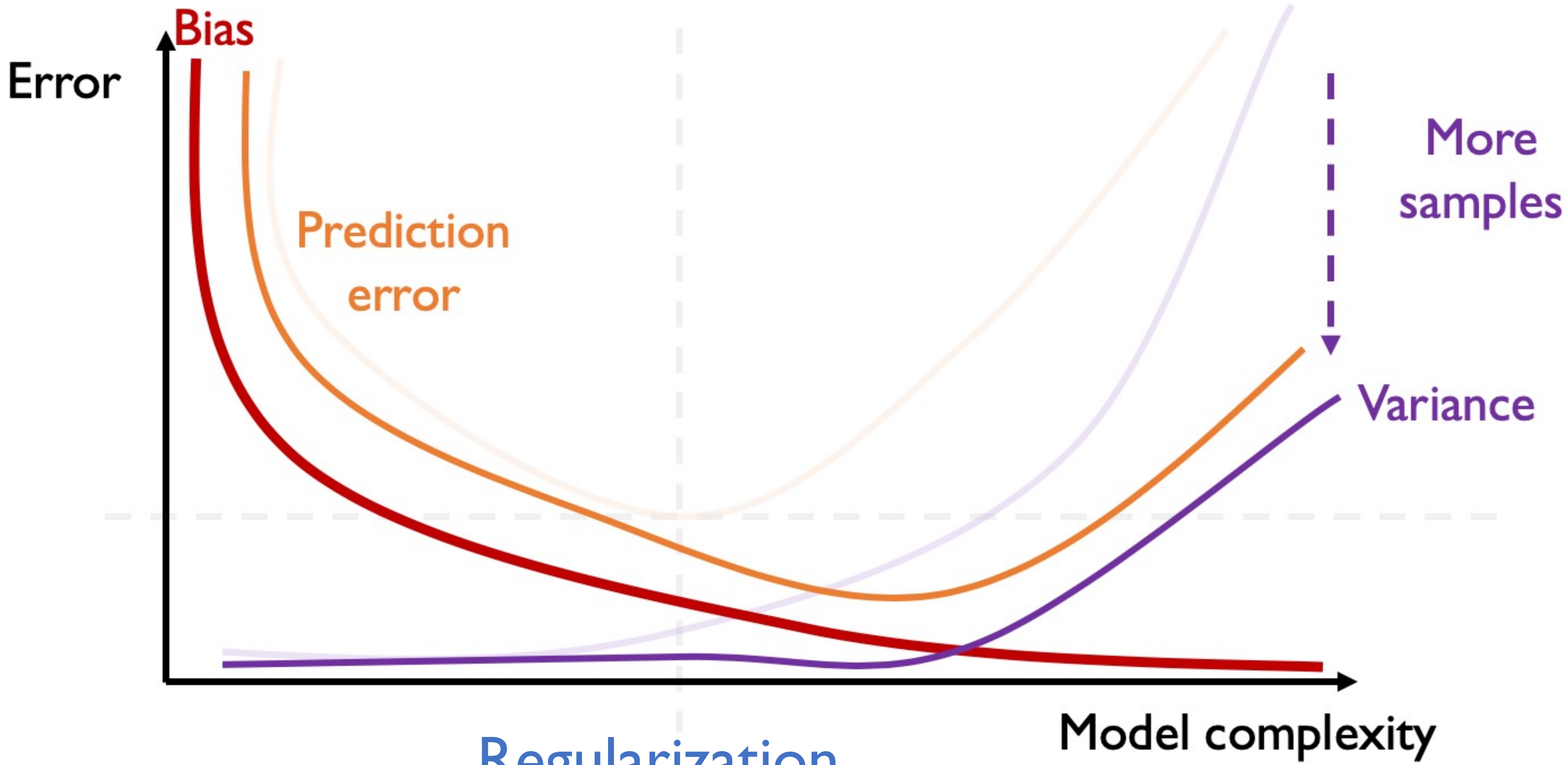
Why Deep NNs work: 1)Large Sample Size



Why Deep NNs work: 1)Large Sample Size

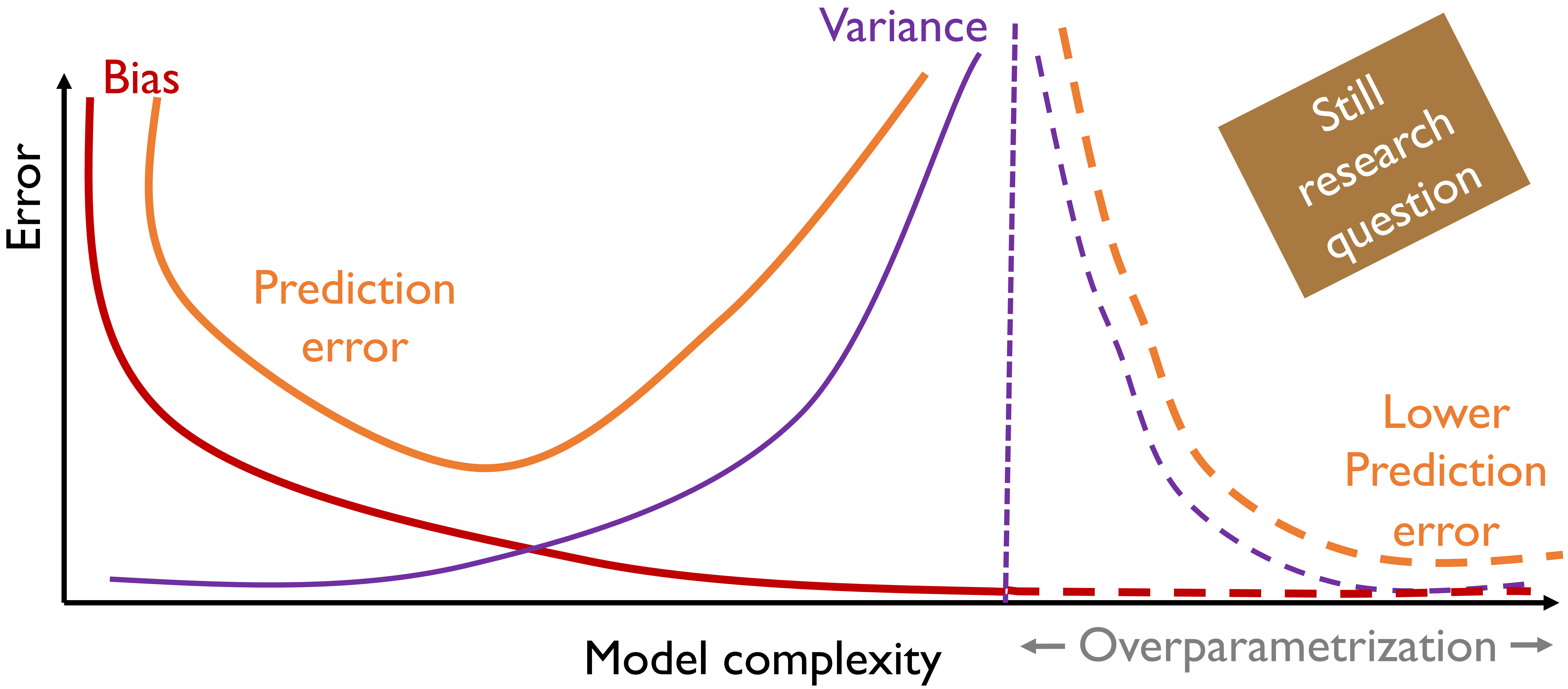


Why Deep NNs work: 2)Regularization

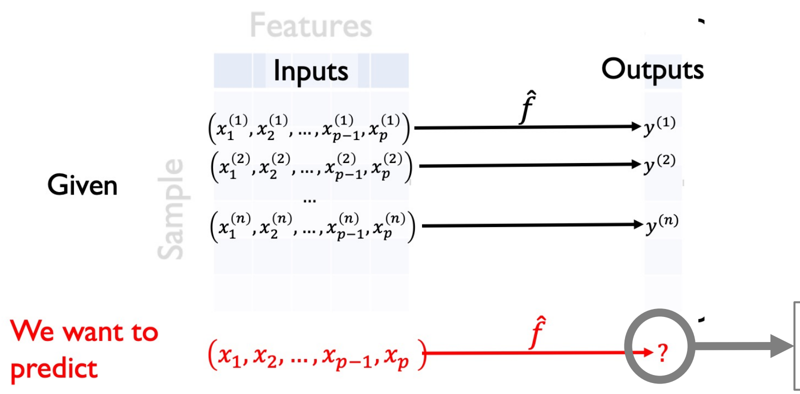


Regularization
Fixed Architecture

Why Deep NNs work: 3)Overparametrization



Why Deep NNs are challenging: Training



Train weights: $$\underset{a, W_1, \dots, W_l}{\text{minimize}} \sum_{i=1}^{N} L\left(y^{(i)}, f\left(x^{(i)}\right)\right)$$

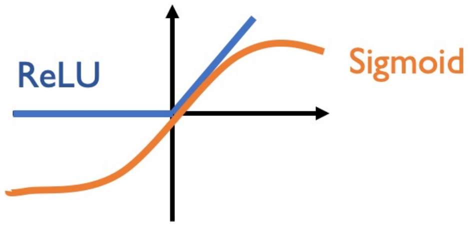
$$\boxed{y \approx f(x) = h(a^T z_l), z_l = g\left(W_l^T z_{l-1}\right), \dots, z_1 = g(W_1^T x)}$$

Choose Hyperparameters

Layers: Deep vs. Shallow

Neurons(units) per layer: Width

Activation Functions $(g(\cdot), h(\cdot))$:

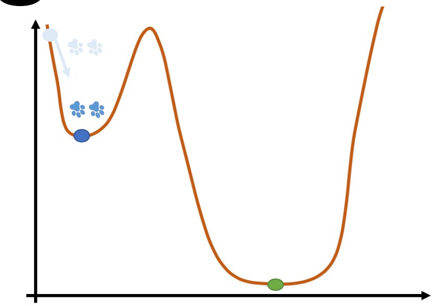


Why Deep NNs are challenging: Training

Train weights:

$$\underset{a, W_1, \ldots, W_l}{\text{minimize}} \sum_{i=1}^{N} L\left(y^{(i)}, f\left(x^{(i)}\right)\right)$$

1) Non-Convex Problem: Use Gradient Descent



2) Large Sample Size: Use Stochastic Gradient Descent

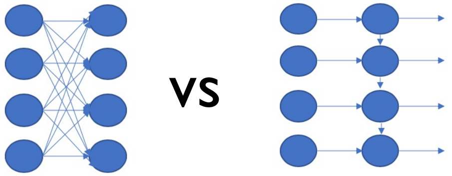
$$\gamma \sum_{k \in data} \nabla_W L(y_k, f(x_k))$$

$$\approx E[\nabla_W L(Y, f(X; W_i))]$$

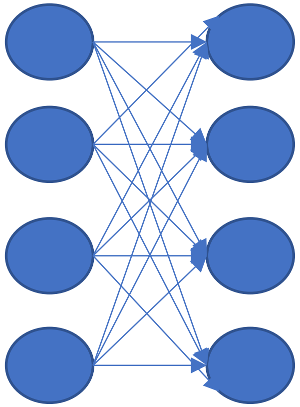
3) Composition of Functions: Back propagation = Chain Rule for derivatives

$$f(x) = h(a^T z_l),\ z_l = \ g\left(W_l^T z_{l-1}\right), \ldots,$$

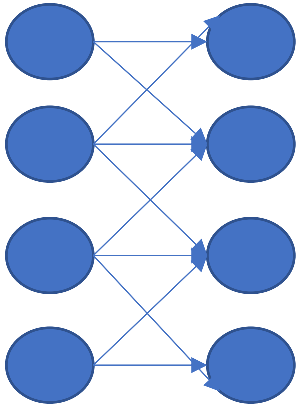
4) Regularization: NN Architecture = Sparsity of weights



Typical NN architectures

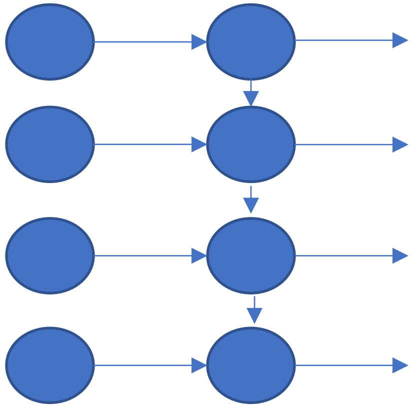


Dense



CNN
Spatial data
(Images)

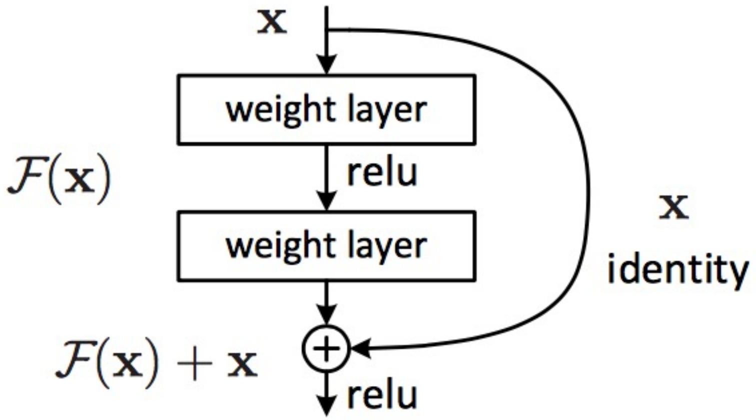
Only connection to neighbors

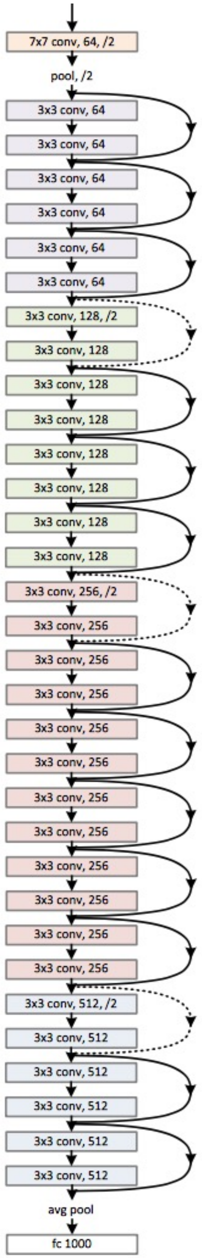


RNN
Sequential Data
(Text)

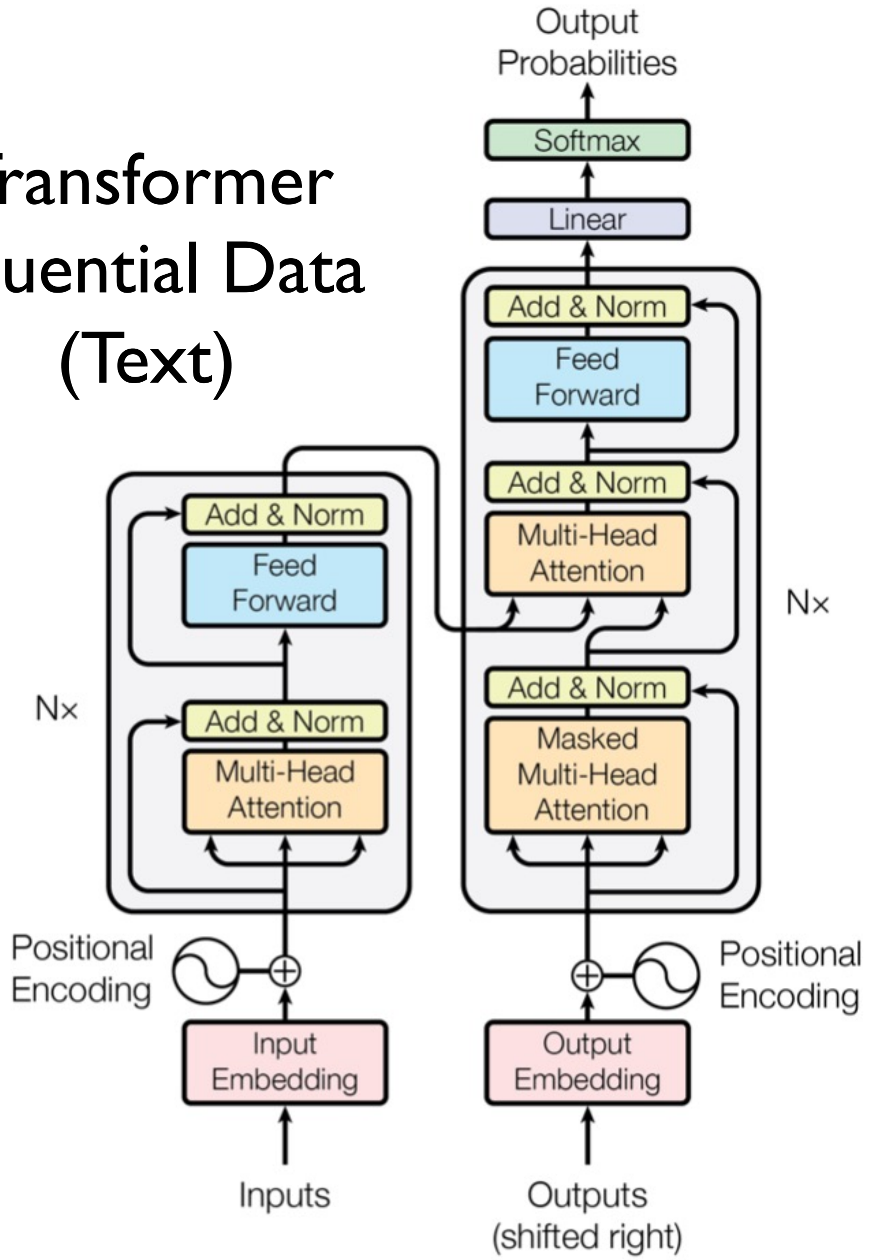
Memory from previous features

ResNet
Image Processing





Transformer
Sequential Data
(Text)



Deep Residual Learning for Image Recognition. He et al. CVPR 2016

Attention is All you Need. Vaswani et al. NeurIPS 2017

Final thoughts

How were these architectures found?

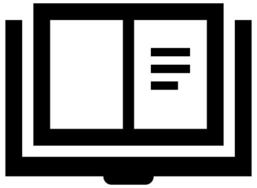
more general …

Why are ML methods so successful?

What happened with theory-based models?

Theory vs Learning from Examples

Theory



Theorems to describe what works best given assumptions

Assumptions are restrictive

Still developing theory (deep learning)

Examples

Try what works for others

Generalization?

Explainability?

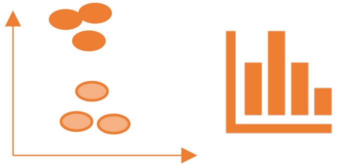
Optimality?

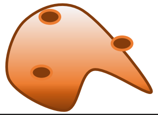
Why learning from examples has worked

Experience

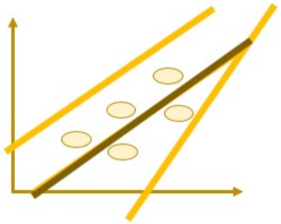
Large amounts of data available

Data Exploration



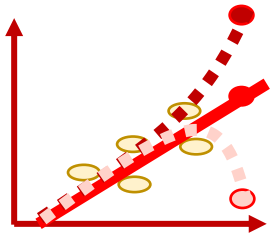


Prediction Models



$y = a\,x + b$

Performance Analysis



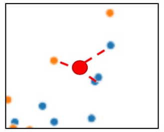
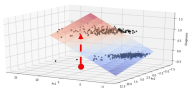
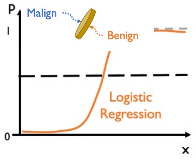
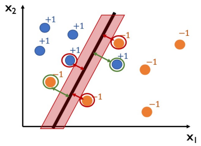
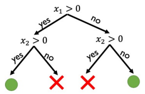
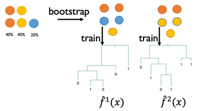
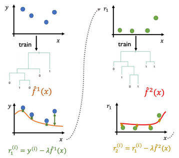
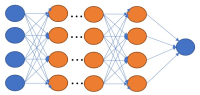
1) Efficient Parallel Hardware + Tools
GPUs, Spark, …

2) Unified Software
sklearn, TensorFlow, …

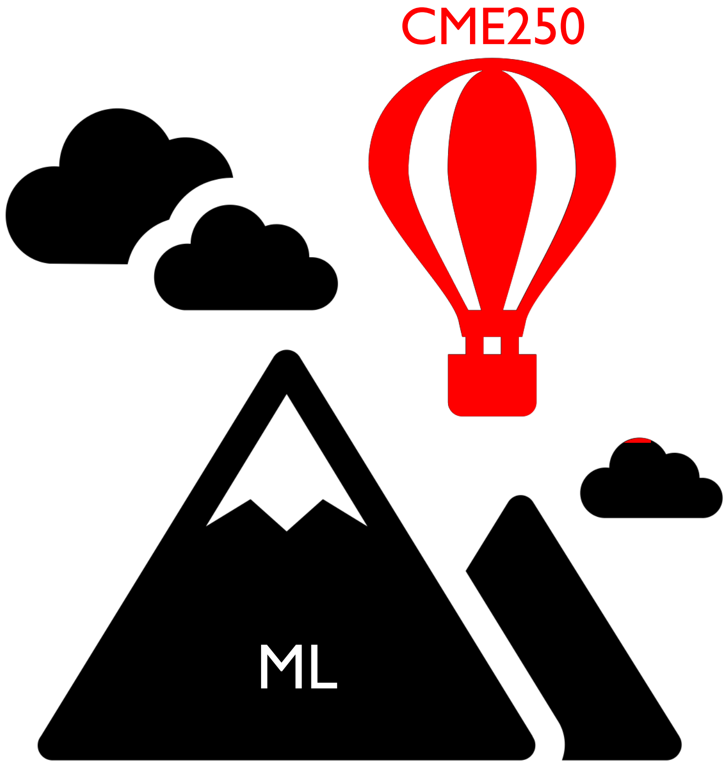
Focus on design rather than technicalities

Task

Openly available benchmarks
Kaggle, …

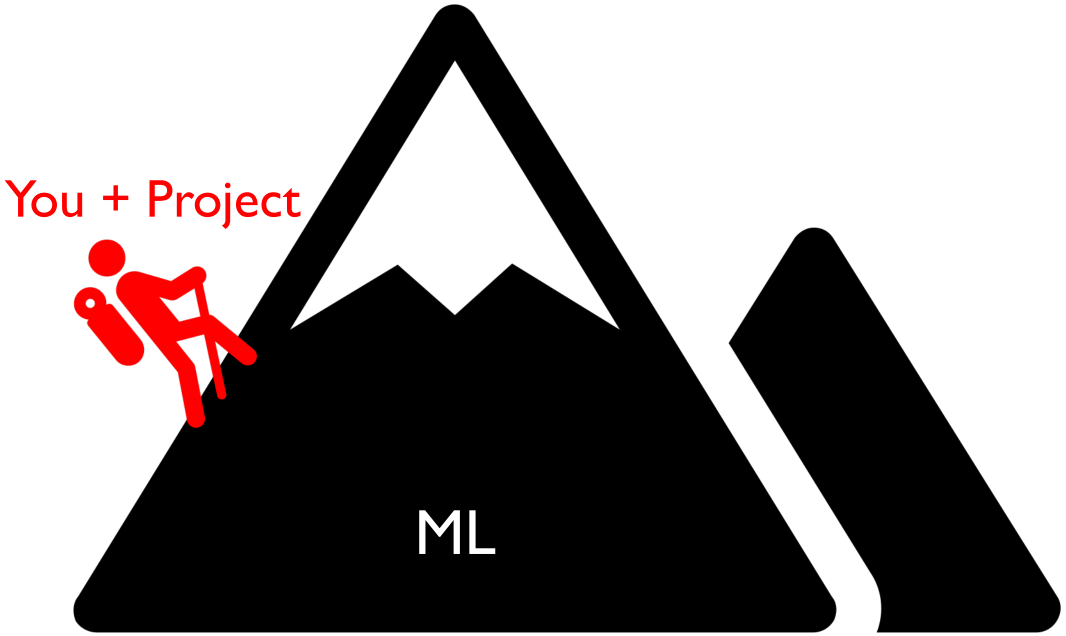
	Regression Y is quantitative	Classification Y is categorical	Interpretability	Flexibility Non-linear boundary	Tuning # Hyperparameters	Training Time
 KNN	✓	✓	✗	✓	😐 #neighbors, Distance	✓
 Linear Regression	✓	😐 Dummy variables	✓	😐 Create additional Features	✓ Features, Regularization	✓
 Logistic Regression	✗	✓	✓	😐	✓	✓
 SVM	✗	✓	✗	✓	😐 Kernel, Regularization	✓
 CART	😐 Overfitting vs pruning	😐	✓	✓	✓ Tree depth	✓
 Random Forest	✓	✓	😐	✓	😐 Tree depth, # trees, # features, learning rate	😐
 Gradient Boosting Trees	✓	✓	😐	✓	😐	😐
 Neural Networks	✓	✓	😐 - ✗	✓	✗ Many hyperparameters, Flexible architecture	✗

What was CME250?



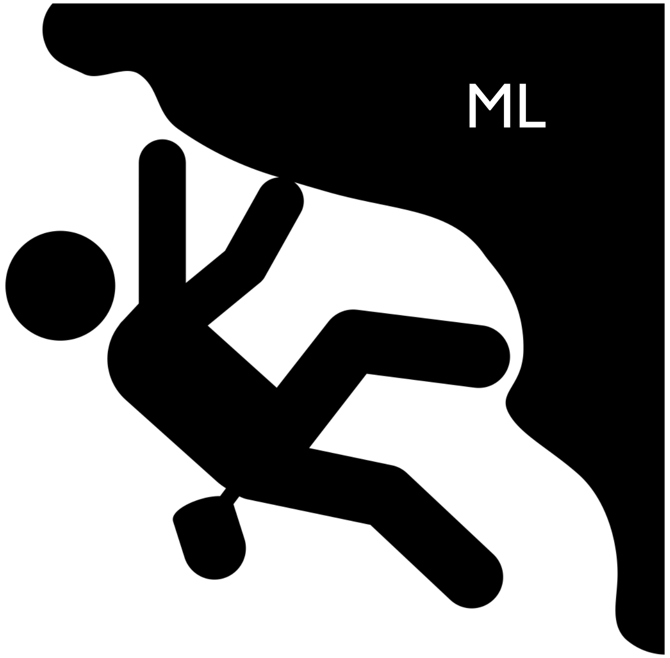
Terminology, Models
Best Practices.

+



Project

What's next?



Mathematical proofs.
Implementation tricks.

Introduction

- CME 250: Introduction to Machine Learning
- CS 229A: Applied Machine Learning

Foundations

- CS 229: Machine Learning
- CS 221: Artificial Intelligence
- CS 230: Deep Learning
- STATS 315A/B: Modern Applied Statistics
- CS 234: Reinforcement Learning

Theory

- CS 229T: Statistical Learning Theory

Applications

- CS 224N: Natural Language Processing with Deep Learning
- CS 325B: Data for Sustainable Development
- CS 231N: Convolutional Neural Networks for Visual Recognition
- CS 273B: Deep Learning in Genomics and Biomedicine
- CS 246: Mining Massive Data Sets
- …and much more

\+ Extensive amount of online courses, blogs, resources

\+ Practice, practice, practice …

Thank you!