outrageously
AMBITIOUS

AIPI 520: Modeling Process & Algorithms

Duke
PRATT SCHOOL *of* ENGINEERING

Context

“It is **easy** to sit in your office and run an algorithm on a data set you downloaded from the web.

It is **very hard** to identify a problem for which machine learning may offer a solution, determine what data should be collected, select or extract relevant features, chose an appropriate learning method, select an evaluation method, interpret the results, involve domain experts, publicize the results to the relevant scientific community, persuade users to adopt the technique, and (only then) to truly have made a difference.”

- *Wagstaff, “Machine Learning that Matters”*

Duke
PRATT SCHOOL *of* ENGINEERING

What we will learn

- Modeling process best practices
  - Modeling, evaluation, interpretation
- Key algorithms – theory, programming with libraries, programming from scratch
- Applying ML to real-world messy data (mostly tabular)

Course organization & content

Week	Topics	Deliverable Due
Week 1	Introduction, ML System Design	
Week 2	Bias-Variance Tradeoff, Evaluating Performance	
Week 3	Linear Regression, Polynomial Regression Regularization	Assignment 1 due
Week 4	Logistic Regression, Gradient Descent	
Week 5	Support Vector Machines, KNN	
Week 6	Midterm exam (No class)	Assignment 2 due
Week 7	Neural networks 1	
Week 8	Neural Networks 2	Assignment 3 due
Week 9	Ensemble Models, Trees and Boosting	
Week 10	Clustering	Assignment 4 due, Kaggle due
Week 11	Dimensionality Reduction, Embeddings	
Week 12	Interpretable ML	Assignment 5 due
Week 13	Final exam released	Project due

Duke
PRATT SCHOOL of ENGINEERING

Grading

- **Assignments (5):**
  - Whiteboard level collaboration
    - Must write your own code and written responses
    - Cannot share your answers with anyone
  - Open-book, open-internet
  - All writing and code MUST be your own (No ChatGPT or copying repos)
  - **Due before start of next lecture. No late submissions**
- **Weekly Knowledge Check Quizzes:**
  - Timed, Closed-book, closed-internet
  - Can drop the lowest score

Component	%
Assignments	25
Midterm exam	15
Final exam	20
Kaggle competition	20
Project	15
Classwise activities	5

CLASSWISE

PRATT SCHOOL of ENGINEERING

outrageously
AMBITIOUS

Introduction to Machine Learning

Duke
PRATT SCHOOL *of* ENGINEERING

What is Machine Learning?

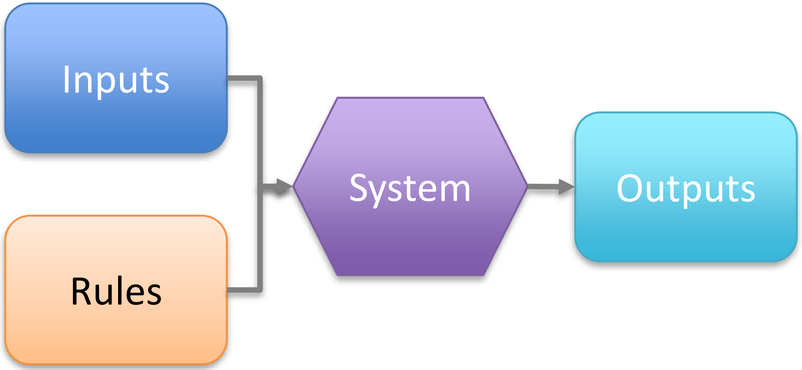
- “Field of study that gives computers the ability to learn **without being explicitly programmed**” – Arthur Samuel, IBM, 1959

- Learning is usually defined as “gaining skill or knowledge through experience”

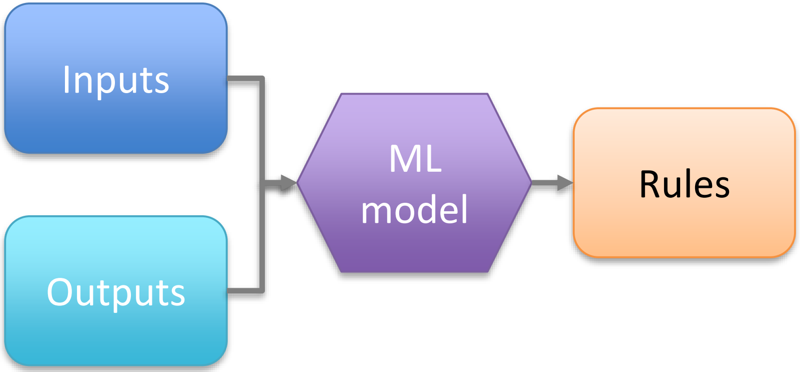
- Two different types of learning:
  - Task-based – build expertise at a task
  - Generalization – transfer of learning from narrow situations to broader ones

ML vs. traditional software

<u>How traditional software generates predictions</u>



<u>How machine learning generates predictions</u>



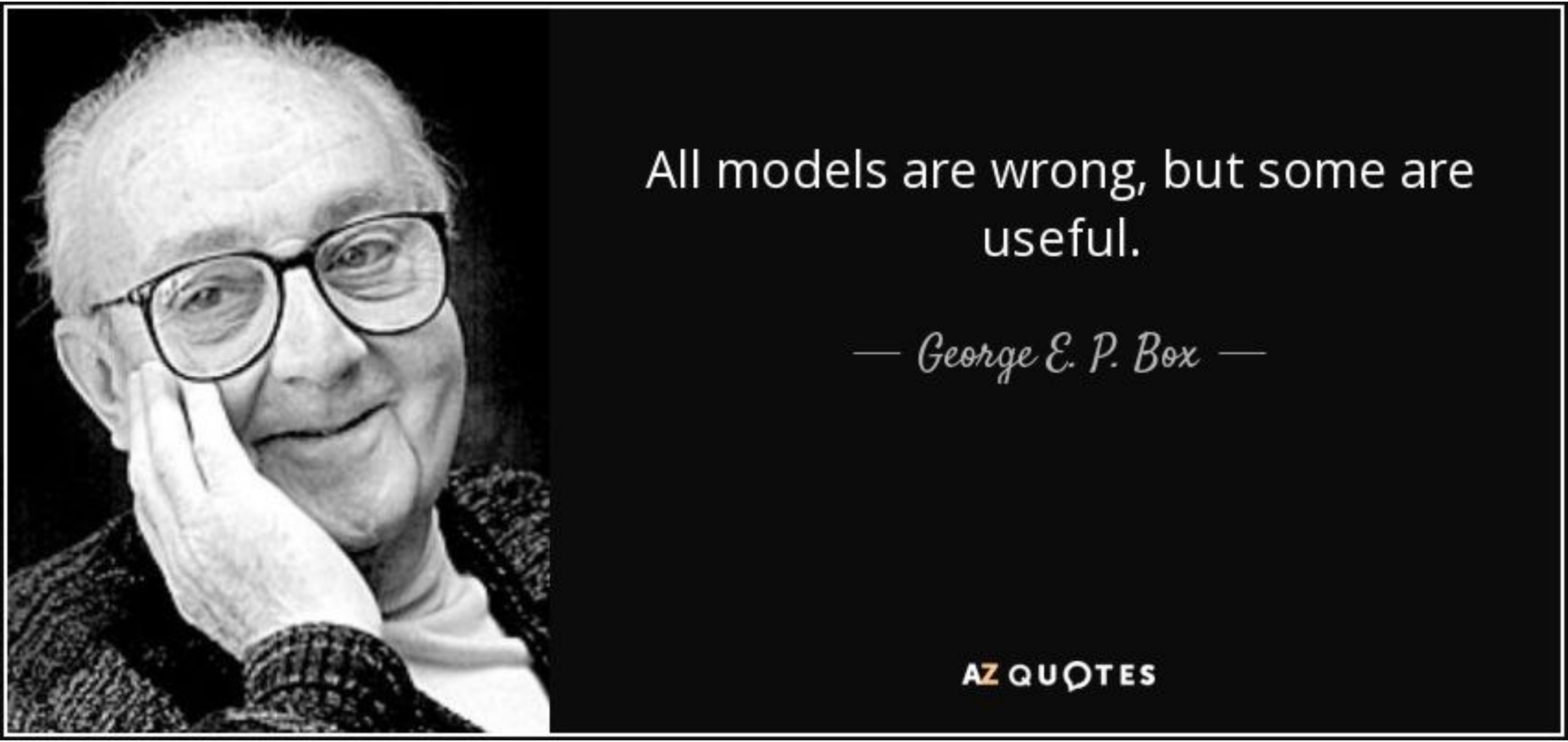
Why do we need ML?

- There is some benefit to using computers to perform tasks
- It is unfeasible to build a set of rules for each task
  - Specifying the rules may be too complex
  - We may not know the rules

outrageously
AMBITIOUS

What is a Model?

Duke
PRATT SCHOOL of ENGINEERING


All models are wrong, but some are useful.
— George E. P. Box —
AZ QUOTES

What is a model?

A model is a **useful approximation** of a non-random phenomenon

Model y=f(x)+ε

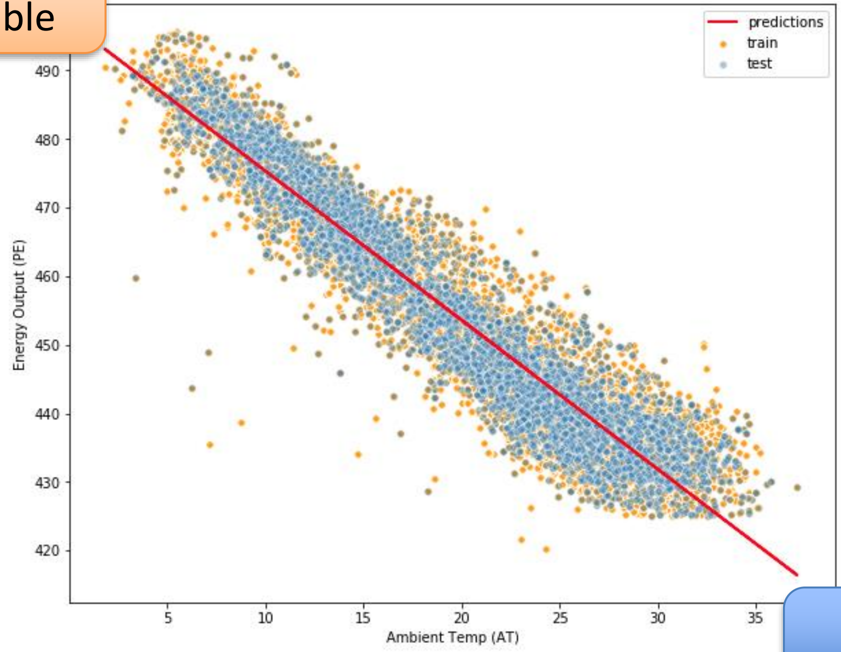
Input variable(s) x →  → Output variable(s) y

We might think of models as a method of compressing knowledge into a finite set of numbers (parameters)

What is a model?

$$\hat{y} = mx + b$$

Output variable



Input variable

Data for modeling

Targets / Labels / Annotations / Response / Y Variable / Dependent Variable

Features / Factor… / X Variables / Independent Varia… / Dimensions

p features
$j = 0,1,2\ldots(p-1)$

n observations
$i = 0,1,2\ldots(n-1)$

Observations / Instances / Examples / Feature Vectors

	Neighbor-hood 0	School district 1	Square footage 2	Number of bedrooms 3	Year built 4	Market sale price
House 1 0	Weycroft $x_{0,0}$	Wake $x_{0,1}$	3400	4	2010 $x_{0,4}$	$612,000 y_0
House 2 1	Horton Creek	Wake	4200	5	2008	$675,000
House 3 2	Cary Park	Chatham	3250	4	2012	$520,000 y_2
…	… $x_{n-1,0}$	…	…	…	…	…

Duke
PRATT SCHOOL of ENGINEERING

Why do we create models?

1. Prediction

- If we can produce a good estimate of the function f, we can reasonably estimate the output y's, even for input X's that the model has never seen before
- We don't care too much about the features of the input, we just want a good estimate of the output to make a decision

2. Insight

- Alternatively, we may be more interested in understanding the relationships between the input x's and output y's
- This is often the case in research, where we want to uncover hidden relationships and learn what affects the output

outrageously
AMBITIOUS

Types of Machine Learning

Duke

PRATT SCHOOL of ENGINEERING

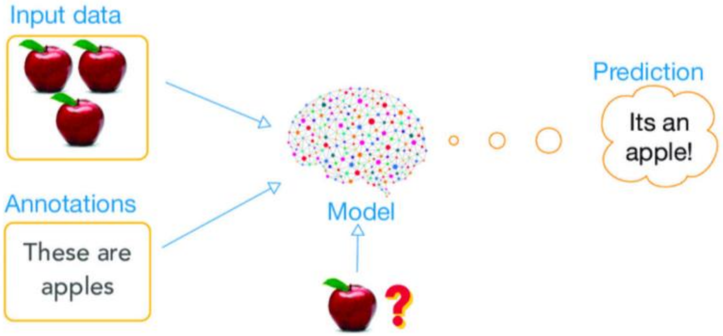
Types of machine learning

	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Objective	Prediction of a target variable	Organize data by inherent structure	Learn strategies via interaction
Learning Task(s)	Classification Regression	Clustering Anomaly detection	Achieve a goal
Target Data Required?	Yes	No	Yes, but delayed
Examples	• Identifying pneumonia from xray images • Predicting real estate prices	• Market segmentation • Identifying fraudulent activity	• AlphaGo • Autonomous vehicles • GPT-4 RLHF

Supervised vs. unsupervised learning

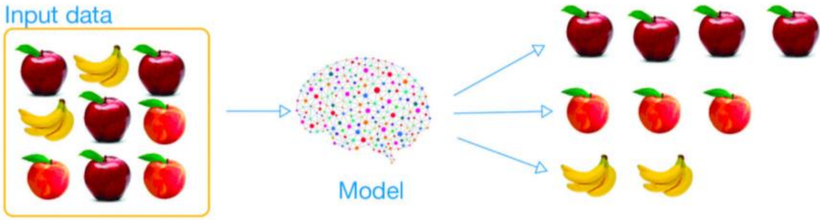
Supervised learning

At least some observations of the features (X_i) and targets (Y_i) are known and used to build a model



Unsupervised learning

We only have observations of the features (X_i). We need to use the observations to guess what the targets (Y_i) would have been and build a model from there

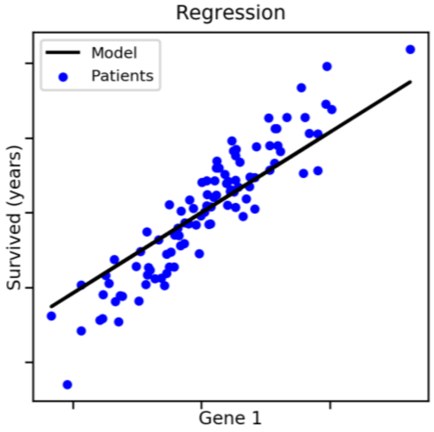


https://www.researchgate.net/figure/Supervised-learning-and-unsupervised-learning-Supervised-learning-uses-annotation_fig1_329533120

Supervised learning types

Regression

- Predict one or more **numerical** target variables
- E.g. home price, number of power outages, product demand



Classification

- Predicts a **class / category** – either binary or out of a set
- E.g. lung disease detection, identifying types of plants, sentiment analysis, detecting spam

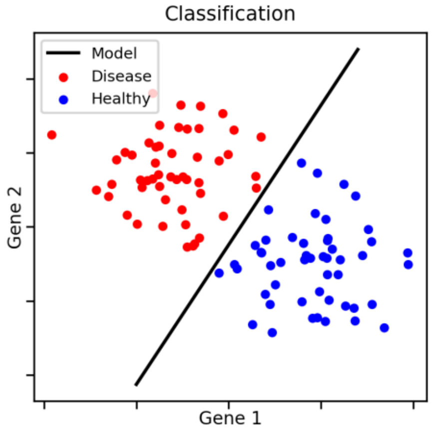
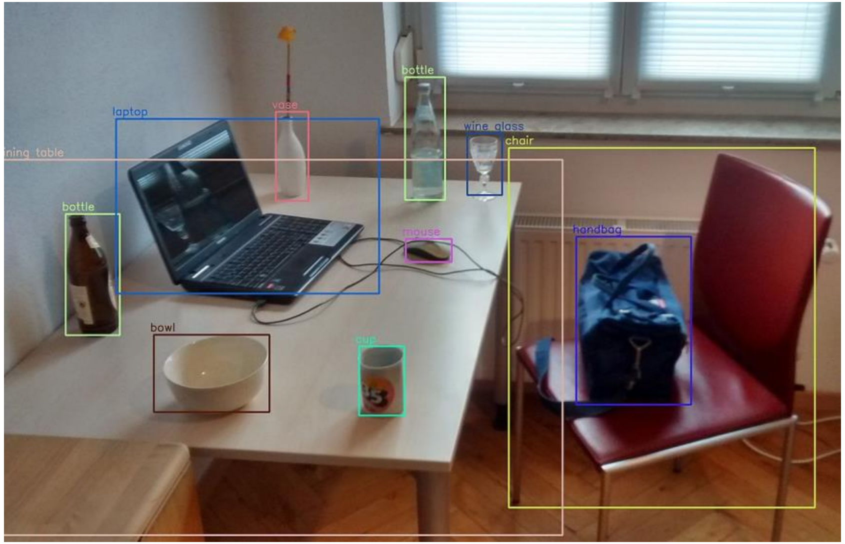


Image source: https://aldro61.github.io/microbiome-summer-school-2017/sections/basics/

Practice: Regression or classification?

- Detecting “fake news”
- Predicting a stock price
- Determining a new driver’s risk level for insurance
- Finding a certain object in camera images – e.g. a self-driving car identifying a pedestrian

Object detection



Input

Prediction

outrageously
AMBITIOUS

When to use ML

Duke
PRATT SCHOOL *of* ENGINEERING

What ML can do well*

- Automate straightforward tasks
- Make predictions by learning input-output relationships
- Personalize for individual users

* Given sufficient quantity and quality of data

What ML cannot do well

- Understand context
- Determine causation from correlation
- Explain “why” things happen
- Find solutions to problems

When to use ML

- ✓ Data is available for training
- ✓ There are patterns to learn (events are not completely random)
- ✓ There is value in getting predictions
- ✓ Predictions are needed at scale
- ✓ The cost of mistakes is low

Alternative: heuristics

- Methods of solving problems using a simplified set of rules based on past experience
- Hard-coded business rules rather than machine learning
- Examples:
  - Demand prediction: Predicting the mean value for sub-groups
  - Product classifier: Classifying based on title (e.g. Walmart)
  - Product recommender: recommend the highest rated
  - App recommender: recommend the most popular (# installs)

Duke
PRATT SCHOOL of ENGINEERING

Heuristics versus ML

Benefits of Heuristics

- Easier to create and maintain
- Minimal computational cost
- High interpretability

Benefits of ML

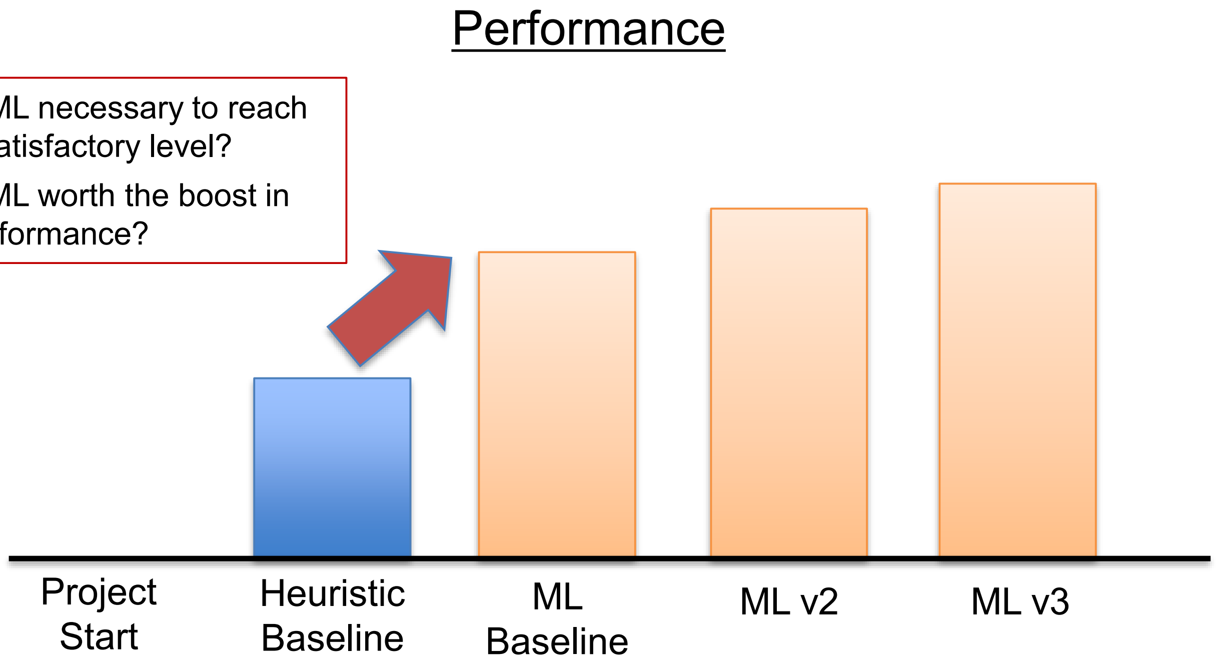
- Often better performing
- Can evolve with re-training
- Suitable for a wider range of problems (e.g. big data, computer vision)

Example: using heuristics

- How might we predict the daily sales at the University Store using heuristics, rather than building a model?

Establishing a baseline

- Is ML necessary to reach a satisfactory level?
- Is ML worth the boost in performance?



Heuristics as a baseline

Question:

"Imagine you're given a new, unfamiliar problem to solve with machine learning. How would you approach it?"

> I'd first try really hard to see if I could solve it without machine learning :D. I'm all about trying the less glamorous, easy stuff first before moving on to any more complicated solutions. — *Vicki Boykis, ML Engineer @ Tumblr*

> I think it's important to do it without ML first. Solve the problem manually, or with heuristics. This way, it will force you to become intimately familiar with the problem and the data, which is the most important first step. Furthermore, arriving at a non-ML baseline is important in keeping yourself honest. — *Hamel Hussain, Staff ML Engineer @ Github*

> First, try to solve it without machine learning. Everybody gives this advice, because it's good. You can write some if/else rules or heuristics that make some simple decisions and take actions as a result. — *Adam Laiacano, Staff Eng (ML platform) @ Spotify*

https://eugeneyan.com/writing/first-rule-of-ml/

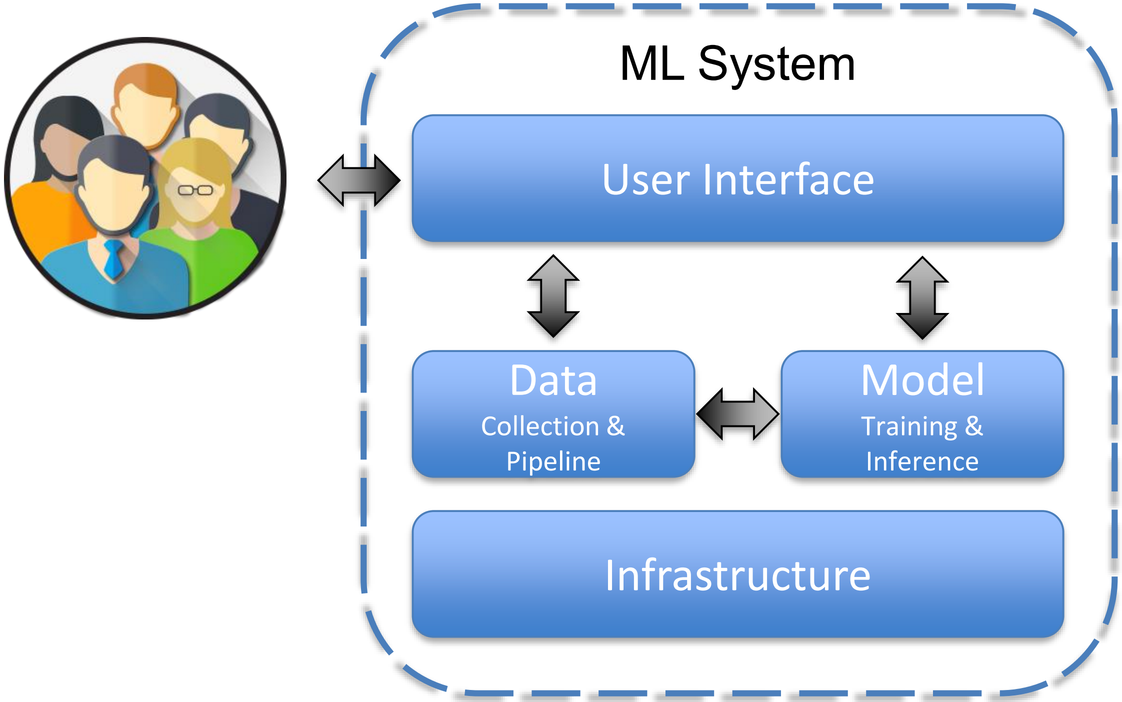
Duke
PRATT SCHOOL of ENGINEERING

outrageously
AMBITIOUS

Types of ML Systems

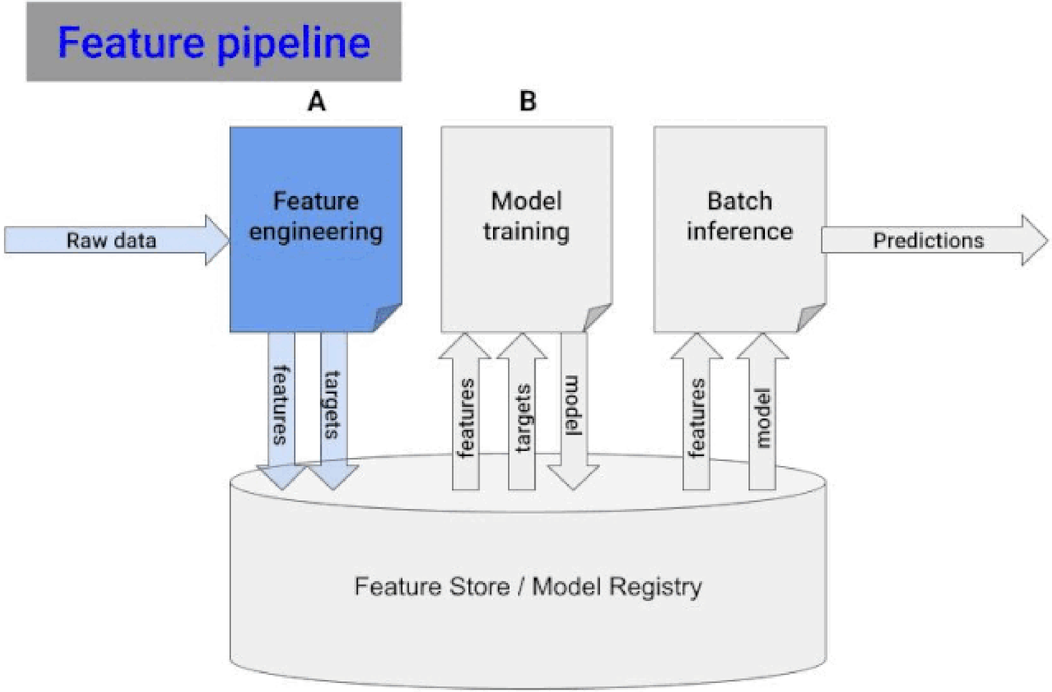
Duke
PRATT SCHOOL *of* ENGINEERING

What is a ML system?



Duke
PRATT SCHOOL of
ENGINEERING

Traditional ML system



ML system design decisions

There are several system design decisions which impact the choice of technologies:

Cloud	vs.	Edge
Offline Learning	vs.	Online Learning
Batch Predictions	vs.	Online Predictions

Duke
PRATT SCHOOL of ENGINEERING

System design process

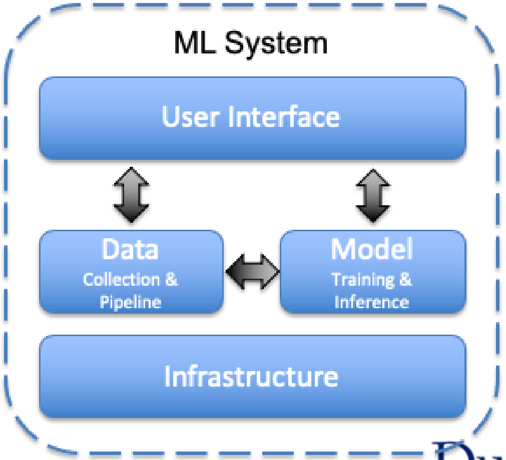
- User requirements and constraints drive system design
- System design drives selection of technologies

User requirements & constraints

System design decisions

Cloud	vs.	Edge
Offline Learning	vs.	Online Learning
Batch Predictions	vs.	Online Predictions

Technology decisions



Duke
PRATT SCHOOL of ENGINEERING

System design examples

	Use Case	Requirements / Constraints	System type
Example 1	Unlocking a phone using a facial recognition model	• Low latency • Cannot require connectivity • Privacy concerns	Cloud vs. Edge Offline Learning vs. Online Learning Batch Predictions vs. Online Predictions
Example 2	Movie recommendation engine	• Can assume connectivity • High computational needs • Low latency	Cloud vs. Edge Offline Learning vs. Online Learning Batch Predictions vs. Online Predictions

Cloud vs. Edge

Edge AI

- Running ML on devices themselves
- Predicted to grow by 25% per year[1]
- Enabled by hardware & software advances
  - Solving power constraints
  - Increased edge computational power
  - Smaller models designed for the edge
- Why edge AI? Eliminates latency & improves privacy
  - Every 100 miles distance from datacenter introduces a latency of > 1.6 milliseconds[2]

1) *Valuates, https://reports.valuates.com/market-reports/QYRE-Auto-4139/global-edge-ai-software*
2) *Techwalla, https://www.techwalla.com/articles/network-latency-milliseconds-per-mile*

Duke
PRATT SCHOOL *of* ENGINEERING

Cloud vs. edge

	Cloud ML	Edge ML
Description	Computations done on cloud and result delivered to end device	Computations done directly on device (phone, sensor, etc)
Requirements	Network connectivity	Sufficient compute power, memory
Benefits	Easier, can use larger models & efficient compute	Low latency, privacy, no need for connectivity, reduce cloud costs
Examples	• Chatbots • Demand prediction	• Quality control • Autonomous driving

Cloud ML example

Movie recommendation system

Edge ML example

Intelligent security system

Duke PRATT SCHOOL of ENGINEERING

Hybrid approaches

- Initiate cloud ML through trigger generated by edge AI
- Store common pre-computed predictions on device
- Exploit many local datacenters/servers to minimize latency

Hybrid example

Smart speaker with voice assistant

Cloud vs. edge AI

- How much does latency matter?
  - A lot -> Edge
  - Not much -> Cloud
- Is reliance on internet connectivity acceptable?
  - No -> Edge
  - Yes -> Cloud
- Are users comfortable sending their data to the cloud?
  - No -> Edge
  - Yes -> Cloud

outraggeously
AMBITIOUS

Online Learning & Inference

Duke
PRATT SCHOOL *of*
ENGINEERING

Offline vs. online models

An important design consideration is whether model training & prediction can be scheduled or must be real-time

	Scheduled	Real-time
Model re-training	Offline learning	Online learning
Prediction	Batch prediction	Online prediction

Duke PRATT SCHOOL *of* ENGINEERING

Offline vs. online learning

	Offline learning	Online learning
Description	Model re-training done on a schedule (weeks/months) using datapoints in many iterations	Continual re-training as new data arrives (mins/hours) using each new datapoint once
Benefits	• Easier to implement in production • Easier to evaluate	• Handles big data • Real-time adaptation to changing environment
Challenges	• Slower to adapt to changes in environment or data distribution	• Harder to implement & evaluate performance
Examples	• Most current applications	• Flagging misinformation in social media

Duke
PRATT SCHOOL of ENGINEERING

Online learning example

News site with personalized recommendations

Duke PRATT SCHOOL of ENGINEERING

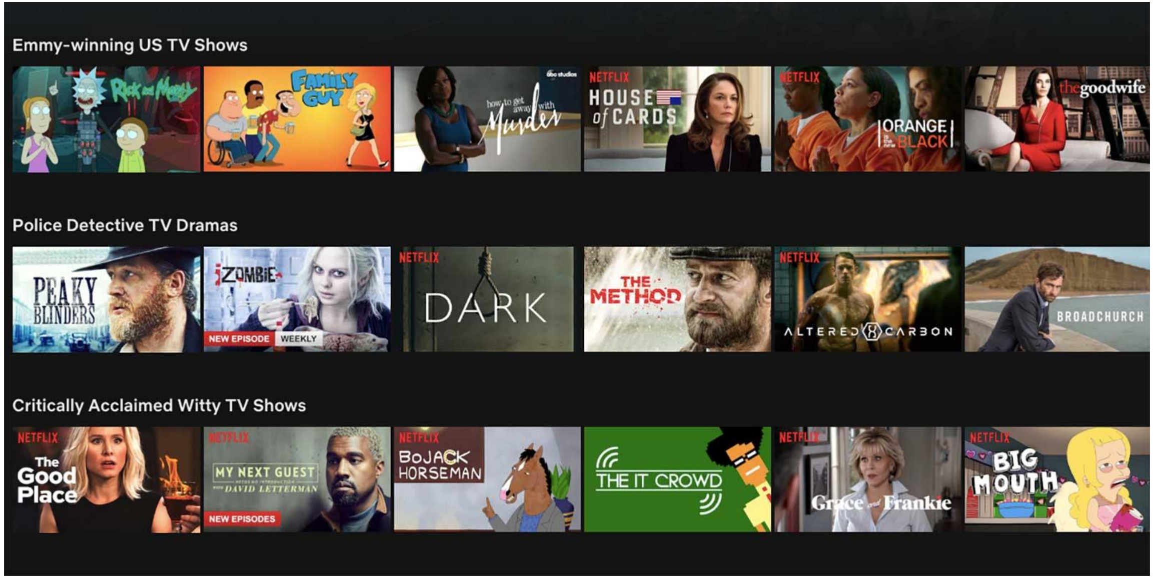
Batch vs. online prediction

	Batch prediction (asynchronous)	Online prediction (synchronous)
Description	Generate predictions on batch of observations on a recurring schedule	Real-time predictions generated upon request
Benefits	• Leverage more efficient operations and technologies • Easier monitoring of drift	• Predictions available immediately
Challenges	• Predictions not immediately available for new data	• Minimizing latency • Monitoring of model drift
Examples	• Recommendation systems • Demand prediction	• Translation app • Autonomous vehicles

Duke
PRATT SCHOOL of ENGINEERING

Batch prediction example

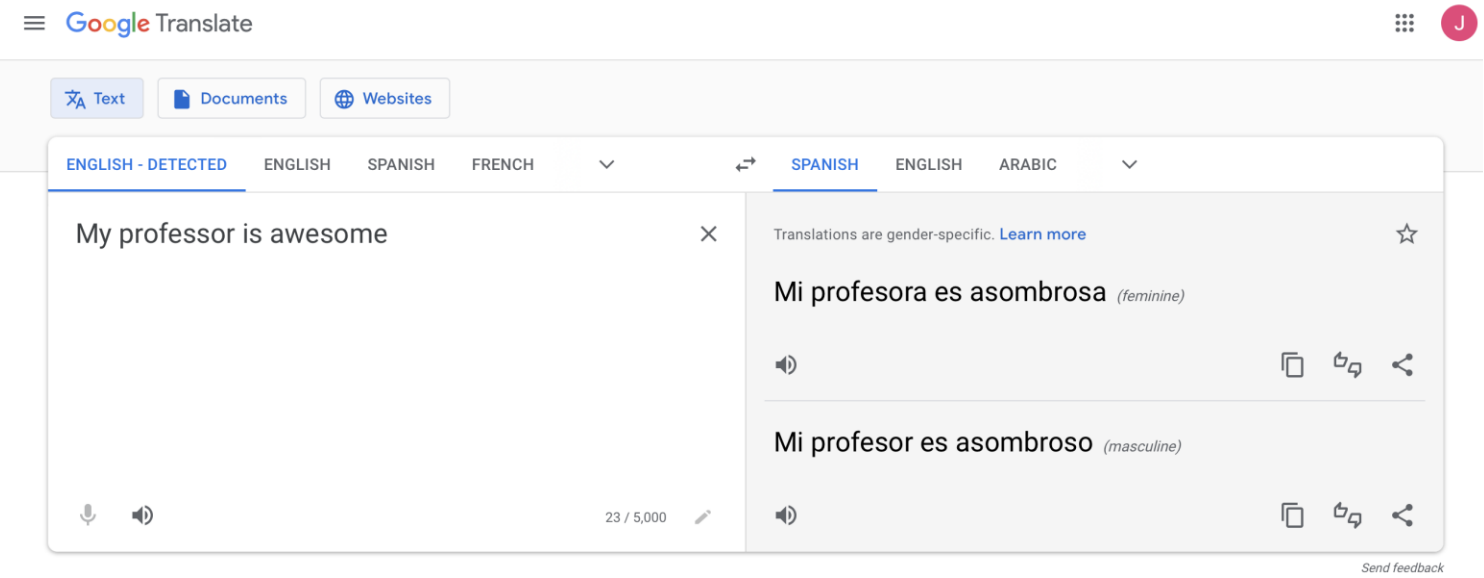
Movie recommendations



Duke
PRATT SCHOOL of ENGINEERING

Online prediction example

Machine translation



Duke
PRATT SCHOOL of ENGINEERING

Introduction to Scikit-Learn

What is Scikit-Learn?

- Python package that provides implementations of a large number of modeling algorithms
- Brief history
  - Started as a Google Summer of Code project by David Cournapeau in 2007
  - Researchers from the French Institute for Research in Computer Science & Automation took over the project and make first release in Feb 2010
  - Continues to be community-supported with funding from various tech companies
- Why is it so popular?
  - Simple API and good documentation
  - Interfaces well with NumPy and pandas (built on SciPy stack)
  - Open source and commercially usable

Scikit-Learn API

- Key characteristics:
  - ***Consistency*** – common interface with limited number of methods
  - ***Limited object hierarchy*** – only algorithms are represented by Python classes. Datasets are represented in arrays/dataframes and parameter names are strings
  - ***Inspection*** – all available parameter values are exposed as public attributes
  - ***Sensible defaults*** – when models require user-specified parameters, the library defines an appropriate default value

Duke PRATT SCHOOL of ENGINEERING

Scikit-Learn API step by step

1. Create features and targets and split the data
2. Select an algorithm
3. Choose model hyperparameters by instantiating the algorithm class
4. Train (fit) the model to your data using the **fit()** method
5. Apply your model to new data using the **predict()** method

DEMO: INTRO TO SCIKIT-LEARN

QUESTIONS?

Duke
PRATT SCHOOL of ENGINEERING

For next week

- Set up your working environment
- Read:
  - Machine learning that matters
  - The first rule of ML: start without ML
  - Google's Rules of ML
- Assignment 1 and Quiz 1 will be released next class