PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Introduction to Machine Learning

MLEARN 510A – Lesson 1

Course Logistics

Time:	Tuesdays 6 – 9pm (10/6/20 – 12/8/20)
Location:	Online
Instructor:	Aitzaz Ahmad (aitzaz@uw.edu)
TA:	Chenwei Lin (chenwl@uw.edu)
Canvas:	https://canvas.uw.edu/courses/1400625
Textbook:	*An Introduction to Statistical Learning (ISLR)*
	Data Mining: Concepts and Techniques, 3rd Edition
Lab:	Python and Jupyter Notebooks
Grading:	10 assignments (90%) and participation (10%)

About Me

- Applied Scientist – Machine Learning @ Amazon
  - Real-time Remittance Matching
  - Named Entity Recognition

- Senior Data Scientist @ Procter & Gamble
  - Deep Learning for
    - Real-time High-Speed Quality Inspection
    - Factory Floor Automation Using Robotics
  - Anomaly Detection and Forecasting

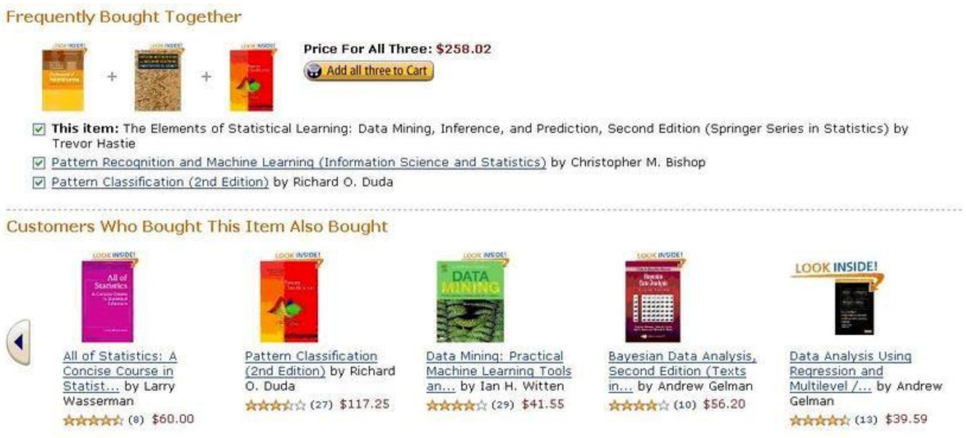
- Ph.D. in Electrical Engineering, Texas A&M University
  - Statistical Inference in Wireless Sensor Networks

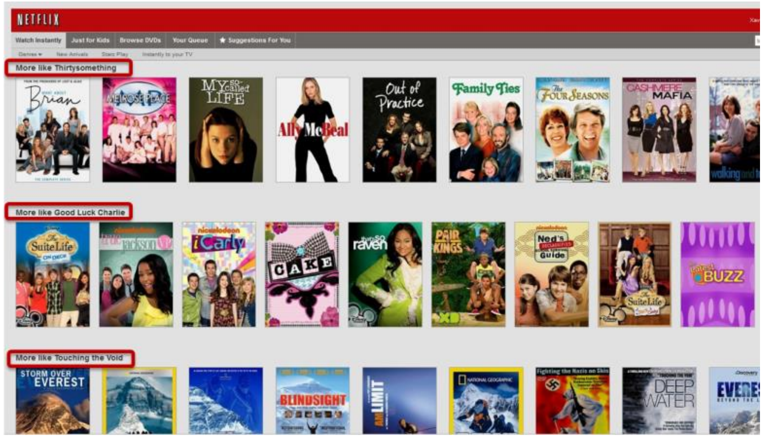
Considerations

- Remember to keep your sense of humor
- Focus on understanding each topic from a mathematical perspective
- Keep up with work every week
- Ask questions! If you have questions, other probably have the same questions

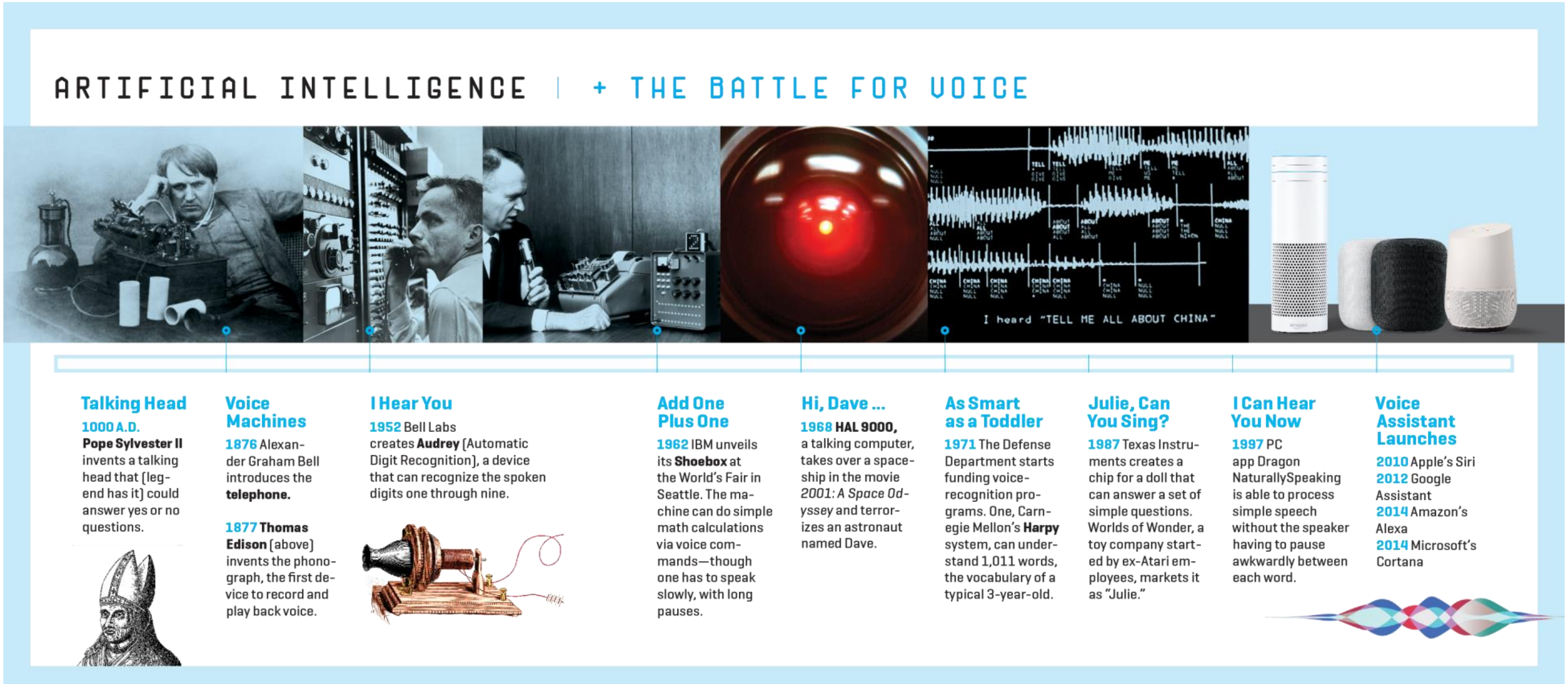
Recommendation Systems: The Battle for Personalization

"People don't know what they want until you show it to them" – Steve Jobs





Voice Recognition



ARTIFICIAL INTELLIGENCE | + THE BATTLE FOR VOICE

I heard "TELL ME ALL ABOUT CHINA"

Talking Head
1000 A.D. **Pope Sylvester II** invents a talking head that (legend has it) could answer yes or no questions.

Voice Machines
1876 Alexander Graham Bell introduces the **telephone.**

1877 **Thomas Edison** (above) invents the phonograph, the first device to record and play back voice.

I Hear You
1952 Bell Labs creates **Audrey** (Automatic Digit Recognition), a device that can recognize the spoken digits one through nine.

Add One Plus One
1962 IBM unveils its **Shoebox** at the World's Fair in Seattle. The machine can do simple math calculations via voice commands—though one has to speak slowly, with long pauses.

Hi, Dave ...
1968 **HAL 9000,** a talking computer, takes over a spaceship in the movie *2001: A Space Odyssey* and terrorizes an astronaut named Dave.

As Smart as a Toddler
1971 The Defense Department starts funding voice-recognition programs. One, Carnegie Mellon's **Harpy** system, can understand 1,011 words, the vocabulary of a typical 3-year-old.

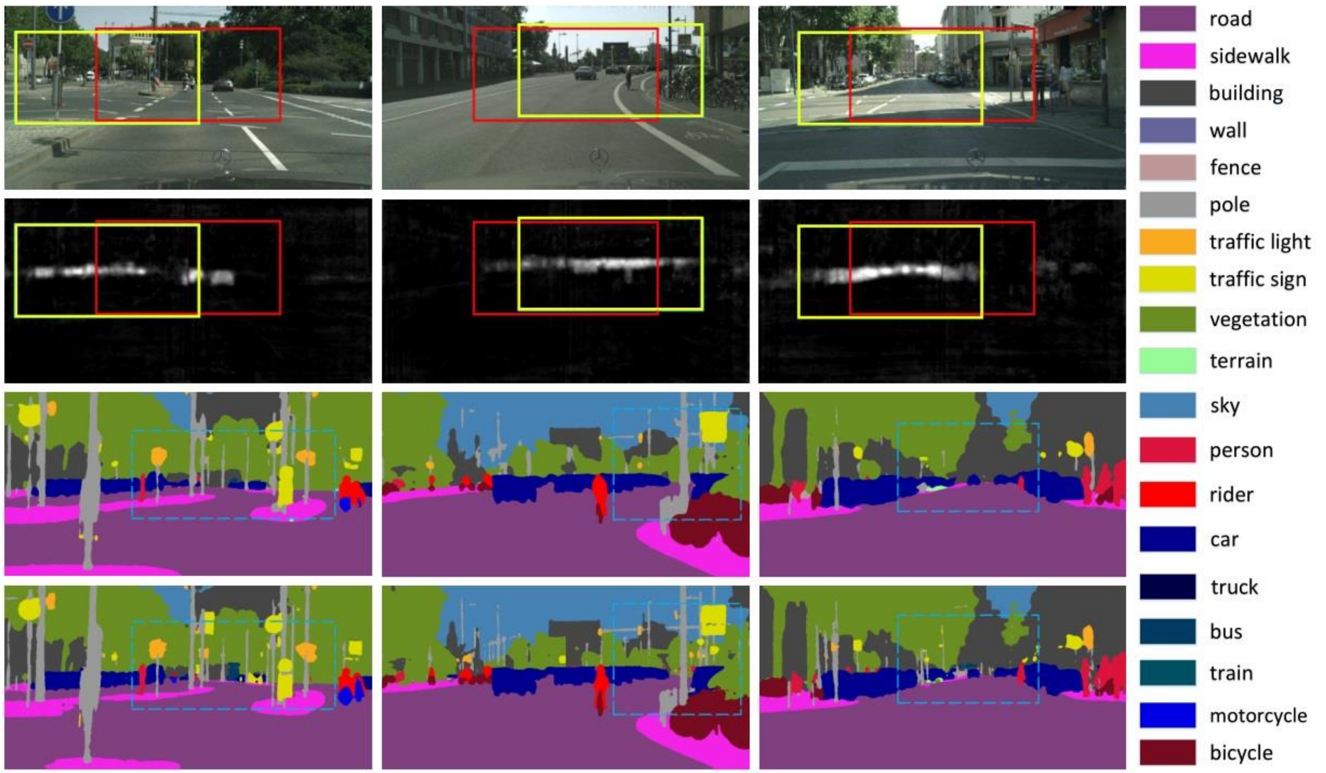
Julie, Can You Sing?
1987 Texas Instruments creates a chip for a doll that can answer a set of simple questions. Worlds of Wonder, a toy company started by ex-Atari employees, markets it as "Julie."

I Can Hear You Now
1997 PC app Dragon NaturallySpeaking is able to process simple speech without the speaker having to pause awkwardly between each word.

Voice Assistant Launches
2010 Apple's Siri
2012 Google Assistant
2014 Amazon's Alexa
2014 Microsoft's Cortana

Image Recognition

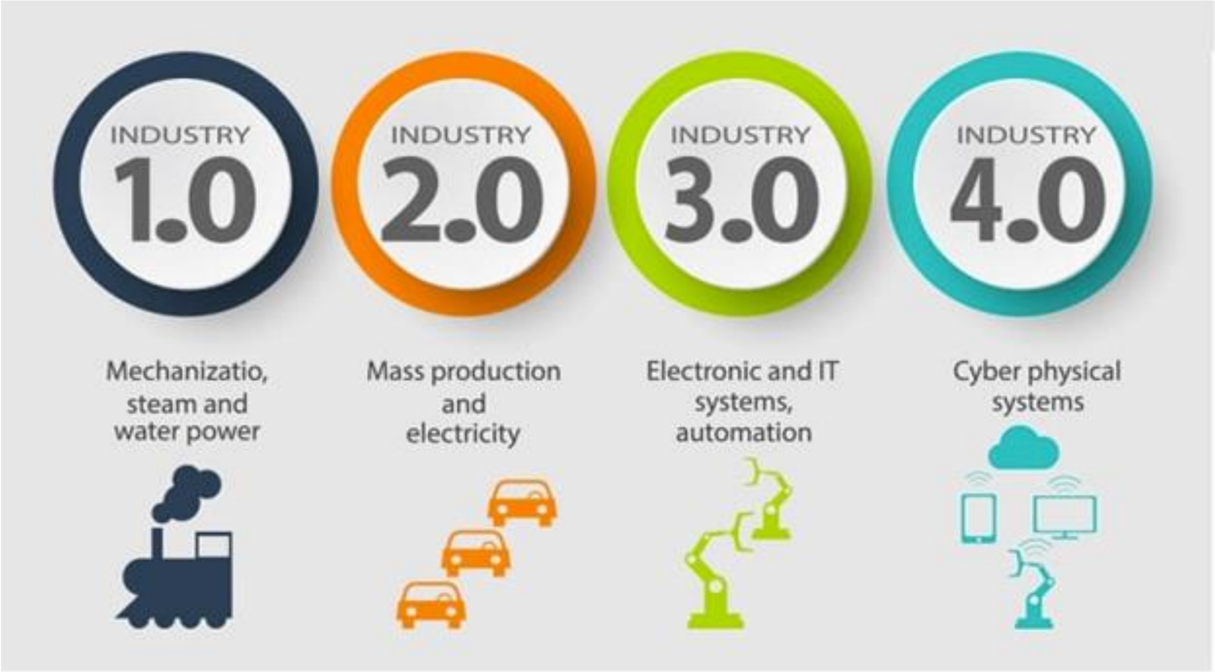
> “Computers used to not be able to see very well, and now they’re starting to open their eyes” – Jeff Dean, Google Senior Fellow



Financial Systems

Industry 4.0

> “As a technologist, I see how AI and the fourth industrial revolution will impact every aspect of people’s lives.” – Fei-Fei Li, Professor of Computer Science at Stanford University.



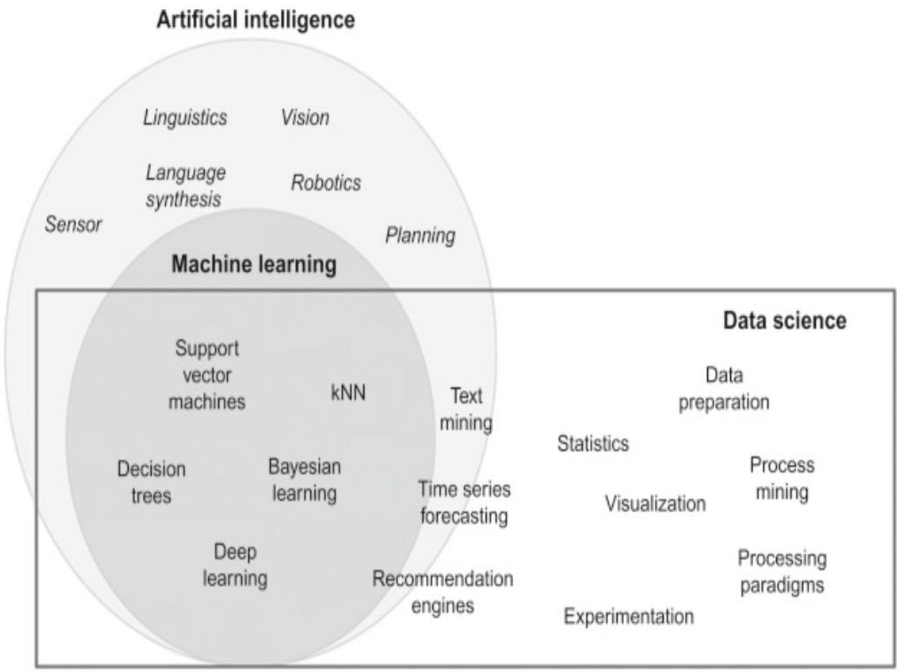
Machine Learning Everywhere!

Quiz

> Can you think of any real-world application driven by ML?
> How do you think they work?

Artificial Intelligence, Machine Learning, Data Science

- > Artificial Intelligence
  - – Machines capable of mimicking human behavior, particularly cognitive functions
- > Machine Learning
  - – Machines capable of learning from experience
- > Data Science
  - – Business application of ML, AI, and other quantitative fields like statistics



*Data Science (Second Edition), Vijay Kotu, Bala Deshpande, 2019

Tools for Machine Learning in Python

- **Environment**
  - **Jupyter:** provides an interactive notebook environment that is useful for exploratory data analysis, as well as creation of interactive documents
- **Data Analysis**
  - **Pandas:** provides a DataFrame object and other powerful set of methods to manipulate, filter, group and transform data
- **ML Modeling**
  - **Scikit-Learn:** provides common machine learning algorithms for modeling
- **Visualization**
  - **Matplotlib:** provides a useful interface for creating high-quality figures and charts
  - **Seaborn:** data visualization library based on matplotlib. Creates attractive and informative statistical charts

Course Overview

1. **Introduction to Statistical Learning**
2. Linear Regression
3. Classification
4. Model Building, Part 1
5. Model Building, Part 2
6. Resampling Methods
7. Linear Model Selection and Regularization
8. Moving Beyond Linearity
9. Unsupervised Learning
10. Dimensionality Reduction

What is Machine Learning?

- *A computer program is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P** if its performance on T, as measured by P, improves with experience E*. – Tom Mitchell

- Example of Spam Filter
  - Classifying an email as spam or ham (Task T)
  - Processing emails for spam or ham (Experience E)
  - Percentage of emails correctly classified as spam or ham (Performance P)

- As we see more examples of spam/ham, our ability to classify them improves

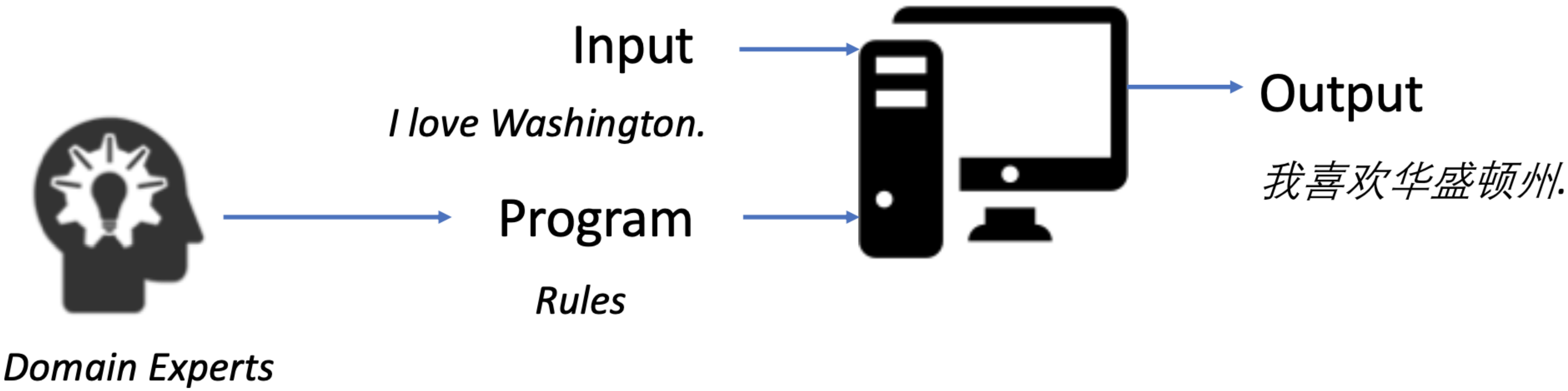
Few More Examples

Task
- Recommend products
- Voice recognition
- Facial recognition
- Credit scoring
- Anomaly detection

Experience
- Customer transactions
- Voice clips
- Images/video clips
- Historical transactions
- Sensor data

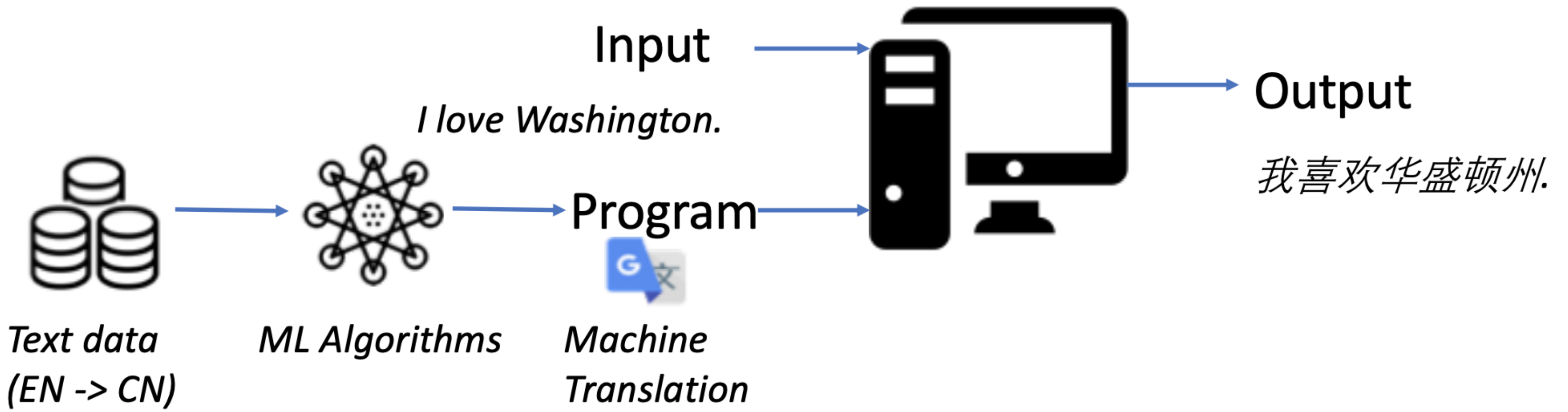
Performance
- Click through rate
- Accuracy in recognition
- Accuracy in face detection
- Lending default
- False positive rate

Machine Translation: Traditional Programming



Rule-based machine translation. (1950 – 2000)

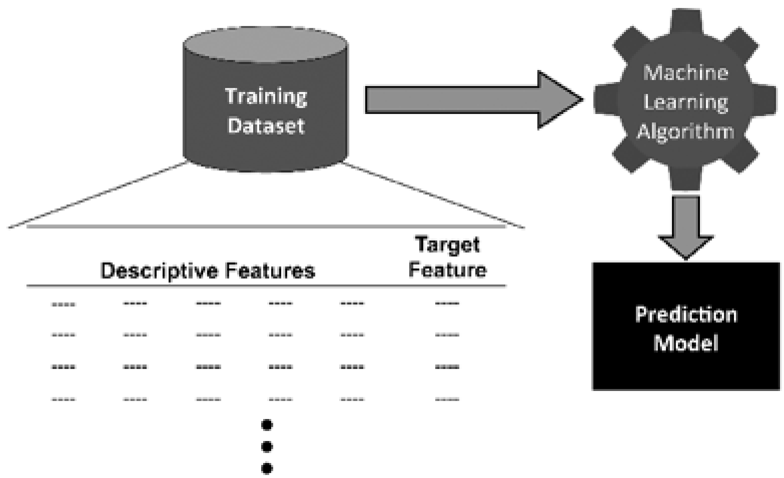
Machine Translation: Machine Learning



Google Machine Translation. (2013 – now)

Machine Learning Workflow

ML Model Training



ML Model Serving

When Do We Need Machine Learning?

- Humans cannot do it or can do it but cannot describe how they do it (thus cannot be programmed)
  - Effectiveness of a new compound treating some disease (cannot do)
  - Object recognition (cannot describe)
- Machines can do better than humans on certain tasks
  - Search page ranking (more data)
  - Play the game of Go (more computing power)
- Machines are cheaper than humans
  - Hand-written digits recognition
  - Voice recognition

When to Use What?

Criteria	Rules	Machine Learning	Deep Learning
Interpretability	✓	✓	
Accuracy		✓	✓
Maintenance		✓	
Speed of Execution	✓		

Quiz

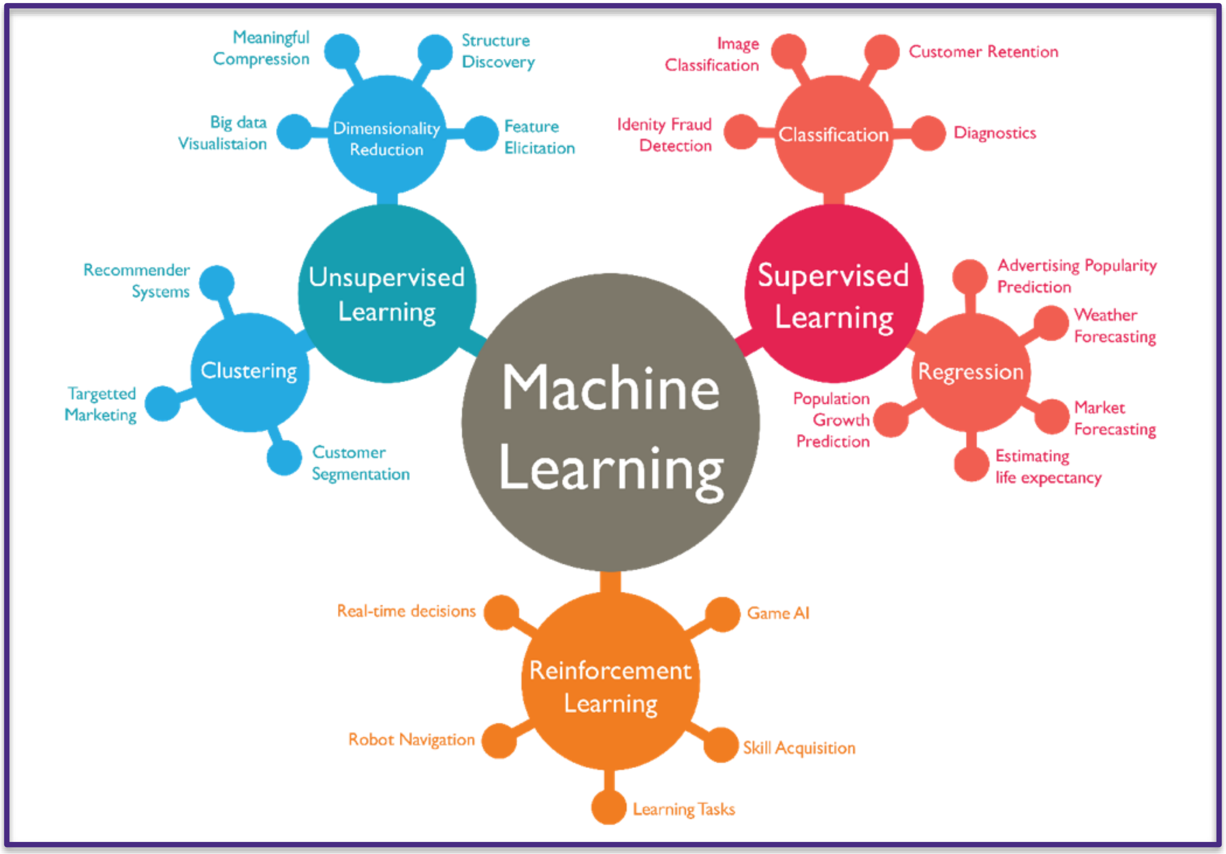
- Can you think of any scenarios where we prefer Machine Learning over Traditional Programming?

- Can you think of any scenarios where we prefer Traditional Programming over Machine Learning?

- TL;DR – Use the right tool for the job
  - Depending on your use case, rule-based systems may be quicker to build and iterate
  - Graduate from rules to machine learning when
    - There are too many rules
    - People are afraid to remove rules

Types of Machine Learning Algorithms

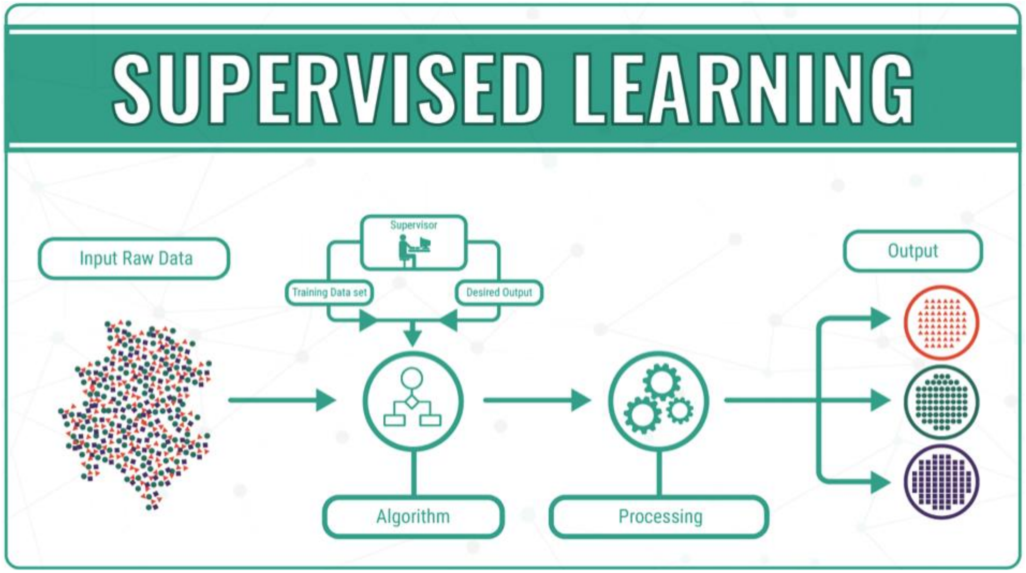
> Broadly divided into three categories



https://blogs.oracle.com/ai/types-of-machine-learning-and-top-10-algorithms-everyone-should-know

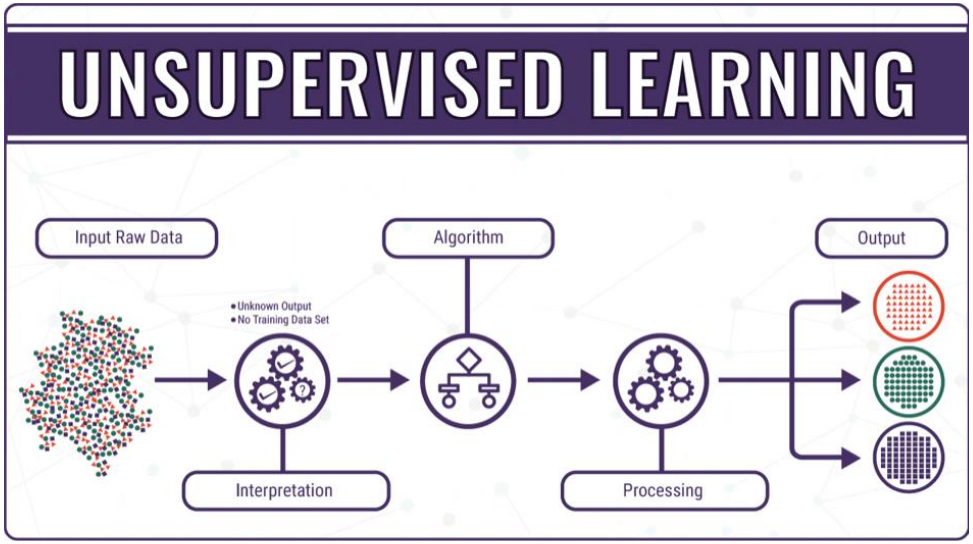
Supervised Learning

> Learn to predict an output from input using a **labeled** dataset
  - Linear/Logistic Regression
  - K Nearest Neighbor
  - Decision Tree
  - Random Forest
  - Gradient Boosting Trees
  - Support Vector Machines
  - (Deep) Neural Networks
  - More …



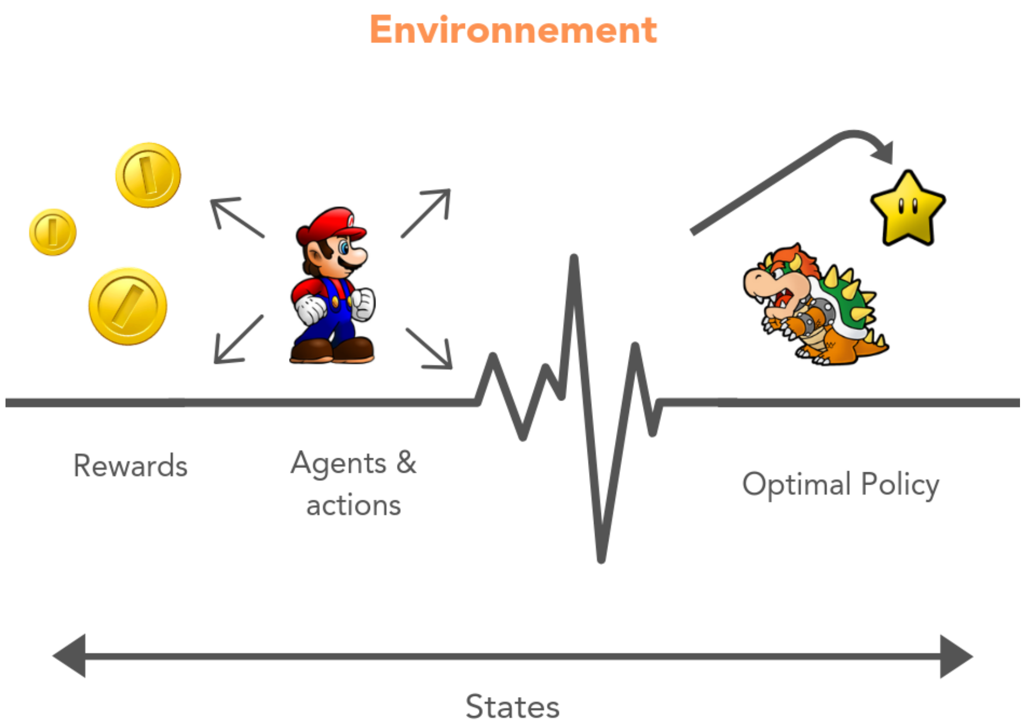
Unsupervised Learning

- Find patterns and structure in an **unlabeled** dataset
  - K-Means/K-Medoids
  - Hierarchical Clustering
  - Gaussian Mixture Models
  - Principal Component Analysis
  - Anomaly detection
  - Autoencoders
  - More …



Reinforcement Learning

> Learn to **take actions** in an environment to maximize expected future rewards



Quiz

Task	Learning Type
Housing price estimates on Redfin	?
Product recommendations on Amazon	?
Voice recognition on Alexa	?
Play the game of Go	?
Grouping Air Bnb listings into neighborhoods	?
Predict delivery times for Uber Eats	?
Segmenting US population for targeted advertising	?
Learn to play Starcraft	?
File a tax return	?

Supervised Learning – Terminology

- Label
  - The outcome we are interested in predicting
  - House prices, cat image, loan default
- Feature
  - An input/transformed variable that is used to predict a label
  - Number of bedrooms, colors in pixels, monthly income
- Training
  - Using labeled data to iteratively learn the function that transforms an input to output
- Inference
  - Using a trained model to make predictions on new, unseen data
- Performance
  - Mean square error, precision, recall, AUC

Supervised Learning

> Two main categories

> Regression
  - Predicting a continuous value
  - House prices in Seattle?
  - Stock prices of YourFavoriteTechCompany.com

> Classification
  - Predicting a discrete value
  - Will this applicant default on a loan?
  - Is this image a cat, a dog, or a pigeon?

Performance Evaluation of an ML Model

- Constructive feedback is important for a successful ML model
  - That's how humans learn too!
- You build a model, get feedback from metrics, make improvements and continue until you achieve a desirable accuracy
- An important aspect of evaluation metrics is their capability to discriminate among model results
- It is very important to define an appropriate metric for ML model evaluation!!

Performance Measures – Classification

- An error occurs in a classification algorithm when an incoming record is assigned a class when it truly belongs to another class
- **Type I Error:** (AKA *False Positive)*
  - Positive class is assigned to a record when the true class is negative
- **Type II Error:** (AKA *False Negative)*
  - Negative class is assigned to a record when the true class is positive

		Actual Class	
		C1 (Positive)	C0 (Negative)
Predicted Class	C1 (Positive)	Number of records with actual class **1** and predicted class **1**	Number of records with actual class **0** and predicted class **1**
	C0 (Negative)	Number of records with actual class **1** and predicted class **0**	Number of records with actual class **0** and predicted class **0**

Confusion Matrix

> Most classification error measures are summarized in the so-called *Confusion Matrix*

> **Accuracy**
  – Ratio of correctly predicted observation to the total observations
  – *(TP + TN)/(TP + FP + TN + FN)*

> **Precision**
  – Ratio of true positives over all true and false positives
  – TP/(TP + FP)

> **Recall**
  – ratio of true positives over the sum of true positives and false negatives
  – TP/(TP + FN)

		Actual Class	
		C1 (Positive)	C0 (Negative)
Predicted Class	C1 (Positive)	True Positive **(TP)**	False Positive **(FP)**
	C0 (Negative)	False Negative **(FN)**	True Negative **(TN)**

Which Classification Metric to Use?

> It is important to understand which classification metric is applicable to a given ML problem

> Accuracy is not always the right metric!

> Suppose we are creating an application that detects a rare disease in a patient. Typically, the disease occurs in 1% of the patients

> What is the accuracy of a *dumb* classifier that says no one has the rare disease?

Precision vs. Recall

- Precision or Recall are more appropriate for problems with imbalanced classes

- Use Precision when
  - The cost of False Positives is high
  - It is more important to be right about our predictions of the positive class
  - For example, an anomaly detection system that shuts down a plant when an anomaly is predicted

- Use Recall when
  - The cost of False Negatives is high
  - It is more important to retrieve/recall all possible records of the positive class
  - For example, clinical decisions usually aim for high recall

Quiz

- Which metric is more important for the following use cases?
- You are building an ML pipeline for cancer detection
- You are building an ML pipeline that predicts which day is suitable for launching a satellite
- You are building an ML pipeline that detects spam email
- You are building an ML pipeline that identifies and places a hold on a possibly fraudulent credit card transaction

Precision vs. Recall – A Tug of War

- We must consider **both** precision and recall to fully evaluate the effectiveness of a model
  - It is possible to create an ML algorithm that has 100% recall and yet 0% precision. Did we already discuss such an example?
- Unfortunately, precision and recall are often in conflict. That is, improving precision typically reduces recall and vice versa
- Increasing classification threshold
  - Increases precision, decreases recall
- Decreasing classification threshold
  - Deceases precision, increases recall
- A good ML algorithm design should strike an acceptable balance between the two

Other Evaluation Measures – F1 Score

> F1-score considers both precision and recall

> Harmonic mean of precision and recall
  - F1 = (2 * P * R)/(P + R)

> Why harmonic mean?
  - Why not arithmetic mean or geometric mean?

> However, F1-score is still dependent on classification threshold

> Another drawback is that precision and recall may not have the same cost and must not be weighed equally in a real application

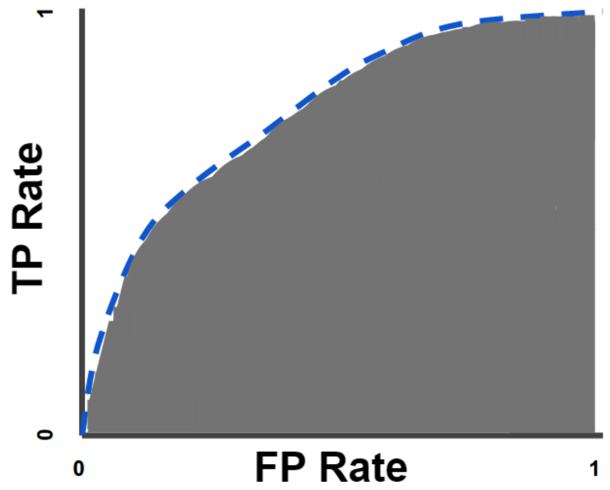
> Can use F_β

The general formula for non-negative real β is:

$$F_\beta = \frac{(1+\beta^2)\cdot(\text{precision}\cdot\text{recall})}{(\beta^2\cdot\text{precision}+\text{recall})}$$

Other Evaluation Measures – ROC/AUC

- **ROC curve** (Receiver Operating Curve) – Graph showing the performance of a classification model at all classification thresholds
- **AUC** (Area under the ROC Curve) – Measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1)



- AUC
  - Provides an aggregate measure of performance across all classification thresholds
  - It is the probability that the model ranks a random positive example more highly than a negative one
- AUC is desirable due to
  - Scale-invariance
  - Classification threshold invariance

Performance Measures – Regression

- Regression error measures the difference between predicted value (z_i) and true value (y_i)
- Mean squared error (MSE)
  - It is the L2 norm of difference between prediction and truth

$$\text{Root Mean-Squared Error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - z_i)^2}$$

- Mean absolute error (MAE)
  - It is the L1 norm of difference between prediction and truth

$$\text{Mean Absolute Error} = \frac{1}{N}\sum_{i=1}^{N}|y_i - z_i|$$

MSE vs. MAE – When to Use Which?

- Use MSE when
  - You want to avoid very large errors and still fit outliers somewhat reasonably
- Use MAE when
  - You think that the outliers are merely corrupted data and should be ignored

- In general, MSE is used more often

MSE	MAE
Best solution is given by the conditional mean	Best solution is given by the conditional median
Penalizes large errors (due to squaring the difference)	Doesn't penalize large errors
Affected by outliers	Robust and resistant to outliers
Closed form solution due to continuous derivatives	Solved iteratively

Model Generalization

- The primary challenge for any ML algorithm is to perform well on *new, unseen* data
- The ability to perform well on unobserved inputs is called **generalization**
- We typically have access to a training set and can evaluate a **training error**
- The distinction between ML and optimization is that in machine learning, we want a low **test error** as well
- A good ML algorithm aims to
  - Reduce the training error
  - Reduce the gap between training and test error

Bias – Variance Tradeoff

If we assume that $Y = f(X) + \epsilon$ and $E[\epsilon] = 0$ and $Var(\epsilon) = \sigma_\epsilon^2$ then we can derive the expression for the expected prediction error of a regression fit $\hat{f}(X)$ at an input $X = x_0$ using squared error loss

$$Err(x_0) = E[(Y - \hat{f}(x_0))^2 | X = x_0]$$

For notational simplicity let $\hat{f}(x_0) = \hat{f}$, $f(x_0) = f$ and recall that $E[f] = f$ and $E[Y] = f$

$$\begin{aligned}
E[(Y - \hat{f})^2] &= E[(Y - f + f - \hat{f})^2] \\
&= E[(y - f)^2] + E[(f - \hat{f})^2] + 2E[(f - \hat{f})(y - f)] \\
&= E[(f + \epsilon - f)^2] + E[(f - \hat{f})^2] + 2E[fY - f^2 - \hat{f}Y + \hat{f}f] \\
&= E[\epsilon^2] + E[(f - \hat{f})^2] + 2(f^2 - f^2 - fE[\hat{f}] + fE[\hat{f}]) \\
&= \sigma_\epsilon^2 + E[(f - \hat{f})^2] + 0
\end{aligned}$$

For the term $E[(f - \hat{f})^2]$ we can use a similar trick as above, adding and subtracting $E[\hat{f}]$ to get

$$\begin{aligned}
E[(f - \hat{f})^2] &= E[(f + E[\hat{f}] - E[\hat{f}] - \hat{f})^2] \\
&= E\left[f - E[\hat{f}]\right]^2 + E\left[\hat{f} - E[\hat{f}]\right]^2 \\
&= \left[f - E[\hat{f}]\right]^2 + E\left[\hat{f} - E[\hat{f}]\right]^2 \\
&= Bias^2[\hat{f}] + Var[\hat{f}]
\end{aligned}$$

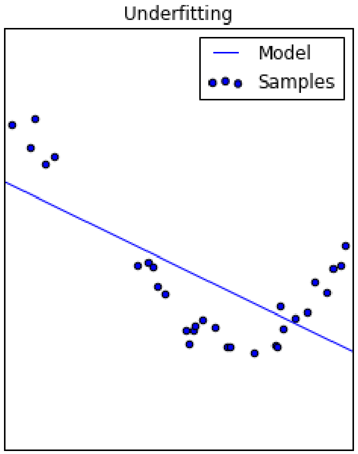
Putting it together

$$\boxed{E[(Y - \hat{f})^2] = \sigma_\epsilon^2 + Bias^2[\hat{f}] + Var[\hat{f}]}$$

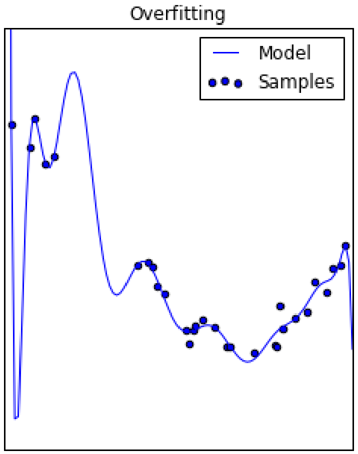
Underfitting and Overfitting

> Bias – Variance tradeoff gives rise to the two central challenges in machine learning i.e., Underfitting and Overfitting

Underfitting
- The model is not able to obtain a sufficiently low training error
- Has high bias

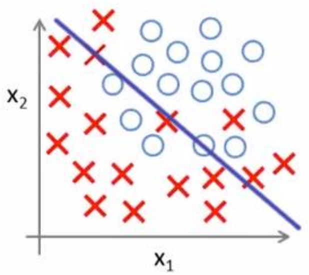


Overfitting
- The gap between training error and test error is large
- Has high variance

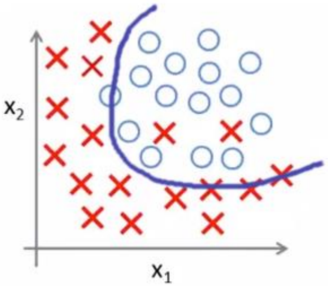


Quiz

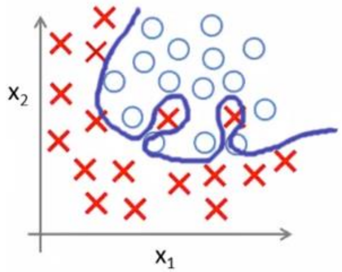
> Identify the underfitting, overfitting and just-right scenarios



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$



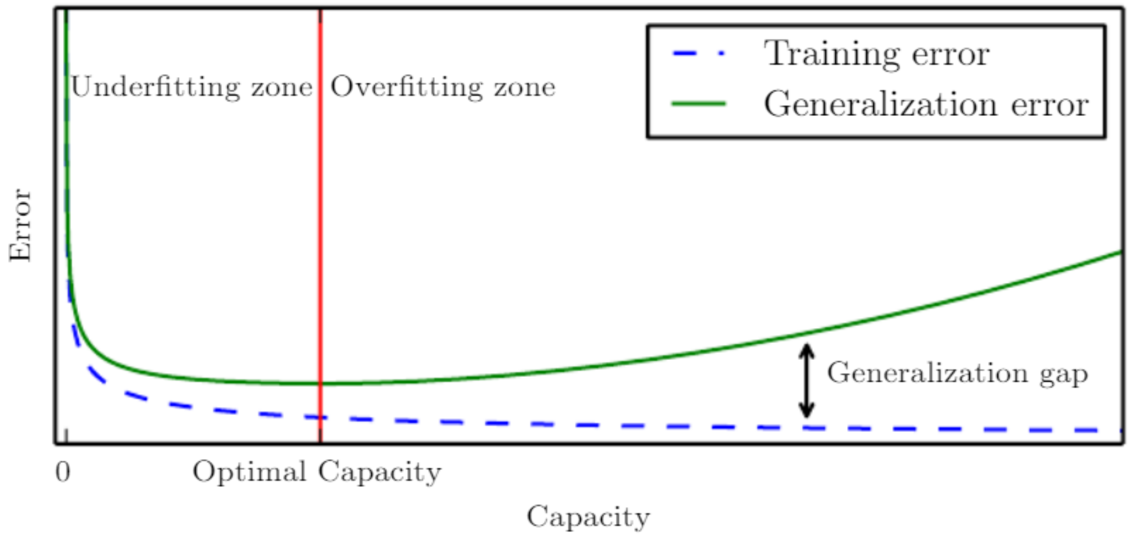
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

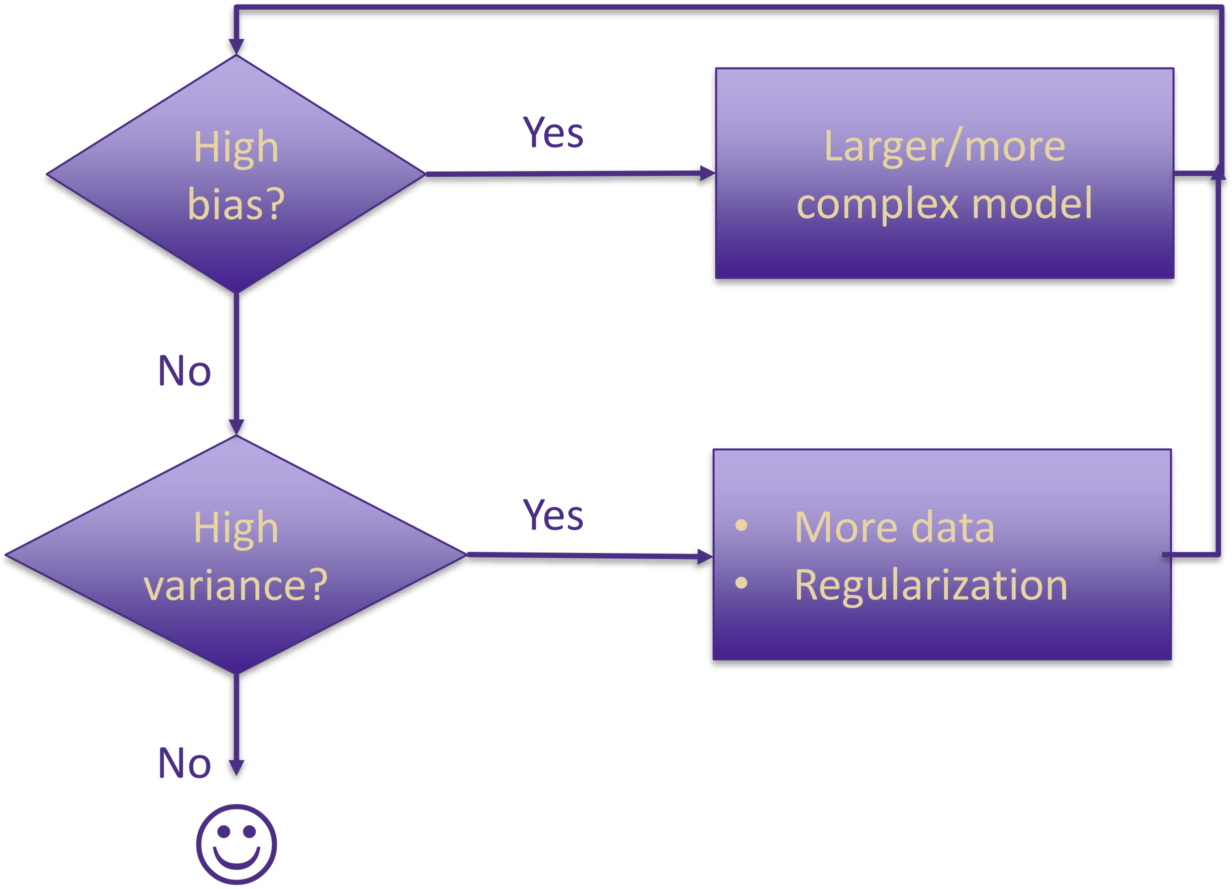
Error vs. Model Capacity/Complexity

- At the left end of the graph, training and test error are both high. This is the **Underfitting Zone**
- Increasing capacity leads to lower training error but larger gap between training and error. This is the **Overfitting Zone**
- **Optimal Capacity** is when both errors are sufficiently low

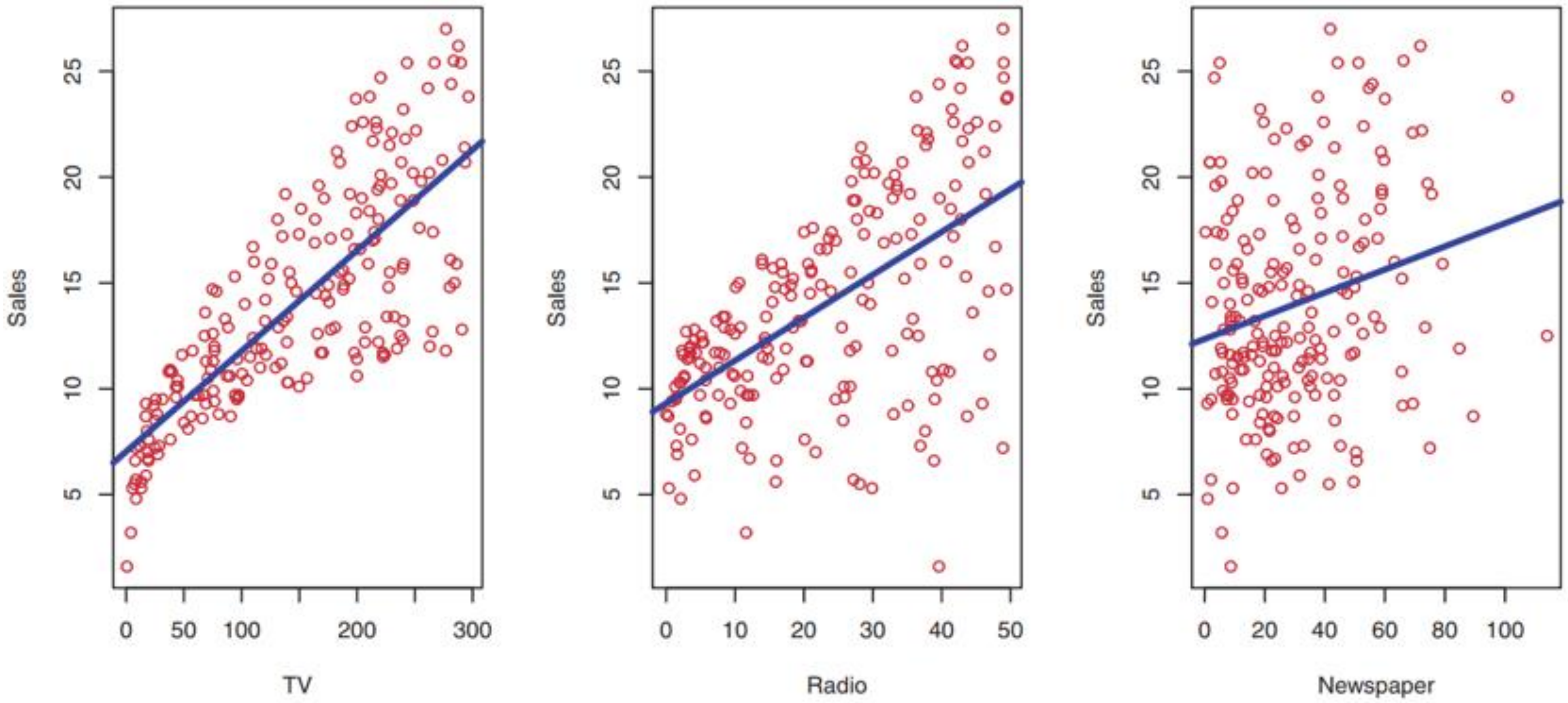


*Deep Learning – Goodfellow, Bengio and Courville

Dealing With Underfitting and Overfitting



What is Statistical Learning?



$$Sales \approx f(TV, Radio, Newspaper)$$

What is Statistical Learning?

- Inputs, also known as predictors, independent variables, features, or more generally, variables
- Outputs, also known as response or dependent variable

Suppose an observed quantitative response Y and p different predictors $x_1, x_2, \dots, xp$. The assumed relationship between Y and $X = x_1, x_2, \dots, xp$ can be generalized as:

$$Y = f(X) + \epsilon$$

Estimating f

- **Parametric Methods:** Utilize a two-step model-based approach

- First, make an assumption about the functional nature, or shape, of *f*. For example, assume that *f* is linear, yielding a linear model

- Once a model has been selected, use training data to fit, or train, the model. In the case of a linear model of the form

$$f(x) = \beta_0 + \beta_1 x_1 +, \dots, +\beta_p x_p$$

The training procedure should yield estimates for the parameters $\beta_0, \beta_1, \dots, \beta_p$ such that

$$Y \approx f(X) \approx \beta_0 + \beta_1 x_1 +, \dots, +\beta_p x_p$$

Estimating f

- **Non-parametric Methods**

- Don't make explicit assumptions about *f* and instead seek to estimate *f* by getting as close to the data points as possible without being too coarse or granular, preferring smoothness instead

- Can fit a wider range of possible shapes for *f* since essentially no assumptions about the form of *f* are made. However, since non-parametric approaches don't simplify the problem of estimating *f*, they tend to require a very large number of observations to accurately estimate *f*

- Example: Nearest Neighbor, Random Forests, etc.

Matrix Algebra – Terminology Note

- **Scalar**: a single numeric value
- **Vector**: a 1-dimensional array of values
- **Matrix**: a 2-dimensional array of values
- **Tensor**: an array of values with 3 or more dimensions (e.g. an array of images)

Notation and Simple Matrix Algebra

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \quad x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix} \qquad \mathbf{x}_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}$$

Row: lower-case, script
$\mathscr{X}$ = values for an observation
$\mathscr{i}$ = an index for the row
$\mathscr{p}$ = the number of predictors

Column: bold, lower-case
X = values for a variable
j = an index for the column
n = number of observation

Output Vector

An output vector is used for supervised learning

- Numeric output values for regression
- Nominal (categorical) output values for classification
- Rank for ranking problems

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Alternative Names

X :

- Input Variable
- Predictor
- Covariate
- Independent
- Exogenous
- Cause
- Manipulated

Y:

- Output Variable
- Response
- Target
- Dependent
- Endogenous
- Effect
- Measured

Matrix Transposition

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \qquad \mathbf{X}^T = \begin{pmatrix} x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \\ \vdots & \vdots & & \vdots \\ x_{1p} & x_{2p} & \dots & x_{np} \end{pmatrix}$$

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix} \qquad x_i^T = \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{ip} \end{pmatrix}$$

Just swap the row and column indices: $new_{j,i} = old_{i,j}$

Alternative Matrix Notation

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_p \end{pmatrix}$$

Matrix expressed as a set of column vectors, where each column is a variable

$$\mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix}$$

Matrix expressed as a set of row vectors, where each row is an observation

Vector Multiplication

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} \qquad x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}$$

$$\beta^T x = \beta_0 * x_0 + \beta_1 * x_1 + \beta_2 * x_2$$

[sometimes called a dot product]

Matrix Multiplication

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$$

$$(\mathbf{AB})_{ij} = \sum_{k=1}^{d} a_{ik} b_{kj}$$

$$\mathbf{AB} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$

$$\boldsymbol{A} \in \mathbb{R}^{n\ x\ p} \qquad \boldsymbol{B} \in \mathbb{R}^{p\ x\ k} \qquad \boldsymbol{AB} \in \mathbb{R}^{n\ x\ k}$$

$\mathbb{R}$: a value from the real number line

Jupyter Notebook

> Install Jupyter using [Anaconda Distribution]
> Install packages necessary for data analysis and machine learning

In [1]:
```
# install packages
import sys

!conda install --yes --prefix {sys.prefix} numpy scipy pandas scikit-learn matplotlib seaborn
```

```
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.4.9
  latest version: 4.5.12

Please update conda by running

    $ conda update -n base conda



# All requested packages already installed.
```

ON-BRAND STATEMENT

FOR GENERAL USE

> What defines the students and faculty of the University of Washington? Above all, it's our belief in possibility and our unshakable optimism. It's a connection to others, both near and far. It's a hunger that pushes us to tackle challenges and pursue progress. It's the conviction that together we can create a world of good. And it's our determination to Be Boundless. Join the journey at **uw.edu**.