https://ttpoll.com/p/911058

COSC 325: Introduction to Machine Learning

Dr. Hector Santos-Villalobos

Dr. Santos

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Lecture E1: Exam 1 Review

THE UNIVERSITY OF TENNESSEE KNOXVILLE

https://ttpoll.com/p/911058

https://ttpoll.com/p/569627

Machine Learning vs Computer Programming

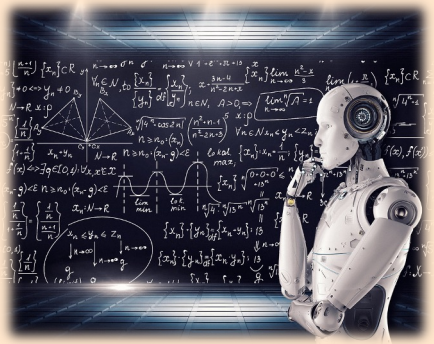
- Traditional Programming:
  - Algorithms are sequences of instructions that are carried out to transform an input into an output
  - Fundamentally, they are lists of instructions
- Machine Learning:
  - The list of instructions is ***Learned*** from data
  - Useful when the sequence of instructions is difficult to define
  - Examples
    - Facial recognition
    - Autonomous driving

https://ttpoll.com/p/569627

AI vs Machine Learning

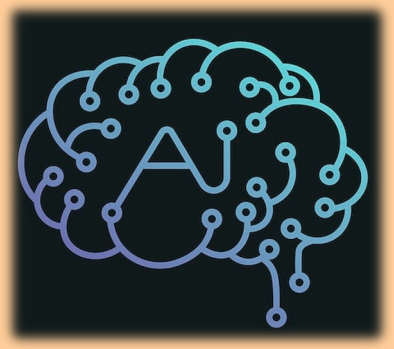
Artificial General Intelligence

Computers "mimic" how humans learn.



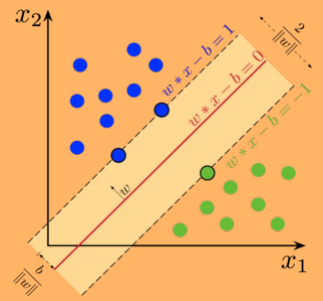
Artificial Intelligence

Computers mimic human behavior.



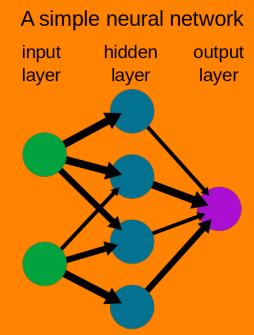
Machine Learning

Ability to learn without explicit hand-made rules.



Deep Learning

Automated extraction of patterns/features from raw data using multi-layer neural networks.



Teaching computers how to **learn a task** directly from **data**.

Figure inspired on MIT 6.S191 course slide.

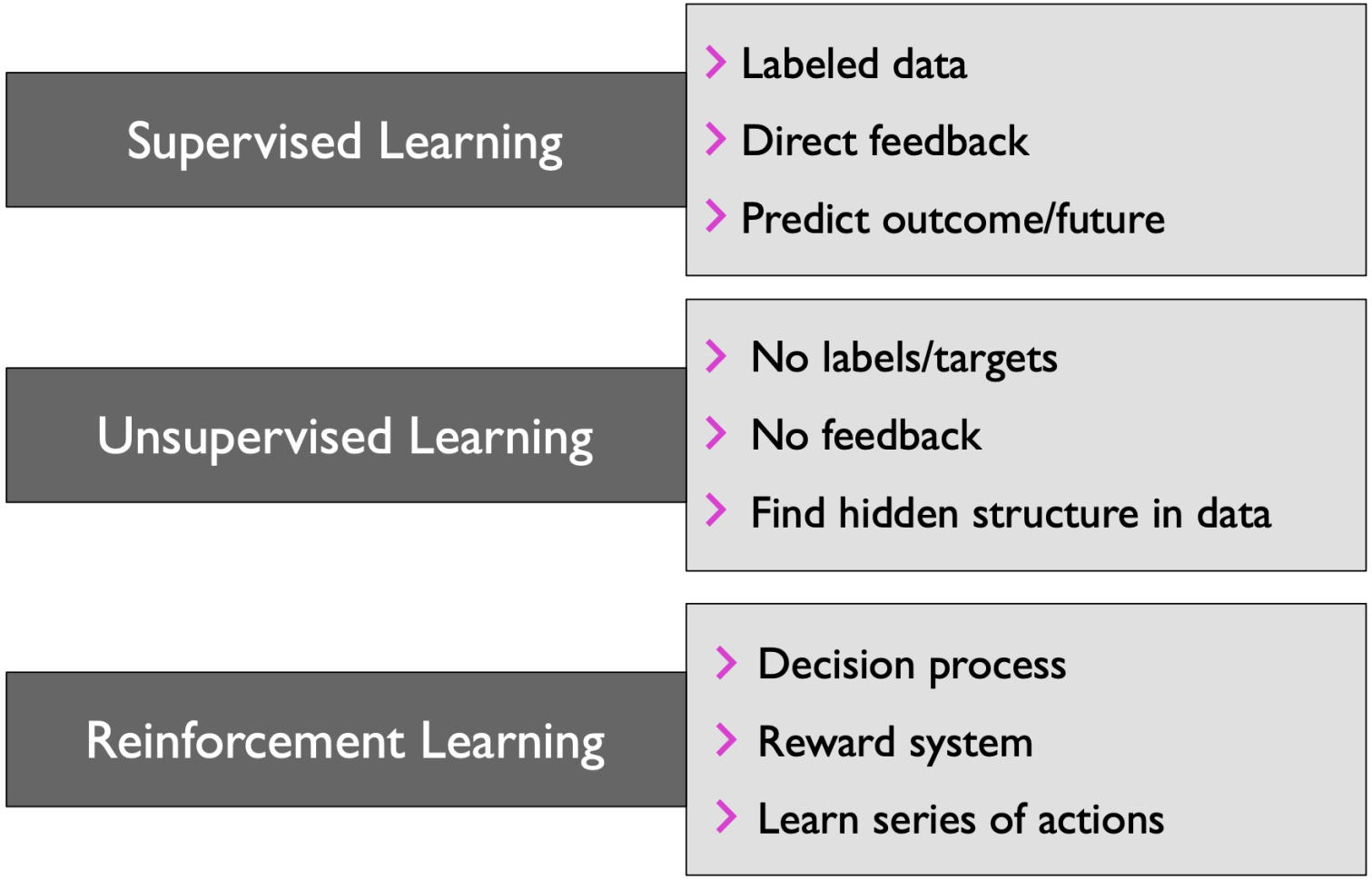
https://ttpoll.com/p/569627

Generalization

- A key component of “learning” is the ability to generalize
  - Take information that has been learned previously and apply it to new but related scenarios
- For a technique to be considered a machine learning approach, it ***must*** be able to generalize
- Thus, we must evaluate its ability to generalize

Slide credit: Dr. Schuman

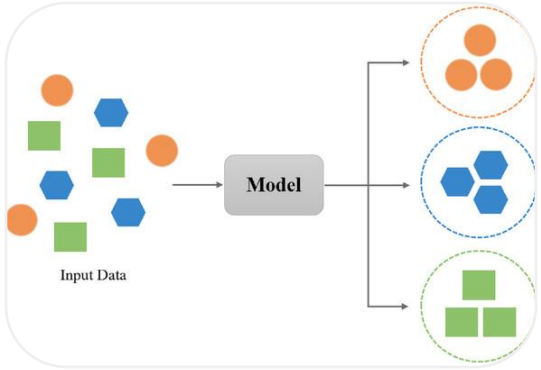
Categories of Machine Learning



Slide credit: Dr. Raschka

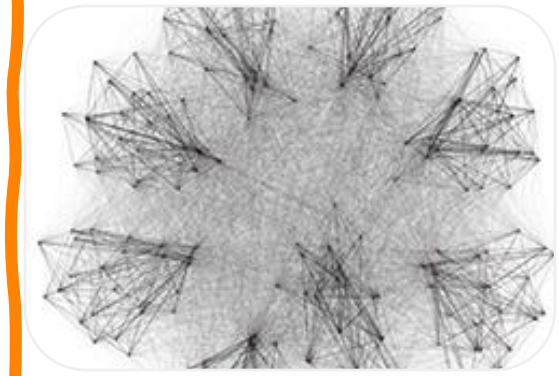
https://ttpoll.com/p/782763

Machine Learning Categories



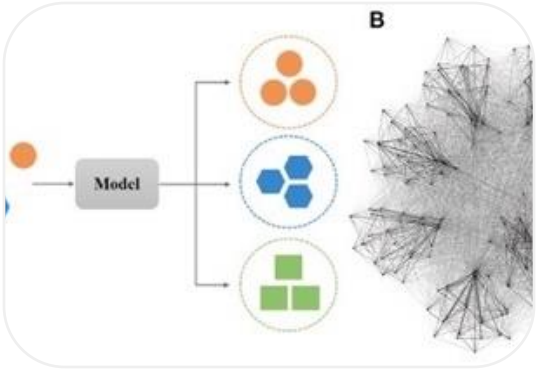
Supervise Learning

- Trained on “Labeled dataset”
- Needs pairs of inputs and outputs (ground truth)



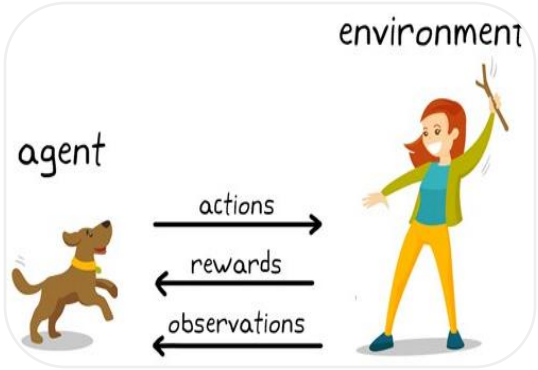
Unsupervised Learning

- The algorithm discovers patterns and relationships in unlabeled data
- It needs inputs only and, most of the time, some context. (e.g., number of unique labels)



Semi-Supervised Learning

- Combines SL and UL
- E.g., a small subset of labeled data is used to label unlabeled data.
- E.g., Generative Adversarial Network mapping blonde to brunette.



Reinforcement Learning

- It learns by interacting with the environment.
- Trial, error, and delay
- Needs well-defined reward feedback.

Notation

Features

Targets/Labels

n training examples

Training set: $D = \{\langle x^{(i)}, y^{(i)} \rangle, i = 1, \ldots, n\}$

Training sample #3: $\langle x^{(3)}, y^{(3)} \rangle$

This is the function/program our ML algorithm will generate.

Unknown function: $f(x) = y$

Hypothesis: $h(x) = \hat{y}$

Classification

Regression

$h: \mathbb{R}^m \rightarrow \{0,1, 3, 4, 5\}$

$h: \mathbb{R}^m \rightarrow \mathbb{R}$

Features

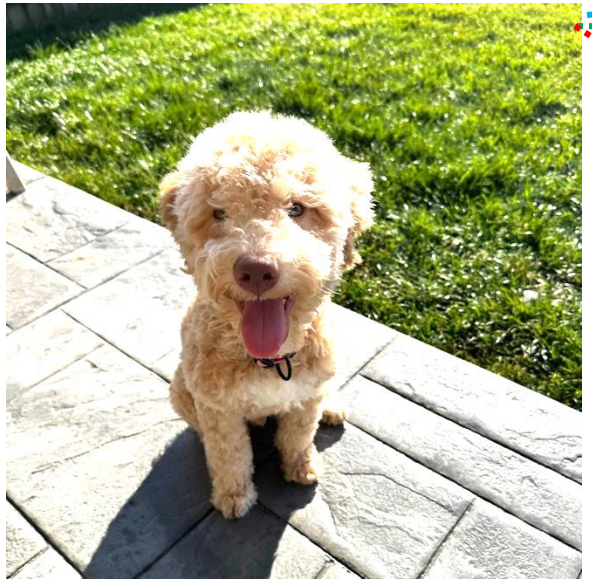
Slide credit: Dr. Raschka

https://ttpoll.com/p/911058

https://ttpoll.com/p/782763

What about color?

Input $x^{(i)}$



100

100

RGB Matrices

10	15	5	0
25	4	30	120
135	130	200	230
145	160	210	250

R

255	255	255	255
255	255	255	30
255	255	255	30
255	255	255	2

G

0	2	15	25
40	40	40	123
30	40	40	123
2	40	40	123

B

Unroll

$$x^{(i)} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 10 \\ 15 \\ \vdots \\ 255 \\ 255 \\ \vdots \\ 0 \\ 2 \\ \vdots \end{bmatrix}$$

$m = 3*100*100$
$=30{,}000$

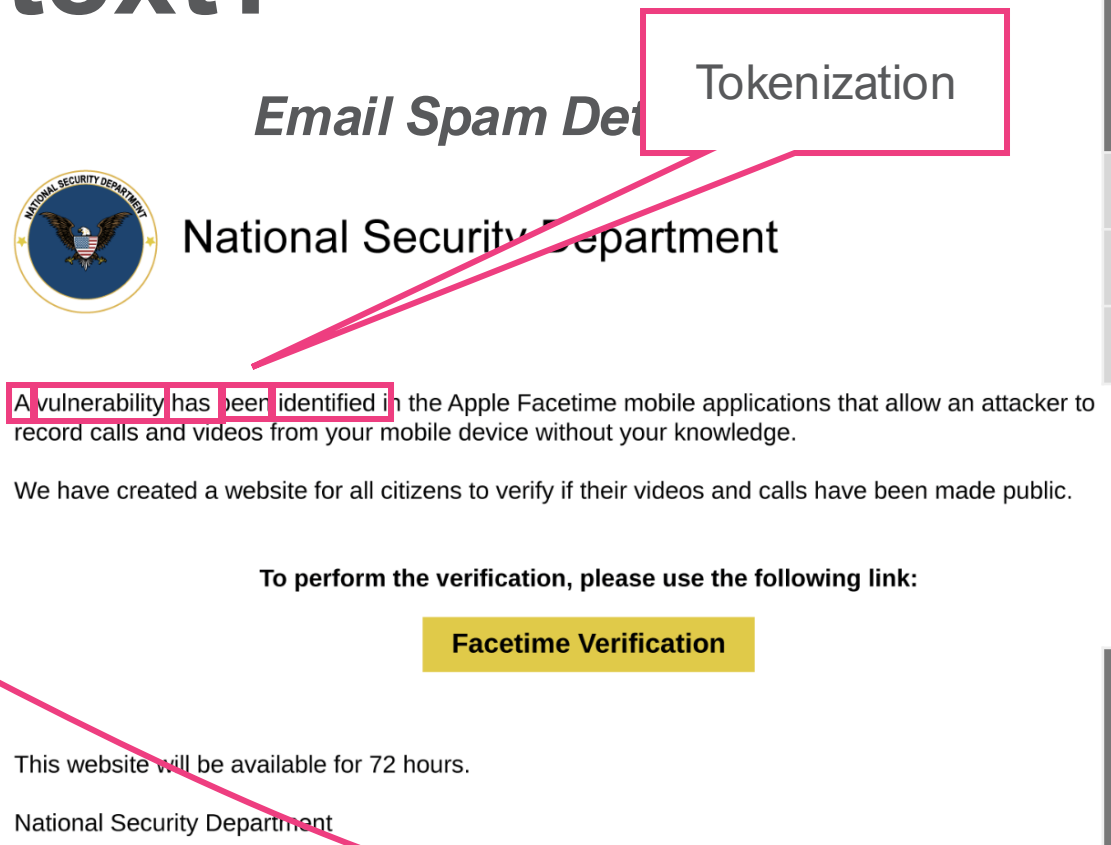
https://ttpoll.com/p/782763

What about text?

$$x^{(i)} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Feature Vector

Tokenization



Bag of Words

	identified	vulnerability	Urgent	Money	Transaction	Pay	link	Password	Verify	Hack
Doc 1	1	1	0	0	0	0	1	0	3	1
Doc 2	0	2	1	0	1	0	1	1	1	1
Doc 3	0	0	1	2	1	2	1	0	0	0

Term Weighting

$$w_j^{[i]} = TF_j^{[i]} * \ln(n/n_j)$$

	identified	vulnerability	Urgent	Money	Transaction	Pay	link	Password	Verify	Hack
Doc 1	0.48	0.18	0.00	0.00	0.00	0.00	0.00	0.00	0.53	0.18
Doc 2	0.00	0.35	0.18	0.00	0.18	0.00	0.00	0.48	0.18	0.18
Doc 3	0.00	0.00	0.18	0.95	0.18	0.95	0.00	0.00	0.00	0.00

The document features

Image source: https://www.terranovasecurity.com/blog/top-examples-of-phishing-emails

https://ttpoll.com/p/782763

Representing the dataset

A sample: (x, y) $\quad x \in \mathbb{R}^m, \quad y \in \{0,1\}$

Input

Ground Truth Label

Set of Samples: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(n)}, y^{(n)})\}$

Matrix representation of inputs: $X = \begin{bmatrix} x^{(1)T} \\ x^{(2)T} \\ \vdots \\ x^{(n)T} \end{bmatrix}$ $\quad X \in \mathbb{R}^{n \times m}$ $\quad$ *Python X.shape* $= (n, m)$

Number of rows

Number of columns

Matrix representation of labels: $Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$ $\quad Y \in \mathbb{R}^{n \times 1}$

Y.shape $= (n, 1)$

https://ttpoll.com/p/911058

Today's Topics

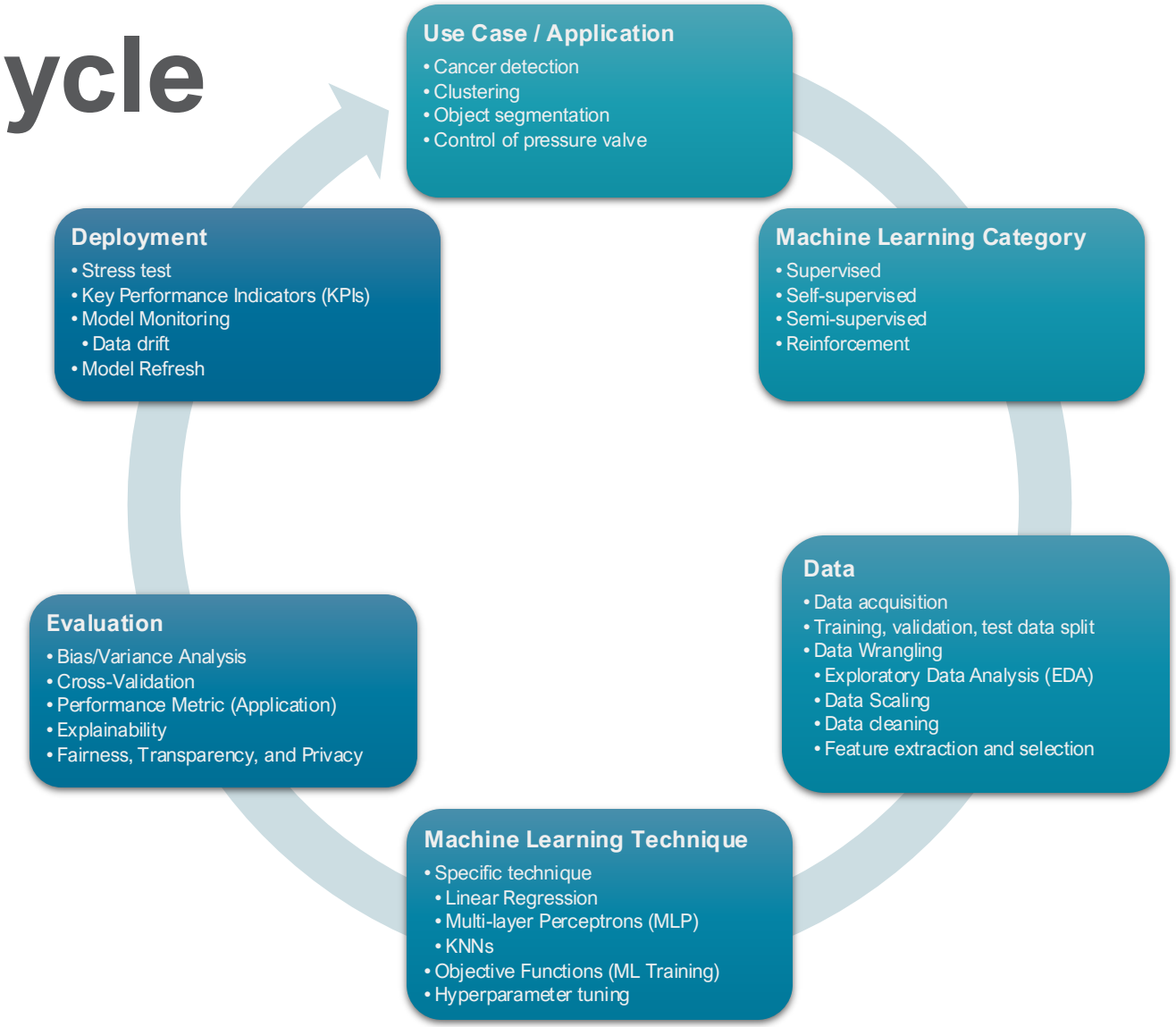
Wrap up Numpy

Matplotlib and Pandas

Scikit-learn

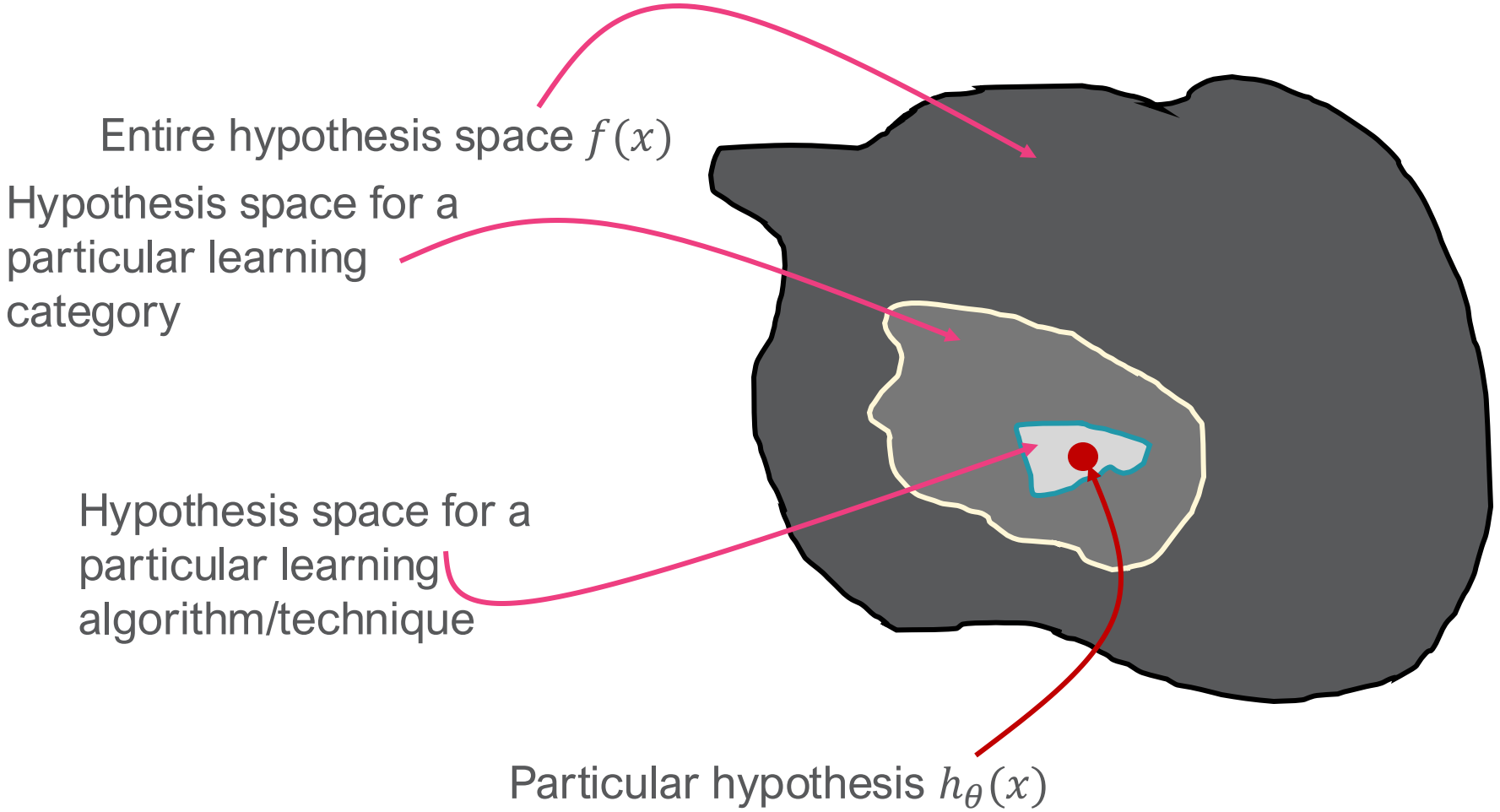
https://ttpoll.com/p/389226

ML Life Cycle

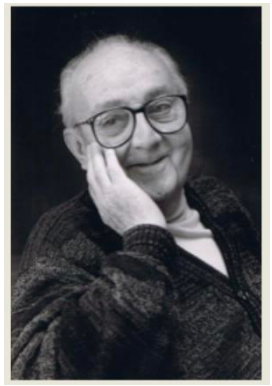


https://ttpoll.com/p/389226

Hypothesis Space



A model cannot make a better hypothesis than one provided by the sample distribution and within the limits of the learning category and technique.



"All models are wrong, but some are useful."
– Prof. George Box

Slide credit: Dr. Raschka

Theorem #1: Bayes Optimal Classifier

- Bayesian optimal classifier:

$$h^{BO}(x) = \arg\max_{y \in C} f(x, y)$$

- Theorem 1: The Bayes Optimal Classifier $h^{(BO)}$ achieves minimal zero/one error of any deterministic classifier.
  - Note: This assumes comparison against deterministic classifiers ($\hat{y}^{(i)} = h\left(x^{(i)}\right)$)

$$0{-}1 \text{ loss} = \text{Zero/One Error} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left(\hat{y}^{(i)} \neq y^{(i)}\right)$$

Slide credit: Dr. Schuman

Loss and Cost Functions

- Loss $\mathcal{L}_\theta\left(y^{(i)}, \hat{y}^{(i)}\right)$ is the error between the ground truth (i.e., expected response) $y^{(i)}$ and the model prediction $\hat{y}^{(i)}$.

- Cost $J(\theta)$ is a measure of overall model error for parameters θ.

Per sample $x^{(i)}$

Expected Performance

We want both to be SMALL

Regression Objective Function Candidates

- Sum of Squared Residuals (Very similar to MSE)

$$\text{SSR} = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

- Mean Absolute Error

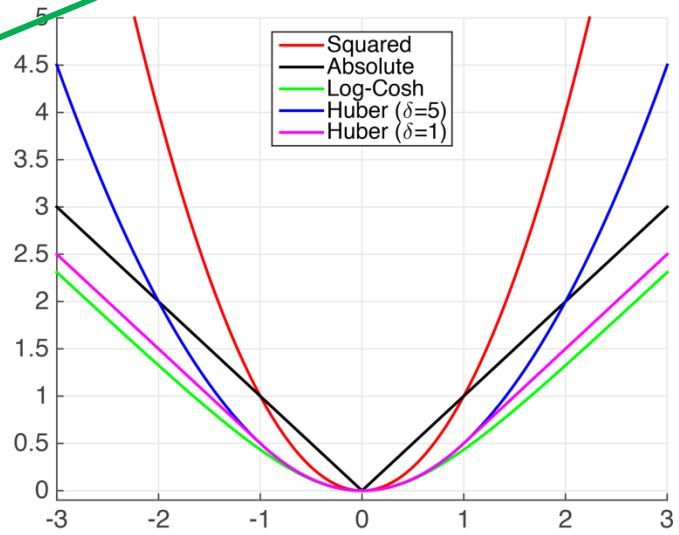
$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y^{(i)} - \hat{y}^{(i)}|$$

- Huber Loss

$$\text{Huber Loss} = \sum_{i=1}^{n} L_\delta(y^{(i)} - \hat{y}^{(i)})$$

$$\text{where } L_\delta(r) = \begin{cases} \frac{1}{2} r^2 & \text{if } |r| \leq \delta \\ \delta(|r| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

Combines the strengths of SSR and MAE (i.e., quadratic for small errors and linear for large errors)



Gradient Descent Algorithm

$X :=$ data features

$y :=$ data targets

$\theta = \theta_0$

Repeat:

$$\hat{y} = h_\theta(X)$$

$$cost = J_\theta(y, \hat{y})$$

$$d\theta = \frac{\partial J_\theta(y,\hat{y})}{\partial \theta}$$

$$\theta := \theta - \alpha(d\theta)$$

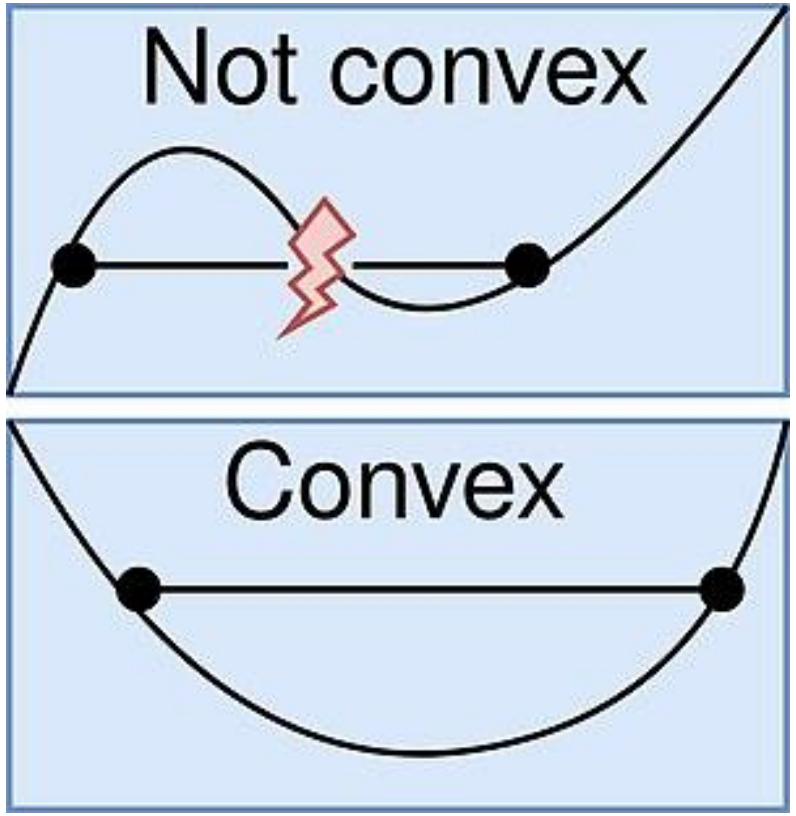
Until a fixed number of iterations or $d\theta$ very small.

https://ttpoll.com/p/911058

https://ttpoll.com/p/389226

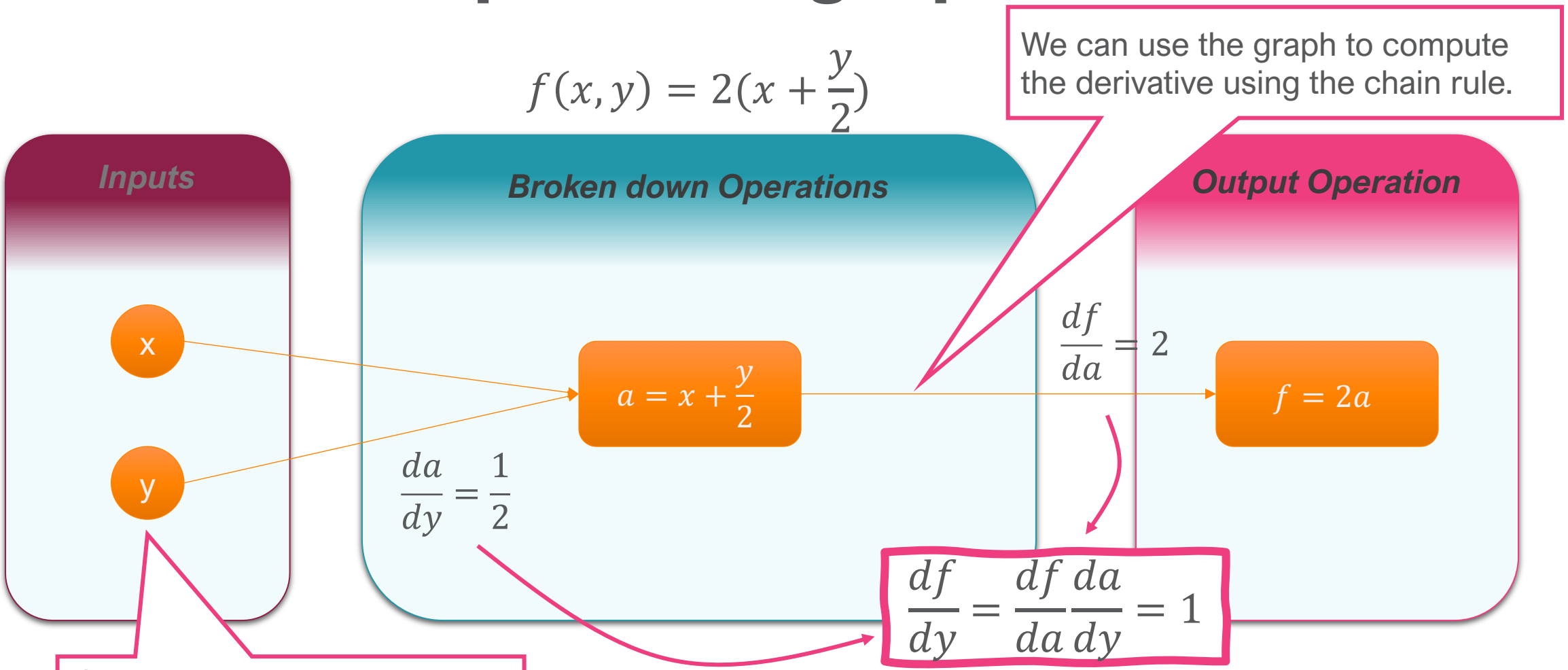
Preferred Objective Functions Characteristics

- Adequate sensitivity to outliers
- Computationally efficient
- Differentiable everywhere
- Interpretable
- Convex
- Aligned with the use case



What is a computation graph?

$$f(x, y) = 2(x + \frac{y}{2})$$



Chain Rule Refresher

- Example

$$f(x) = \frac{1}{(1-x)^2} \qquad g(x) = u = 1 - x \qquad f(u) = \frac{1}{u^2}$$

$$\frac{du}{dx} = -1 \qquad \frac{df}{du} = \frac{-2}{u^3}$$

$$\frac{df}{dx} = \frac{df}{du}\frac{du}{dx} = \frac{-2}{(1-x)^3} \times -1 = \frac{2}{(1+x)^3}$$

Linear Regression

- For $m = 1$:

$$y^{(i)} = \theta_1 x^{(i)} + \theta_0$$

θs are the parameters to learn

For $m = n_x$:

$$y^{(i)} = \theta_0 1 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \cdots + \theta_m x_m^{(i)}$$

$$y^{(i)} = [1, x^{(i)T}]\theta$$

Shape: $(1,1)$

Shape: $(1, m+1)$

Shape: $(m+1, 1)$

Exact Solution vs. Gradient Descent

$$\theta = (X^T X)^{-1} X^T y$$

- This gives an exact solution (modulo numerical inaccuracy for inverting the matrix)
- Gradient descent gives you progressively better solutions and eventually gets to an optimum

Qualitative Features

- Example: investigate differences in credit card balance between Asian, Caucasian, and African American.
  - We create an extra dummy variables
    - Dummy variables = Qualitative classes – 1
    - Variable 1: Asian
    - Variable 2: Caucasian
    - Baseline: African American

1s for Asian samples, 0s otherwise

$$y = \theta_0 + \theta_1 x_{1,1} + \theta_2 x_{1,2}$$

1s for Caucasian samples, 0s otherwise

Slide Credit: James, et. al., An Intro to Statistical Learning, 2023.

https://ttpoll.com/p/911058

Matrix Design

- We have an input matrix X with shape (n, m)
- We want to fit a polynomial of degree d
- Polynomial feature extraction process
  - Columns for each feature polynomial power (e.g., x_1^3, x_4^5)
  - Plus, columns for each feature interaction up to $d-1$(e.g., $x_1 x_4, x_1^2 x_4$)
- Example for data with n samples, $\mathrm{m} = 2$ features, and polynomial degree $d = 3$.

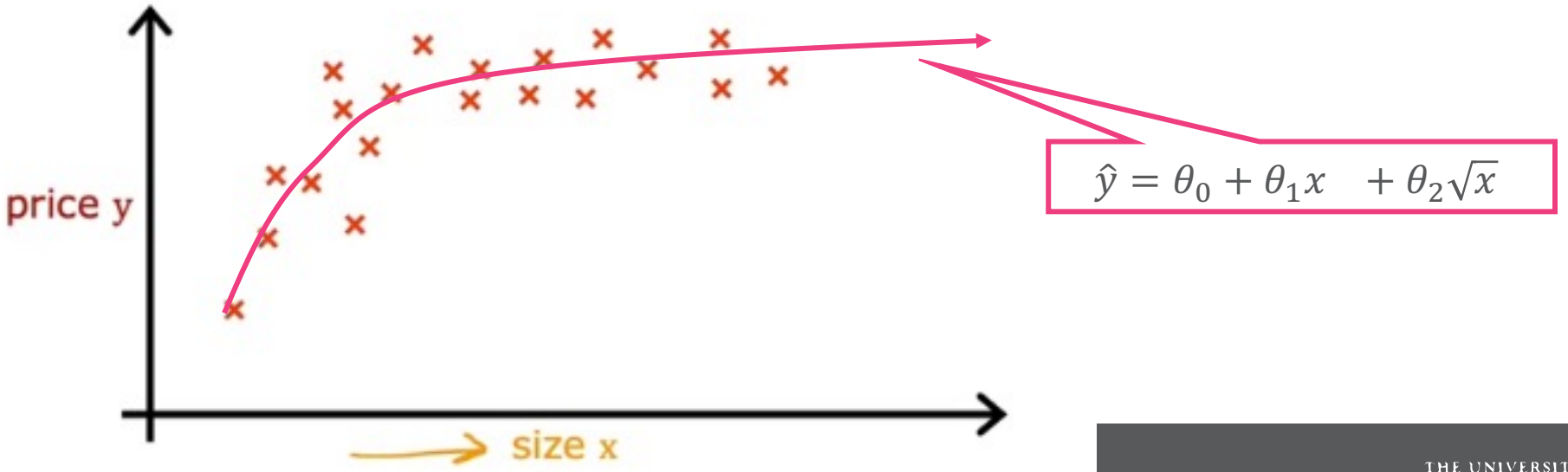
$$X_{new} = [1 \quad x_1 \quad x_2 \quad x_1^2 \quad x_1 x_2 \quad x_2^2 \quad x_1^3 \quad x_1^2 x_2 \quad x_1 x_2^2 \quad x_2^3]$$

- Then, apply linear regression algorithm on X_{new}

Slide Credit: James, et. al., An Intro to Statistical Learning, 2023.

Thinking outside the box

- Look at your data
- Learn about the domain
- Use equations that match your data and domain knowledge



https://ttpoll.com/p/911058

Issues with Linear Regression for classification

- For balanced binary classification problems, linear regression is a good classifier.
- Since in the population $E(y\,|X = x) = \Pr(Y = 1|X = x)$, we might think that regression is perfect for this task.
- However, linear regression might produce probabilities less than zero or bigger than one.

Slide Credit: James, et. al., An Intro to Statistical Learning, 2023.

Issues with Linear Regression for classification

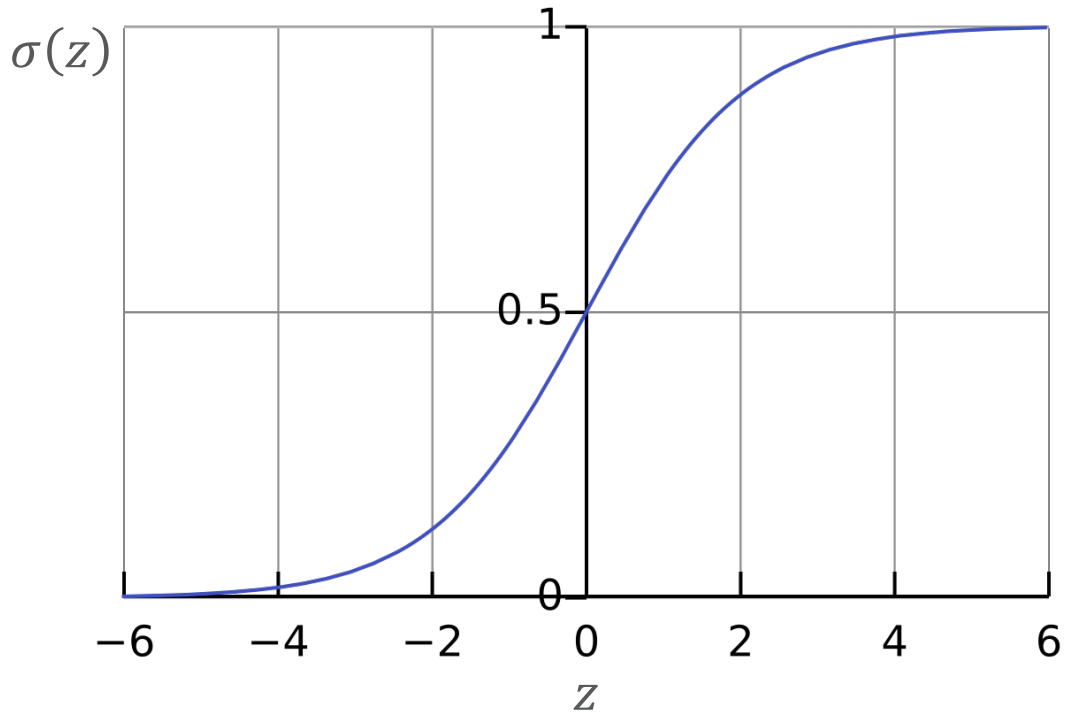
- Now suppose we have a response variable with three possible values. A patient presents at the emergency room, and we must classify them according to their symptoms.

$$y = \begin{cases} 1 & if \quad stroke \\ 2 & if \quad drug\ overdose \\ 3 & if \quad epileptic\ seizure \end{cases}$$

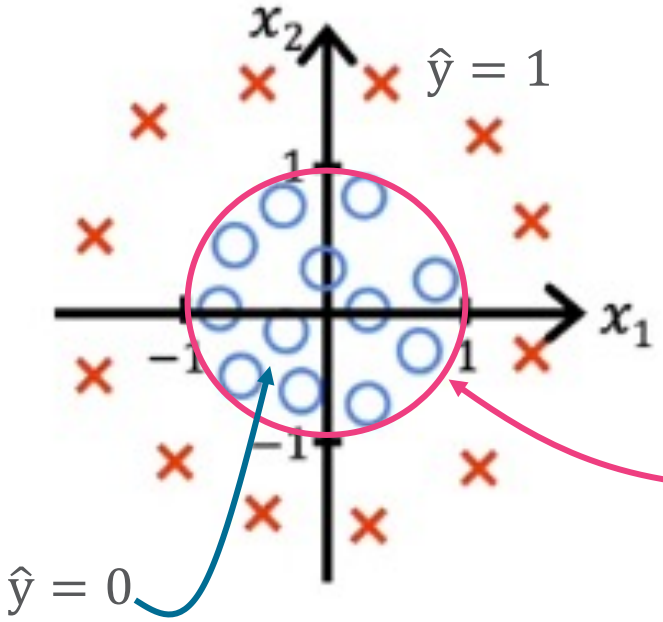
- Any issues with this coding?
  - Suggests an ordering
  - Implies that the difference between ***stroke*** and ***drug overdose*** is the same as between ***drug overdose*** and ***epileptic seizure***.

Logistic Regression

- Linear regression: $\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_m x_m = X\theta$
- Logistic regression:
  - $z = X\theta$
  - $\hat{p} = \sigma(z)$
  - $\sigma(z) = \frac{1}{1+e^{-z}} \Rightarrow \hat{p} = \frac{1}{1+e^{-X\theta}}$
  - $\hat{y} = \begin{cases} 1, \hat{p} \geq 0.5 \\ 0, \hat{p} < 0.5 \end{cases}$



Geometry of Logistic Regression



$$\sigma(z) = \sigma(\theta_0 + \theta_1 x_1^2 + \theta_2 x_2^2)$$
$$= \sigma(-1 + (1)x_1^2 + (1)x_2^2)$$

$\hat{y} = 1$:

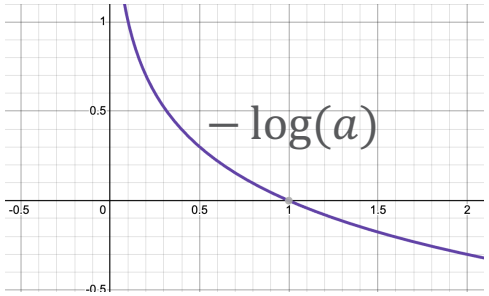
$$z = -1 + x_1^2 + x_2^2 = 0 \Rightarrow$$
$$x_1^2 + x_2^2 = 1$$

https://ttpoll.com/p/911058

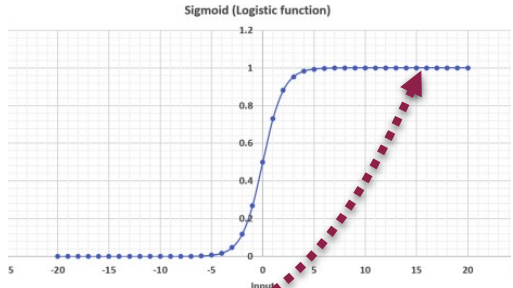
Loss Function Intuition

$$\mathcal{L}(\hat{y}, y) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

- If $y = 1$: $\mathcal{L}(\hat{y}, 1) = -\big((1) \log(\hat{y}) + (1 - 1) \log(1 - \hat{y})\big) = -\log(\hat{y})$



As $\hat{y} \to 1, \mathcal{L}(\hat{y}, 1) \to 0$

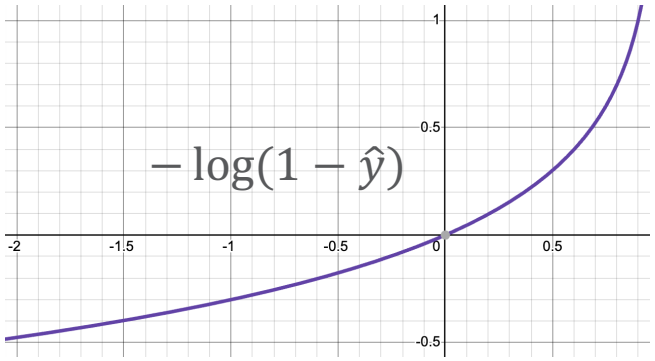


As $\hat{y} \to 0, \mathcal{L}(\hat{y}, 1) \to \infty$

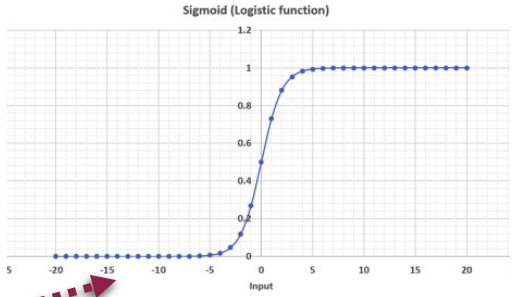
Loss Function Intuition

$$\mathcal{L}(\hat{y}, y) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

- If $y = 0$: $\mathcal{L}(\hat{y}, 0) = -\big((0) \log(\hat{y}) + (1 - 0) \log(1 - \hat{y})\big) = -\log(1 - \hat{y})$



$As\ \hat{y} \to 0, \mathcal{L}(\hat{y}, 0) \to 0$



$As\ \hat{y} \to 1, \mathcal{L}(\hat{y}, 0) \to \infty$

https://ttpoll.com/p/817711

Training, Validation, and Test Sets

- **Training set:** samples drawn from $f(x, y)$ used to train/adjust the *parameters* in model $h(x)$.
- **Validation set:** samples drawn from $f(x, y)$ used to evaluate model performance and adjust the *hyperparameters* in model $h(x)$.
- **Test set:** samples drawn from $f(x, y)$ used to evaluate the final model with unseen data.

Practical Advice on Data Splits

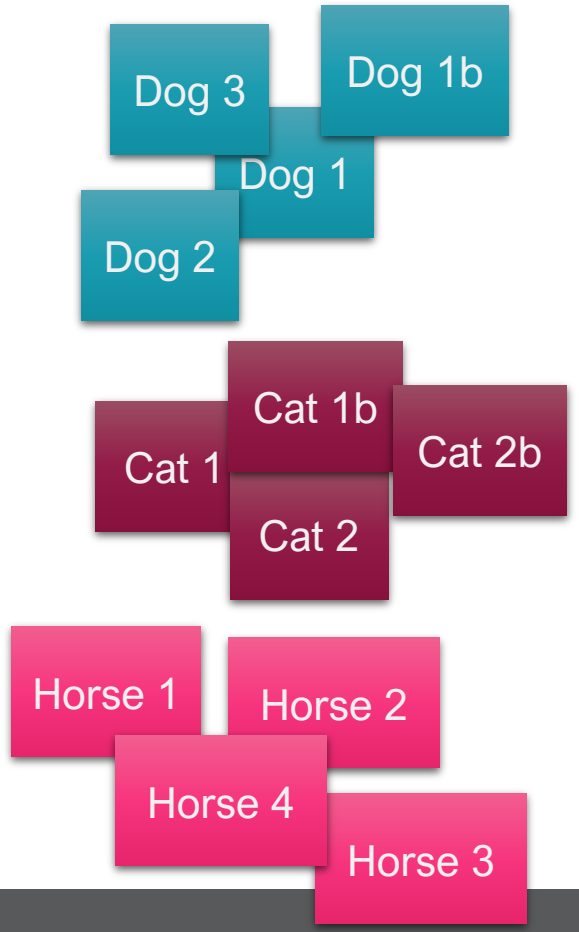
- Most times, random sampling works fine unless…
  - Unbalanced classes – Stratified split
  - Differences in the data (e.g., quality)
- Typical splits {Training, Validation, Testing}
  - {60, 20, 20}, {70,15,15}, {80,10,10}
  - Validation and testing set splits are about adequate data representation
- Avoid data leakage
  - E.g., time series data split chronologically
  - E.g., instances of the same sample assign to same set.

https://ttpoll.com/p/817711

Random Sampling

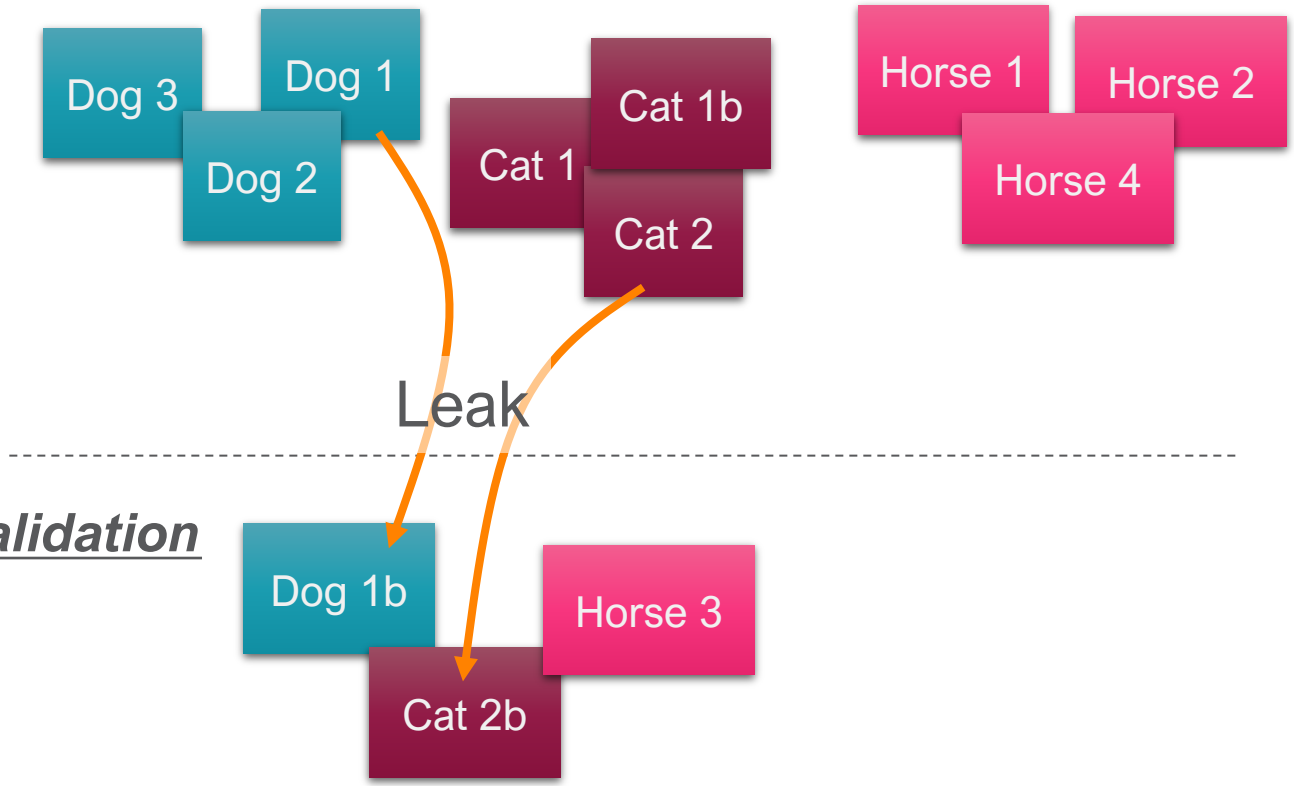
Data Leakage

Dataset



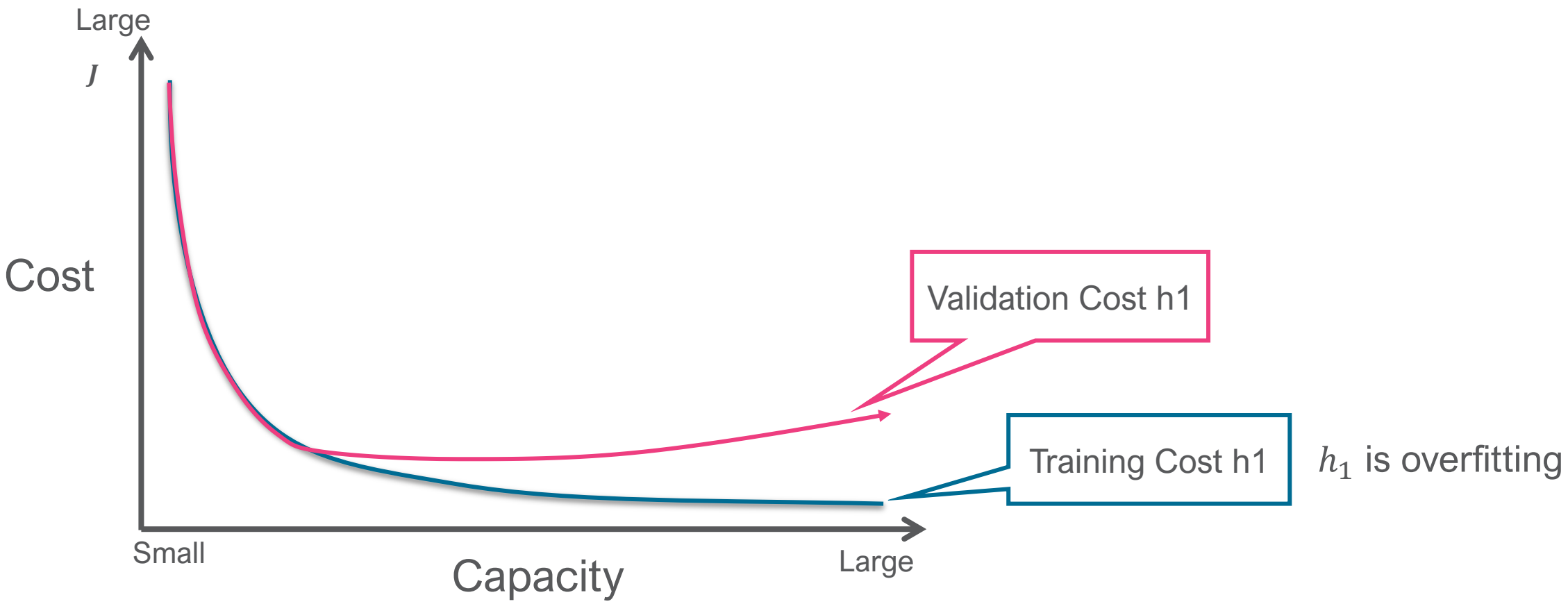
Training

Validation



https://ttpoll.com/p/817711

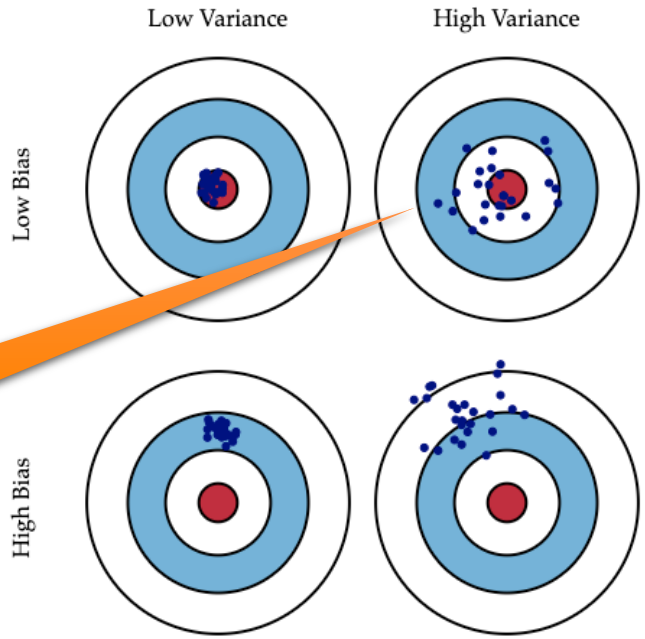
Overfitting and Underfitting

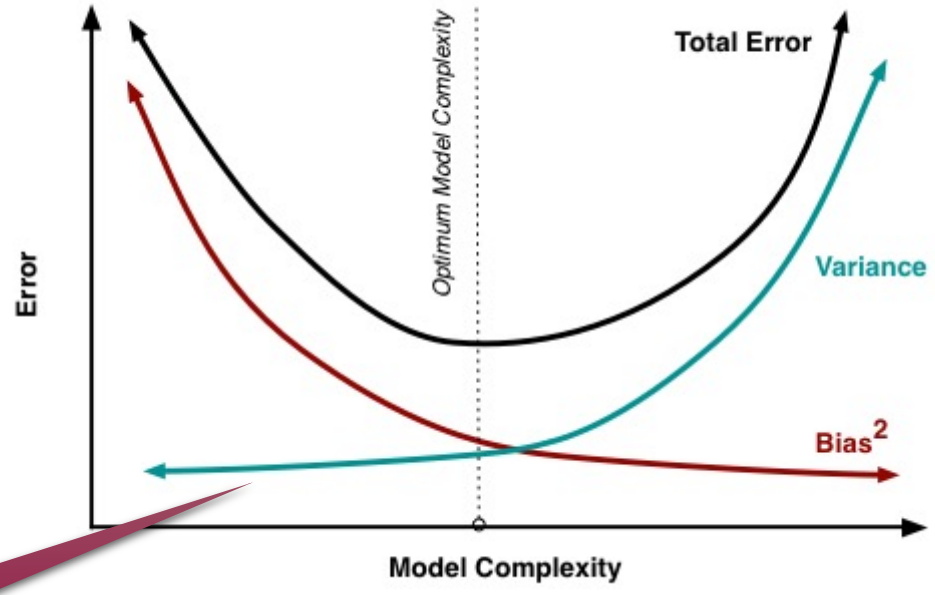


https://ttpoll.com/p/548136

Bias and Variance

- Sources of error
  - Bias
  - Variance
  - Irreducible Error





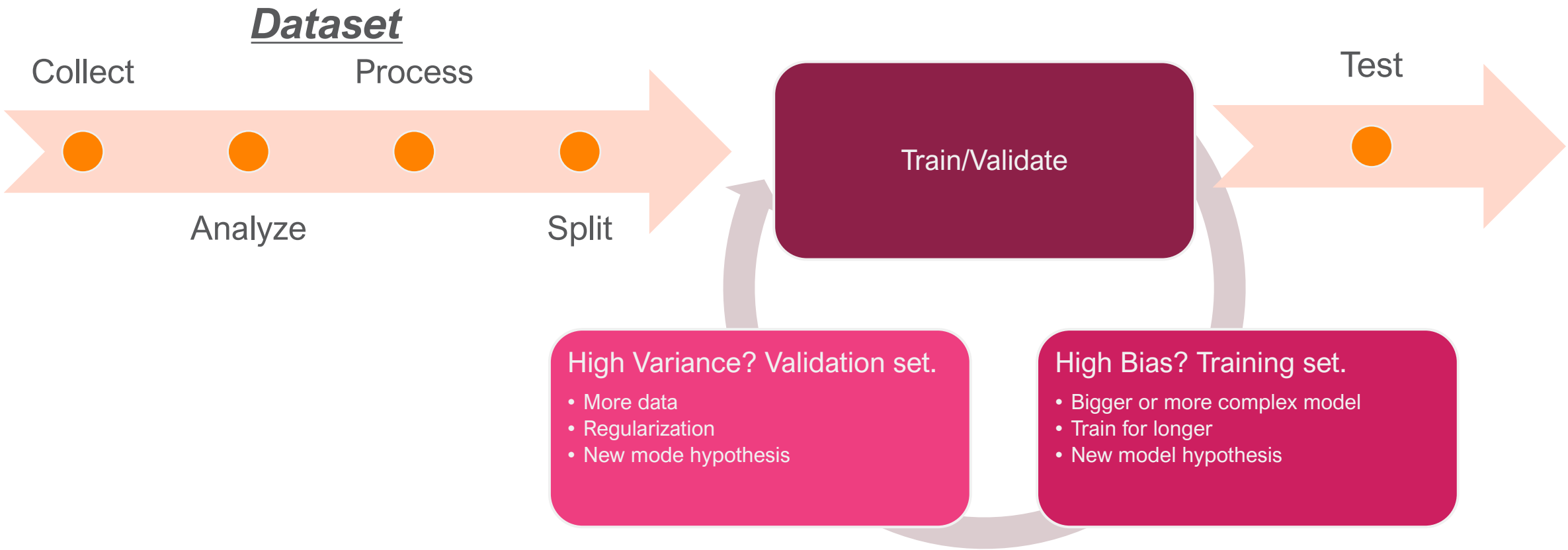
Should we optimize for low bias?

No, both are equally important.

Source figures: https://scott.fortmann-roe.com/docs/BiasVariance.html

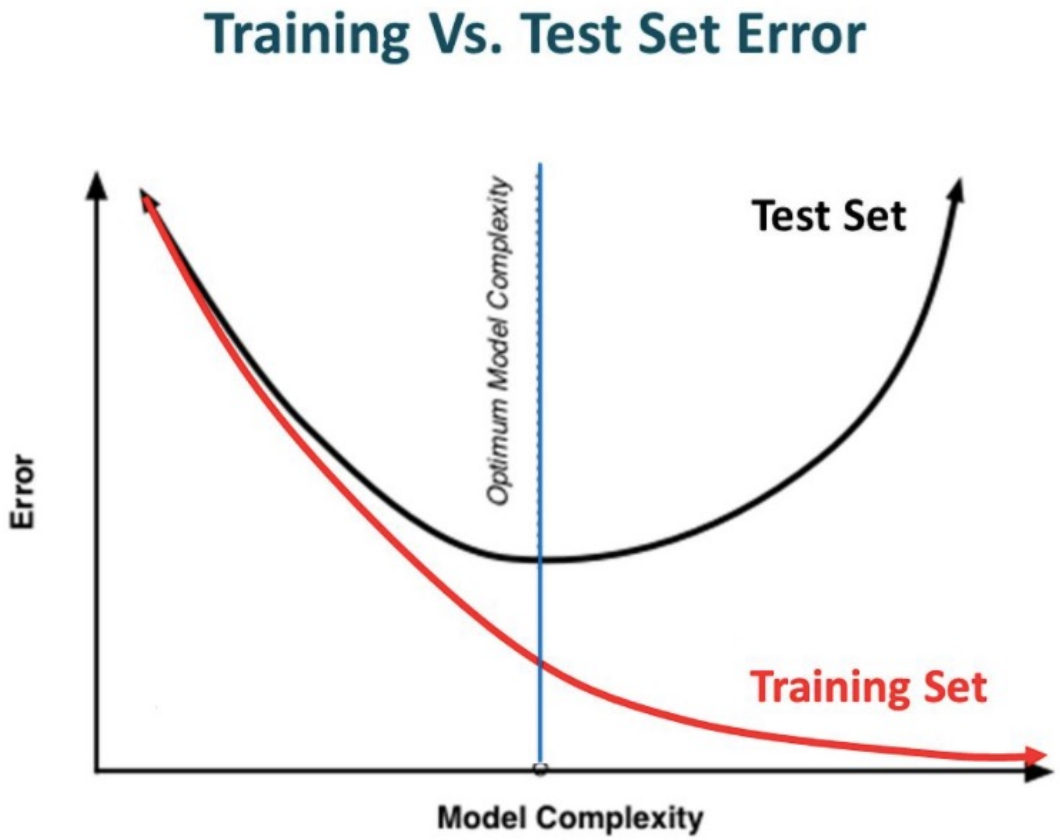
https://ttpoll.com/p/548136

So far

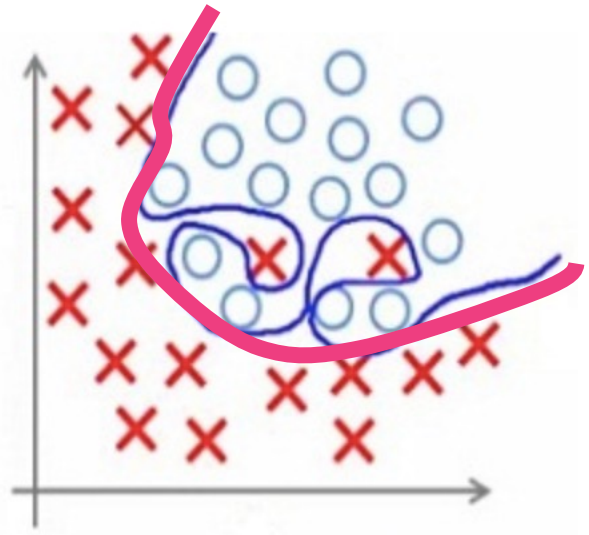


https://ttpoll.com/p/548136

Regularization and Model Error



Regularization



Good-fitting

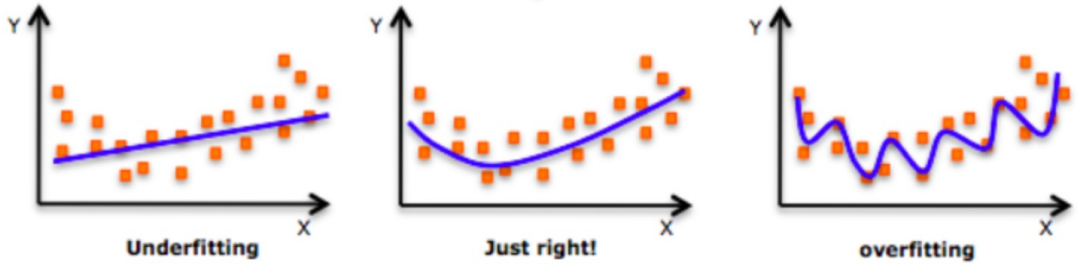


Image source: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

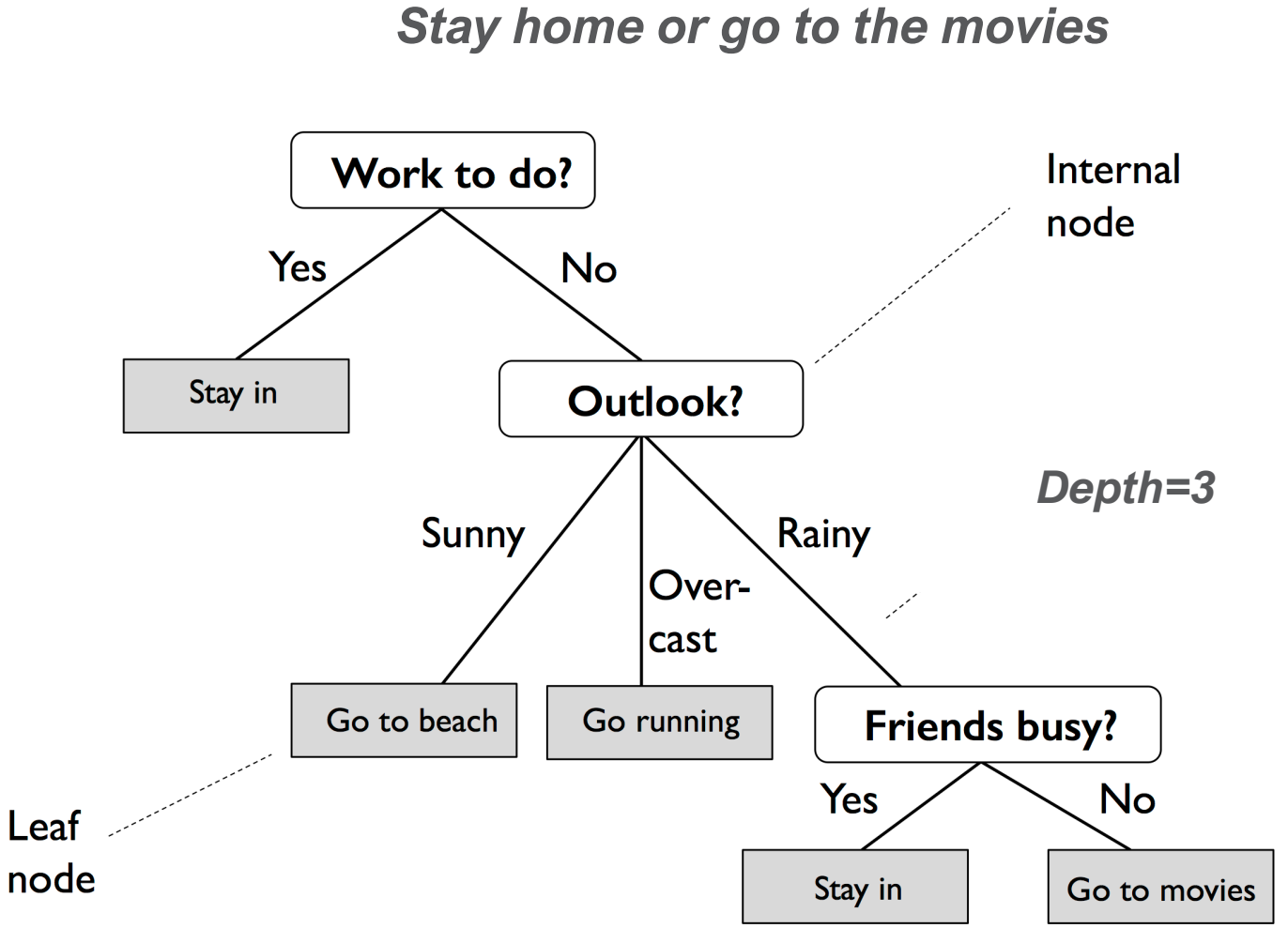
https://ttpoll.com/p/548136

Popular Regularization/Penalty Terms

Technique	Formula	Type	Effect	Common use cases
Ridge (L2)	$\frac{\lambda}{2}\sum_{i=1}^{m} w_j^2 = w^T w$	Penalizes squared weights	Rewards smaller weights, smoother transitions.	Linear/Logistic Regression, Neural Networks
Lasso (L1)	$\lambda \sum_{j=1}^{m} \lvert w_j \rvert$	Penalizes absolute weights	Rewards sparsity (feature space reduction)	High-dimensional data
ElasticNet	$\frac{\lambda_1}{2}\lVert w \rVert_2^2 + \lambda_2 \lVert w \rVert_1$	Combines Ridge and Lasso	Balances sparsity (L1) and smoothness (L2)	High-dimensional data with correlated features
Early Stopping	N/A	Stops training after specified cost event.	Prevents overfitting by using an earlier checkpoint.	Neural Networks

Decision Trees

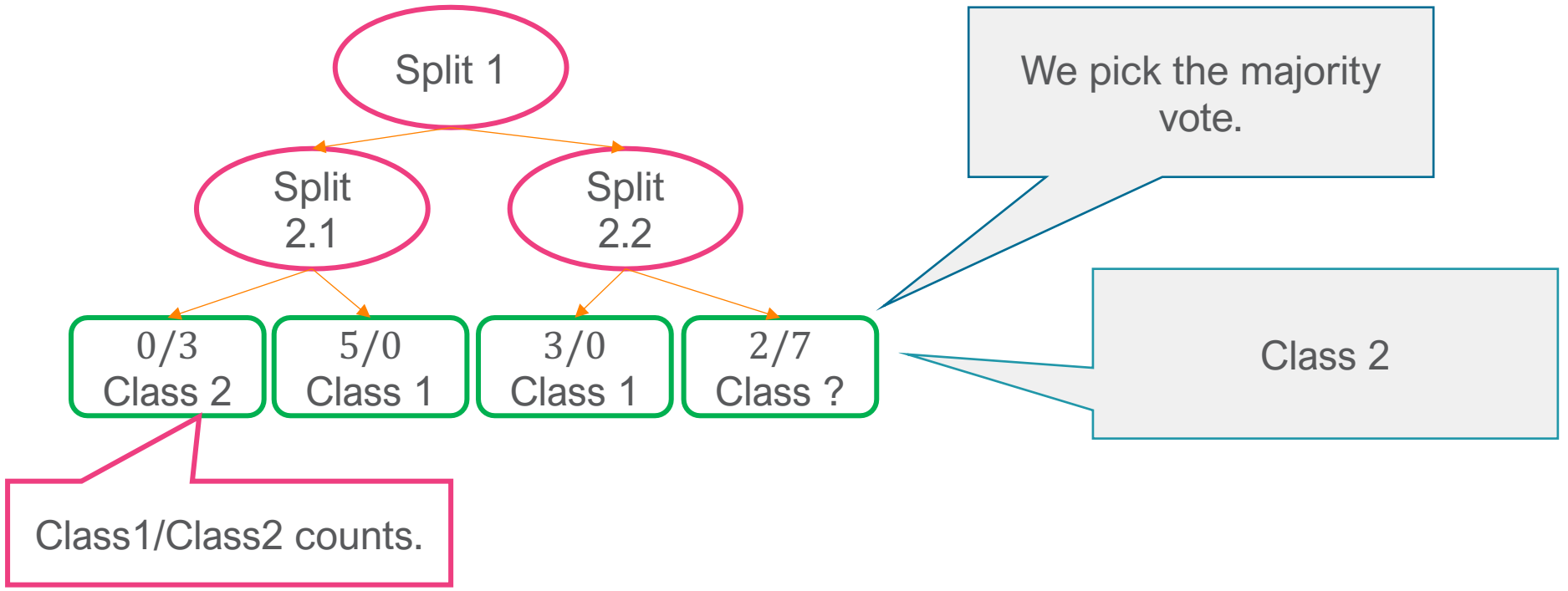
- Iterative top-down creatio of hypothesis (Classifier)
- Hierarchy of decisions
  - We ask questions to split th dataset.
- Highly explainable



Source: Dr. Raschka, Machine Learning Book

https://ttpoll.com/p/643115

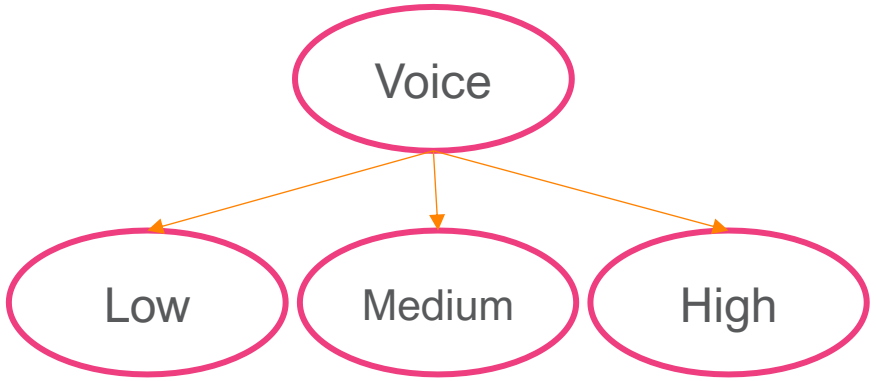
How to handle decisions at non-pure leaf nodes?



https://ttpoll.com/p/643115

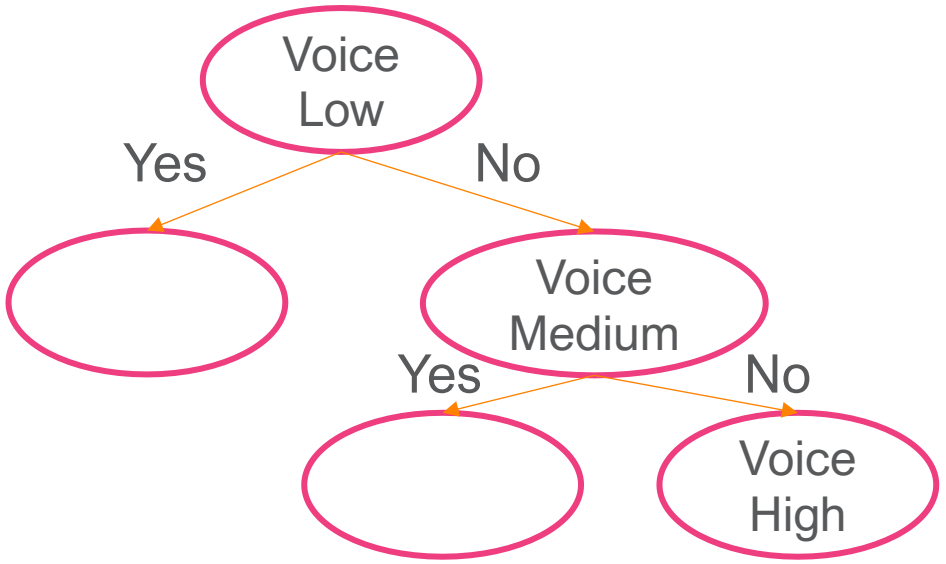
Binary vs Categorical

Voice Pitch: {Low, Medium, High}



=

One-Hot Encoding

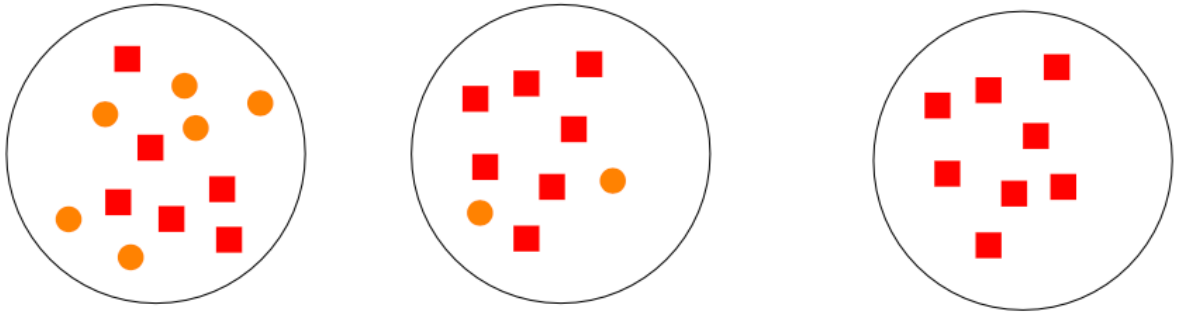


Binary trees are more efficient.

Similar to linear regression: Add a new feature per category (i.e., new columns in X).

When to stop growing the tree?

- Node is pure
  - Leaf node contains only examples of the same class
- x_j feature values are the same for all examples
- Statistical significance test
  - E.g., Chi-Square: Are parent and child class distributions significantly different?



https://ttpoll.com/p/643115

Choosing the *"best"* attribute

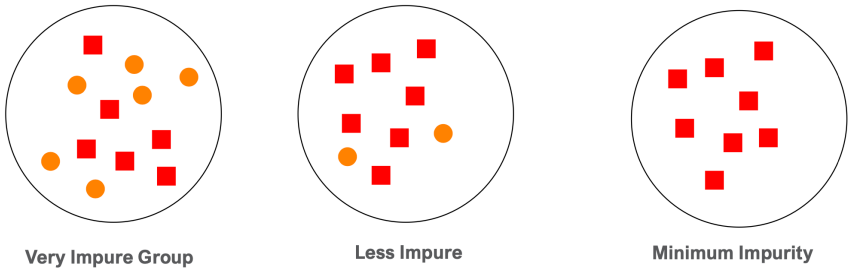
- **Key problem:** choosing which attribute to split a given set of examples
- Some possibilities are:
  - Random: Select any attribute at random
  - Least-Values: Choose the attribute with the smallest number of possible values
  - Most-Values: Choose the attribute with the largest number of possible values
  - Max-Gain: Choose the attribute that has the largest expected information gain
    - i.e., the attribute that results in the smallest expected size of the subtrees rooted at its children

Slide Credit: Dr. Schumman

Information Gain

If we have a ***delta*** between the parent node impurity and the child nodes cumulative impurity, we ***gain information***.

$$IG(D_p, V) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$



V: Feature to split

D_p: dataset of parent node

D_j: dataset of child node j

I: Impurity measurement

N_p: Number of training examples for parent node

N_j: Number of training examples for child node j

m: Number of child nodes

Impurity Metrics

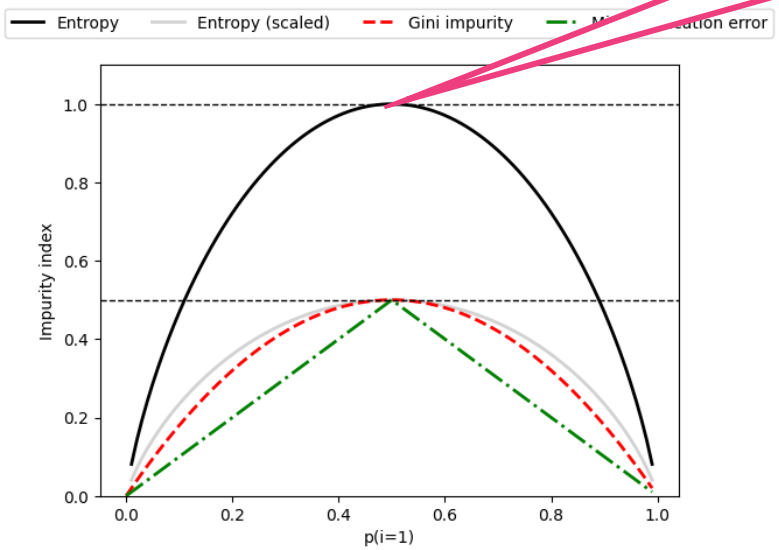
- Entropy (I_H):
  - Attempts to maximize mutual information.
  - How much knowledge about y we gain from knowing split D_j?
- Gini (I_G):
  - Minimizes the probability of misclassification
  - Produces very similar results to Entropy.
- Classification Error (I_E):
  - Less sensitive to changes in the node class distribution
  - Useful when pruning the tree

Entropy (I_H) - Shannon

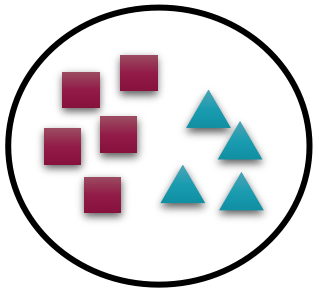
- From information theory—the higher the entropy the more information.

$$I_H = -p\log_2(p) - (1-p)\log_2(1-p)$$

p: Proportion of the samples in node t that belong to the positive (1) class.



t



$$p = \frac{5}{9} = 0.56$$

$$I_H = -(0.56)\log_2(0.56) - (1-0.56)\log_2(1-0.56)$$
$$= 0.468 + 0.521$$
$$= \mathbf{0.99}$$

Conditional Entropy

$$I_H(D|V) = \sum_{v \in V} p(V = v) I_H(D|V = v)$$

$\frac{N_j}{Np}$

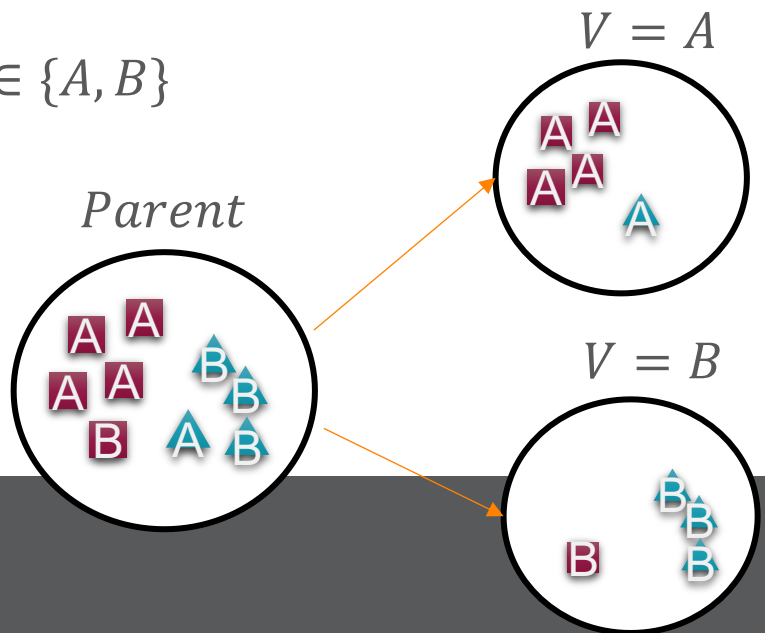
Computed in the previous slide.

$$I_H(D|V = A) = -\frac{4}{5}\log_2\left(\frac{4}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right) = 0.72$$

$$I_H(D|V = B) = -\frac{1}{4}\log_2\left(\frac{1}{4}\right) - \left(\frac{3}{4}\right)\log_2\left(\frac{3}{4}\right) = 0.81$$

$$I_H(D|V) = \frac{5}{9}(0.72) + \frac{4}{9}(0.81) = \mathbf{0.76}$$

$V = x_j \in \{A, B\}$

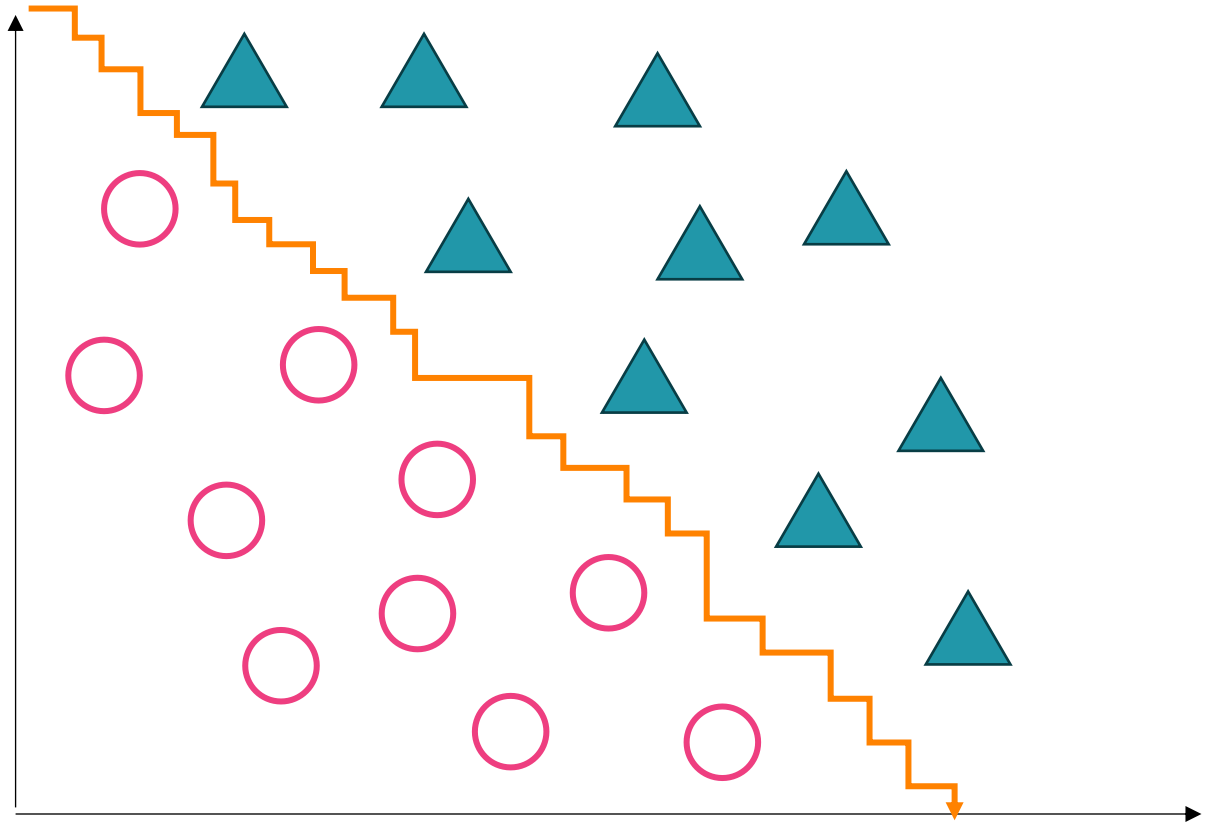


Computed on the parent node.

$p(V = A) = 5/9$

$p(V = B) = 4/9$

Diagonal Boundaries



Tree will become too large.

Slide credit: Dr. Raschka

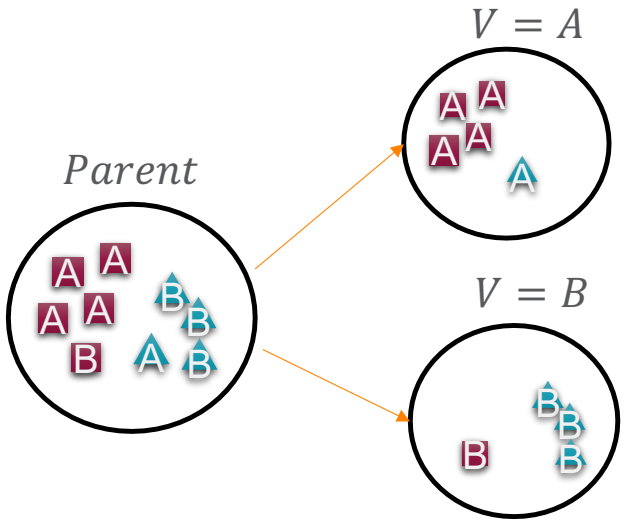
Mutual Information

Mutual Information (I) is the amount of information that one random variable Y contains about another random variable X.

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

$$H(X,Y) = H(Y) + H(X|Y) \Rightarrow$$

$$\boldsymbol{I(X,Y) = H(X) - H(X|Y)}$$

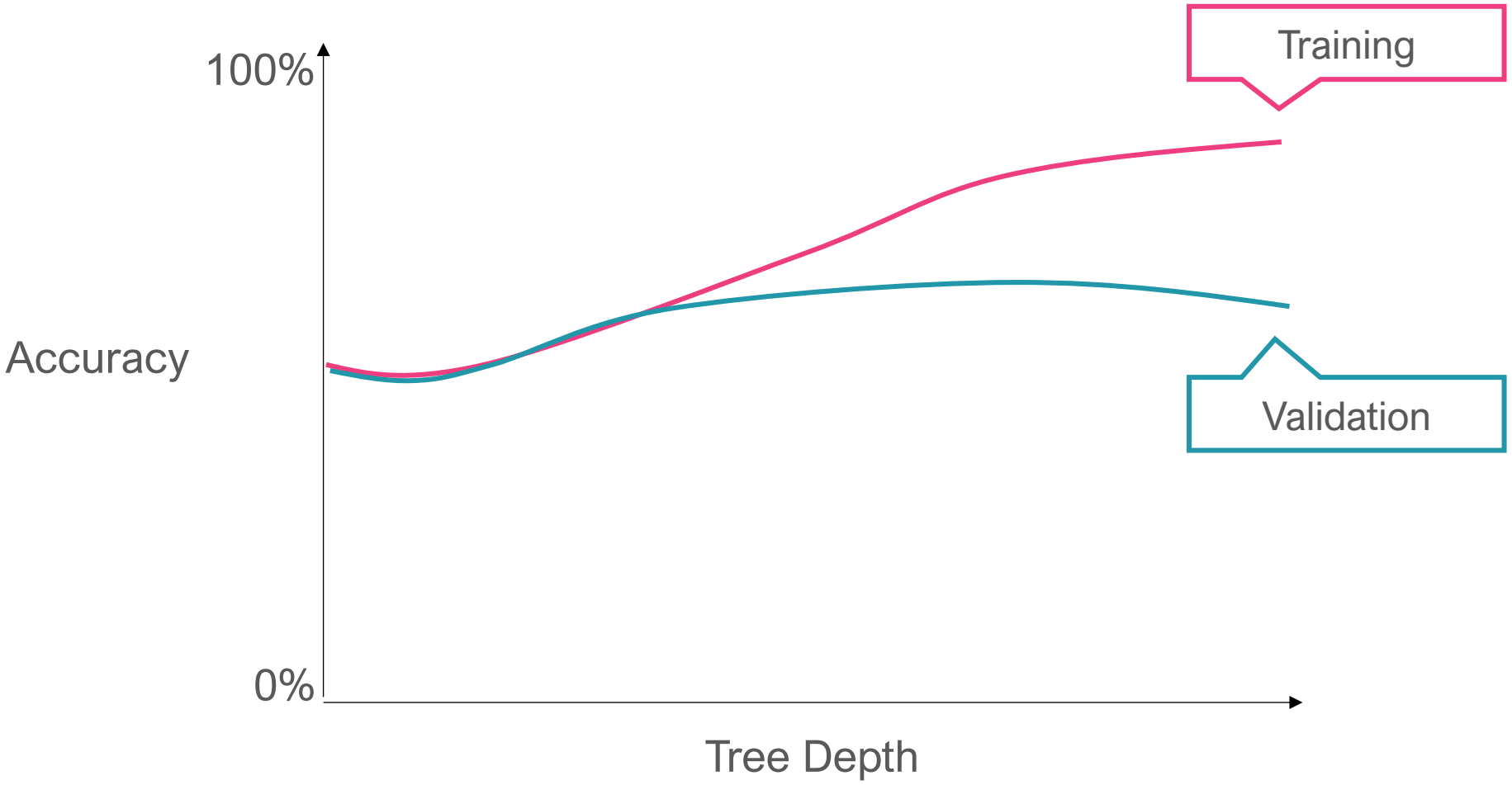


$$I_H(D,V) = I_H(D) - I_H(D|V) = I_H(D) - \sum_{v \in V} p(V = v) I_H(D|V = v) = 0.99 - 0.76 = 0.23$$

This is our information gain.

https://ttpoll.com/p/643115

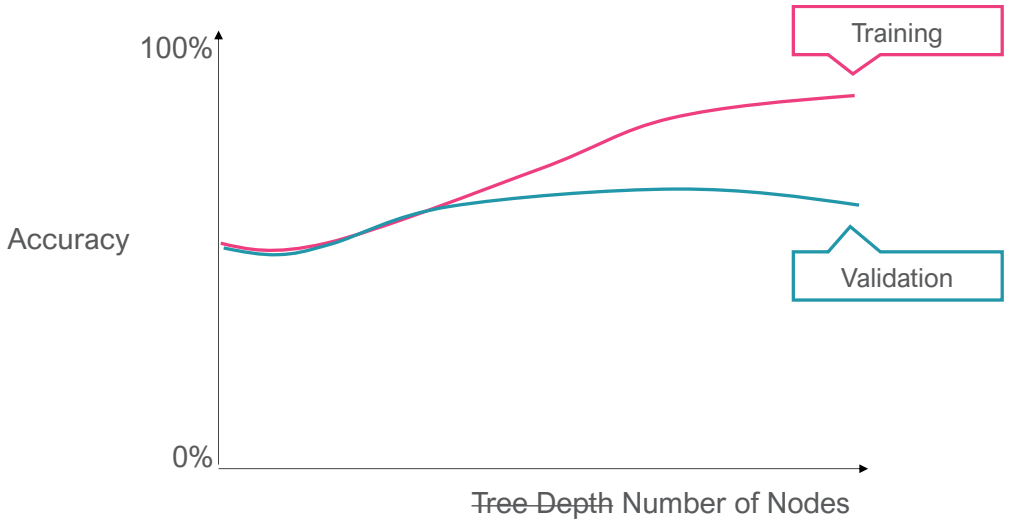
Overfitting



Slide credit: Dr. Raschka

Post-Pruning (After Training)

- Acquire more training data
- Grow full tree first, then remove nodes
- ***Reduced-error pruning:*** remove nodes via validation set evaluation
  - Requires a test set
  - Greedily remove node that most improves validation set accuracy



Slide credit: Dr. Raschka

Pre-Pruning (Before we grow tree)

- Set a depth cut-off (maximum tree depth)
- Cost-complexity pruning, where we set a total number of nodes.
- Stop growing if split is not statistically significant
  - (e.g., χ^2 test)
- Set a minimum number of data points for each node
  - Addresses labeling errors
- Remove irrelevant attributes

Slide credit: Dr. Raschka

Review

- (+) Easy to interpret and communicate
- (+) Can represent "complete" hypothesis space
- (-) Easy to overfit
- (-) Elaborate pruning required
- (-) Expensive to just fit a "diagonal line"
- (-) Output range is bounded in regression trees by input range.