Machine Learning

Laura Brown

UN5550

Overview

- What is Machine Learning?
- Supervised Learning
  - K Nearest Neighbors
  - Decision Trees

Feel free to stop me to ask questions,
I would like a dialogue with you.

Machine Learning

- Wikipedia: “ML is the scientific study of algorithms and statistical models that that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.”

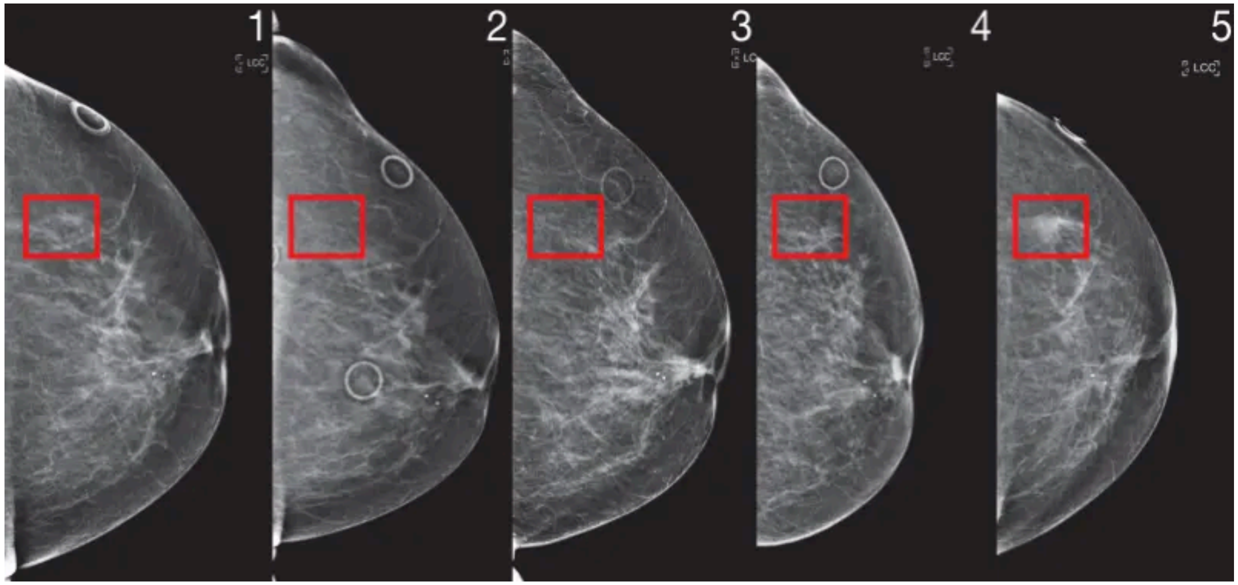
Enabling Trends

- Progress in algorithms and theory
- Flood of data: BIG DATA!
- Computational power is cheap
- High demand by industry / researchers / governments

Three Niches of ML

- Data mining: using historical data to improve decisions
  - Medical records to medical knowledge
- Software applications that are difficult to program (or solve) by hand
  - Self-driving cars
  - Image classification
- User modeling
  - Automatic recommender systems (Netflix, Amazon, etc.)

Data Mining Example



- Mirai – models patient's risk of cancer across multiple future time points
- Given
  - 200,000+ mammograms + clinical data
  - Algorithm can optionally benefit from clinical risk factors like family history, and mammography machine

Yala, et. al, 2021 DOI: 10.1126/scitranslmed.aba4373

Credit Risk Analysis

Customer103: (time=t0)

Years of credit: 9
Loan balance: $2,400
Income: $52k
Own House: Yes
Other delinquent accts: 2
Max billing cycles late: 3
Profitable customer?: ?

Customer103: (time=t1)

Years of credit: 9
Loan balance: $3,250
Income: ?
Own House: Yes
Other delinquent accts: 2
Max billing cycles late: 4
Profitable customer?: ?

...

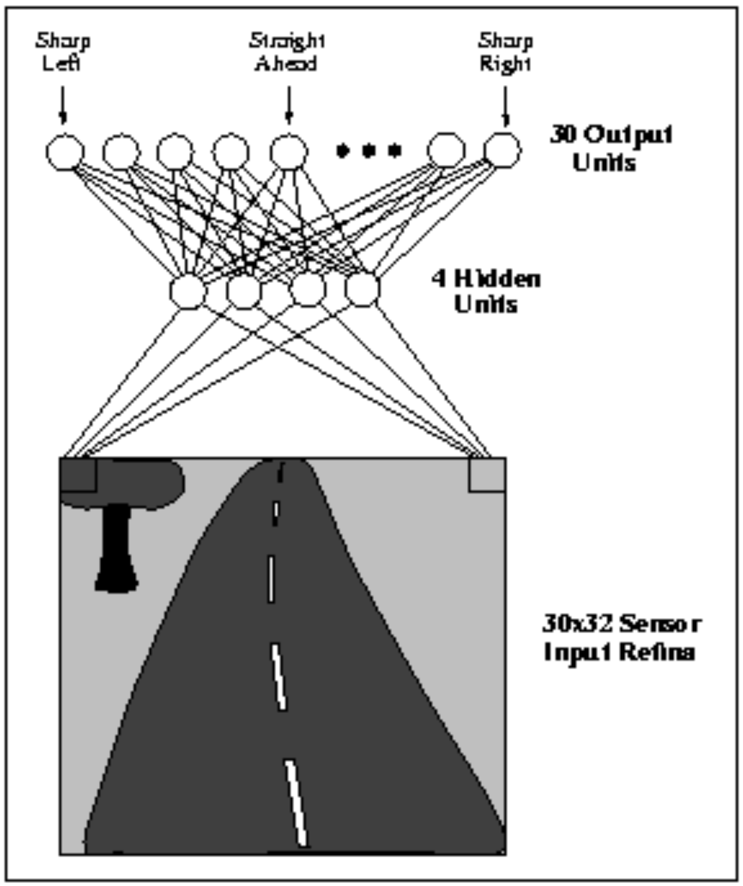
Customer103: (time=tn)

Years of credit: 9
Loan balance: $4,500
Income: ?
Own House: Yes
Other delinquent accts: 3
Max billing cycles late: 6
Profitable customer?: No

- Learned rules
  - IF other-delinquent-account > 2 AND number-delinquent-billing-cycles > 1 THEN profitable-customer = NO
  - IF other-delinquent-account = 0 AND (income > $30K OR years-of-credit > 3) THEN profitable- customer = YES

Difficult to program

- ALVINN drives 70mph on highways



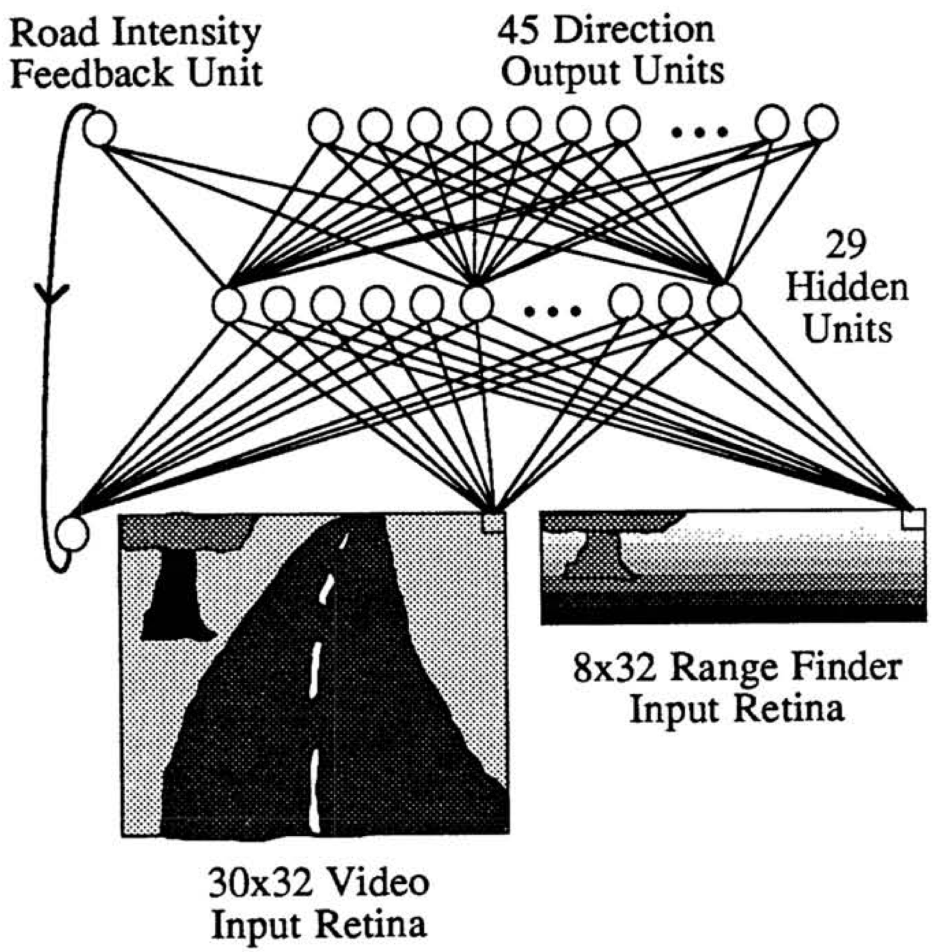
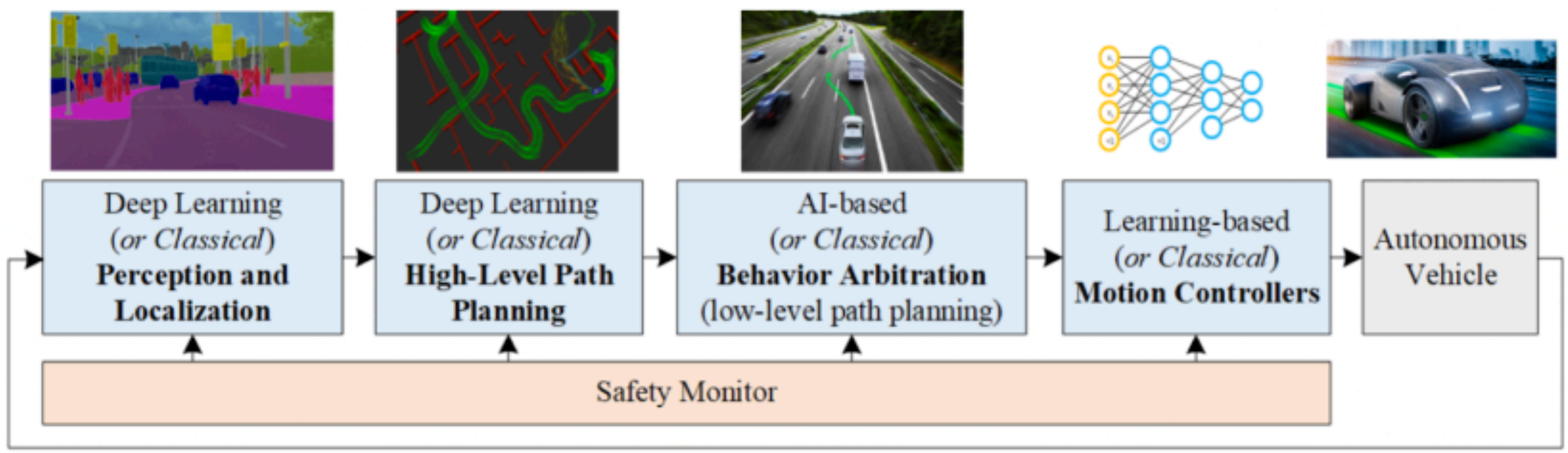
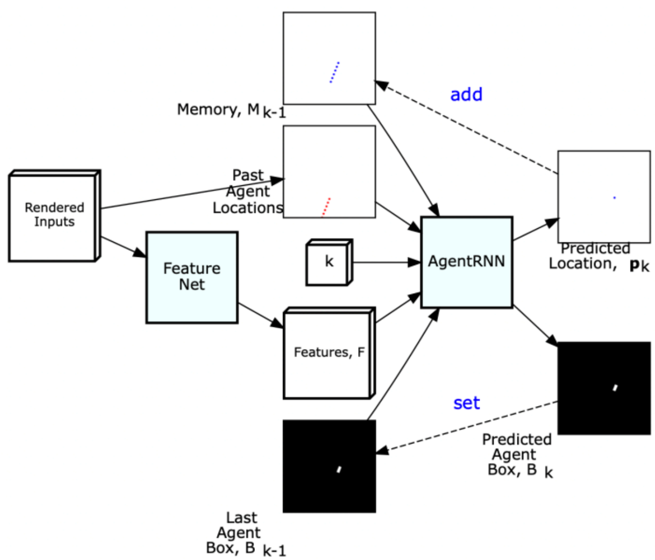


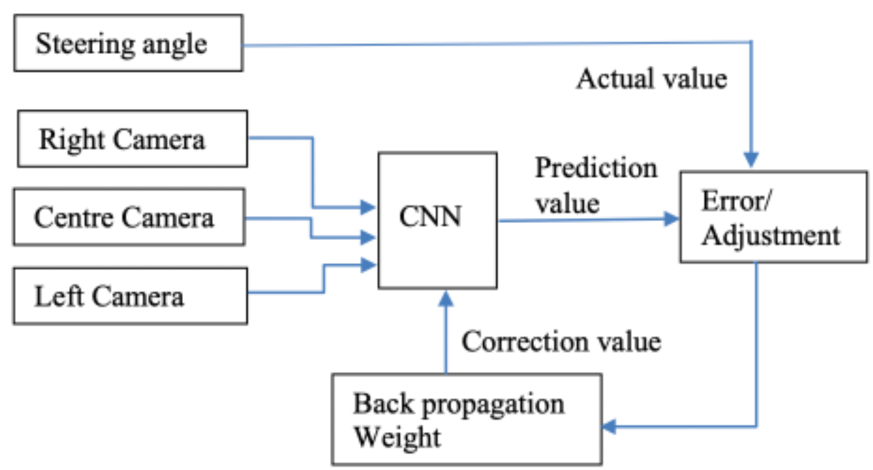
Figure 1: ALVINN Architecture

Historical Work, ALVINN 1989

Modern Systems



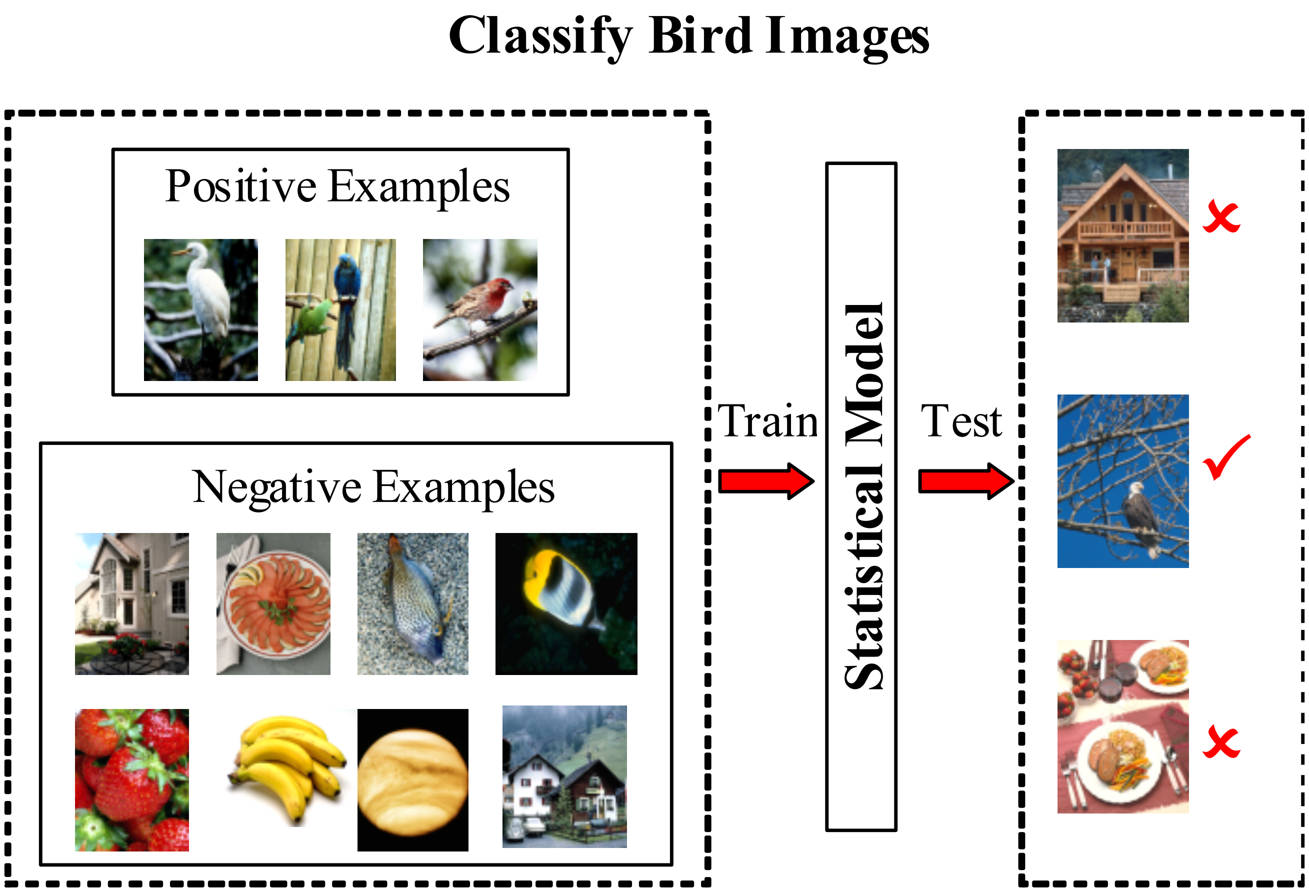




https://neptune.ai/blog/self-driving-cars-with-convolutional-neural-networks-cnn

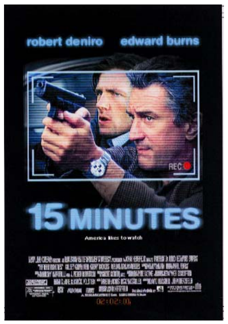
Visual Object Recognition

- Classify bird images



Software the Models Users

History

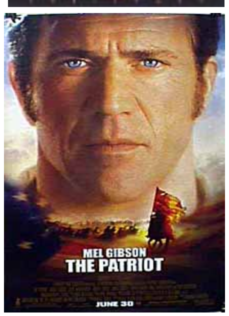


Description: A homicide detective and a fire marshall must stop a pair of murderers who commit videotaped crimes to become media darlings

Rating:

Description: A biography of sports legend, Muhammad Ali, from his early days to his days in the ring

Rating:



Description: Benjamin Martin is drawn into the American revolutionary war against his will when a brutal British commander kills his son.

Rating:

What to Recommend?

Description: A high-school boy is given the chance to write a story about an up-and-coming rock band as he accompanies it on their concert tour.

Recommend: No

Description: A young adventurer named Milo Thatch joins an intrepid group of explorers to find the mysterious lost continent of Atlantis.

Recommend: Yes

Machine Learning Problems

- Supervised Learning
  - Classification / Regression
  - Given a set of example input/output pairs, find a rule to predict output of a new input
- Unsupervised Learning
  - Given set of examples with no labeling, group examples into categories
- Reinforcement Learning
  - Agent interacts with the world receiving occasional rewards or punishments for its actions, learn to choose actions to maximize reward
- Semi-supervised Learning and more …

Example of Supervised Learning

AUTOMATING TINDER WITH EIGENFACES

While my friends were getting sucked into "swiping" all day on their phones with Tinder, I eventually got fed up and designed a piece of software that **automates everything on Tinder**.

An update on Tinderbox: Tinderbox has always been a strictly fun project and I've been glad to share it with the community. From the beginning, I've stated that my support would be very limited. Moving forward, I'd like to openly communicate that I will be dropping any further support for the software. I'm happy that there has been so much interest and support, and moving forward I am putting my energy towards other very promising projects. The code will continue living on Github and feel free to fork it and edit it yourself. Happy Tindering!

http://crockpotveggies.com/2015/02/09/automating-tinder-with-eigenfaces.html

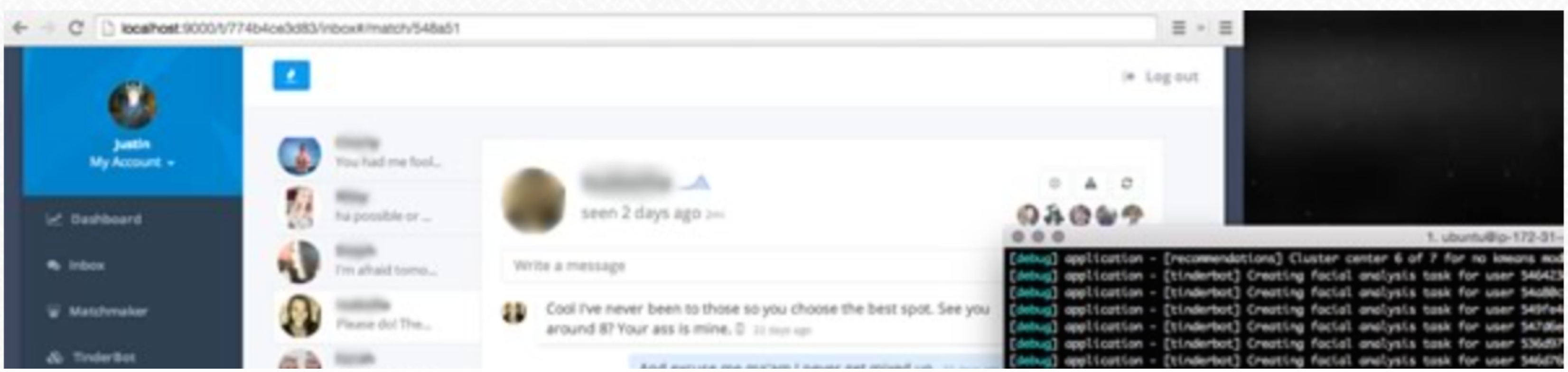


Image Classification Problem

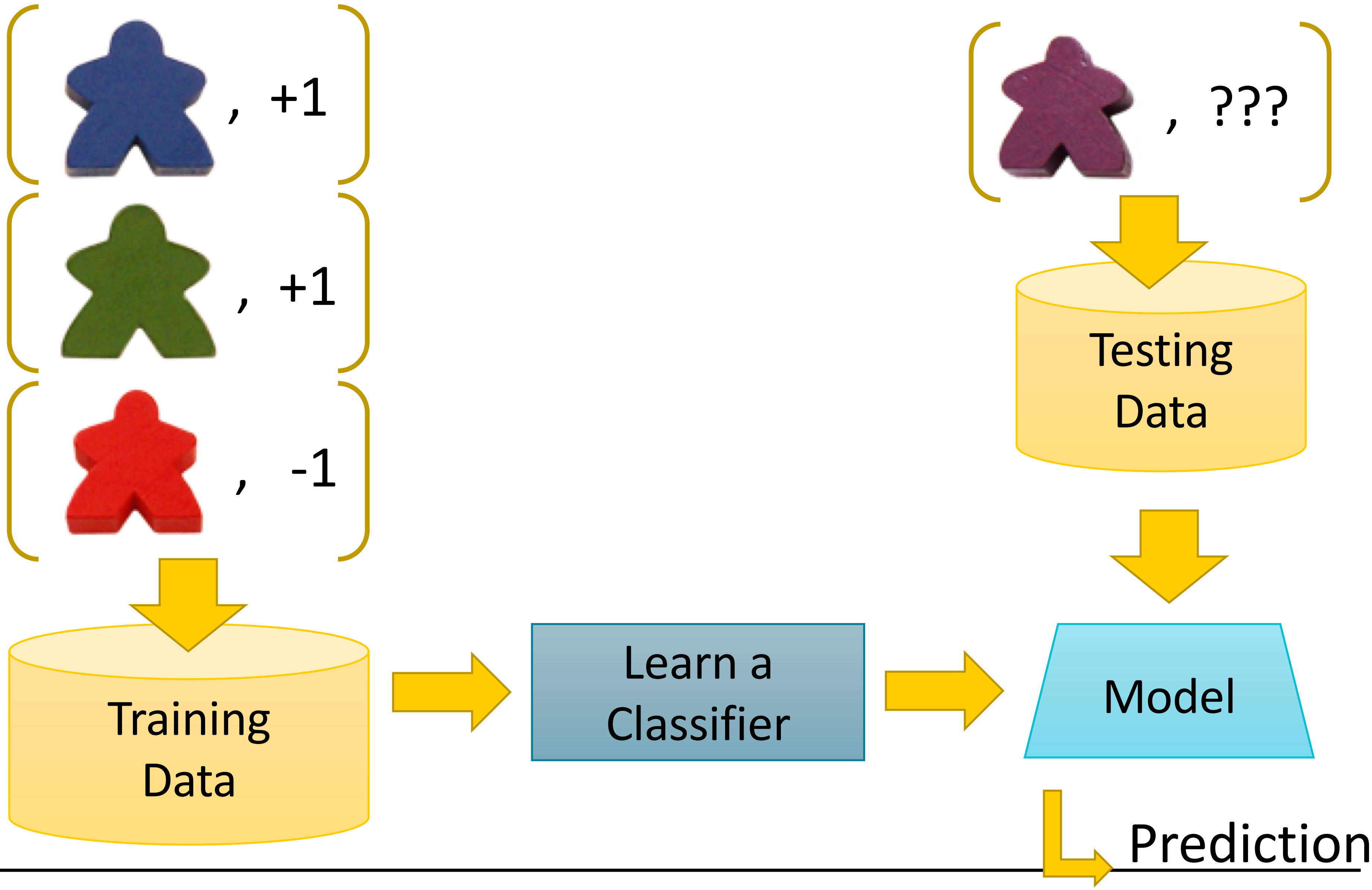
- Image Classification

Which images do you swipe left or swipe right?

- Inputs: $\mathbf{X}$
  - Profile pictures

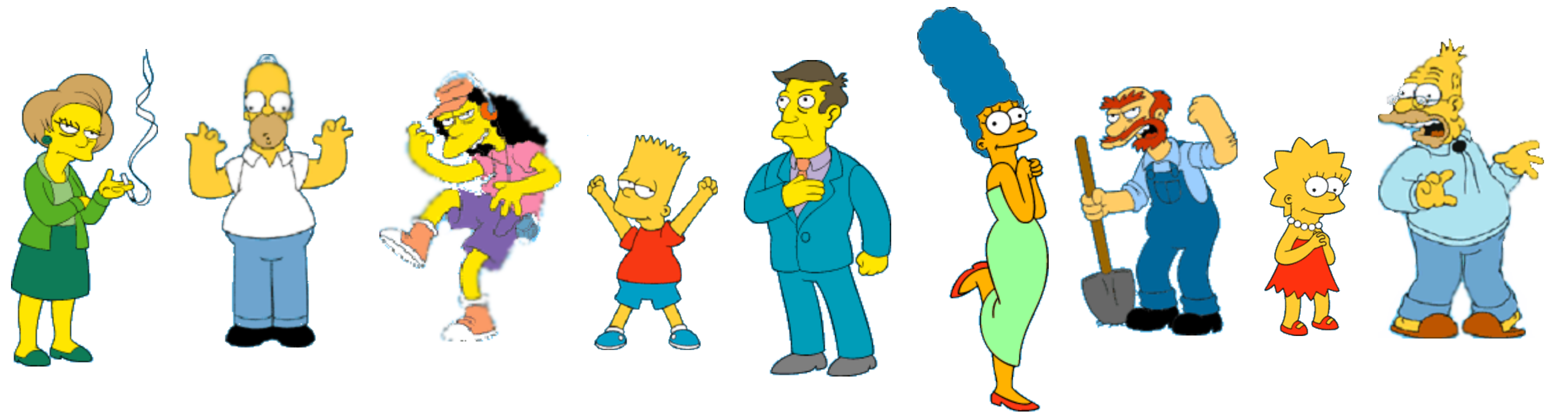
- Class label: $\mathcal{Y}$
  - "Swipe right", +1 "Swipe left", -1

Classification Process



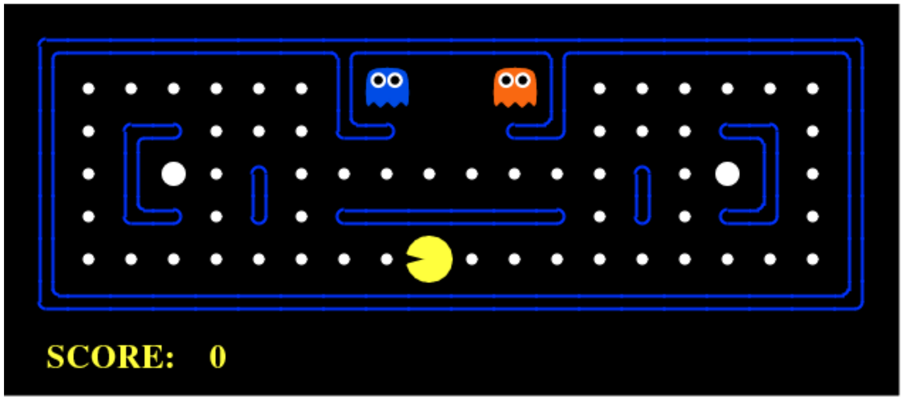
Example of Clustering

What is a natural grouping among these object?



Slide from Eamonn Keogh

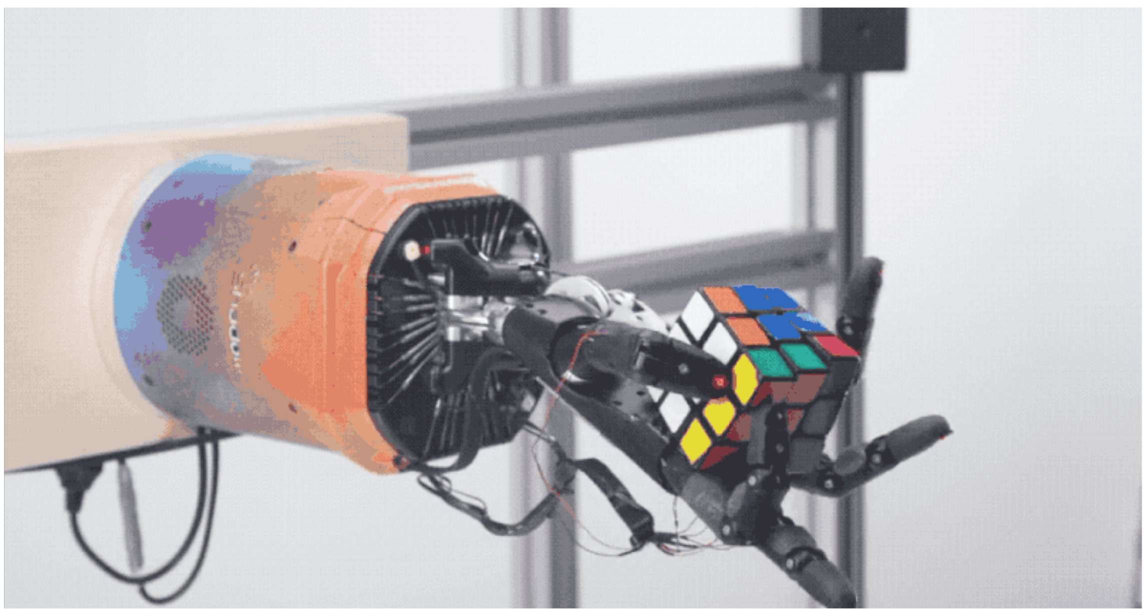
Pacman – Reinforcement Learning



Hide and Seek - RL

https://openai.com/blog/emergent-tool-use/

Solve Rubik's Cube



https://www.theverge.com/2019/10/15/20914575/openai-dactyl-robotic-hand-rubiks-cube-one-handed-solve-dexterity-ai

Classification Problem

- Given a collection of records (training set)

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\} \text{ where}$$

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$$

- Each record contains a list of attributes, and a class label, $y \in \mathcal{Y}$
- Find a model for the class label as a function of the attributes

$$\hat{f}(\boldsymbol{x}) \colon \mathbb{R}^p \longmapsto \mathcal{Y}$$

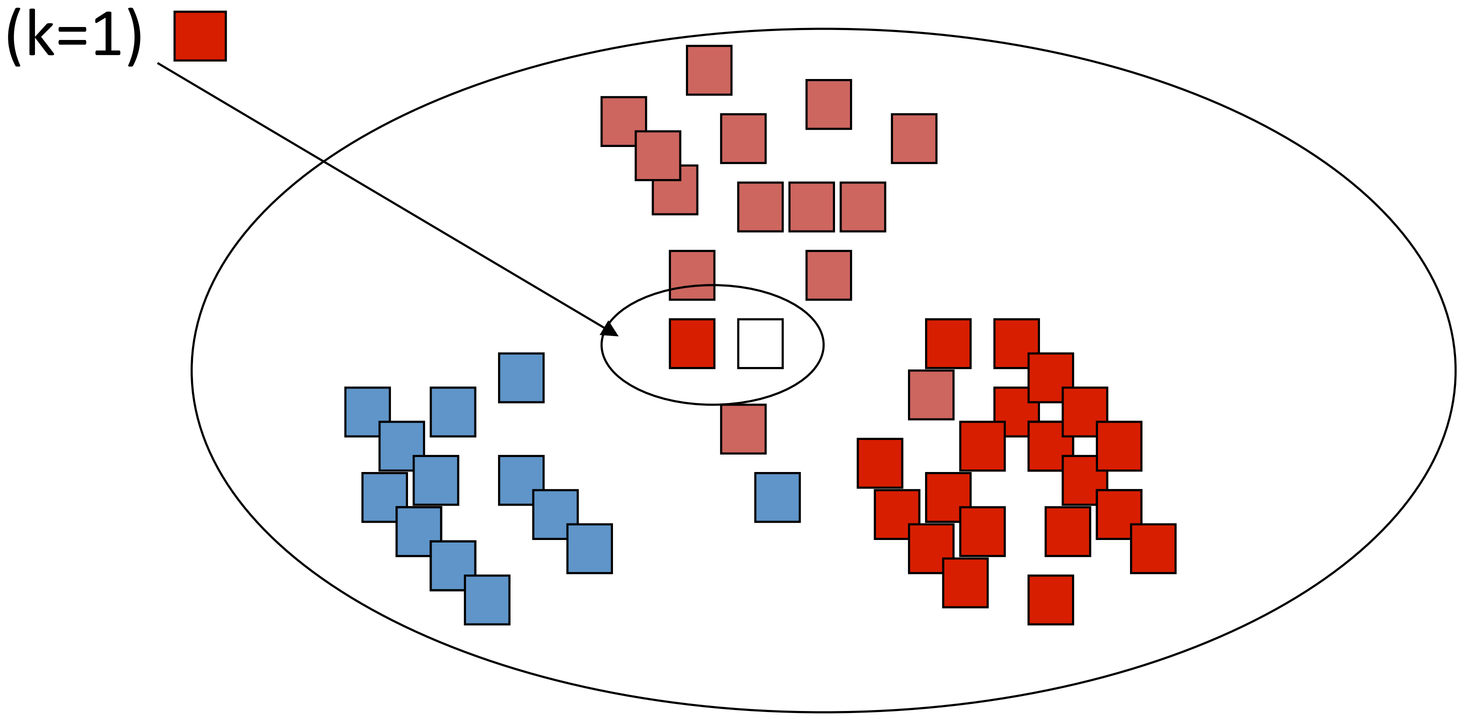
Supervised Learning

- Training Examples

$$\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$$

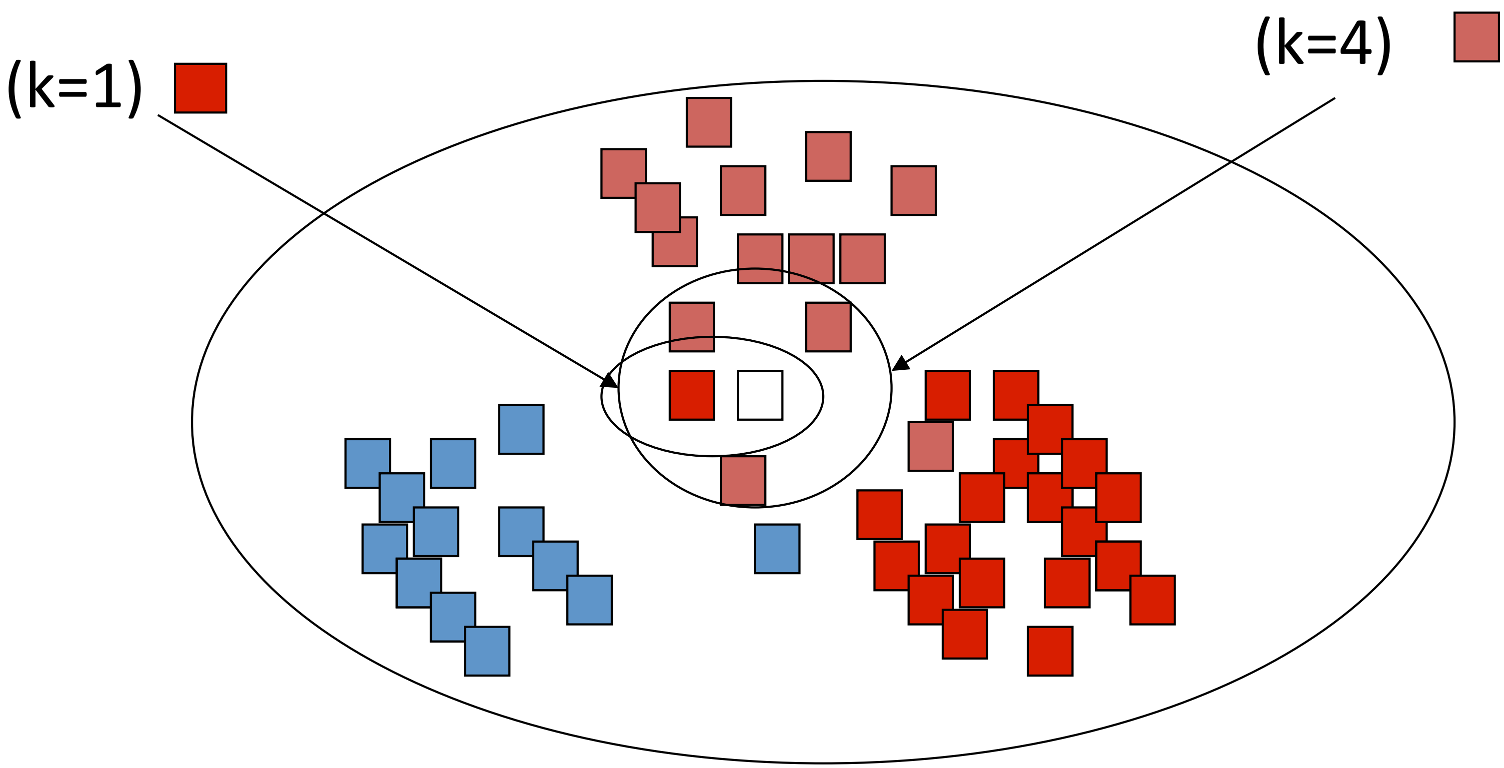
- Identical independent distributed (i.i.d.) assumption
  - Critical for machine learning theory
- Binary classification $\mathcal{Y} = \{-1, +1\}$
- Multi-class classification $\mathcal{Y} = \{1, 2, \ldots, C\}$

K Nearest Neighbor (kNN)



kNN Classifier

How many neighbors should we count ?

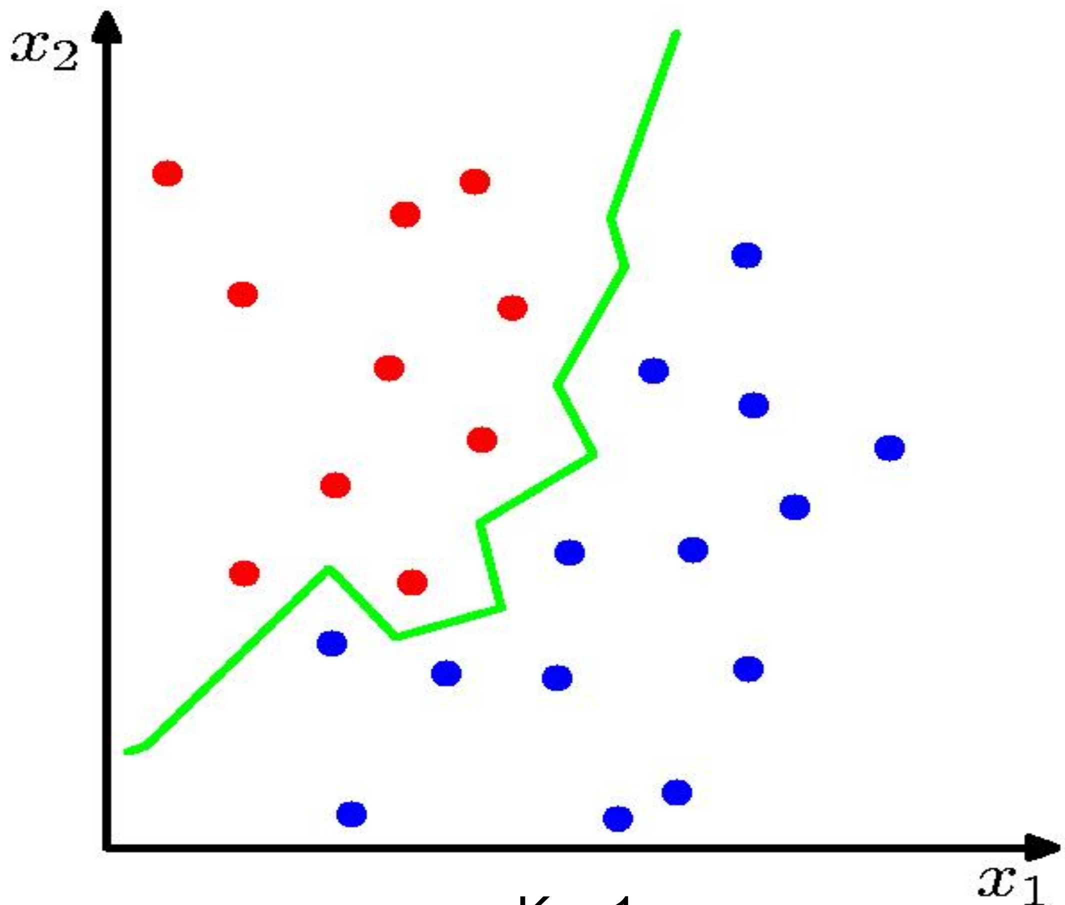


kNN Algorithm

Given: training sample $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,n}$
distance function, *k*, and new input x

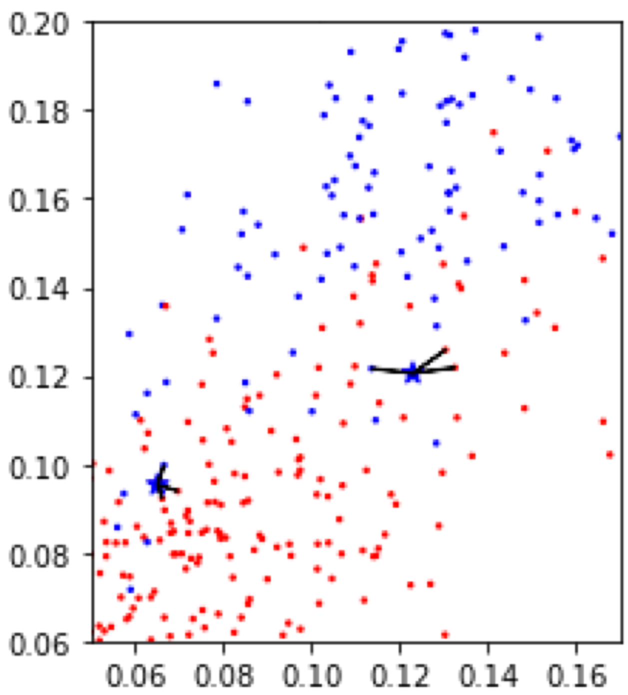
- Find the *k* closest examples with respect to the distance function, $\{j_1, \ldots, j_k\}$
    - Return majority of class labels

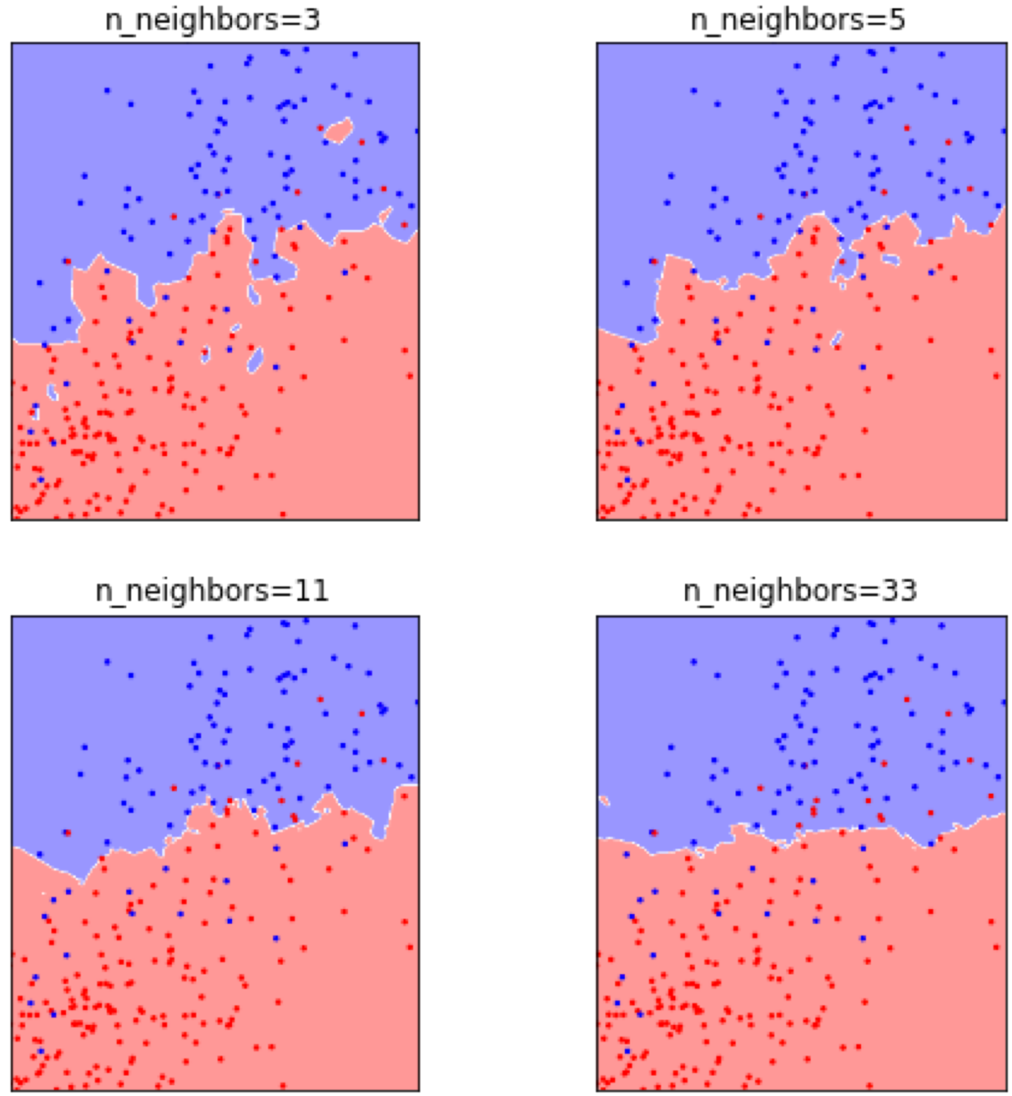
kNN Classifier



K = 1

kNN Classifier





- K acts as a smoother

Generalization and Overfitting

- A model should learn something about the data beyond the specific examples it has been presented (training data)
- The model should be able to predict the correct output for a number of inputs (not only previously seen examples) – this is the property of generalization
- Overfitting: Model is too complex and matches training data well, but not on new data
- Underfitting: Model is too simple and performs poorly overall

Fitting a Line

- Which hypothesis to select?

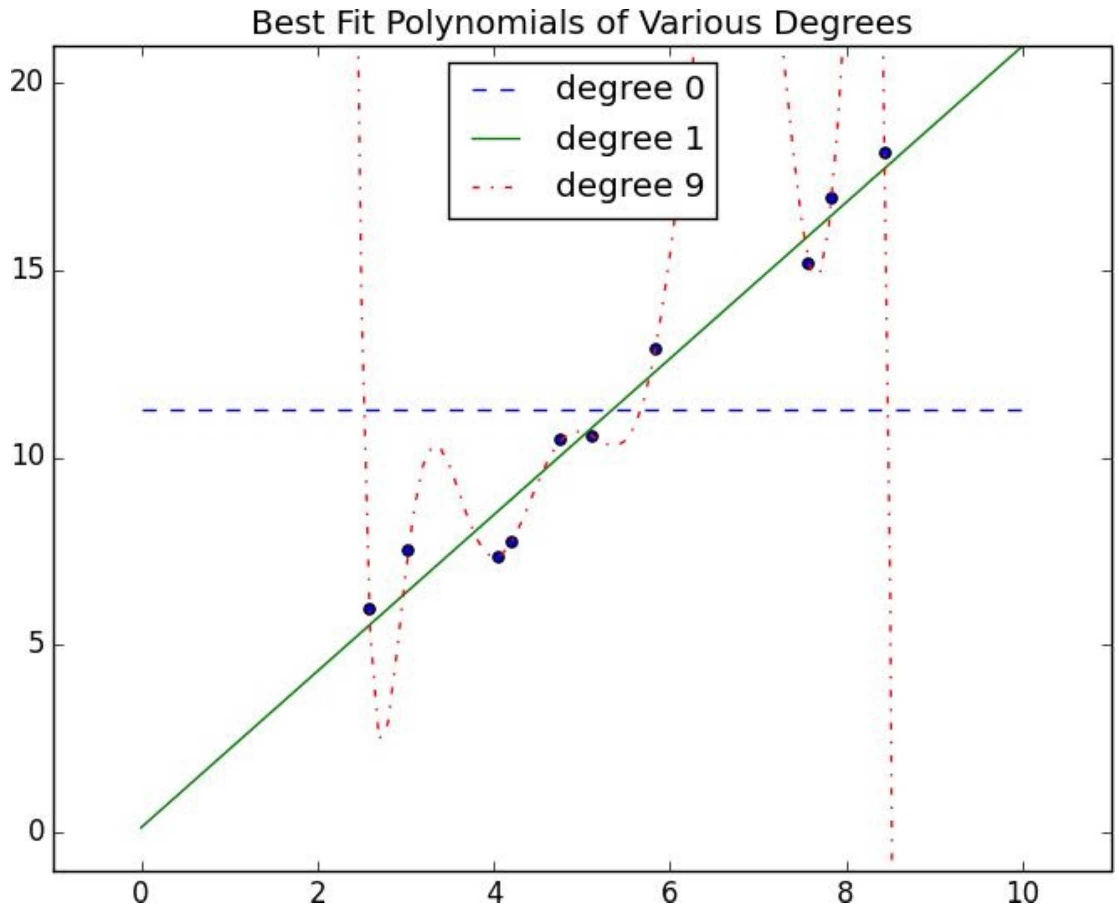


Figure 11-1. Overfitting and underfitting

Training vs. Test sets

training set

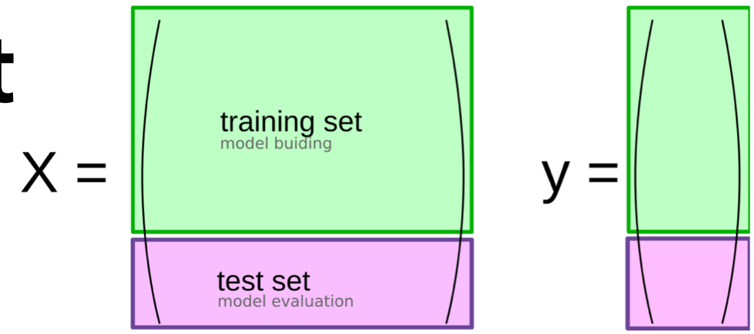
$$X = \begin{pmatrix} 1.1 & 2.2 \\ 6.7 & 0.5 \\ 2.4 & 9.3 \\ 1.5 & 0.0 \\ 0.5 & 3.5 \\ 5.1 & 9.7 \\ 3.7 & 7.8 \end{pmatrix} \qquad y = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

test set

If we have a good model, it should predict well when we have new data

- In machine learning terms, we compute our statistical model y(·) from the ***training set***
- A good estimator y(·) should then perform well on a new, independent set of data
- We “test” or assess how well y(·) performs on the new data, which we call the ***test set***

Limitations of Train/Test Model Evaluation



- Many models have hyper-parameters to select, e.g., k for KNN, or we want to choose between a KNN and DT model

- If we want to select the *best k*, learn a model for each value of k on the training set, and evaluate it on the test set

- What’s the issue?

Limitations of Train/Test Model Evaluation

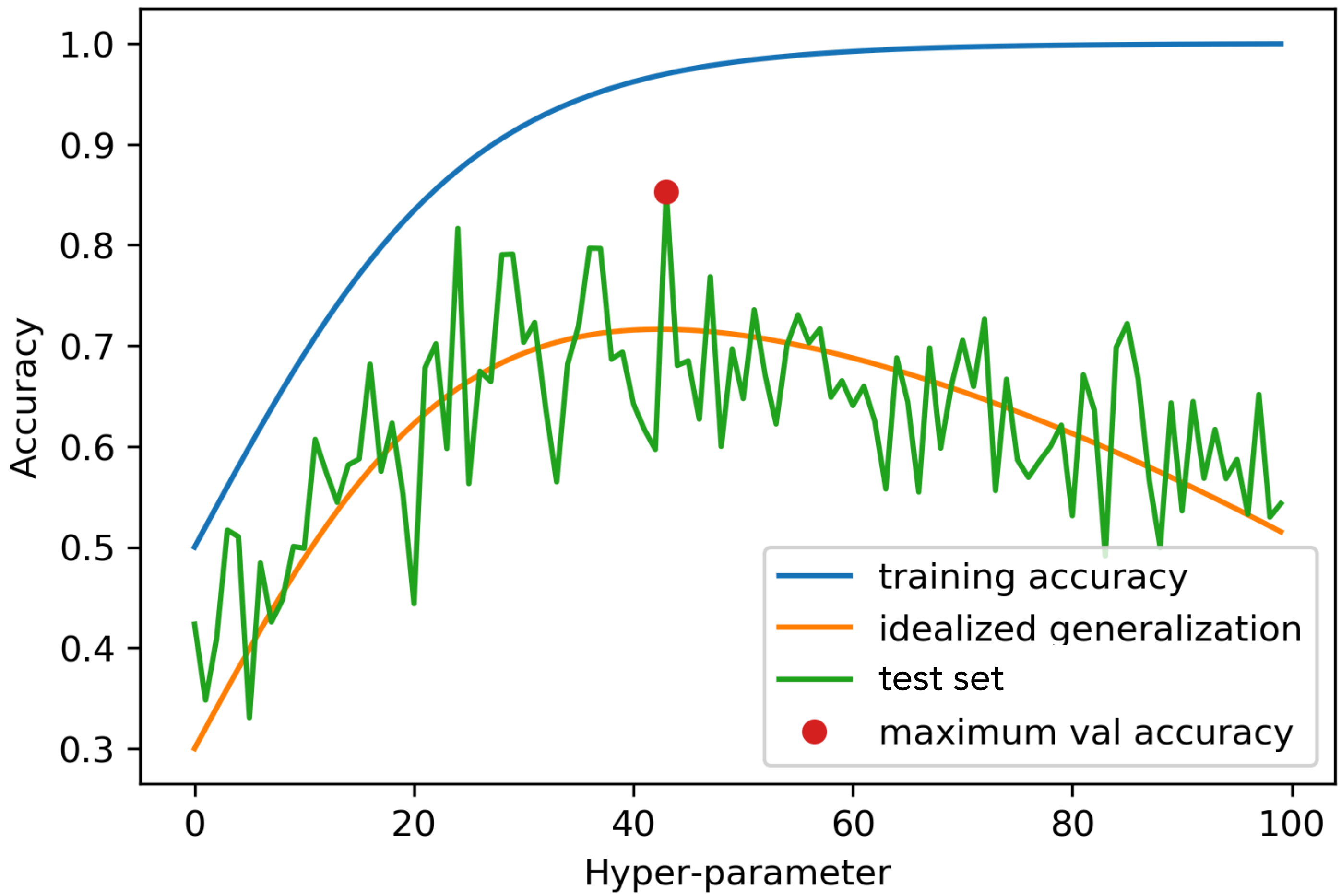


Image from Muller, AMLP

Threefold Split

- For many models we have:
  - Parameters – to specify a particular model
  - Hyperparameters – options for the model, learning process
- What should we learn where?
  - Learn parameters on training data
  - Tune hyperparameters on hold-out/validation set
  - Estimate generalization performance on test data

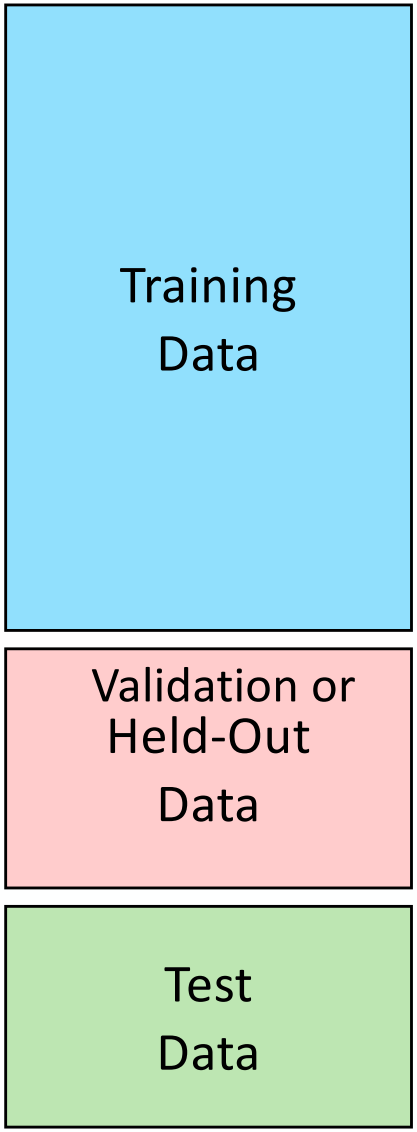
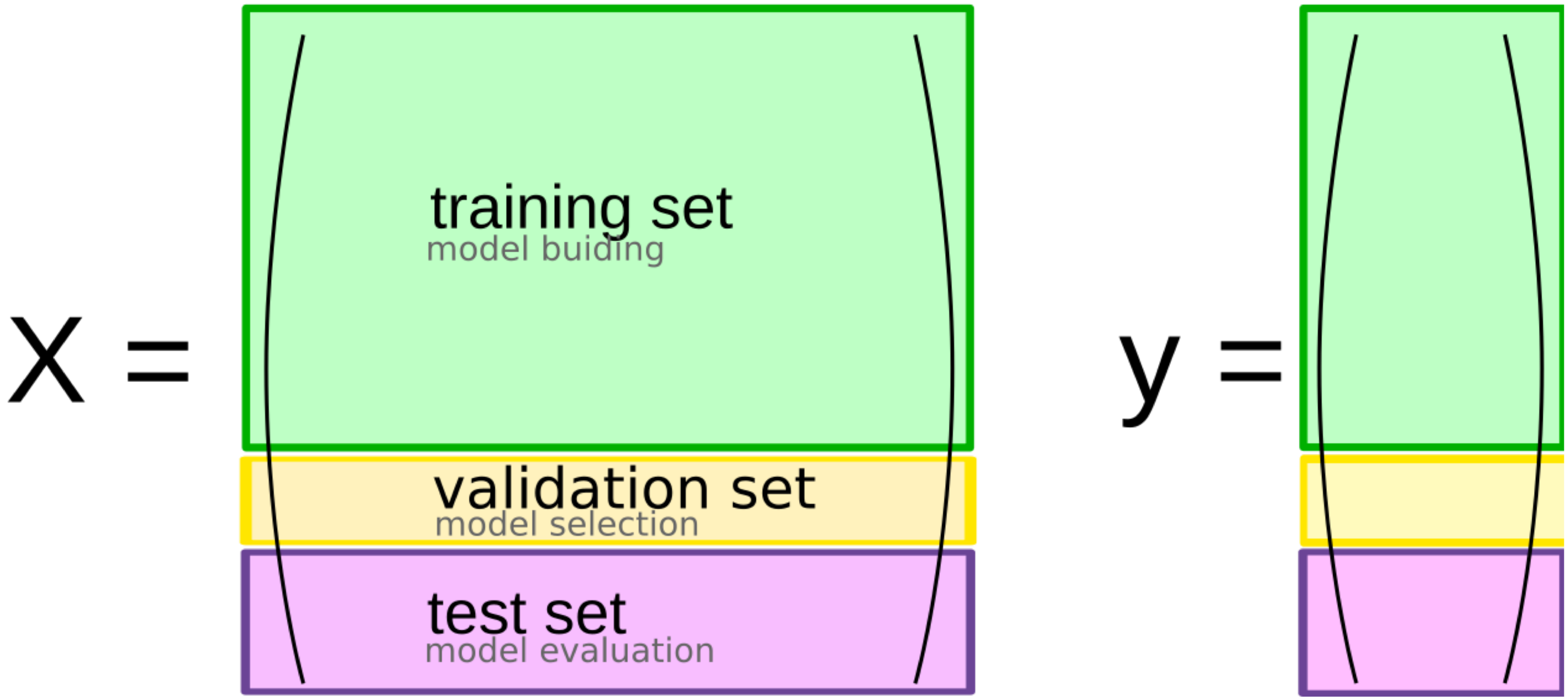


Image from CS188

Threefold Split – Best Practice



- Use Three Sets:
  - Training set – model building
  - Validation set – model selection
  - Test set – model evaluation

Threefold Split

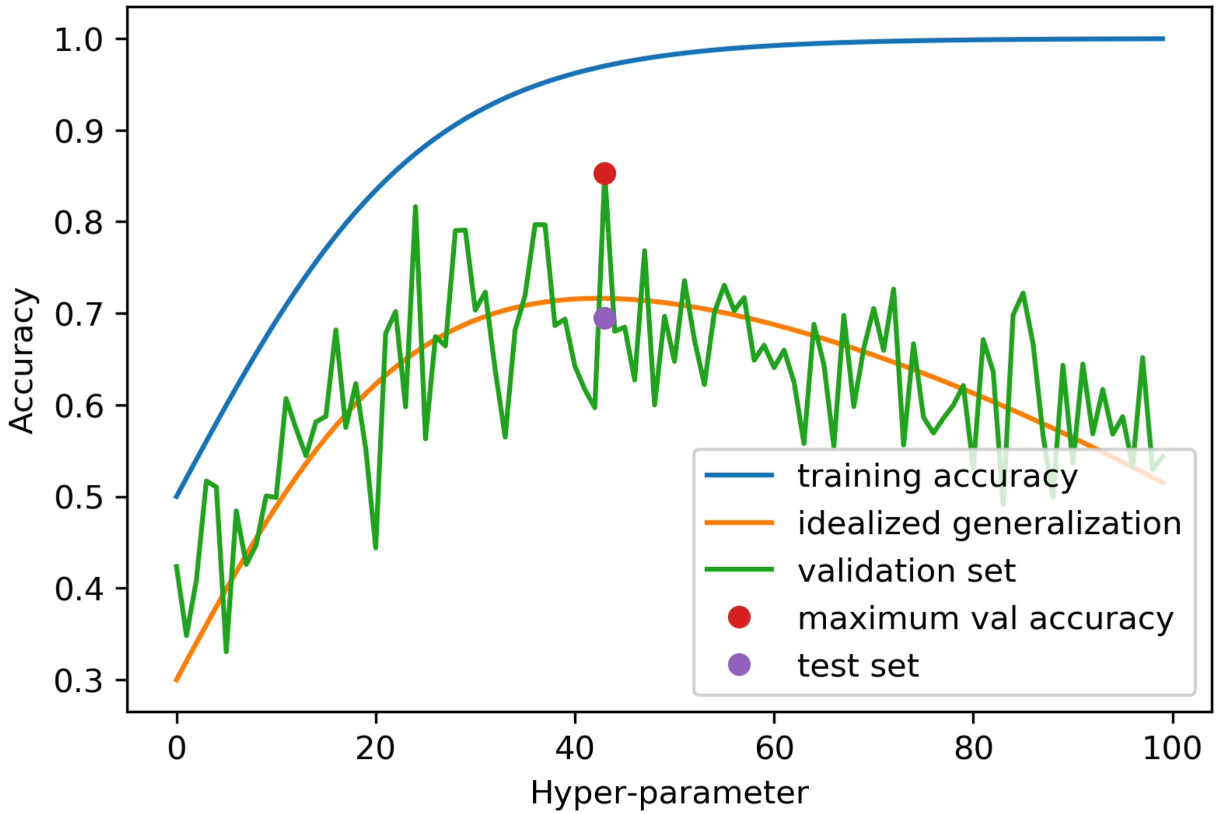


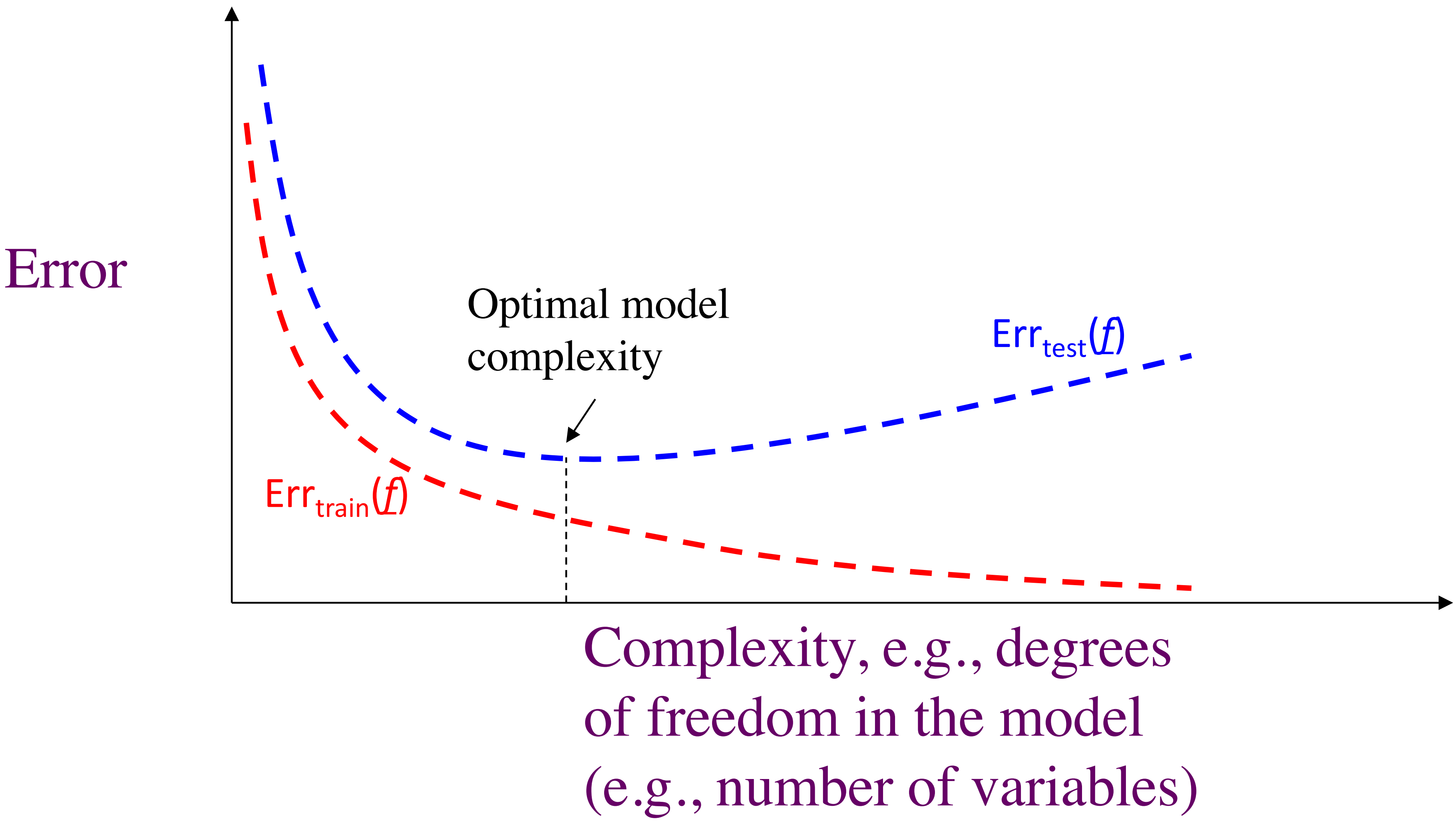
Image from Muller, AMLP

- Use validation set to select hyper-parameter; test set provides unbiased estimate of generalization performance

---

Only use the test set once!

Complexity and Generalization



Classification Errors

- Consider a problem where we want to *classify* e-mails as spam or not
- There are four types of decisions we can make:
  - True positive: Classify spam as spam
  - False positive (type-1 error): Classify not-spam as spam
  - False negative (type-2 error): Classify spam as not-spam
  - True negative: Classify not-spam as not-spam

Confusion Matrix

- M – model predicted class
- A – actual class

		Actual Outcome		
		A+	A-	
Model Outcome	M+	TP	FP	TP + FP
	M-	FN	TN	FN + TN
		TP + FN	FP + TN	*n*

- TP-true positive
- TN-true negative
- FP-false positive
- FN-false negative

Evaluation - Accuracy

- M – model predicted class
- A – actual class

		Actual Outcome		
		A+	A-	
Model Outcome	M+	TP	FP	TP + FP
	M-	FN	TN	FN + TN
		TP + FN	FP + TN	*n*

- TP-true positive
- TN-true negative
- FP-false positive
- FN-false negative

Accuracy

$$\frac{Num.\,of\;\; correct\;\, classifications}{Num.\,of\;\; total\;\, classifications} = \frac{TP + TN}{TP + FP + FN + TN}$$

Evaluation – Error Rate

- M – model predicted class
- A – actual class

		Actual Outcome		
		A+	A-	
Model Outcome	M+	TP	FP	TP + FP
	M-	FN	TN	FN + TN
		TP + FN	FP + TN	*n*

- TP-true positive
- TN-true negative
- FP-false positive
- FN-false negative

Error Rate

$$\frac{Num.\ of\ errors}{Num.\ of\ total\ classifications} = 1 - accuracy = \frac{FP + FN}{TP + FP + FN + TN}$$

Evaluation – Precision

- M – model predicted class
- A – actual class

		Actual Outcome		
		A+	A-	
Model Outcome	M+	TP	FP	TP + FP
	M-	FN	TN	FN + TN
		TP + FN	FP + TN	*n*

- TP-true positive
- TN-true negative
- FP-false positive
- FN-false negative

Precision – num. samples predicted positive really are?

$$Precision = \frac{TP}{TP + FP}$$

Evaluation – Recall

- M – model predicted class
- A – actual class

		Actual Outcome		
		A+	A-	
Model Outcome	M+	TP	FP	TP + FP
	M-	FN	TN	FN + TN
		TP + FN	FP + TN	*n*

- TP-true positive
- TN-true negative
- FP-false positive
- FN-false negative

Recall – num. samples really +, that you predicted?

$$Recall = \frac{TP}{TP + FN}$$

Evaluation – Sensitivity

- M – model predicted class
- A – actual class

		Actual Outcome		
		A+	A-	
Model Outcome	M+	TP	FP	TP + FP
	M-	FN	TN	FN + TN
		TP + FN	FP + TN	*n*

- TP-true positive
- TN-true negative
- FP-false positive
- FN-false negative

Sensitivity – true positive rate

$$Sensitivity = \frac{TP}{TP + FN}$$

Evaluation – Specificity

- M – model predicted class
- A – actual class

		Actual Outcome		
		A+	A-	
Model Outcome	M+	TP	FP	TP + FP
	M-	FN	TN	FN + TN
		TP + FN	FP + TN	*n*

- TP-true positive
- TN-true negative
- FP-false positive
- FN-false negative

Specificity – true negative rate, proportion of TN found

$$Specificity = \frac{TN}{FP + TN}$$

Evaluation – F1-score

- M – model predicted class
- A – actual class

		Actual Outcome		
		A+	A-	
Model Outcome	M+	TP	FP	TP + FP
	M-	FN	TN	FN + TN
		TP + FN	FP + TN	*n*

- TP-true positive
- TN-true negative
- FP-false positive
- FN-false negative

F_1-measure – harmonic mean of precision and recall

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}$$