INTRODUCTION TO MACHINE LEARNING

ASTR 324

STEPHEN PORTILLO

SUPPLEMENTAL READING

Ch 1, 5.1-5.3

Goodfellow, Bengio, and Courville (2016)

https://www.deeplearningbook.org/

Ch 8.4, 17.1, 18

Efron and Hastie (2016)

https://web.stanford.edu/~hastie/CASI_files/PDF/casi.pdf

WHAT IS MACHINE LEARNING?

"A computer program is said to learn[…] if its performance at tasks[…] as measured by [some performance measure], improves with experience."

Mitchell (1997)

RULE-BASED SYSTEMS

Does this tweet contain the string “dog”?

This task can be easily codified into a rule:

```
return "dog" in tweet
```

We often use computers to automate tasks that can be codified, like PSF photometry

LIMITS OF RULE-BASED SYSTEMS

Does this image contain a dog?

This task is easy to perform for humans, but how can we get a computer to do it?

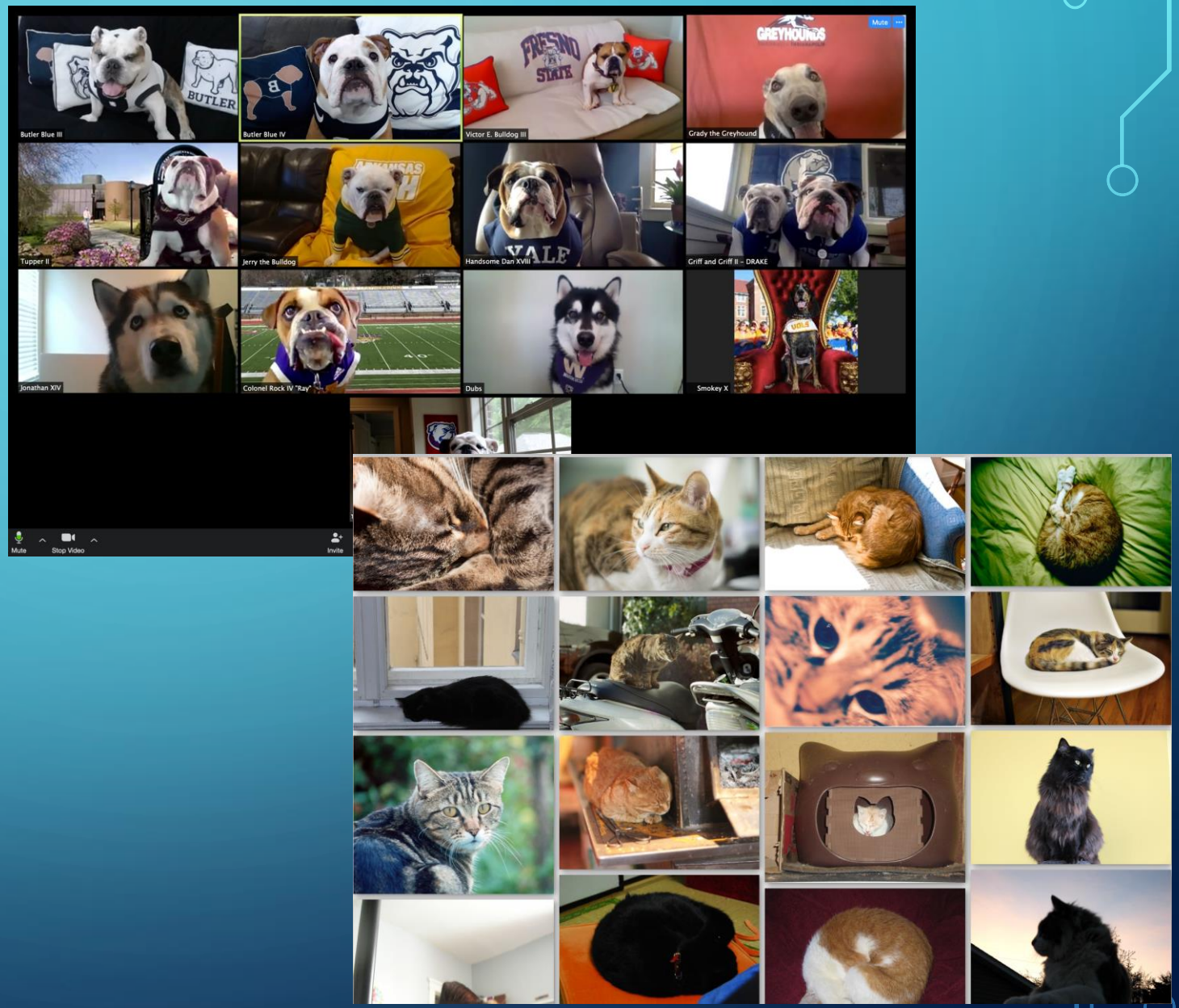
```
return dog in image #?!
```

Can we use computers to automate these messier tasks, like classifying spiral/elliptical galaxies?

MACHINE LEARNING

Instead, we can collect a **dataset** of images that do or do not contain dogs

Using machine learning, we can **train** a **model** that learns how to identify dogs



@UWDubs, Open Images Dataset V6

LIMITS OF MACHINE LEARNING

A machine learning model is only as good as the dataset it learns from

Goodfellow+ recommends 10M labelled examples with 5K per category

Togootogtokh & Amartuvshin (2018)

AGE OF "BIG DATA"

Astronomy already has large, complex datasets

Rubin Observatory/LSST will have trillions of observations of tens of billions of objects

Massive amounts of complex data are available, from genomic data to credit card transactions

We can use machine learning to use this data more effectively

LSST Project/NSF/AURA

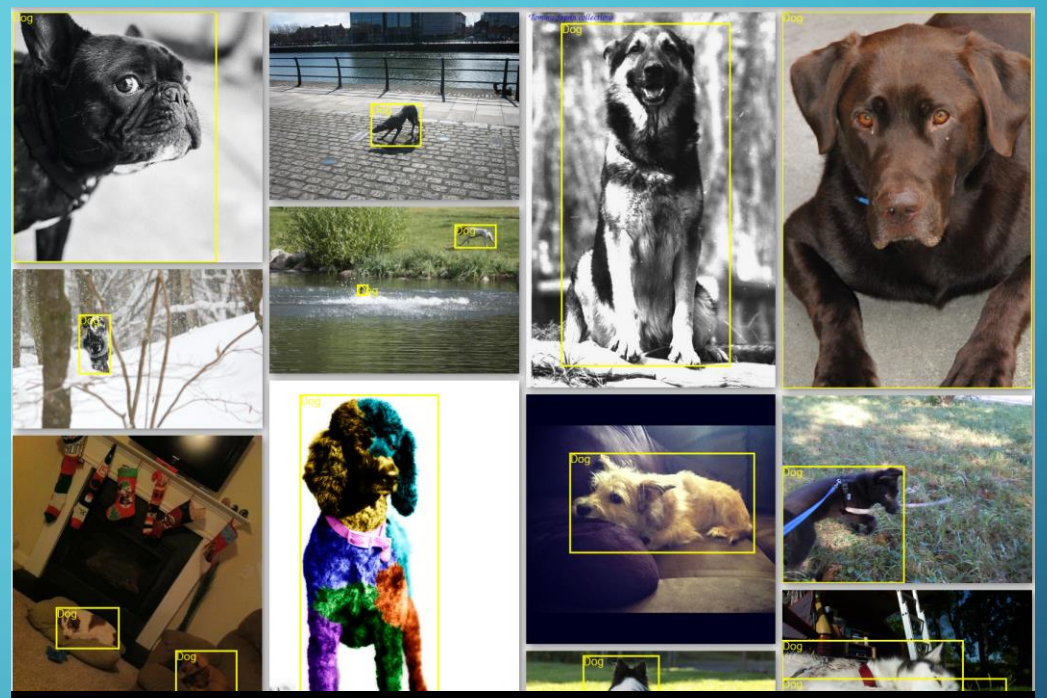
SUPERVISED LEARNING

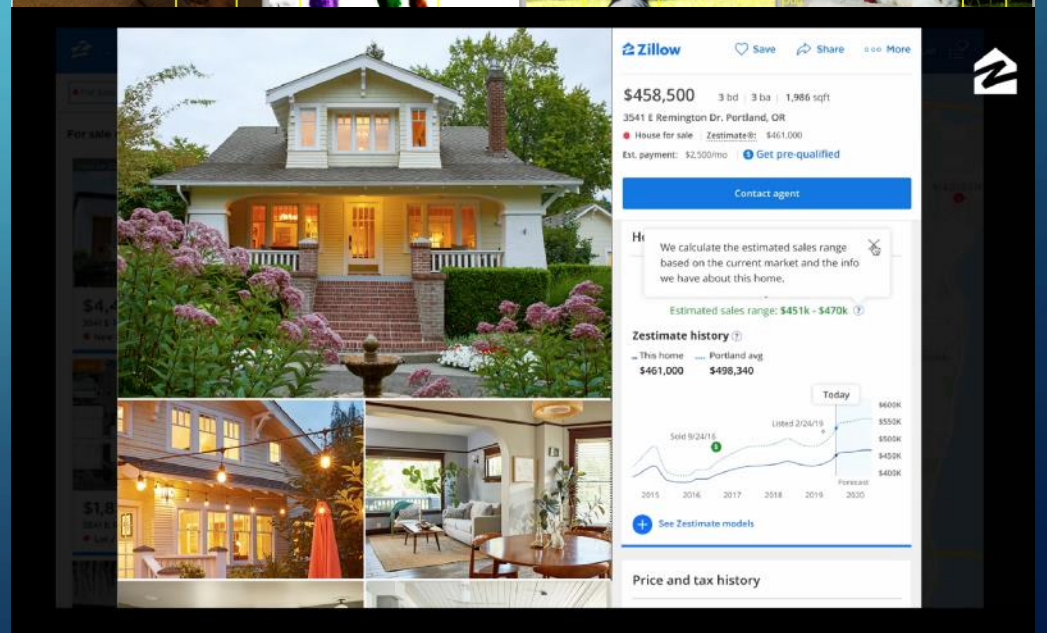
Dataset has **features** and a **label/target**

We want the model to use the features to predict the target

Classification: target is a category

Regression: target is a number





Open Images Dataset V6, Zillow

UNSUPERVISED LEARNING

Dataset has features and the model learns something useful about the structure of the dataset

Generative Adversarial Networks (GANs) trained on a dataset of faces can generate new faces

Supervised/unsupervised is a spectrum

GPT-2 is trained to predict only the next word in text, but learns to mimic writing

SYSTEM PROMPT (HUMAN-WRITTEN)	*Legolas and Gimli advanced on the orcs, raising their weapons with a harrowing war cry.*
MODEL COMPLETION (MACHINE-WRITTEN, FIRST TRY)	The orcs' response was a deafening onslaught of claws, claws, and claws; even Elrond was forced to retreat. "You are in good hands, dwarf," said Gimli, who had been among the first to charge at the orcs; it took only two words before their opponents were reduced to a blood-soaked quagmire, and the dwarf took his first kill of the night. The battle lasted for hours until two of the largest Orcs attempted to overwhelm Aragorn. When they finally stopped, they lay defeated and lifeless for miles and miles.

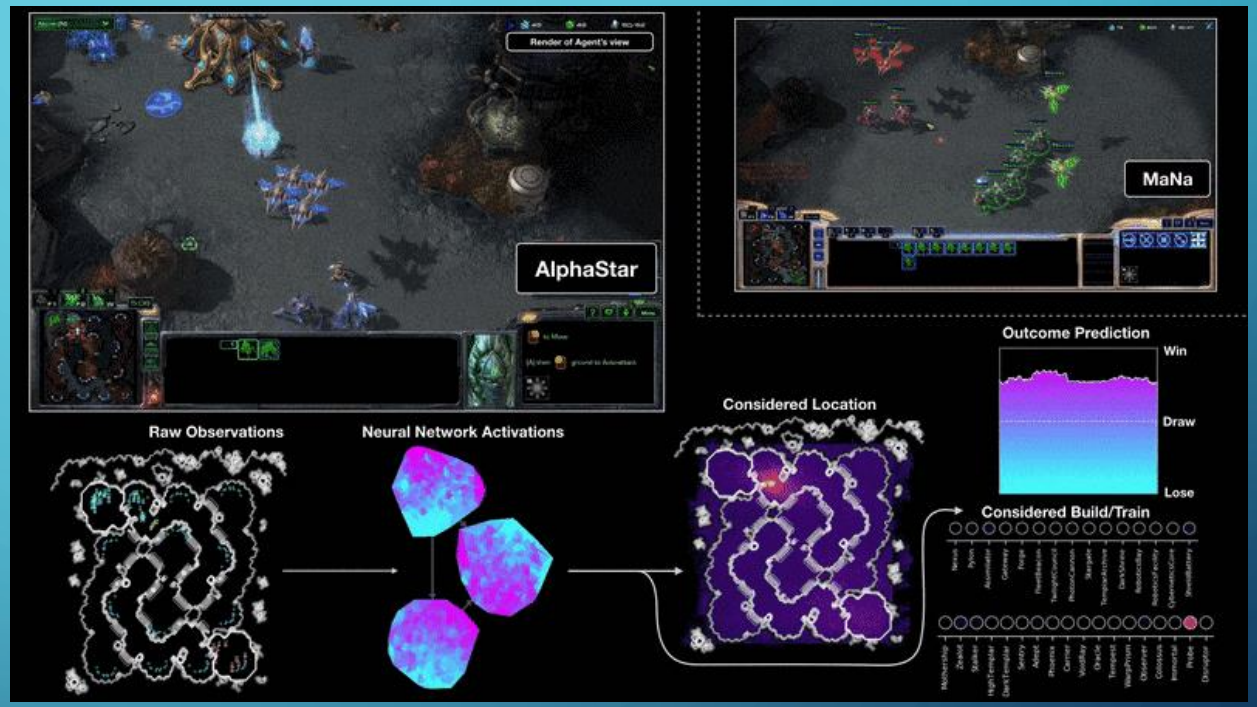
https://www.thispersondoesnotexist.com/
https://talktotransformer.com/

Karras et al. (2018), Radford (2019)

REINFORCEMENT LEARNING

The dataset is not fixed – instead the model interacts with an environment

AlphaStar achieved Grandmaster level in Starcraft II by learning from human games and self-play



Vinyals et al. (2019)
https://youtu.be/nbiVbd_CEIA

BUILDING A MACHINE LEARNING MODEL

Dataset

Loss Function

Model

Optimization Procedure

LINEAR REGRESSION

Dataset: measurements of the redshift z and distance modulus μ of Type Ia supernovae

z	μ
0.4686	41.97
0.7455	43.10
0.0294	35.69
0.3832	41.10
0.2622	40.95
0.2116	39.92

Loss Function: we want a prediction $\hat{\mu}(z)$ – let's use mean squared error

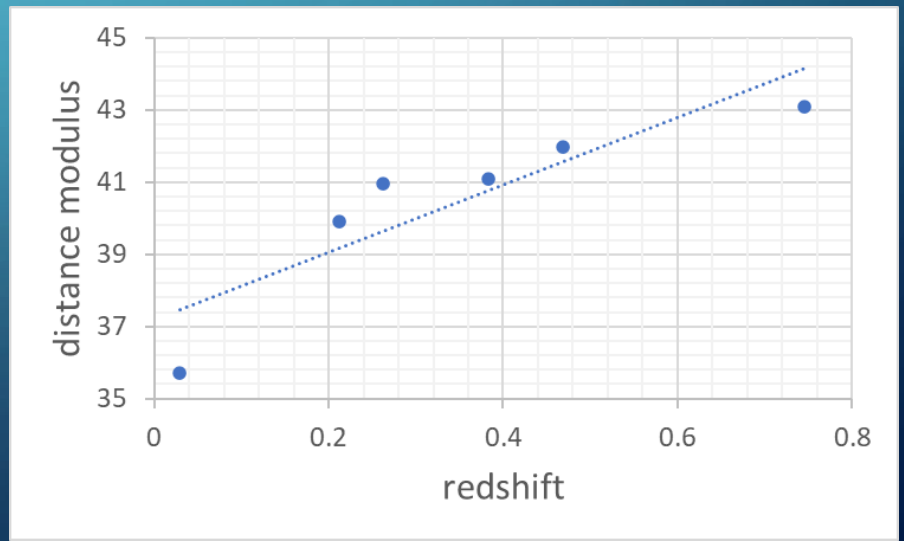
$$L = \frac{1}{N}\sum\left(\mu - \hat{\mu}(z)\right)^2$$

Model: Let's start simple – linear regression

$$\hat{\mu}(z) = a + b\,z$$

Optimization procedure:

$$\frac{\partial L}{\partial a} = 0 \quad \frac{\partial L}{\partial b} = 0$$



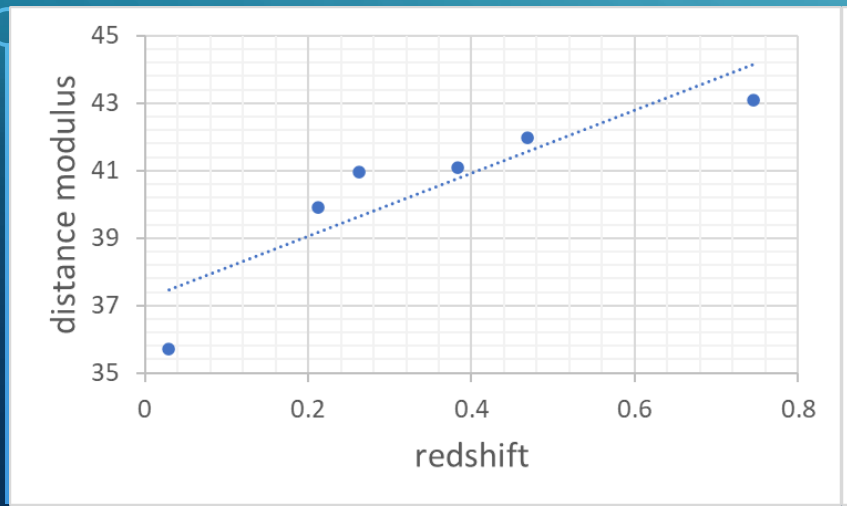
CAPACITY

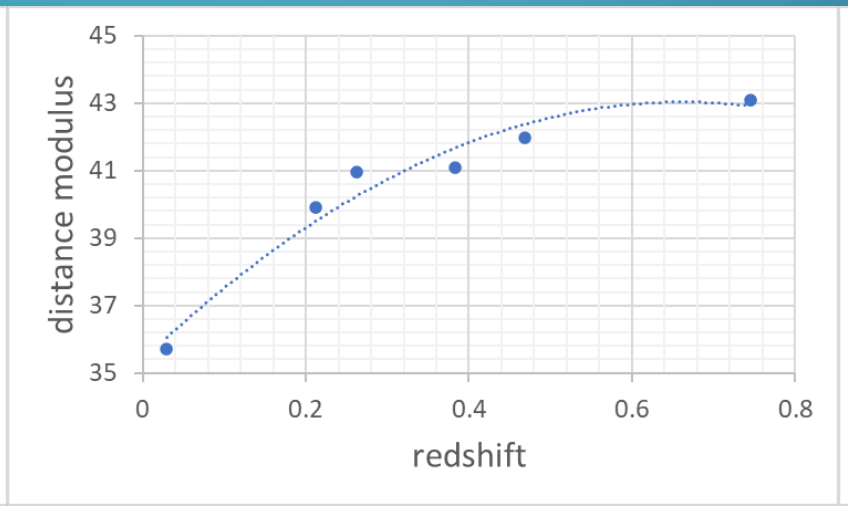
We can get a better fit to the dataset by making the model more flexible, increasing its **capacity**

We can increase capacity by using higher order polynomials

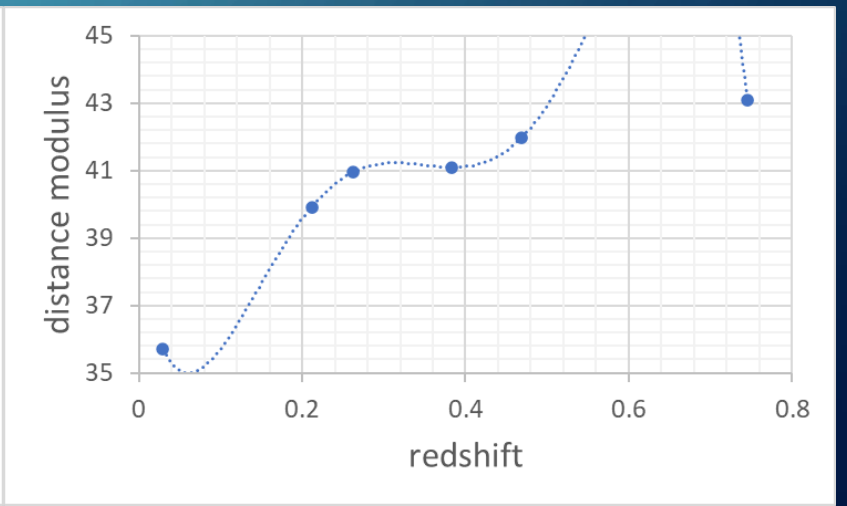
In fact, using a 5^{th} order polynomial as the model gives us zero loss

low capacity





high capacity



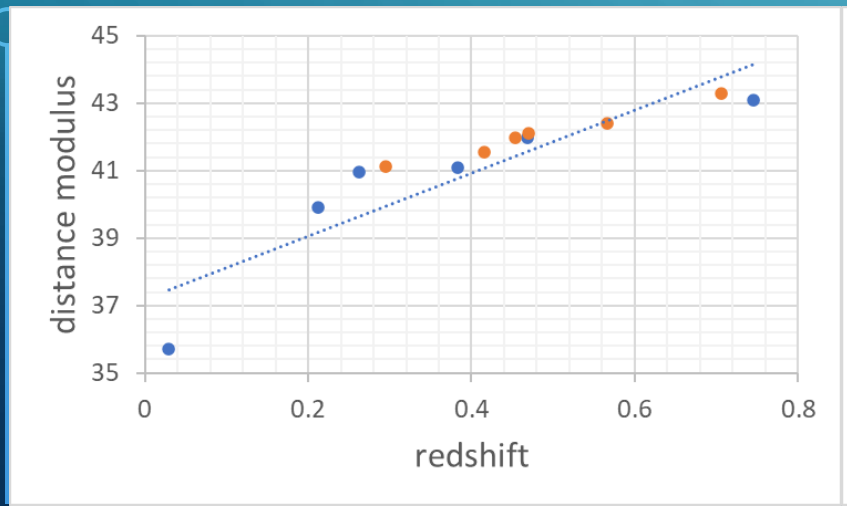
GENERALIZATION ERROR

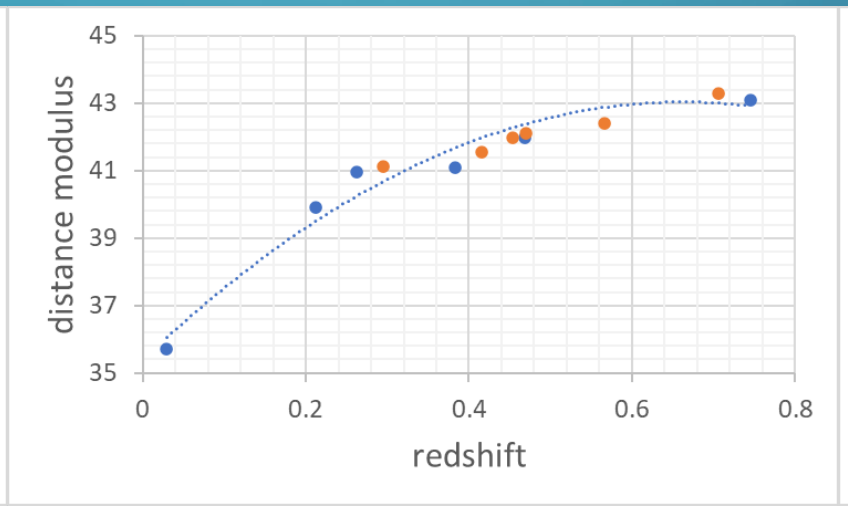
But we want our model to perform well on data it was not trained on

Use a separate **validation set** to see how well the models generalize

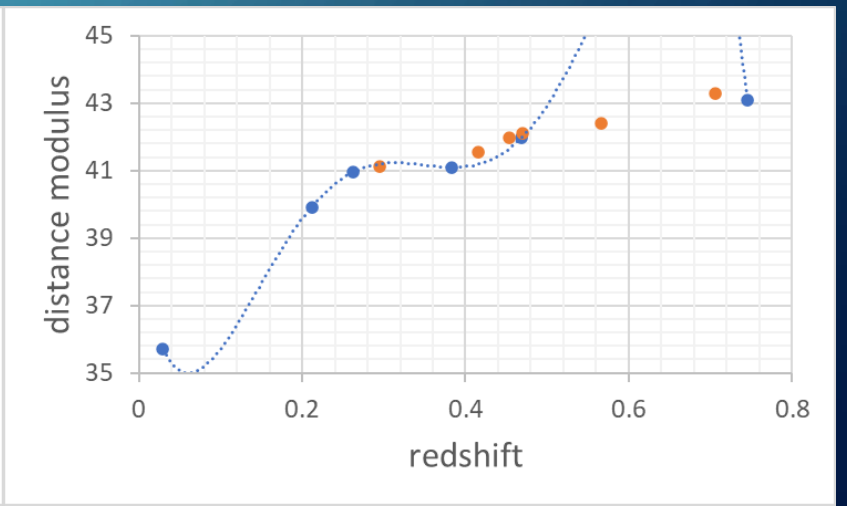
Underfitting occurs when capacity is too low, **overfitting** when it is too high

underfitting





overfitting

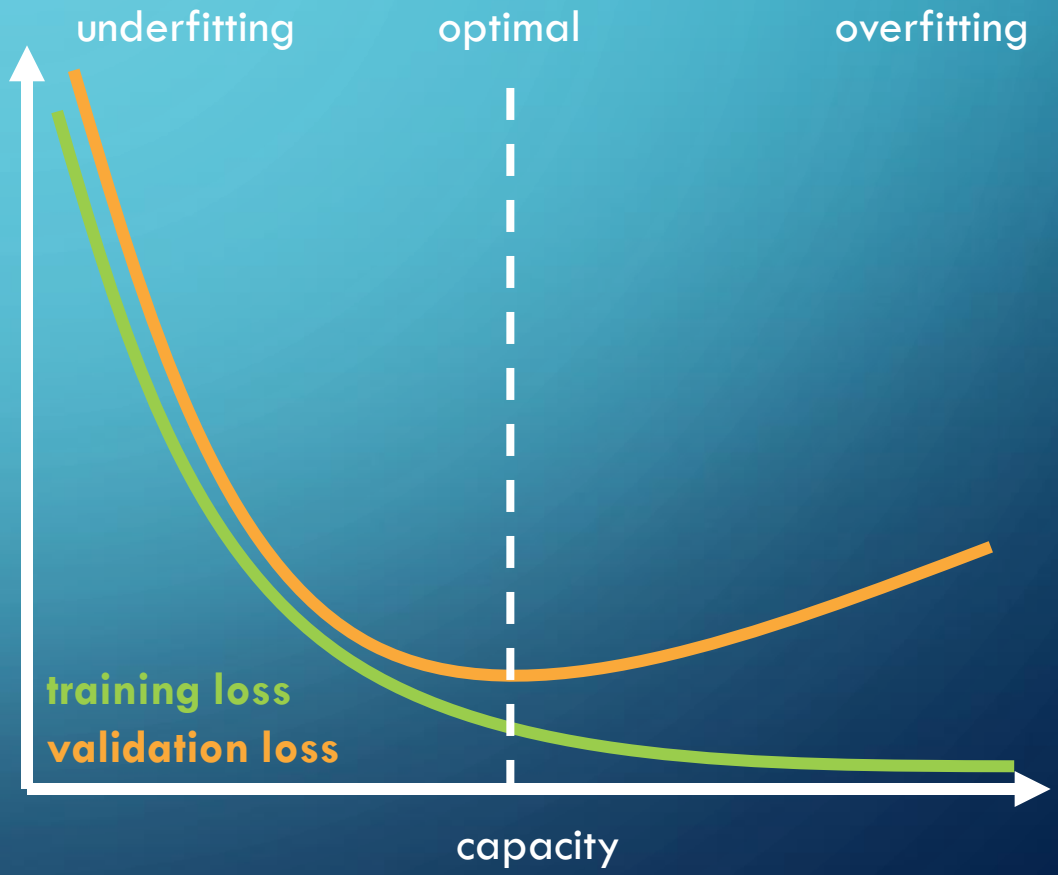


VALIDATION AND TEST SETS

Model choices (like polynomial order) are often called **hyperparameters**

Choose hyperparameters that minimize validation loss

When comparing to others' work, compare the loss on a separate **test set**



ISN'T THIS JUST STATISTICS?

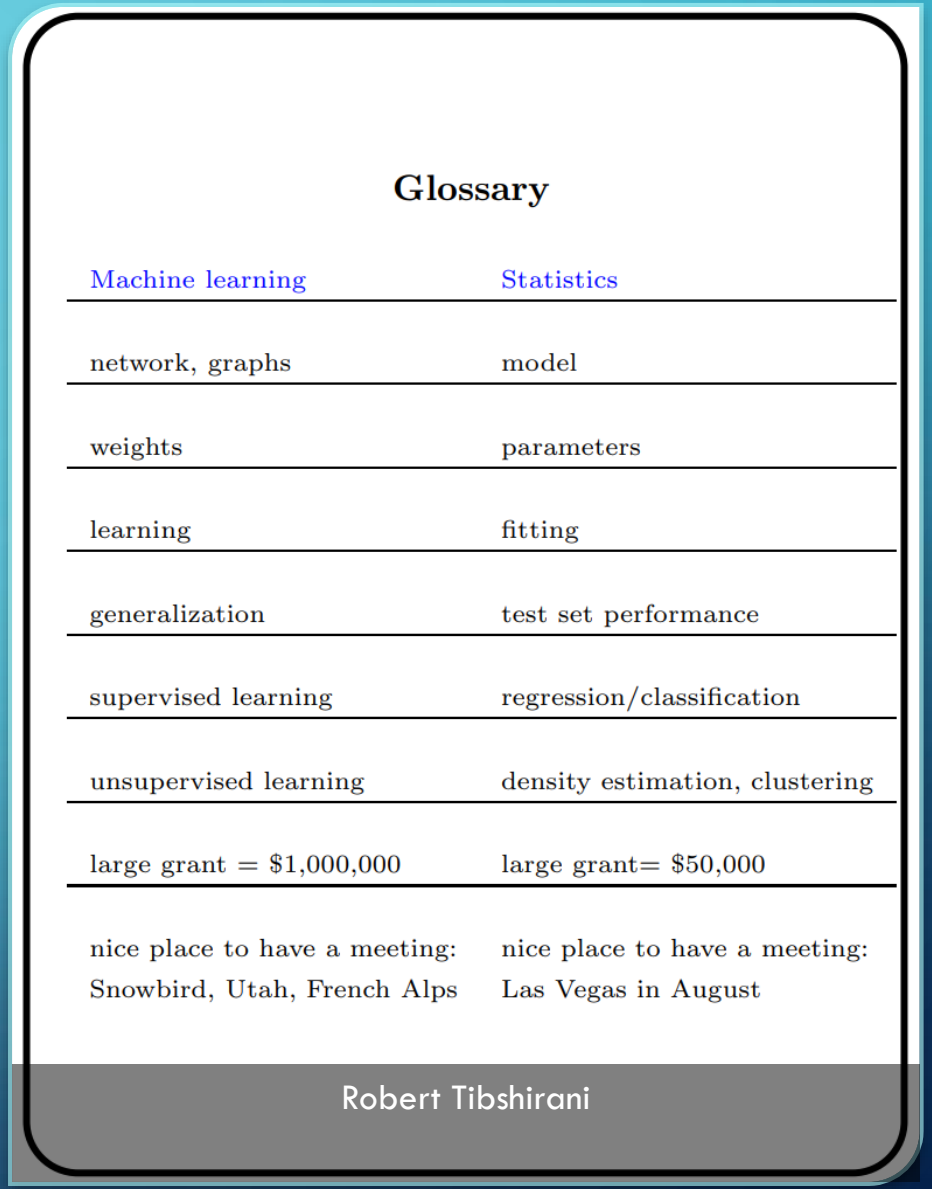
Larry Wasserman: "They are both concerned with the same question: how do we learn from data?"

My take:

Statistics – you have a model you might actually believe

$$H^2 = H_0^2(\Omega_m(1+z)^3 + \Omega_\Lambda)$$

Machine Learning – optimize a really flexible model using a lot of data



Glossary

Machine learning	Statistics
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering
large grant = \$1,000,000	large grant= \$50,000
nice place to have a meeting: Snowbird, Utah, French Alps	nice place to have a meeting: Las Vegas in August

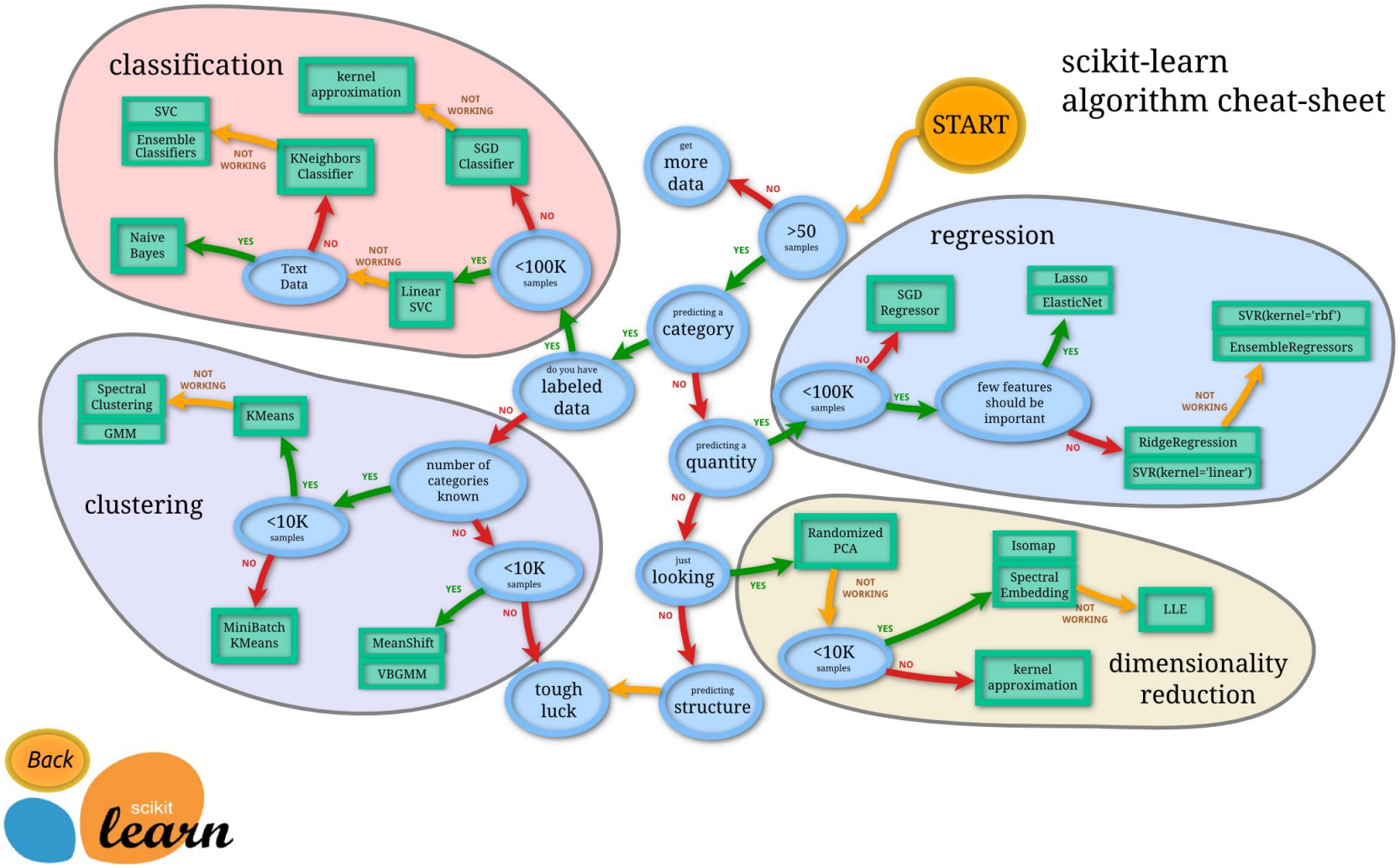
Robert Tibshirani

CLASSIC MACHINE LEARNING

Classic machine learning depends on **feature engineering**

Many different machine learning models

We'll focus on decision trees



RR LYRAE CLASSIFICATION*

dataset – features: colors, target: RR Lyrae or not

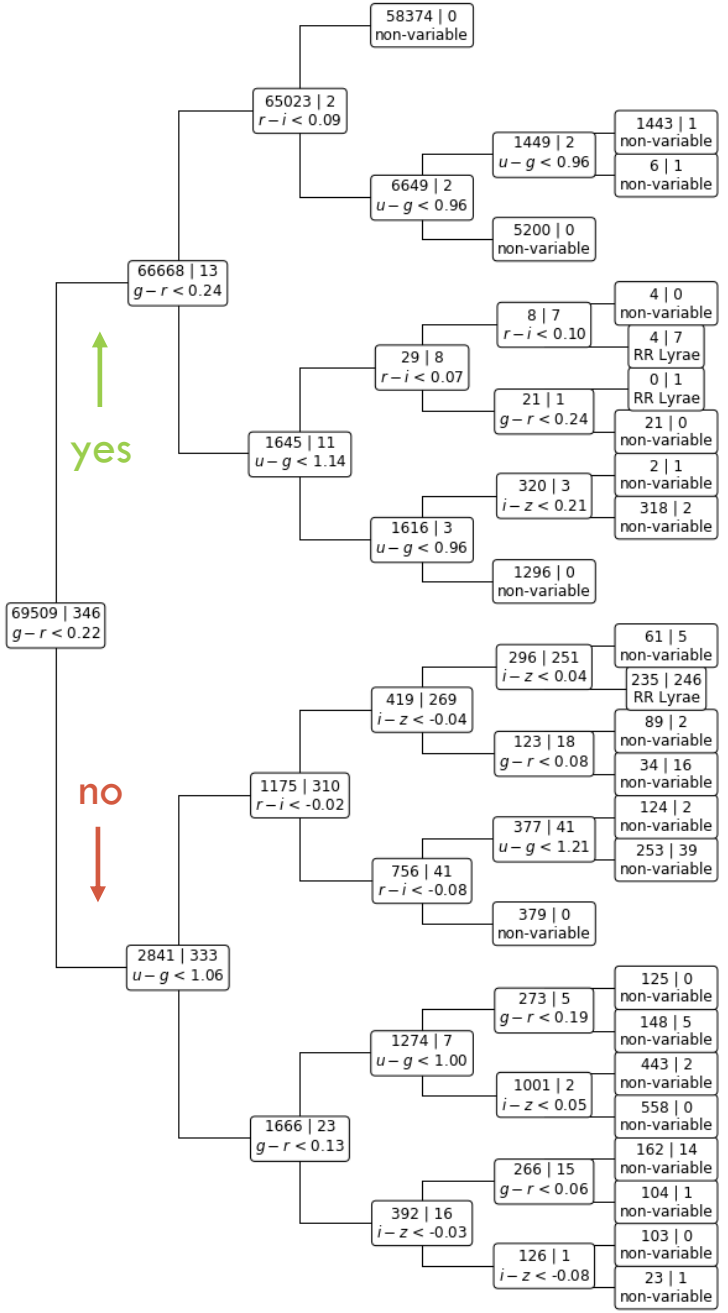
in training set, we have both, but in general, colours are easier to get than an RR Lyrae classification

model – decision tree

loss function – Gini impurity $\frac{N_- \times N_+}{(N_- + N_+)^2}$

optimization – choose the split that minimizes Gini impurity in the next level of the tree

capacity set by tree depth

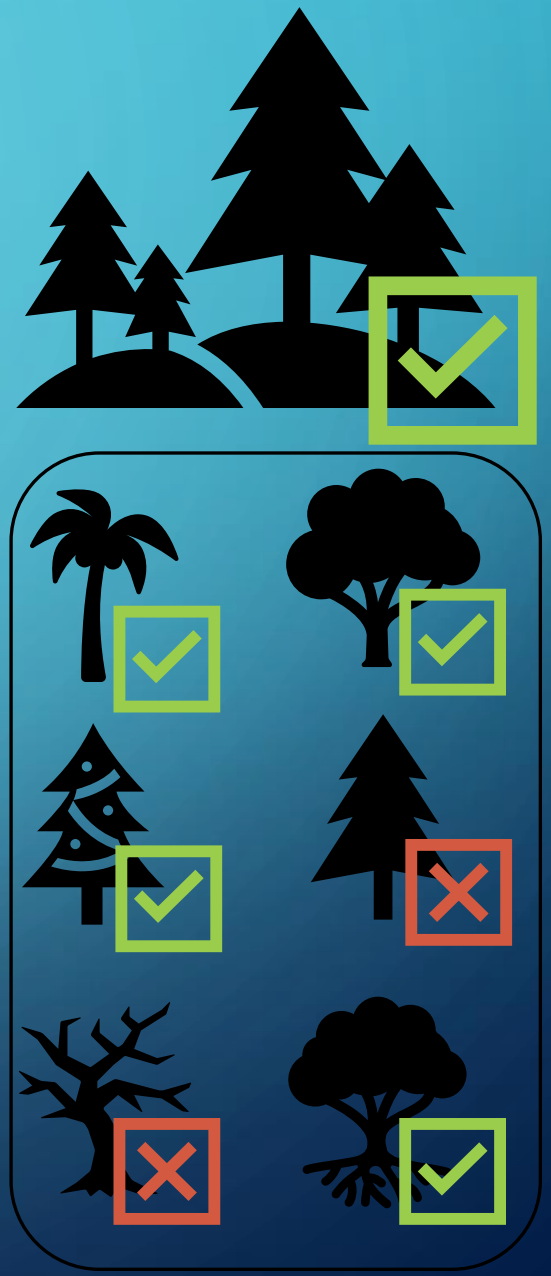


RANDOM FORESTS

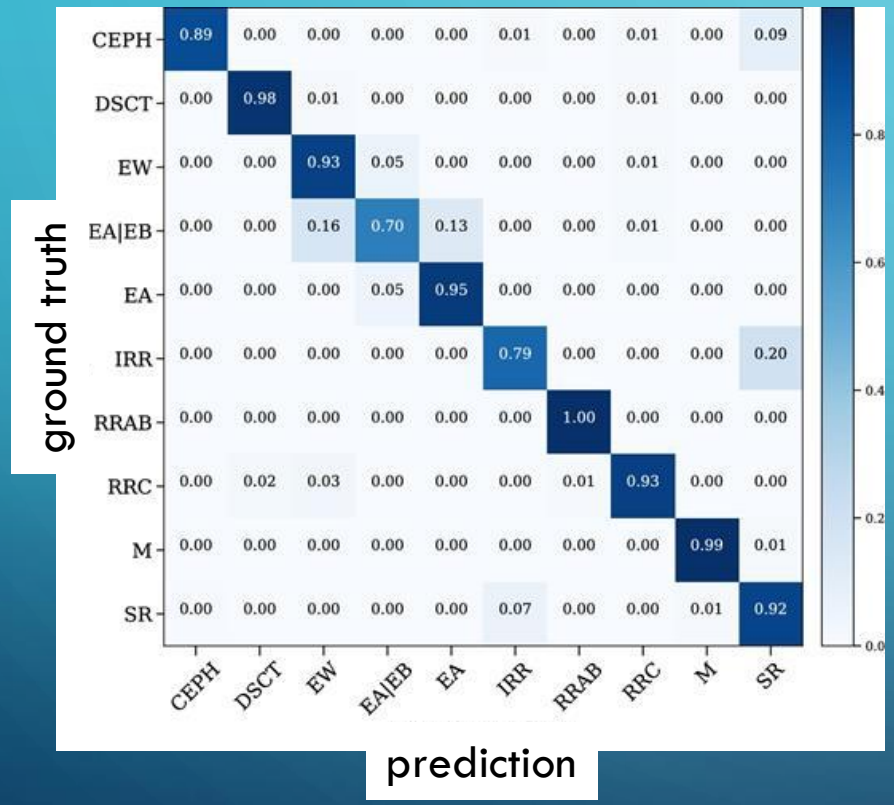
bagging (bootstrap aggregating) – make an ensemble of trees by giving each a different subset of the dataset

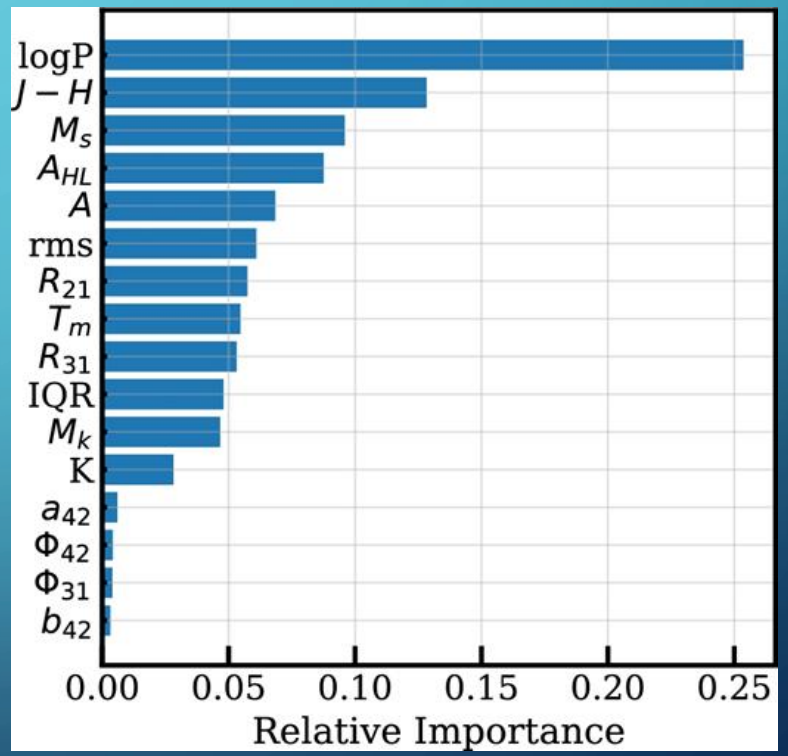
feature bagging – when creating a new split, select a random subset of the features to consider

classify examples by having the trees vote – tends to reduce overfitting



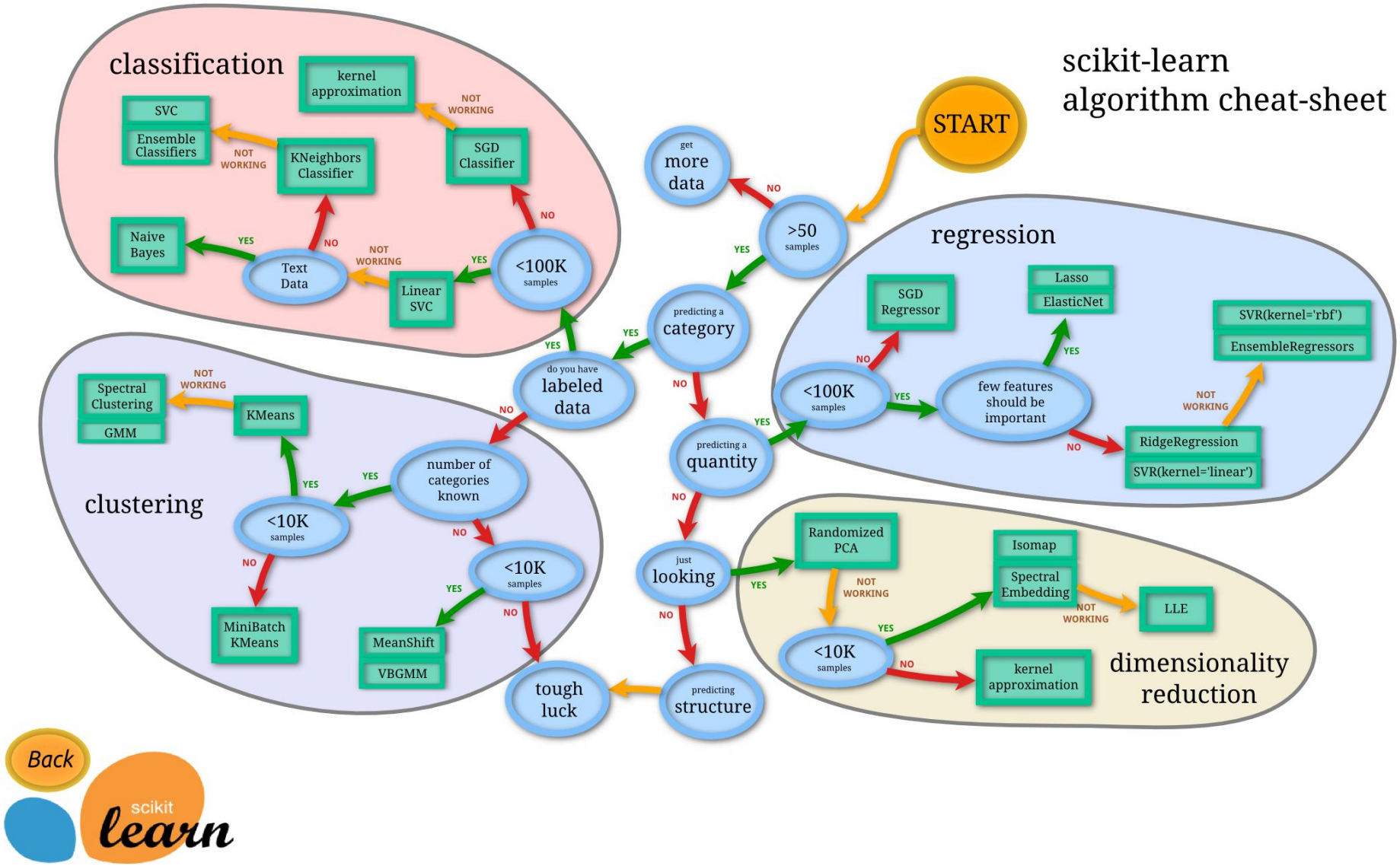
VARIABILITY CLASSIFICATION IN ASAS-SN





Jayasinghe et al. (2018)

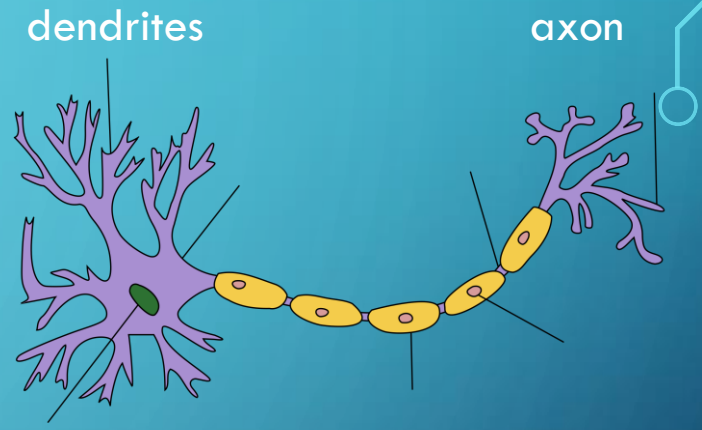
scikit-learn algorithm cheat-sheet

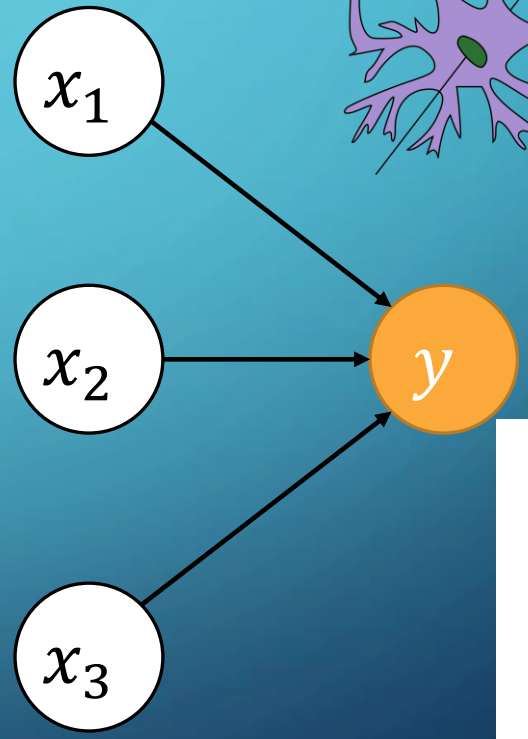


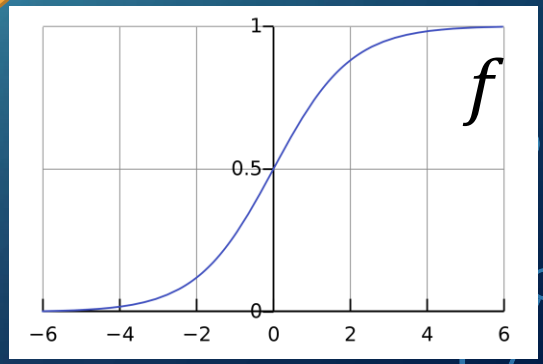
ARTIFICIAL NEURAL NETWORKS

Neurons take in multiple inputs and give an output **activation**

1. Multiply each input by a weight
2. Add the weighted inputs with a bias term
3. Apply a non-linear activation function to the sum







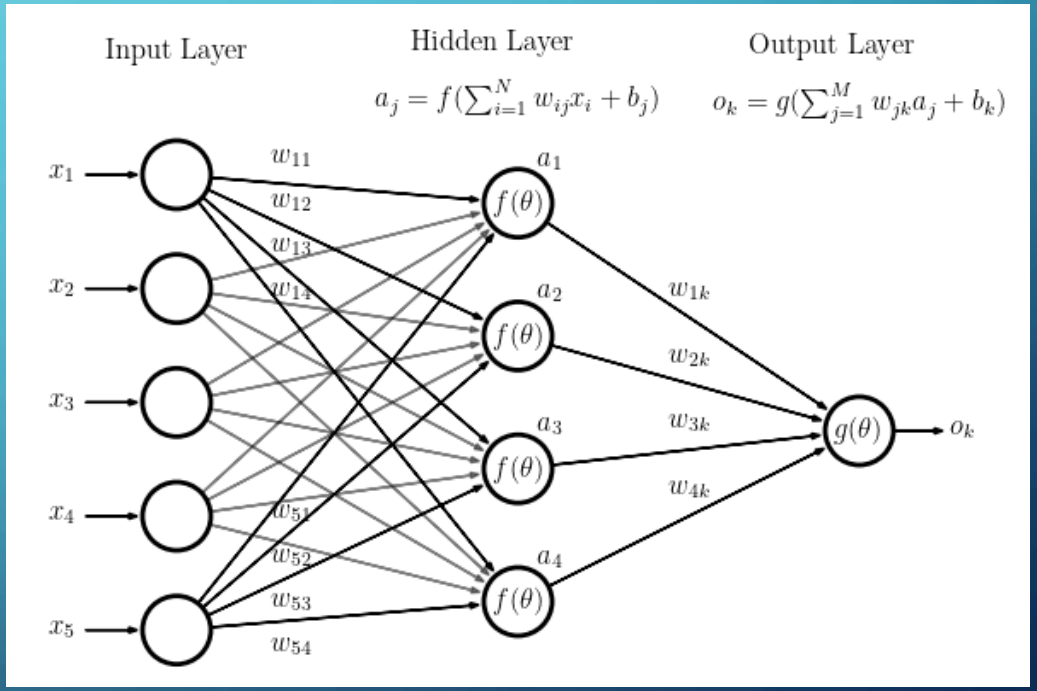
$$y = f(x_1 w_1 + x_2 w_2 + x_3 w_3 + b)$$

ARTIFICIAL NEURAL NETWORKS

Neurons in a **layer** take in the same inputs but can apply different weights and biases

If it is wide enough, a network with one hidden layer can approximate any function of the inputs

The ultimate in flexible models?

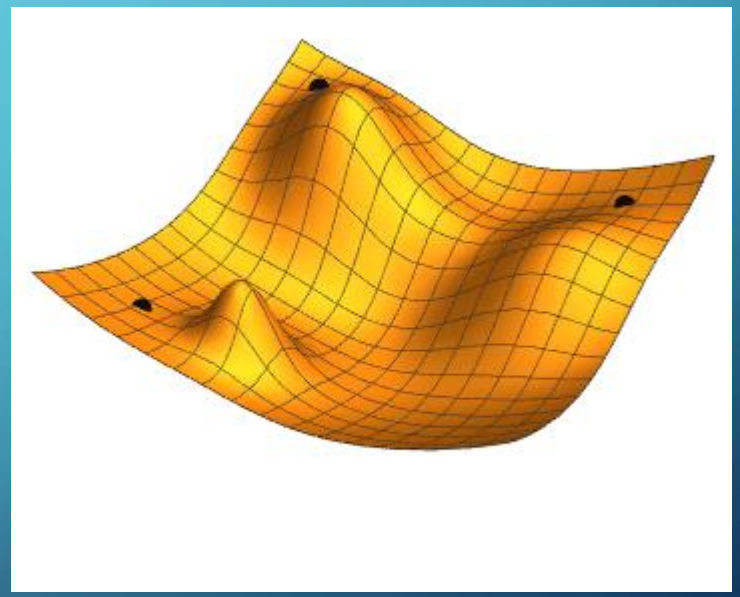


GRADIENT DESCENT

Need a way to optimize the weights and biases: **stochastic gradient descent (SGD)**

Gradient descent takes steps in the direction of the gradient of the loss function

SGD adds some stochasticity, avoiding local minima



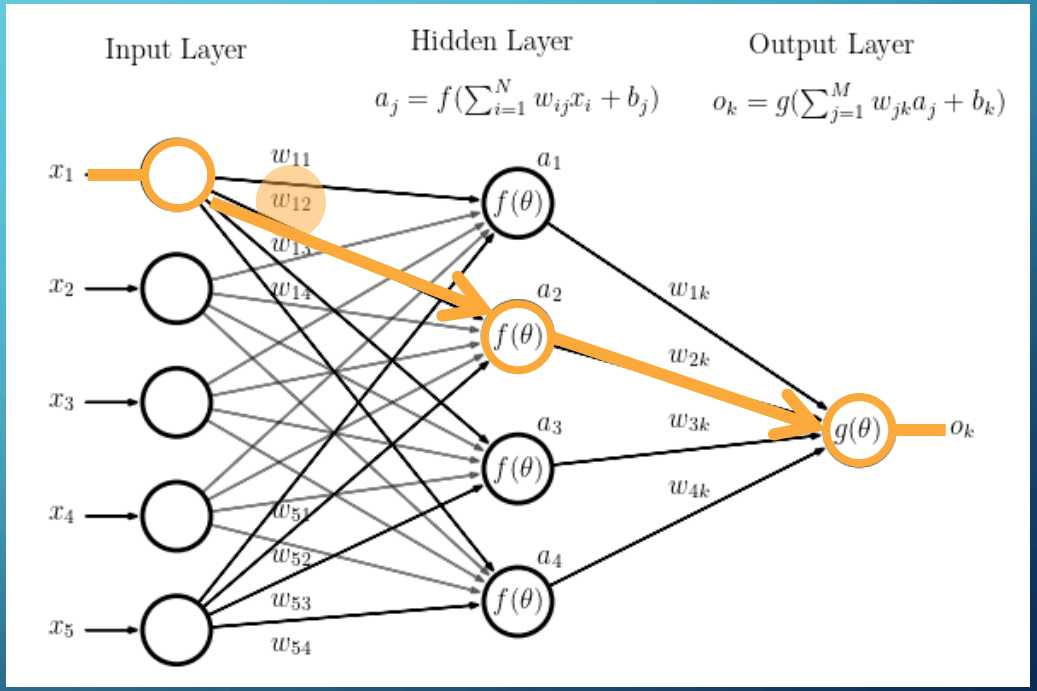
BACKPROPAGATION

Calculating gradients can be done by traversing network backwards: **backpropagation**

$$\frac{\partial L}{\partial w_{12}} = \frac{\partial L}{\partial o_k} \cdot \frac{\partial o_k}{\partial a_2} \cdot \frac{\partial a_2}{\partial w_{12}} = \frac{\partial L}{\partial o_k} \cdot g' \cdot w_{2k} \cdot f' \cdot x_1$$

The ReLU activation function makes derivatives really easy

$$\mathrm{ReLU}(x) = \max(0, x)$$



PHOTOMETRIC REDSHIFTS*

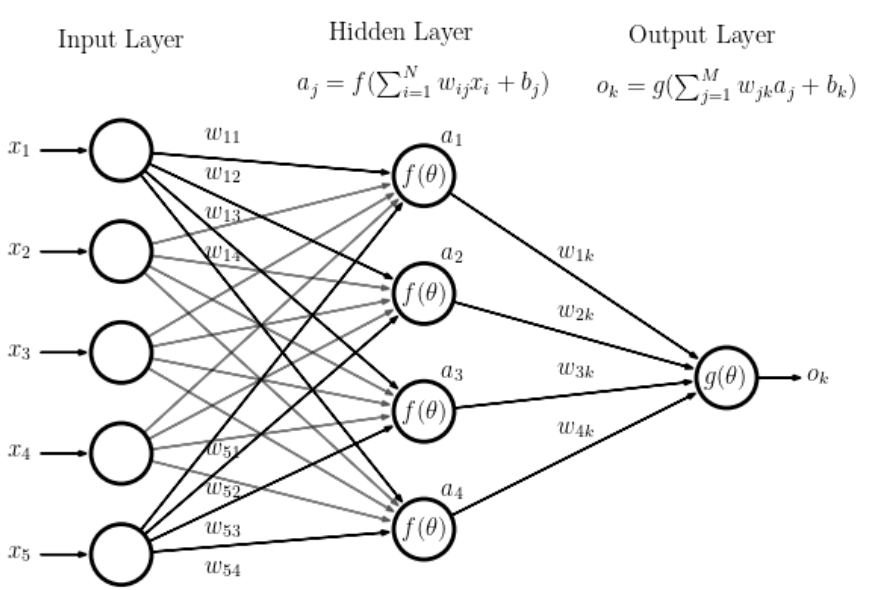
features: *ugriz* photometry

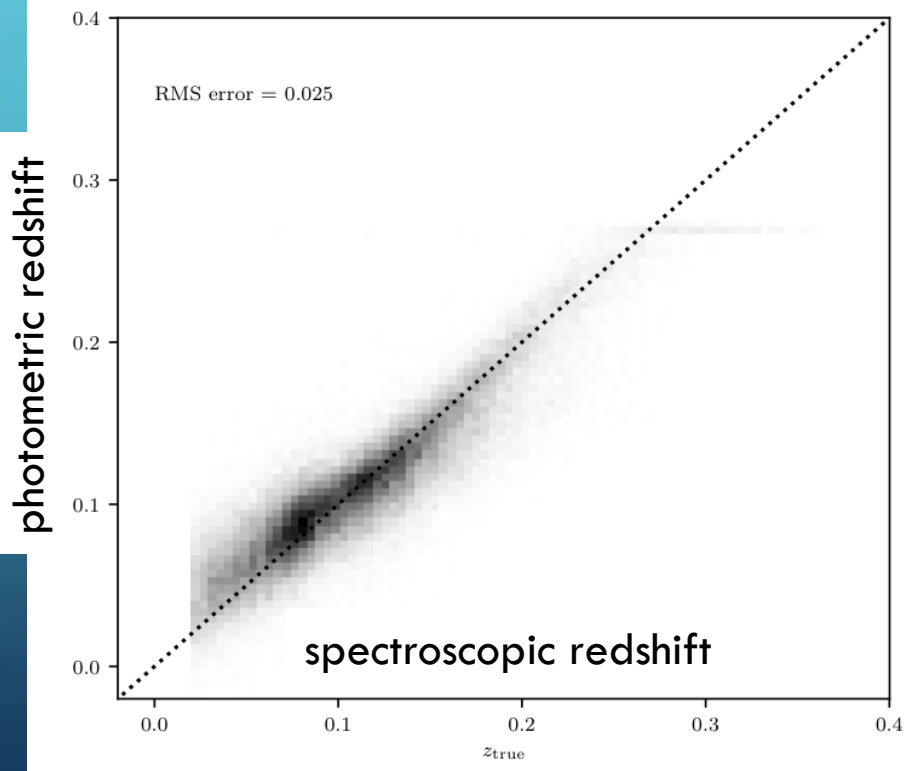
target: spectroscopic redshift

loss function: mean squared error

model: neural network

optimization method: SGD





NEURAL NETWORK HYPERPARAMETERS

Number of layers and their widths

Activation functions

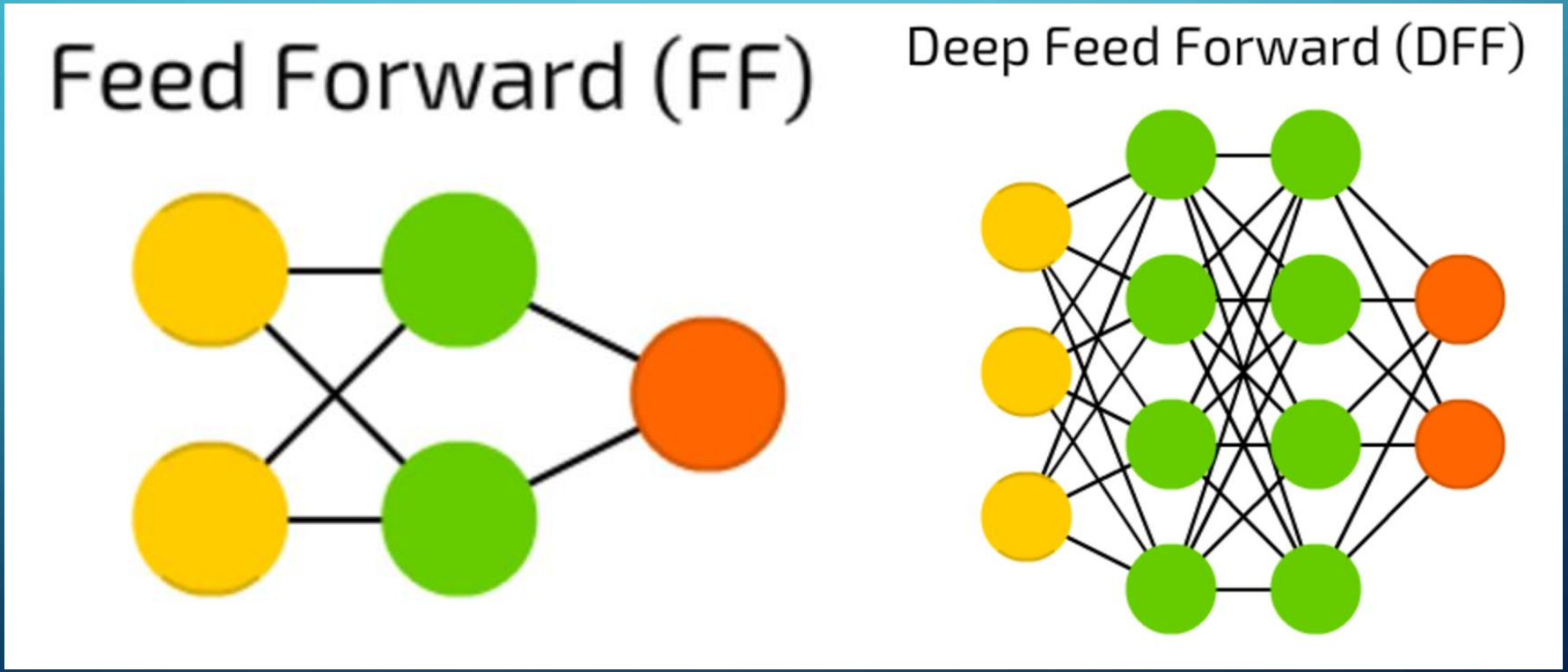
Optimizer and its hyperparameters

When to stop training

Random search – train many times, trying random values for the hyperparameters each time!

DEEP LEARNING

A deep neural network has more than one hidden layer

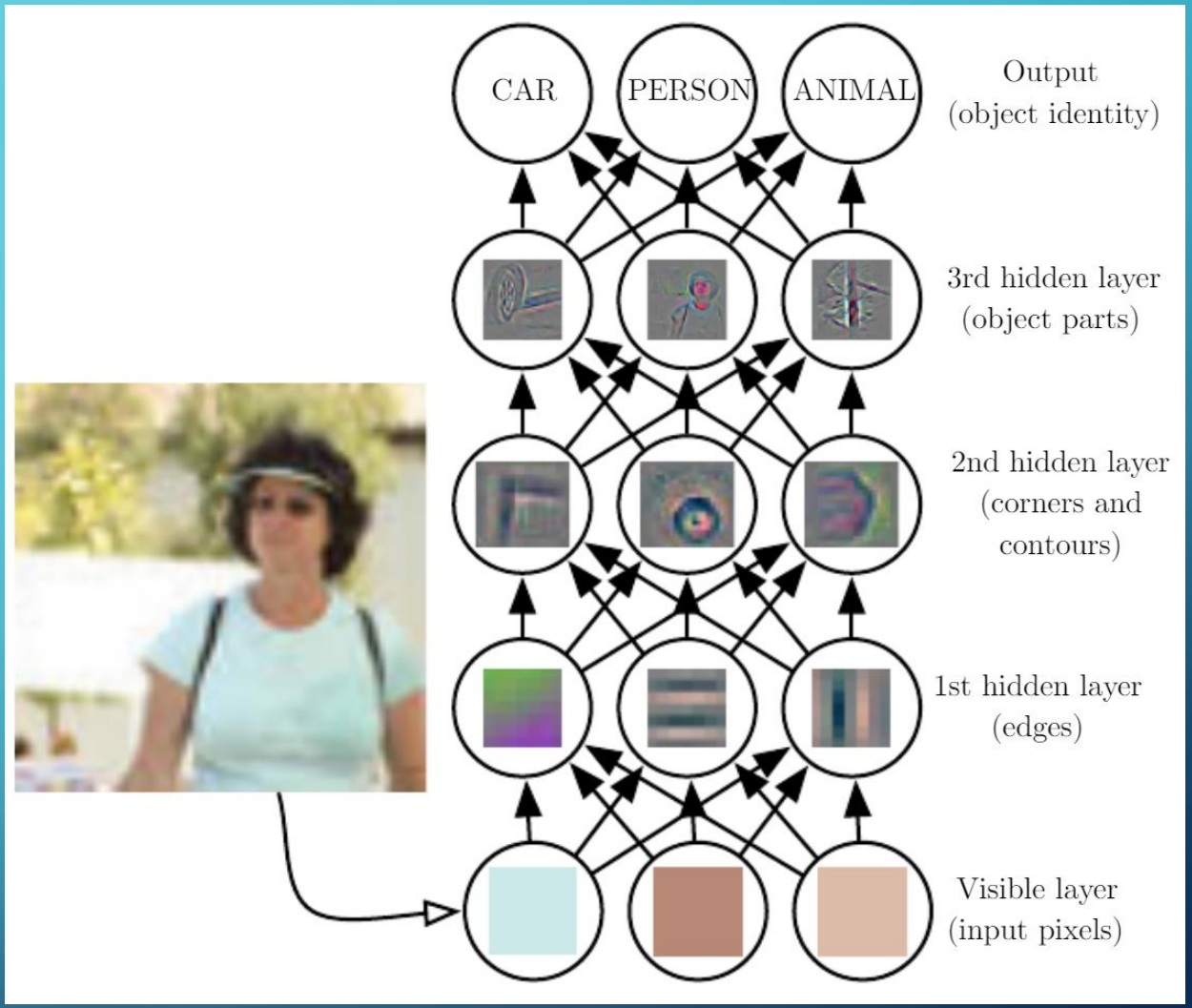


Neural Network Zoo, Fjodor van Veen

DEEP LEARNING

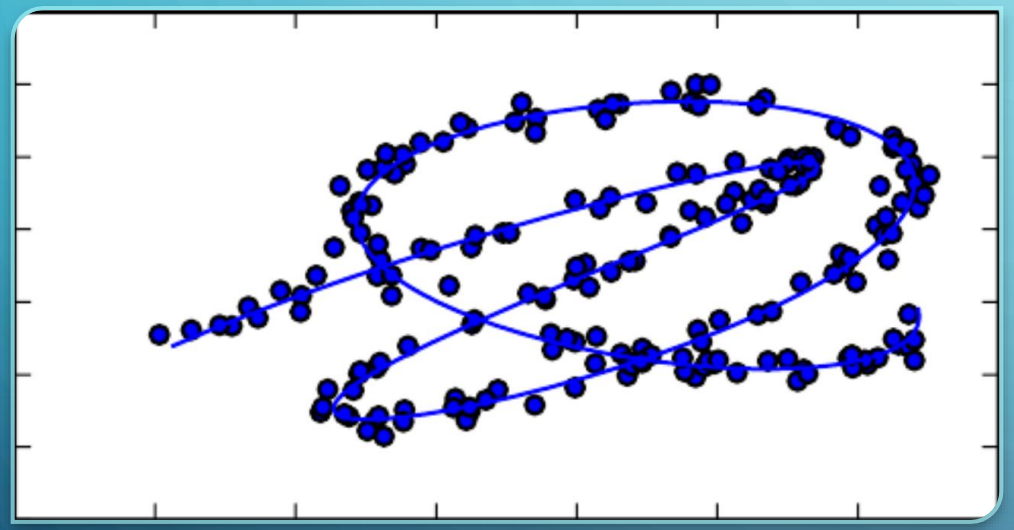
Each hidden layer extracts increasingly abstract information

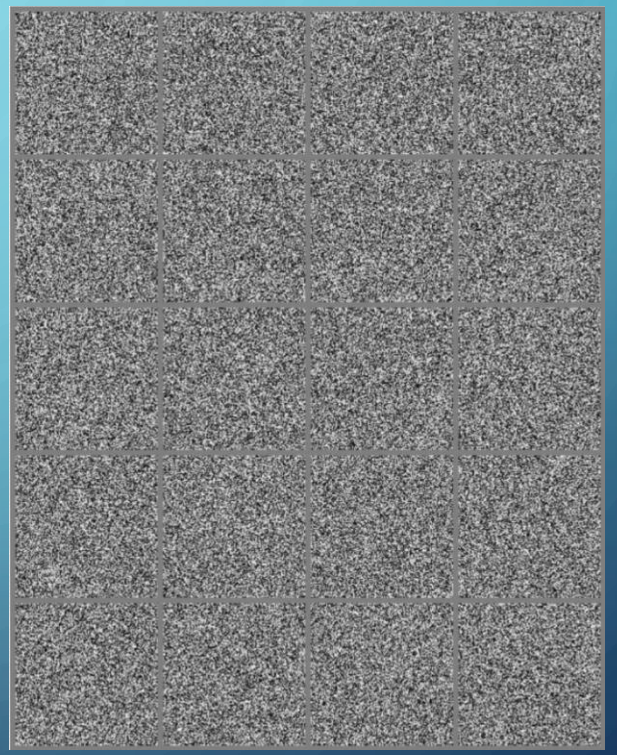
Don't need to hand-craft features – the network will make its own



Goodfellow, Bengio, and Courville (2016)
Zeiler and Fergus (2014)

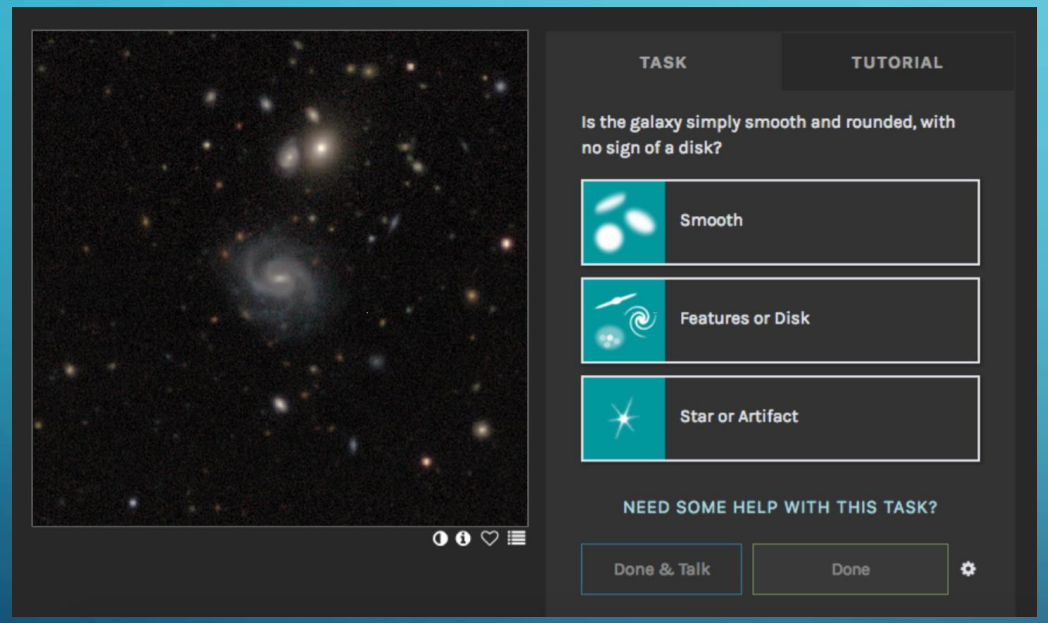
MANIFOLD HYPOTHESIS

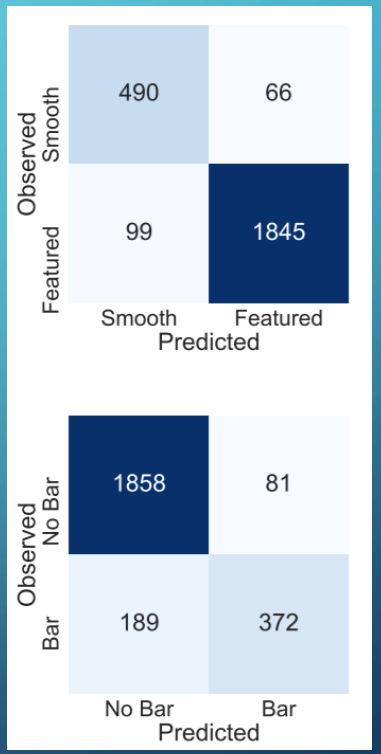




Goodfellow, Bengio, and Courville (2016)

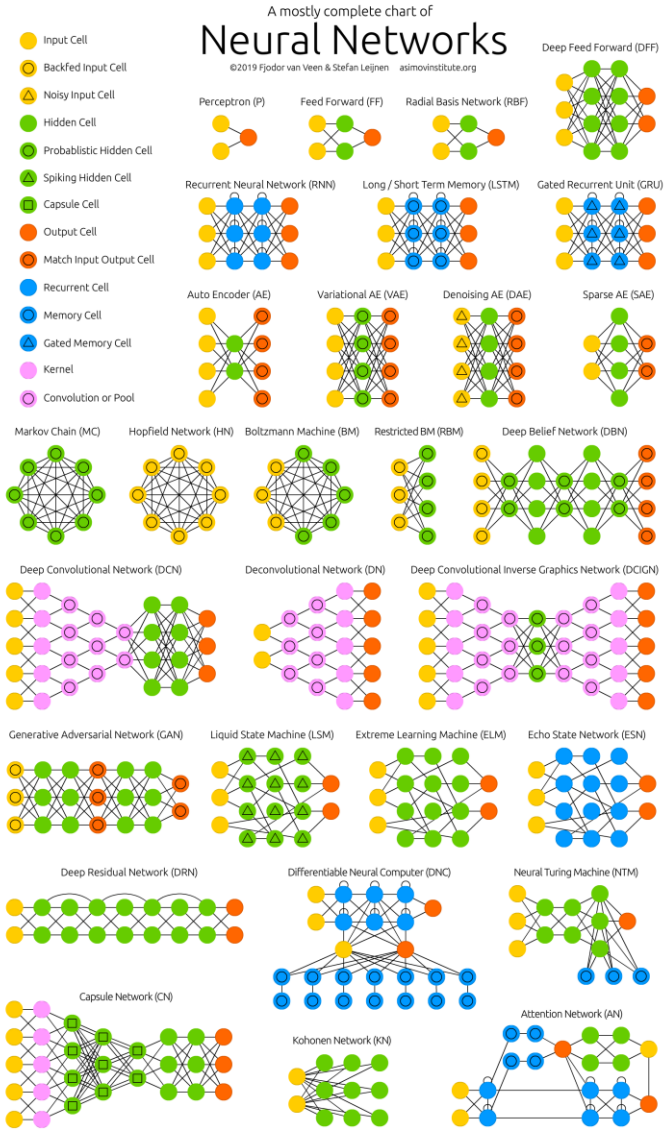
GALAXY ZOO





Walmsley et al. (2019)

THE NEURAL NETWORK ZOO



Neural Network Zoo, Fjodor van Veen

APPLICATIONS OF MACHINE LEARNING

1. Where a human does well, but rules are hard to codify

2. Datasets with complex correlations that are hard for humans to handle

In either case, lots of data is needed!