An Introduction to Machine Learning

Sundar Srinivasan
IRyA-UNAM
s.srinivasan@irya.unam.mx

https://tinyurl.com/EdV2025

Escuela de Verano de Astrofísica 2025, IRyA-UNAM

About you

Do you speak Python?

Have you heard of ML?

Have you used ML?

Outline

1. Machine learning: what and why
    a. Machine learning is prediction
    b. Why does astronomical data need machine learning?

2. Getting into the details
    a. Some terminology
    b. The train/test split (and validation)
    c. Classification/regression, supervised/unsupervised learning, parameters/hyperparameters
    d. The need for a metric

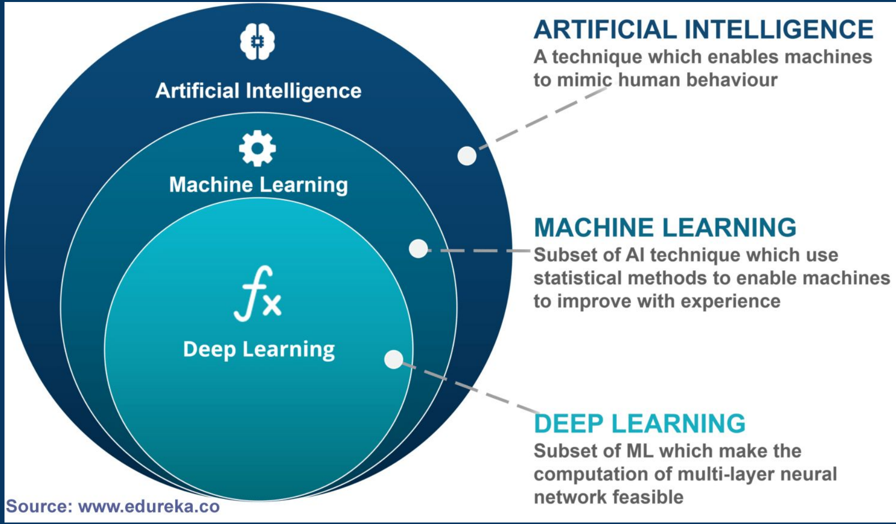
3. Machine learning is NOT a black box! It’s a… grey box
    a. Overfitting, underfitting, bias, variance, and the Bias-Variance Tradeoff
    b. Cross validation
    c. Model selection and hyperparameter tuning

4. Real data is more complicated!

5. Python demos

Machine learning: what and why

AI, ML, and Deep Learning



SU ELI5 the difference between AI, ML, and deep learning

Sure! I can explain the difference between AI (Artificial Intelligence), ML (Machine Learning), and deep learning in a way that a 5-year-old could understand:

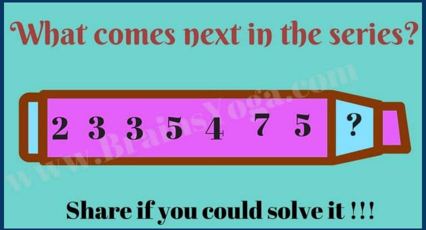
Imagine you have a toy car. AI is like the brain of the car. It helps the car make decisions and do things on its own, like figuring out how to avoid obstacles or find the fastest route.

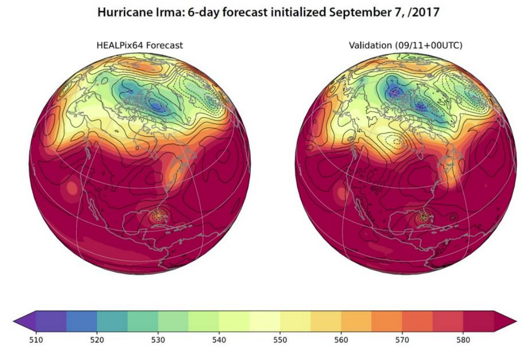
ML is like a teacher that helps the toy car learn from experience. The teacher shows the car many examples of driving, and the car learns to drive better over time. It's like the car is learning from its mistakes and getting better at driving.

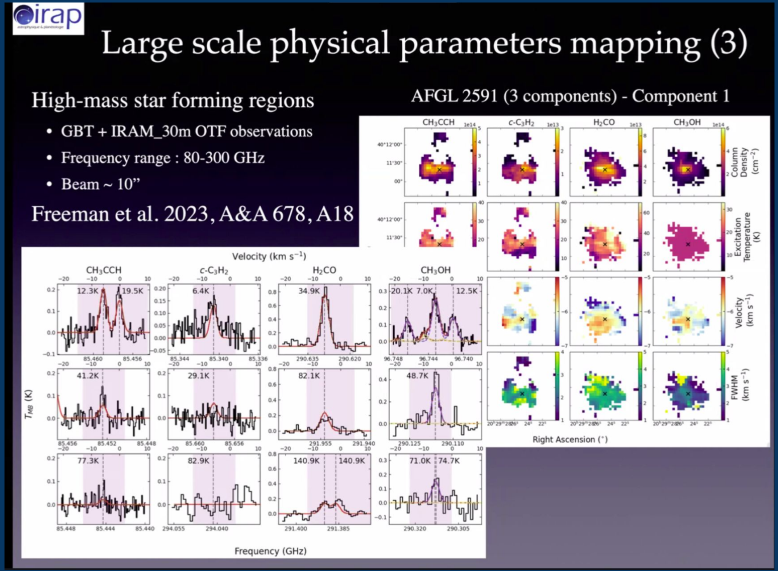
Deep learning is a special kind of ML that is even more powerful. It's like having a really smart teacher with lots of layers of knowledge. This teacher can understand more complex things and can teach the toy car to do really advanced tasks, like recognizing different objects or even driving itself without any human help.

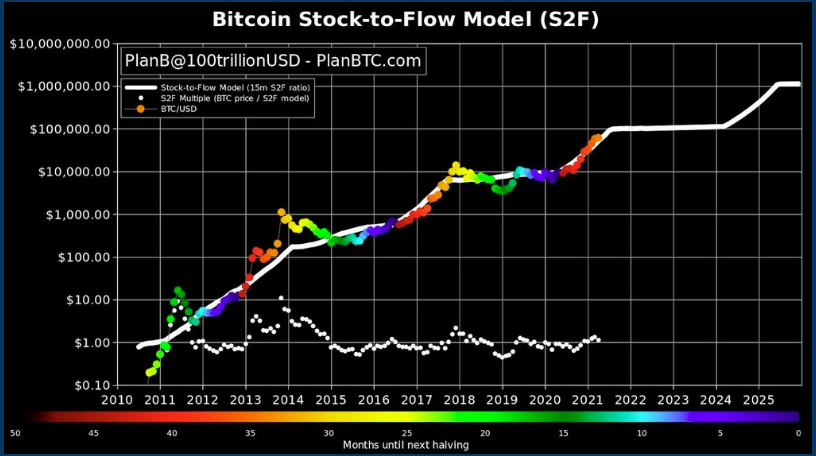
So, to sum it up: AI is the overall concept of making machines smart, ML is a way for machines to learn from examples, and deep learning is a more advanced type of ML that can tackle complex tasks.

Machine learning is prediction



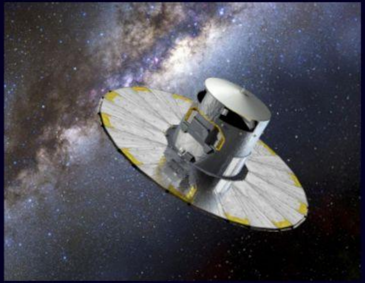






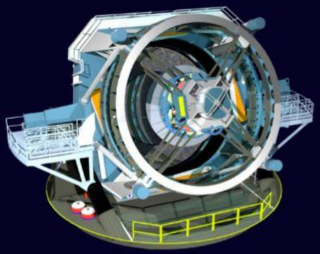
Why does astronomy need machine learning?

Gaia (2013–c. 2022)
European Space Agency



Astrometry for >10^9 objects, 60 TB @ 1 Mb/s

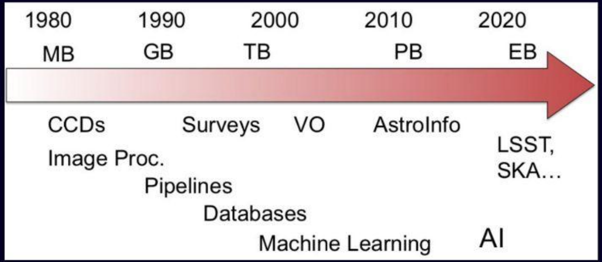
Vera C. Rubin Observatory (2020s)
LSST.org / CC BY-SA 3.0



30 TB/night, 100 Gb/s

Square Kilometer Array (2020s)
SKAtelescope.org / CC BY-SA 3.0

~EB of data, ~Pb/s



Increasing computing capability, storage, memory, and speed of connectivity has led to increase in **quality and quantity** of digital archives.

Improved internet access and speeds have improved accessibility of these archives.

Increase of **data-driven science** instead of hypothesis-driven science.
Rise of **citizen science** (e.g., zooniverse.org — classify galaxies on your phone).

Increase in information content: Most data will never be seen by humans.

Increase in information complexity: Patterns in these data cannot be directly comprehended by humans.

We need YOU!

The astronomical community is still slow in taking up statistics and automation. Most of us still do things manually.

The hope lies with the next generation of researchers.

Getting into the details

Terminology: features and labels

Mass (?units?)	**Animal**
1.63	Cat
44.2	Dog
2.35	Cat
19.7	Dog
3.7	Cat
25.9	???

- Learned the pattern from existing data, predicted result for "future" data
- Learning/prediction in general requires quantitative knowledge
- Extrapolation more difficult than interpolation [why?]

T_{eff} (K)	$\log_{10}(L_{star}/L_{sun})$	Metallicity	M_{star}/M_{sun}
5800	1.021	0.05	1.8
6500	0.317	0.12	1.2
20000	2.352	0.08	???
5000	-0.183	0.02	0.9
6700	0.704	0.15	1.5
6000	1.51	0.09	???
5500	0.0	0.04	1.0

Terminology: features and labels

Independent variables **(Features)**

Mass (?units?)	**Animal**
1.63	Cat
44.2	Dog
2.35	Cat
19.7	Dog
3.7	Cat
25.9	???

T_{eff} (K)	$\log_{10}(L_{star}/L_{sun})$	Metallicity	M_{star}/M_{sun}
5800	1.021	0.05	1.8
6500	0.317	0.12	1.2
20000	2.352	0.08	???
5000	-0.183	0.02	0.9
6700	0.704	0.15	1.5
6000	1.51	0.09	???
5500	0.0	0.04	1.0

Dependent variables **(Labels)**

Terminology: classification and regression

Mass (?units?)	Animal
1.63	Cat
44.2	Dog
2.35	Cat
19.7	Dog
3.7	Cat
25.9	???

- In general, we have **more than one feature** which we use to explain/predict the **label(s)**.

- Discrete label: **classification**
  Continuous label: **regression**

T_{eff} (K)	$\log_{10}(L_{star}/L_{sun})$	Metallicity	M_{star}/M_{sun}
5800	1.021	0.05	1.8
6500	0.317	0.12	1.2
20000	2.352	0.08	???
5000	-0.183	0.02	0.9
6700	0.704	0.15	1.5
6000	1.51	0.09	???
5500	0.0	0.04	1.0

Questions #1

```
class = 'cat'
```

How many features does this data have?

How many labels?

Terminology: training, test, and validation samples

class = ‘cat’

class = ‘dog’

class = ??

Terminology: training, test, and validation samples

Training sample

class = ‘cat’

class = ‘dog’

Test sample

class = ??

Terminology: training, test, and validation samples

Training sample

class = ‘cat’

class = ‘dog’

Validation sample

class = ??

Test sample

class = ??

The train/validation/test split

Standard procedure when using ML!

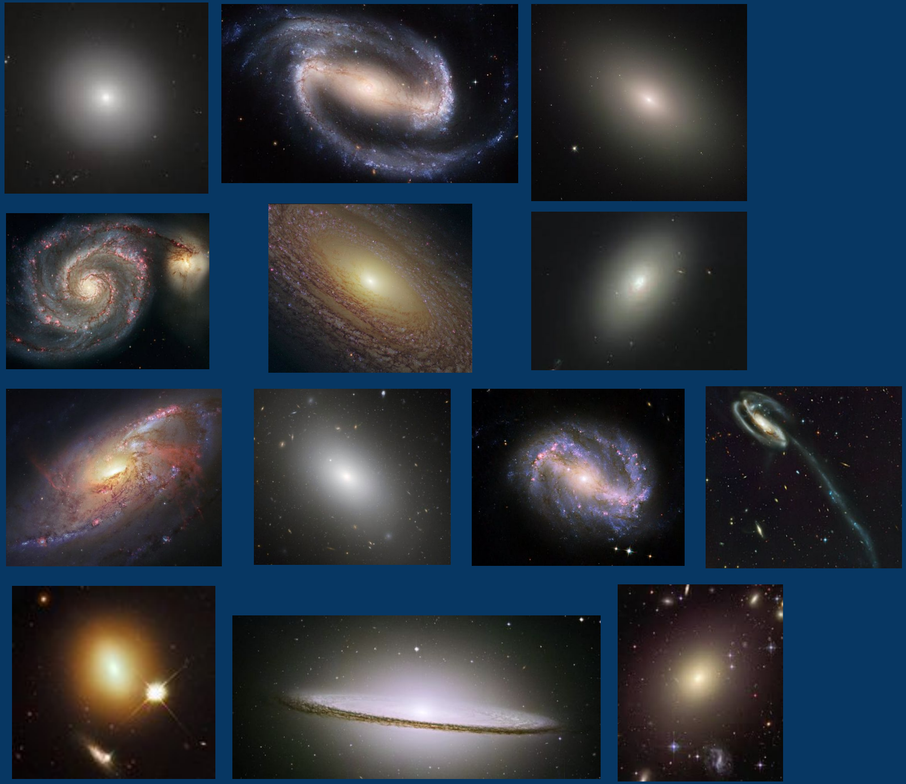
Randomly split your input data into a training sample and a validation/test sample (typically 80:20).

Learn from the training sample, predict labels for the validation/test sample.

Estimate accuracy of the learning by comparing predictions with true labels.

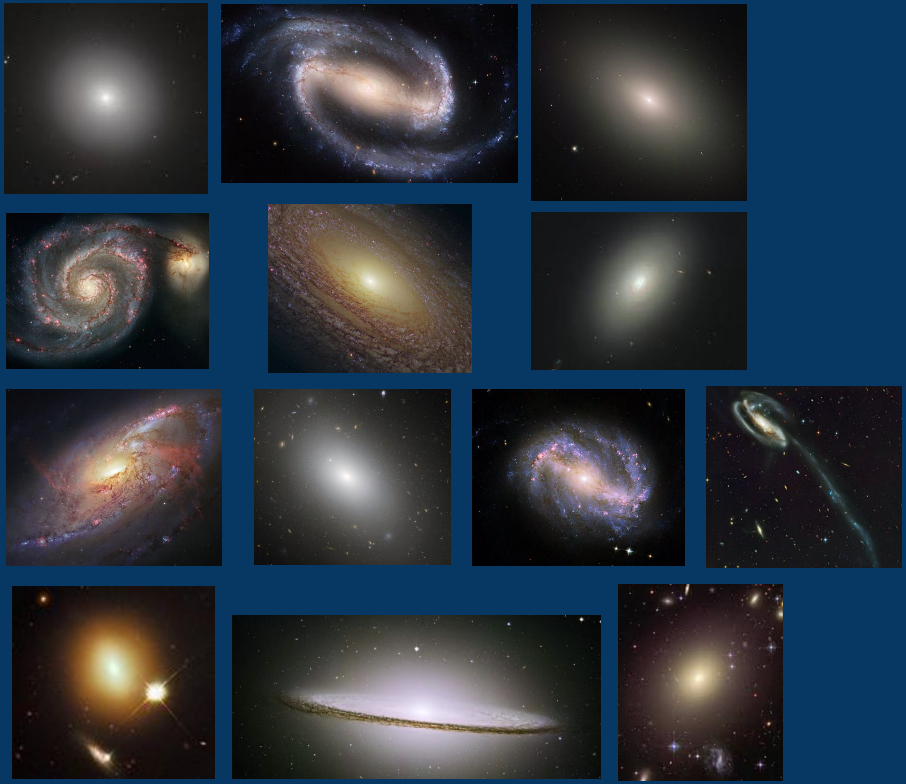
Split these images into two groups

Training sample



Split these images into two groups

Training sample



Separation criterion [model]

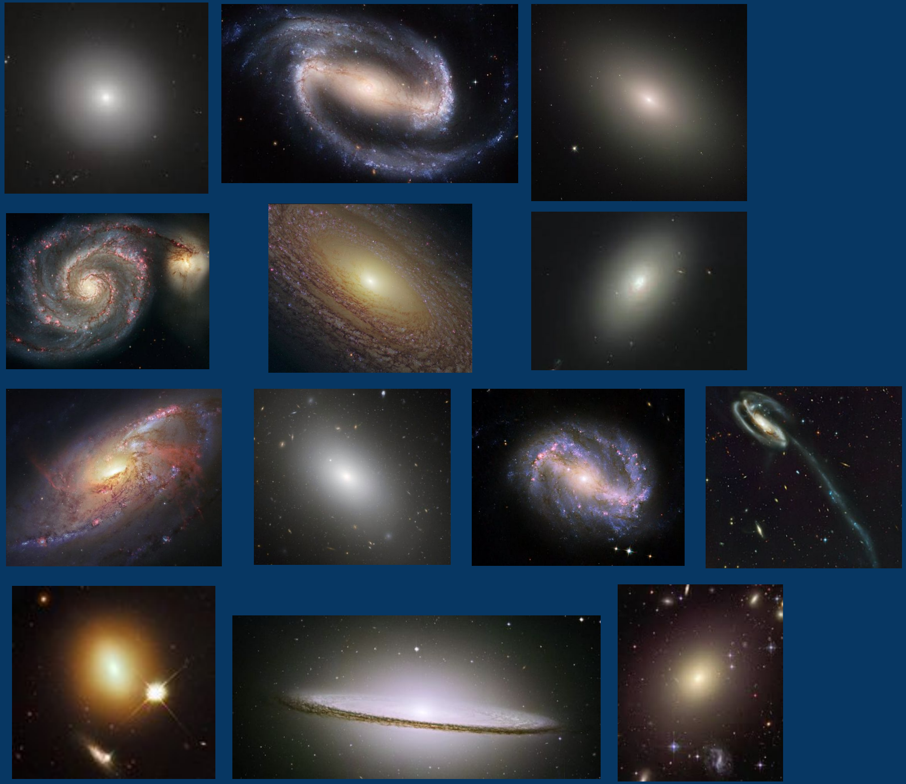
Does it have a disk?

Is it a spiral?

Is it alone?

To which group does this image belong?

Training sample



Test sample

Answer may depend on model.

Terminology: supervised and unsupervised learning

Supervised learning

Learn the model parameters from existing data (training sample has labels) and predict labels where they're not available (target/prediction sample).

Accuracy of model parameters can be checked by comparing label predictions with true labels on a validation sample.

Unsupervised learning

Refine model based on data features alone.

"Let the data speak for itself" (no labels).

Terminology: parameters and hyperparameters

Parameters

Internal variables of the ML model that are learned by training.

e.g., the slope/intercept of the best-fit line to some data.

Hyperparameters

External settings/configuration choices specified before training the model.

e.g., the metric used to determine the best-fit line to some data,

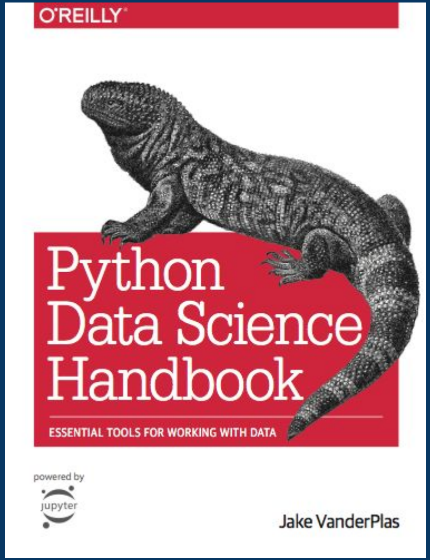
the optimisation algorithm used to find the optimum value of the metric,

the model/criterion used to group galaxies in the previous example.

ML = building models to explain and predict data

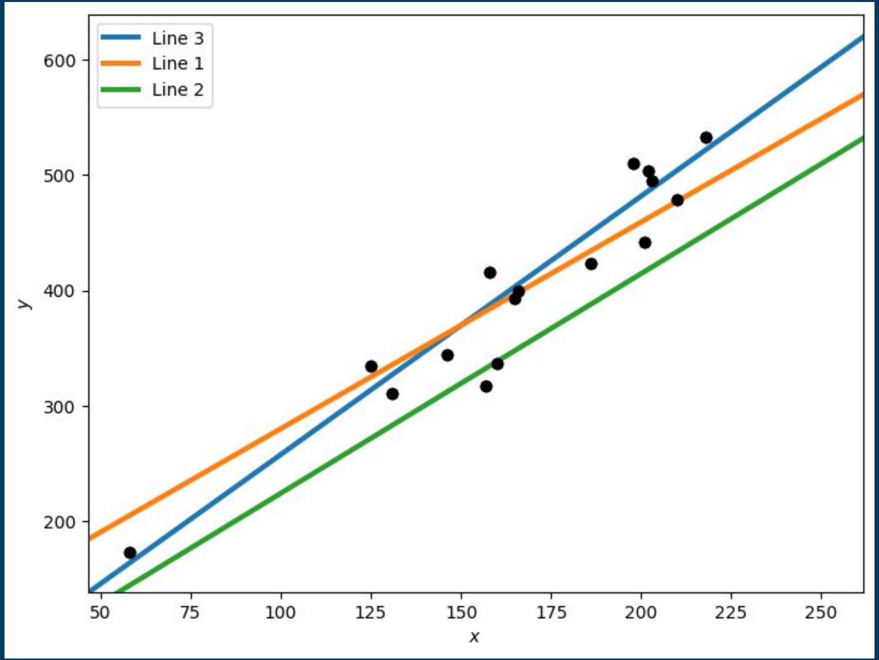
“[I]t's more helpful to think of machine learning as a means of *building models of data.*

Fundamentally, machine learning involves building mathematical models to help understand data. ‘Learning’ enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be ‘learning’ from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data.”



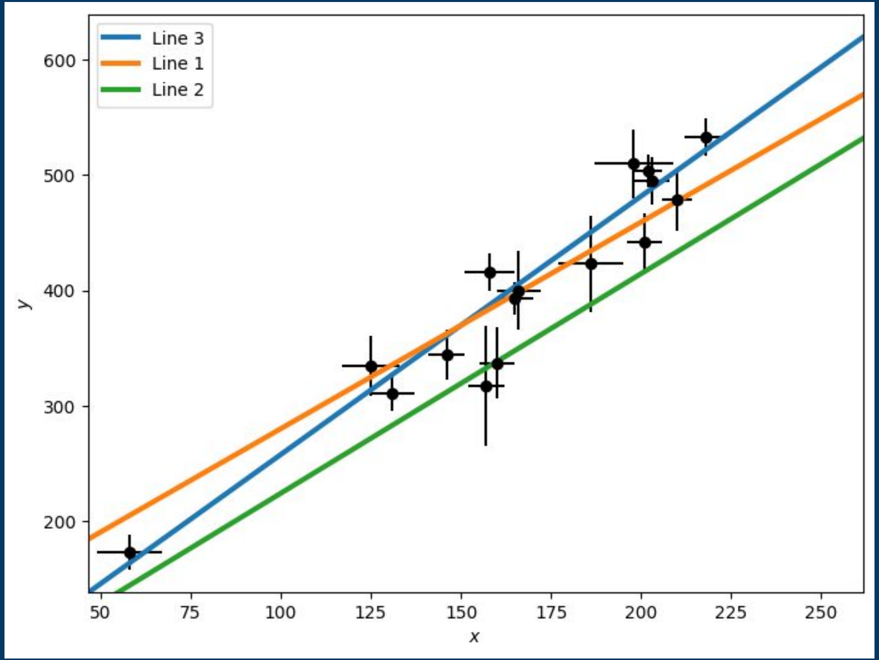
– J. Vanderplas, *Python Data Science Handbook*
https://github.com/jakevdp/PythonDataScienceHandbook

Questions #2



Which line best represents the trend in the data? (Which line is the “best fit”?)

Questions #3

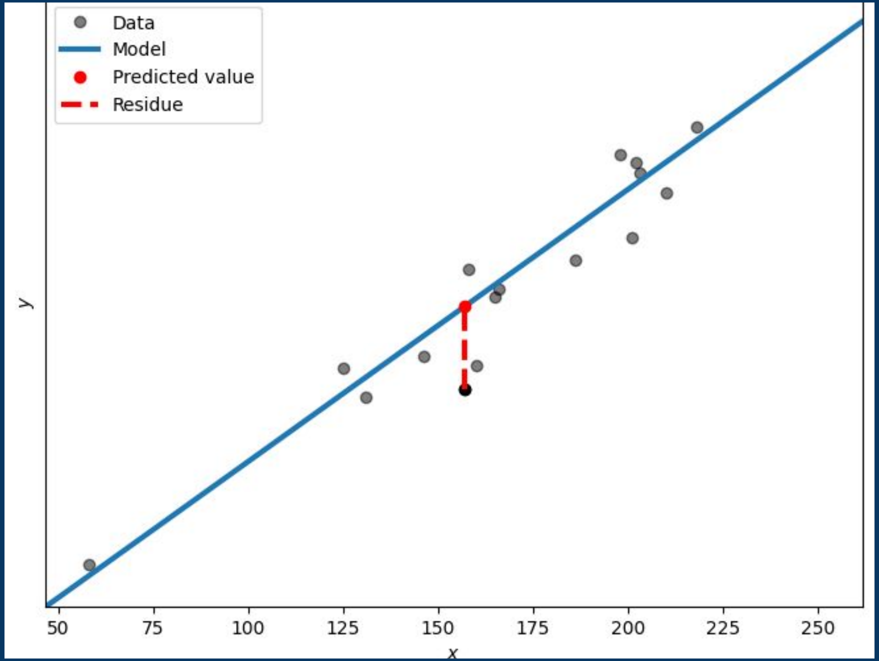


The data now have uncertainties.

Which line best represents the trend in the data? (Which line is the “best fit”?)

How did you determine the best-fit line?

Metric



How do I know the algorithm actually learned the trends in the data?

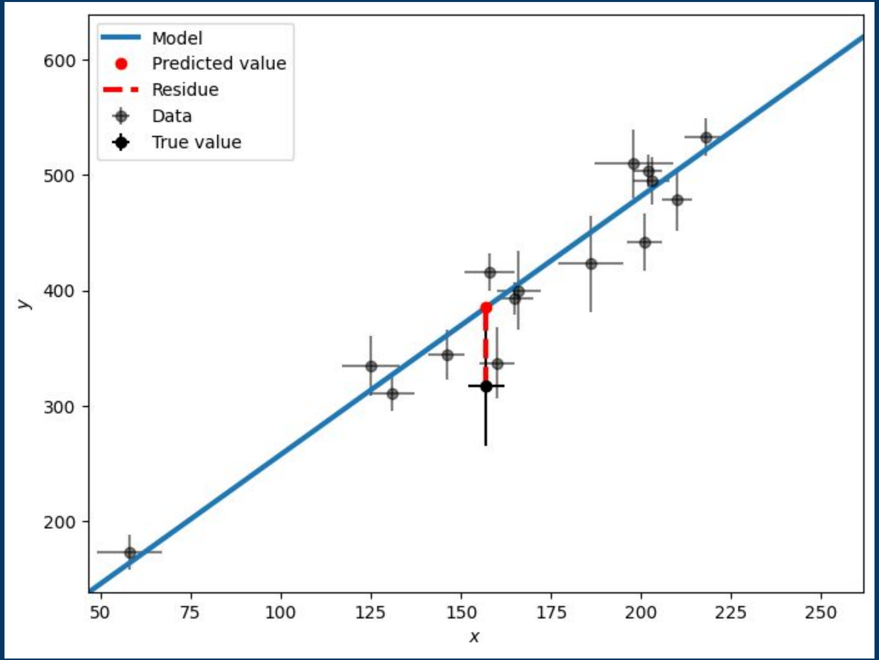
Compare prediction to actual (true) value.

The best-fit (intercept, slope) combination will be such that the average residue will be minimum.

Note: square the residues before summing them!

Sum of squares of residues = mean squared error (MSE)

Metric



For data with uncertainties, we must compare the residue to the size of the error bar (uncertainty).

This is called the standardised sum of squares of residues [because we divided by the standard deviation], or chi-squared.

For astronomical applications, we will typically have uncertainties, so don't stick with the default MSE!

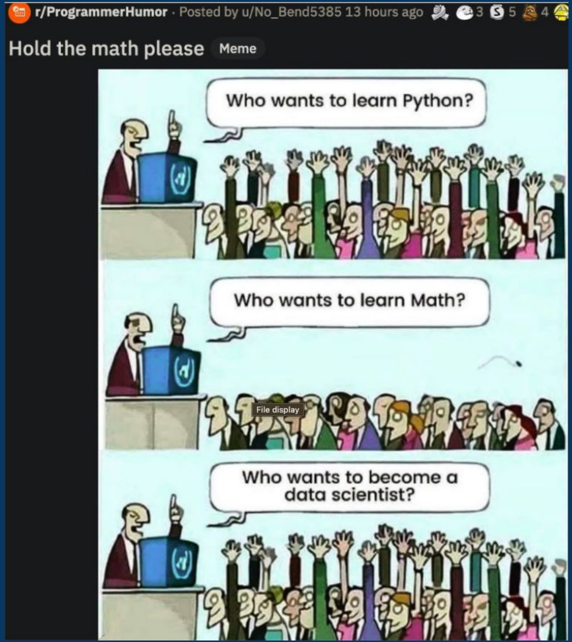
Metric

The MSE is only one of many possible metrics. The metric is any function of the data and the model.

Example: instead of squaring, take the absolute value and then average it.
OR median of squares OR median absolute value....

Machine learning is NOT a black box
(a.k.a. statistics matters)

Brush up on your mathematics!



Astronomers NEED mathematics to properly interpret ML results!

We are not only interested in fitting observations, we want to use the models to understand Nature!!

Astronomical data naturally contain statistical effects such as noise/uncertainty, missing or incomplete information, unbalanced/underrepresented classes, and observational biases.

A mathematical treatment is required in order to correctly apply ML algorithms to these data!

We can use physical and probabilistic laws to compute meaningful models and to eliminate unrealistic ones.

Linear algebra (matrices, vectors), calculus, probability, and statistics are handy tools to have on your side!

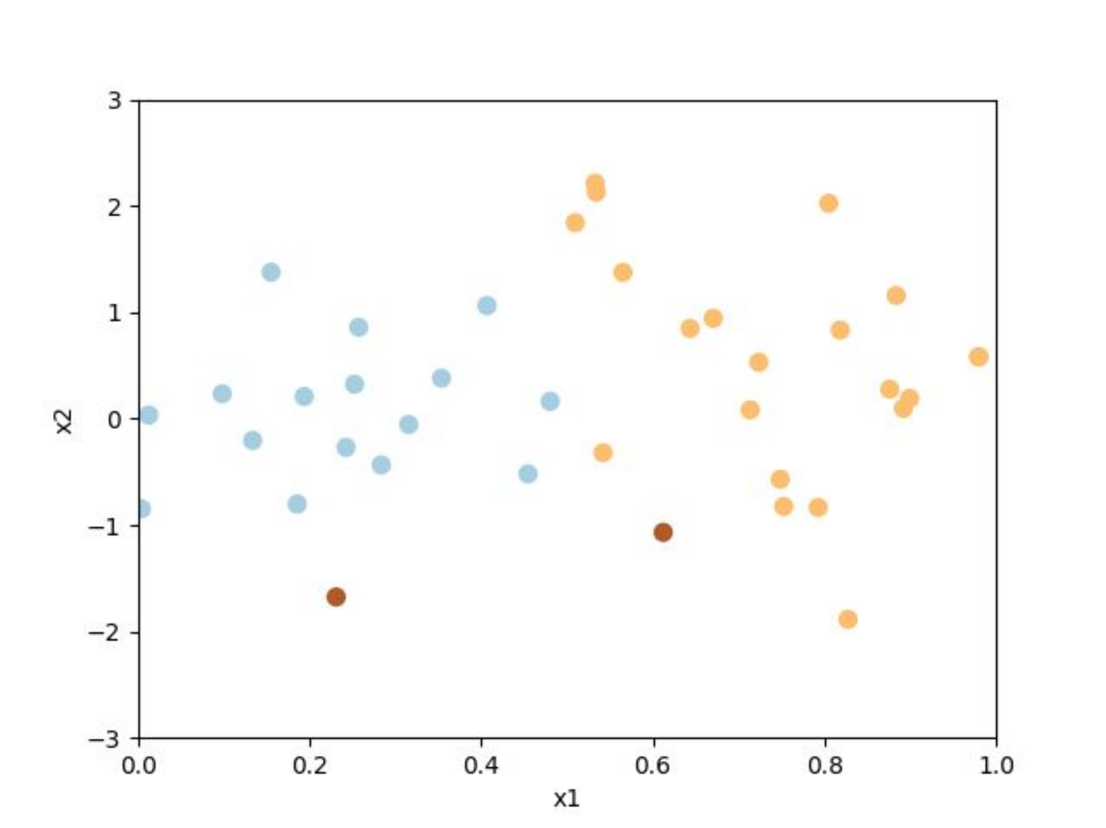
Cross validation

Watch this video for a basic introduction and demo

Setting up ML or neural network codes for astronomical datasets: many switches to flip.

Cross validation is ESSENTIAL to select the “best combination” of settings.

Illustration

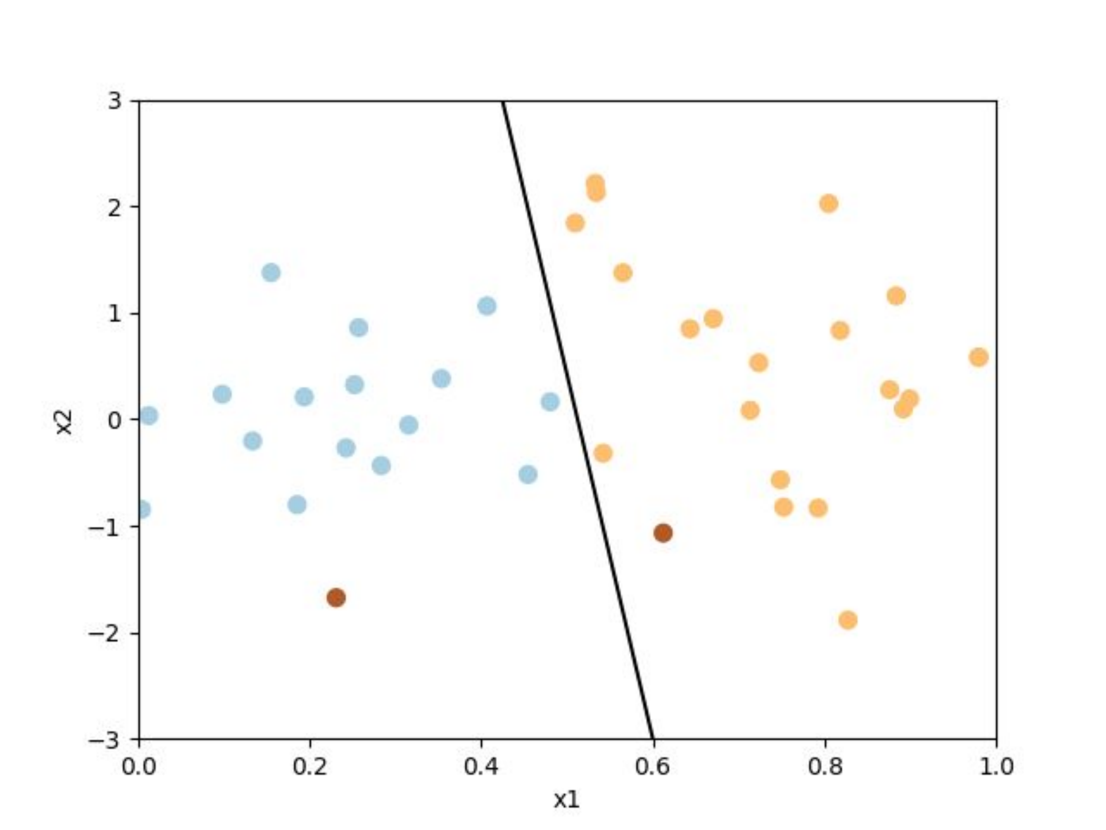


38 data points with two features x1, x2 and one label y (blue or orange).

Label predictions desired for two points (brown).

Model: straight line to separate the two classes. Points to the “left” will be classified as “blue”.

Illustration



This line is a good decision boundary for the given data, as it correctly classifies all blue points as blue.

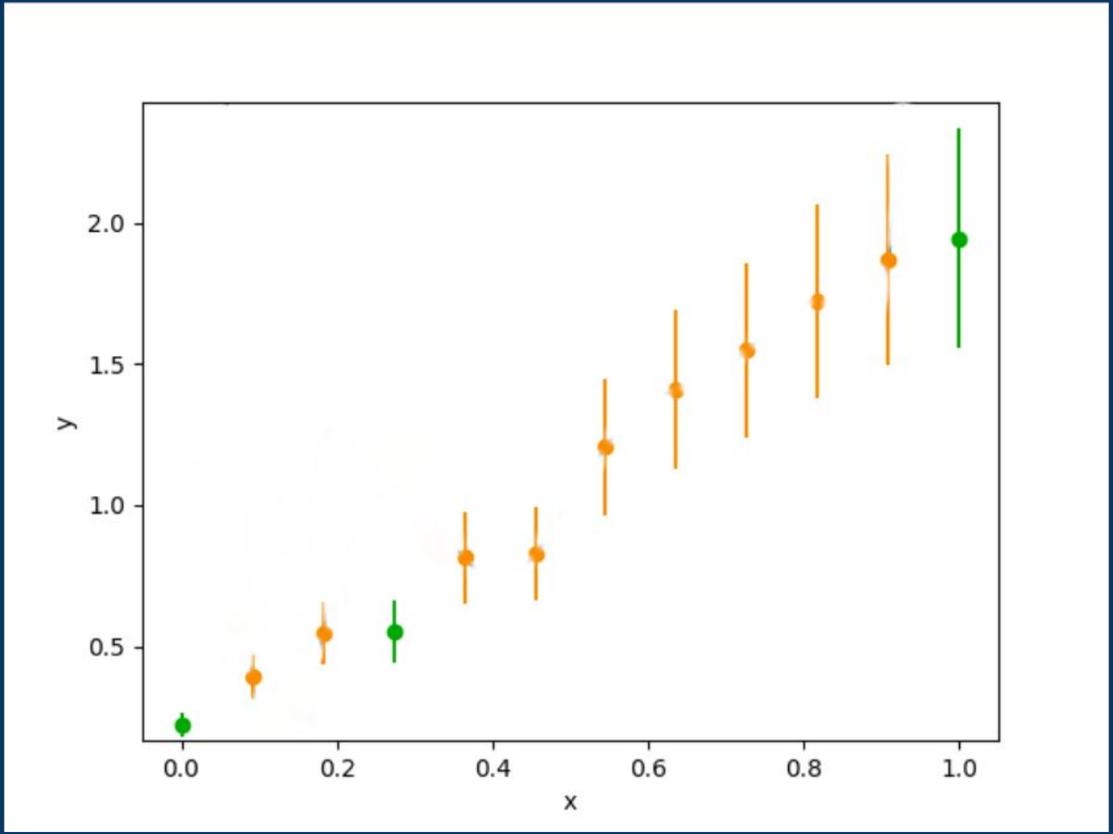
How to pick this boundary?

Model score (how well model reproduces the training data).

Score depends on model complexity (Bias-Variance Tradeoff).

We can now predict labels for the two brown points!

Overfitting

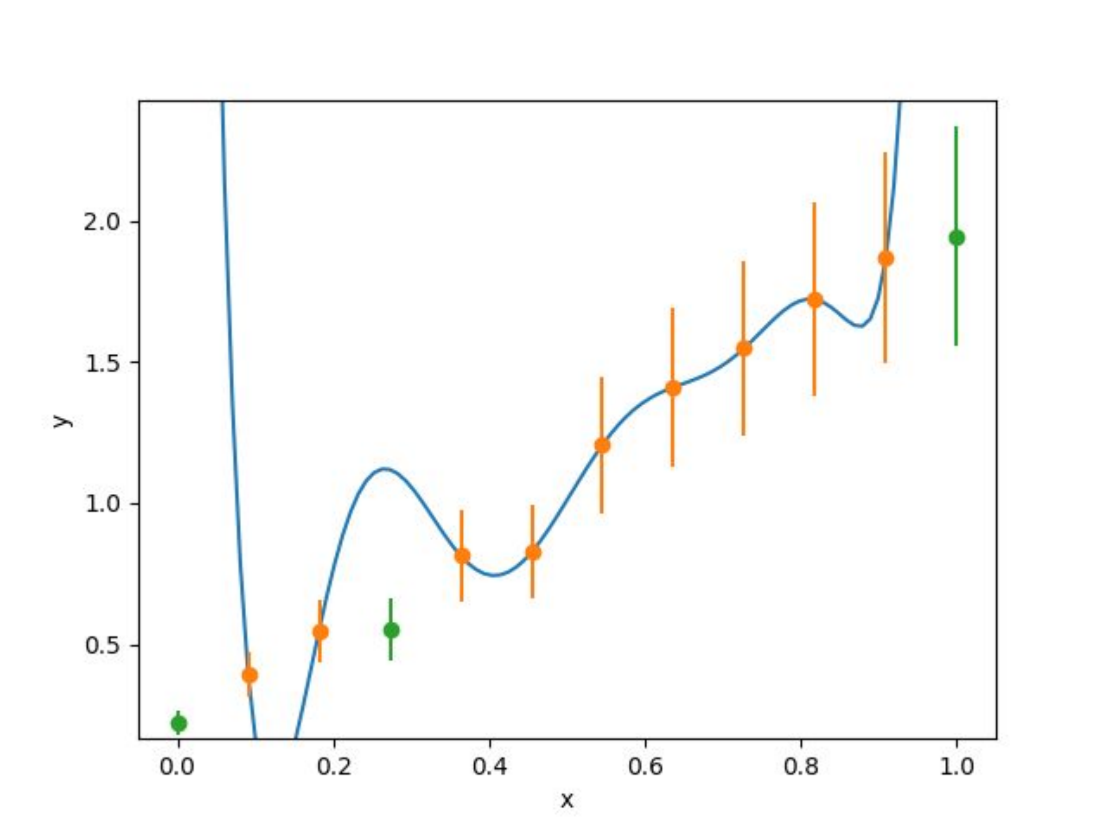


Training data = orange
Validation data = green

We will use the training data to learn the best-fit model and then predict the values for the validation data.

We could fit a straight line, but why stop there? Why not a more complex model that will pass through all the training points??

Overfitting

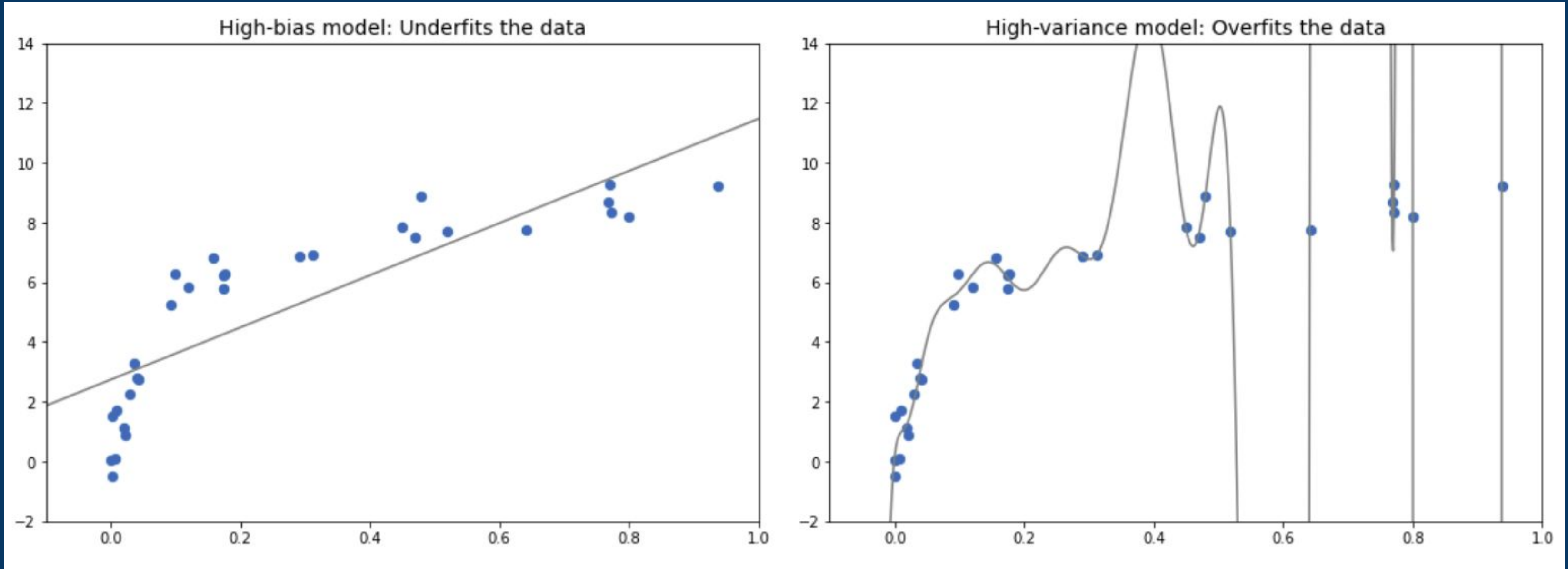


This model is an exact fit to the training data (orange points)!

BUT its out-of-sample (green points) predictive ability decreases, or its variance increases

Model complexity also function of number of data points (e.g., you can't fit a 4th-degree polynomial to 5 data points)

The Bias-Variance Tradeoff



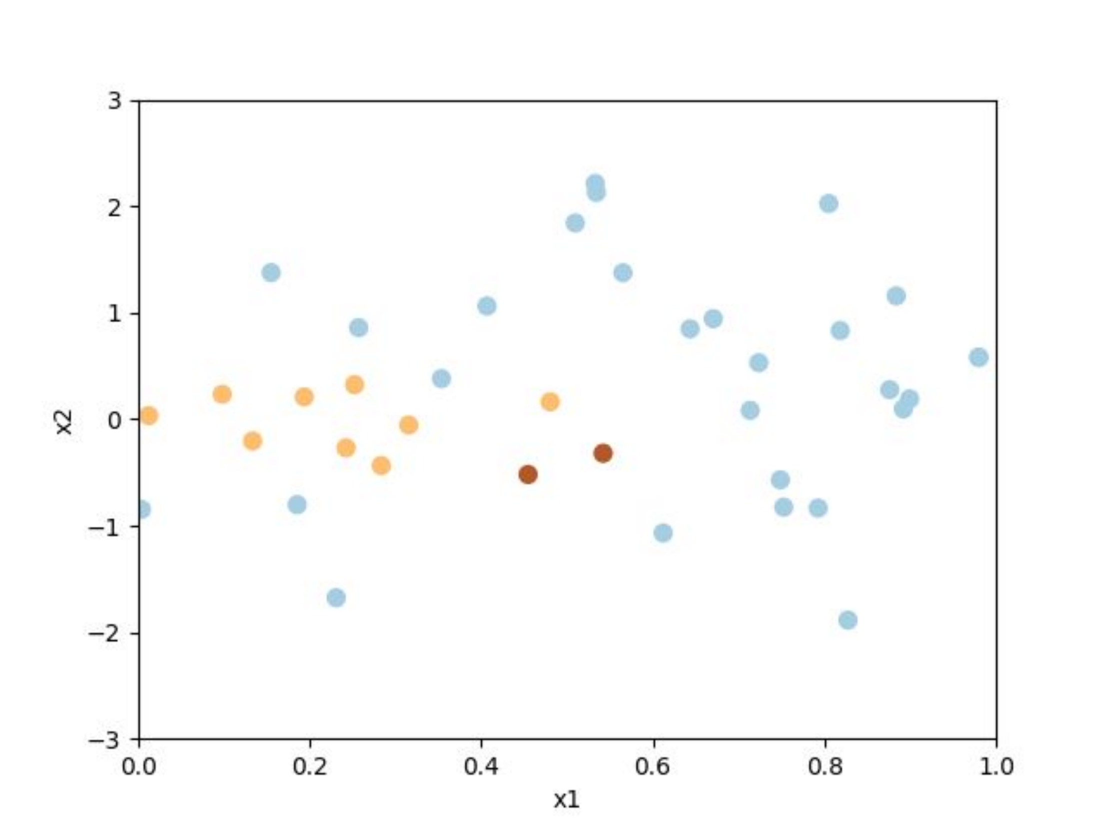
Bias: how well you can reproduce the training sample.

Less complexity = more bias.

Variance: deviation of model prediction from the true values for the validation sample (any data with labels that wasn't part of the training sample).

More complexity = more variance.

Illustration: a more complicated case

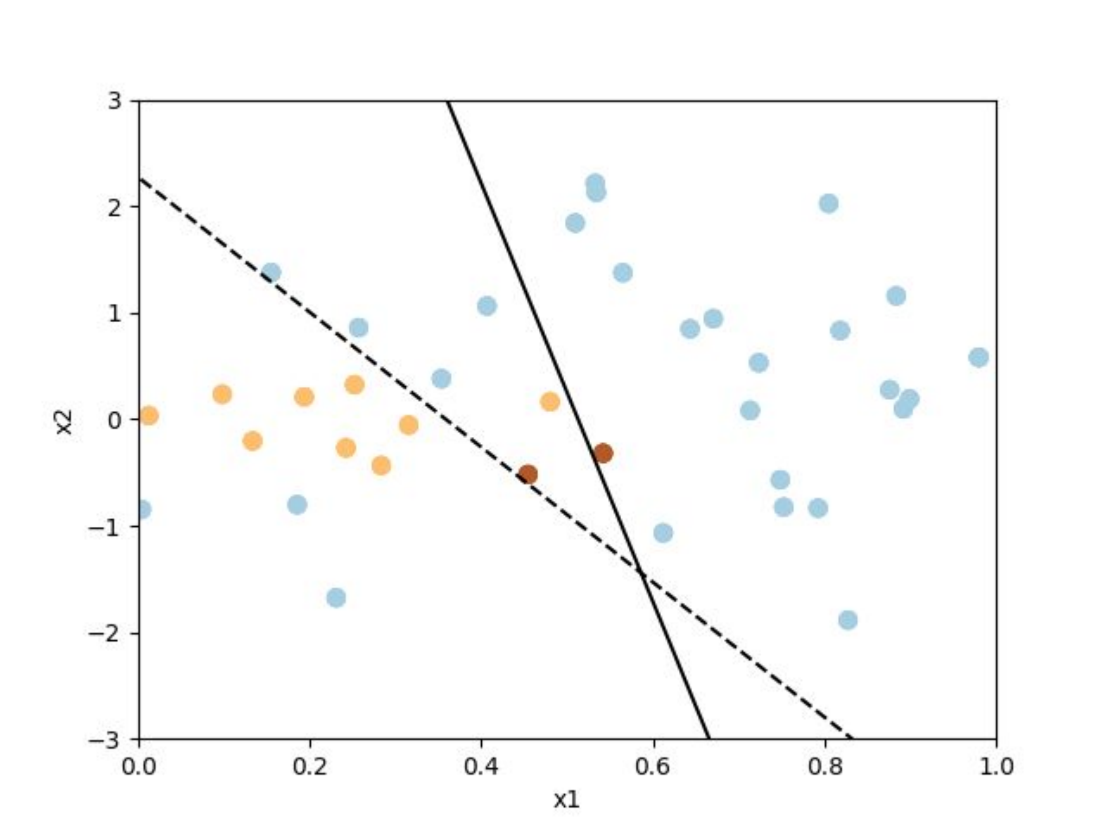


- Unbalanced sample (many more blue points than orange ones)

- Straight line can't perfectly separate the labels.

Prediction for the two brown points also affected.

The best-fit model depends on utility

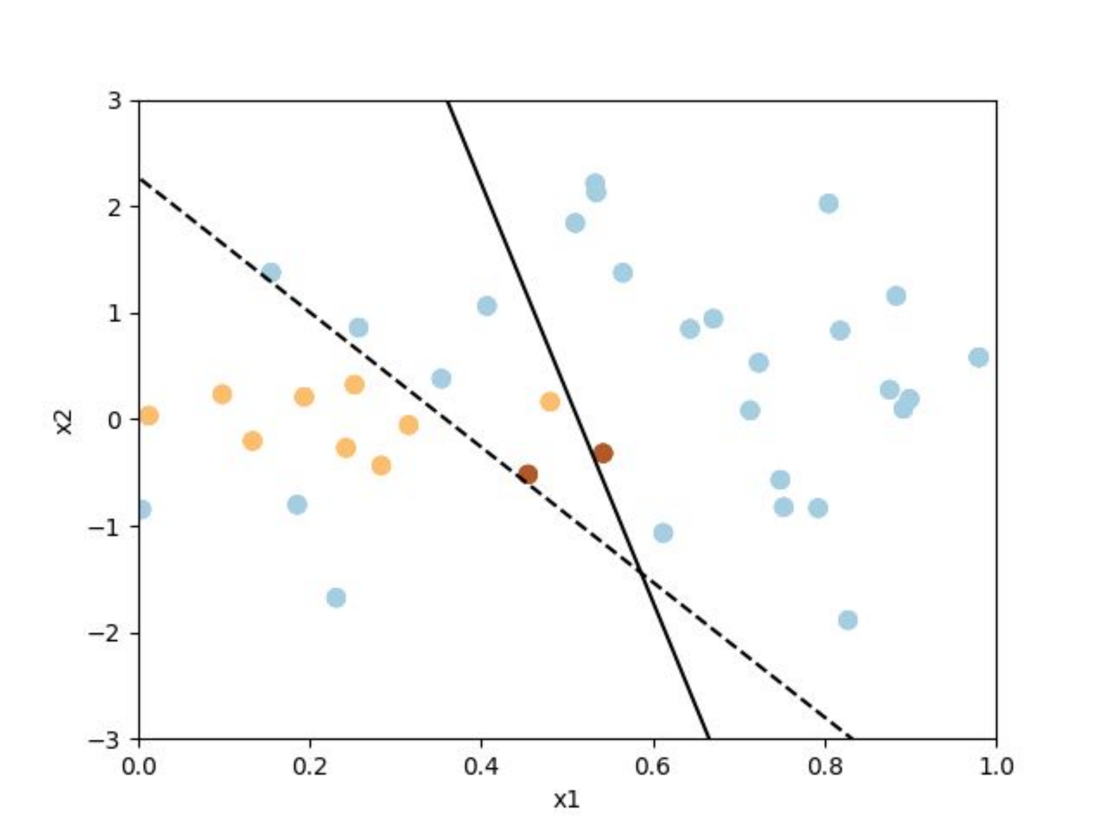


Solid line correctly identifies all orange points (true positives), but also incorrectly identifies six blue points as orange (false positives).

Dashed line has a lower false positive rate, but also a lower true positive rate. It also incorrectly classifies an orange point as blue (false negative).

Which one should we pick?

The best-fit model depends on choice of metric



Accuracy = Fraction of total points that are correctly classified
(solid: 31/38, dashed: 34/38)
Dashed line gives higher accuracy

Precision = TP / (TP + FP)
(solid: 9/16, dashed: 8/11)
Dashed line gives higher precision

Recall = TP / (TP + FN)
(solid: 1, dashed: 8/9)
Solid line gives higher recall

Choice of metric depends on how results will be used.
Choice of metric determines choice of model!

Real data is more complicated!

What standard ML tutorials don't include

1. **Astronomical data is inherently RANDOM. It has uncertainties.**

   There are many techniques to incorporate randomness into ML algorithms.
   Statistics helps!

2. **MISSING DATA is also USEFUL INFORMATION!**

   If a star is not detected at a certain wavelength, it tells us something useful about its properties. We need to be able to incorporate this into our learning.
   A useful ML algorithm must be OK with NaNs in the data.

3. **Astronomical data may be UNBALANCED.**

   If you only have images of dogs in your training sample and the test sample has an image of a crow, your reaction will be, "Huh, that's a very strange dog."
   Oversampling/undersampling techniques must be used for proper learning.

Let's work together!

https://www.irya.unam.mx/gente/s.srinivasan/Research/index.html

Key member of the Nearby Evolved Stars Survey (NESS; https://evolvedstars.space).

Research interests:

- Observations and modeling of cosmic dust in various environments
- Evolved (especially asymptotic giant branch) stars
- Statistics, data science/analysis

Statistics and machine-learning applications to research:

- Classification of photometry and spectra
- Neural networks for regression
- Grid search cross-validation for model selection/validation
- Markov Chain Monte Carlo sampling for solutions to astrophysical problems

Github: https://github.com/sundarjhu/

After the break: Python demos

Notebooks are based on those by Jake Vanderplas from his book,
"*Python Data Science Handbook*".
[text: CC-BY-NC-ND, code: MIT]

Notebooks: Scikit-Learn intro

X → [n_samples x n_features], y → [n_samples]

The Estimator API

Supervised regression

Notebooks: Hyperpars/model validation

Cross validation

Model complexity, bias-variance tradeoff

Grid search

Notebooks: Linear regression

Cross validation

Model complexity, bias-variance tradeoff

Grid search