An Introduction to Machine Learning

("sólo para el sabor")

Sundar Srinivasan
s.srinivasan@irya.unam.mx

with student volunteers Natalia Osorio & Alejandro Vasquez

Escuela de Verano 2023, IRyA-UNAM

About me

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

Research interests: observations and modeling of dust in various environments, stellar evolution, evolved (especially asymptotic giant branch) stars, statistics, data science/analysis.
Key member of the Nearby Evolved Stars Survey (NESS; https://evolvedstars.space).

ML application to research: random forest for spectroscopic classification, neural networks for regression, grid search cross-validation for model selection/validation, support vector machine + SMOTE for photometric classification of unbalanced samples.

Python knowledge: beginner/intermediate

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

About you

Do you speak Python?

Have you heard of ML?

Have you used ML?

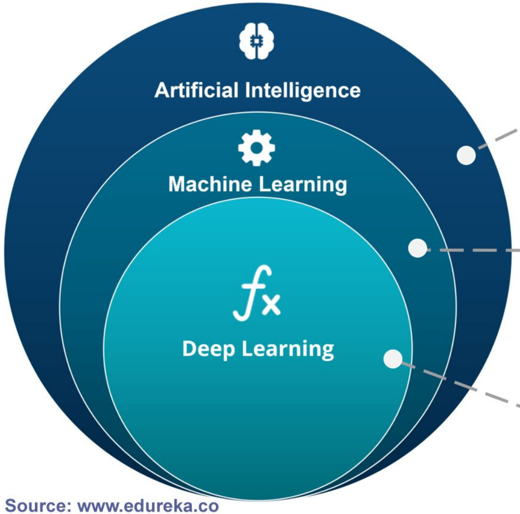
About this workshop

Please follow updates to the Github repository!

https://www.github.com/sundarjhu/EscueladeVerano_2023

Final update on Thursday to include assignments and other information.

AI, ML, and Deep Learning



Source: www.edureka.co

ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour

MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience

DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

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.

Intuition

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

Probability

- **Learned** the pattern from existing data, **predicted** result for “future” data → **Supervised** learning
- 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	30.32	0.05	1.8
6500	29.54	0.12	1.2
20000	44.51	0.08	???
5000	28.96	0.02	0.9
6700	29.68	0.15	1.5
6000	30.08	0.09	???
5500	29.76	0.04	1.0

Terminology

Features

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

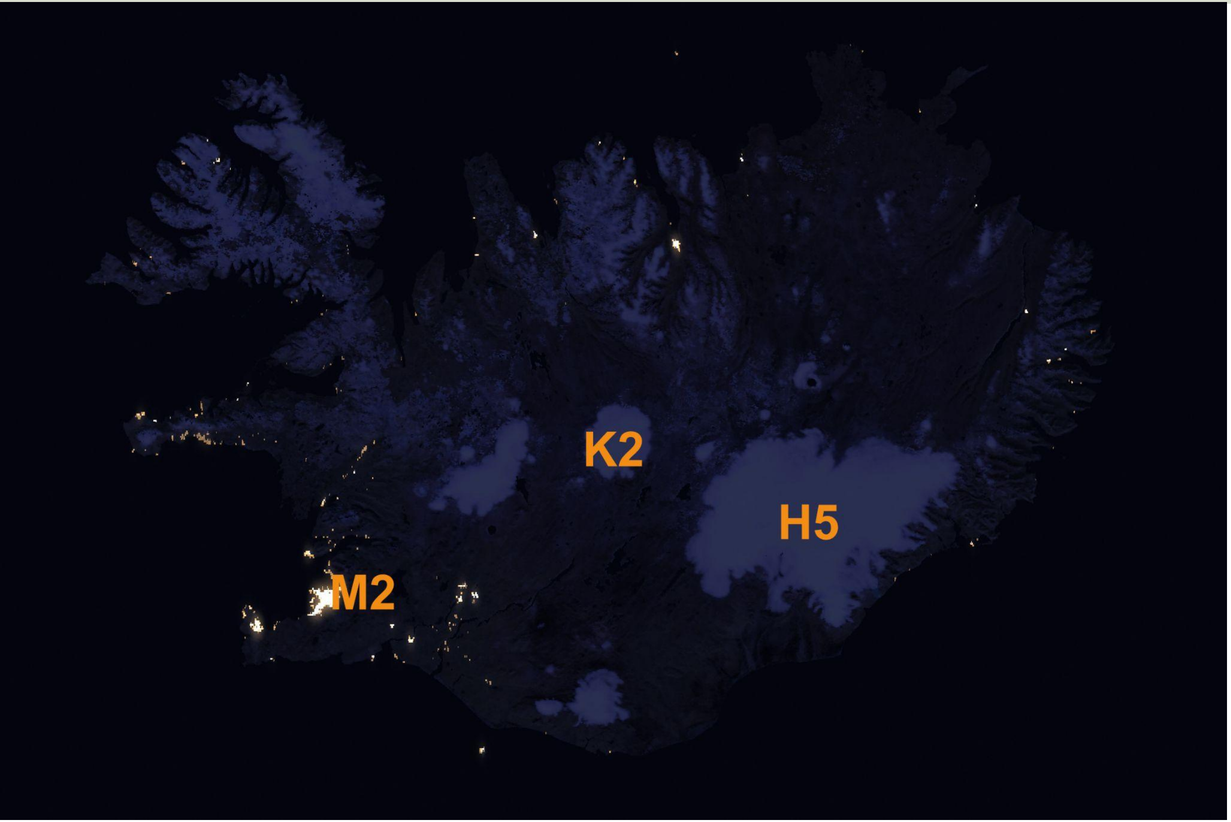
T_{eff} (K)	$Log_{10}(L_{star}/L_{sun})$	Metallicity	M_{star}/M_{sun}
5800	30.32	0.05	1.8
6500	29.54	0.12	1.2
20000	44.51	0.08	???
5000	28.96	0.02	0.9
6700	29.68	0.15	1.5
6000	30.08	0.09	???
5500	29.76	0.04	1.0

Labels

Intuition: *Althings* Iceland

Locate Iceland's two largest cities and the international airport for its capital.

Unsupervised learning



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

Clustering, density estimation, dimensionality reduction, etc.

e.g., "is that tiny dot a suburb of Reykjavik, or a neighbouring town?"

ML problems: an incomplete categorisation

Supervised

Learn the model parameters from existing data (**training sample**, has labels for each combination of the features) 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 **test sample**.

Unsupervised

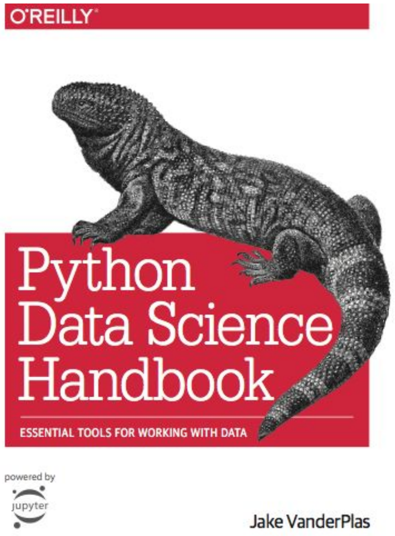
Refine model based on data features alone.

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*.

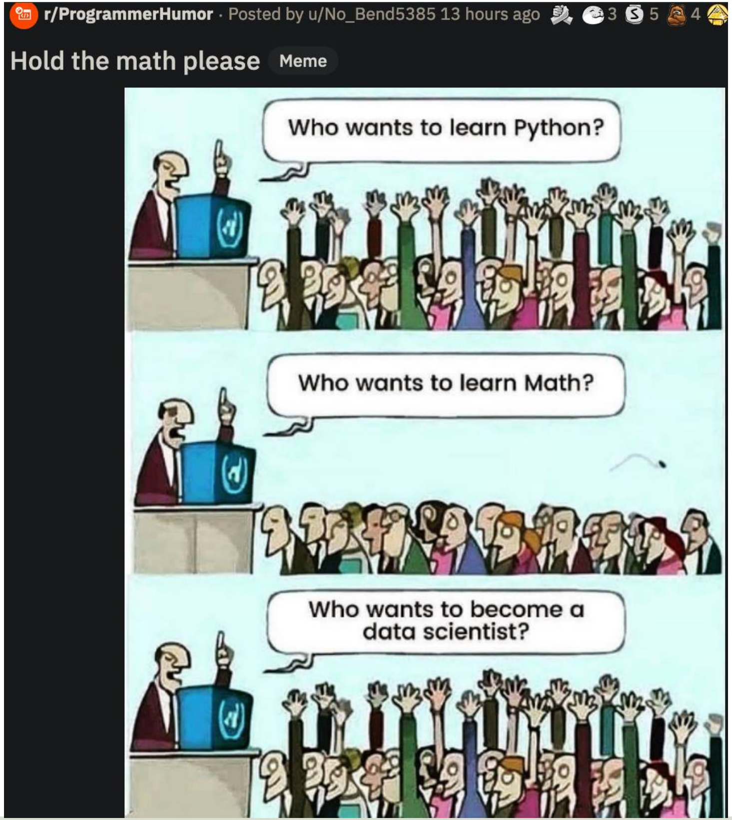
Fundamentally, machine learning involves building mathematical models to help understand data. ‘Learning’ enters the fray when we give these models ***tunable parameters*** that can be adapted to observed data; in this way the program can be considered to be ‘learning’ from the data. Once these models 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



Terminology:
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, or
the ***optimisation algorithm*** used to find the optimum value of this metric.

Brush up on your maths!



Astronomers NEED mathematics to understand ML. Why?

We are interested in the reasons for the existence of patterns in the data. We are interested in describing our observations of Nature!

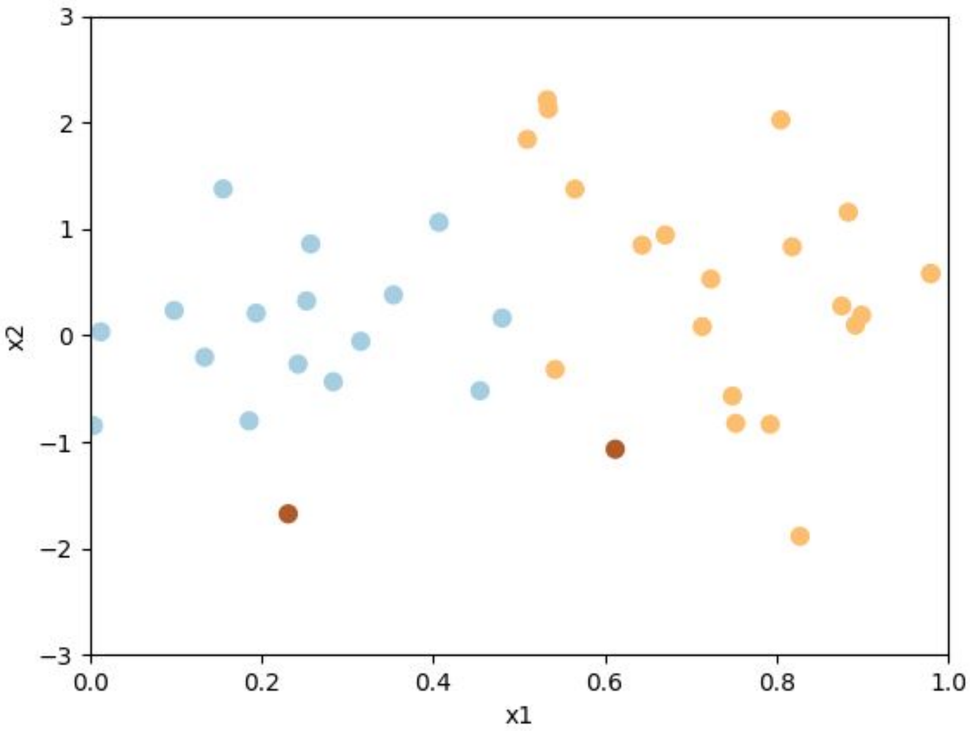
Astronomical data naturally contains statistical effects such as noise/uncertainty, missing or incomplete information, unbalanced/underrepresented classes, and observational biases. A mathematical treatment is required in order to correctly apply ML algorithms to these data!

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!

Supervised learning: classification

Illustration

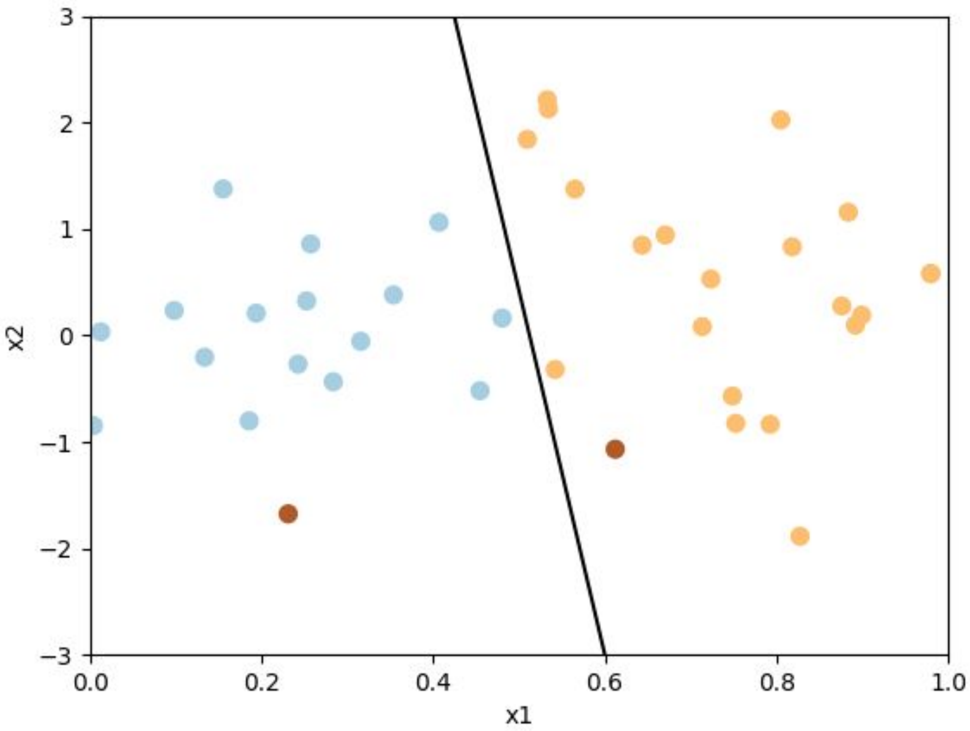


38 data points with two features x1, x2 and one label y (blue or orange).

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

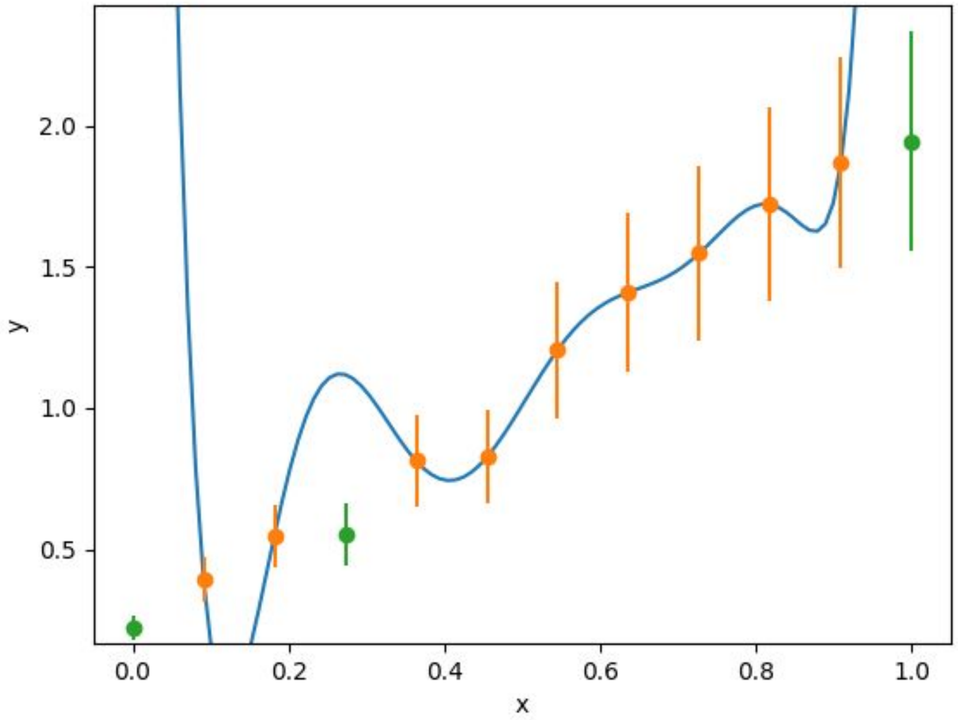


This line is a good **decision boundary** for the given data, as it correctly classifies all blue points as blue.

How to pick this boundary? **Model score** [e.g., MSE]. Depends on **model complexity** (**bias-variance tradeoff**).

We can now predict labels for the two brown points!

Aside: overfitting

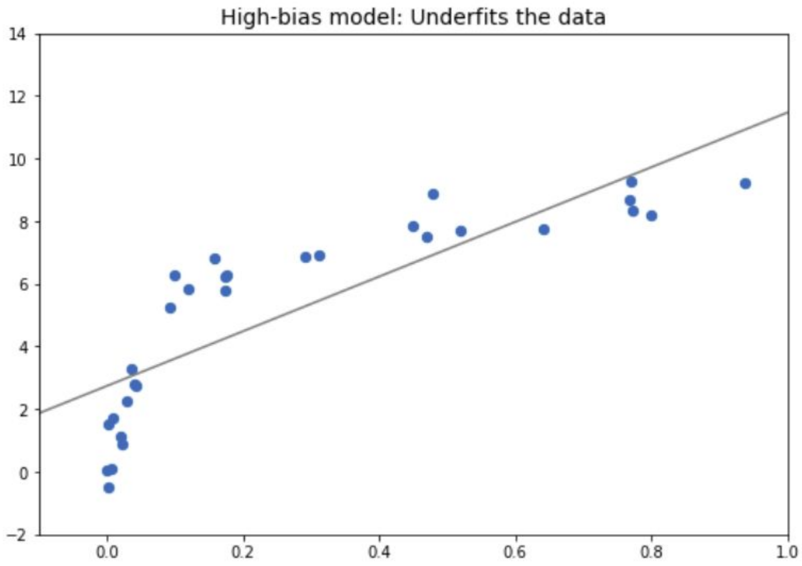


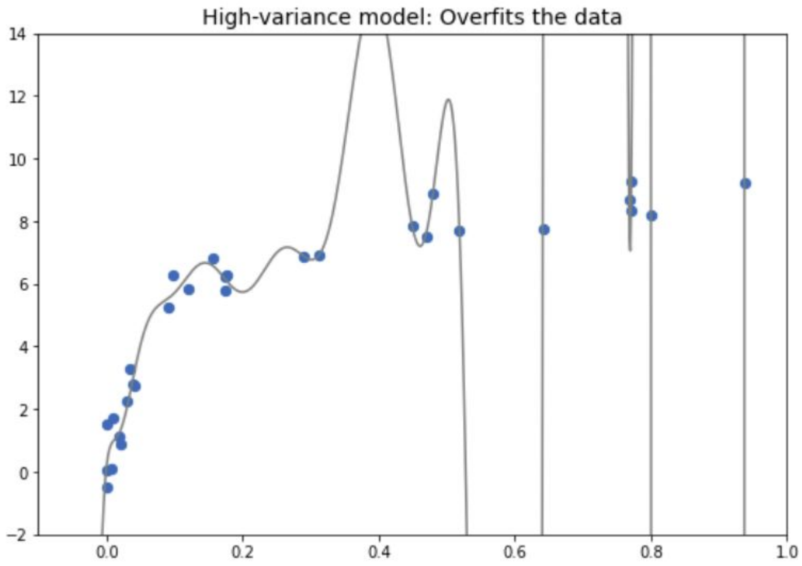
Why not make our model more complex? We could choose a curve such that the boundary is much more stringent!

The model is then trained to exactly fit the current dataset, but its out-of-sample predictive ability will decrease!

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

Aside: bias-variance tradeoff





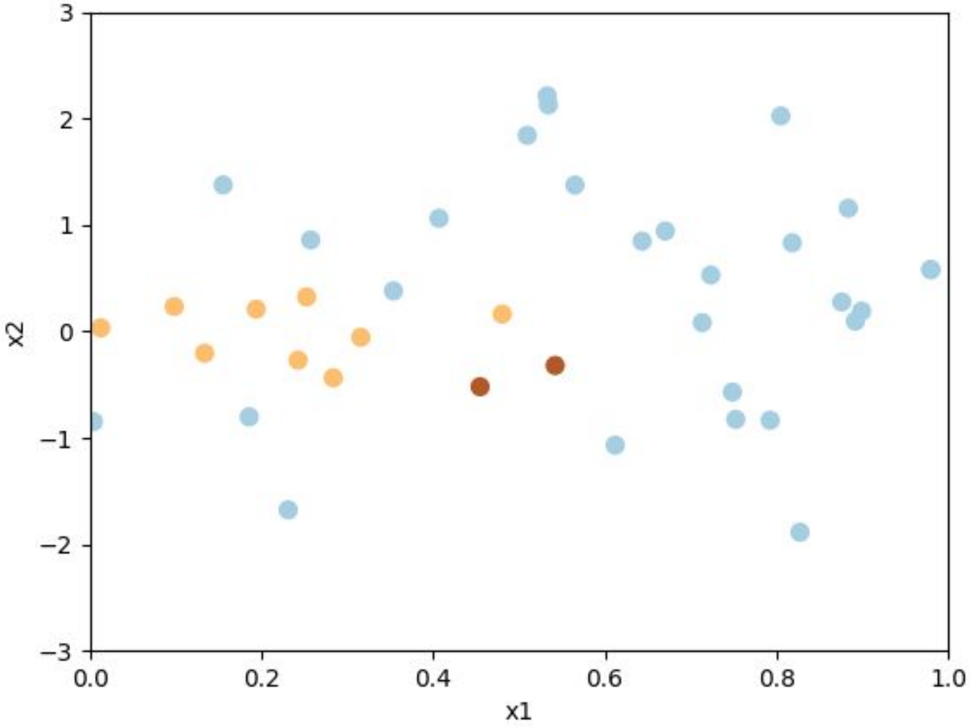
Bias: how well you can reproduce the training sample.

Less complexity = more bias.

Variance: how much the model prediction changes for the **validation sample** (any data **with labels** that **wasn't part of the training sample**).

More complexity = more variance.

Illustration



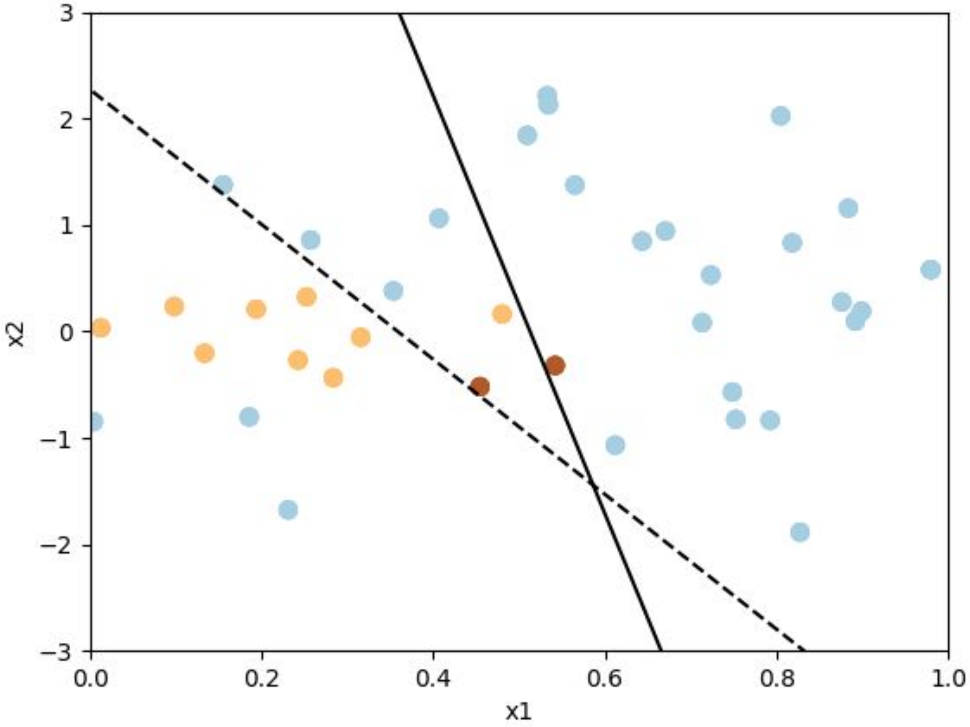
More complicated case:

– Unbalanced sample (way more blue points than orange ones)

– Straight line can’t perfectly separate the labels.

Prediction for the two brown points also affected.

Illustration

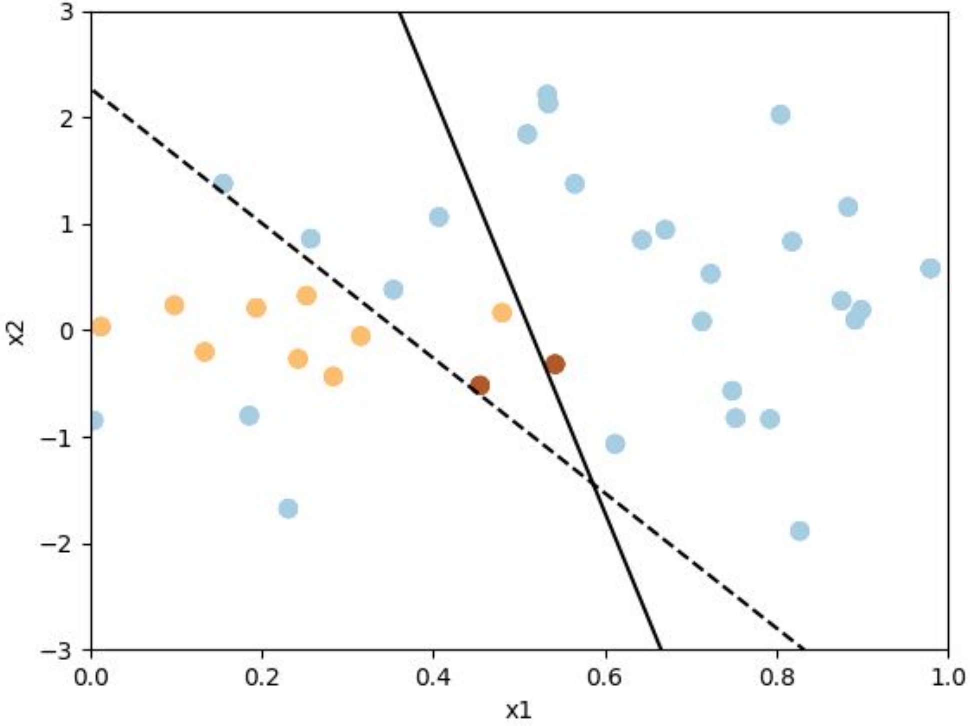


Solid line correctly identifies all orange points (**true positives**), but also incorrectly identifies six blue points as orange (**false positives**).

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?

Illustration



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!

On to the Python notebooks!

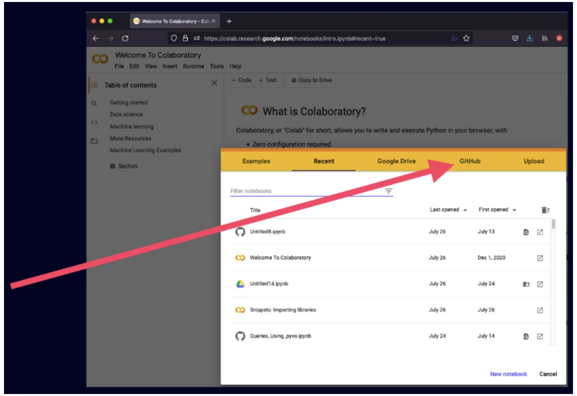
The Python notebooks are based on those by Jake Vanderplas from his book, "Python Data Science Handbook".
[text: CC-BY-NC-ND, code: MIT]

Starting a Google Colab session

1. Sign into your Google account.

2. Navigate to https://colab.research.google.com

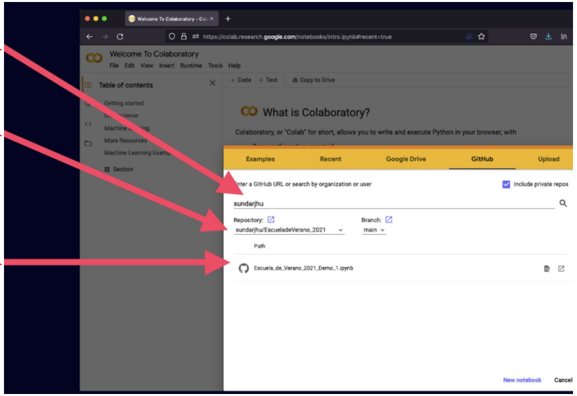
3. In the menu that pops up, select the “**`Github`**” tab



4. Search for username “**`sundarjhu`**” and hit ENTER

5. Select the “**`sundarjhu/EscueladeVerano_2023`**” repository from the dropdown menu

6. Select the Python notebook in that repository



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

Assignment/assessment

Check out the Github repository on Thursday morning!

https://github.com/sundarjhu/EscueladeVerano_2023/tree/main/assignments