Machine Learning Overview

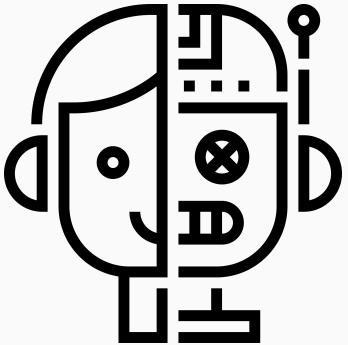
Valentina Staneva

Senior Data Scientist

eScience Institute, University of Washington

vms16@uw.edu

10/23/23



Learning Objectives

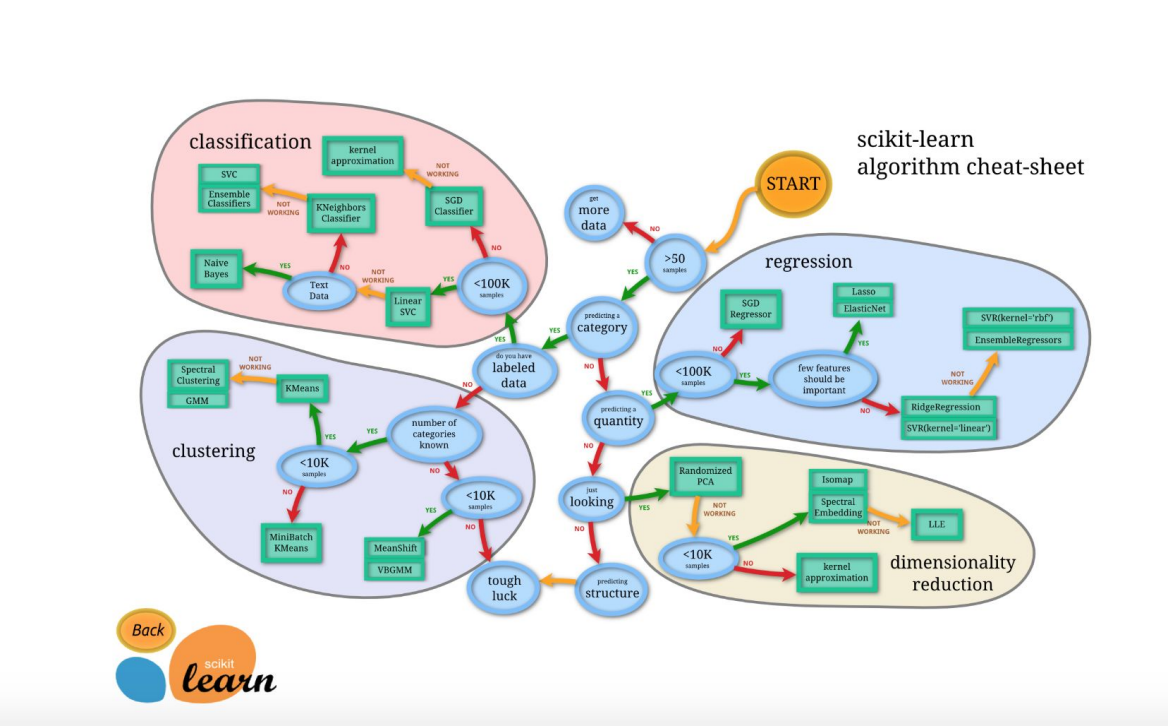
- Identify and formulate major types of ML problems from scientific questions
- List major differences between types of methods, and steps to proceed to evaluate their performance
- Recognize contexts where deep learning can be useful
- Identify popular computer vision/time series tasks and popular deep learning frameworks for them
- Able to organize labeled datasets in format for those frameworks
- Outline elements of ML pipelines

What I will not talk about

- Details of applying specific methods (follow up tutorials with Jupyter notebooks)
- Advanced uses of ML, such as robot perception, reinforcement learning, generative adversarial networks
- Unsupervised learning
- Computational aspects
- Domain specifics: you will do this through your projects!

Machine Learning Algorithms

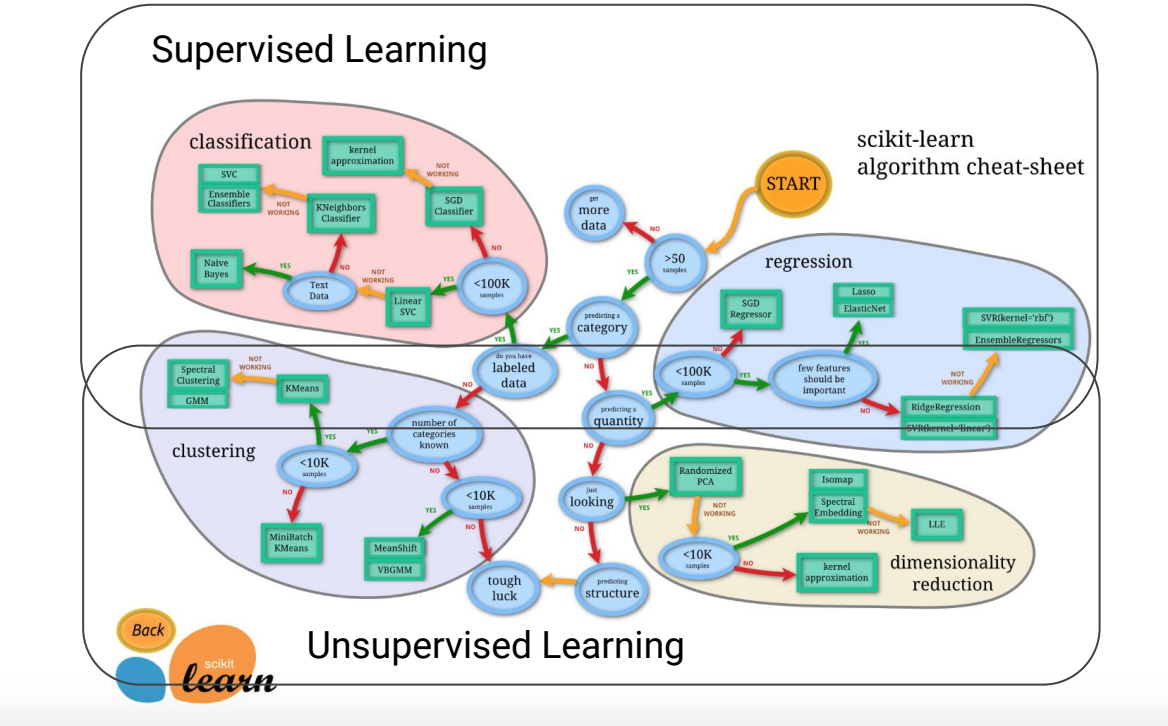
Choosing a Machine Learning Algorithm in 2013:



by Andreas Mueller,
scikit-learn developer

Machine Learning Algorithms

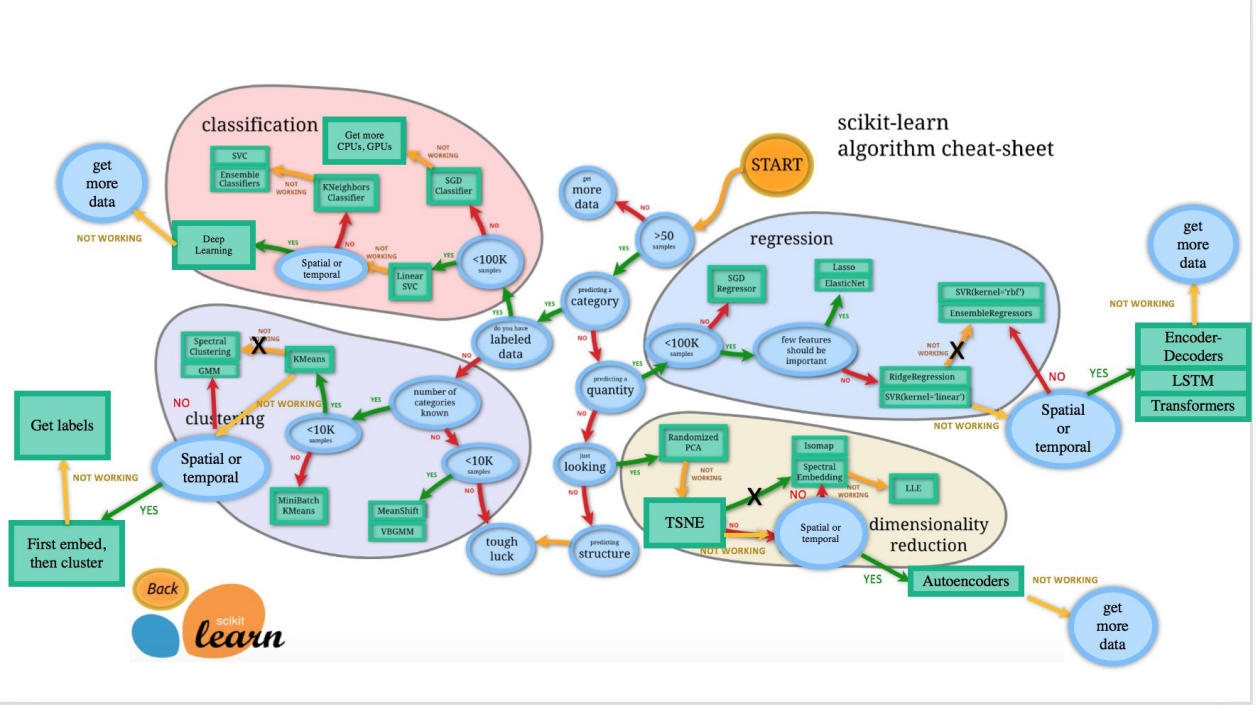
Choosing a Machine Learning Algorithm in 2013:



by Andreas Mueller,
scikit-learn developer

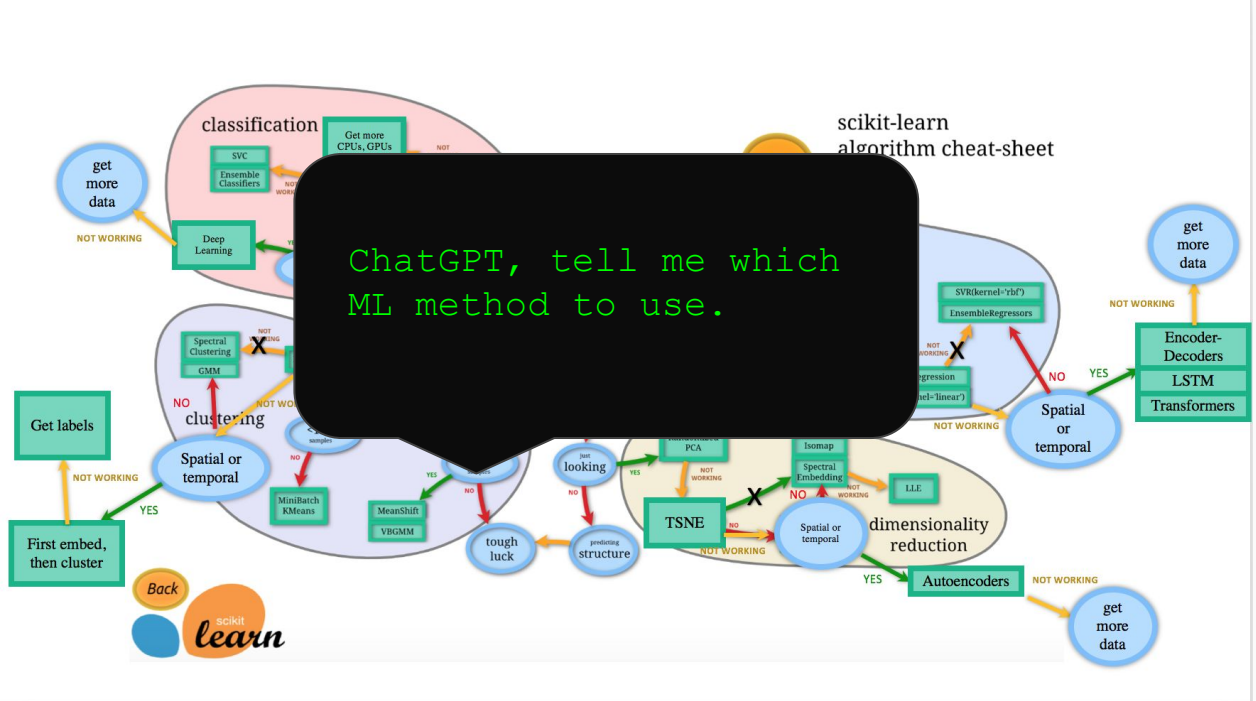
Machine Learning Algorithms

Deep Learning Advances as of 2020:



Machine Learning Algorithms

AI in 2023:



Algorithms change often, but the fundamental types of problems are still prevalent today, and the algorithms should be applied with care.

ML in Your Work

- What kind of problems in your work/field you think you can solve through ML?
- What challenges do you encounter/envision in applying ML in your work/field?

SharedGoogleDoc

(open in incognito/private mode if you want your answer to be anonymous)

Machine Learning Problems

Regression:

- Find relationship between a set of variables and outcomes
- Predict outcomes on new variables

Classification:

- Based on a set of observations with attached categories, learn how to predict categories of new observations which do not have labels

Clustering:

- Find groups of objects which are similar to each other

Dimensionality Reduction:

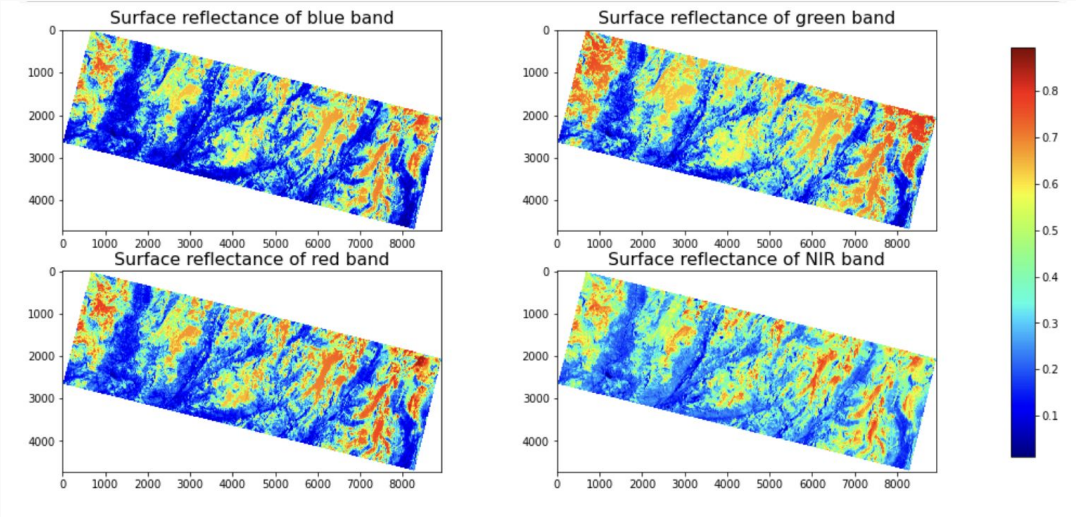
- Convert data into a new format which represents the data in a more compact format without much loss of information

In practice category boundaries are not sharp and several methods can be used simultaneously!

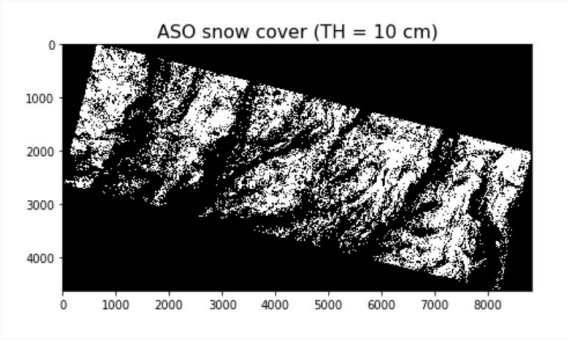
Classification Example: Snow Cover Prediction

Task: predict snow coverage **(T/F)** from satellite imagery

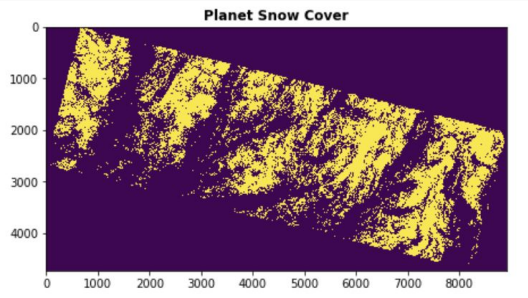
Planet Data:



Snow Cover Label:



Predicted Snow Cover:

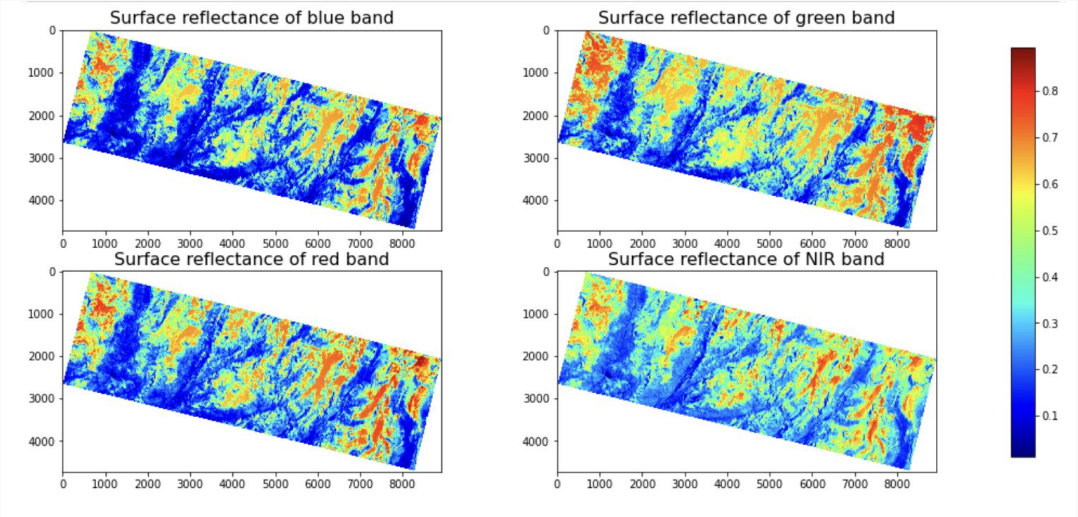


Snow Cover Mapping by Kehan Yang, Aji John, Nicoleta Cristea
https://geo-smart.github.io/scm_geosmart_use_case/chapters/one.html

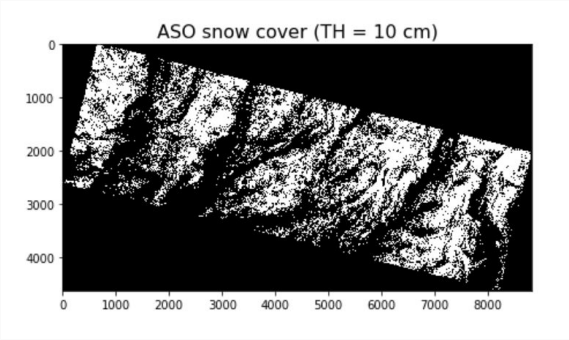
Regression Example: Snow Cover Prediction

Task: predict snow **depth** from satellite imagery

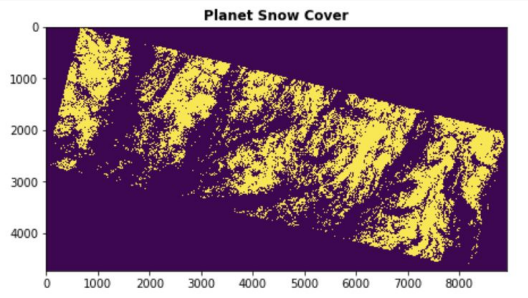
Planet Data:



Snow Cover Label:



Predicted Snow Cover:



Snow Cover Mapping by Kehan Yang, Aji John, Nicoleta Cristea
https://geo-smart.github.io/scm_geosmart_use_case/chapters/one.html

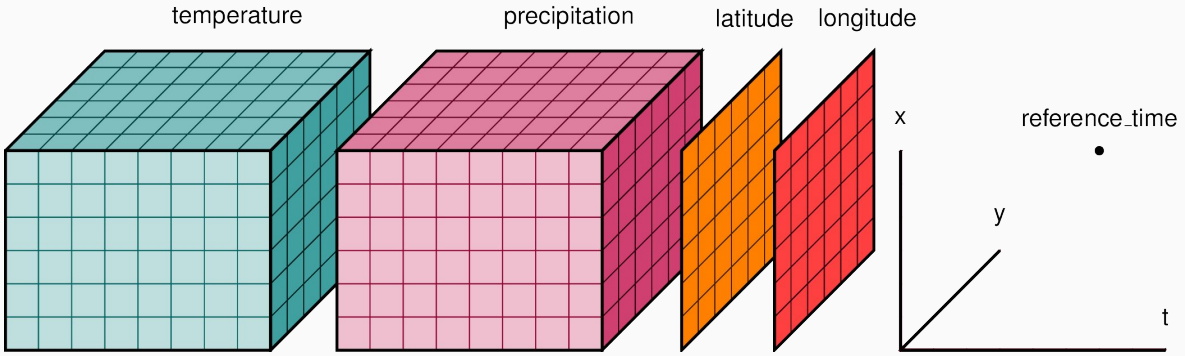
Data Challenges

Data in ML tutorial:
independent observations, uncorrelated features.

DATASET					
	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
Observation 1	1.29	Yes	B	0	6.6
Observation 2	3.56	No	C	1	7.2
Observation 3	7.89	Yes	B	0	3.4
Observation 4	0.53	Yes	A	1	5.5
Observation 5	6.44	No	A	1	8.1
.	.	.	.	.	.
.	.	.	.	.	.

Scikit-learn format:
`(n_samples, n_features)`

Data in practice:
often correlated in time, space, measurement protocols, etc ….



Data in Earth Sciences

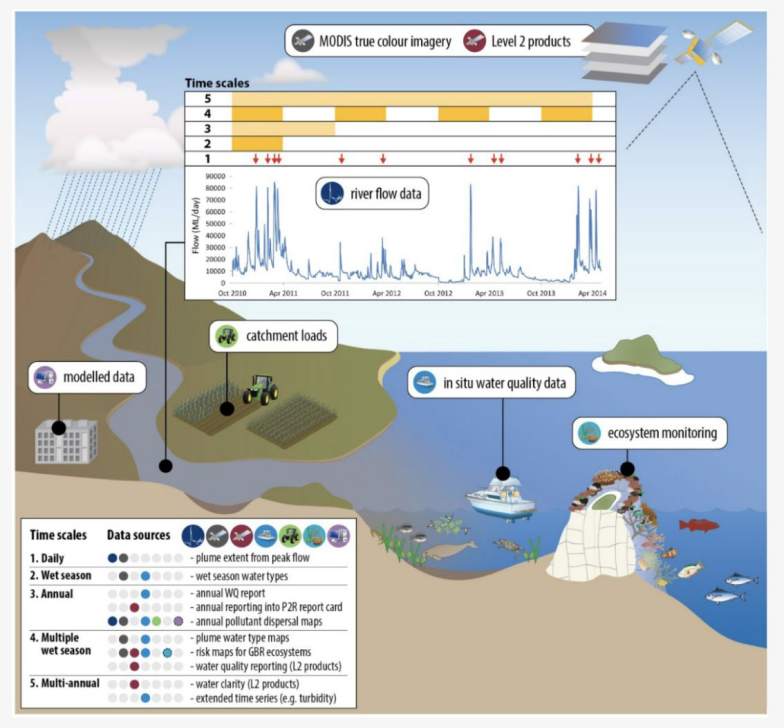
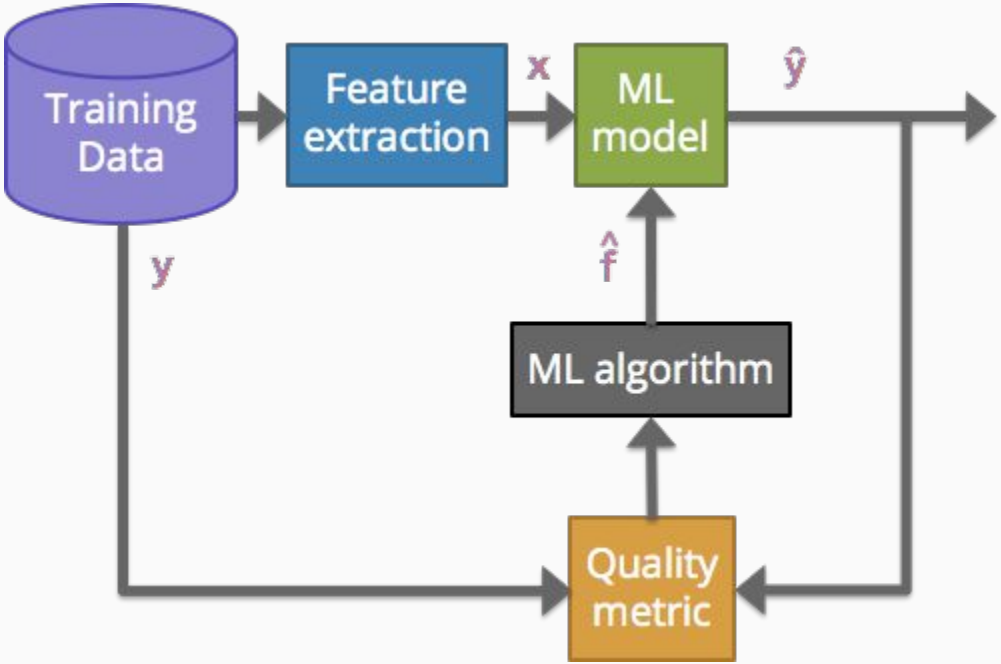


Image Source: Devlin M., et. al. *"Water Quality and River Plume Monitoring in the Great Barrier Reef: An Overview of Methods Based on Ocean Colour Satellite Data"*

Data Comes in all Shapes, Sizes and Scales!

Supervised Learning Pipeline



Machine Learning Debt

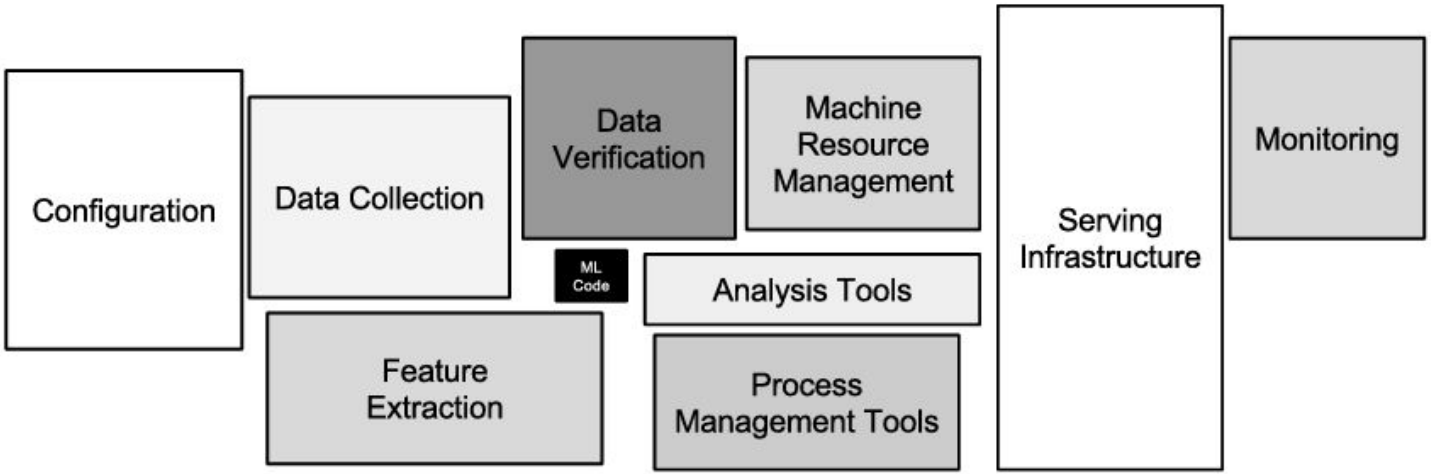
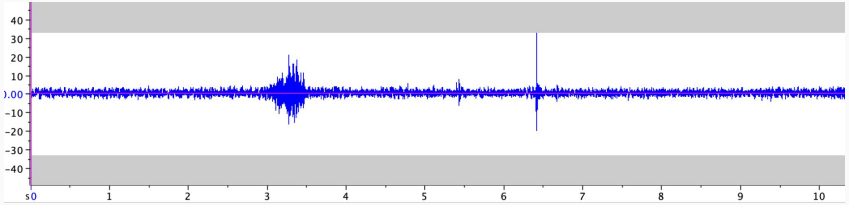


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Sculley et. al., Hidden Technical Debt in Machine Learning Systems

Feature Engineering: Icequake Sound

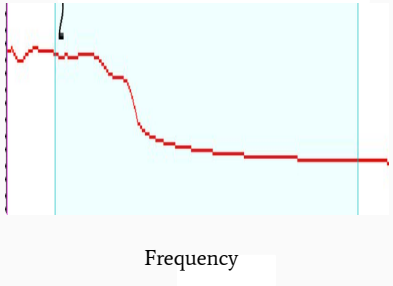
1D Icequake (Bloop) Sound:



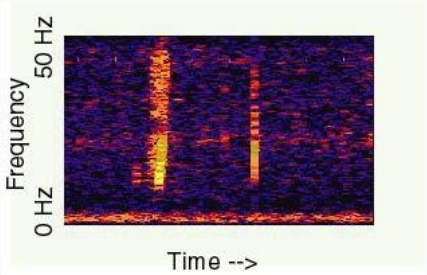
Basic Features:

Group	Low-level descriptor
Basic	AudioWaveform AudioPower
BasicSpectral	AudioSpectrumEnvelope AudioSpectrumCentroid AudioSpectrumSpread AudioSpectrumFlatness
SpectralBasis	AudioSpectrumBasis AudioSpectrumProjection
SignalParameters	AudioHarmonicity AudioFundamentalFrequency
TimbralTemporal	LogAttackTime TemporalCentroid
TimbralSpectral	SpectralCentroid HarmonicSpectralCentroid HarmonicSpectralDeviation HarmonicSpectralSpread HarmonicSpectralVariation

1D Power Spectrum



2D Spectrogram



Use simple methods:

- Logistic Regression
- Decision Tree

Treat as 1D vector, use:

- SVM/Random Forest
- 1D CNN

Treat as 2D array, use:

- Apply first PCA, then SVM/Random Forest
- 2D CNN

Sound Source: https://www.pmel.noaa.gov/acoustics/sounds_cryogenic.html

Feature Engineering: Image Example

2D satellite images:
snow, clouds, clear land?

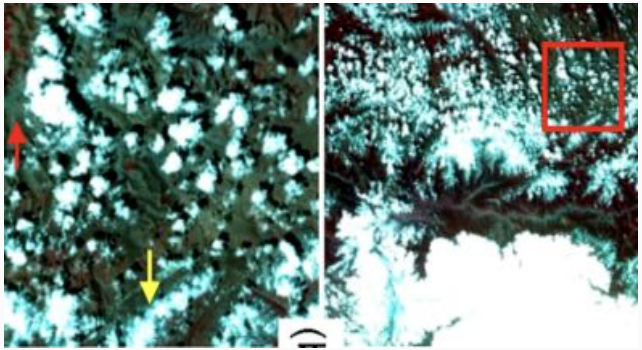
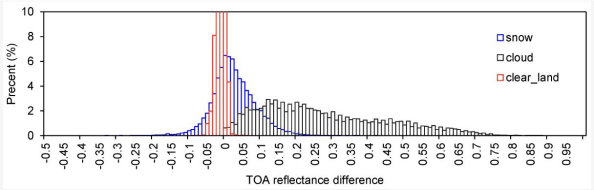


Image Source: Bian'16



Color Histograms
1D vectors

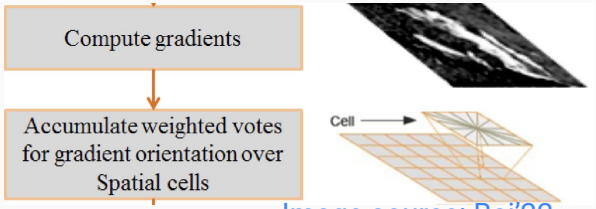


Image source: Bai'22

Histograms of Oriented Gradients (HOGs)
1D vector for each color

snow - persistent in winter
cloud - not as persistent

Principal Component Analysis
a set of templates & weights

Loss Functions

Loss between actual and predicted value: $L(y_i, \hat{y}_i) = L(y_i, \hat{f}(x_i))$

Regression Loss

Mean Squared Error

$$\mathcal{L}_{\mathcal{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2$$

Classification Loss

Cross-Entropy Loss

$$\mathcal{L}_{\mathcal{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}.log(f(x_{ij}))$$

$$N : samples; M : classes$$

Minimize loss == “learning” parameters of estimators

Loss Optimization

Gradient Descent:

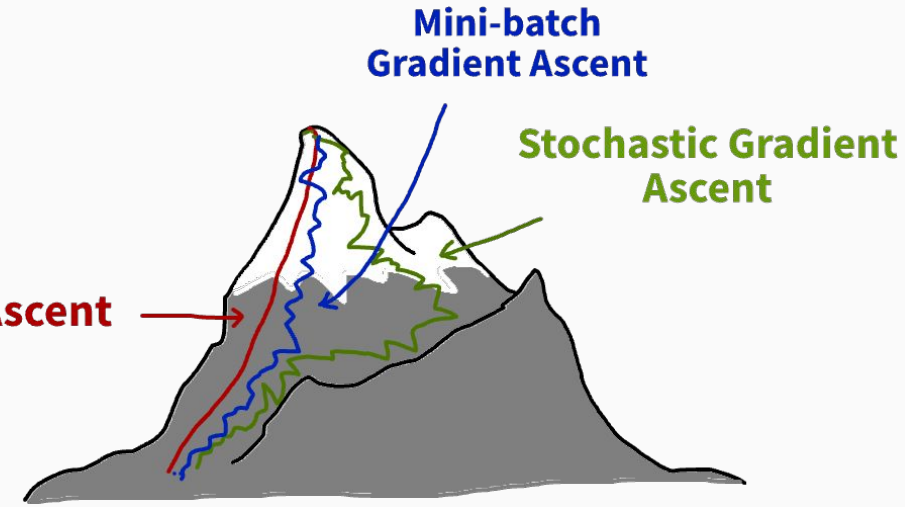
- update *f* in the direction of the gladient
- calculate gradient using **all observations**
- each step decreases the loss
- calculation of each step could be expensive

Stochastic Gradient Descent:

- calculate gradient using **one observation** at a time
- over time the algorithm will see all the data, and can minimize the loss (some guarantees)
- not every step decreases the loss, the path is noisy and long, but could avoid local minima

Mini-batch Gradient Descent:

- evaluate the gradient at a **random subset of the data** (mini-batch)
- a good compromise!



$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta L_\theta(\{x_i, y_i\})$$

Classification Algorithms

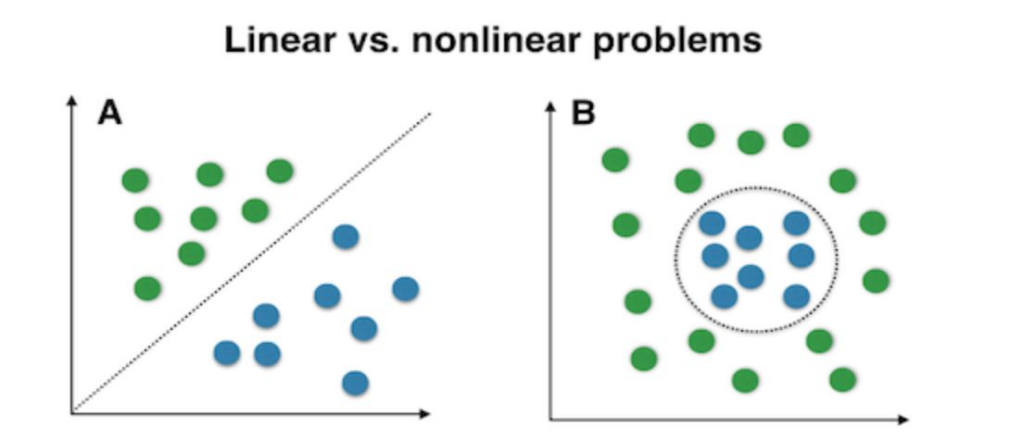


Image Source

Linear Classifiers:

- Logistic Regression
  - interpretable features
- Support Vector Machines
  - data coming from same sensors
  - can be slow
- Naive Bayes
  - counts, scalable

Nonlinear Classifiers:

- Decision Trees
  - interpretable features
- Random Forests
  - all kinds of inputs
  - combines the power of many trees
- Gradient boosting
  - all kinds of inputs, high accuracy

Establish a baseline, compare performance!

Feature Engineering & Model Complexity

Machine Learning in the past:

- create a few highly crafted features
- apply simple models (e.g. linear regression, decision tree)

Machine Learning today:

- keep data as raw as possible: let the algorithm figure out the features
- use complex model to discover nonlinear relationships in the data

No free lunch: Bias-Variance trade-off!

Prediction Error = Variance + Bias^2 + Irreducible error

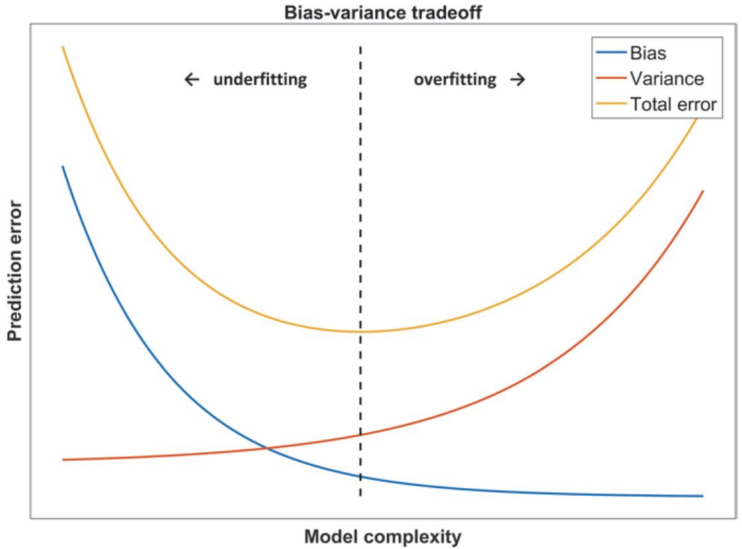


Fig. 8.3 The bias-variance tradeoff. With increased model complexity the model can more accurately match the underlying relation at the risk of increasing the variance (amount of overfitting). The bias-variance tradeoff corresponds to minimizing the total prediction error (which is the sum of bias and variance)

Image Source

Evaluation: metrics

Translating ML problems back to science problem:

Our algorithm achieved accuracy of 99%.

On which set?

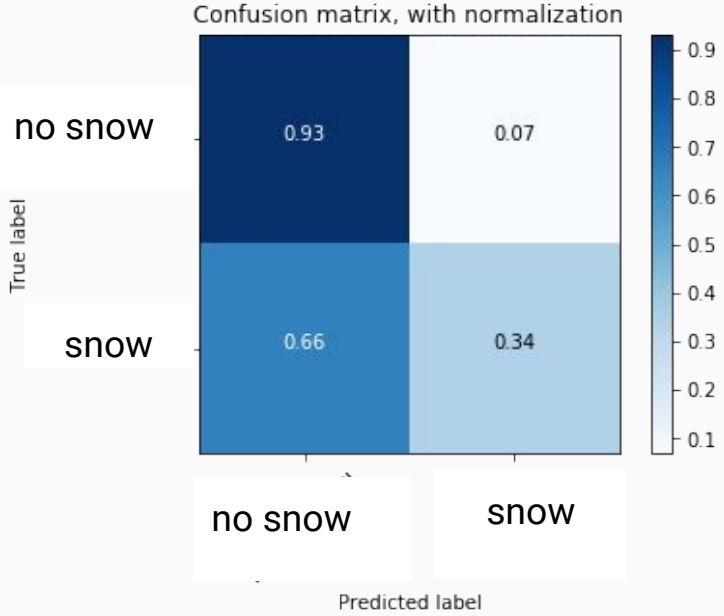
Did you use that set for parameter selection?

What accuracy will you achieve if you only predict No/Yes? What if you randomly guess?

Can you tune your detection threshold to select only detections with very high confidence?

Can you ensure you retrieve most objects, even if some detections are wrong (you can remove the wrong ones manually after that)?

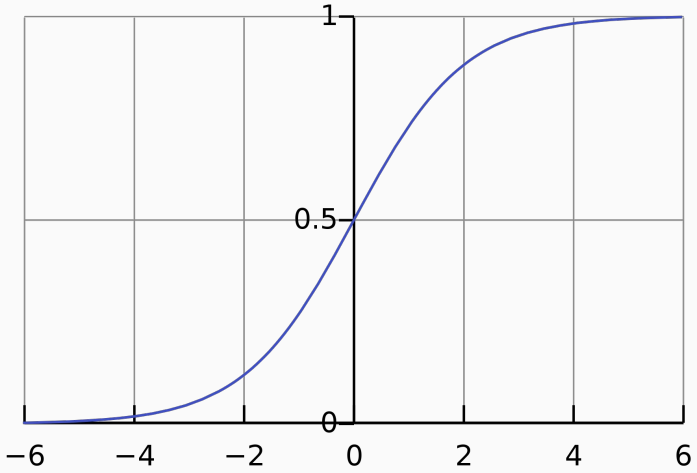
Will it work on a new site, different ship, during a difference season, etc.?



Accuracy: #correctly predicted snow/#all pixels
Precision: # correctly predicted snow/all snow pixels
Recall: # correctly predicted snow #predicted snow pixels
Reduce threshold!

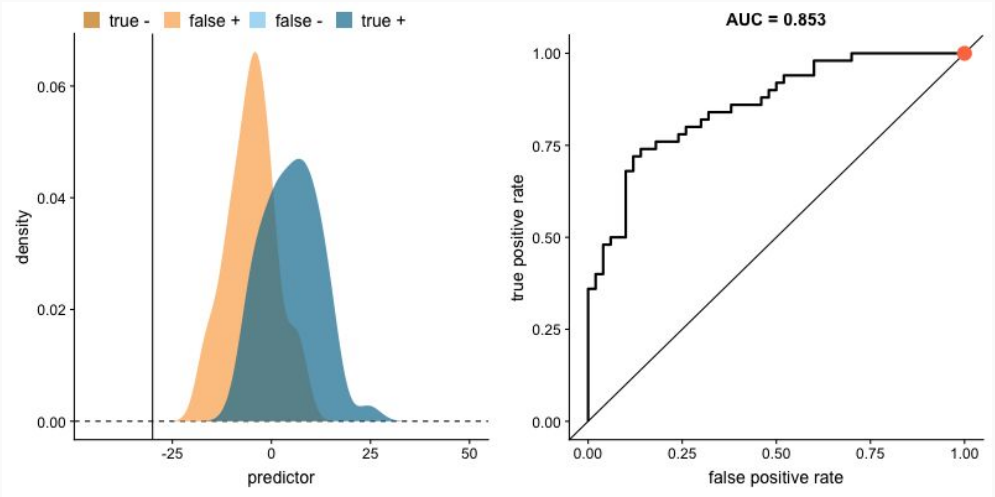
From Scores to Predictions

Many classification algorithms output scores which are converted to predictions through the **Sigmoid Function**.



[-Inf, +Inf] -> [0, 1]

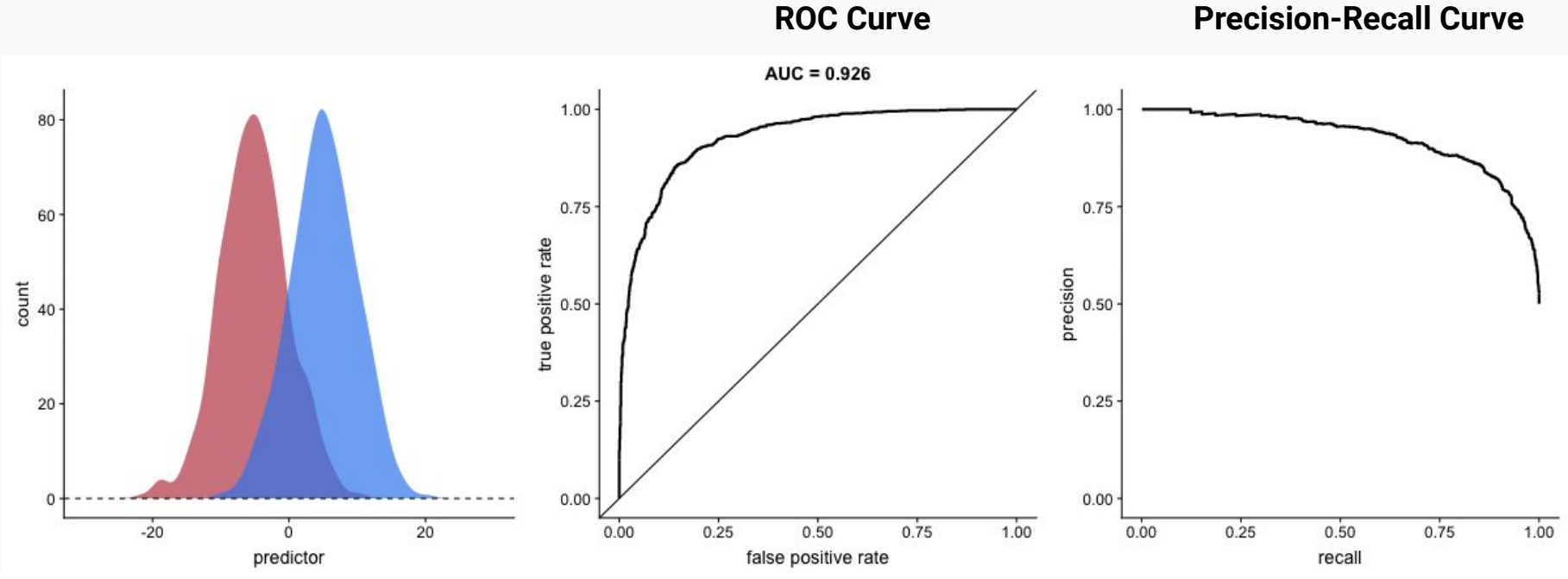
ROC Curve



https://github.com/dariyasydykova/open_projects/

From Scores to Predictions

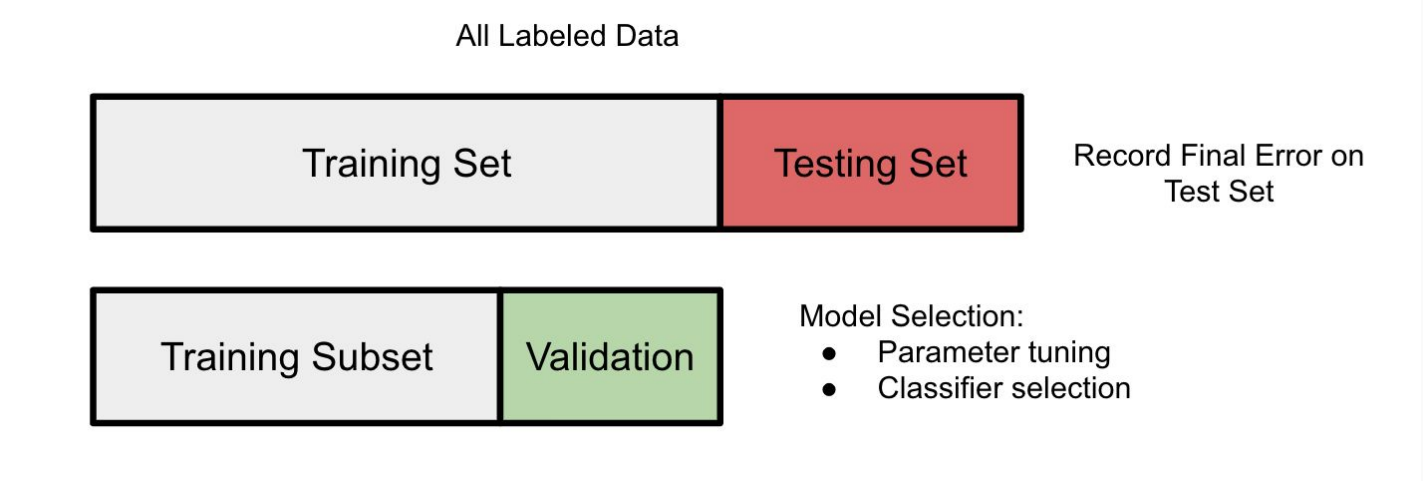
AUC (Area Under the Curve) is not a good measure for unbalanced datasets.



https://github.com/dariyasydykova/open_projects/

Evaluation Sets

- Put away a dataset for testing
  - ➢ don’t use it for model selection/parameter tuning
  - ➢ use the testing set once!
  - ➢ use validation set many times!

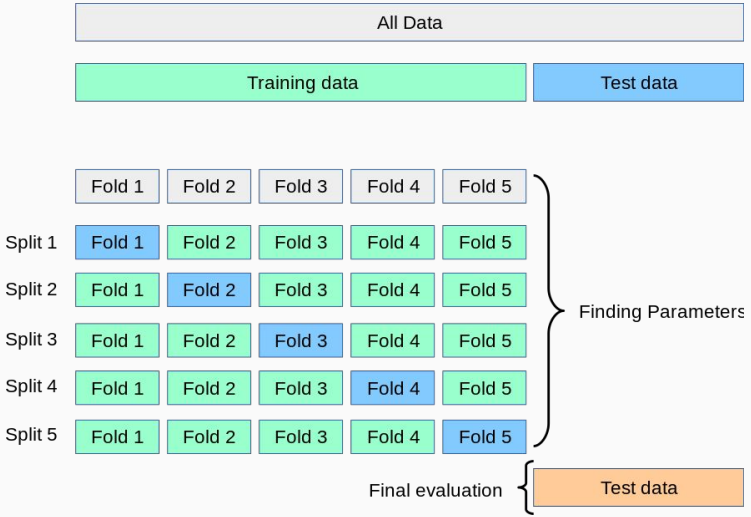


Cross-validation

Using one validation set can make our results sensitive to it, instead create multiple validation sets

- K-fold cross-validation
- Leave-one-out cross-validation

- Average the results
- Can estimate variance!

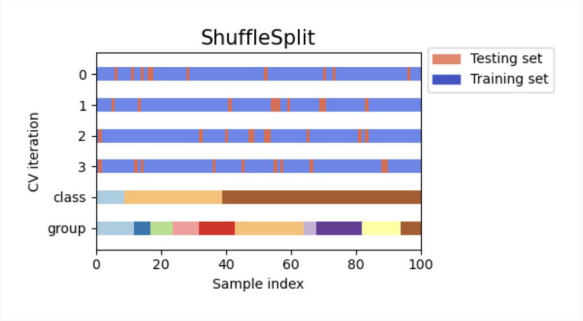


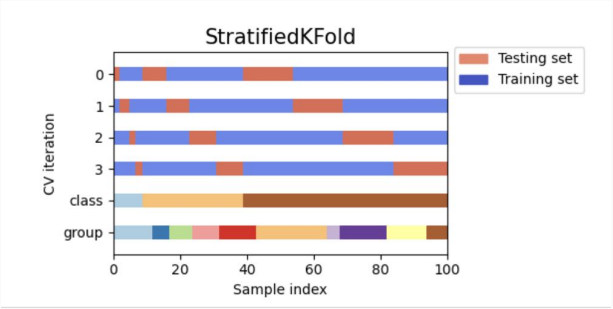
https://scikit-learn.org/stable/modules/cross_validation.html

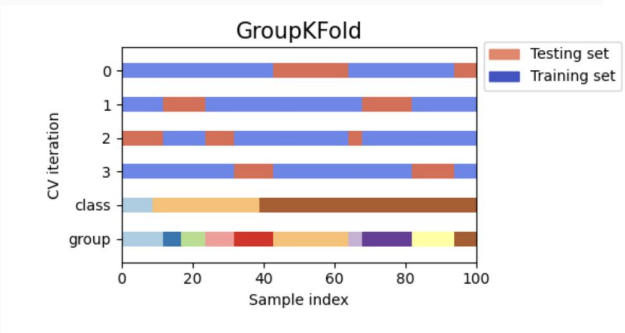
Cross-validation Sampling Strategies

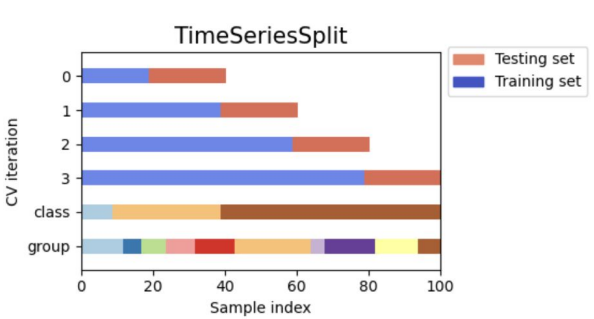
Sampling strategies:

- stratified samples
- balanced samples
- group cross-validation
- time-series split





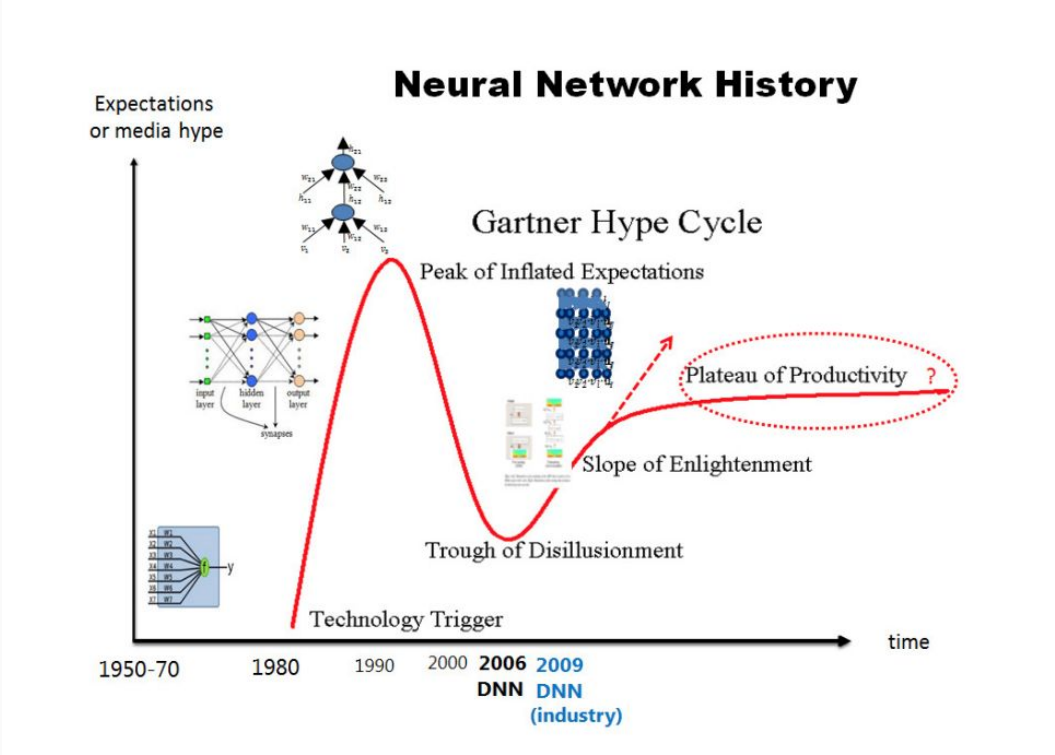




[Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure](), by Roberts et.al.

What about Deep Learning?

Deep learning is the methodology of using deep neural networks to solve machine learning problems.



[Image Source]

What about Deep Learning?

Last 10 Years:

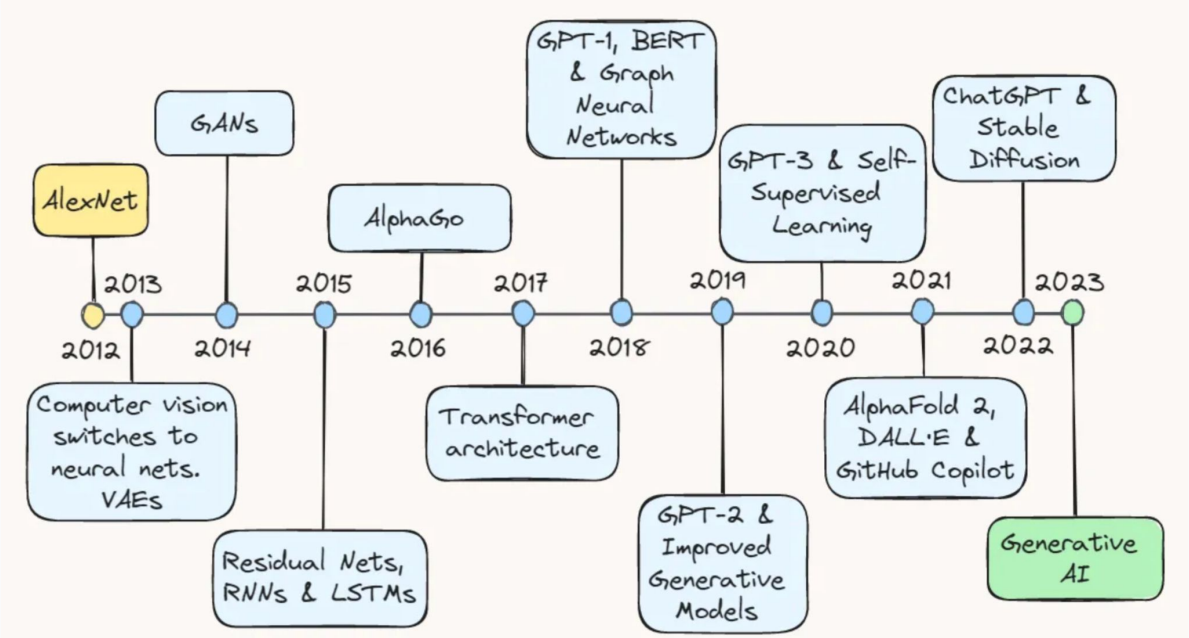
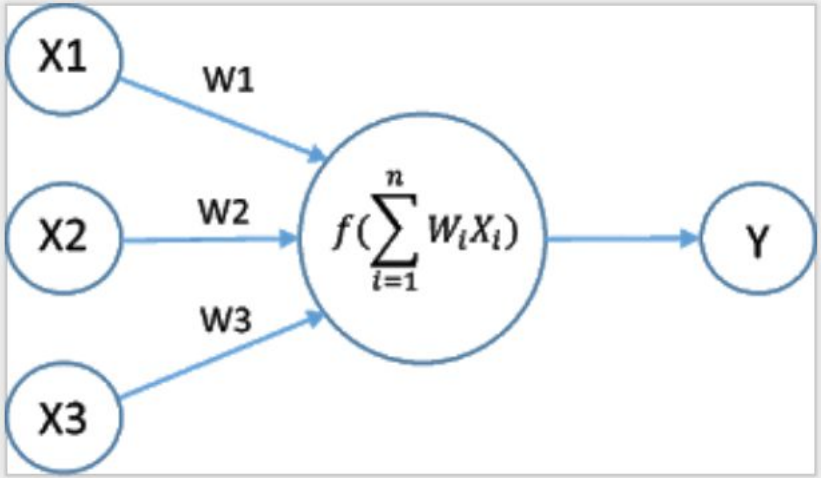


Image Source: Thomas Dorfer's Blog

Shallow Learning vs Deep Learning

Logistic Regression as a Neural Network:



Fully Connected Neural Network as nested regressions:

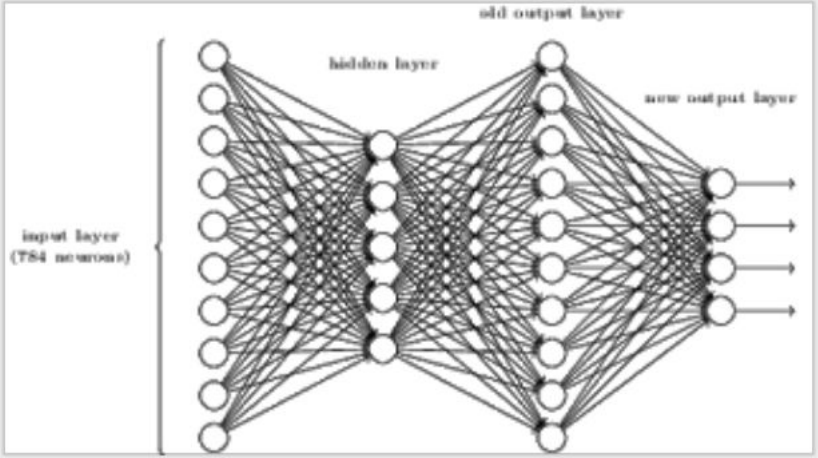


Image Source: http://samuelhermanblog.blogspot.com/2017/01/deep-learning-part-1-logistic.html

parameters per layer = # neurons in layer 0 x # neurons in layer 1 + # neurons in layer 1

Fully Connected Networks are not very practical!

Weight Sharing

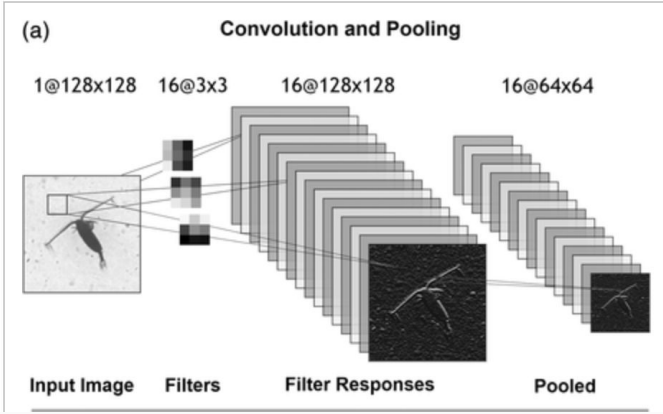
Weight sharing:

- reduce the number of parameters
- explore correlation structure

Spatial Correlation Structure

Convolutional Neural Networks (CNN):

- local patterns (1D, 2D, 3D,...)
- images, spatial/grid data



Temporal Correlation Structure

Recurrent NN, LSTM, Transformers:

- long and short range relations
- time series

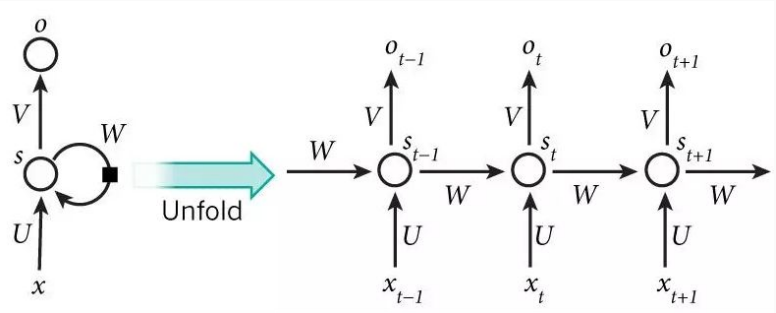


Image Source: Wikipedia

Convolutional Neural Networks

CNNs are capable of learning a hierarchy of features:

from **local** (such as edges) in the first layers to **global** in the deeper layers (such as petals, leaves)

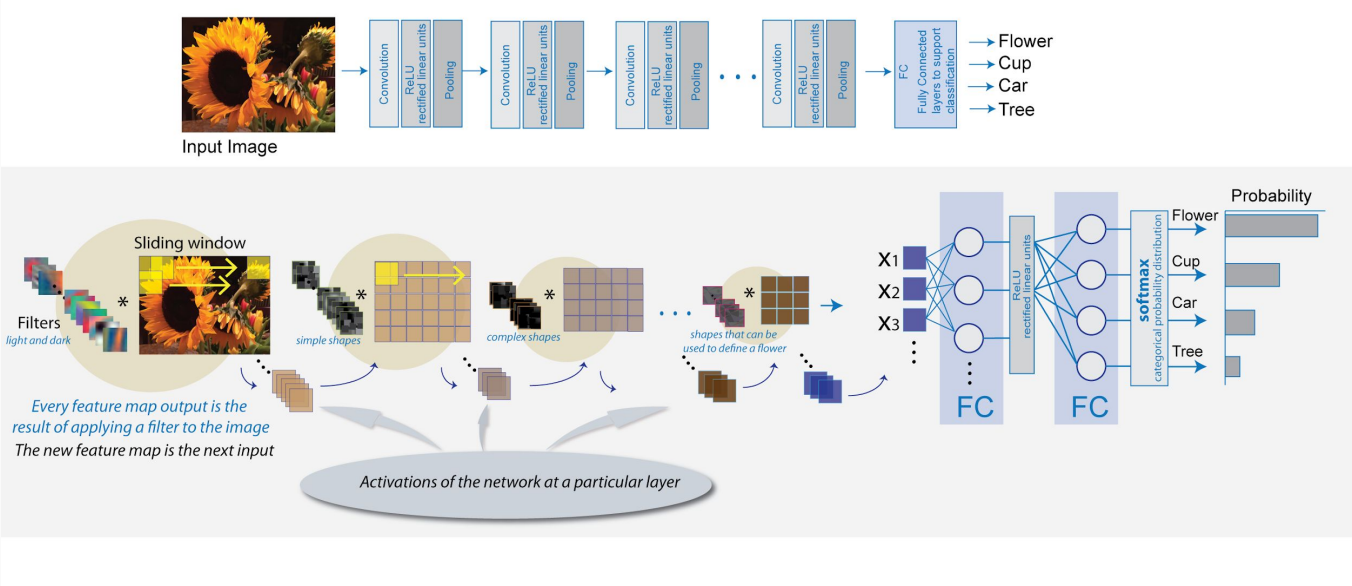


Image Source: [Mathworks]

Transfer Learning

Initial success: **big training sets** and **fast GPUs**.

Subsequent success: a model trained on one dataset can be useful for solving another ML problem with a small training dataset -> learnt features are **transferable**!

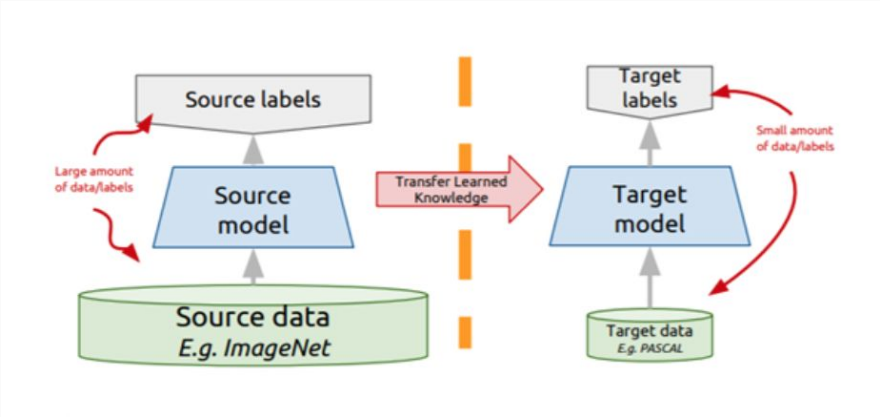


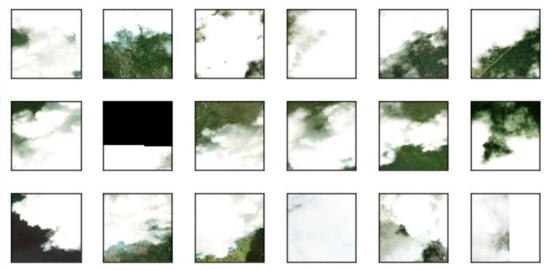
Image Source

Example Notebook: https://www.tensorflow.org/tutorials/images/transfer_learning

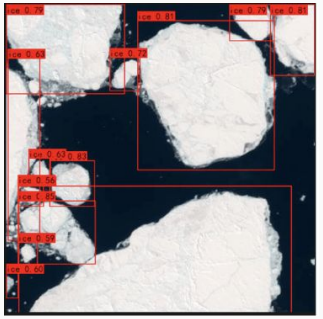
Repositories with trained models: https://www.tensorflow.org/hub, https://huggingface.co/

Computer Vision Tasks

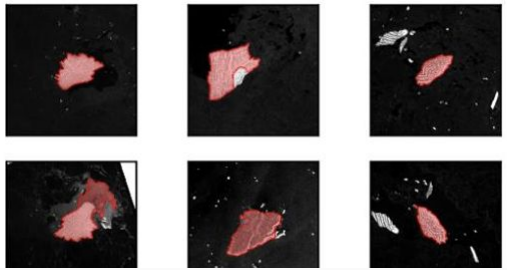
Image Classification
predicting the category of an entire image



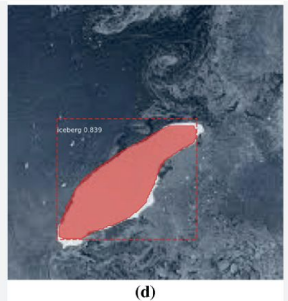
Object Detection
localization and identification of objects in an image



Semantic Segmentation
predicting the category of every pixel



Instance Segmentation
detection and delineation of objects in an image

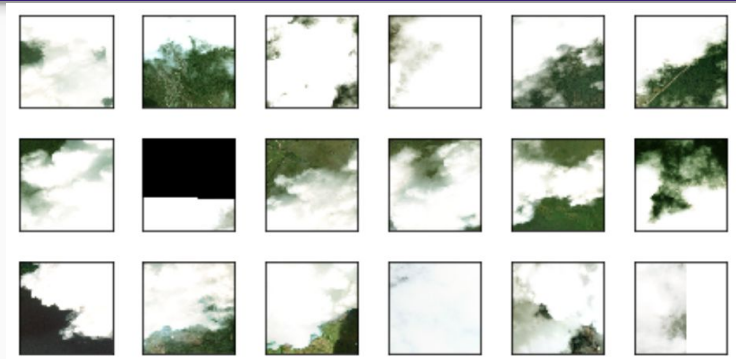


Terminology

Do we speak the same language?

- prediction
- classification
- detection
- forecasting
- simulation
- state estimation
- segmentation
- pixel classification
- object detection, region detection
- data assimilation
- filtering
- interpolation
- extrapolation
- smoothing
- kriging
- hindcasting, nowcasting
- stochastic smoothing
- regression
- function approximation
- learning
- parameter estimation
- downscaling/upscaling
- downsampling/upsampling
- online vs offline learning
- data augmentation
- synthetic data generation
- inference
- localization

Image Classification



How can we assign a label to the image? (ex. cloud/no cloud)

Approach 1:

- Extract features & apply scikit-learn classifiers

Approach 2:

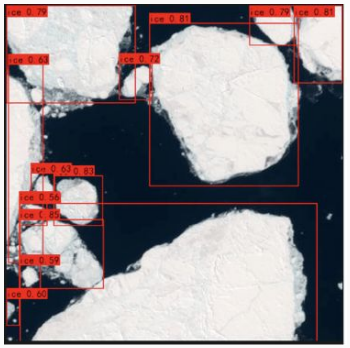
- Use Convolutional Neural Networks on the full images
  - Easy implementation in Keras
  - Pretrained models:
    - VGG, ResNet, MobileNet

Evaluation:

- Standard classification metrics apply:
  - Accuracy, precision, recall, F1 score

Object Detection

How can we identify more than one object in an image?



Source: [Multi-Scale Polar Object Detection Based on Computer Vision]

Approach 1:

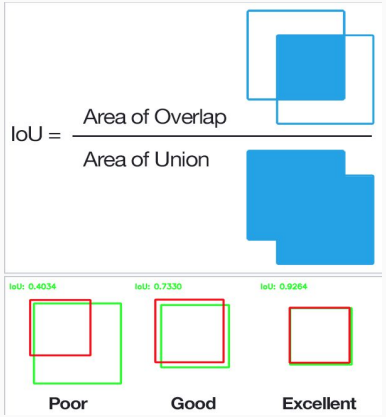
1) propose regions (using a small network)
2) classify only those regions, combine bounding boxes
3) Faster RCNN (Region Conv. Neural Networks)

Approach 2:

1) Split image into grid
2) Detect boxes in each grid and weight predictions
   Yolov8 (You Only Look Once)

https://keras.io/examples/vision/yolov8/

Evaluation: Intersection over Union, mean Average Precision (mAP)



Semantic Segmentation

How can we segment regions in the image?

x_i - image, y_i - mask (objects of same type labeled the same)

Approach 1:

- Pixel by pixel classification (assume independence)

Approach 2:

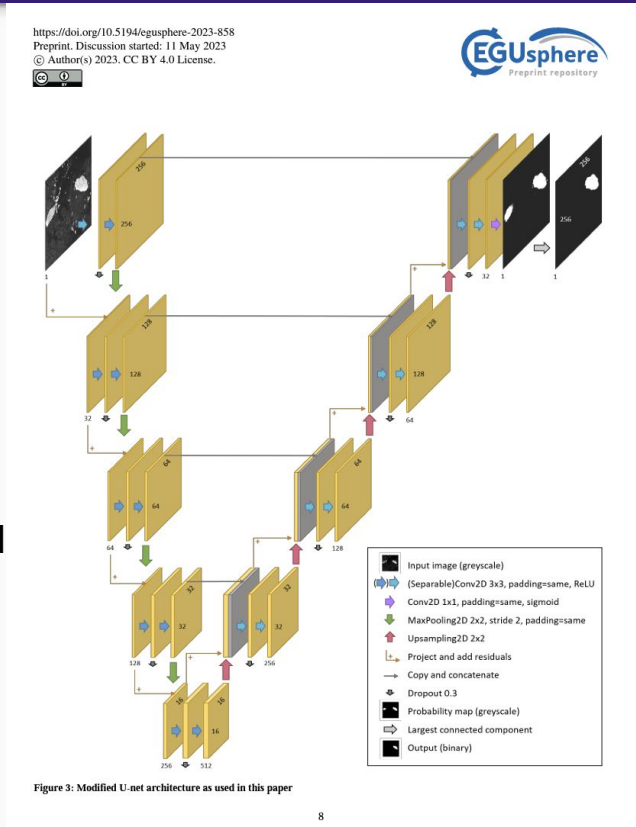
- Split into small windows and classify each window
  - correlation within a window

Approach 3:

- U-net (developed for biomedical imaging, works with little training d
- Easy Implementation in keras

Evaluation:

- Evaluation based on standard metrics (like pixel accuracy)
- Geometric: intersection/total area, boundary F1 scores

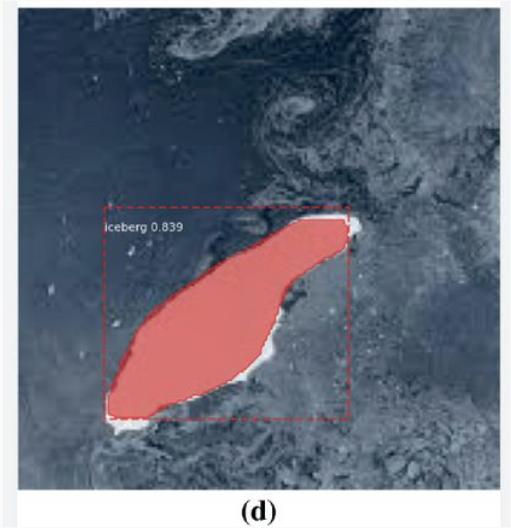


Source: https://egusphere.copernicus.org/preprints/2023/egusphere-2023-858/

Instance Segmentation

How can we detect objects and segment their boundaries?

x_i - image, y_i - instance id, mask (each object labeled differently)



Source: Ice Berg Detection in SAR Images Using Mask R-CNN

Approach 1:

- Segment directly

Approach 2:

- Detect bounding box
- Segment within bounding box

  https://github.com/matterport/Mask_RCNN

Beyond Typical Classification and Regression

Multi-resolution: Generative Adversarial Neural Nets

- Deep learning models for generation of precipitation maps based on numerical weather prediction

Numerical Analysis: I have no data, I simulate models!

- speed up solutions of differential equations
  - fit a deep NN between inputs and outputs of a system
  - slow at training, fast at prediction
- Machine Learning-accelerated computational fluid dynamics
- Can deep learning beat numerical weather prediction?

Physics Informed ML:

- AI for Physics Inspired Hydrology Modeling

Multi-Modal Sensor Fusion:

- Keras models with multiple inputs and outputs
- Deep learning in multimodal remote sensing data fusion (review)

Data Preparation: Satellite Imaging Example

Geospatial Data Annotation:

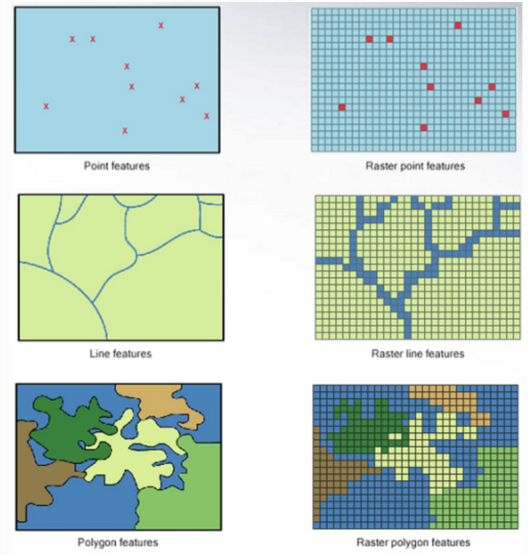


Image Source

Getting data into the right format for Machine Learning:

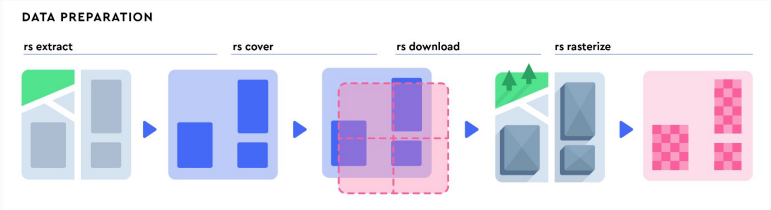
- Chipping images and preserving labels
- Geospatial coordinates ⇔ pixel coordinates
- Mask ⇔ boundary coordinates ⇔ geospatial coordinates
- Overlapping geospatial regions on geotiff images
- Handle missing data (regions, channels)
- Merging chips for mapping

Tools:

- rasterio, geopandas, shapely, rioxarray, regionmask, xbatcher, ...

Special Formats:

- COCO (Common Objects in Context) format
- STAC (Spatio-temporal Access Catalogue) extensions



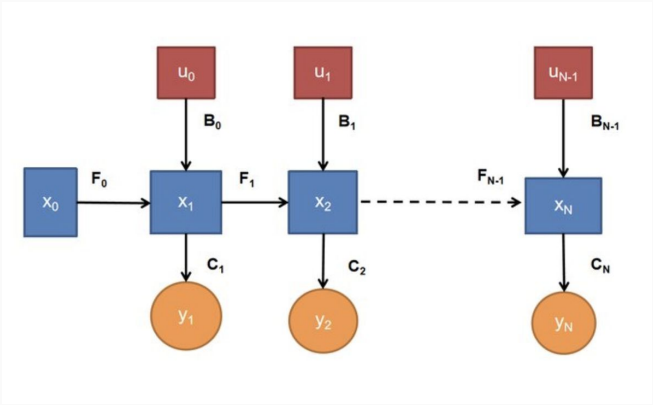
Robosat Pipeline

Time Series Modeling

- **(Extended) Kalman Filter:**

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{w}_k$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k$$



- **ARIMA models**
- **Hidden Markov Models**
- **ODEs/PDEs**

Image Source

- **Neural Networks can have many hidden layers**

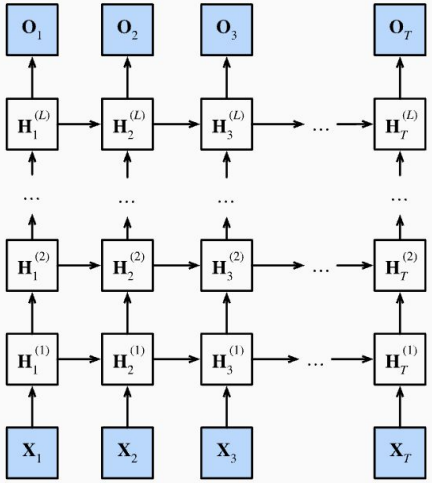
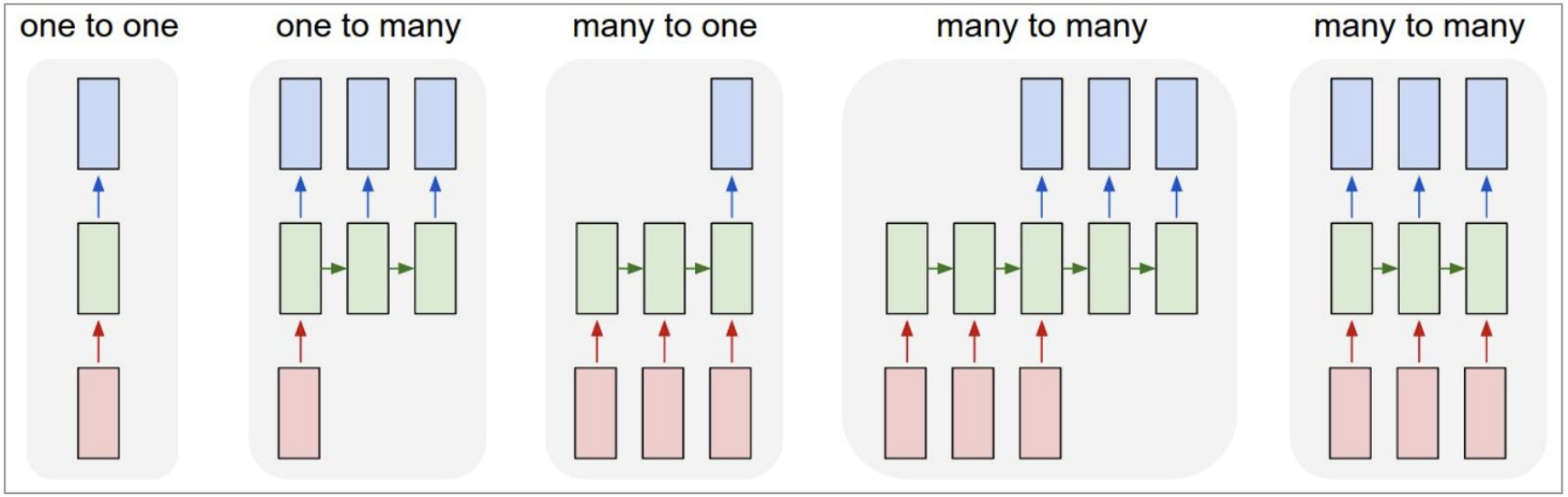


Image Source: Dive into Deep Learning

- no explicit probabilistic/physical model

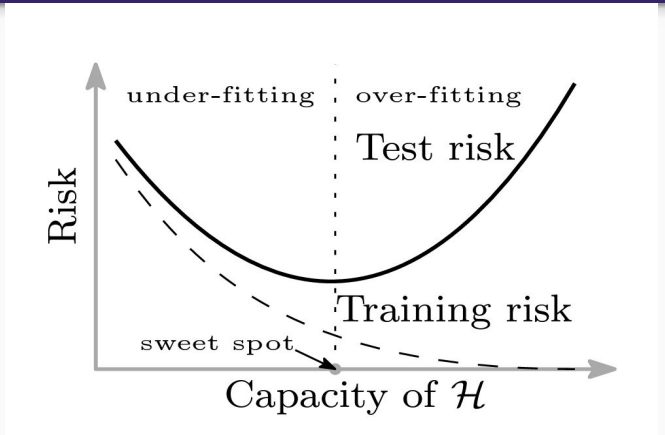
Time Series Tasks



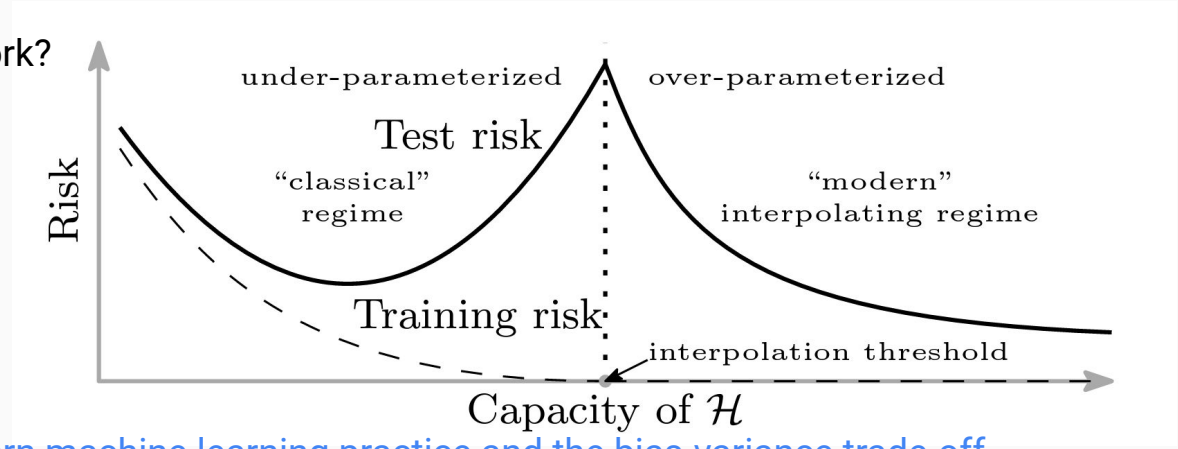
- **one to many:** from radar image of a hurricane predict the future path
- **many to one:** predict water level fo next day based on meteorological inputs
- **many to many:** forecast rainfall for next 5 days based on last 5 days

Deep Learning and Overfitting

Training vs Testing Error



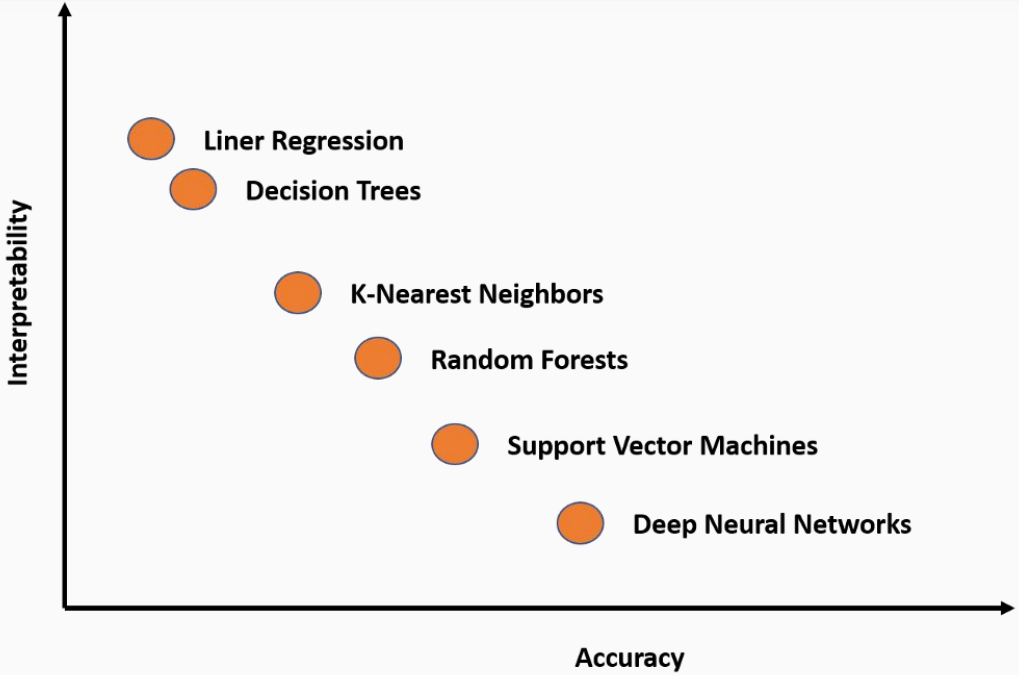
Why does training deep models work?



Belkin, et. al., Reconciling modern machine learning practice and the bias-variance trade-off

Interpretability: can we open the black box?

- Regression:
  - significance of coefficients
- Lasso Regression:
  - few positive coefficients
- Decision Trees:
  - split on important variables
- Random Forest:
  - variable importance
- Hierarchical clustering
  - dendrogram structure
- PCA
  - biplots
- Neural Networks:
  - embeddings layers,
  - activation
  - attention
  - ...



Source: https://mc.ai/interpretability-vs-accuracy-the-friction-that-defines-deep-learning/

Machine Learning Reproducibility

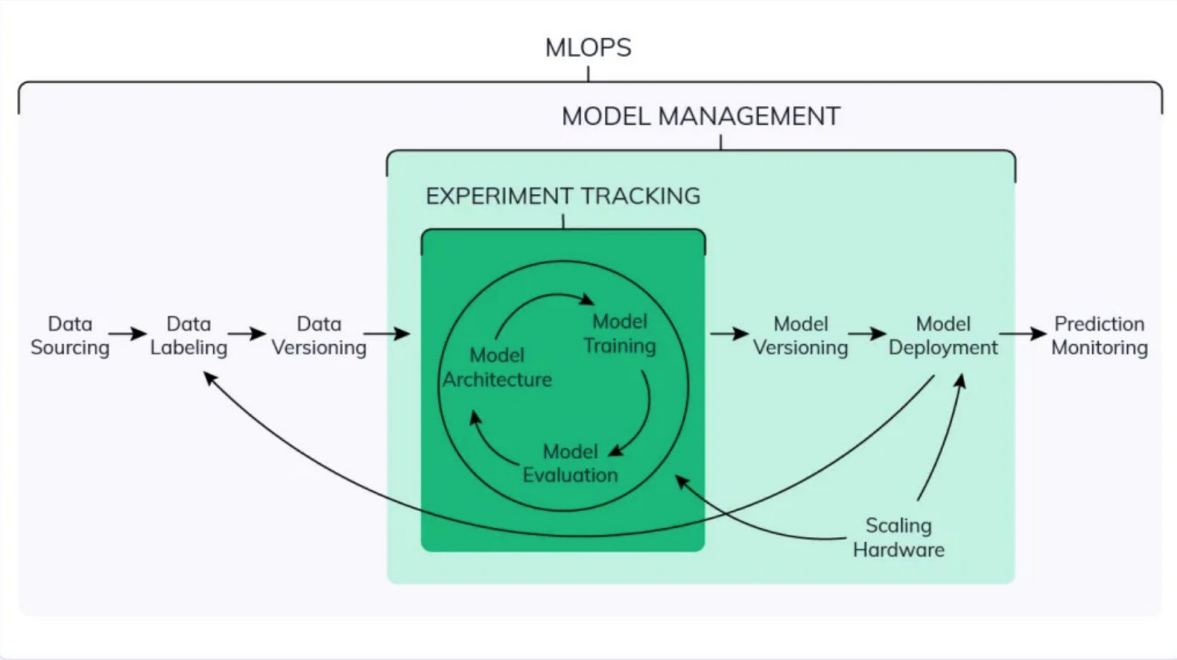
Field	Paper	Number of papers reviewed	Number of papers with pitfalls	[L1.1] No test set	[L1.2] Pre-proc. on train-test	[L1.3] Feature sel. on train-test	[L1.4] Duplicates	[L2] Illegitimate features	[L3.1] Temporal leakage	[L3.2] Non-ind. b/w train-test	[L3.3] Sampling bias	Comput. reproducibility issues	Data quality issues	Metric choice issues	Standard dataset used?
Medicine	Bouwmeester et al. (2012)	71	27	○								○			
Neuroimaging	Whelan & Garavan (2014)	–	14	○		○									
Autism Diagnostics	Bone et al. (2015)	–	3			○				○		○	○	○	
Bioinformatics	Blagus & Lusa (2015)	–	6		○										
Nutrition Research	Ivanescu et al. (2016)	–	4	○								○	○		
Software Eng.	Tu et al. (2018)	58	11					○			○	○		○	
Toxicology	Alves et al. (2019)	–	1			○					○	○			
Satellite Imaging	Nalepa et al. (2019)	17	17						○			○		○	
Tractography	Poulin et al. (2019)	4	2	○							○	○	○	○	
Clinical Epidem.	Christodoulou et al. (2019)	71	48			○						○			
Brain-computer Int.	Nakanishi et al. (2020)	–	1	○										○	
Histopathology	Oner et al. (2020)	–	1						○						
Neuropsychiatry	Poldrack et al. (2020)	100	53	○	○							○	○		
Medicine	Vandewiele et al. (2021)	24	21			○			○	○	○	○		○	
Radiology	Roberts et al. (2021)	62	62	○			○			○	○			○	
IT Operations	Lyu et al. (2021)	9	3					○						○	
Medicine	Filho et al. (2021)	–	1				○								
Neuropsychiatry	Shim et al. (2021)	–	1			○					○				
Genomics	Barnett et al. (2022)	41	23			○						○			
Computer Security	Arp et al. (2022)	30	30	○	○	○		○		○	○		○	○	

- No test set
- Pre-processing on train–test
- Feature Selection on train-test
- Duplicates
- Illegitimate features
- Temporal Leakage
- Non-independence b/w train-test
- Sampling bias

- Computational Reproducibility
- Data Quality Issues
- Metric Choice Issues

Kapoor S. and Narayan A.,Leakage and the Reproducibility Crisis in ML-based Science

ML Experiment Tracking



Source: [neptune.ai]

- Version Control for models, data, results + monitoring
- Facilitating the reproducibility of the ML experiments

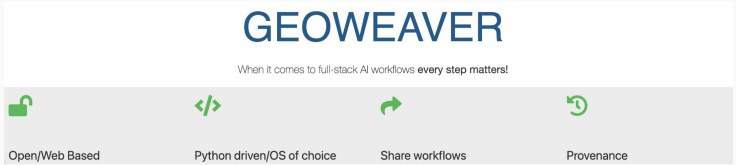
ML Experiment Tracking Tools

Explosion of Experiment Tracking Tools:

	mlflow	DVC	CLEARML	TensorBoard	Weights & Biases	comet	DagsHub Logger	DagsHub MLflow
Open Source	✅ Apache	✅ Apache	✅ Apache	✅ Apache	❌	❌	✅ MIT	✅ MIT
Platform & language agnostic	✅	✅	❌	❌	❌	❌	✅	✅
Experiment Data Access Local / Cloud								
Remote server set up for the user	❌	❌	✅	❌	✅	✅	✅	✅
Custom Visualizations	✅	✅	✅	✅	✅	✅	❌	❌
Scalable for a large number of experiments	✅	❌	✅	❌	✅	✅	✅	✅
Auto-logging	✅	✅	✅	❌	❌	✅	✅	✅
Collaboration features	✅	❌	✅	❌	✅	✅	✅	✅
Team-based access	❌	✅	✅	❌	✅	❌	✅	✅

DagsHub Blog

Geoweaver: OS workflow management system for earth scientists



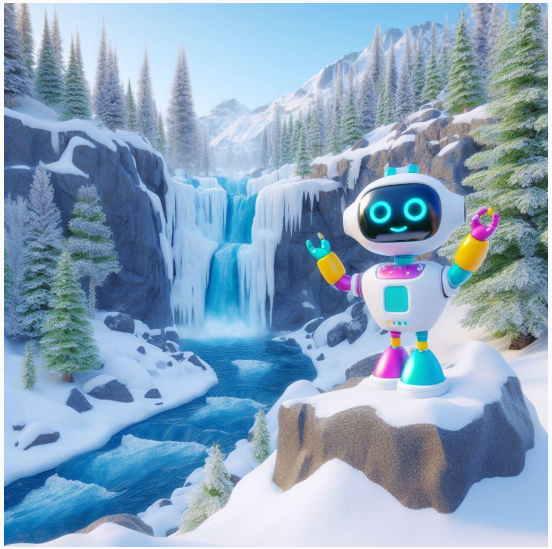
Python Machine Learning Ecosystem

- General Machine Learning Libraries:
  - Scikit-Learn, PyCaret
- Deep Learning Libraries:
  - Tensorflow, PyTorch, Pytorch Lightning, Keras, Fast.AI
- Parallelizing machine learning workflows
  - Dask ML, Ray
- Free Jupyter Environments with GPU:
  - Kaggle, Colab, Planetary Computer
- Cloud Computing for ML:
  - AWS Sage Maker, Azure Machine Learning, Cloud ML
- Deployment:
  - Kubernetes, Kubeflow, Tensorflow TFX
- Orchestration Tools:
  - Airflow, Dagster, Prefect, Flyte

Resources

Extra resources:

- Scikit-learn cheat sheet
- Cross-Validation Schemes
- Precision-Recall Animations
- Python Data Science Handbook ML chapter with notebooks
- Deep Learning Book
- Overview of Deep Semi-Supervised Learning
- Human in the Loop Machine Learning
- Trustworthy AI for Environmental Science



generated with AI with keywords:
AI, robot, glacier, waterfall, cartoon

Happy (Machine) Learning!