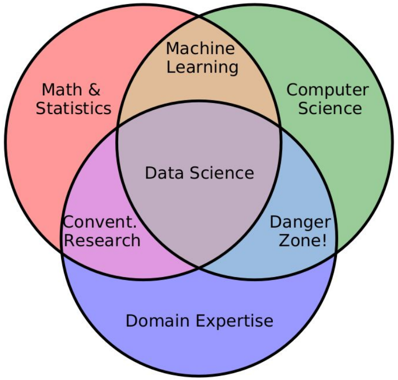
Openness wins!

Introduction

- Introduction
- Use case: cloud detection from satellite imagery
- Basics of a few ML algorithms
- Cloud detection using ML
- Results and conclusions



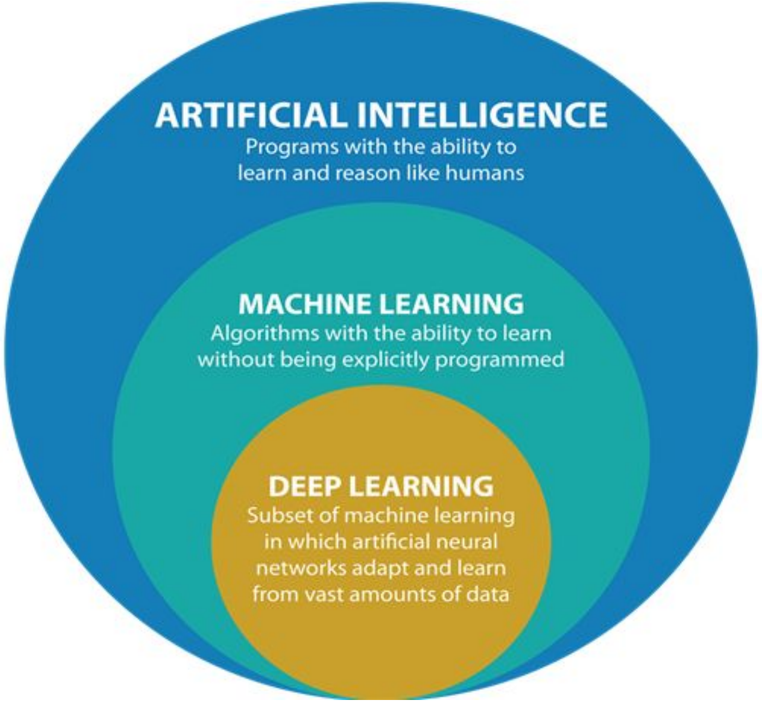
Data vs knowledge

Data	Knowledge
- is raw; it does not have meaning of itself - has no significance beyond its existence (in and of itself) - can exist in any form, usable or not	- refers to classes of instances and describes general patterns, structures, laws, principles etc. - consists of as few statements as possible (Occam's razor) - often difficult and time-consuming to find or to obtain - allows us to make predictions and forecasts

"We are drowning in data, but starved for knowledge"

Machine learning & Artificial Intelligence

- ML focuses on applications that learn from experience and improve their performance level on a given task over time
- ML algorithms can be used to predict the outcome in new situation or to understand and explain how prediction is derived
  - Many state-of-the-art ML methods produce non-interpretable ”black-box” models
- Algorithms acquire structural descriptions (representing patterns explicitly) from examples



Marvin Minsky: Learning is making useful changes in our minds

Supervised vs unsupervised learning

Classical division in machine learning

1) Supervised learning

- Classification
- Regression (like classification, but with numerical outcome)

2) Unsupervised learning (tools for exploratory data analysis)

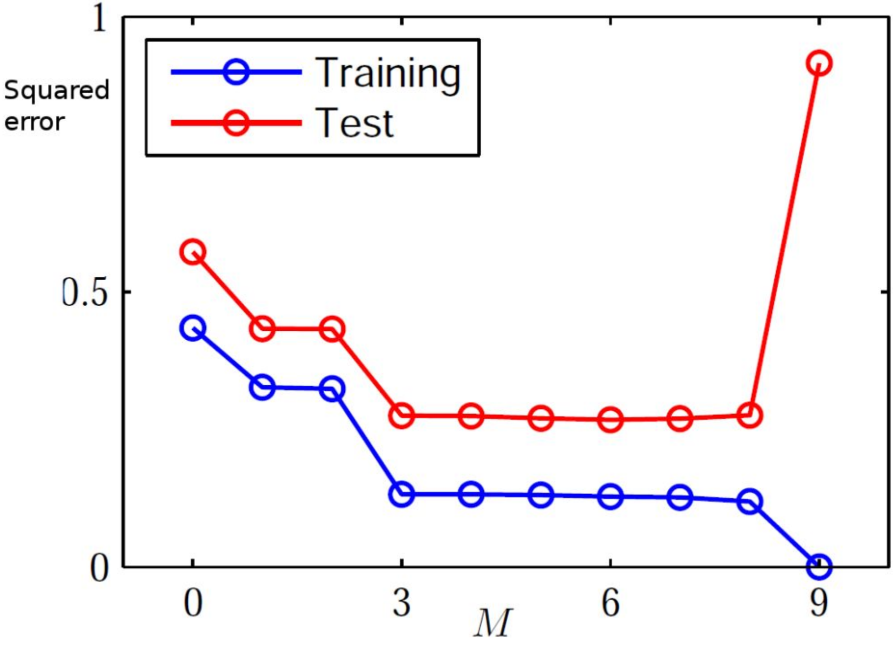
- Clustering, segmentation
- Association or deviation analysis (outlier detection)
- Most dimensionality reduction methods

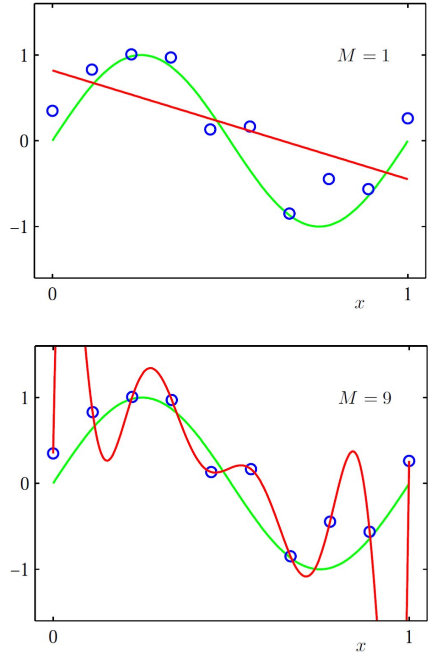
Building blocks of supervised ML algorithm

- Model class: general structure of the model (hypothesis set)
  - E.g. linear or quadratic function, decision tree, neural network
- Error measure (Score function): evaluates quality of different models
  - E.g. squared error
- Algorithm: finds good model, as defined by score function
  - Mathematical optimization
- Validation
  - Finding best fit to training data does not guarantee accurate predictions on new data
  - Danger of overfitting
    - Worst case: model just memorizes training data; does not generalize beyond it
  - Alternatively underfitting:
    - Model class not expressive enough (linear functions on non-linear problems)

Approximation-generalization trade-off:

The aim of learning is to approximate the real function behind the explored phenomena as closely as possible but still yield a small prediction error for test samples too.





Illustrative toy example

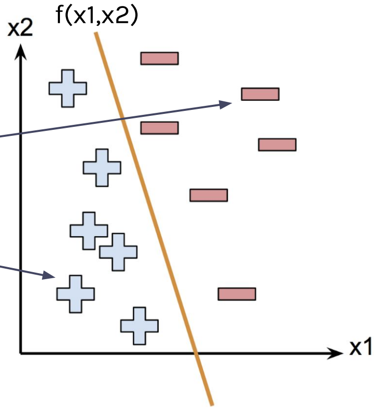
1) Training phase
- Each training sample gets represented by a list of features

Training data

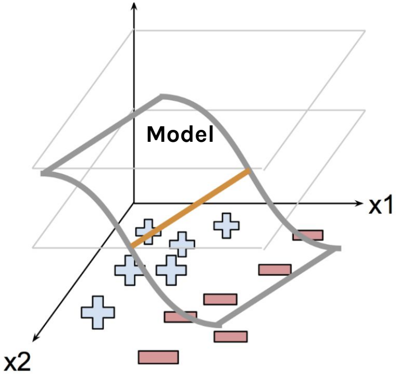
Training labels

+, +, -, +, -

Create a function f(x1,x2) for separating the samples according to their class labels



Value of the function f



2) Prediction phase
- Use the fitted function f to predict the class of the new sample

f(x1_test, x2_test) ⟶ +/-

3) Evaluation phase
- Does the predicted label match with the original one?

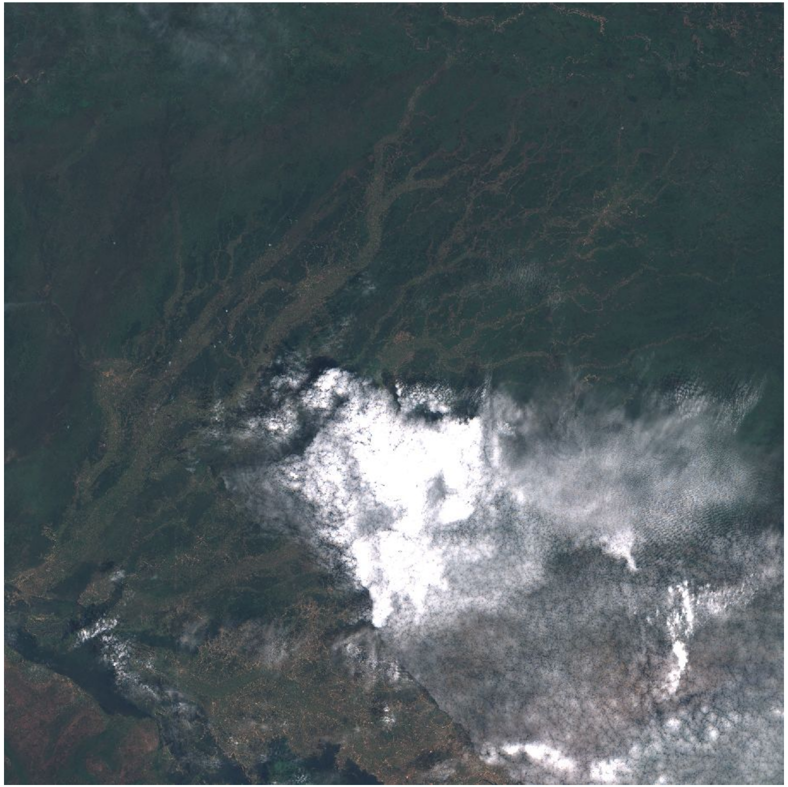
Use case: Cloud Detection

Goal: Detect cloud coverage in satellite images

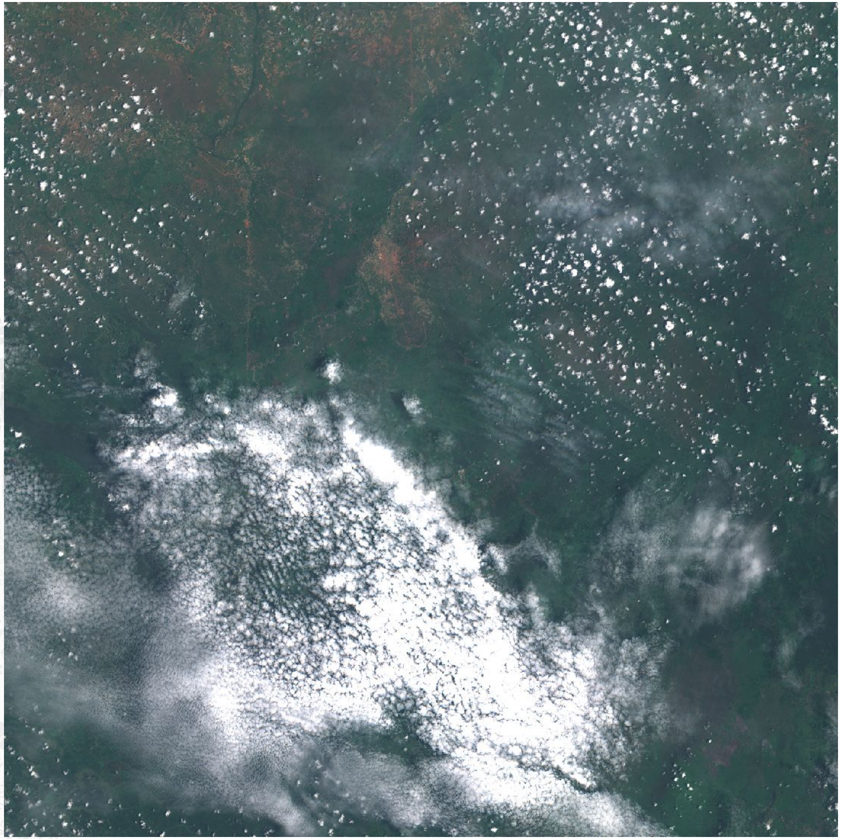
Solution:

1. Detect clouds
2. Create cloud masks

- UN CFSI pilot project: s2cloudless, Fmask
- Resulting in mosaic images with most-recent cloudless data



From satellite image to feature vectors



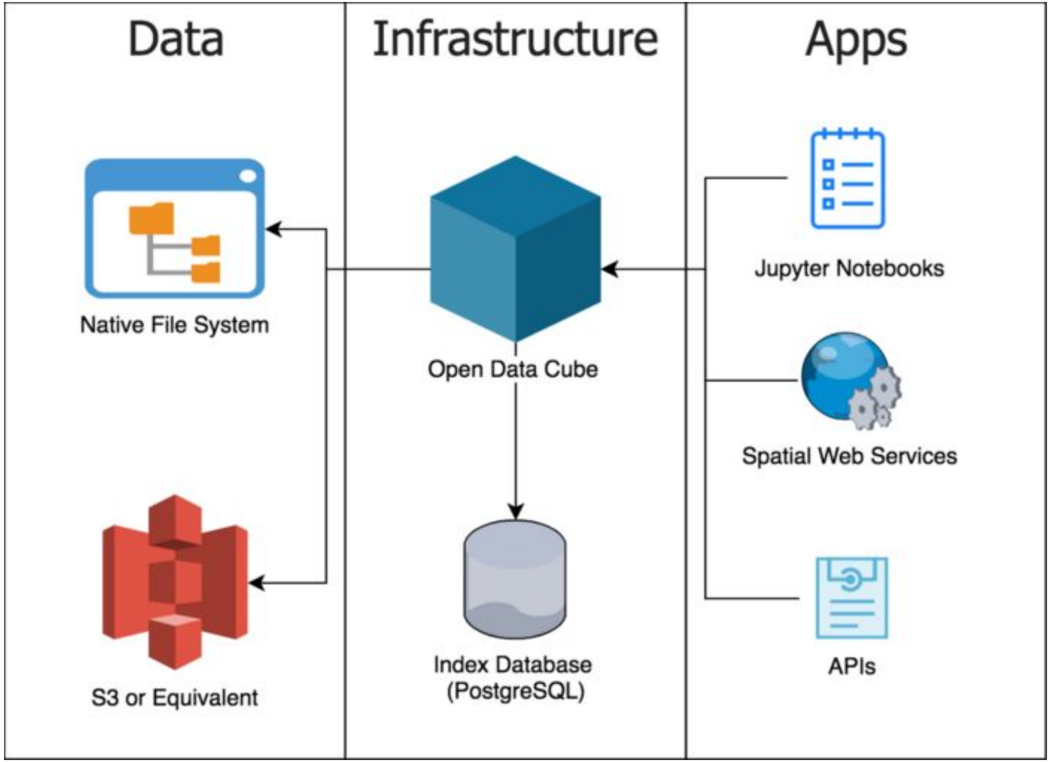
```
                 Band 1                                      Band 13
Pixel 1          [[1190  948  899 ... 1788  844 2596]
                  [1183  978 1036 ... 1800  804 3273]
                  [1176  951  974 ... 1766  786 3132]
                  ...
                  [1183  975  977 ... 1659  741 3062]
                  [1179  934  909 ... 1373  621 2578]
Pixel 3 350 000   [1187  950  925 ... 1450  646 2817]]
```

```
Label for pixel 1          [[0]
                            [1]
                            [0]
                            ...
                            [1]
                            [0]
Label for pixel 3 350 000   [0]]
```

OpenDataCube

- Open-source Python library for handling RS data
- Supports data indexing, ingesting, and querying
- Creates spatio-temporal datacubes
- Can be used as a base for various applications
    - Cloudless mosaics
    - Change detection
    - …

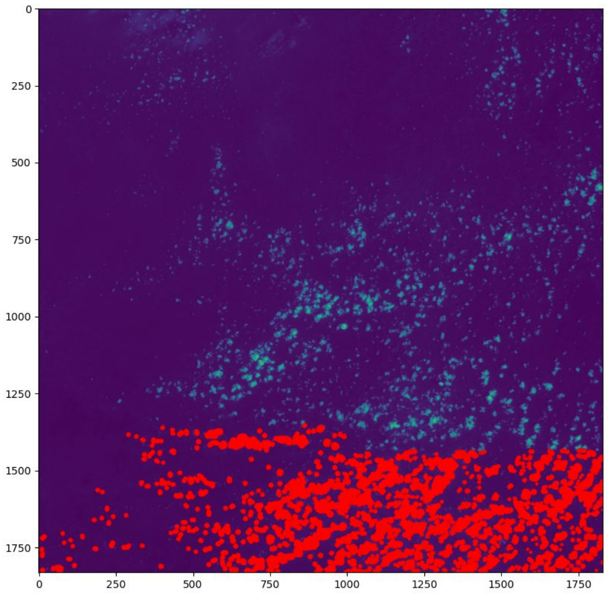
OpenDataCube



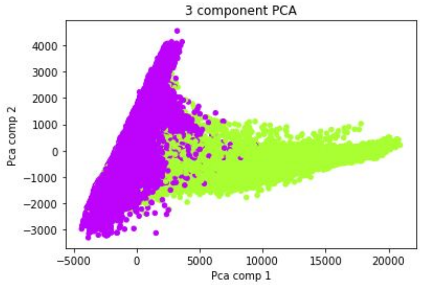
https://medium.com/opendatacube/what-is-open-data-cube-805af60820d7

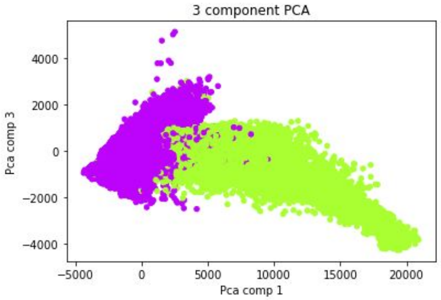
Training data

- Basis for teaching any ML algorithm
- Created manually using mpl-pixel-picker
- Alternative: use existing cloud masks
- Output: JSON of cloudy pixels
- Transform to raster and eventually to feature vectors with python

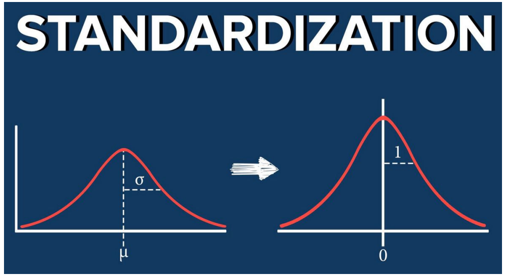


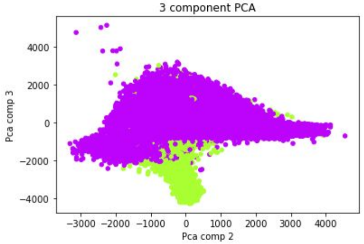
Preprocessing





- Principal Component Analysis
- Standardization
  - Normalization



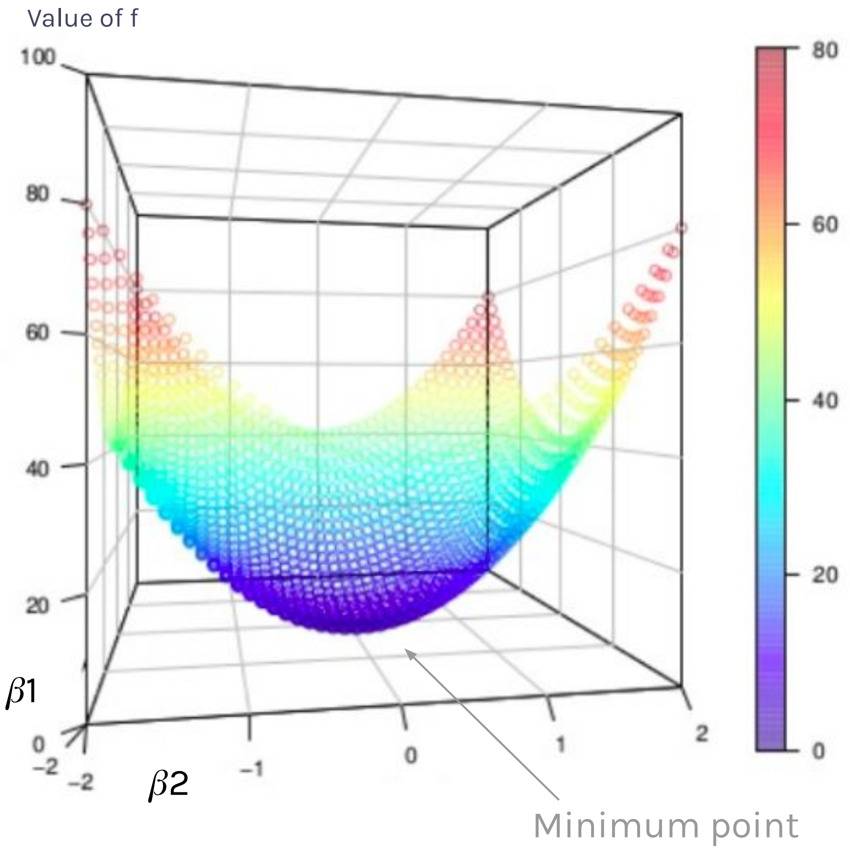


Note. Both of these methods need to be applied separately to the training and test sets in order to not leek valuable information from the test samples into the training phase of the algorithms!

Ridge Regression algorithm

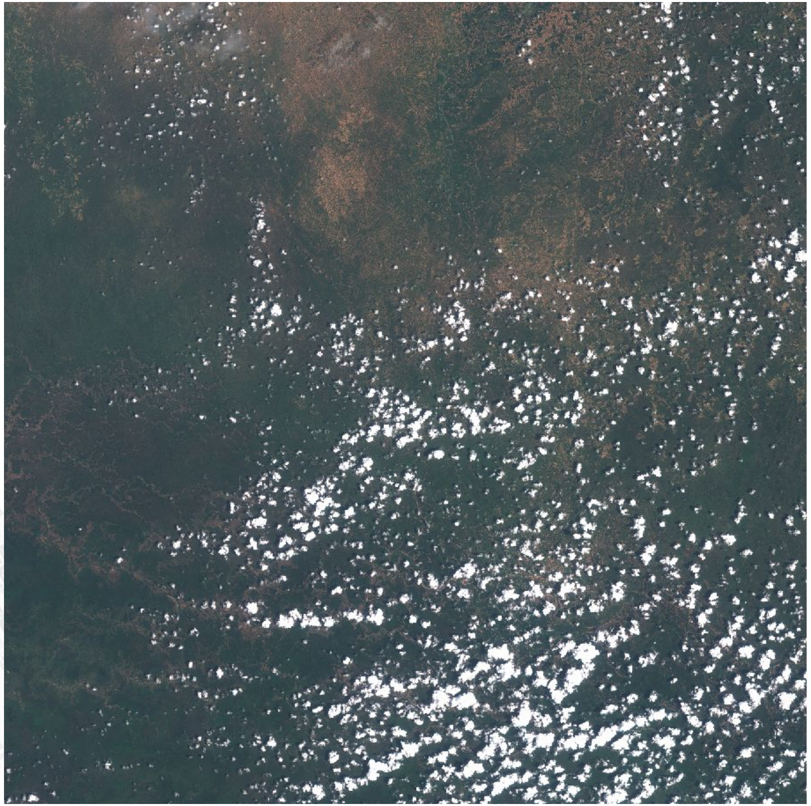
Minimize function f determined as
Loss function (Ordinary Least Squares) +
Penalty term (L2-norm)

$$\underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{N} \Big[\underset{\text{Ground truth}}{y^i} - \underbrace{sign\left(\beta_1 x_1^i + \beta_2 x_2^i\right)}_{\text{Prediction}} \Big]^2 + \lambda(\beta_1{}^2 + \beta_2{}^2)$$

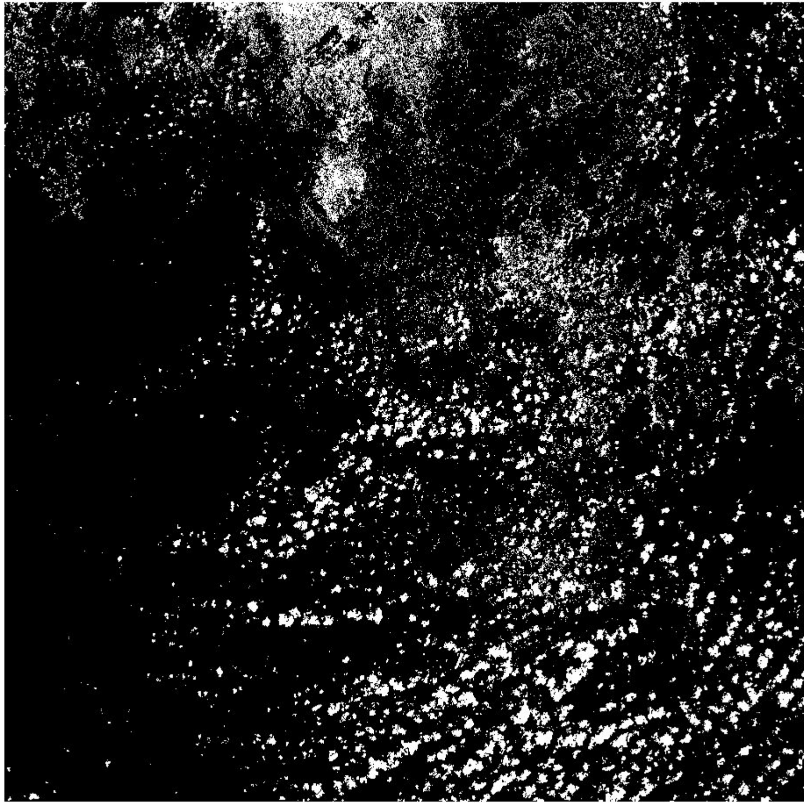


Ridge Regression prediction result, OA: 0.86

Ground truth

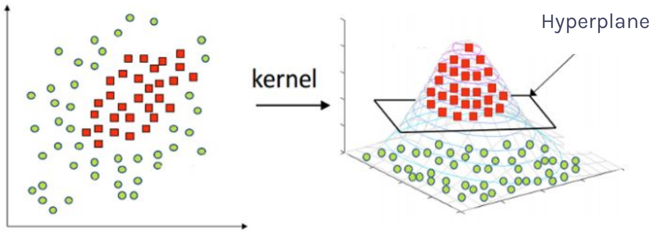


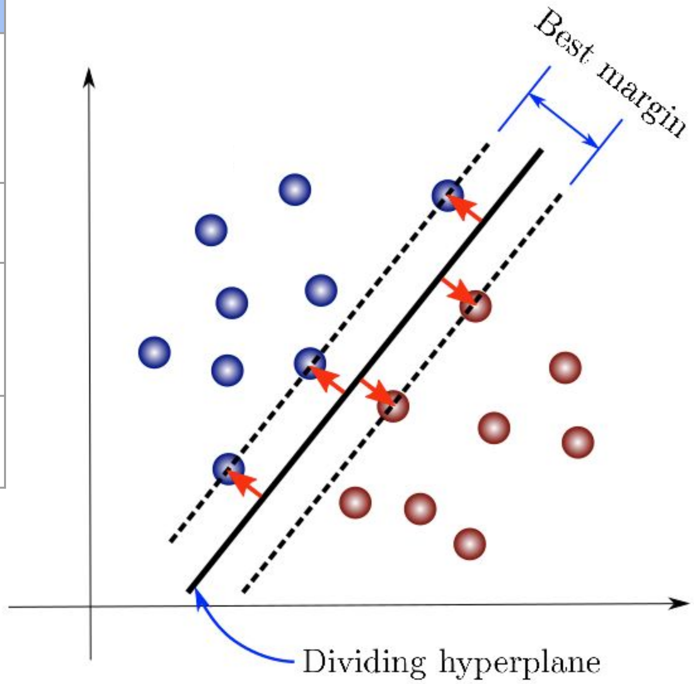
Predictions



Support Vector Machine algorithm

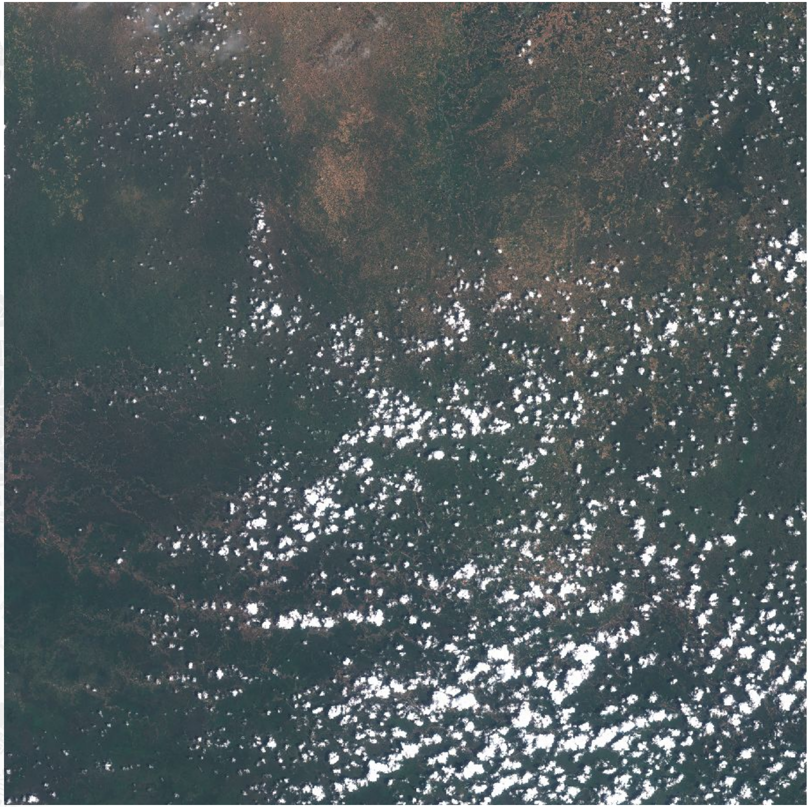
Primal	Dual
Minimize $\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$ s.t. $\forall i:\ y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + w_0\right) \geq 1$	Maximize $\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)$ s.t. $\forall i:\ \alpha_i \geq 0 \ and \ \sum_{i=1}^{N} y_i \alpha_i = 0$
More intuitive	Utilizes Lagrange multipliers
Variables define the hyperplane formula	More efficient (if # features > # of samples)
	Allows to apply kernel trick



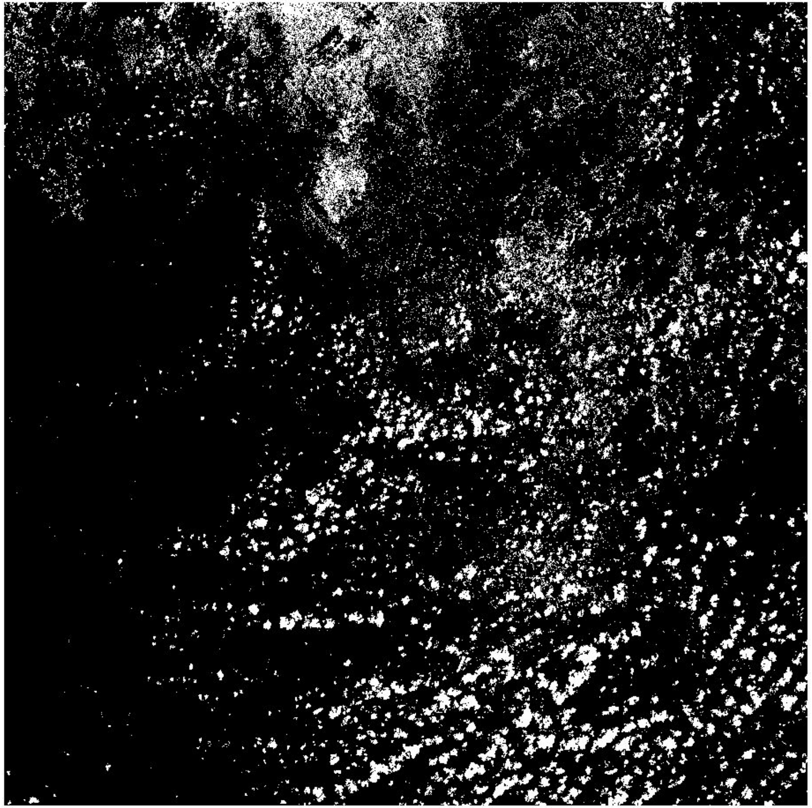


SVM prediction result, OA: 0.86

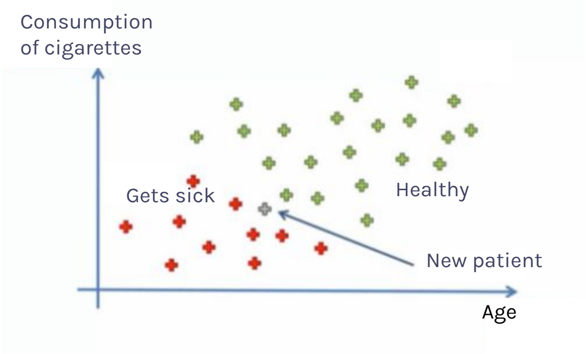
Ground truth



Predictions



Naive Bayes algorithm



1. Calculate Prior Probability
- **P(Healthy)** = the total number of healthy persons divided by the total number of observations
2. Calculate Marginal Likelihood (Evidence)
- Select a radius of our own choice and draw a circle around the new data point
- Deem every other data point in that circle to be about similar in nature
- **P(X)** defines the likelihood of any new random variable added to this dataset that falling inside this circle
3. Calculate Likelihood
- Draw a circle of our radius of our choice
- **P(X|Healthy)** defines the Likelihood that a randomly selected red point falls into the circle area

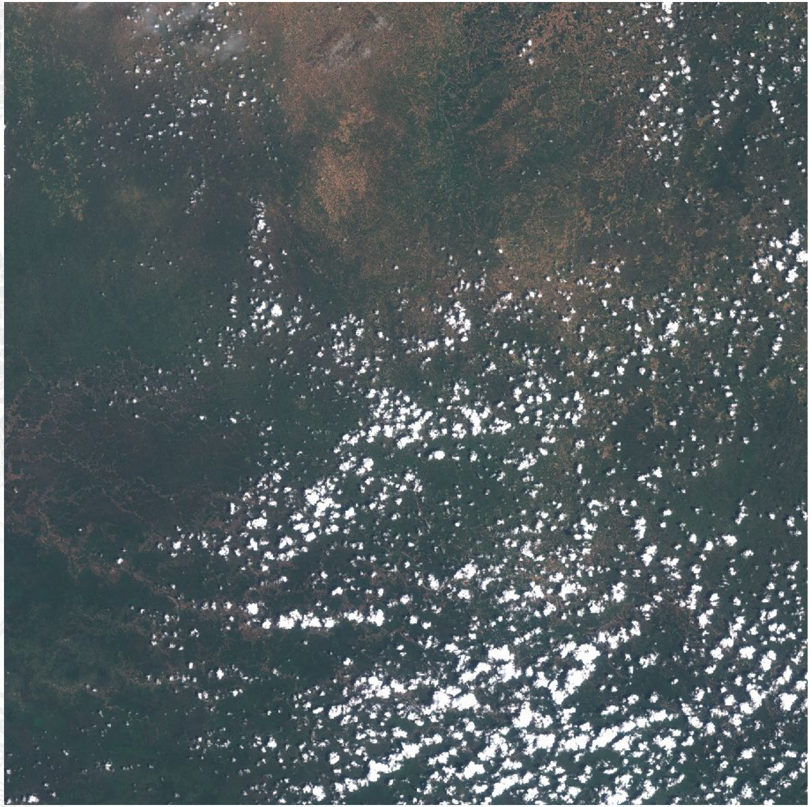
  In other words the likelihood that somebody who does not get sick exhibits feature X

Get Posterior Probability

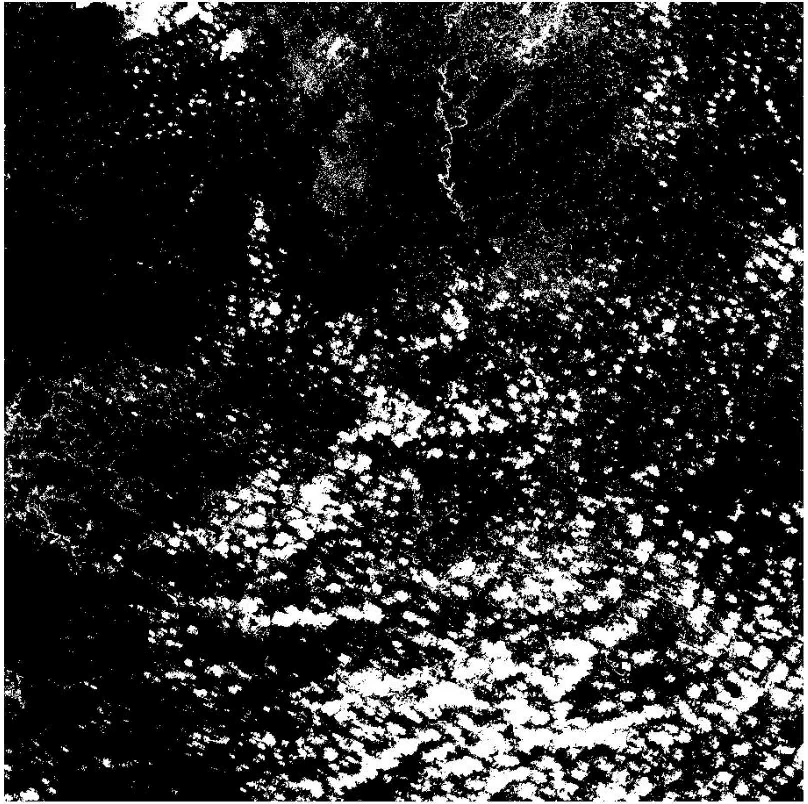
$$\overset{\text{Posterior}}{P(A|B)} = \frac{\overset{\text{Likelihood}}{P(B|A)} * \overset{\text{Prior}}{P(A)}}{\underset{\text{Evidence}}{P(B)}}$$

Naive Bayes prediction result, OA: 0.89

Ground truth

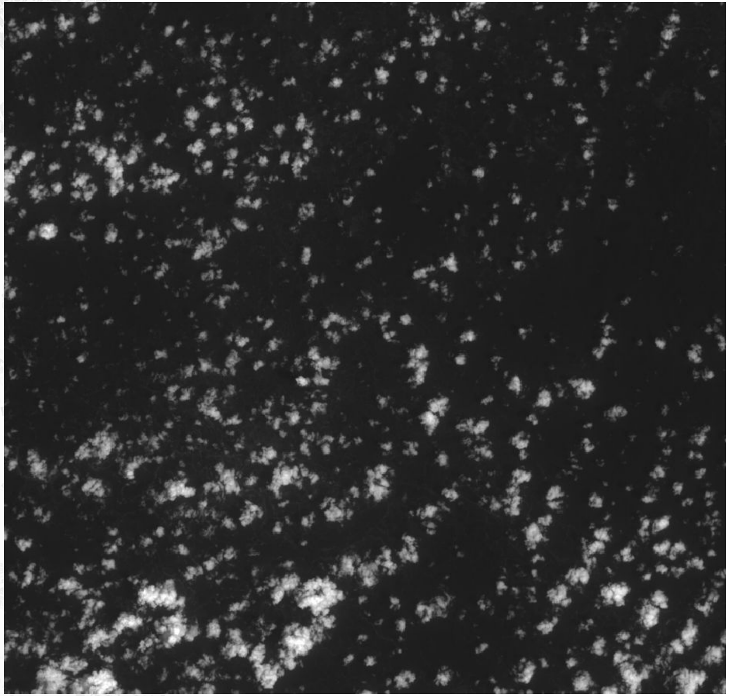


Predictions

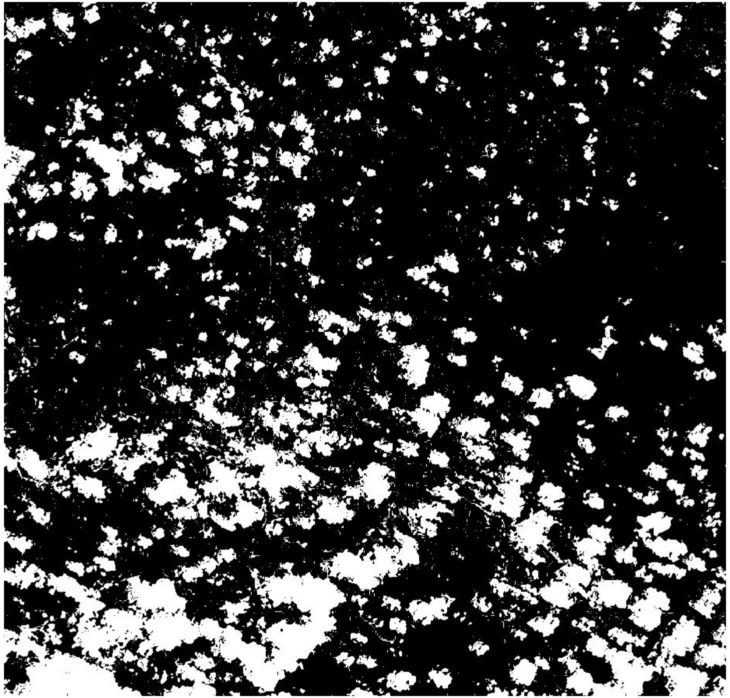


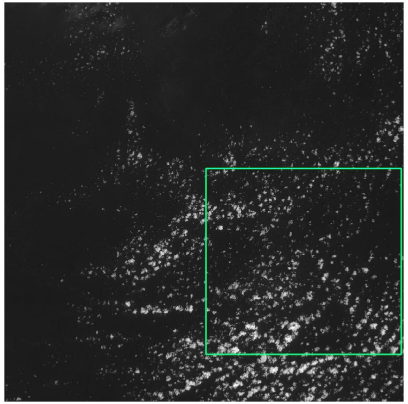
Naive Bayes prediction results
- more closely

Ground truth for subarea



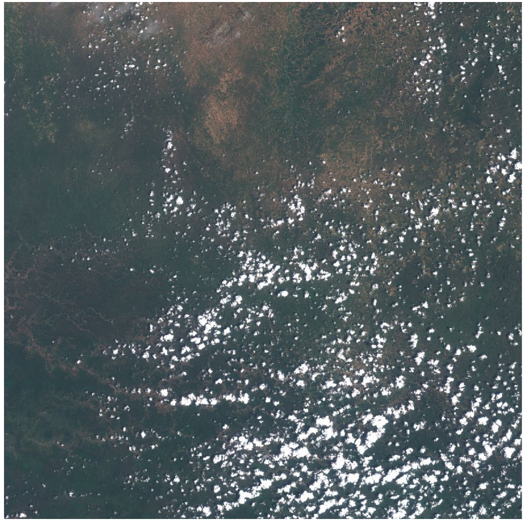
Predictions for subarea

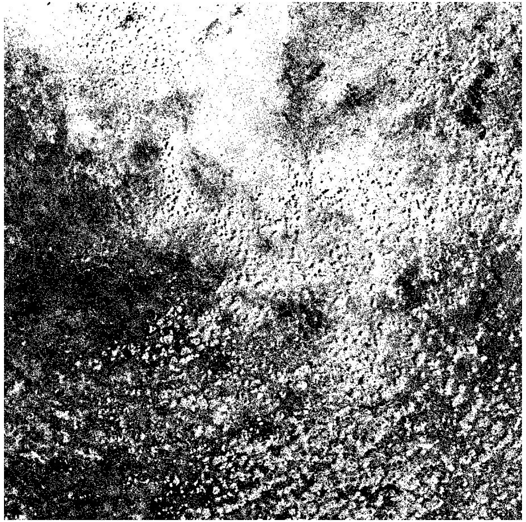




Other popular machine learning algorithms

- K Nearest Neighbors
- Decision Trees →
- Random Forests
- Multilayer Perceptron
  - Field of artificial neural networks
  - Got Killed during execution





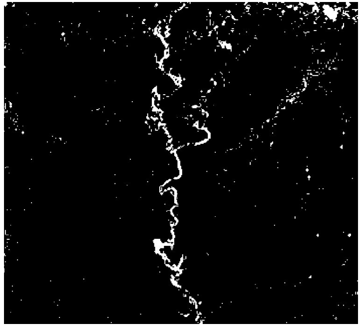
Note. There is no need to limit yourself in using just one ML algorithm

→ Consider Ensemble Learning!

Results (with 60m resolution)

	Training accuracy	Test accuracy	Running time
Ridge Regression	0.91	0.86	Few seconds
Logistic Regression	0.87	0.86	< ½ min
SVM (LinearSVC)	0.91	0.86	< ½ min
Naive Bayes	0.93	0.89	~ ½ min
Voting Classifier	0.91	0.86	~ 1 min
KNN	0.97	0.67	~ 2 mins
Decision Tree	1.00	0.61	~ 2 mins
Random Forest	0.96	0.70	~ 5 mins

Conclusions



- OpenDataCube

The world is not ready; Naive Bayes predicts rivers as clouds

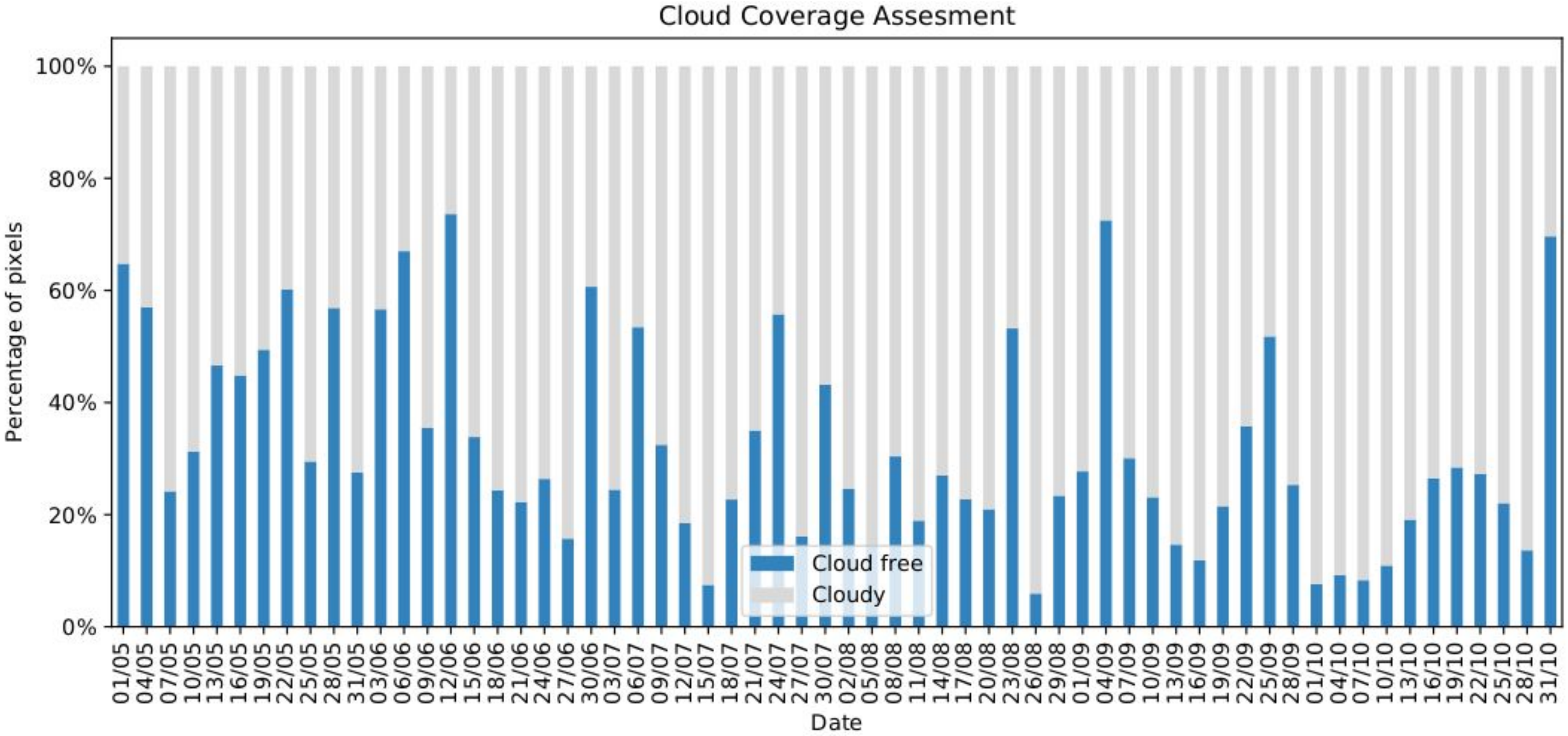
Possible ways to achieve better performance

- Tuning of hyperparameters
    - Crossvalidation
- Tuning of model architectures (especially with neural networks)
- Trying other regularizers (L1 norm, L1+L2 norm, L0 pseudonorm)
- Feature generation (e.g. moments)

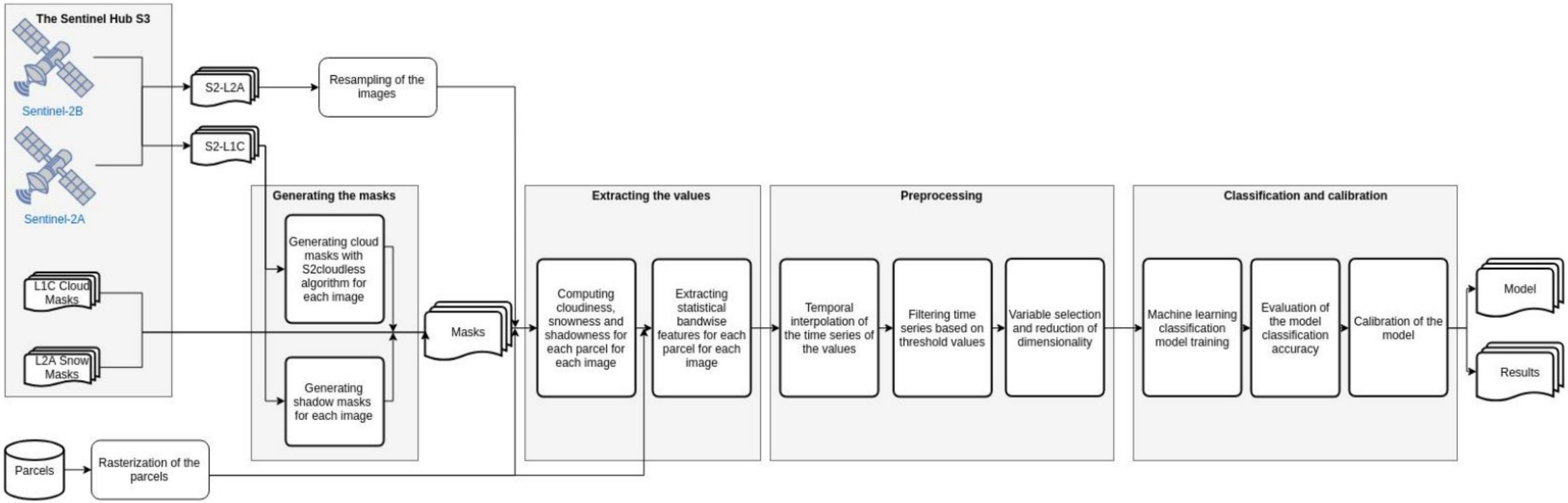
Case: Crop Identification

References: Photo by Tom Fisk from Pexels

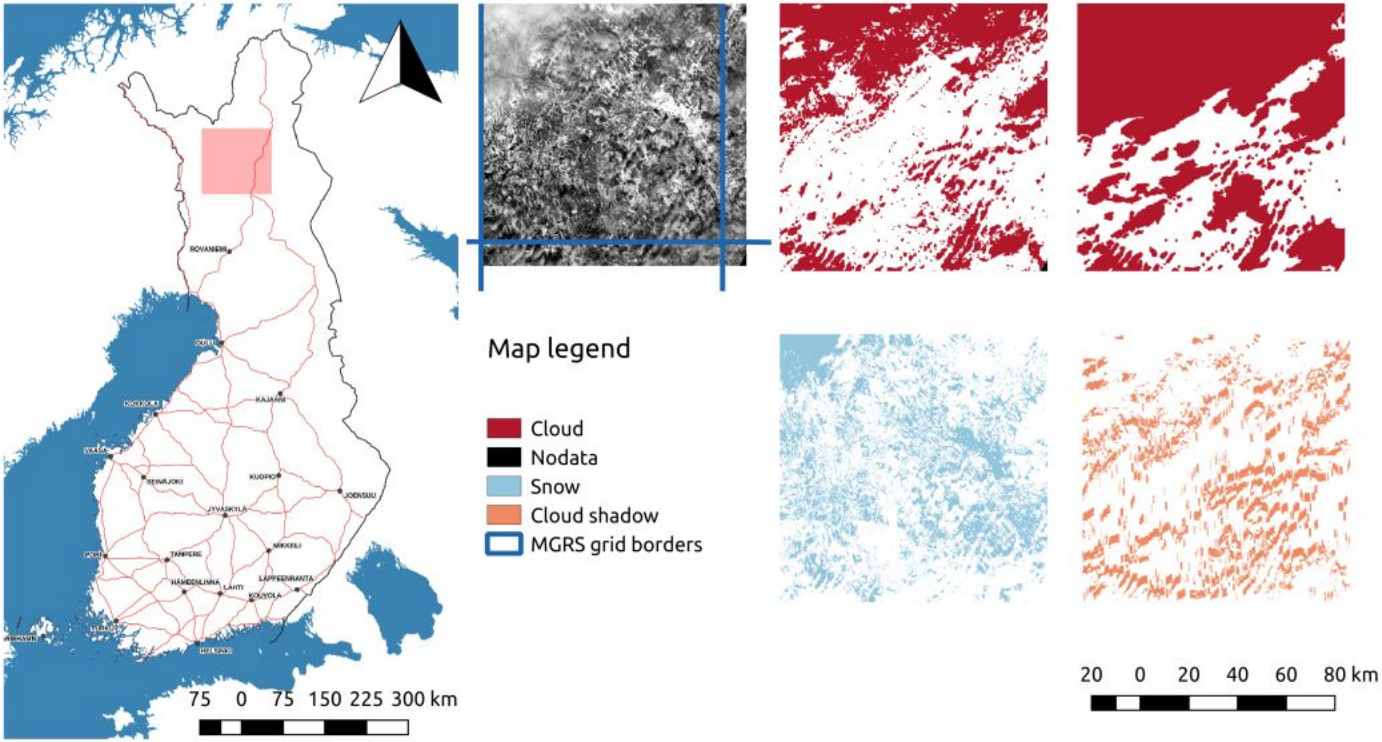
ML & AI: use cases and challenges



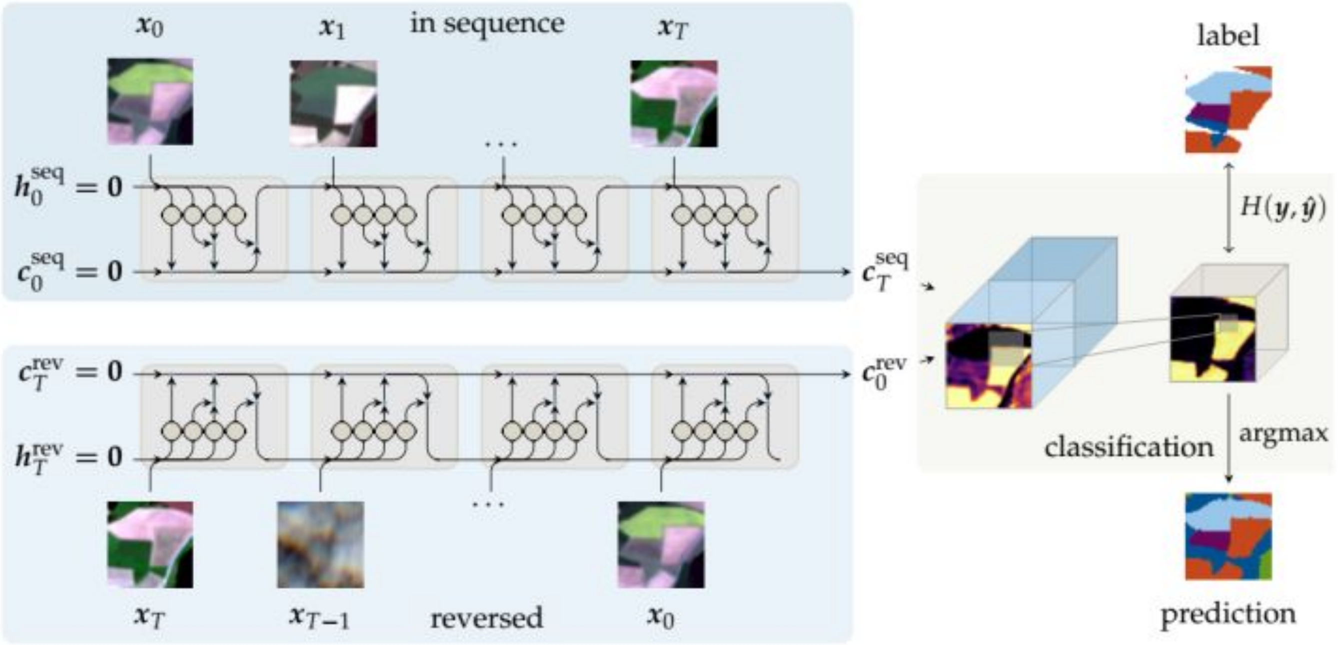
ML & AI: use cases and challenges



ML & AI: quality of data



ConvRNN



x_0
x_1
in sequence
x_T
$\dots$
$h_0^{seq} = 0$
$c_0^{seq} = 0$
c_T^{seq}
$c_T^{rev} = 0$
c_0^{rev}
$h_T^{rev} = 0$
$\dots$
x_T
x_{T-1}
reversed
x_0
label
$H(y, \hat{y})$
classification
argmax
prediction

Thank you!

Kalevankatu 31, 00100 Helsinki

Köydenpunojankatu 14, 20100 Turku

info@gispo.fi

+358 207 310 360

www.gispo.fi

References

Antti Airola: Course materials for Data Analysis and Knowledge Discovery. University of Turku, Department of Information Technology, Spring 2019