# An Introduction to Machine Learning with Python and scikit-learn

Julien Simon
Principal Technical Evangelist, AI and Machine Learning, AWS
@julsimon

# Agenda

- Machine Learning in 5 minutes
- Scikit-learn
- Algos & Demos: Linear Regression, Logistic Regression, Decision Trees, K-Means, Principal Component Analysis
- Scikit-learn on Amazon SageMaker
- Resources

# Machine Learning in 5 minutes

Artificial Intelligence: design software applications which exhibit human-like behavior, e.g. speech, natural language processing, reasoning or intuition

Machine Learning: using statistical algorithms, teach machines to learn from featurized data without being explicitly programmed

Deep Learning: using neural networks, teach machines to learn from complex data where features cannot be explicitly expressed

# Types of Machine Learning

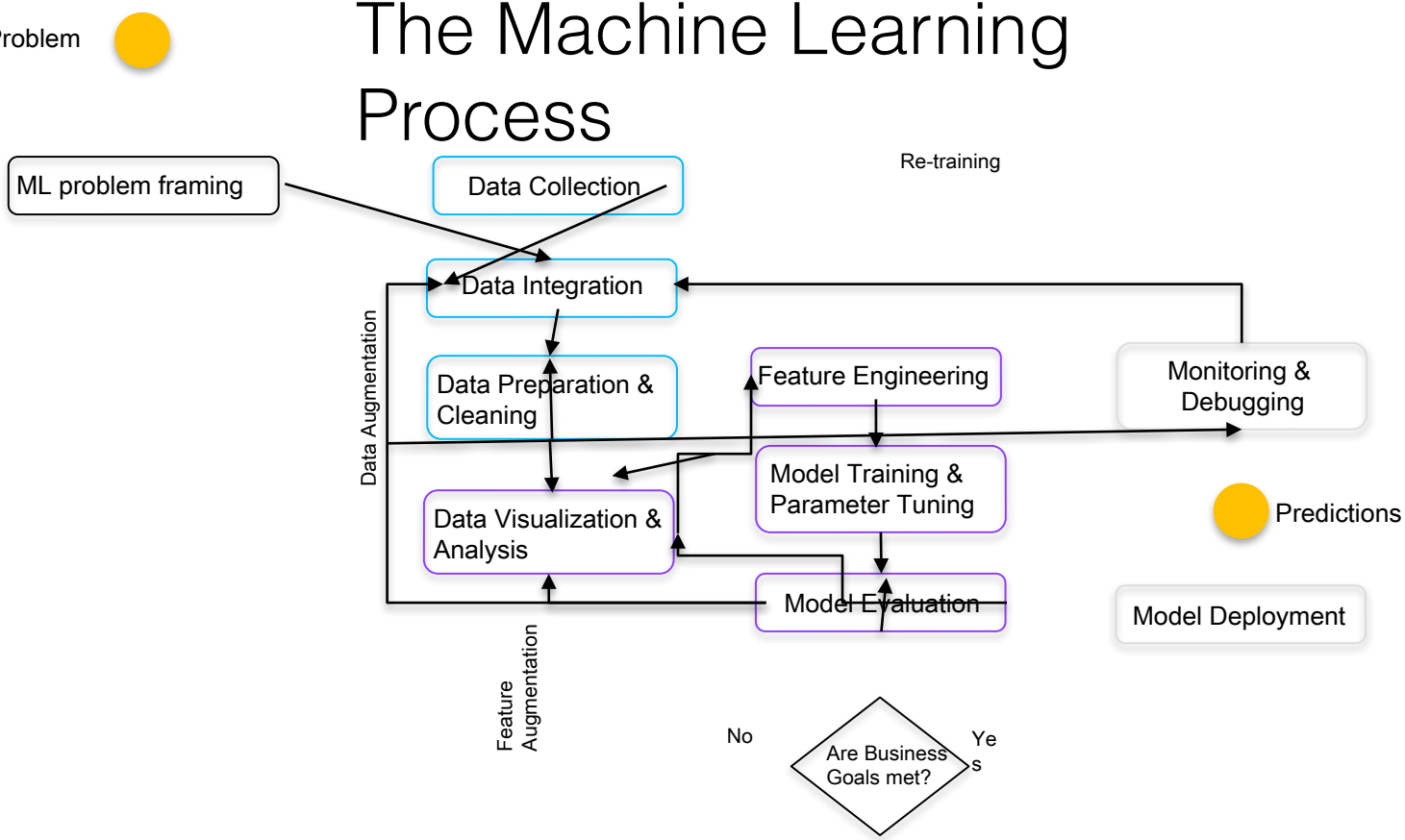
## Supervised learning

- Run an algorithm on a labeled data set.
- The model learns how to correctly predict the right answer.
- Regression and classification are examples of supervised learning.

## Unsupervised learning

- Run an algorithm on an unlabeled data set.
- The model learns patterns and organizes samples accordingly.
- Clustering and topic modeling are examples of unsupervised learning.

# The Machine Learning Process

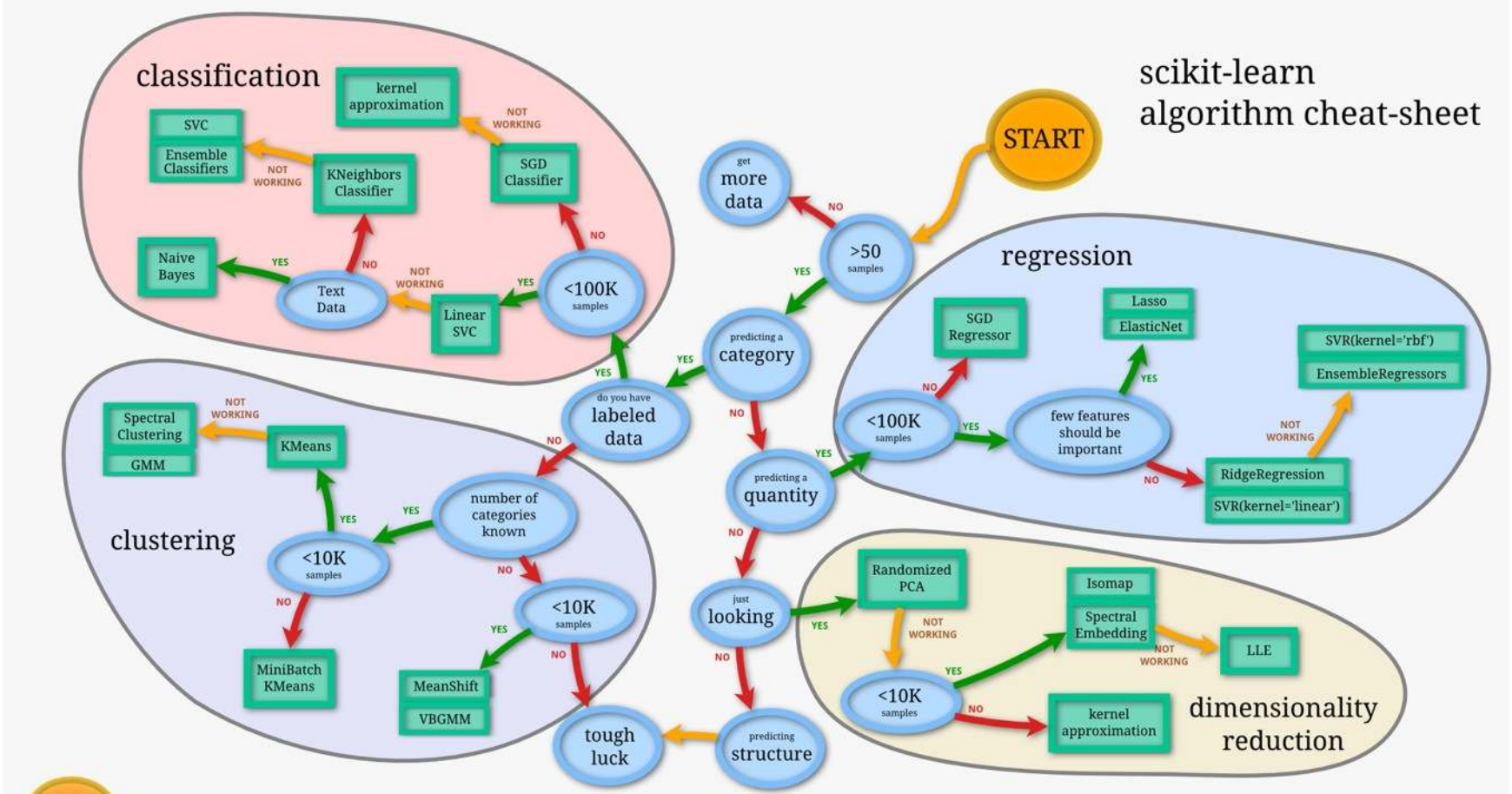


# Scikit-learn

# Scikit-learn


- Open Source library in Python released in February 2010
- Built on NumPy, SciPy, and matplotlib
- Simple tools for data analysis and Machine Learning
- Excellent collection of algorithms
- Very good documentation, tons of tutorials

- *Limited scalability for data sets that don't fit in RAM*
- *Not appropriate for Deep Learning (no GPU support)*



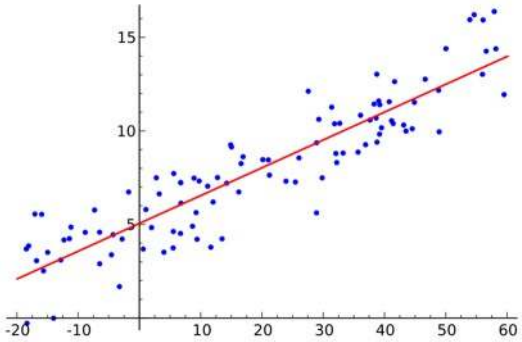



http://scikit-learn.org/stable/tutorial/machine_learning_map/

# Linear Regression

https://en.wikipedia.org/wiki/Linear_regression

- Supervised learning algorithm
- Goal: fit data to a linear function in order to predict numerical values
- Data set: features + target (scalar or scalar vector)
- 1 feature → line, 2 features → plane, etc.
- Intuition: minimize the "distance" between data points and the linear function



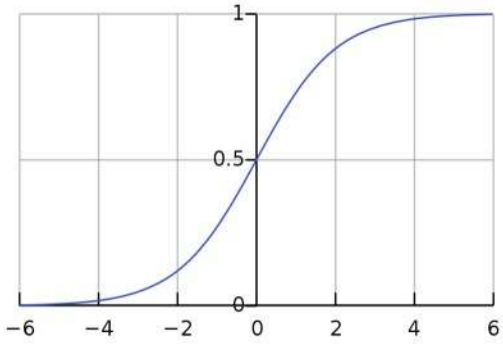

$$y_i = \beta_0 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

*This can also be used for binary classification: a sample is either "above" or "below" the linear function*

# Logistic Regression (1958)

https://en.wikipedia.org/wiki/Logistic_regression

- Supervised learning algorithm
- Goal: fit data to a linear function in order to predict the class of a sample
- Data set : features + binary label (yes/no, true/false, etc.)
  - This algorithm can be extended to more than two classes
- Intuition: find a function computing a score between 0 and 1
  and set a threshold separating both classes

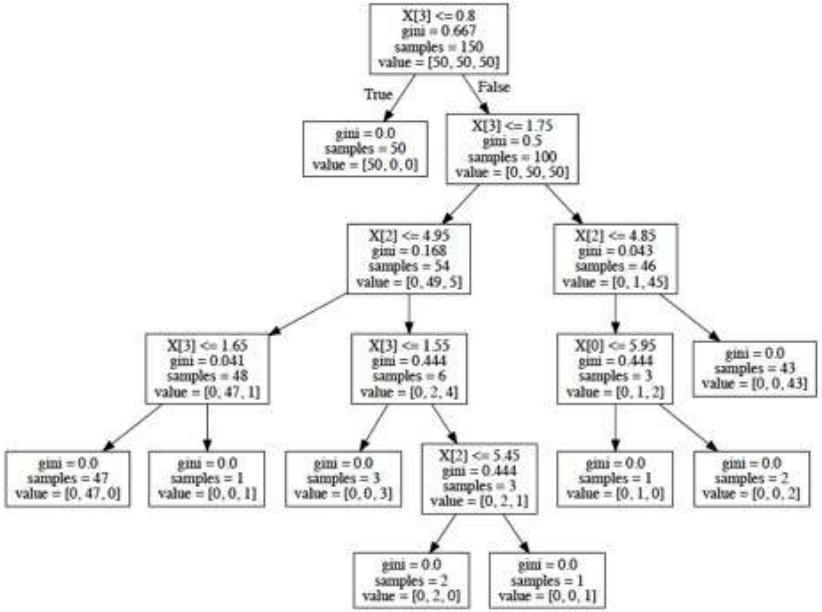



$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

# Decision Trees

https://en.wikipedia.org/wiki/Decision_tree

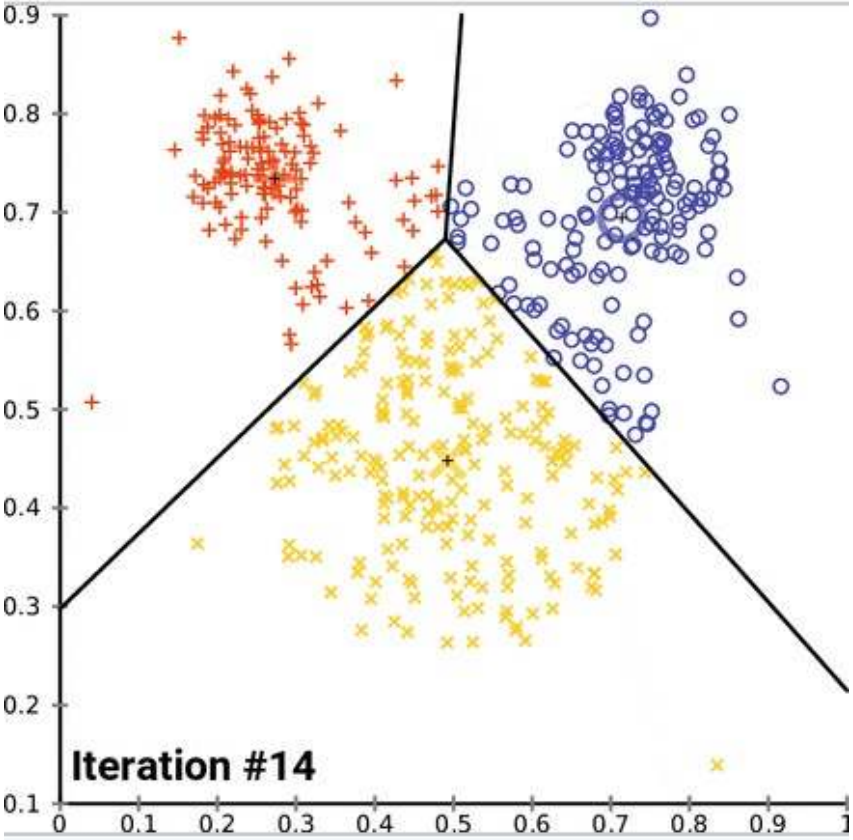
- Supervised learning algorithm
- Goal: build a decision tree for regression or classification
- Data set : features + target value / class
- Intuition: find the “best” feature thresholds to go left or right
- “Easy” to interpret, but prone to overfitting
- Plenty of advanced variants with multiple trees: Random Forests, XGBoost (2016), etc.



# K-means (1957)

https://en.wikipedia.org/wiki/K-means_clustering

- Unsupervised learning algorithm
- Goal: group samples in ‘k’ clusters
- Data set : features only
- Intuition: find ‘k’ cluster centers that minimize the “distance” to their respective samples
- This assumes “spherical” clusters of similar ”radius”: maybe, maybe not!



# Principal Component Analysis aka PCA (1901!)

https://en.wikipedia.org/wiki/Principal_component_analysis

- Unsupervised learning algorithm
- Goal: build a new data set with a smaller number of uncorrelated features (aka Dimensionality Reduction)
  - …keeping as much variance as the number of new features will allow
- Data set : features only
- Sample use cases:
  - Visualize high-dimension data sets in 2D or 3D
  - Remove correlation in high-dimension datasets
  - Preliminary step to building linear models

# Demos

- Linear Regression
- Logistic Regression
- Decision Trees
- K-Means
- PCA
- PCA + Logistic Regression on MNIST

https://gitlab.com/juliensimon/aws --> ML/scikit

# Scaling scikit-learn

- Scikit-learn runs on a single machine, loading the full data set in memory
- Scaling options are quite limited
  http://scikit-learn.org/stable/modules/computing.html
  - Some algorithms can leverage multi-core (*joblib)*
  - Some algorithms support incremental training
- Amazon SageMaker can help
  - Use ML-optimized multi-core instances (C5)
  - Use pipe mode, i.e. the ability to stream data from Amazon S3

# Beyond scikit-learn

- Amazon SageMaker
  - Train models on fully-managed infrastructure at any scale
  - Built-in algorithms (17) for regression, classification, etc.
  - Built-in environments for Deep Learning
- Apache Spark MLLib
  - Available in Amazon Elastic Map Reduce (EMR)
  - Distributed processing by design
  - Nice collection of Machine Learning algorithms
  - Seamless integration with Amazon SageMaker (Scala / PySpark SDK)

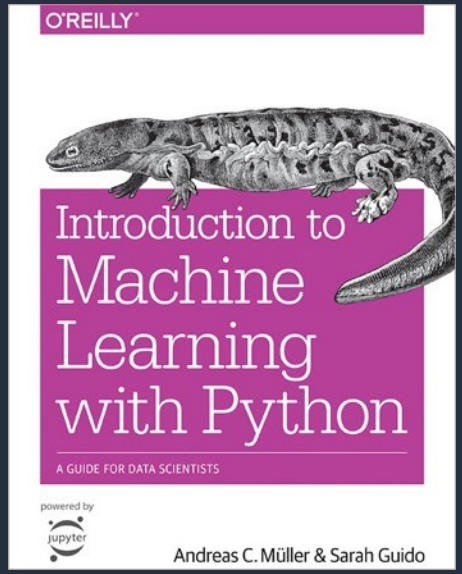
# Resources

https://ml.aws
https://aws.amazon.com/sagemaker

https://scikit-learn.org
https://www.numpy.org

https://machinelearningmastery.com

https://medium.com/@julsimon
https://gitlab.com/juliensimon/aws



# Thank you!

Julien Simon

Principal Technical Evangelist, AI and Machine Learning, AWS

@julsimon