DAT200 – Applied Machine Learning I

Chapter 1 in “Python Machine Learning“ book

Giving Computers the Ability to Learn from Data

Topics of Ch. 01 – Giving Computers the Ability to Learn from Data

- General **concepts** of machine learning
- Three **types** of machine learning
- **Basic terminology**
- **Building blocks** for machine learning systems
- **Setting up Python** for data analysis and machine learning

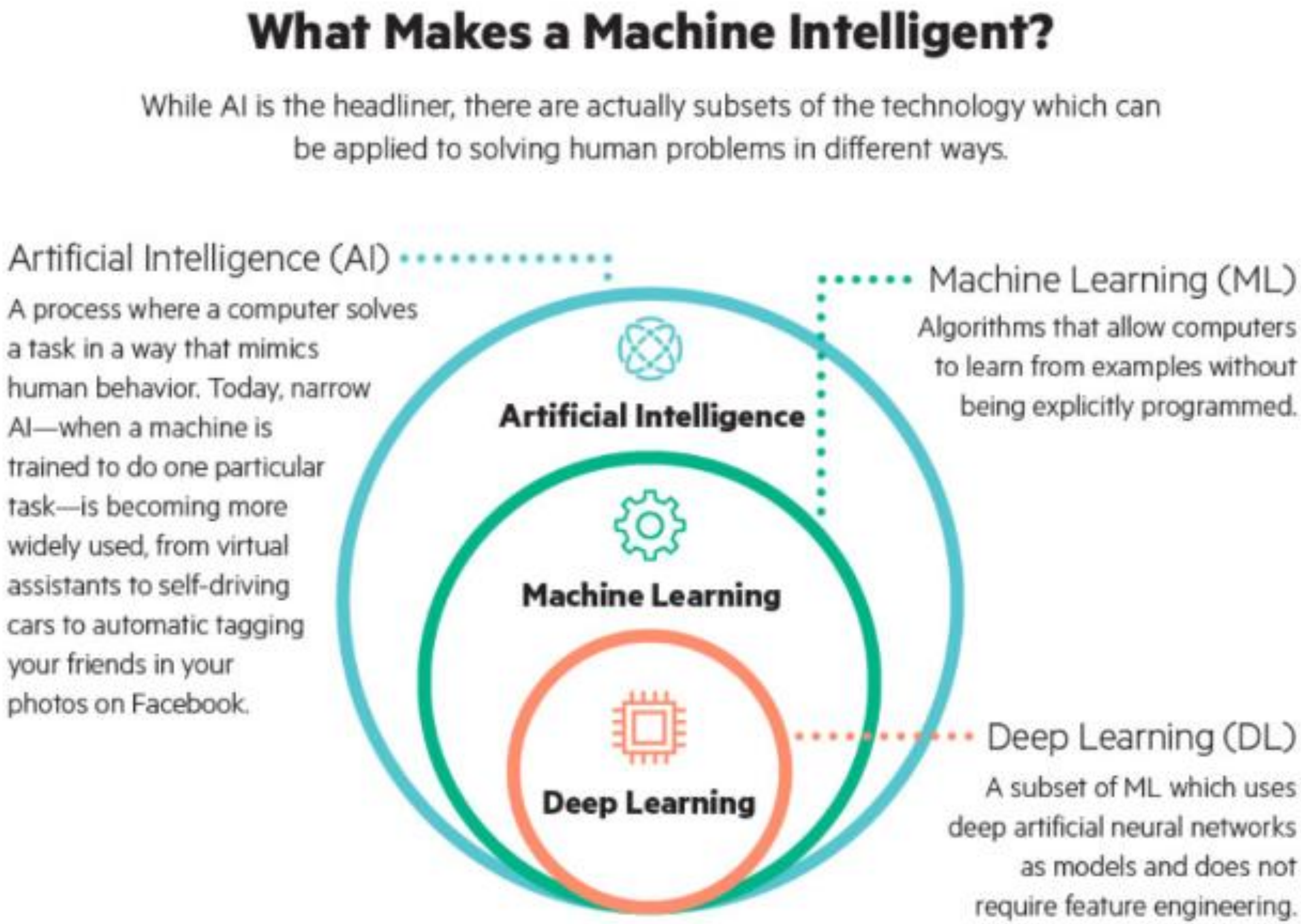
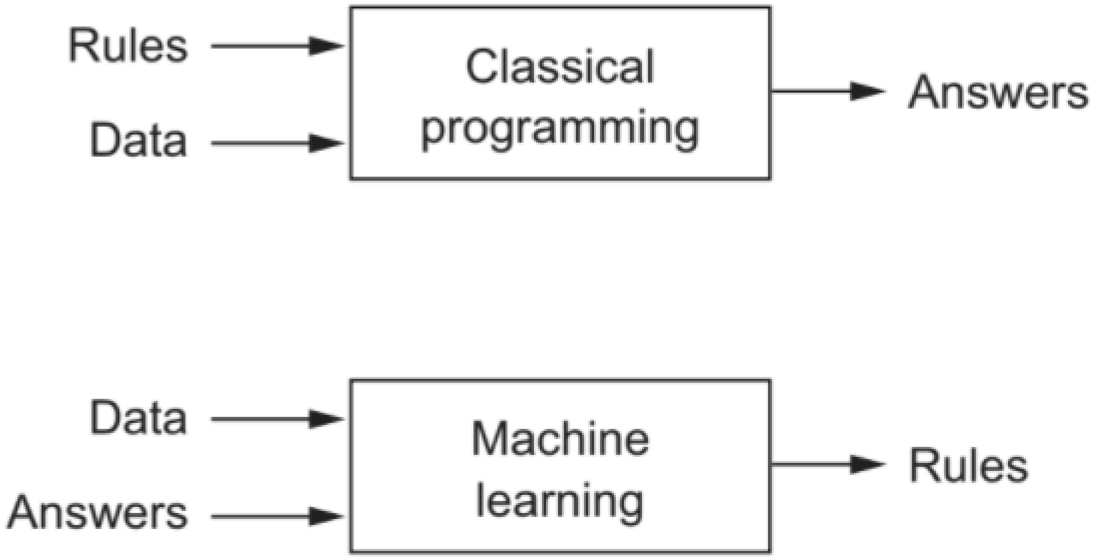


Image: https://community.hpe.com/t5/Behind-the-scenes-Labs/Labs-Deep-Learning-Cookbook-headlines-the-launch-of-HPE-s-AI/ba-p/6981300#.WmS9oqjiWUk

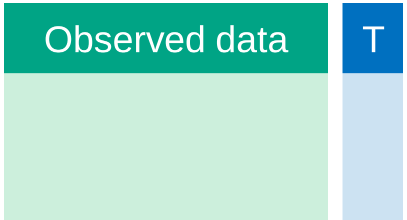
New programming paradigm



Three different types of machine learning

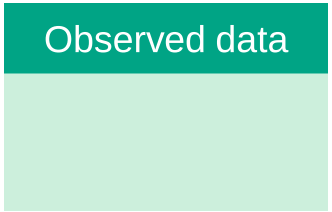
Supervised Learning
- Labeled data
- Direct feedback
- Predict outcome/future

Learning from labelled data



Unsupervised Learning
- No labels/targets
- No feedback
- Find hidden structure in data

Discover structure in unlabelled data



Reinforcement Learning
- Decision process
- Reward system
- Learn series of actions

Learning by "doing" with delayed reward

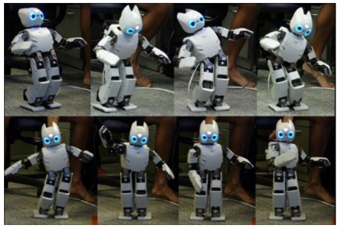


Image: S. Raschka, V. Mirjalili. 2017. «Python Machine Learning», Chapter 1, page 2

Multivariate data – basic terminology

Instances (samples, observations, objects**)**

Target (class, category**)**

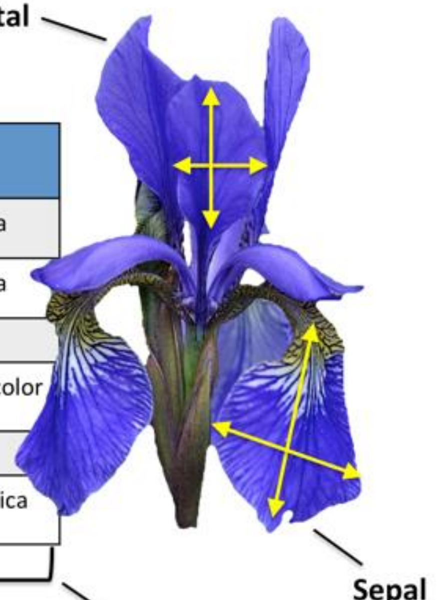
Features (attributes, variables, dimensions**)**

Patient	Diagnose	Age	Sex	Comorbidity	Prev. admissions
#1	I21	78	male	3	4

Basic terminology and notations

Samples
(instances, observations)

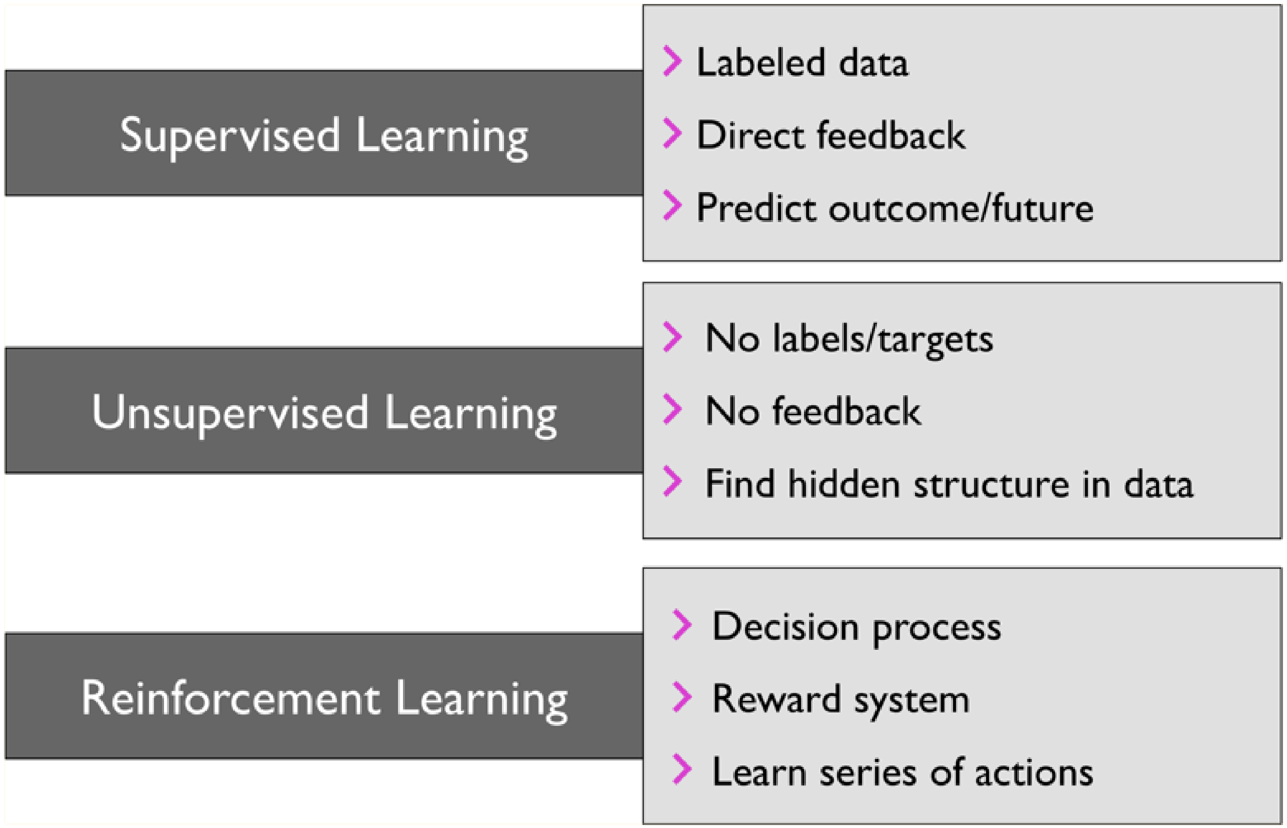
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica



Class labels
(targets)

Features
(attributes, measurements, dimensions)

Three different types of machine learning



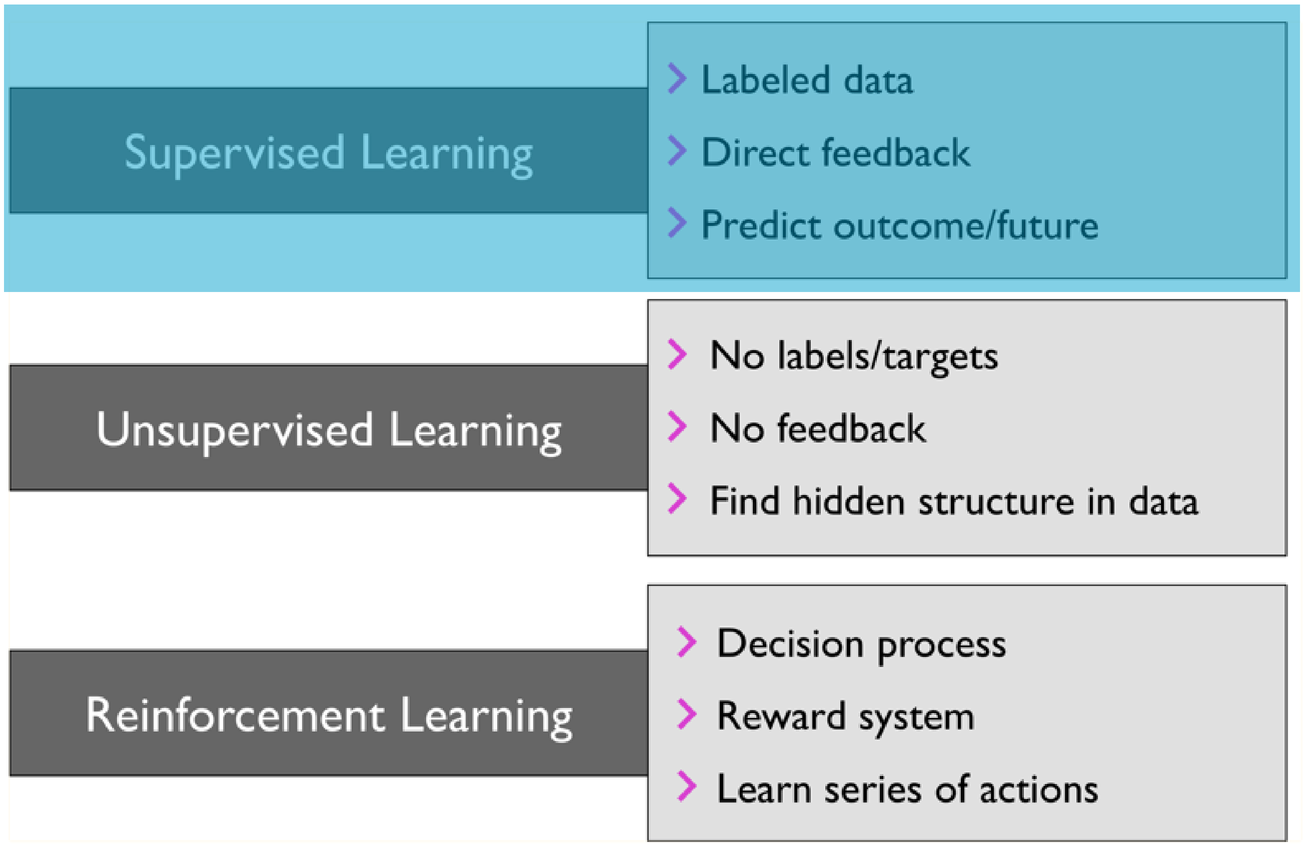
Ch. 02 – 06: classification
Ch. 05: dim. reduction
Ch. 10: regression

Ch. 05: dim. reduction
Ch. 11: clustering

Not part of DAT200

Image: S. Raschka, V. Mirjalili. 2017. «Python Machine Learning», Chapter 1, page 2

Three different types of machine learning



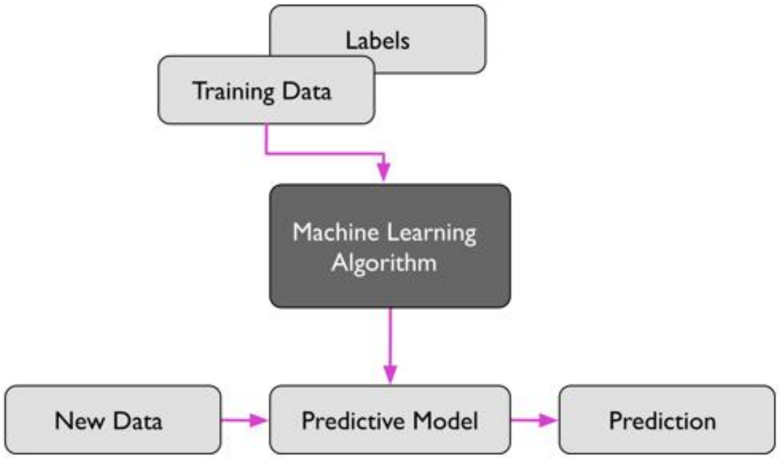
Ch. 02 – 06: classification
Ch. 05: dim. reduction
Ch. 10: regression

Ch. 05: dim. reduction
Ch. 11: clustering

Not part of DAT200

Image: S. Raschka, V. Mirjalili. 2017. «Python Machine Learning», Chapter 1, page 2

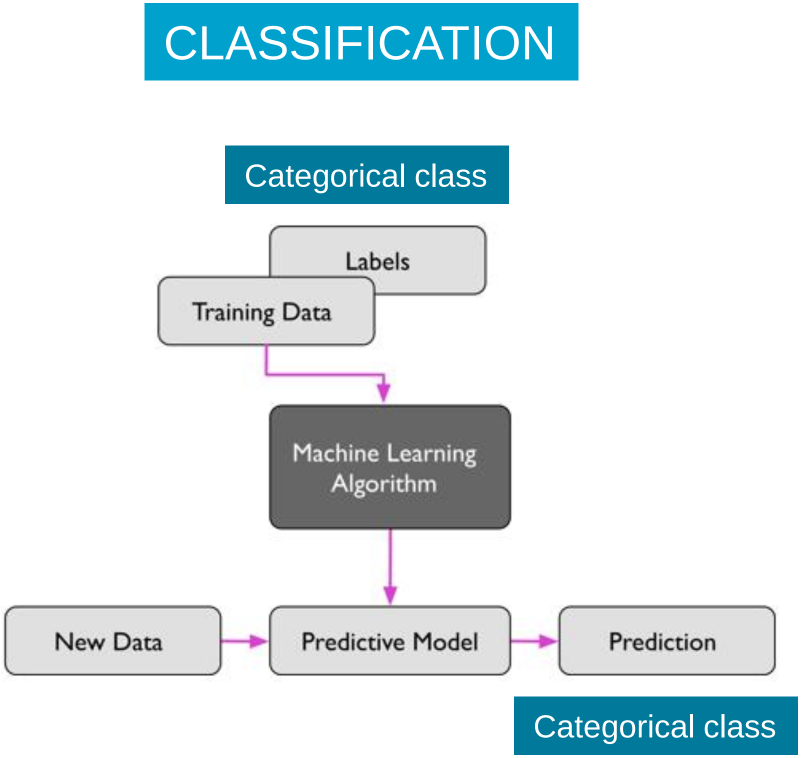
Supervised learning - Making predictions about the future



- Main goal in supervised learning
  - **Learn model** from labeled training data
  - Use model to **make predictions** on unseen or future data
  - Term «Supervised»: refers to a set of samples where desired **output signals** (labels) **are already known**

Image: S. Raschka, V. Mirjalili. 2017. «Python Machine Learning», Chapter 1, page 3

Supervised learning –
Classification for predicting class labels

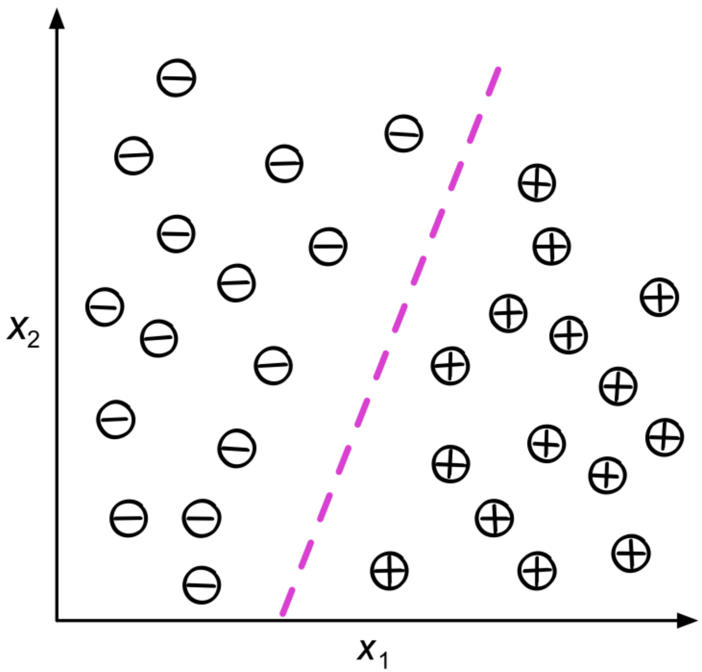


- Classification is **subcategory** of **supervised learning**
- Goal: predict categorical class labels
  - For **new instances**
  - Based on **past observations**
- Provided data
  - Training data: explanatory variables
  - Labels: **Class** labels
- Class labels
  - discrete, unordered values
  - «Group memberships» of instances

Image: S. Raschka, V. Mirjalili. 2017. «Python Machine Learning», Chapter 1, page 3

Supervised learning –
Classification for predicting class labels

Binary classification task



- 30 training samples
  - 15 training samples labeled as negative class (minus signs)
  - 15 training samples labeled as positive class (plus signs)
- Two-dimensional dataset (x_1 and x_2)
- Supervised machine learning algorithms learns rule (decision boundary)
- Classify new data into each of those two classes, given values x_1 and x_2

- Example of binary classfication task
  - Distinguish between spam and non-spam emails

Supervised learning - multiclass classification example

Patient admission to hospital

Input

Observed data

- Diagnosis
- Age
- Sex
- Comorbidity
- Previous admissions
- Blood pressure
- etc.

Classifier

Output

Class or **category label**

- Category 1: High risk patient
- Category 2: Medium risk patient
- Category 3: Low risk patient

Treatment category 1

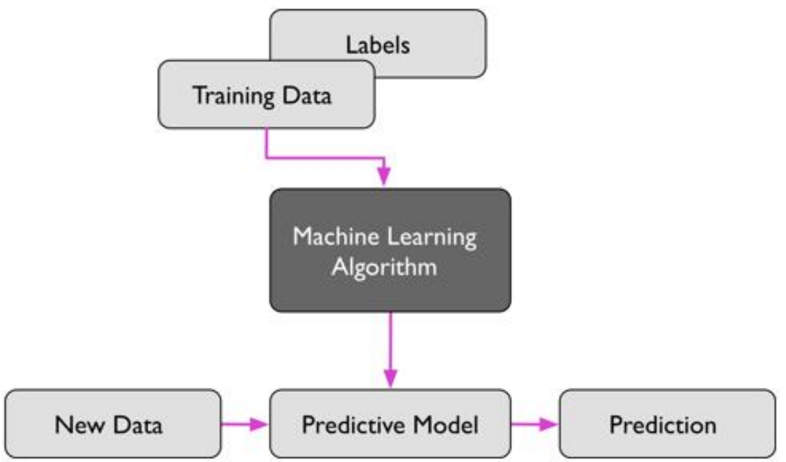
Treatment category 2

Treatment category 3

Supervised learning –
Regression for predicting continuous outcomes

REGRESSION

Continuous outcome

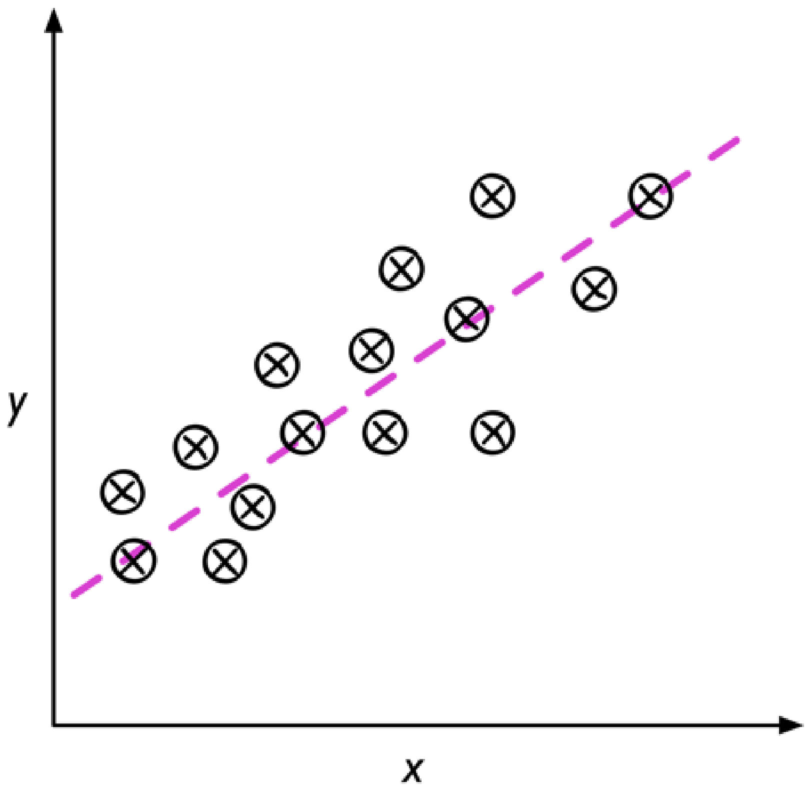


Continuous outcome

- Regression analysis is a **subcategory** of supervised learning
- Goal: predict continuous outcome
  - For **new instances**
  - Based on **past observations**
- Provided data
  - Training data: explanatory variables
  - Labels: **Continuous response variable** (outcome or target)

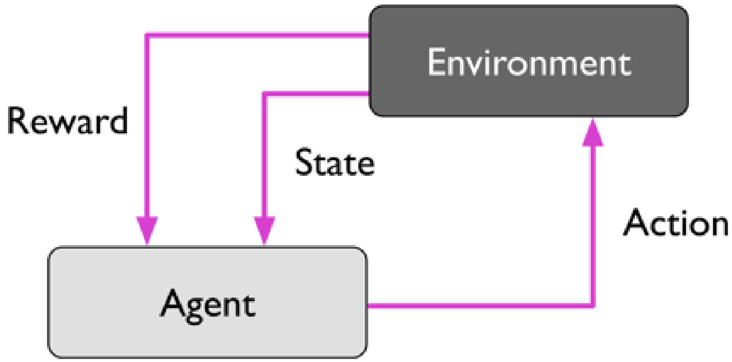
Image: S. Raschka, V. Mirjalili. 2017. «Python Machine Learning», Chapter 1, page 3

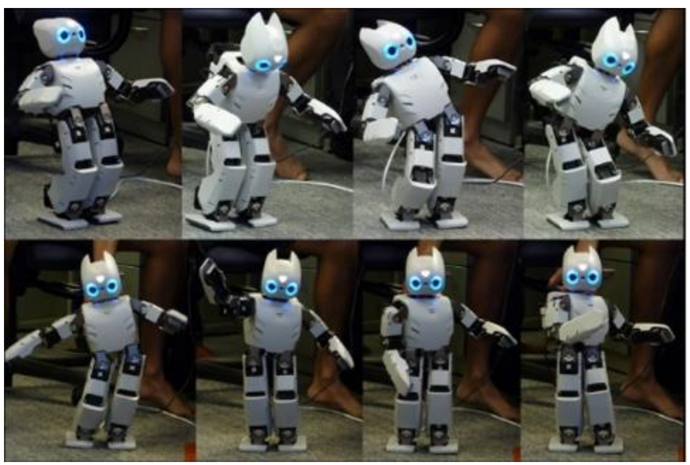
Supervised learning –
Regression for predicting continuous outcomes



- Predictor variable x
- Response variable y
- Fit model – a line that **minimises distance** between sample points and and fitted line
- Use intercept and slope learned from data for **predicting** outcoume of new data

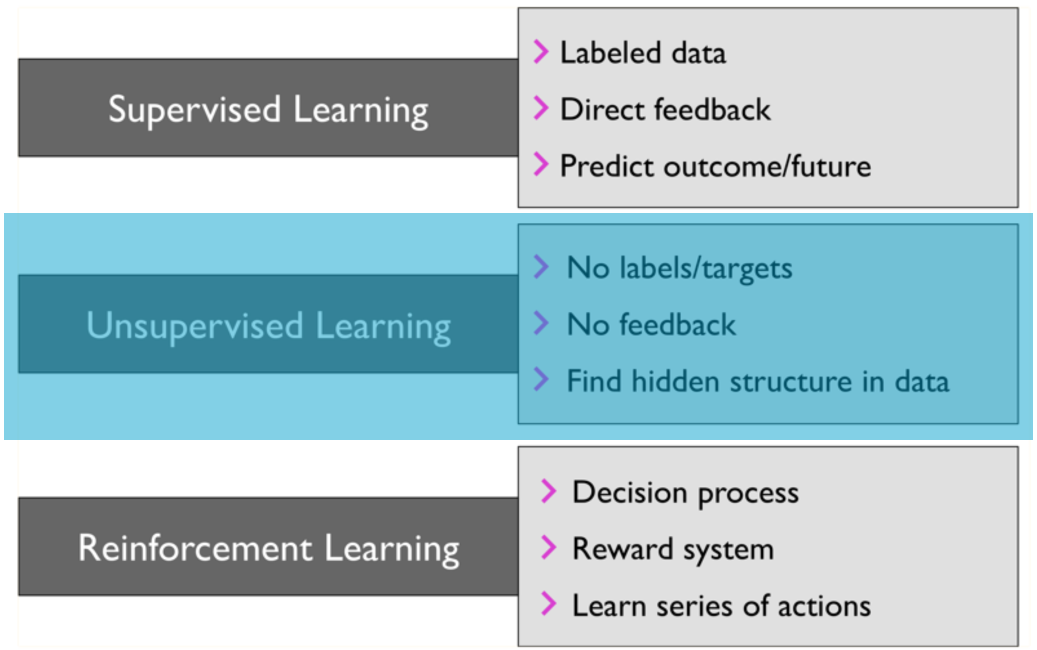
Reinforcement learning -
solving interactive problems





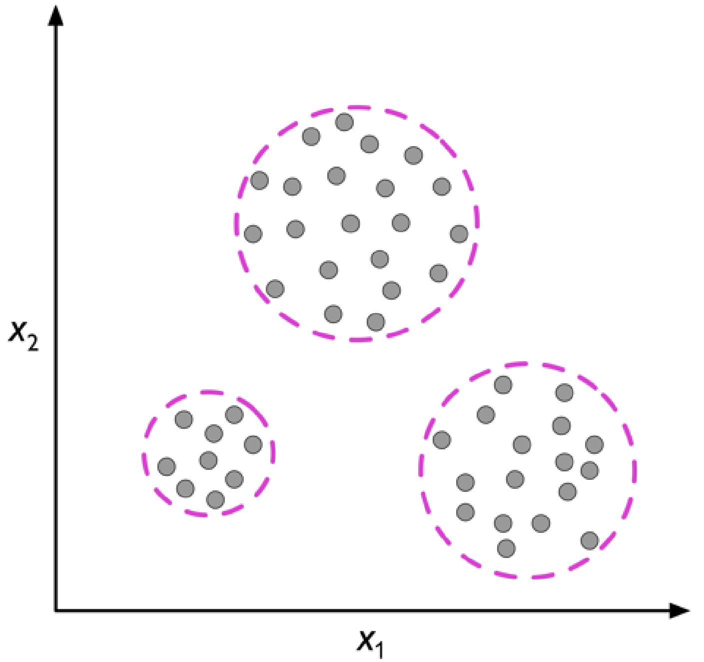
- Goal: develop a system (**agent**) that **improves performance** based on **interaction** with **environment**
- Agent processes information on **current state** of environment
- Define a **measure** of reward for **particular actions** by the agent
- State can be associated with **postive** or **negative** reward
- A reward can be defined as **accomplishing** an **overall goal**
- Concerned with learning a **series of steps** my **maximising a reward** based on
  - Immediate feedback
  - Delayed feedback

Unsupervised learning – Finding hidden structures with clustering



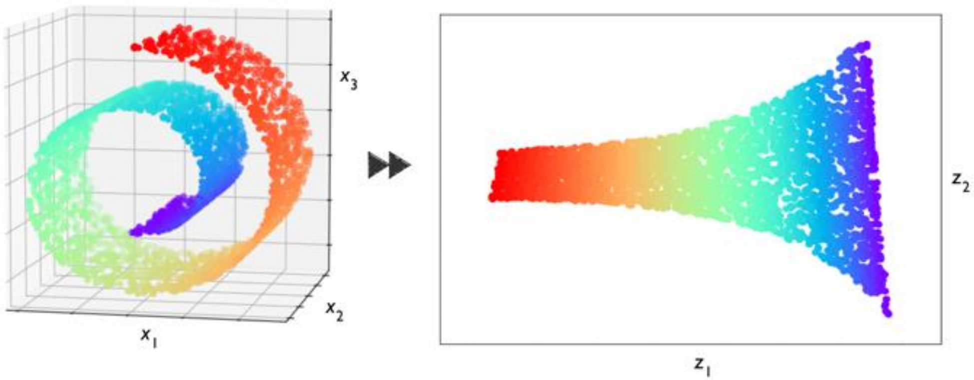
- **Unlabeled** data of **unknown** structure
- **Explore** structure of data
  - extract **meaningful** information
  - do so **without guidance** of known outcome variable or reward function
- Subfields of unsupervised learning
  - Clustering (Ch. 11)
  - Dimensionality reduction (Ch. 5)

Unsupervised learning –
Finding subgroups with clustering



- Clustering is an **exploratory** data analysis technique
- Clustering organises information into **meaningful distinct subgroups** based on their features
- Clusters:
  - objects **within** cluster share a certain **degree of similiarity**
  - objects from one cluster are **dissimilar** to objects from other clusters

Unsupervised learning –
Dimensionality reduction for data compression



- Data are often **high-dimensional** (many features / variables)
- High dimensionality may be challanging with **limited storage space**
- High dimensionality may **hamper computational performance** of machine learning algorithms
- Unsupervised dimensionality reduction a **common approach** in **feature processing**
  - **Remove noise** from data (noise can degrade predictive peformance)
  - **Compress data** into smaller dimensional subspace while retaining most of relevant information
- Unsupervised dimensionality reduction useful for **visualisation**

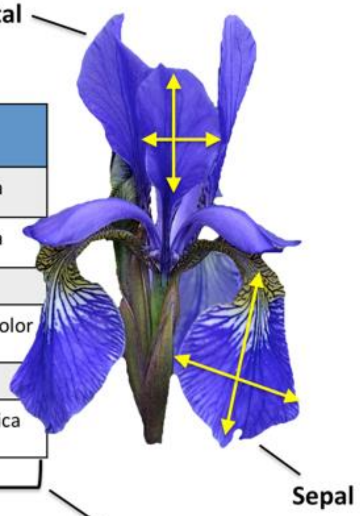
Mathematical notations

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)



Iris data set

150 **instances** / objects / rows
4 **features** / variables / columns
3 **classes** / targets

- https://archive.ics.uci.edu/ml/datasets/Iris

UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Mathematical notations

- Use matrix and vector notation to refer to data
- Iris data set can be written as a 150×4 matrix $\boldsymbol{X} \in \mathbb{R}^{\mathbf{150 \times 4}}$

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

- Index i: refers to the ***i***th training sample
- Index j: refers to the ***j***th dimension of the training data set
- Lowercase bold-face letters refer to **vectors** $\left(\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n} \times 1}\right)$
- Uppercase bold-face letters refer to **matrices** $(\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{m}})$
- Single element n in a vector $x^{(n)}$
- Single element n in a matrix $x_{(m)}^{(n)}$

Mathematical notations

- **Row** in iris data matrix X
  - Represents **one** flower instance
  - Can be written as a four-dimensional row-vector $\boldsymbol{x}^{(i)} \in \mathbb{R}^{\mathbf{1\times4}}$

$$\boldsymbol{x}^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} \end{bmatrix}$$

- **Column** in iris data matrix X
  - Represents **one** feature
  - Can be written as a 150-dimensional column vector $\boldsymbol{x}_j \in \mathbb{R}^{\mathbf{150\times1}}$

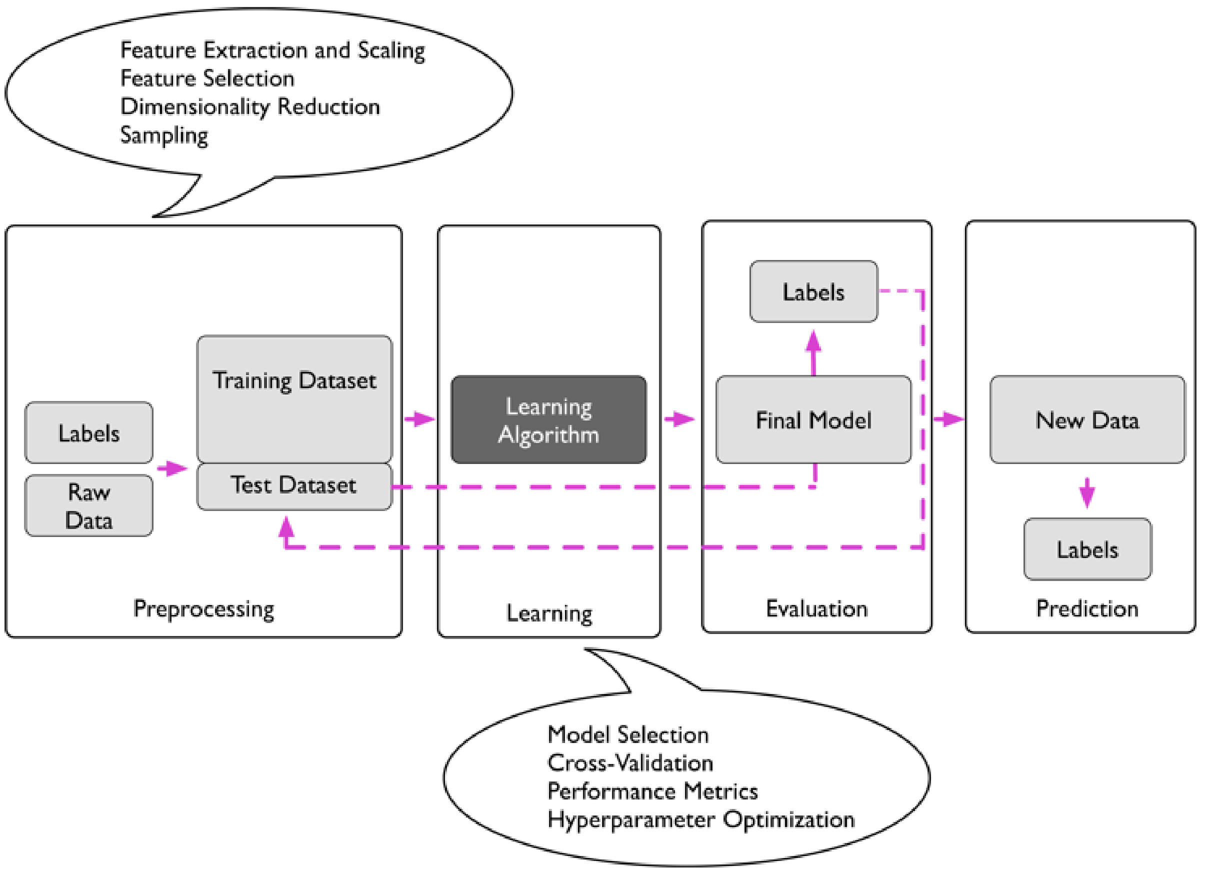
$$\boldsymbol{x}_j = \begin{bmatrix} x_j{}^{(1)} \\ x_j{}^{(2)} \\ \vdots \\ x_j{}^{(150)} \end{bmatrix}$$

Mathematical notations

- **Target variable y**
  - Contains either classes (classification) or continuous outcomes (regression)
  - In iris data set target variable y contains classes
    - Setosa
    - Versicolour
    - Virginica
  - Can be written as a 150-dimensional vector

$$\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(150)} \end{bmatrix} \left(y \in \{\text{Setosa, Versicolor, Virginica}\} \right)$$

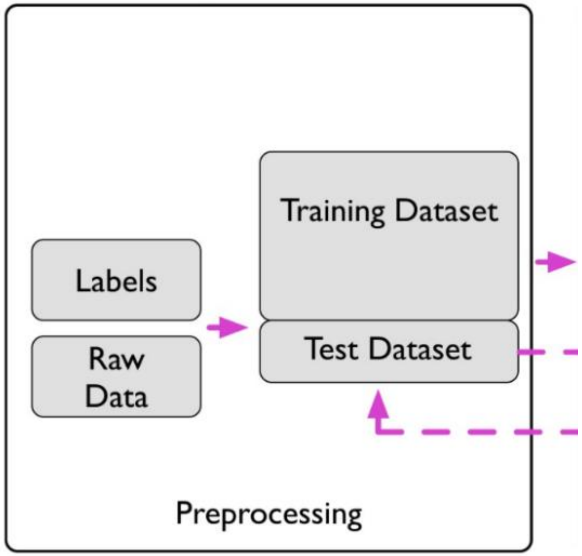
Roadmap for building machine learning systems



Typical workflow for machine learning in predictive modelling

Roadmap for building machine learning systems – Preprocessing

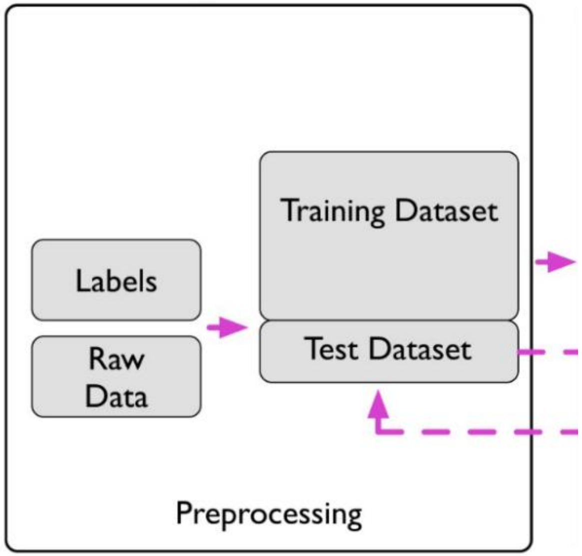
- Feature extraction and scaling
- Feature selection
- Dimensionality reduction
- Sampling



- Preprocessing of data is one of the most cruicial steps in any machine learning application
- Raw data often requires preprocessing to get it into wanted format
- Many ML algorithms require features to be on same scale for optimal performance
- Dimensionality reduction
    - Leaving out highly correlated variables that may be redundant
        - → Less storage space
        - → Shorter computation times
    - Leaving out irrelevant/noisy features → may improve predictive performance

Roadmap for building machine learning systems – Preprocessing

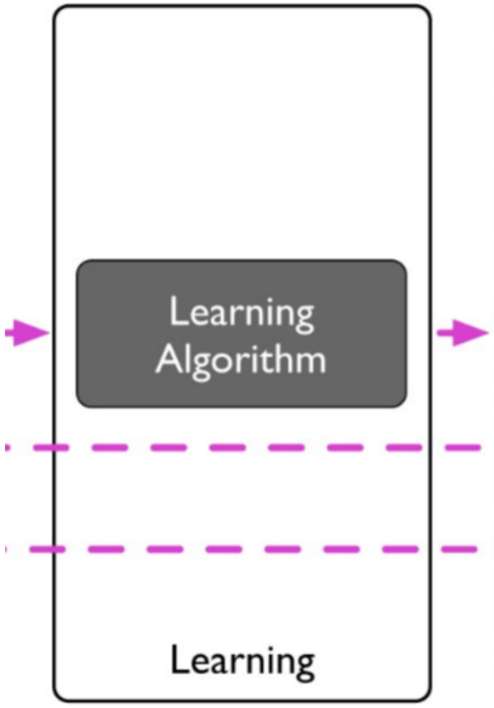
- Feature extraction and scaling
- Feature selection
- Dimensionality reduction
- Sampling



- Separation of data into training and test set
- Model needs to generalise well
  - Good performance on training data
  - AND good performance on test data
- Optimise ML model to achieve that
- Feature engineering
  - Obtain new features from raw data

Roadmap for building machine learning systems – Learning algorithm

- Model selection
- Cross-validation
- Performance metrics
- Hyperparamter optimisation



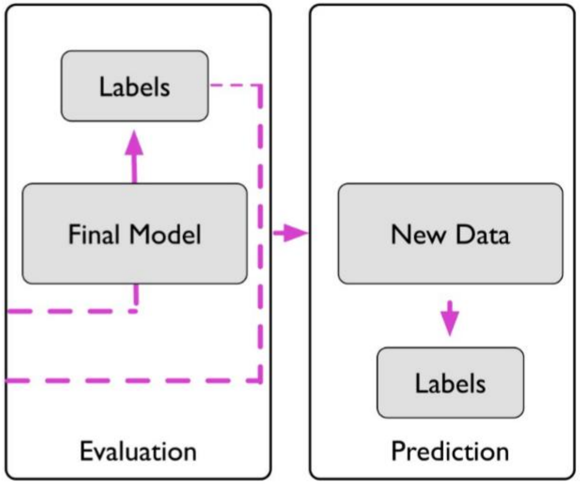
- Many different machine learning algorithms are available
- Each algorithm / method has strengths and weaknesses (inherent bias, see examples in next slide)
- Choose metric to measure performance (accuracy, AUC of ROC, etc.)
- Compare performance of several algorithms to find best performing model
- Use cross-validation for estimation of generalisation performance
- Hyperparameter optimisation for fine tuning performance of model (default values for learning alogrithms may not be optimal)

Algorithm	Type	Class	Restriction bias	Preference bias
K-Nearest Neighbors	Supervised	Instance based	Generally speaking, KNN is good for measuring distance-based approximations; it suffers from the curse of dimensionality	Prefers problems that are distance based
Naive Bayes	Supervised	Probabilistic	Works on problems where the inputs are independent from each other	Prefers problems where the probability will always be greater than zero for each class
Decision Trees/ Random Forests	Supervised	Tree	Becomes less useful on problems with low covariance	Prefers problems with categorical data
Support Vector Machines	Supervised	Decision boundary	Works where there is a definite distinction between two classifications	Prefers binary classification problems
Neural Networks	Supervised	Nonlinear functional approximation	Little restriction bias	Prefers binary inputs
Hidden Markov Models	Supervised/ Unsupervised	Markovian	Generally works well for system information where the Markov assumption holds	Prefers time-series data and memoryless information
Clustering	Unsupervised	Clustering	No restriction	Prefers data that is in groupings given some form of distance (Euclidean, Manhattan, or others)
Feature Selection	Unsupervised	Matrix factorization	No restrictions	Depending on algorithm, can prefer data with high mutual information
Feature Transformation	Unsupervised	Matrix factorization	Must be a nondegenerate matrix	Will work much better on matricies that don't have inversion issues
Bagging	Meta-heuristic	Meta-heuristic	Will work on just about anything	Prefers data that isn't highly variable

Restriction and preference biases of a set of common algorithms.

Table: M.Kirk. 2017. «Thoughtful Machine Learning», Chapter 2, page 17

Roadmap for building machine learning systems – Evaluation and prediction



- Estimate generalisation error with unseen test data
- Use model with satisfactory prediction performance for prediction of new future data
- All transformations of training data are applied to the test data – using the parameters acquired from transformation of training data

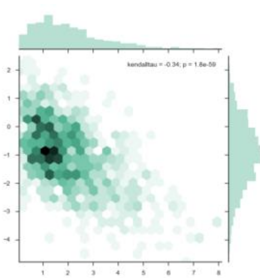
Python for machine learning

- Python has become one of the most popular languages for data science
- Many add-on packages for specific tasks
- We will use scikit-learn in DAT200
  - Popular among data scientists
  - Often preferred in “production” over other programming languages
  - Streamlined user interface – many tedious tasks are automated

Python for machine learning

- Other packages used in DAT200
  - Numpy
  - SciPy
  - Matplotlib
  - Pandas
  - Seaborn
  - Jupyter
  - Altair

seaborn

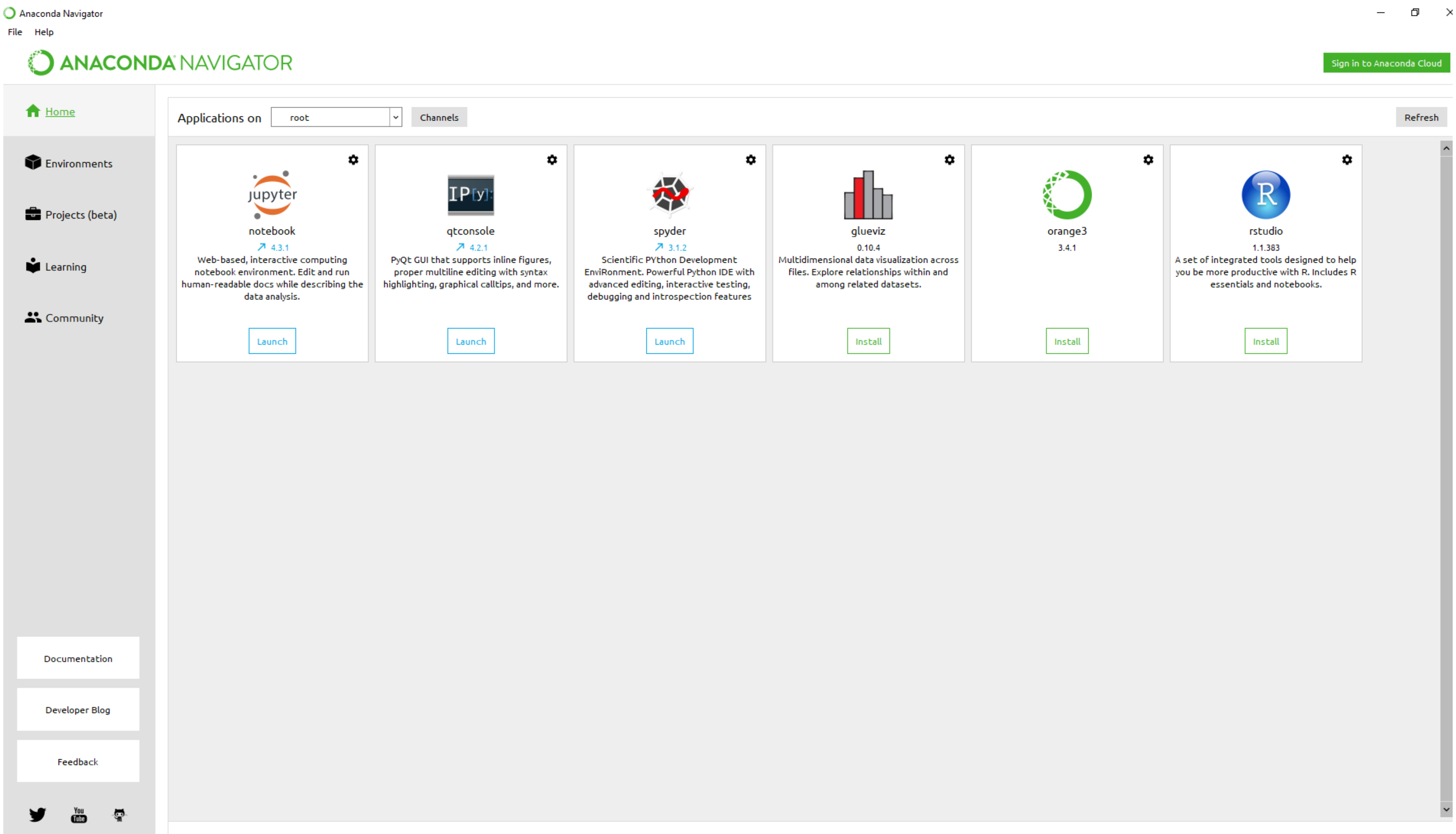


Altair

Anaconda Navigator

- Convenient working environment
  - Installing and updating various add-on package is straightforward
  - Availability of various useful coding tools
    - Spyder
    - Jupyter notebooks
    - Jupyterlab
    - R + Rstudio
    - Etc.
  - Independent working environments for projects
  - Availability of documentation
  - Links to various communities

Anaconda Navigator



Resources

- Python Machine Learning SE, Chapter 1, pages 1 – 16
  - Jupyter notebook: https://github.com/rasbt/python-machine-learning-book-2nd-edition/tree/master/code/ch01
- Anaconda: https://www.anaconda.com/
- scikit-learn: http://scikit-learn.org
- Other scikits: https://scikits.appspot.com/scikits
- CS229 Machine learning, Stanford: http://cs229.stanford.edu/
  - Video lectures: https://www.youtube.com/playlist?list=PLA89DCFA6ADACE599

NMBU