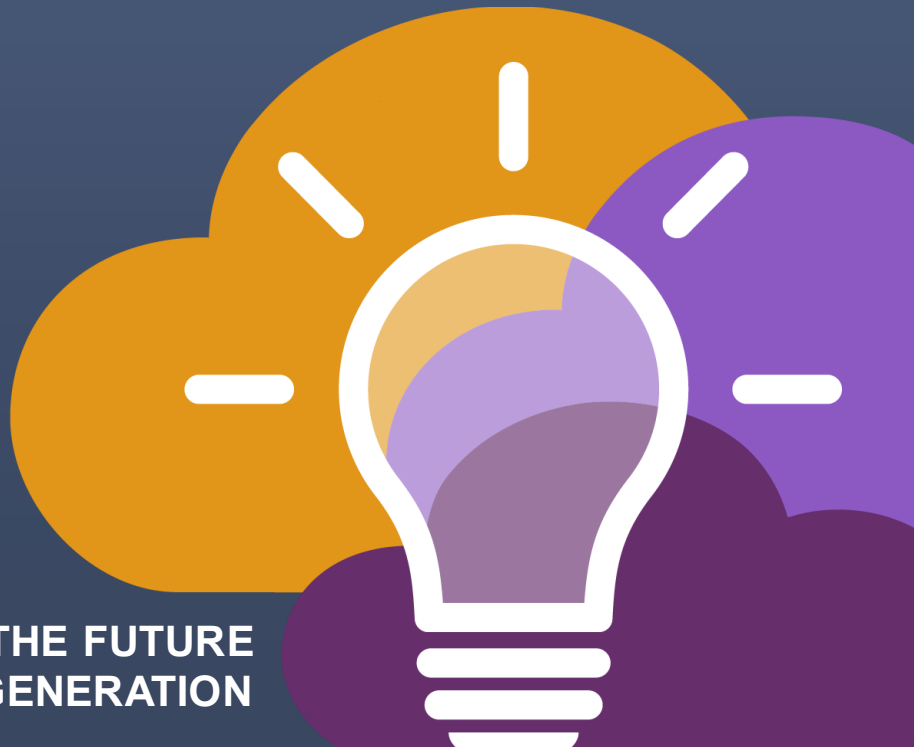
ScHARe

Computational strategies: AI and Machine Learning



BE A PART OF THE FUTURE
OF KNOWLEDGE GENERATION

Artificial Intelligence (AI)

AI is defined as:

*"machines that respond to stimulation **consistent with traditional responses from humans**, given the human capacity for contemplation, judgment, and intention."*

This definition emphasizes several qualities that separate AI from traditional computer software:

- **Intentionality**
- **Intelligence**
- **Adaptability**

AI-based computer systems **can learn from data, text, or images and make intentional and intelligent decisions** based on that analysis.

Many AI/ML projects are built using Python.

ScHARe fully supports the **Python libraries** most commonly used for AI tasks.

The role of AI

***AI is an outcome**—the ability of machines to perform tasks that typically require human-level intelligence*

perception

Describe and understand surroundings

notification

Provide alerts, reminders, etc.

suggestion

Build on past preferences and modify over time

automation

Follow routine steps to accomplish an objective

prediction

Forecast the likelihood of future events based on past events

prevention

Apply cognitive reckoning to identify potential threats

situational awareness

Summarize the current, and likely future, environment

Key Questions Answered							
	What's happening now?	What do I need to know?	What do you recommend?	What should I do?	What can I expect to happen?	What can/should I avoid?	What do I need to do now?

THE CURRENT ROLE OF AI:

Curator — Recommender — Orchestrator

NOT THE ROLE OF AI:

Critical Thinker — Decision Maker

Machine Learning (ML)

Machine learning is a subset of artificial intelligence (AI) that involves training algorithms to recognize patterns and make decisions based on data.

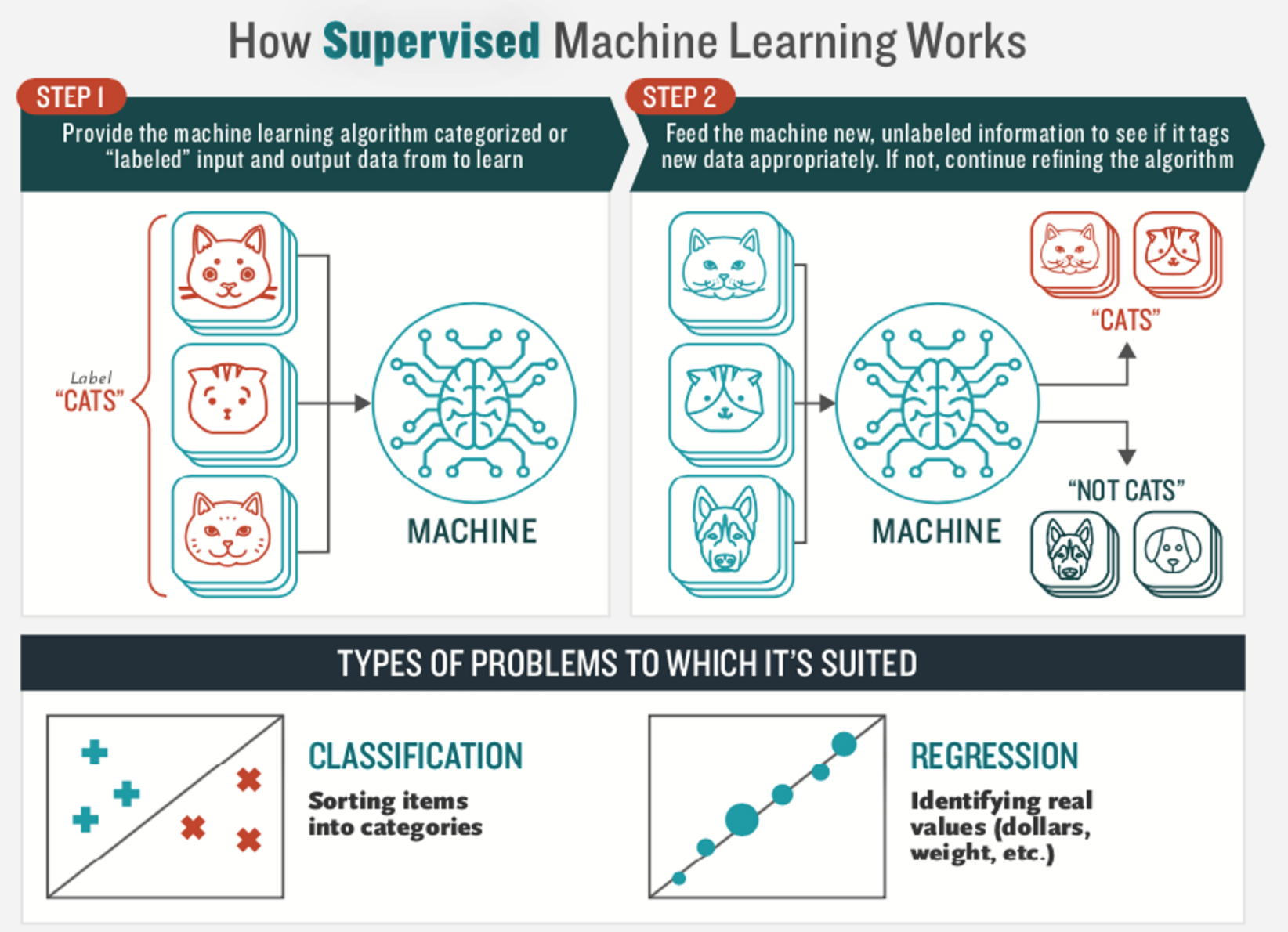
It represents **a way to classify data/objects without detailed instruction**.

The algorithm learns in the process so that new objects can be identified using the learned info.

ML is “based on **algorithms that can learn from data** without relying on rules-based programming.”

Unlike **traditional programming, where explicit instructions are given**, machine learning models **learn from examples and improve their performance** over time.

Supervised Learning



Supervised Learning: regression

Supervised learning utilizes a dataset which includes both input features as well as the output class or target which are **labeled at the start of training**. Supervised ML algorithms subsequently train on the input data set to produce a model which will differentiate among the output labels.

Common Algorithms: Linear Regression, Support Vector Regression, Random Forest Regression, Decision Tree Regression, Ridge Regression, Support Vector Regression (SVR)

Machine-learning Prediction For Hospital Length Of Stay Using A French Medico-administrative Database

Objective: The objective of this study is to explore ML models that best predict Prolonged Hospital Length of Stay for patients based on clinical and demographic features.

Algorithm Used: Random Forest (RF), Neural Networks (NN), Gradient Boosting (GB), Decision Trees (CART), Logistic Regression (LR).

Dataset: 27 predictor variables including sociodemographic features (age, gender, state-funded medical assistance), disease category, patient origin (home or other hospital institution), hospitalization via emergency departments, destination after hospital discharge, and hospitalization via emergency departments in the previous 6 months.

www.ncbi.nlm.nih.gov/pmc/articles/PMC9707380

Supervised Learning: classification

In classification based supervised learning, the model learns to **map input data to specific categories (classes) by studying examples where the correct output is known**. This process involves creating a set of rules or a classifier that can predict the correct category for new, unseen data based on the patterns learned from the training examples.

Common Algorithms: Logistic Regression, Decision Trees, Random Forest, Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Naive Bayes, Neural Networks

Machine Learning-Based Prediction Model of Preterm Birth Using Electronic Health Record

Objective: Preterm birth (PTB) was one of the leading causes of neonatal death. Predicting PTB in the first trimester and second trimester will help improve pregnancy outcomes. The aim of this study is to propose a prediction model based on machine learning algorithms for PTB.

Algorithm Used: Six algorithms, including Naive Bayesian (NBM), support vector machine (SVM), random forest tree (RF), artificial neural networks (ANN), K-means, and logistic regression, were used to predict PTB.

Dataset: Demographic factors (i.e., age), physical examination, blood test, white blood cell count, and plateletcrit, urine test strip (urine pH, urine WBC, and glycosuria), and gynecological examination.

pubmed.ncbi.nlm.nih.gov/35463669/

Supervised Learning: regression & classification

Ensemble methods combine the predictions of several base estimators built with a given learning algorithm to improve generalizability and robustness over a single estimator. They are often used for classification and regression tasks, where they can reduce bias and variance to improve model accuracy

Common Algorithms: Ensemble boosting methods (ADA Boost, Gradient Boosting) Ensemble Bagging Methods (Random Forest), Artificial Neural Networks

A Study Of Generalizability Of Recurrent Neural Network-based Predictive Models For Heart Failure Onset Risk Using A Large And Heterogeneous EHR Data Set

Objective: Recurrent neural networks (RNNs) have been applied in predicting disease onset risks with Electronic Health Record (EHR) data to test generalizability and transferability of existing models and its applicability to different patient populations across hospitals.

Algorithm Used: Recurrent Neural Networks

Dataset: Number of Visits, Diagnoses, Medication, Surgery, Gender, Race, Age

pubmed.ncbi.nlm.nih.gov/29908902/

Supervised Learning: Python libraries

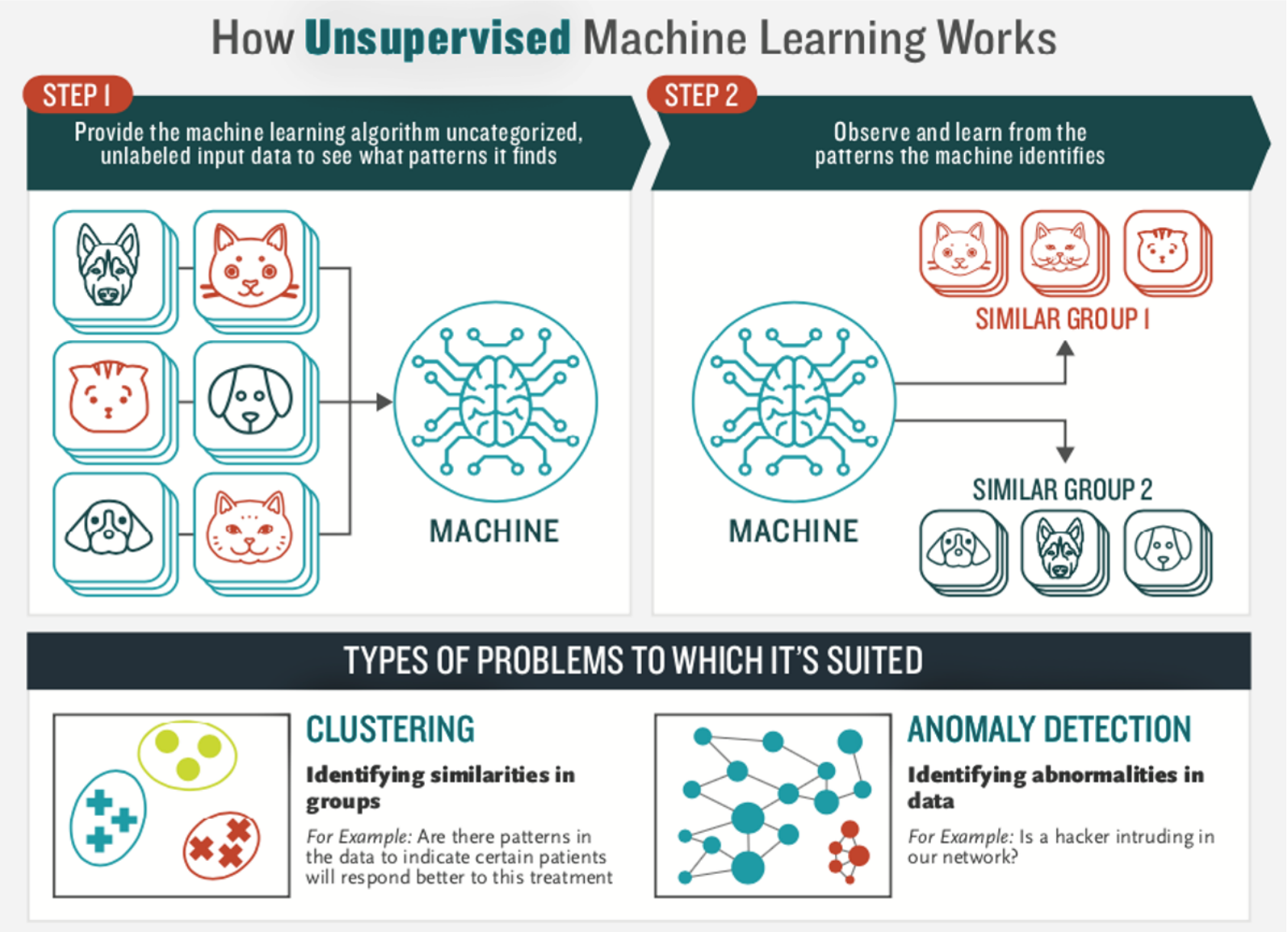
Python offers a range of data science libraries suitable for supervised learning:

1. **scikit-learn**:
    1. Comprehensive library for machine learning with tools for classification, regression, and clustering.
    2. Provides efficient implementations of algorithms like SVM, decision trees, random forests, and more.
2. **TensorFlow**:
    1. Open-source library for deep learning developed by Google.
    2. Suitable for building and training neural networks for tasks like image and speech recognition.
3. **Keras**:
    1. High-level neural networks API, running on top of TensorFlow.
    2. Simplifies the creation of deep learning models with an intuitive interface.
4. **PyTorch**:
    1. Open-source deep learning library developed by Facebook's AI Research lab.
    2. Known for its dynamic computation graph and ease of use in research and production.
5. **XGBoost**:
    1. Optimized gradient boosting library designed for speed and performance.
    2. Effective for regression and classification problems, often used in machine learning competitions.

Supervised Learning: Python libraries

6. **LightGBM**:
    1. Gradient boosting framework that uses tree-based learning algorithms.
    2. Known for its efficiency and scalability, suitable for large datasets.
7. **CatBoost**:
    1. Gradient boosting library that handles categorical features well.
    2. Developed by Yandex, it is user-friendly and powerful for classification and regression tasks.
8. **pandas**:
    1. Essential for data manipulation and analysis.
    2. Provides data structures like DataFrame, making it easier to clean and preprocess data for supervised learning.
9. **NumPy**:
    1. Fundamental package for numerical computing.
    2. Provides support for arrays, matrices, and many mathematical functions, often used for preprocessing.
10. **Matplotlib** and **Seaborn**:
    1. Visualization libraries useful for exploring and understanding data.
    2. Helps in identifying patterns and preparing data for modeling.

Unsupervised Learning



Unsupervised Learning: Classification

Unsupervised learning with classification involves **grouping similar data points together based on their features without using labeled outcomes**. The goal is to find inherent structures or patterns in the data.

Unsupervised ML **can complement supervised ML approaches, since it can be used to initially determine the most critical features** prior to supervised ML approaches which will build models to discriminate among the classes of interest.

Common Algorithms: K-Means Clustering, Hierarchical Clustering, Gaussian Mixture Models (GMM)

Using Unsupervised Learning To Identify Clinical Subtypes Of Alzheimer's Disease In Electronic Health Records

Objective: Primary care electronic health records from the CALIBER resource were used to identify and characterize clinically-meaningful clusters of patients using unsupervised learning approaches.

Algorithm Used: MCA and K-Means Clustering

Dataset: Symptoms, comorbidities and demographic and lifestyle factors including age of onset, gender, drinking status and smoking status.

pubmed.ncbi.nlm.nih.gov/32570434/

Unsupervised Learning: dimension reduction

Dimensionality reduction in unsupervised learning **involves reducing the number of random variables under consideration, making the dataset easier to explore** and visualize while preserving as much variance as possible.

Common Algorithms: Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), Linear Discriminant Analysis (LDA), Autoencoders, Similarity Network Fusion (SNF)

A Fusion Framework To Extract Typical Treatment Patterns From Electronic Medical Records

Objective: Reduce the dimensionality of Electronic Medical Records (EMRs) contain temporal and heterogeneous doctor order information to identify “right patient”, “right drug”, “right dose”, “right route”, and “right time” from doctor order information

Algorithm Used: Multi-view similarity Network Fusion (SNF) method

Dataset: The EMR data included five types of information including demographic information, laboratory indicators, diagnostic information, doctor orders, and treatment outcome.

www.sciencedirect.com/science/article/abs/pii/S0933365718304184

Unsupervised Learning: Python libraries

Python offers several data science libraries that are well-suited for unsupervised learning:

1. **scikit-learn**:
    1. Provides a wide range of tools for clustering (e.g., K-means, DBSCAN, hierarchical clustering) and dimensionality reduction (e.g., PCA, t-SNE).
    2. Comprehensive and user-friendly, suitable for many unsupervised learning tasks.
2. **TensorFlow**:
    1. Deep learning library that can be used for advanced unsupervised learning techniques, such as autoencoders and generative models.
    2. Offers flexibility for building custom unsupervised learning algorithms.
3. **PyTorch**:
    1. Similar to TensorFlow, used for building and training deep learning models, including those for unsupervised learning.
    2. Known for its dynamic computation graph, making it easier to experiment with new models.
4. **hdbscan**:
    1. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is a clustering algorithm that can find clusters of varying densities.
    2. Effective for identifying complex cluster structures in data.

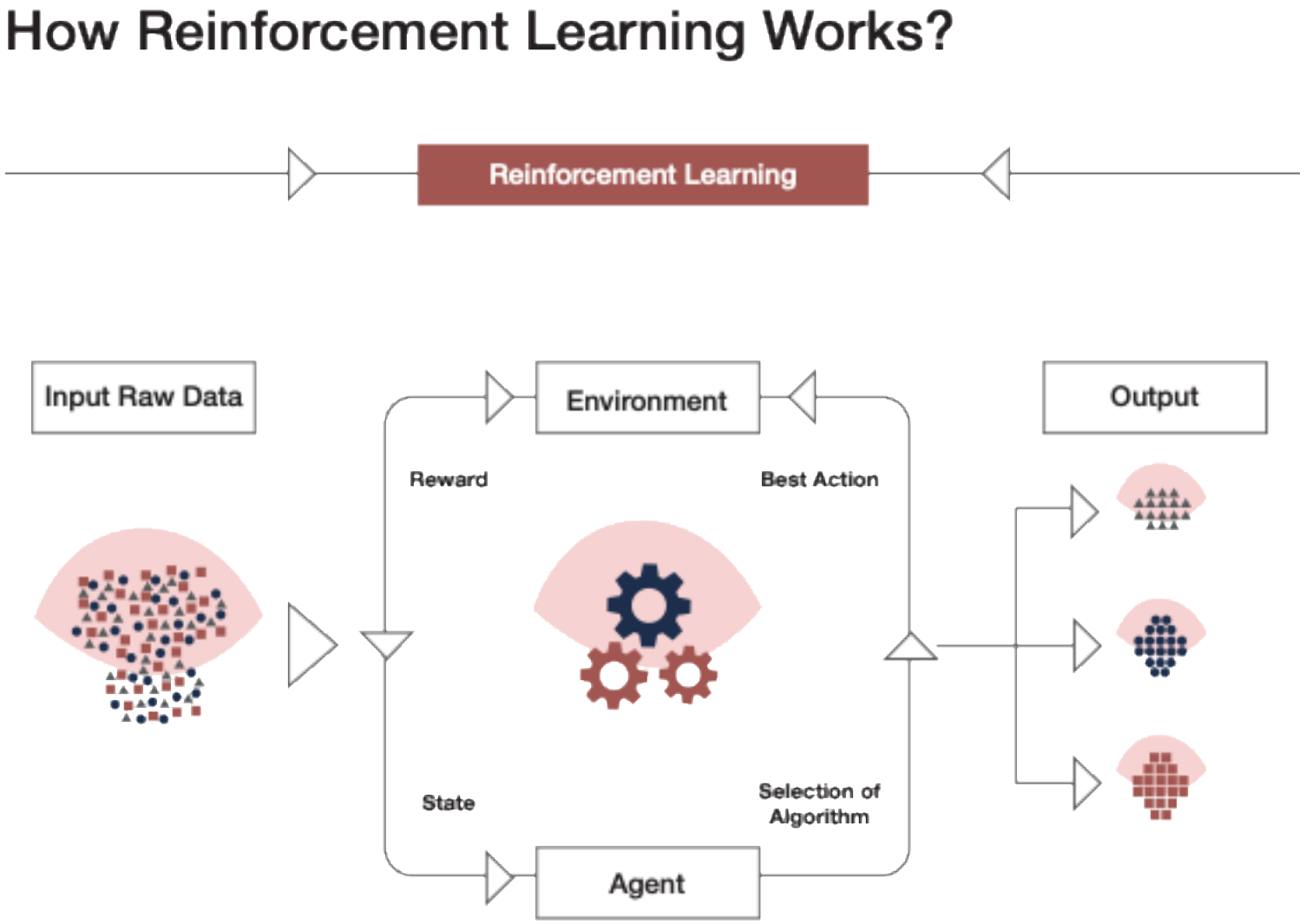
Unsupervised Learning: Python libraries

5. **OpenCV**:
    1. Primarily used for computer vision tasks, OpenCV also offers tools for unsupervised learning like K-means clustering.
    2. Useful for image processing and pattern recognition tasks.
6. **Yellowbrick**:
    1. A visualization library that extends scikit-learn and provides visual diagnostic tools for model selection, including tools for unsupervised learning.
    2. Helps in visualizing the performance of clustering and dimensionality reduction techniques.
7. **SciPy**:
    1. A scientific computing library that includes clustering and hierarchical clustering algorithms.
    2. Often used in conjunction with NumPy for numerical operations.
8. **NMF (Non-Negative Matrix Factorization)**:
    1. Used for dimensionality reduction and feature extraction.
    2. Available through libraries like scikit-learn and specialized packages.
9. **UMAP (Uniform Manifold Approximation and Projection)**:
    1. A dimensionality reduction technique that is particularly good for visualizing high-dimensional data.
    2. Often used as an alternative to t-SNE.

Unsupervised Learning: Python libraries

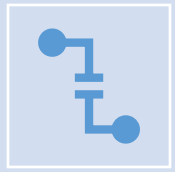
10. **Gensim**:
    1. A library for topic modeling and document similarity analysis, particularly useful for natural language processing (NLP) tasks.
    2. Implements algorithms like Latent Dirichlet Allocation (LDA) for discovering topics in text data.
11. **PyCaret**:
    1. An open-source, low-code machine learning library that simplifies the end-to-end machine learning process.
    2. Includes modules for clustering and anomaly detection.

Reinforcement Learning



Reinforcement Learning

Reinforcement Learning (RL) **involves training an agent to make a sequence of decisions by rewarding it for good actions and penalizing it for bad ones.** The agent learns to maximize cumulative rewards through trial and error interactions with the environment.



Common Algorithms: Q-Learning, Deep Q-Networks (DQN), Policy Gradient Methods, Actor-Critic Methods

A Value-based Deep Reinforcement Learning Model With Human Expertise In Optimal Treatment Of Sepsis

Objective: Develop personalized treatment strategies for sepsis patients using RL to maximize patient outcomes. The RL model learns to suggest personalized treatment plans that can improve survival rates and reduce recovery times for sepsis patients.

Algorithm Used: Deep Q-Network (DQN).

Dataset: EHR data from sepsis patients, including treatment actions, physiological measurements, and outcomes.

www.nature.com/articles/s41746-023-00755-5

Reinforcement Learning: Python libraries

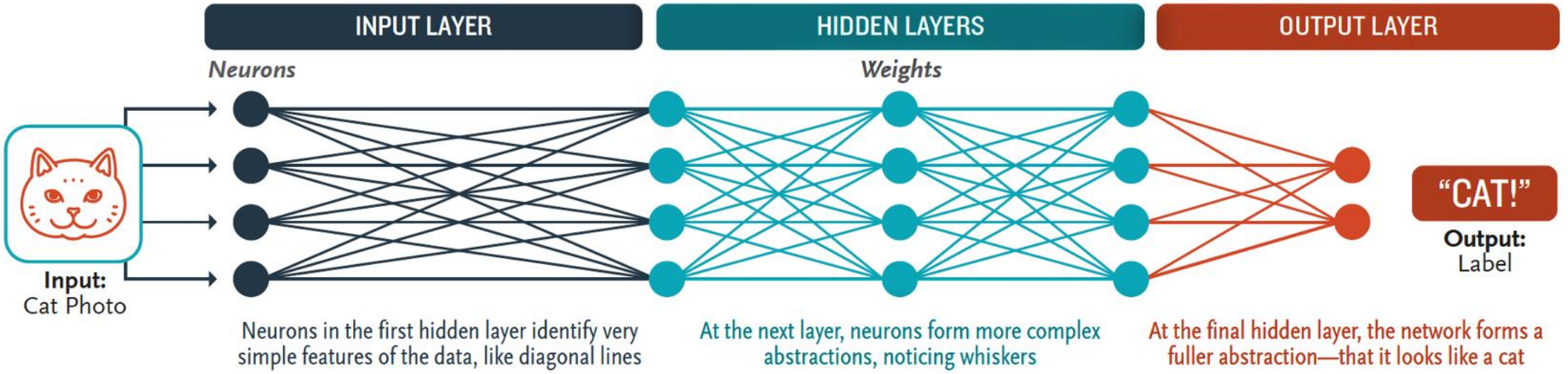
Python offers several libraries that are well-suited for reinforcement learning:

1. **OpenAI Gym**:
    1. A toolkit for developing and comparing RL algorithms.
    2. Provides a variety of environments (e.g., classic control tasks, Atari games) to test and benchmark RL algorithms.
2. **Stable Baselines3**:
    1. A set of reliable implementations of reinforcement learning algorithms in PyTorch.
    2. Built on top of OpenAI Baselines, it offers implementations of algorithms like PPO, A2C, DDPG, and more.
3. **RLlib**:
    1. Part of the Ray framework, RLlib is a scalable reinforcement learning library.
    2. Supports a wide range of RL algorithms and is designed for both single-machine and distributed training.
4. **Keras-RL2**:
    1. Builds on Keras and TensorFlow, providing simple and easy-to-use RL algorithms.
    2. Supports a variety of RL algorithms, making it easy to experiment and integrate with Keras models.

Reinforcement Learning: Python libraries

5. **TF-Agents**:
    1. A library for reinforcement learning in TensorFlow.
    2. Provides well-documented components for building, training, and evaluating RL agents.
6. **Baselines**:
    1. Originally developed by OpenAI, it offers high-quality implementations of standard RL algorithms.
    2. Useful for researchers and practitioners to benchmark new algorithms against established ones.
7. **Dopamine**:
    1. A research framework for fast prototyping of RL algorithms, particularly focused on simplicity and flexibility.
    2. Developed by Google Research, it provides implementations of several baseline algorithms.
8. **Coach**:
    1. An RL research framework by Intel AI Lab.
    2. Provides a collection of RL algorithms and environments with a focus on usability and performance.
9. **Horizon**:
    1. An open-source RL platform developed by Facebook.
    2. Designed for production use cases, particularly for large-scale applications.
10. **Tianshou**:
    1. A reinforcement learning library based on PyTorch.
    2. Designed to be efficient, flexible, and easily extensible.

Deep Learning



Deep Learning

Deep Learning is a subset of machine learning that uses **neural networks with many layers (deep neural networks) to model complex patterns in data**. It can be used for identifying objects within images, transcribing spoken language into text, analyzing medical scans for diagnostics, and understanding and generating human language.

Common Algorithms: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs)

Clinical Relation Extraction Toward Drug Safety Surveillance Using Electronic Health Record Narratives: Classical Learning Versus Deep Learning

Objective: To evaluate natural language processing and machine learning approaches using the expert-annotated medical entities and relations in the context of drug safety surveillance, and investigate how different learning approaches perform under different configurations.

Algorithm Used: Support vector machines, Deep neural networks, Supervised Descriptive Rule Induction.

Dataset: Healthcare notes with medication, indication, severity, and adverse drug events (ADE), medication-dosage, medication-ADE, and severity-ADE.

publichealth.jmir.org/2018/2/e29/

Deep Learning: Python libraries

Python offers a robust ecosystem of libraries for deep learning:

1. **TensorFlow**:
    1. Developed by Google, TensorFlow is an open-source deep learning library.
    2. Supports building and training neural networks for various tasks such as image recognition, natural language processing, and more.
    3. TensorFlow 2.x integrates Keras as its high-level API, making it easier to develop models.
2. **Keras**:
    1. A high-level neural networks API, written in Python and capable of running on top of TensorFlow, Microsoft Cognitive Toolkit (CNTK), or Theano.
    2. Simplifies building and training deep learning models with a user-friendly and modular approach.
3. **PyTorch**:
    1. Developed by Facebook's AI Research lab, PyTorch is an open-source deep learning library.
    2. Known for its dynamic computation graph, which makes it easier and more intuitive to build and modify neural networks.
    3. Popular in both academia and industry for research and production.

Deep Learning: Python libraries

4. **MXNet**:
    1. An open-source deep learning framework developed by Apache.
    2. Supports a wide range of languages including Python, and provides efficient tools for building and training deep learning models.
    3. Known for its scalability and performance, especially in distributed computing environments.
5. **Caffe**:
    1. Developed by the Berkeley Vision and Learning Center (BVLC), Caffe is a deep learning framework focused on speed and modularity.
    2. Suitable for image classification and convolutional neural networks (CNNs).
6. **Theano**:
    1. One of the earliest deep learning libraries, developed by the Montreal Institute for Learning Algorithms (MILA) at the University of Montreal.
    2. While it has been succeeded by other libraries like TensorFlow and PyTorch, Theano laid the groundwork for many subsequent frameworks.
7. **Chainer**:
    1. A flexible and intuitive deep learning framework that supports dynamic computation graphs (define-by-run).
    2. Particularly well-suited for complex and varied network architectures.

Deep Learning: Python libraries

8. **Fastai**:
    1. Built on top of PyTorch, Fastai provides high-level components that make it easy to train state-of-the-art deep learning models.
    2. Known for its user-friendly API and focus on making deep learning accessible.
9. **DL4J (DeepLearning4J)**:
    1. An open-source, distributed deep learning library for the Java Virtual Machine (JVM), but also offers APIs in Python.
    2. Integrates well with big data tools like Apache Hadoop and Apache Spark.
10. **Gluon**:
    1. A deep learning library jointly developed by AWS and Microsoft, providing an easy-to-use interface for building neural networks.
    2. Integrates with Apache MXNet to combine ease of use with performance.

Poll

Which type of machine learning involves the model being trained on a labeled dataset where output for each input is known?

a) Supervised learning

b) Unsupervised learning

c) Semi-supervised learning

d) Reinforcement learning

Poll

Which type of machine learning involves the model learning by interacting with an environment and receiving rewards or penalties based on its actions?

a) Supervised learning

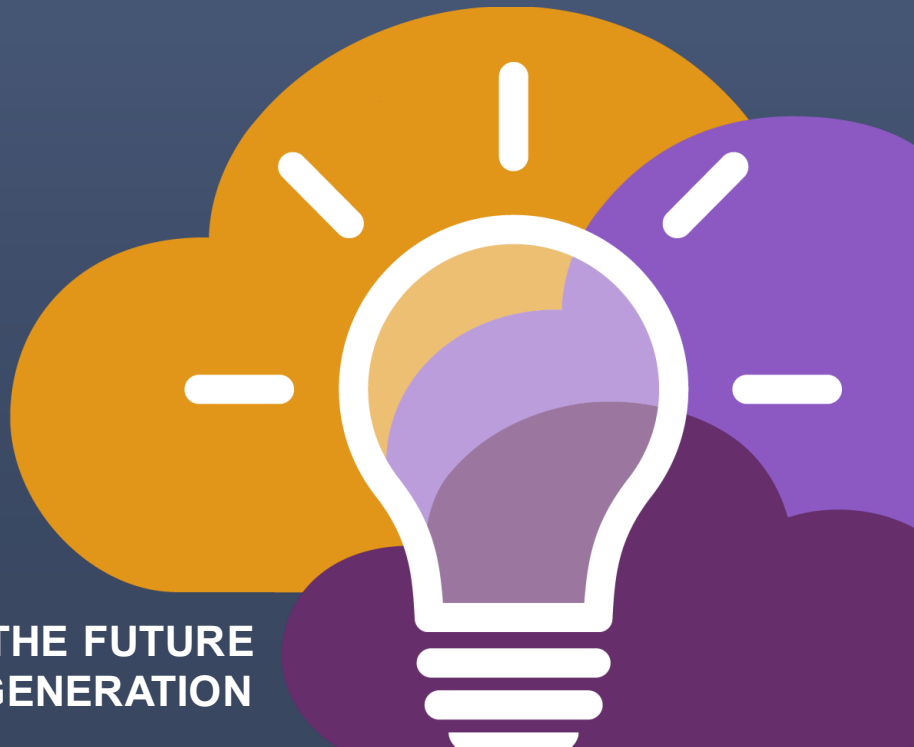
b) Unsupervised learning

c) Semi-supervised learning

d) Reinforcement learning

ScHARe

Resources



BE A PART OF THE FUTURE
OF KNOWLEDGE GENERATION

ScHARe resources

Support made available to users:

ScHARe-specific

- ScHARe documentation
- Email support

Platform-specific

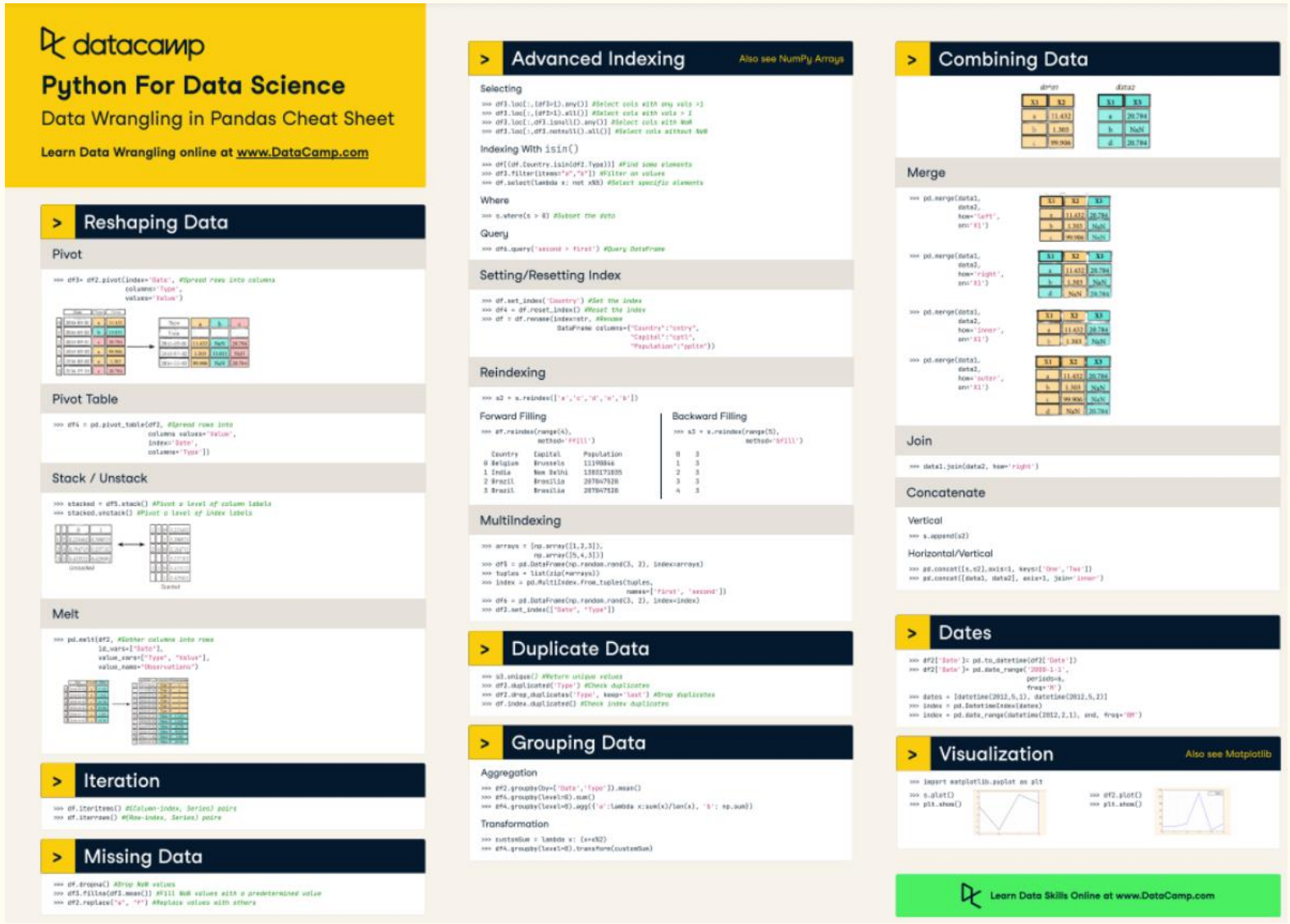
- Terra-specific support
- Terra-specific documentation

ScHARe resources

Training opportunities made available to users:

- Monthly **Think-a-Thons**
- **Instructional materials** and slides made available online on NIMHD website
- **YouTube videos**
- **Links to relevant online resources** and training on NIMHD website
- **Pilot credits** for testing ScHARe for research needs
- **Instructional Notebooks** in ScHARe Workspace with instructions for:
  - Exploring the data ecosystem
  - Setting your workspace up for use
  - Accessing and interacting with the categories of data accessible through ScHARe

ScHARe resources: cheatsheets



Credits: datacamp.com

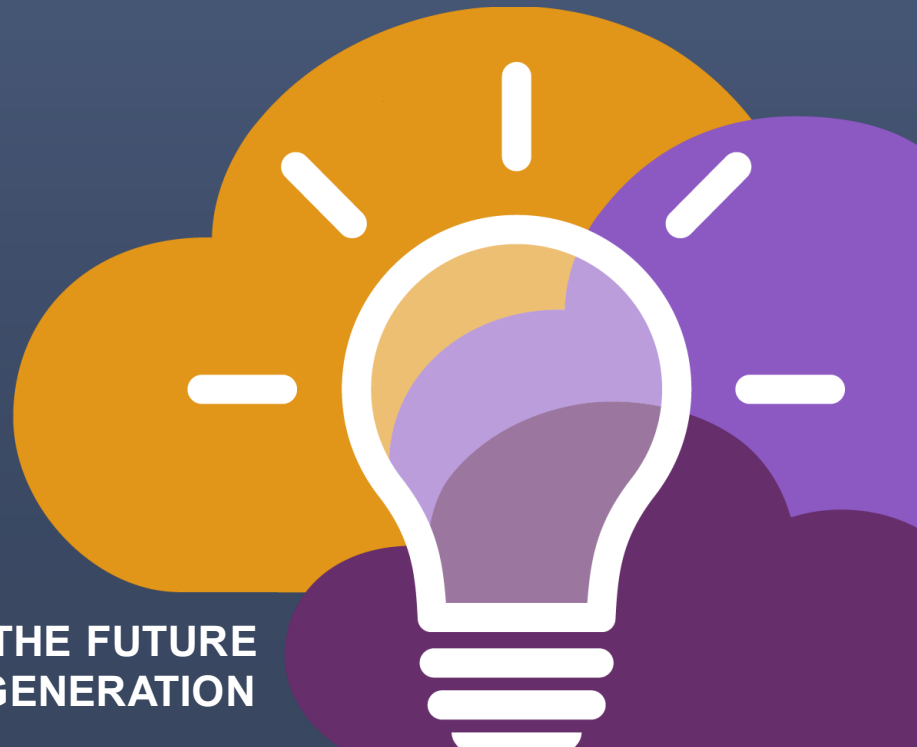
Terra resources

If you are new to Terra, we recommend exploring the following resources:

- Overview Articles: Review high-level docs that outline what you can do in Terra, how to set up an account and account billing, and how to access, manage, and analyze data in the cloud
- Video Guides: Watch live demos of the Terra platform's useful features
- Terra Courses: Learn about Terra with free modules on the Leanpub online learning platform
- Data Tables QuickStart Tutorial: Learn what data tables are and how to create, modify, and use them in analyses
- Notebooks QuickStart Tutorial**:** Learn how to access and visualize data using a notebook
- Machine Learning Advanced Tutorial: Learn how Terra can support machine learning-based analysis

ScHARe

Thank you



BE A PART OF THE FUTURE
OF KNOWLEDGE GENERATION

Think-a-Thon poll

1. **Rate how useful this session was:**

- ☐ Very useful
- ☐ Useful
- ☐ Somewhat useful
- ☐ Not at all useful

Think-a-Thon poll

2. Rate the pace of the instruction for yourself:

- ☐ Too fast
- ☐ Adequate for me
- ☐ Too slow

Think-a-Thon poll

3. **How likely will you participate in the next Think-a-Thon?**

- ☐ Very interested, will definitely attend
- ☐ Interested, likely will attend
- ☐ Interested, but not available
- ☐ Not interested in attending any others

ScHARe

Next Think-a-Thons:

bit.ly/think-a-thons

Register for ScHARe:

bit.ly/join-schare

schare@mail.nih.gov