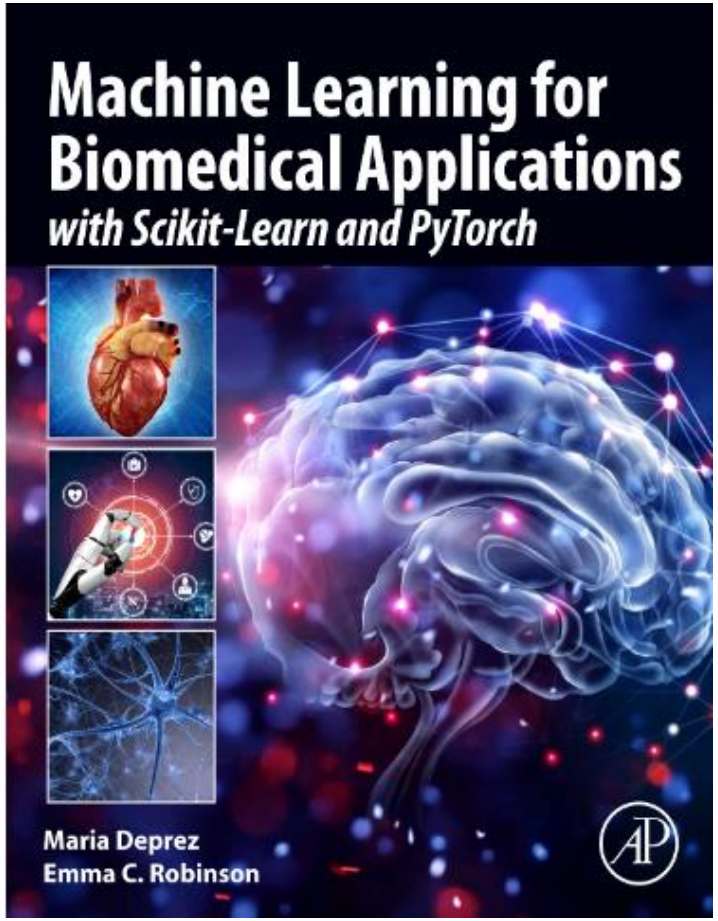
AI in Biomedical Data

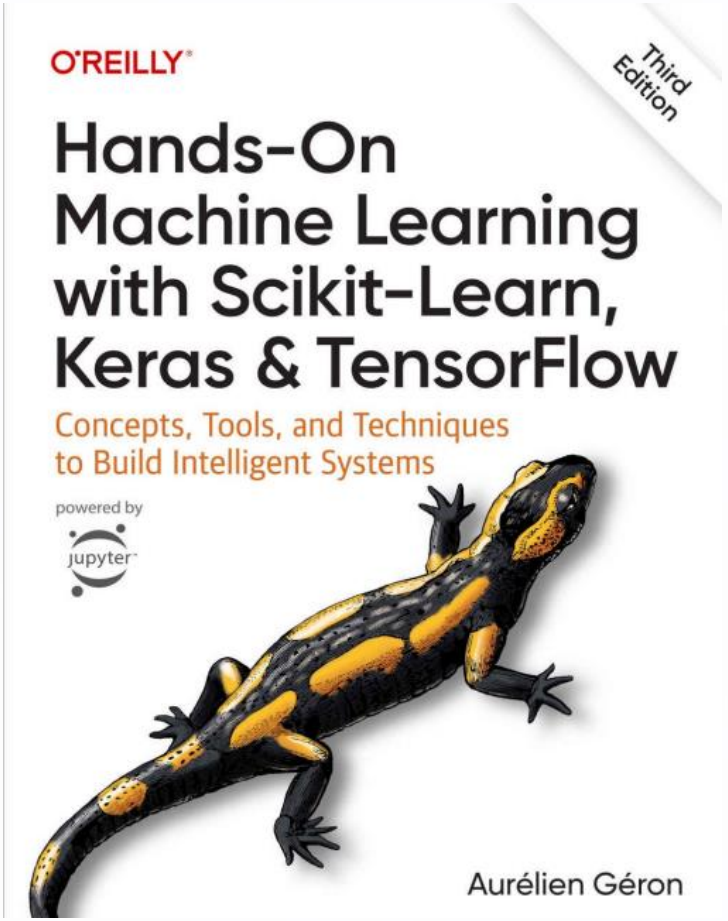
Dr. M.B. Khodabakhshi
Amir Hossein Fouladi
Alireza Javadi

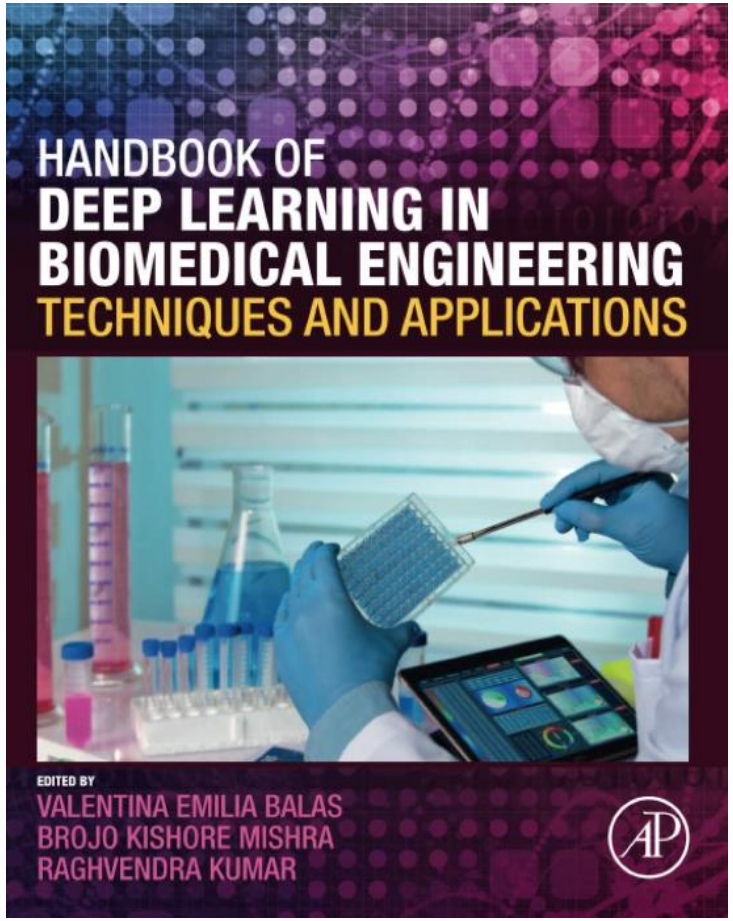
github.com/mbkhodabakhshi/AI_in_BiomedicalData

یادگیری ماشین در زیست پزشکی

دکتر محمدباقر خدابخشی
mb.khodabakhshi@gmail.com


Machine Learning for
Biomedical Applications
with Scikit-Learn and PyTorch
Maria Deprez
Emma C. Robinson


O'REILLY
Third Edition
Hands-On
Machine Learning
with Scikit-Learn,
Keras & TensorFlow
Concepts, Tools, and Techniques
to Build Intelligent Systems
powered by
jupyter
Aurélien Géron


HANDBOOK OF
DEEP LEARNING IN
BIOMEDICAL ENGINEERING
TECHNIQUES AND APPLICATIONS
EDITED BY
VALENTINA EMILIA BALAS
BROJO KISHORE MISHRA
RAGHVENDRA KUMAR

Objective and Approach

Objective and Approach This book assumes that you know close to nothing about machine learning. Its goal is to give you the concepts, tools, and intuition you need to implement programs capable of learning from data.We will be using production-ready Python frameworks:

- Scikit-Learn is very easy to use, yet it implements many machine learning algorithms efficiently, so it makes for a great entry point to learning machine learning. It was created by David Cournapeau in 2007, and is now led by a team of researchers at the French Institute for Research in Computer Science and Automation (Inria).
- TensorFlow is a more complex library for distributed numerical computation. It makes it possible to train and run very large neural networks efficiently by distributing the computations across potentially hundreds of multi-GPU (graphics processing unit) servers. TensorFlow (TF) was created at Google and supports many of its large-scale machine learning applications. It was open sourced in November 2015, and version 2.0 was released in September 2019.
- Keras is a high-level deep learning API that makes it very simple to train and run neural networks. Keras comes bundled with TensorFlow, and it relies on TensorFlow for all the intensive computations.

Prerequisites

This book assumes that you have some Python programming experience. If you don't know Python yet, https://learnpython.org is a great place to start. The official tutorial on Python.org is also quite good.

This book also assumes that you are familiar with Python's main scientific libraries—in particular, NumPy, Pandas, and Matplotlib. If you have never used these libraries, don't worry; they're easy to learn, and I've created a tutorial for each of them. You can access them online at https://homl.info/tutorials.

Moreover, if you want to fully understand how the machine learning algorithms work (not just how to use them), then you should have at least a basic understanding of a few math concepts, especially linear algebra. Specifically, you should know what vectors and matrices are, and how to perform some simple operations like adding vectors, or transposing and multiplying matrices. **If you need a quick introduction to linear algebra (it's really not rocket science!), I provide a tutorial at https://homl.info/tutorials**. You will also find a tutorial on differential calculus, which may be helpful to understand how neural networks are trained, but it's not entirely essential to grasp the important concepts.

Roadmap

This book is organized in two parts. Part I, "The Fundamentals of Machine Learning", covers the following topics:

- What machine learning is, what problems it tries to solve, and the main categories and fundamental concepts of its systems
- The steps in a typical machine learning project
- Learning by fitting a model to data
- Optimizing a cost function
- Handling, cleaning, and preparing data
- Selecting and engineering features
- Selecting a model and tuning hyperparameters using cross-validation
- The challenges of machine learning, in particular underfitting and overfitting (the bias/variance trade-off)
- The most common learning algorithms: linear and polynomial regression, logistic regression, *k*-nearest neighbors, support vector machines, decision trees, random forests, and ensemble methods
- Reducing the dimensionality of the training data to fight the "curse of dimensionality"
- Other unsupervised learning techniques, including clustering, density estimation, and anomaly detection

Part II, "Neural Networks and Deep Learning", covers the following topics:

- What neural nets are and what they're good for
- Building and training neural nets using TensorFlow and Keras
- The most important neural net architectures: feedforward neural nets for tabular data, convolutional nets for computer vision, recurrent nets and long short-term memory (LSTM) nets for sequence processing, encoder–decoders and transformers for natural language processing (and more!), autoencoders, generative adversarial networks (GANs), and diffusion models for generative learning
- Techniques for training deep neural nets
- How to build an agent (e.g., a bot in a game) that can learn good strategies through trial and error, using reinforcement learning
- Loading and preprocessing large amounts of data efficiently
- Training and deploying TensorFlow models at scale

پایتون: کلید طلایی دنیای تحلیل داده

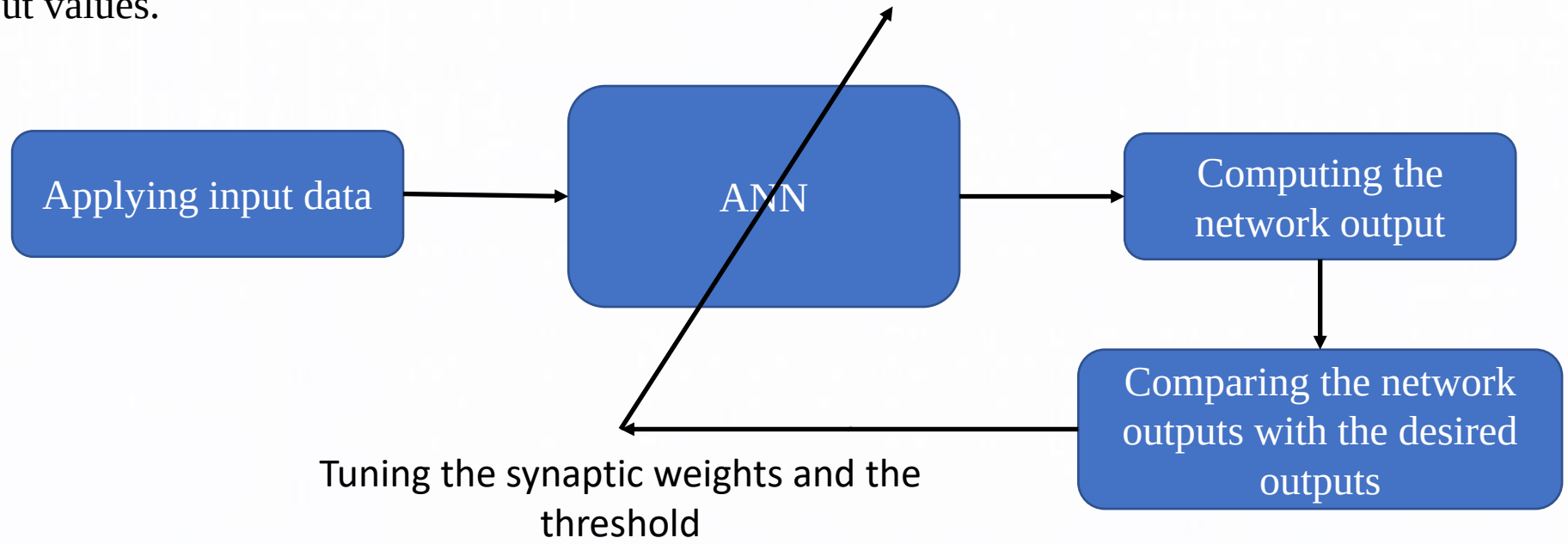
ستاد توسعه فناوری های هوش مصنوعی و رباتیک

- **سادگی و خوانایی:** سینتکس ساده و نزدیک به زبان طبیعی، یادگیری سریع و کاهش خطاها.
- **جامعه بزرگ و فعال:** دسترسی آسان به منابع آموزشی، کتابخانه‌ها و پشتیبانی جامعه.
- **کتابخانه‌های قدرتمند:** NumPy، Pandas، Matplotlib، Seaborn و بسیاری دیگر برای عملیات ریاضی، پردازش داده، و تجسم.
- **کاربرد گسترده:** از تحلیل داده‌های کوچک تا پروژه‌های بزرگ یادگیری ماشین .

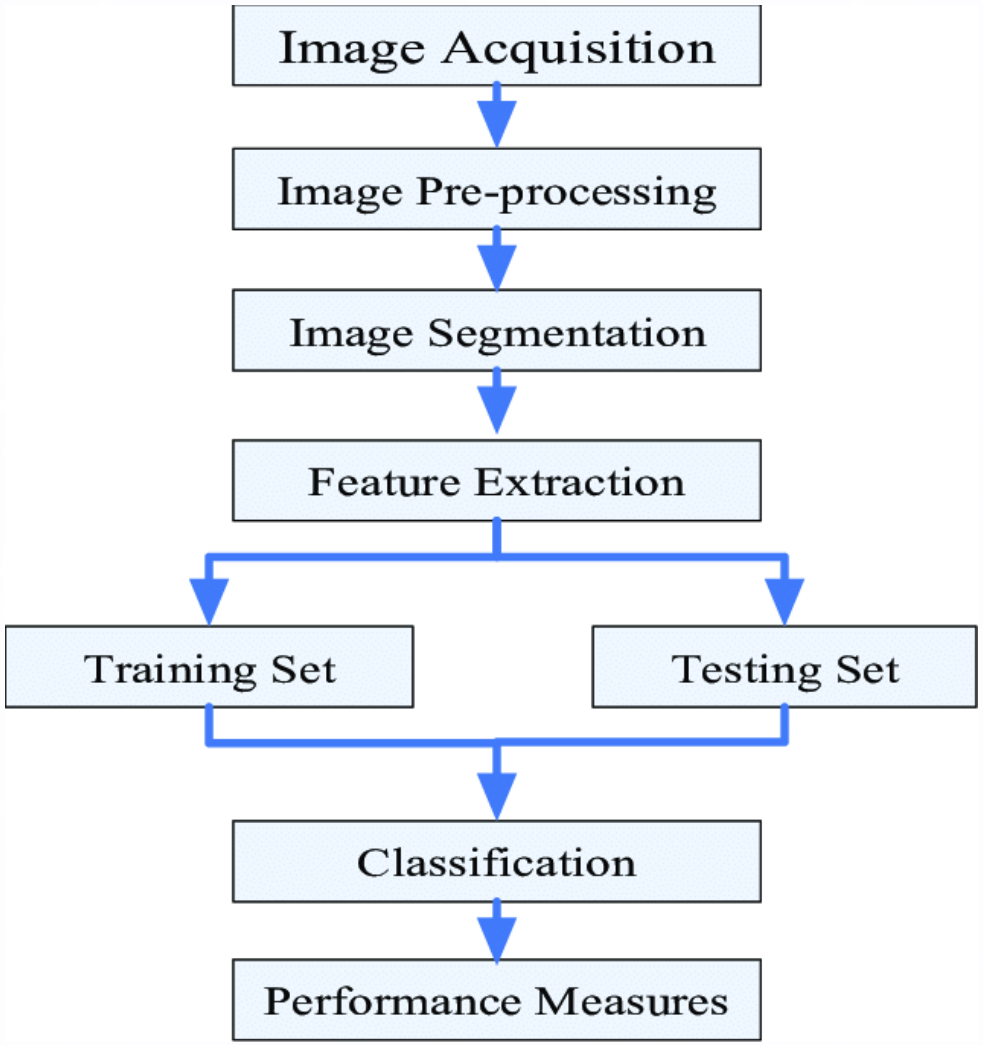
هوش مصنوعی و یادگیری ماشین

- One of the most relevant features of artificial neural networks is their capability of **learning from the presentation of samples (patterns)**.

- After the network has learned the relationship between inputs and outputs, it can **generalize solutions**, meaning that **the network can produce an output which is close to the expected** (or desired) output of any given input values.



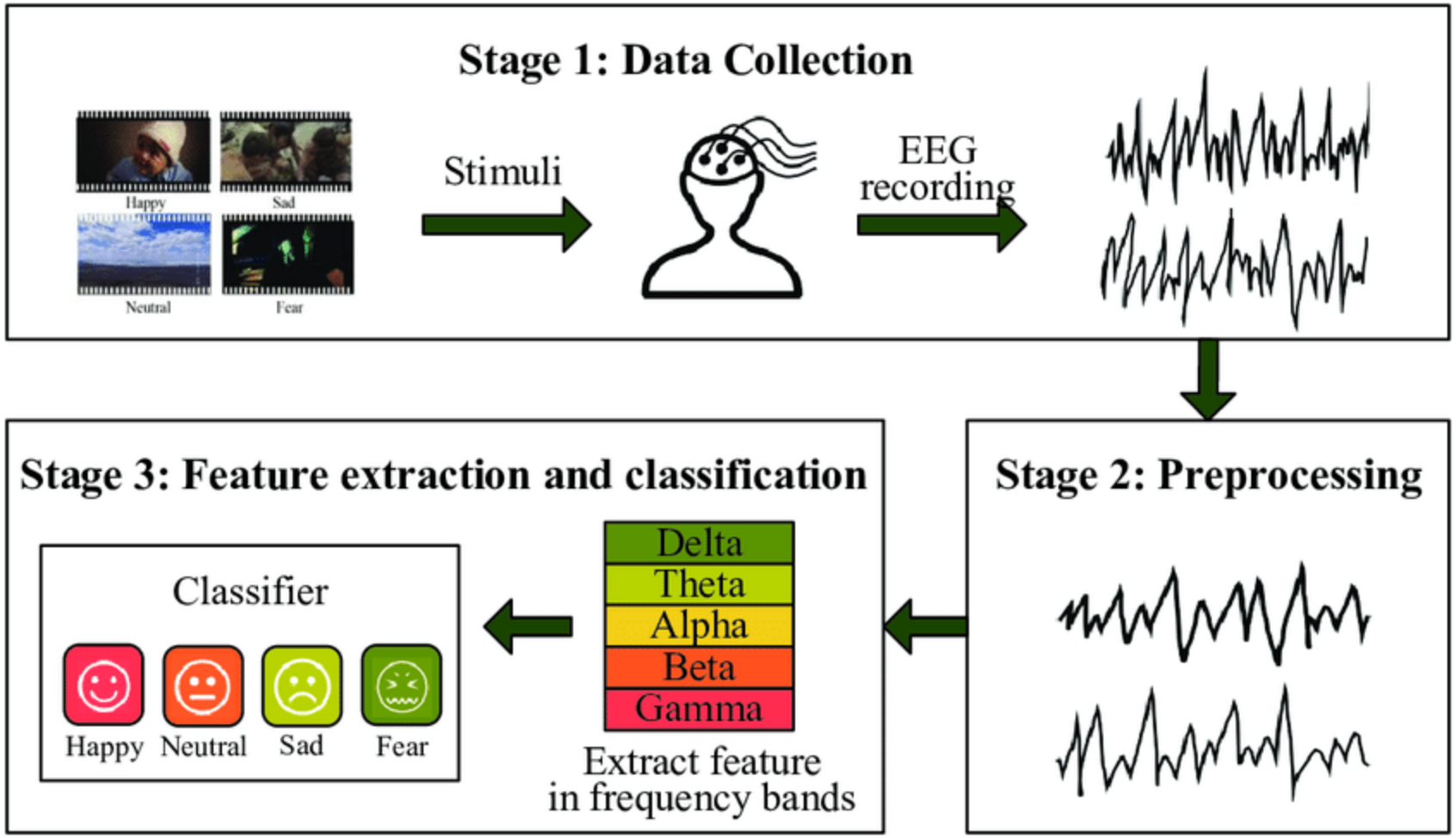
هوش مصنوعی، رویکرد سنتی



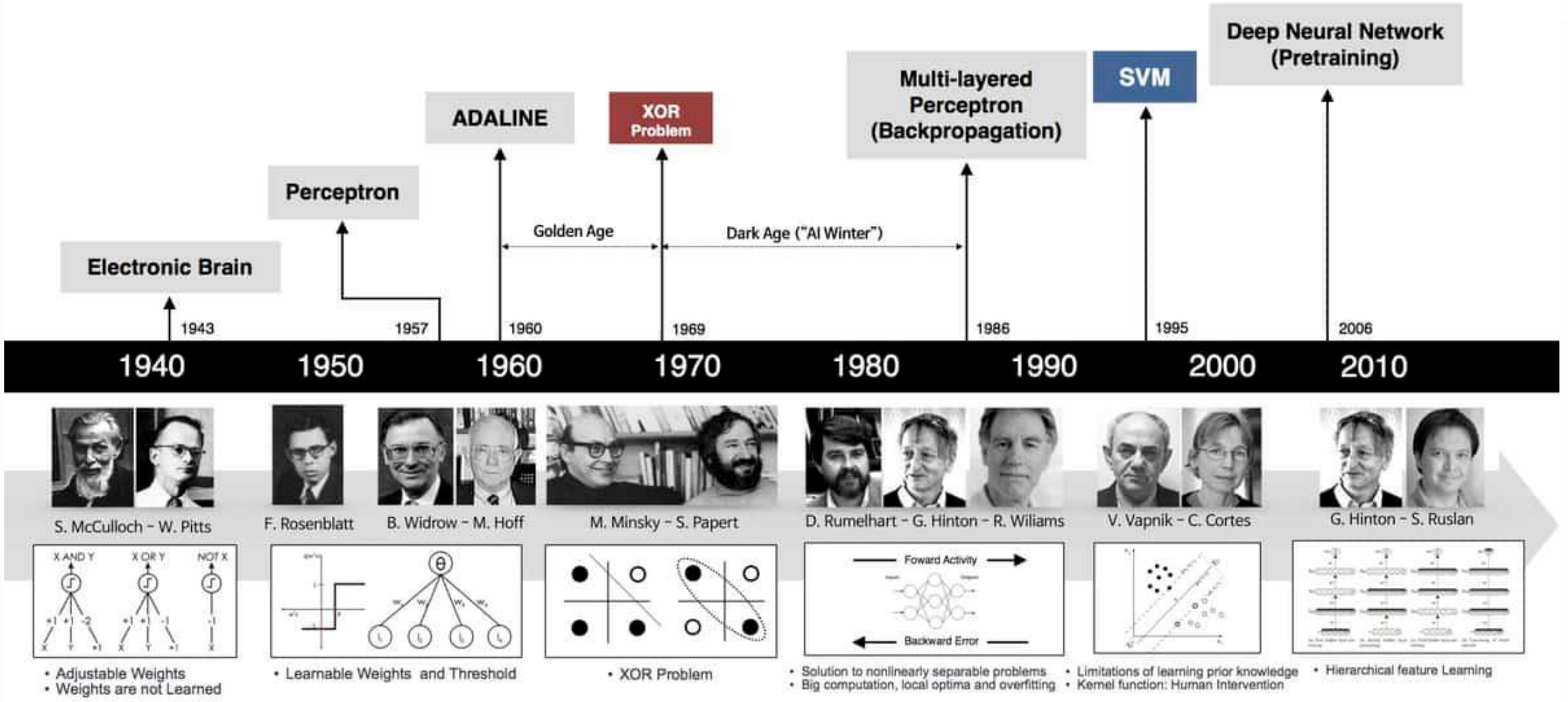
- اکتساب داده
- پیش پردازش (تمیز کردن، استخراج زیرباندهای فرکانسی و ...)
- استخراج ویژگی (بروز تمایزات بین دسته های داده ها)
- الگوریتم هوشمند طبقه بندی

استخراج ویژگی های توانی از سیگنال EEG

تشخیص خودکار احساسات



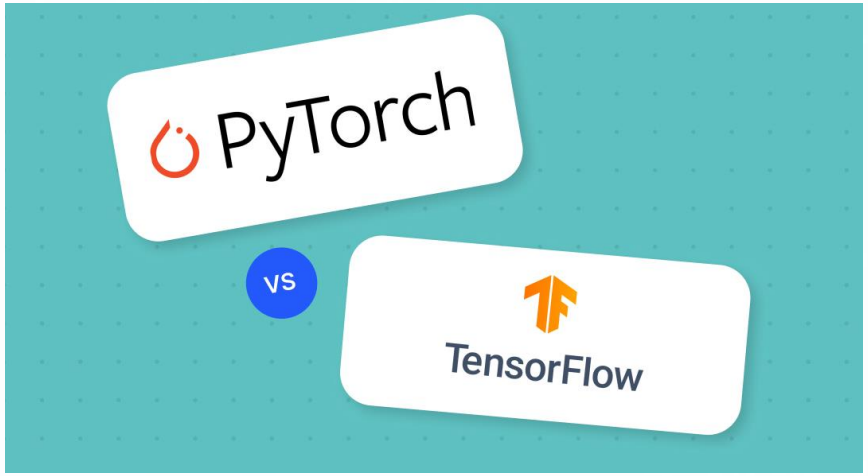
تاریخچه هوش مصنوعی



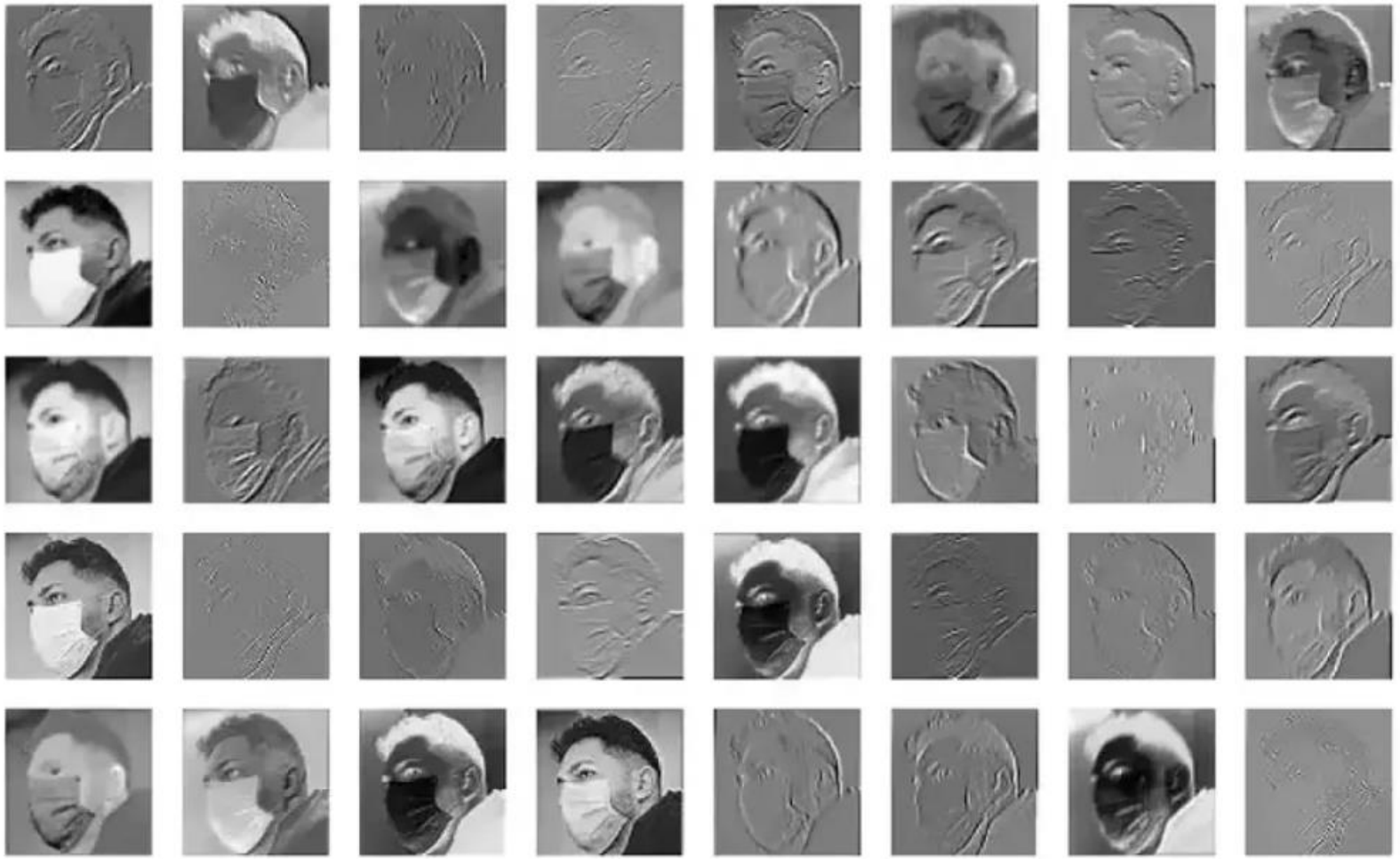
انقلاب شبکه های عصبی عمیق در علوم اعصاب

- **استخراج خودکار ویژگی‌ها:** شبکه‌های عصبی به صورت خودکار ویژگی‌های مهم را از تصاویر استخراج می‌کنند.
- **دقت بالا:** شبکه‌های عمیق می‌توانند دقت تشخیص را به میزان قابل توجهی افزایش دهند.
- **انعطاف‌پذیری:** می‌توانند برای انواع مختلف داده‌های تصویربرداری پزشکی استفاده شوند.

- **پایتون و یادگیری عمیق:** کتابخانه‌های قدرتمندی مانند TensorFlow و PyTorch برای پیاده‌سازی و آموزش شبکه‌های عصبی

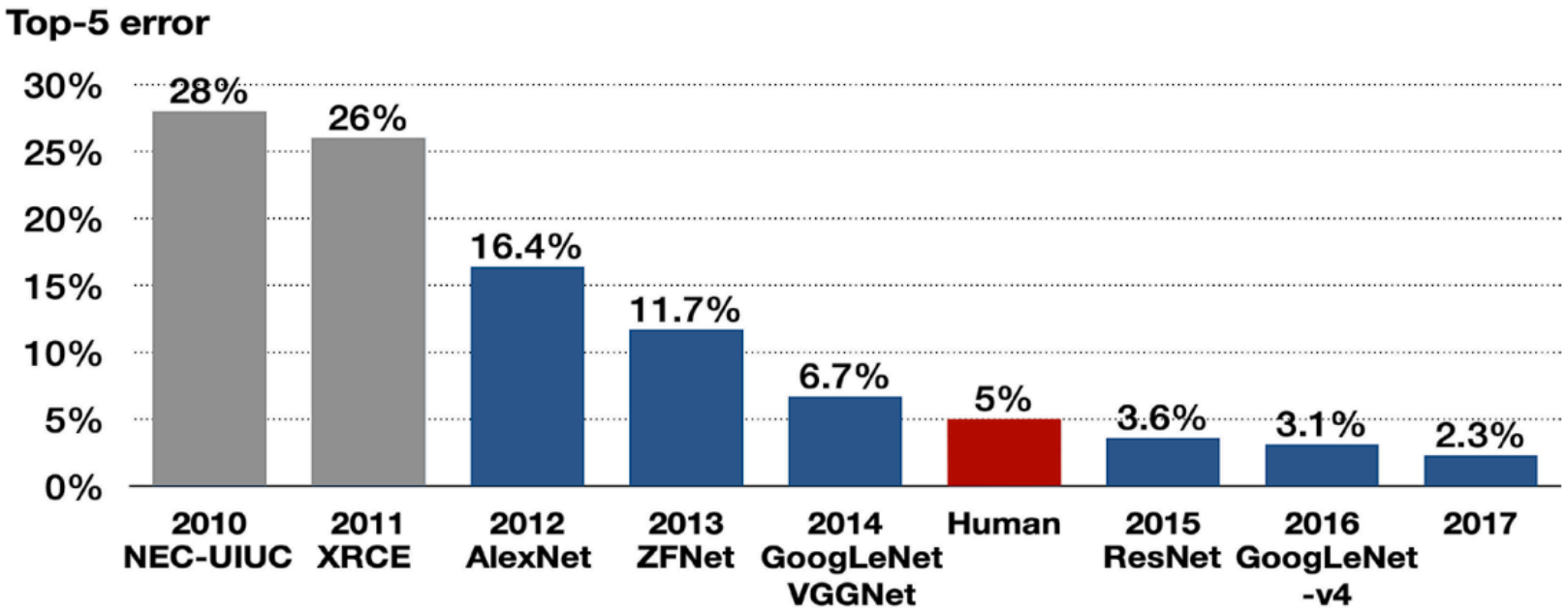


شبکه های کانولوشنی بدون استخراج ویژگی



حرکت از رویکرد سنتی به یادگیری عمیق

مسابقه ImageNetیک چالش دسته‌بندی با ۱/۲ میلیون تصویر در ۱۰۰۰ کلاس بود. شبکه AlexNet با خطای ٪۱۶.۴ به رتبه اول مسابقه رسید. اختلاف خطای حداقل ۱۰ درصدی بین شبکهCNN با سایر روش‌ها...

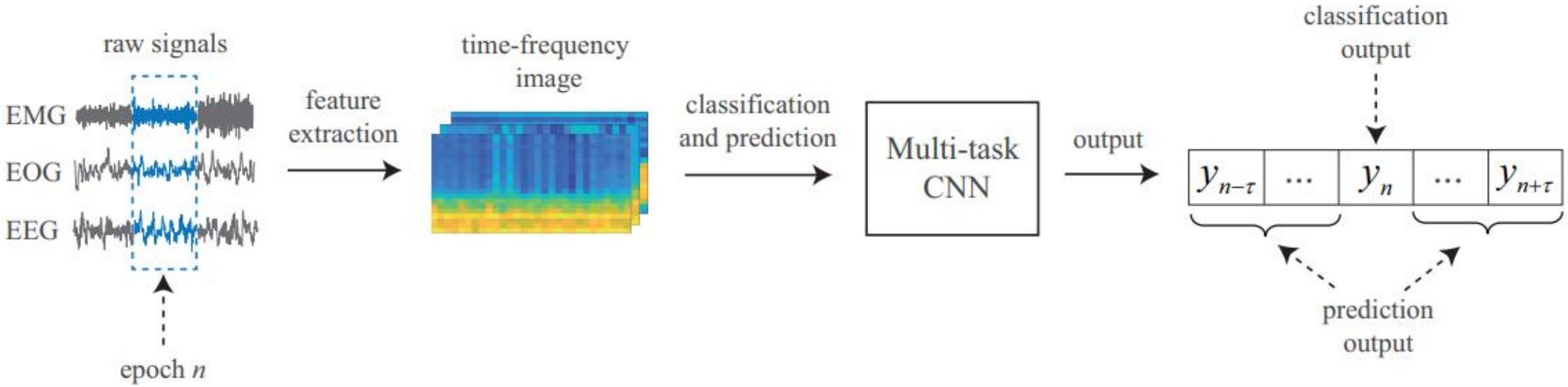


Presenter

تحلیل مراحل خواب

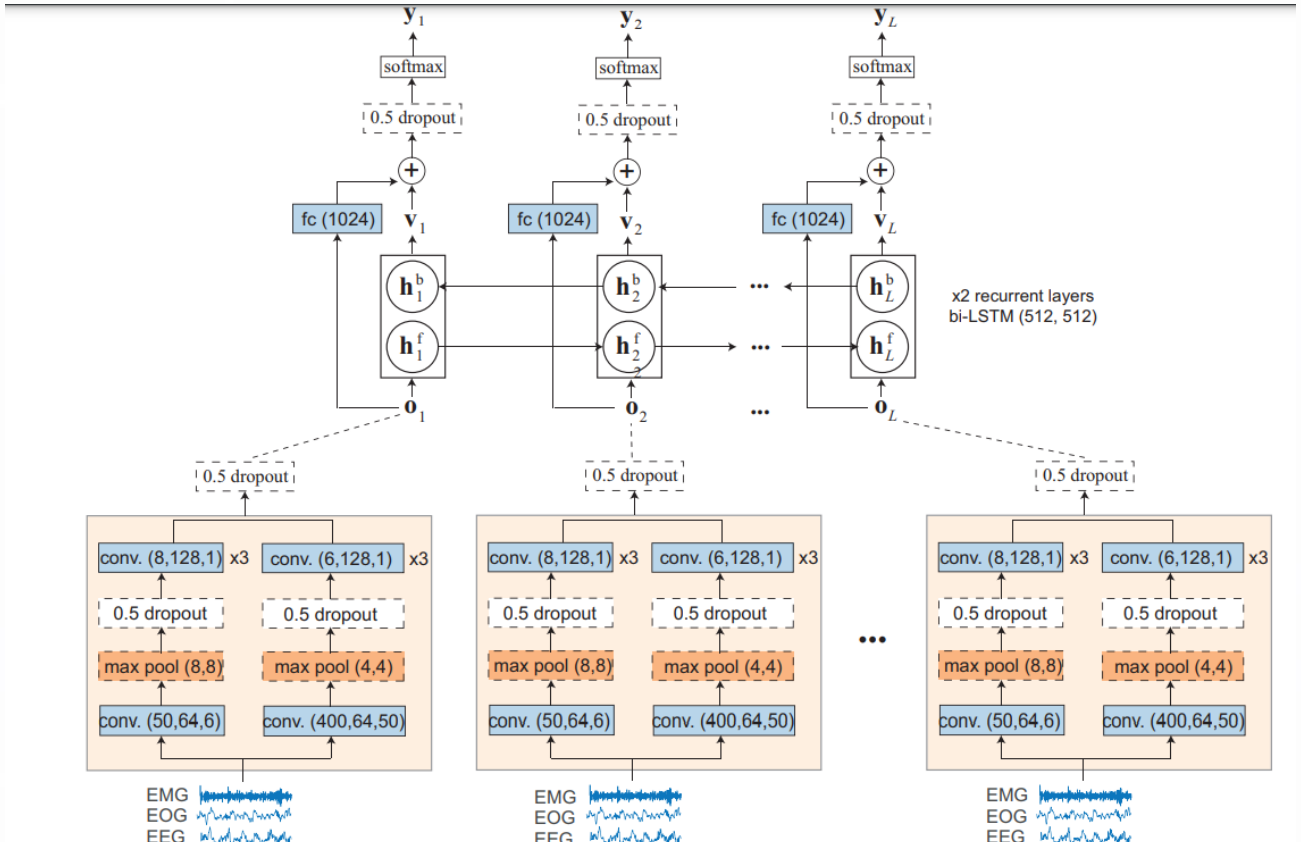
Sleep-EDF dataset

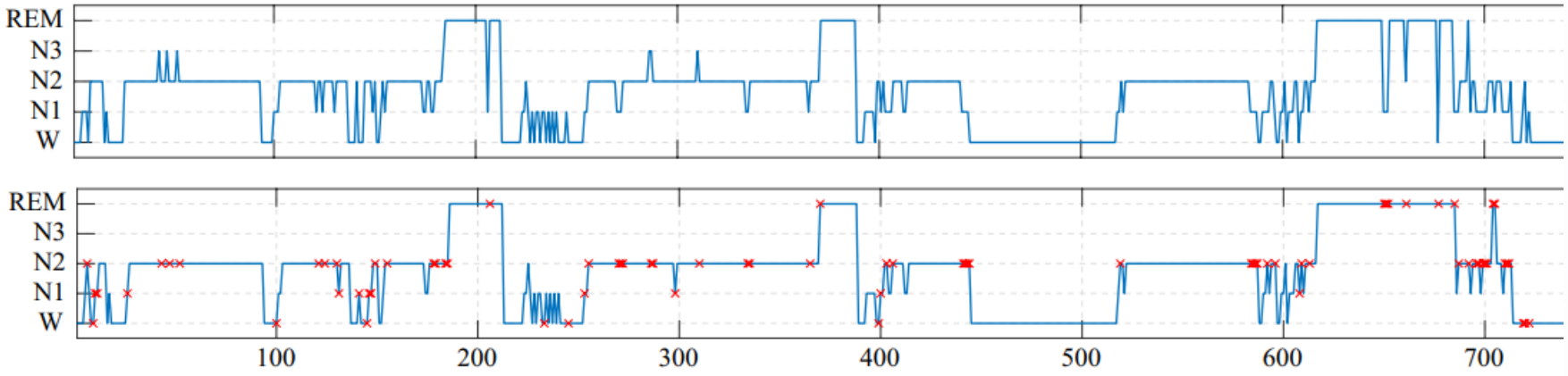
Each 30-second epoch of the recordings was manually labelled by sleep experts into one of eight categories {W, N1, N2, N3, N4, REM, MOVEMENT, UNKNOWN}.



Phan, Huy, et al. "Joint classification and prediction CNN framework for automatic sleep stage classification." *IEEE Transactions on Biomedical Engineering* 66.5 (2018): 1285-1296.

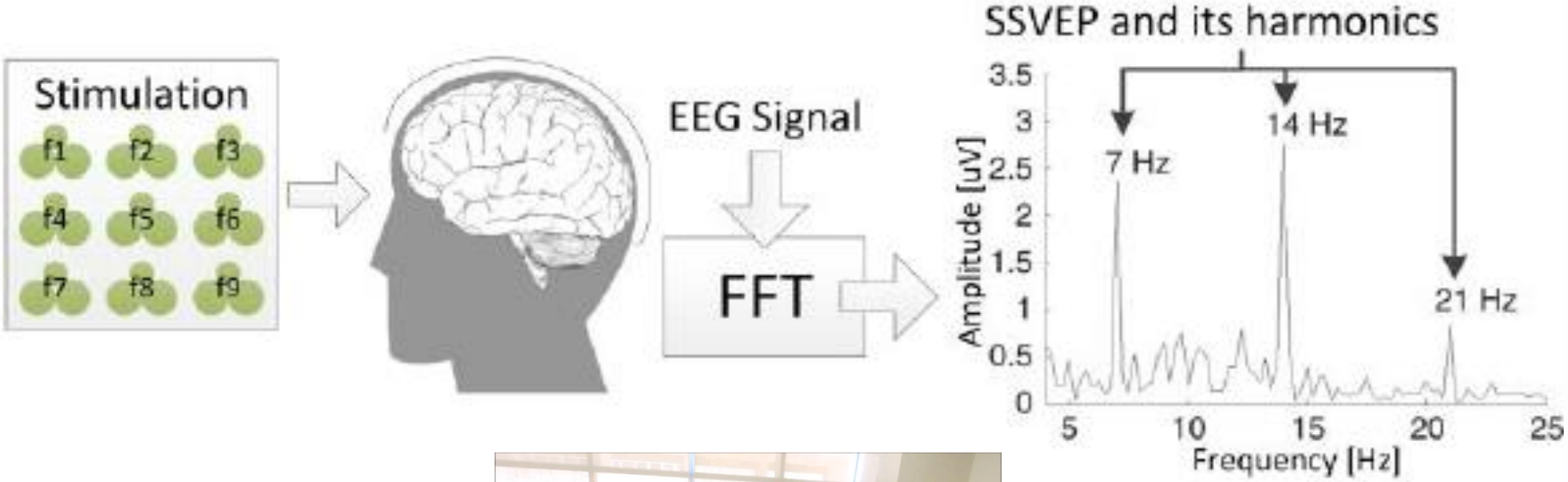
شبکه های بازگشتی در تحلیل مراحل خواب





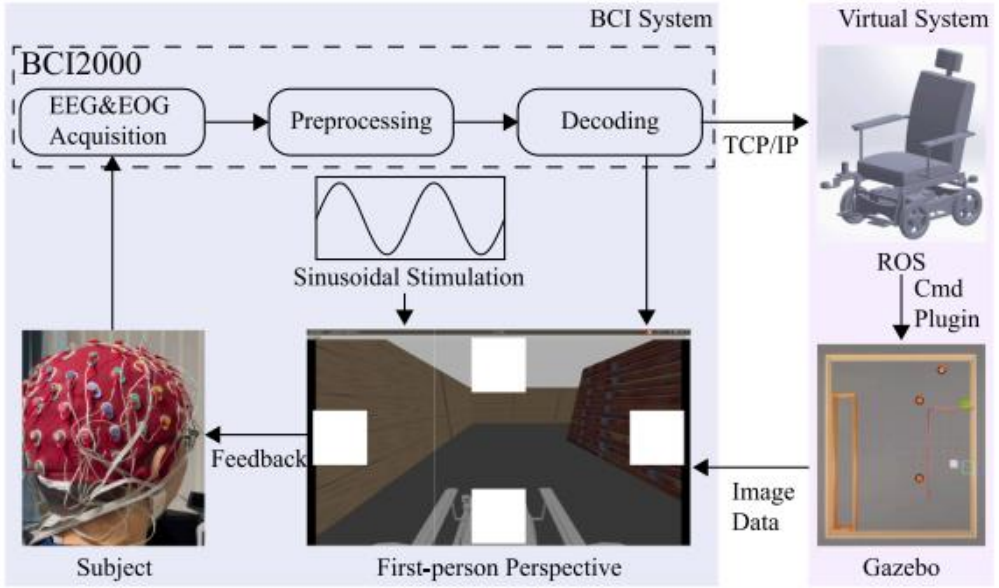
Phan, Huy, et al. "SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.3 (2019): 400-410.

واسط مغز و کامپیوتر: پتانسیل برانگیخته ماندگار

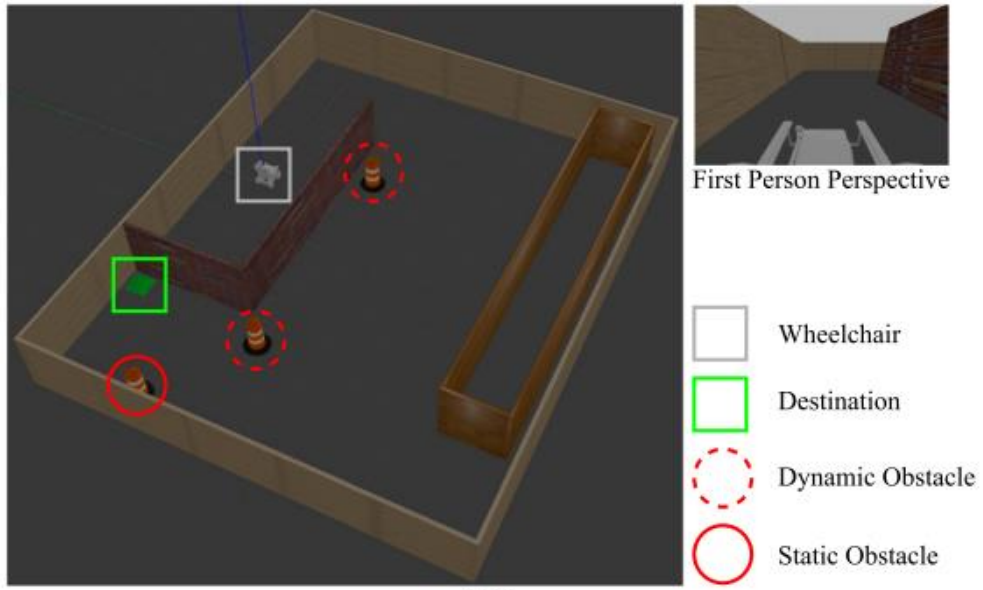


کنترل ویلچر از طریق SSVEP

Mai, Ximing, et al. "A hybrid BCI combining SSVEP and EOG and its application for continuous wheelchair control." *Biomedical Signal Processing and Control* 88 (2024): 105530.

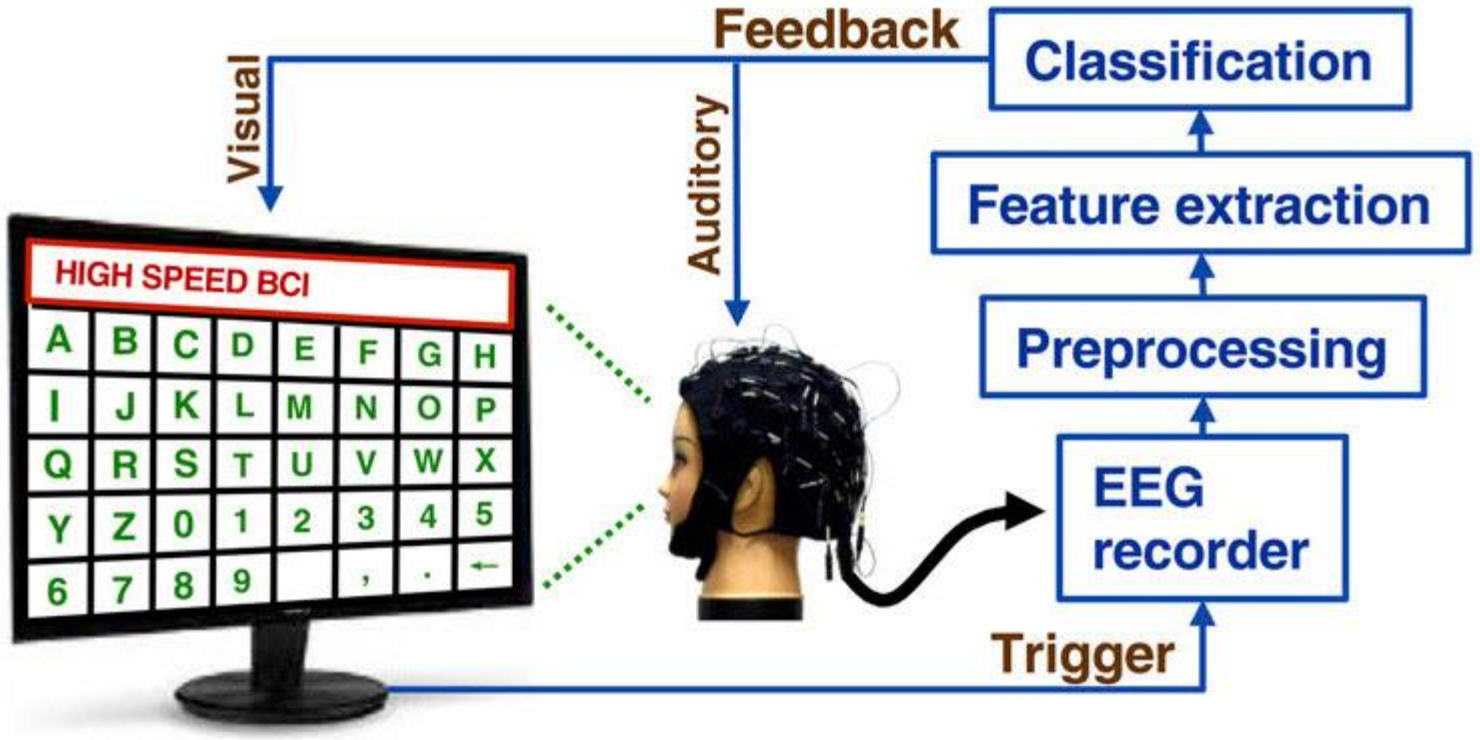


(a)



(b)

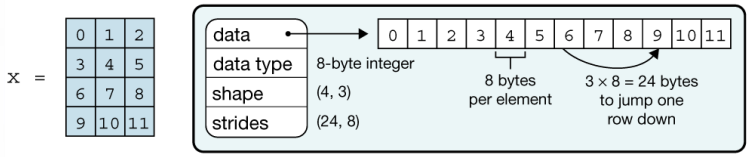
SSVEP BCI Speller



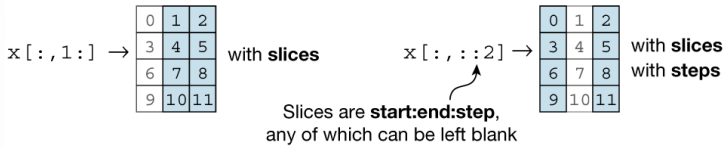
برخی ابزارهای قدرتمند پایتون

NumPy

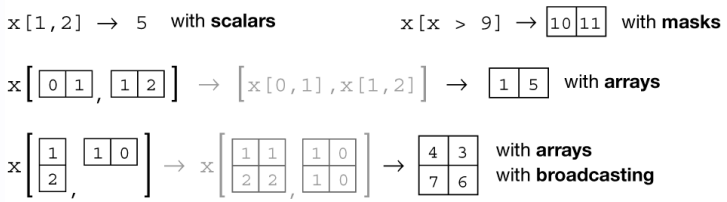
a Data structure



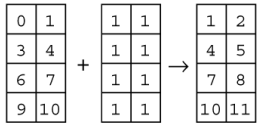
b Indexing (view)



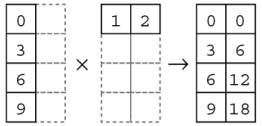
c Indexing (copy)



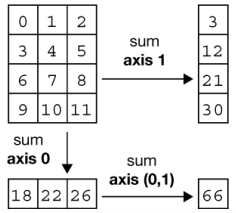
d Vectorization



e Broadcasting



f Reduction

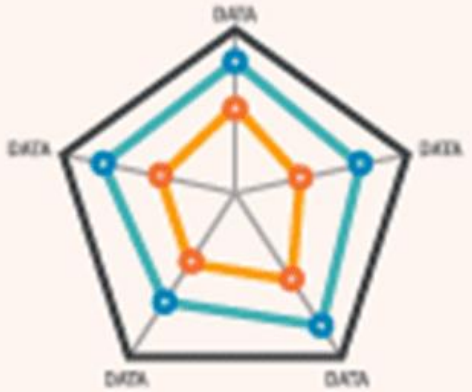


g Example

```
In [1]: import numpy as np

In [2]: x = np.arange(12)

In [3]: x = x.reshape(4, 3)

In [4]: x
Out[4]:
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])

In [5]: np.mean(x, axis=0)
Out[5]: array([4.5, 5.5, 6.5])

In [6]: x = x - np.mean(x, axis=0)

In [7]: x
Out[7]:
array([[-4.5, -4.5, -4.5],
       [-1.5, -1.5, -1.5],
       [ 1.5,  1.5,  1.5],
       [ 4.5,  4.5,  4.5]])
```

برخی ابزارهای قدرتمند پایتون

matplotlib



کاربردهای عملی پایتون در تحلیل فرآیندهای شناختی

scikit-learn

Machine Learning in Python

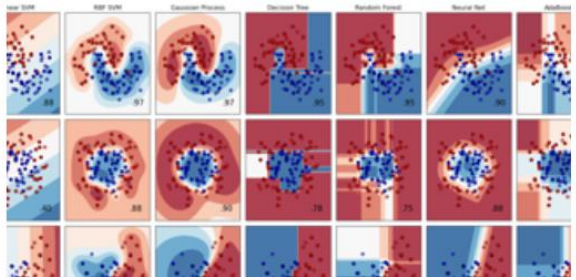
Getting Started | Release Highlights for 1.5

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

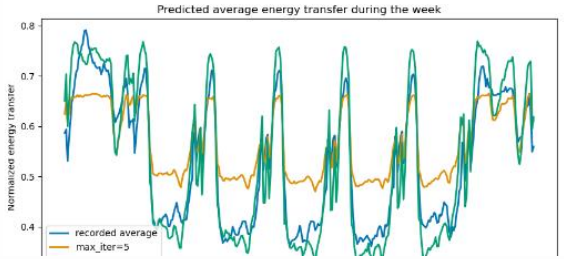
Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...



Regression

Predicting a continuous-valued attribute associated with an object.

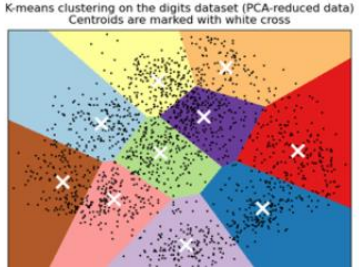
Applications: Drug response, stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...



Clustering

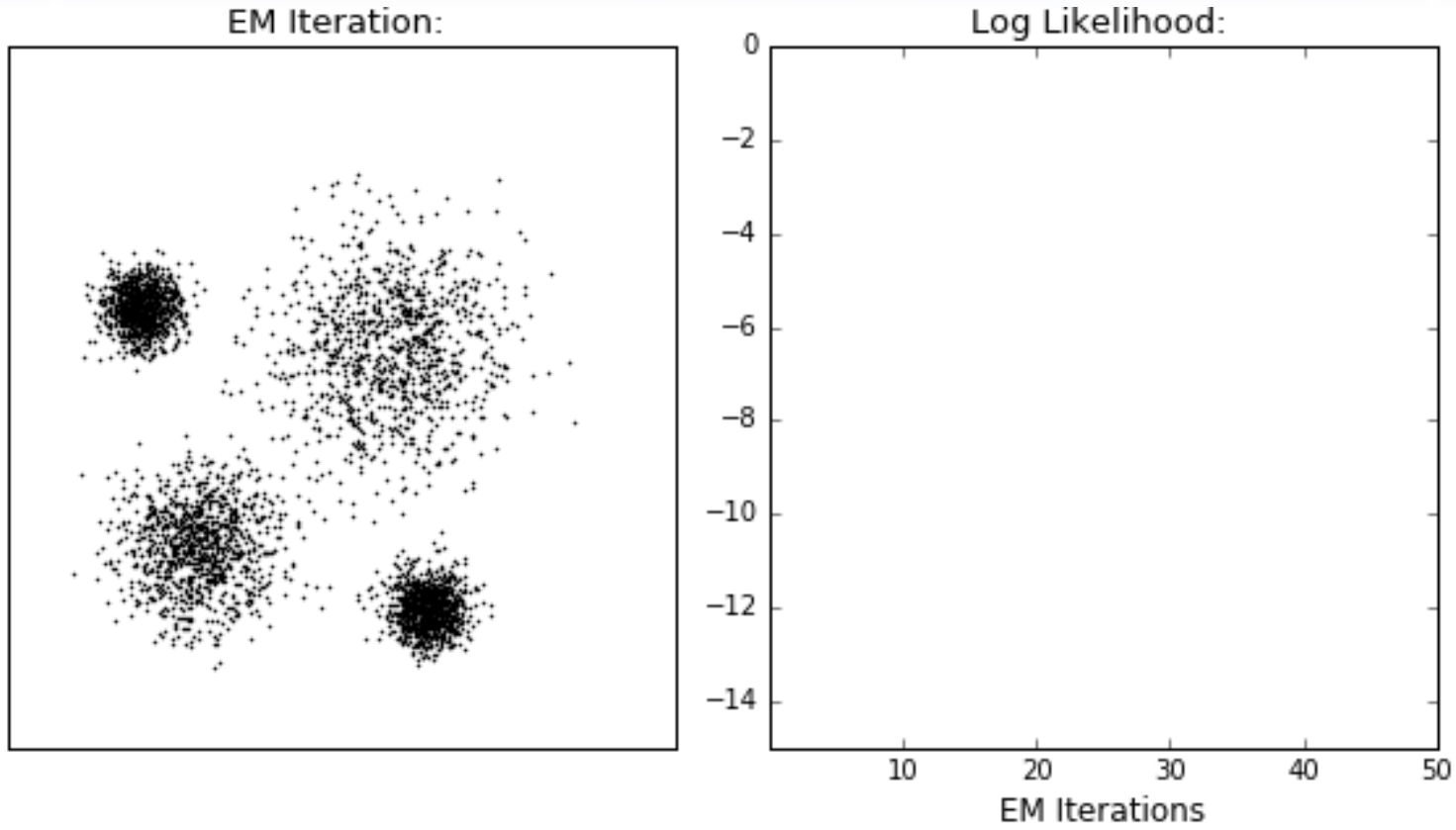
Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



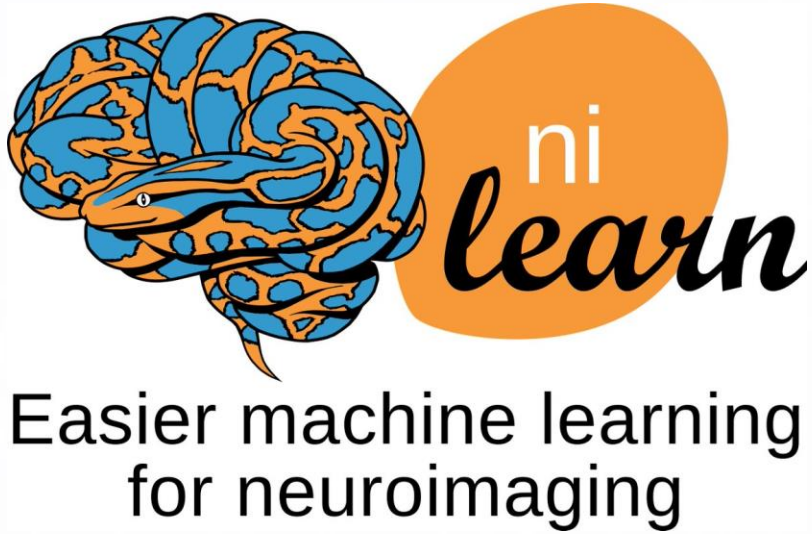
Presenter: Dr.Khodabakhshi

مثالی از خوشه بندی با ابزارهای Scikitlearn



کاربردهای عملی پایتون در تحلیل فرآیندهای شناختی

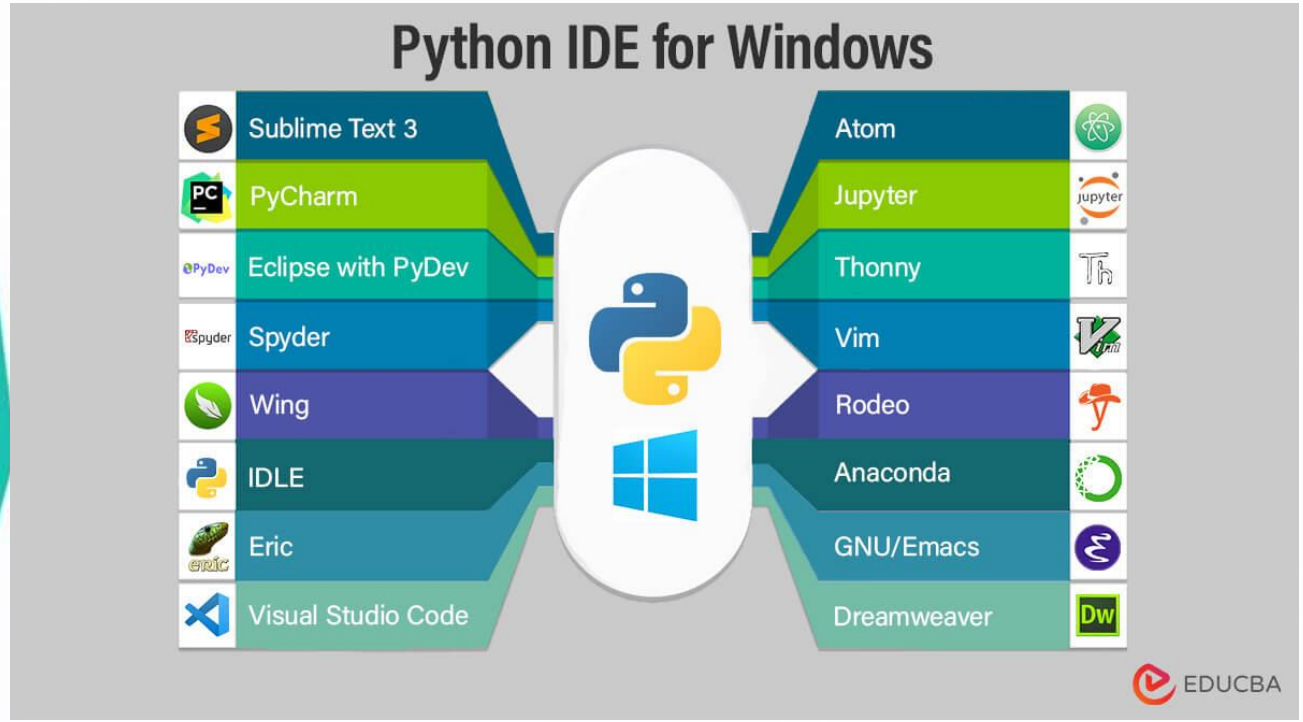
Open-source Python package for exploring, visualizing, and analyzing human neurophysiological data: MEG, EEG, sEEG, ECoG, NIRS, and more.



- **تشخیص بیماری:**
  - تشخیص آلزایمر از طریق تحلیل تصاویر **MRI**
  - مکان یابی تومورهای سرطانی در تصاویر پزشکی
  - تجزیه و تحلیل دادگان **fMRI**

- **مطالعه فرآیندهای شناختی:**
  - اختلالات توجه و تمرکز در کودکان
  - طراحی سیستم های هجی کننده حروف
  - کمی سازی فرآیند یادگیری در افراد
  - طراحی درمان های شخصی سازی شده برای بیماران مبتلا به افسردگی

محیط های برنامه نویسی پایتون



Google colab VS jupyter

Colab.research.google.com

پکیج MNE در colab

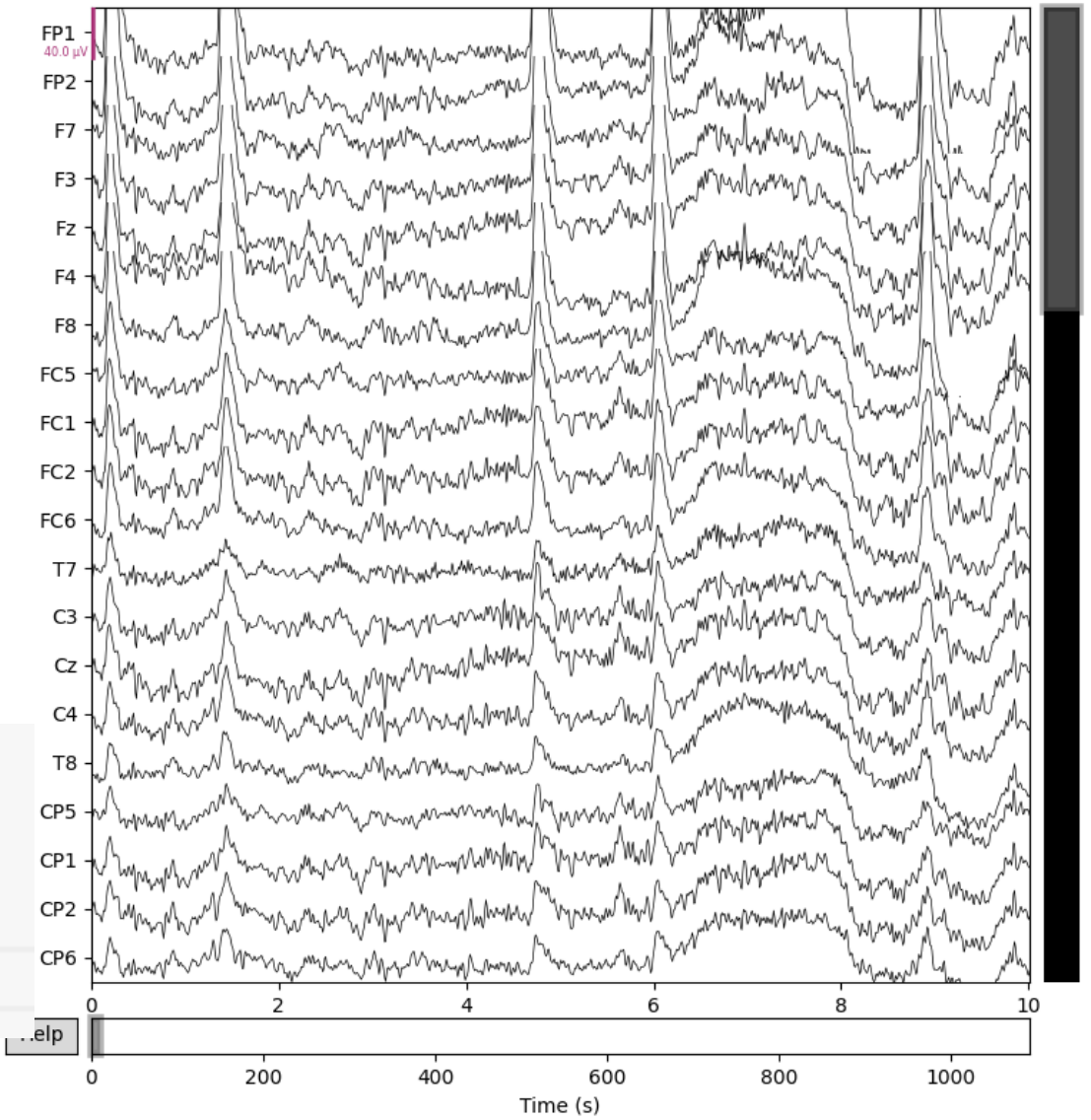
```
pip install mne
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
import mne
import matplotlib.pyplot as plt
```

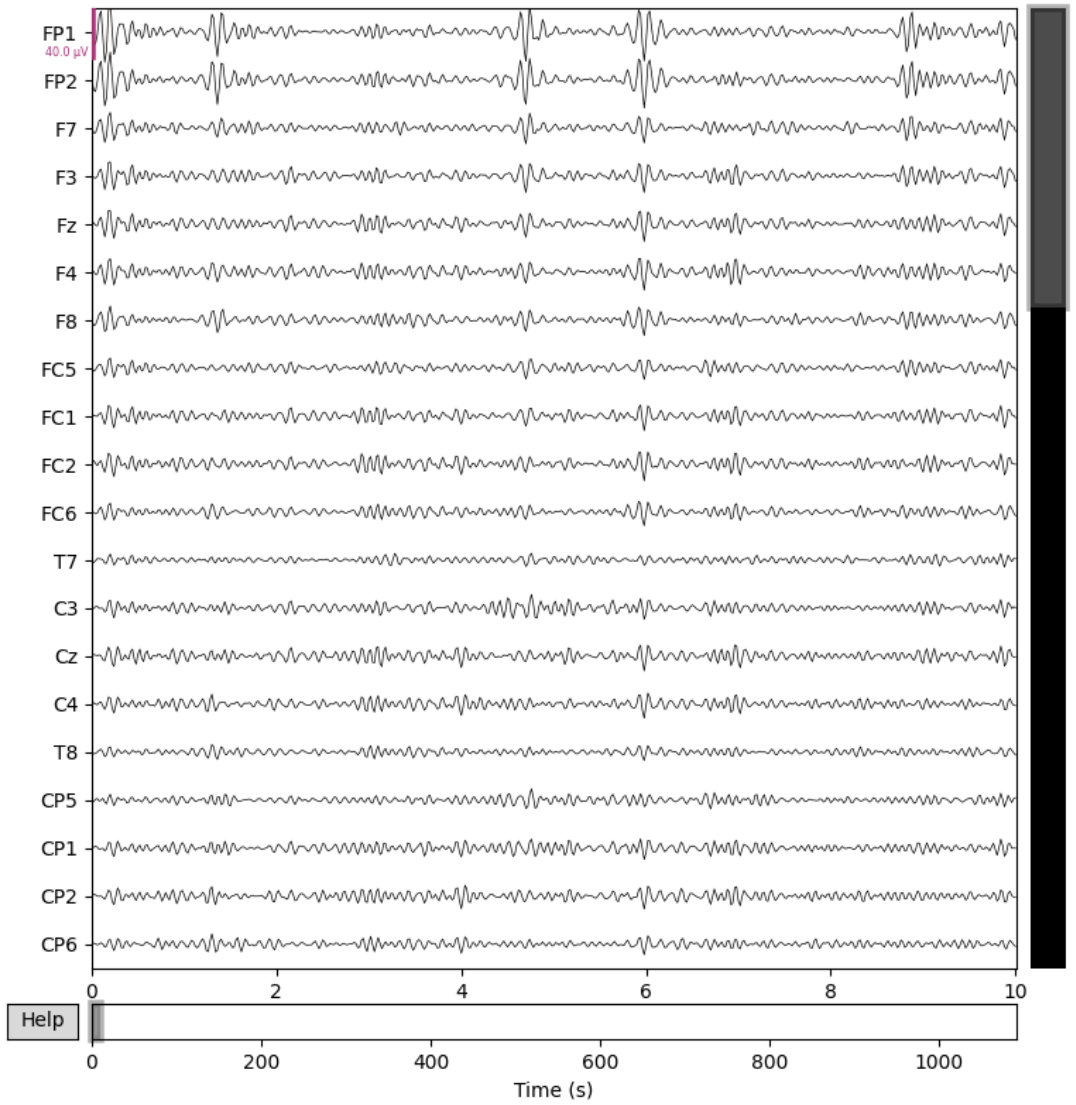
```
fname = "oddball_example_small-fif.gz"
```

```
raw  = mne.io.read_raw_fif(fname,preload=True)
raw.info
raw.plot()
raw.plot_psd()
```

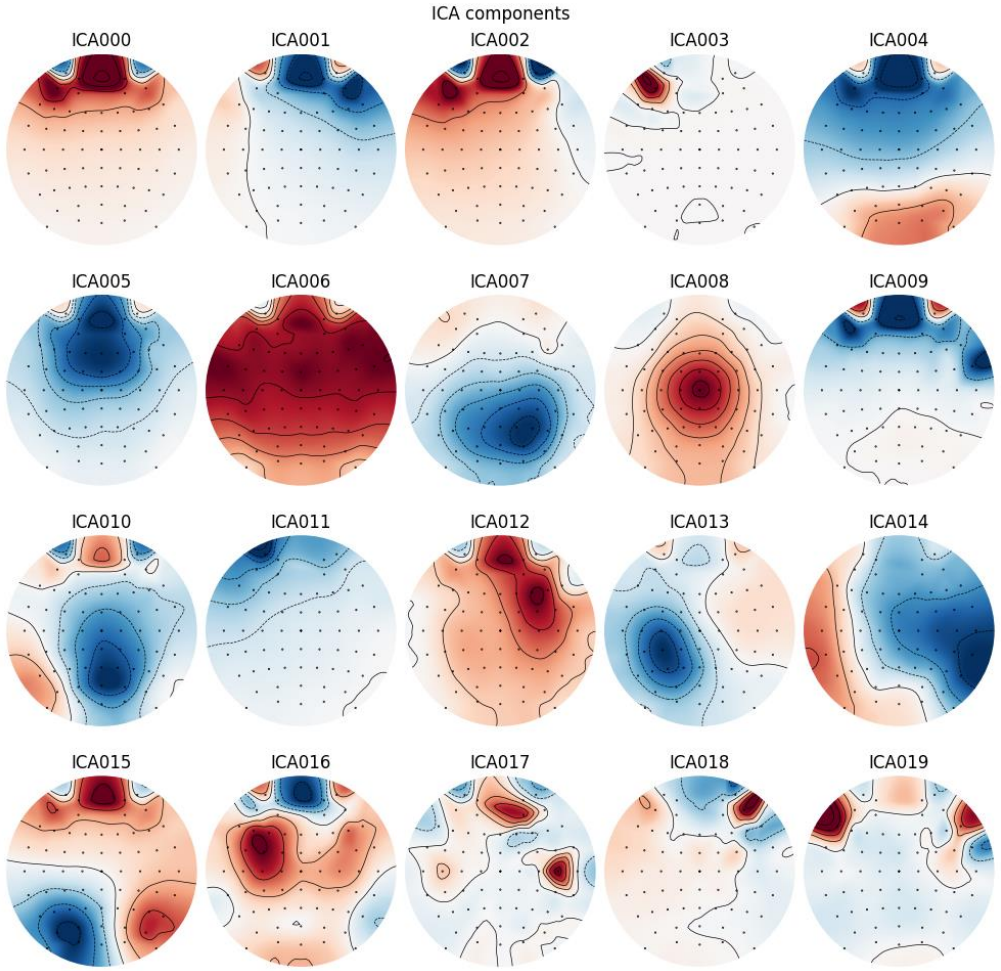


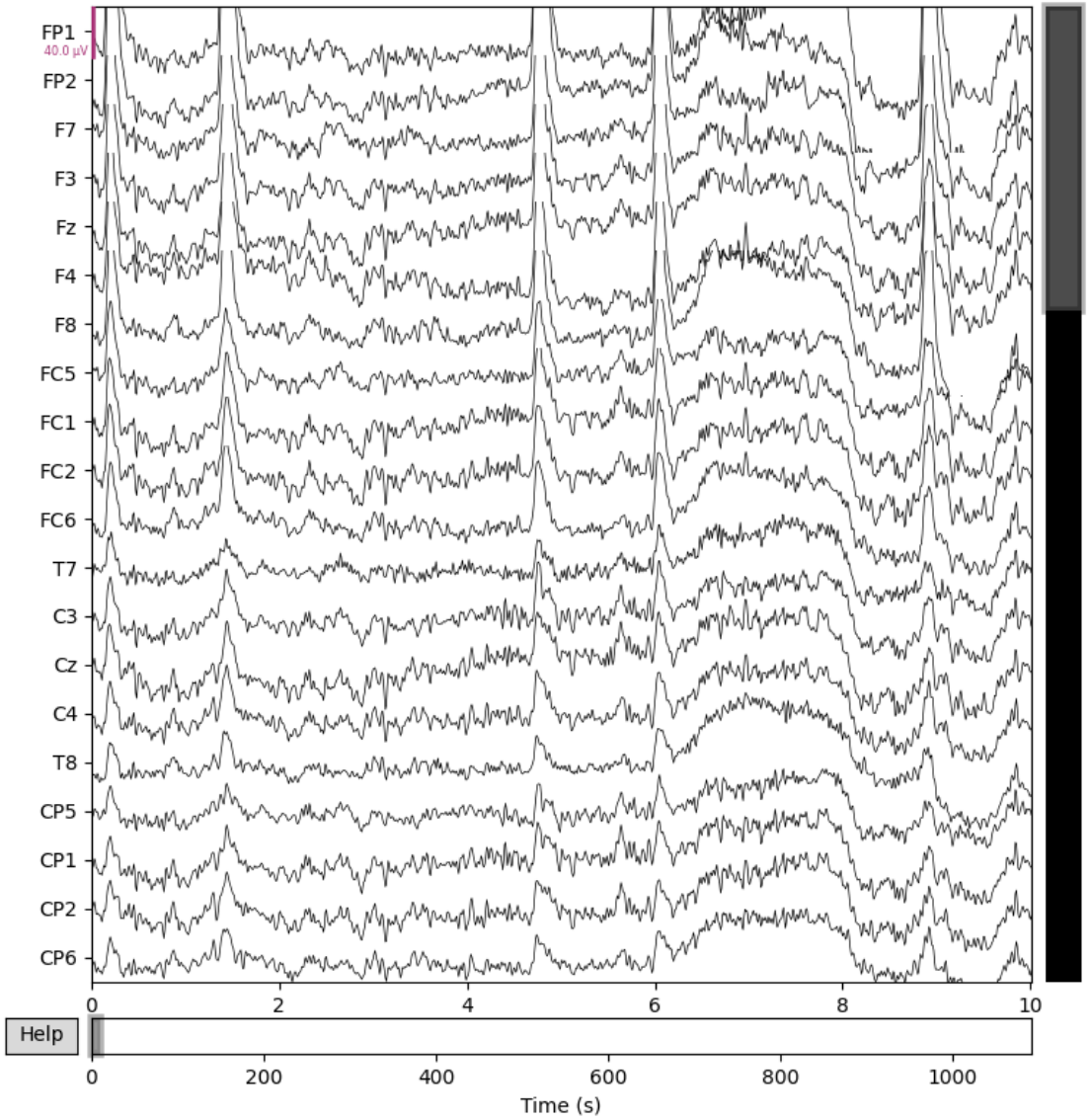
پکیج MNE در colab

```
# raw.filter?
raw.filter(8,14 )
raw.plot()
```

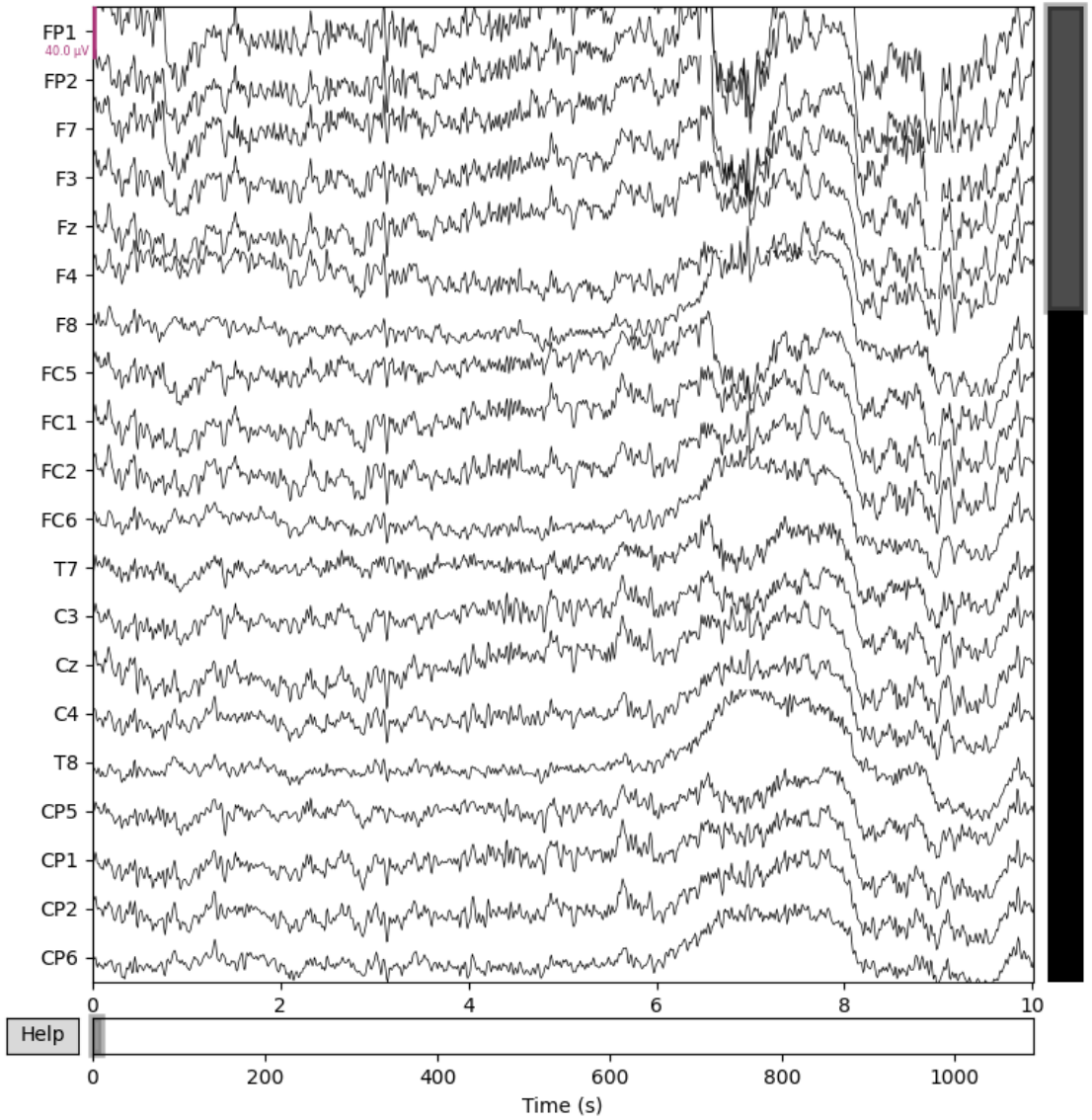


```
# cleaning eye blinks and ECG artifacts
ica = mne.preprocessing.ICA(n_components=20, random_state=0)
ica.fit(raw)
ica.plot_components()

ica.exclude = [0,1]
ica.plot_properties(raw, picks=ica.exclude)
ica.apply(raw)
raw.plot()
```





Before removing eye blink



After removing eye blink

ستاد توسعه فناوری های
هوش مصنوعی و رباتیک

ارزشیابی:
آزمون پایان ترم
تمرین شبیه سازی و تحلیلی
پروژه پایانی و ارائه سمینار

مهلت انتخاب مقاله برای پروژه پایانی:
هفته سوم مهر ماه

Presenter: Dr.Khodabakhshi