

# بوت کمپ «کاربرد پایتون در هوش مصنوعی»

Bootcamp "Applying Python  
to Artificial Intelligence"



# فهرست مطالب

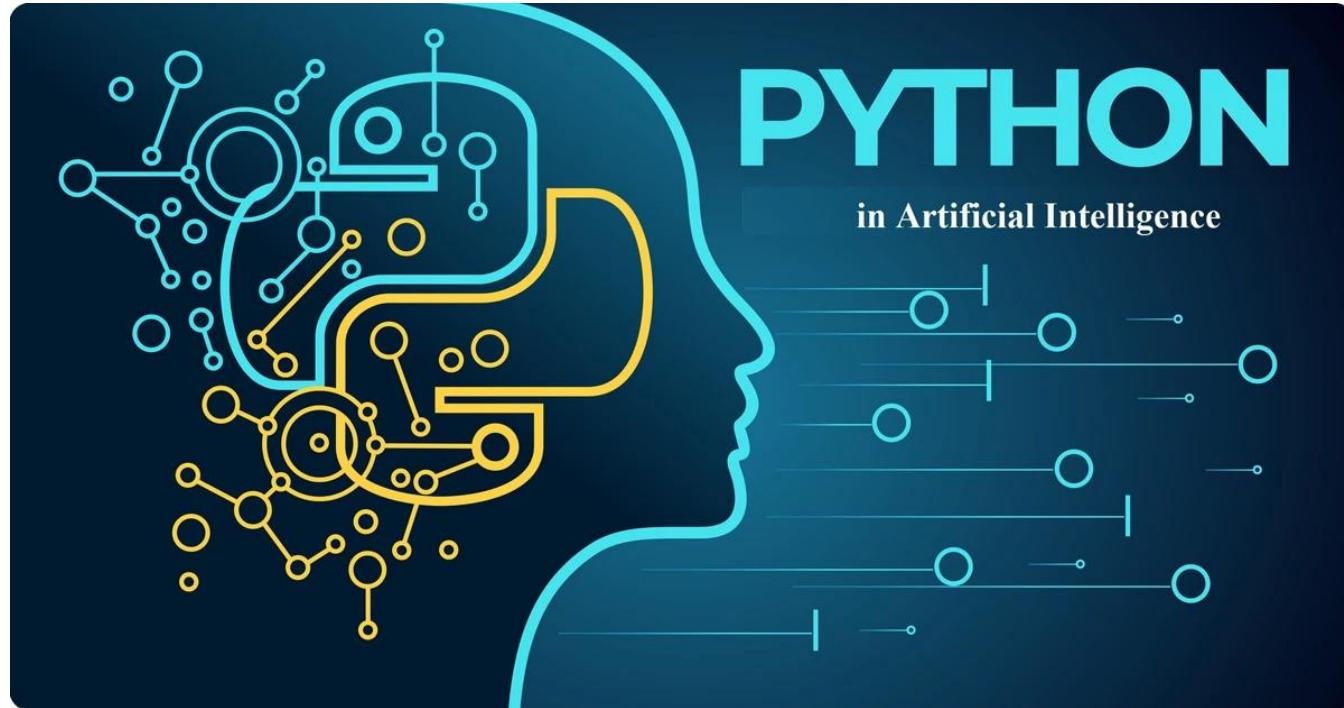
**scikit-learn**

- بخش ۴: یادگیری ماشین با scikit-learn (۵۰ دقیقه)
  - مبانی یادگیری ماشین (۱۰ دقیقه)
  - آشنایی با scikit-learn (۱۰ دقیقه)
  - مثال عملی: رگرسیون خطی (۱۵ دقیقه)
  - مثال عملی: طبقه‌بندی (۱۵ دقیقه)
- بخش ۵: معرفی یادگیری عمیق و کتابخانه‌های آن (۵۰ دقیقه)
  - مبانی یادگیری عمیق (۱۵ دقیقه)
  - معرفی کتابخانه‌های یادگیری عمیق TensorFlow، Keras، PyTorch (۱۵ دقیقه)
  - مثال ساده شبکه عصبی (۲۰ دقیقه)
- بخش ۶: آینده شغلی و کاربردهای عملی (۲۰ دقیقه)
  - بازار کار هوش مصنوعی (۱۰ دقیقه)
  - حوزه‌های کاربردی هوش مصنوعی (۱۰ دقیقه)

- بخش ۱: مقدمه هوش مصنوعی و پایتون (۴۵ دقیقه)
  - معرفی هوش مصنوعی و مفاهیم اساسی (۱۵ دقیقه)
  - چرا پایتون برای هوش مصنوعی؟ (۱۰ دقیقه)
  - معرفی اکوسیستم پایتون برای هوش مصنوعی (۱۰ دقیقه)
  - آشنایی با Google Colab برای یادگیری عملی (۱۰ دقیقه)
- بخش ۲: پردازش داده با Pandas و NumPy (۵۰ دقیقه)
  - بررسی سریع NumPy (۲۰ دقیقه)
  - کار با داده‌ها در Pandas (۲۰ دقیقه)
  - تمرین عملی ساده (۱۰ دقیقه)
- بخش ۳: تحلیل و مصورسازی داده (۳۵ دقیقه)
  - مصورسازی داده با Matplotlib (۱۵ دقیقه)
  - مصورسازی پیشرفته‌تر با Seaborn (۱۰ دقیقه)
  - تمرین عملی (۱۰ دقیقه)
- استراحت (۱۰ دقیقه)



# مقدمه هوش مصنوعی و پایتون



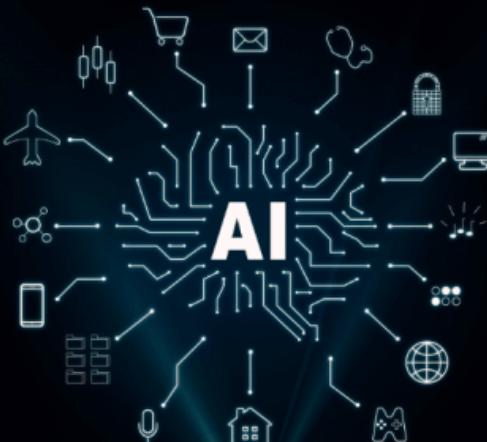
# هوش مصنوعی (AI) چیست؟

Artificial intelligence or AI is the branch of computer science that studies machine intelligence.

## EXAMPLES OF APPLICATIONS

- Search engines (Google)
- Content recommendations (Netflix, YouTube)
- Self-driving vehicles
- Automatic language translation
- Facial recognition
- Computer games
- Spam filters

An AI is a computer system that performs tasks that usually require human intelligence.



# AI in everyday life

From sources across the web



Voice assistants



Facial recognition



Smart home



Chatbots



Navigation



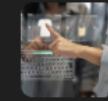
Social Media



E-Commerce



Google Maps



AI in online shopping



Autonomous vehicles



Cybersecurity



AI in banking and finance



Fraud detection



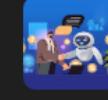
Healthcare



Search engines



AI in social media



Banking



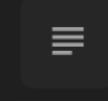
Education



Video game



Robotics



AI in healthcare



Automobiles



Weather forecasting

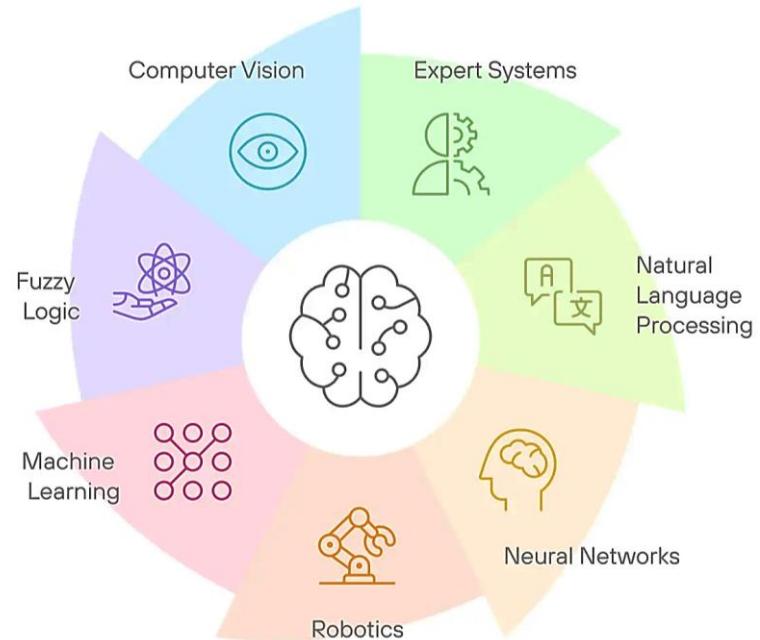


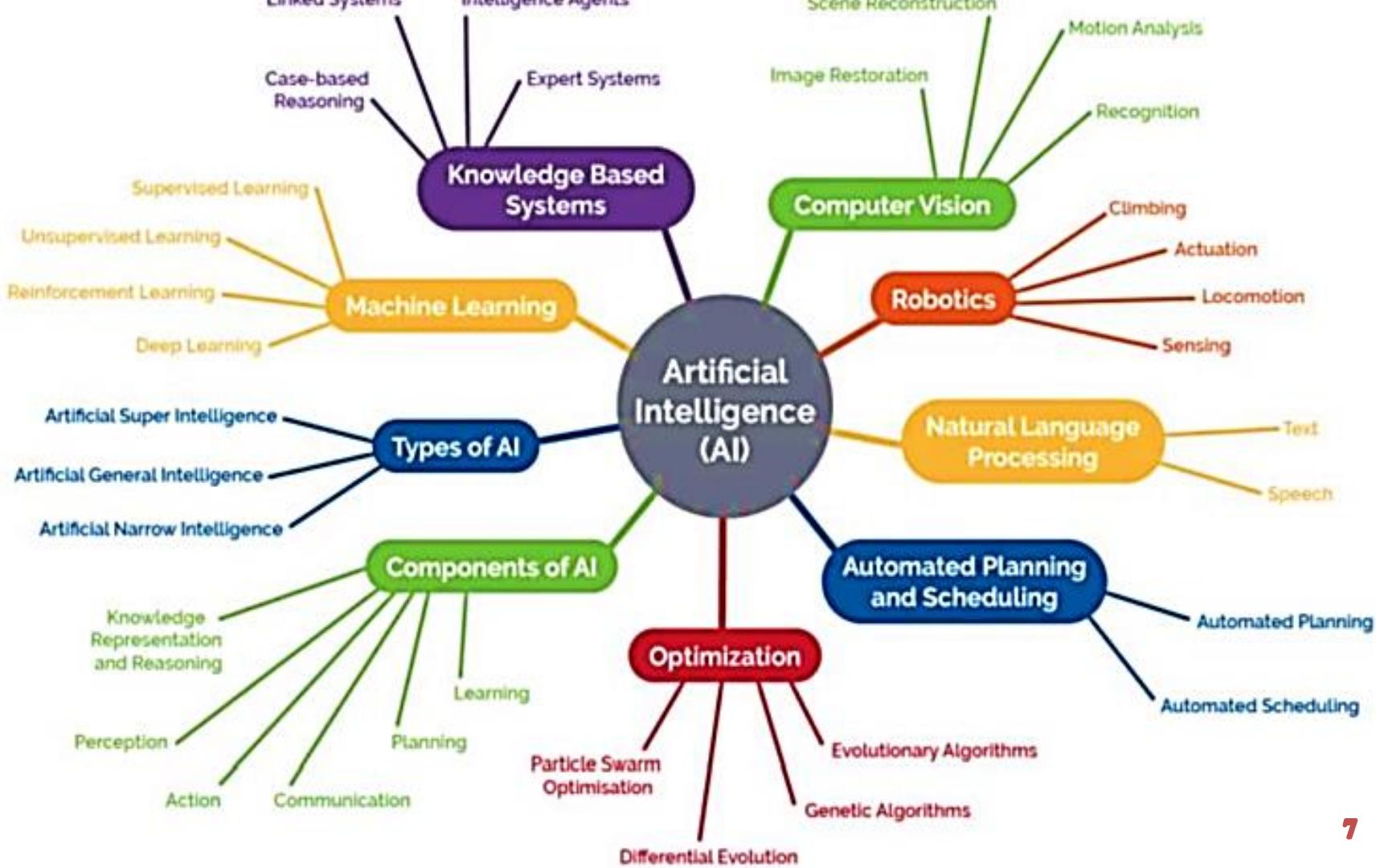
Write an email

# مفاهیم اساسی هوش مصنوعی

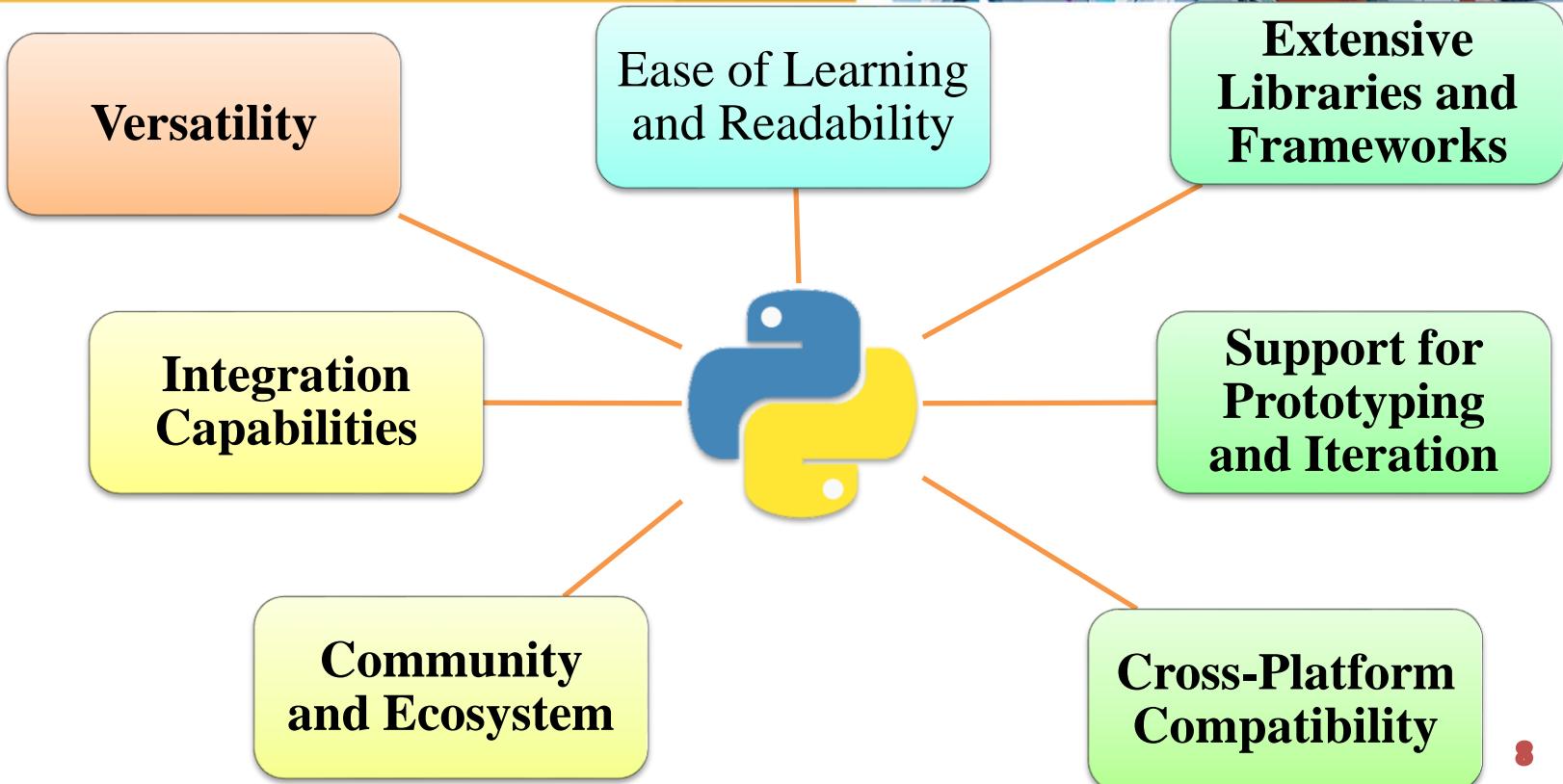


## Top Branches of AI

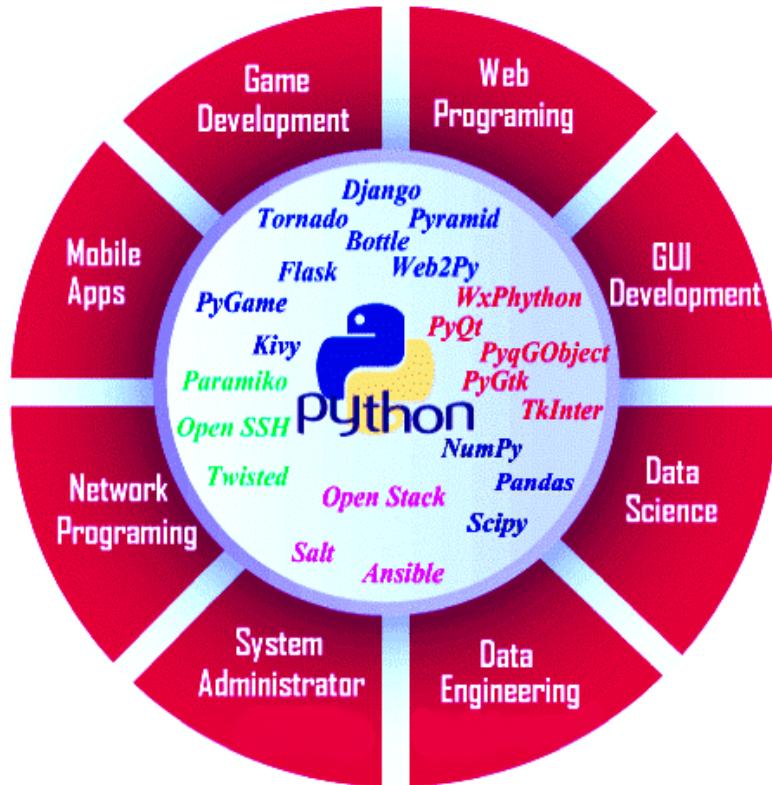




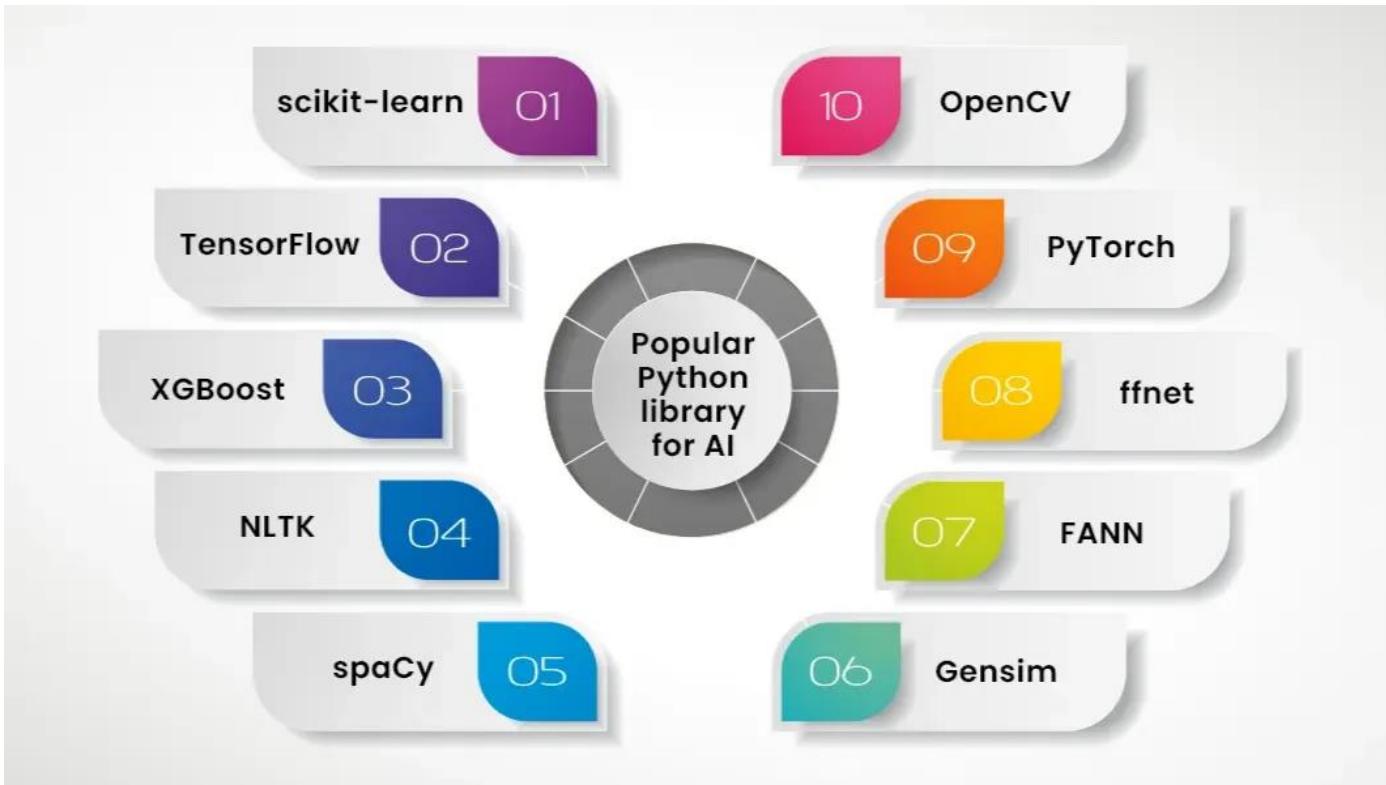
# چرا پایتون برای هوش مصنوعی؟



# اکوسیسٹم پایتون



# اکوسیستم پایتون برای هوش مصنوعی



# Top 10 Python IDE and Code Editors



Visual Studio Code

Sublime Text

Vim

GNU Emacs



SPYDER



Atom



Jupyter



Eclipse



IntelliJ IDEA



Notepad++

# Google Colab



**Google Colab for Python** 





**Exercise\_1.ipynb**  

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Connect  

**Release notes** 

Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

**2025-04-09**

- Colab Data Science Agent [launched!](#)
- Julia language runtimes available with GPU and TPU support [GitHub](#).
- Now your popular Kaggle datasets are cached for quick retrieval.
- Upgraded Colab runtimes to Python 3.11.

**Python package upgrades**

- bigframes 1.29.0 → 1.42.0
- TensorFlow 2.17.1 → 2.18.0
- tensorboard 2.17.1 → 2.18.0
- keras 3.5.0 → 3.8.0
- torch 2.5.1 → 2.6.0
- torchaudio 2.5.1 → 2.6.0
- torchvision 0.20.1 → 0.21.0
- fastai 2.7.18 → 2.7.19
- ipykernel 5.5.0 → 6.17.1
- google-genai 0.3.0 → 1.9.0
- google-auth 2.27.0 → 2.38.0
- Tornado 6.3.3 → 6.4.2
- jax 0.4.33 → 0.5.2
- accelerate 1.2.1 → 1.5.2
- transformers 4.47.1 → 4.50.3
- openai 1.57.4 → 1.70.0
- kagglehub 0.3.6 → 0.3.11
- earthengine-api 1.4.3 → 1.5.9
- google-cloud-bigquery 3.29.0 → 3.31.0

{ } Variables

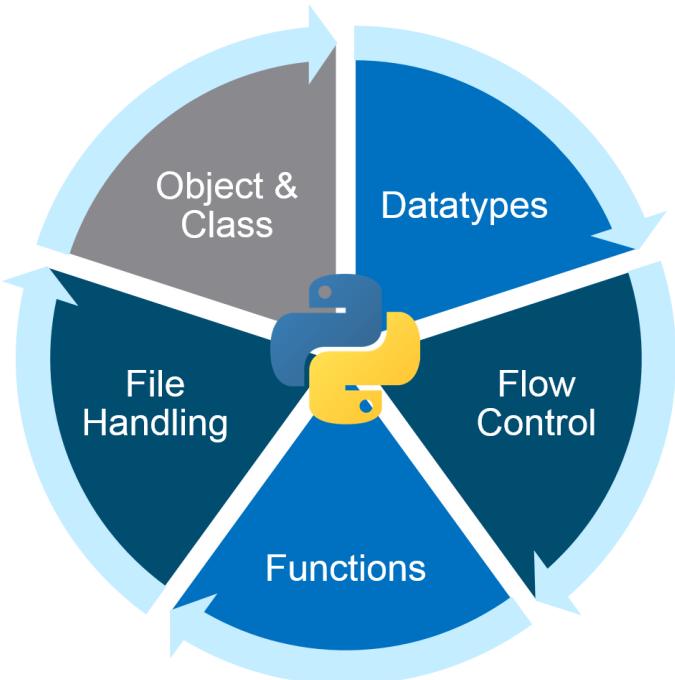
# آشنایی با مبانی پایتون



## Python Programming Fundamentals



# مبانی پایتون

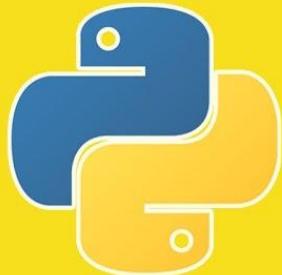


## FUNDAMENTALS

Datatypes	Flow Control	Functions	File Handling	Object & Class
Numbers Strings Lists Dictionaries	If Else For While Continue	Definition Function Call Docstring Return	Reading Writing Editing	Variables Functions

تمرين پايتون

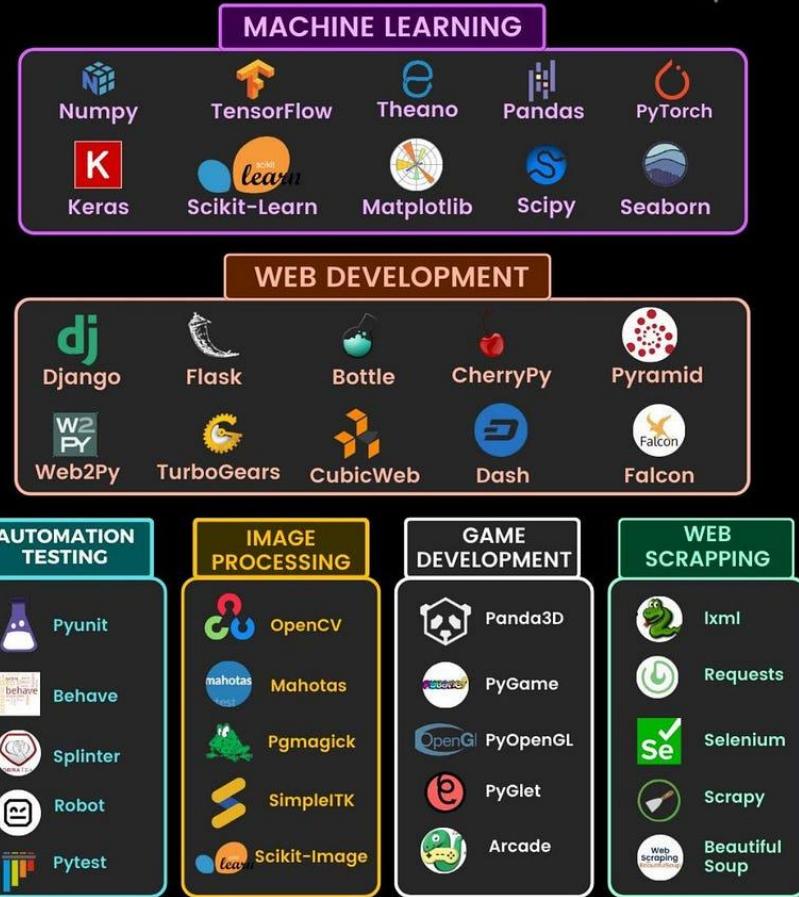
PYTHON



Exercises



# PYTHON LIBRARIES AND FRAMEWORKS



# پردازش داده با Pandas و NumPy

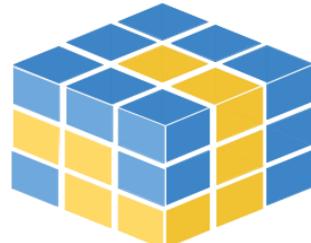
NumPy



Pandas



# آشنایی با NumPy (Numerical Python)



# NumPy

- کتابخانه اساسی پایتون برای محاسبات عددی و علمی
- پشتیبانی از آرایه‌ها و ماتریس‌های بزرگ و چندبعدی
- طیف گسترده‌ای از توابع ریاضی برای کار مؤثر روی این آرایه‌ها
- data analysis, machine learning, and scientific research, as it allows for fast and memory-efficient array operations



## NumPy Array Creation Methods

Method	Description
<code>np.array(&lt;list&gt;)</code>	NumPy array from Python list
<code>np.array(&lt;list-of-lists&gt;)</code>	NumPy array from list of lists
<code>np.array(&lt;pandas-series&gt;)</code>	NumPy array from PD Series
<code>df.values</code>	NumPy array from DataFrame
<code>np.zeros(&lt;size&gt;)</code>	NumPy array of all zeros
<code>np.ones(&lt;size&gt;)</code>	NumPy array of all ones
<code>np.eye(&lt;size&gt;)</code>	Identity NumPy array
<code>np.arange(&lt;start&gt;, &lt;stop&gt;, &lt;step&gt;)</code>	Equally spaced NumPy array with specific step
<code>np.linspace(&lt;start&gt;, &lt;stop&gt;, &lt;count&gt;)</code>	Equally spaced NumPy array with specific size
<code>np.random.randint(&lt;low&gt;, &lt;high&gt;, &lt;size&gt;)</code>	NumPy array of random ints
<code>np.random.random(&lt;size&gt;)</code>	NumPy array of random floats

## NumPy Array Manipulation Methods

Method	Description
<code>array.reshape(&lt;new-shape&gt;)</code>	Reshape NumPy Array
<code>array.transpose() OR array.T</code>	Transpose NumPy Array
<code>np.concatenate(&lt;np-arrays&gt;, &lt;axis&gt;)</code>	Concatenate NumPy Arrays
<code>np.flatten(&lt;Nd-np-array&gt;)</code>	Flatten a NumPy Array
<code>np.unique(&lt;np-array&gt;, &lt;axis&gt;)</code>	Find unique elements
<code>array.tolist()</code>	NumPy Array to List

## Search Methods

Method	Description
<code>np.argmax(&lt;np-array&gt;, &lt;axis&gt;)</code>	Max Element Index
<code>np.argmin(&lt;np-array&gt;, &lt;axis&gt;)</code>	Min Element Index
<code>np.where(&lt;condition&gt;, &lt;true-return-value&gt;, &lt;false-return-value&gt;)</code>	Conditional Search and Replacement
<code>np.nonzero(&lt;np-array&gt;)</code>	Index of non-zero elements

## Mathematical Operations

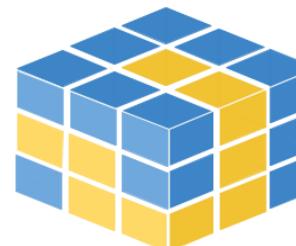
Method	Description
<code>np.sin(&lt;np-array&gt;)</code>	
<code>np.cos(&lt;np-array&gt;)</code>	Trigonometric Functions
<code>np.tan(&lt;np-array&gt;)</code>	
<code>np.floor(&lt;np-array&gt;)</code>	Element-wise floor value
<code>np.ceil(&lt;np-array&gt;)</code>	Element-wise ceiling value
<code>np.rint(&lt;np-array&gt;)</code>	Round to nearest int
<code>np.round(&lt;np-array&gt;, &lt;decimal-places&gt;)</code>	Round to decimal places
<code>np.exp(&lt;np-array&gt;)</code>	Element-wise exponent
<code>np.log(&lt;np-array&gt;)</code>	Element-wise logarithm
<code>np.sqrt(&lt;np-array&gt;)</code>	Element-wise square root
<code>np.sum(&lt;np-array&gt;, &lt;axis&gt;)</code>	Sum along an axis
<code>np.mean(&lt;np-array&gt;, &lt;axis&gt;)</code>	Mean along an axis
<code>np.std(&lt;np-array&gt;, &lt;axis&gt;)</code>	Std. dev along an axis

## Matrix and Vector Operations

Method	Description
<code>np.dot(&lt;np-array1&gt;, &lt;np-array2&gt;)</code>	Dot Product
<code>np.matmul(&lt;np-array1&gt;, &lt;np-array2&gt;)</code>	Matrix Multiplication
<code>np.array1 @ np.array2</code>	
<code>np.linalg.norm(&lt;np-array&gt;)</code>	Vector Norm

## Sorting Methods

Method	Description
<code>np.sort(&lt;np-array&gt;, &lt;axis&gt;)</code>	Sort Array
<code>np.argsort(&lt;np-array&gt;, &lt;axis&gt;)</code>	Return the order of indices that sort the array



# NumPy

# آشنایی با Pandas



یک کتابخانه قدرتمند برای تحلیل، «پیش‌پردازش» (PreProcessing) و «بصری‌سازی» (Visualization) داده‌ها

ابزار علم مهندسی داده

پشتیبانی از فرمتهای گوناگونی شامل CSV, TSV, MS Excel

دو ساختار اصلی برای ذخیره‌سازی داده‌ها است که عبارتند از:

Series  
(DataFrame)

# Pandas با آشنایی



## Applications of Pandas

# تمرین ساده عملی



# تحلیل و مصورسازی داده

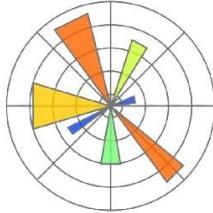


*matplotlib*



*seaborn*





**matplotlib**

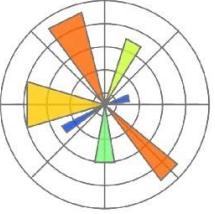
# Matplotlib



- در پایتون، می‌توانید از مازول‌ها یا کتابخانه‌های مختلفی برای مصورسازی داده‌ها استفاده کنید.
- هدف از مصورسازی داده‌ها، ارائه داده‌ها به صورت ساده‌تر، مانند نمودار پراکندگی، نمودار چگالی، نمودار میله‌ای و غیره است.
- تجسم داده‌ها با استفاده از Matplotlib در سبک‌های مختلف ترسیم

# matplotlib

Cheat sheet  
Version 3.7.4



# matplotlib

## Quick start

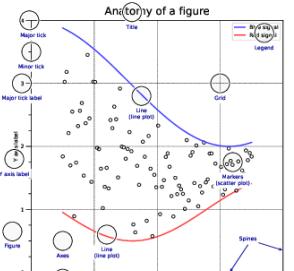
```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt

X = np.linspace(0, 2*np.pi, 100)
Y = np.cos(X)

fig, ax = plt.subplots()
ax.plot(X, Y, color='green')

fig.savefig("figure.pdf")
plt.show()
```

## Anatomy of a figure



## Subplots layout

```
subplot[s](rows, cols, ...)
fig, axs = plt.subplots(3, 3)

G = gridspec(rows,cols, ...)
ax = G[0, :]

ax.inset_axes(extent)

d=make_axes_locatable(ax)
ax = d.new_horizontal('10%')
```

## Getting help

- matplotlib.org
- github.com/matplotlib/matplotlib/issues
- discourse.matplotlib.org
- stackoverflow.com/questions/tagged/matplotlib
- https://gitter.im/matplotlib/matplotlib
- twitter.com/matplotlib
- matplotib.org/gallery/index.html

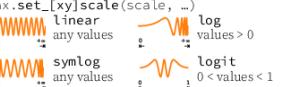
## Basic plots



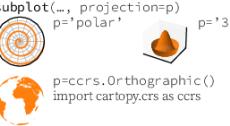
## Advanced plots



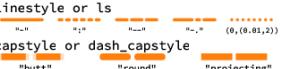
## Scales



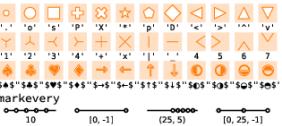
## Projections



## Lines



## Markers



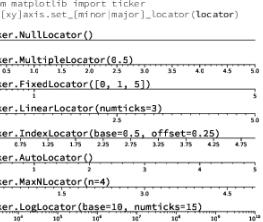
## Colors



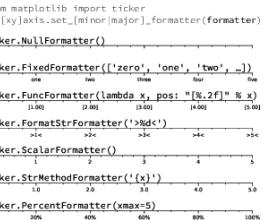
## Colormaps



## Tick locators

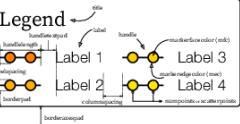


## Tick formatters



## Ornaments

```
ax.legend(...)
handles, labels, loc, title, frameon
```



```
ax.colorbar(...)
mappable, ax, cax, orientation
```

```
ax.annotate(...)
text, xy, xytext, xycoords, textcoords, arrowprops
```

```
text
text
textcoords
xy
xycoords
```

## Event handling

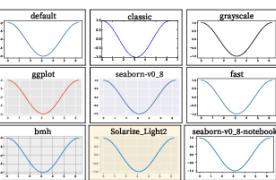
```
fig, ax = plt.subplots()
def on_click(event):
    print(event)
fig.canvas.mpl_connect('button_press_event', on_click)
```

## Animation

```
import matplotlib.animation as mpl
T = np.linspace(0, 2*np.pi, 100)
S = np.sin(T)
line, = plt.plot(T, S)
def animate(i):
    line.set_ydata(np.sin(T+i/50))
anim = mpl.FuncAnimation(
    plt.gcf(), animate, interval=50)
plt.show()
```

## Styles

```
plt.style.use(style)
```



## Quick reminder

```
ax.grid()
ax.set_xlim(vmin, vmax)
ax.set_xlabel(label)
ax.set_xticks(ticks, [labels])
ax.set_xticklabels(labels)
ax.set_title(title)
ax.tick_params(width=10, ...)
ax.set_axis_on/off()
```

```
fig.suptitle(title)
fig.tight_layout()
plt.gcf(), plt.gca()
mpl.rc('axes', linewidth=1, ...)
[fig, ax].patch.set_alpha(0)
text=r'$\frac{e^{-i\pi}}{2^n}$'
```

## Keyboard shortcuts

<b>ctrl + s</b>	Save	<b>ctrl + w</b>	Close plot
<b>r</b>	Reset view	<b>f</b>	Fullscreen 0/1
<b>f</b>	View forward	<b>b</b>	View back
<b>p</b>	Pan view	<b>z</b>	Zoom to rect
<b>x</b>	X pan/zoom	<b>y</b>	Y pan/zoom
<b>g</b>	Minor grid 0/1	<b>G</b>	Major grid 0/1
<b>l</b>	X axis log/linear	<b>L</b>	Y axis log/linear

## Ten simple rules

- Know your audience
- Identify your message
- Adapt the figure
- Captions are not optional
- Do not trust the defaults
- Use color effectively
- Do not mislead the reader
- Avoid "chartjunk"
- Message trumps beauty
- Get the right tool

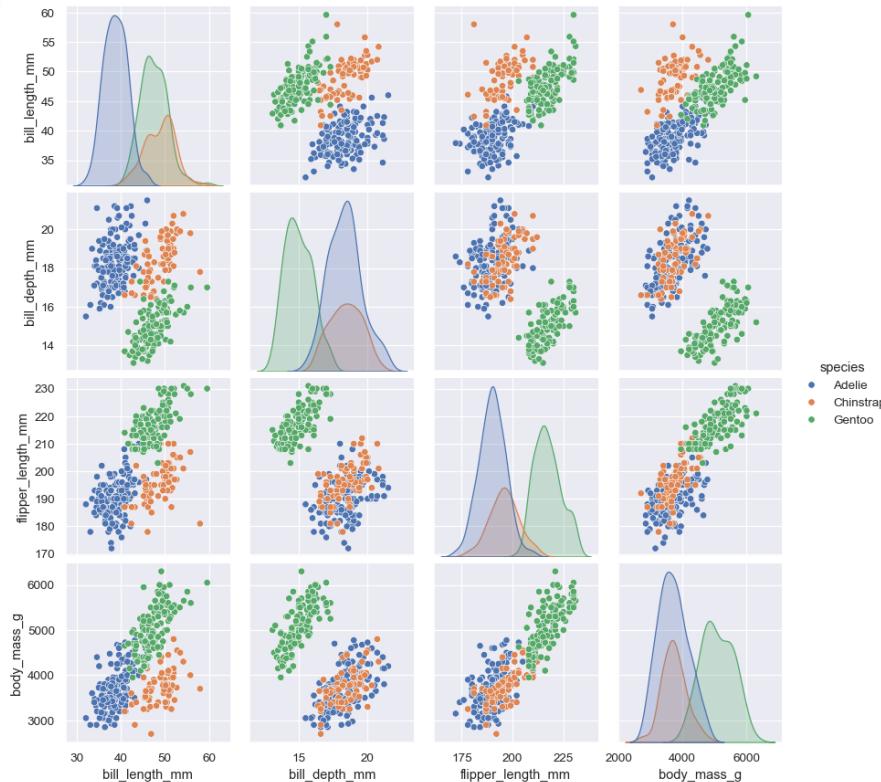
READ



# seaborn



- کتابخانه‌ای برای ساخت نمودارهای آماری
- این کتابخانه بر پایه matplotlib ساخته شده و با ساختارهای داده pandas به طور نزدیکی ادغام می‌شود.





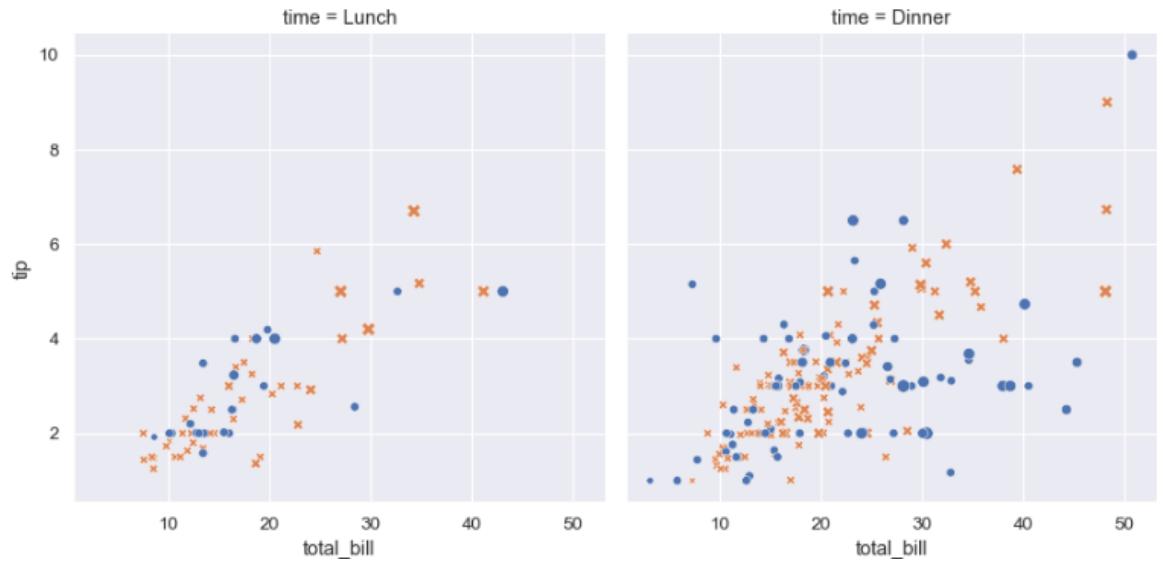
# seaborn

```
# Import seaborn
import seaborn as sns

# Apply the default theme
sns.set_theme()

# Load an example dataset
tips = sns.load_dataset("tips")

# Create a visualization
sns.relplot(
    data=tips,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```



# تمرین عملی



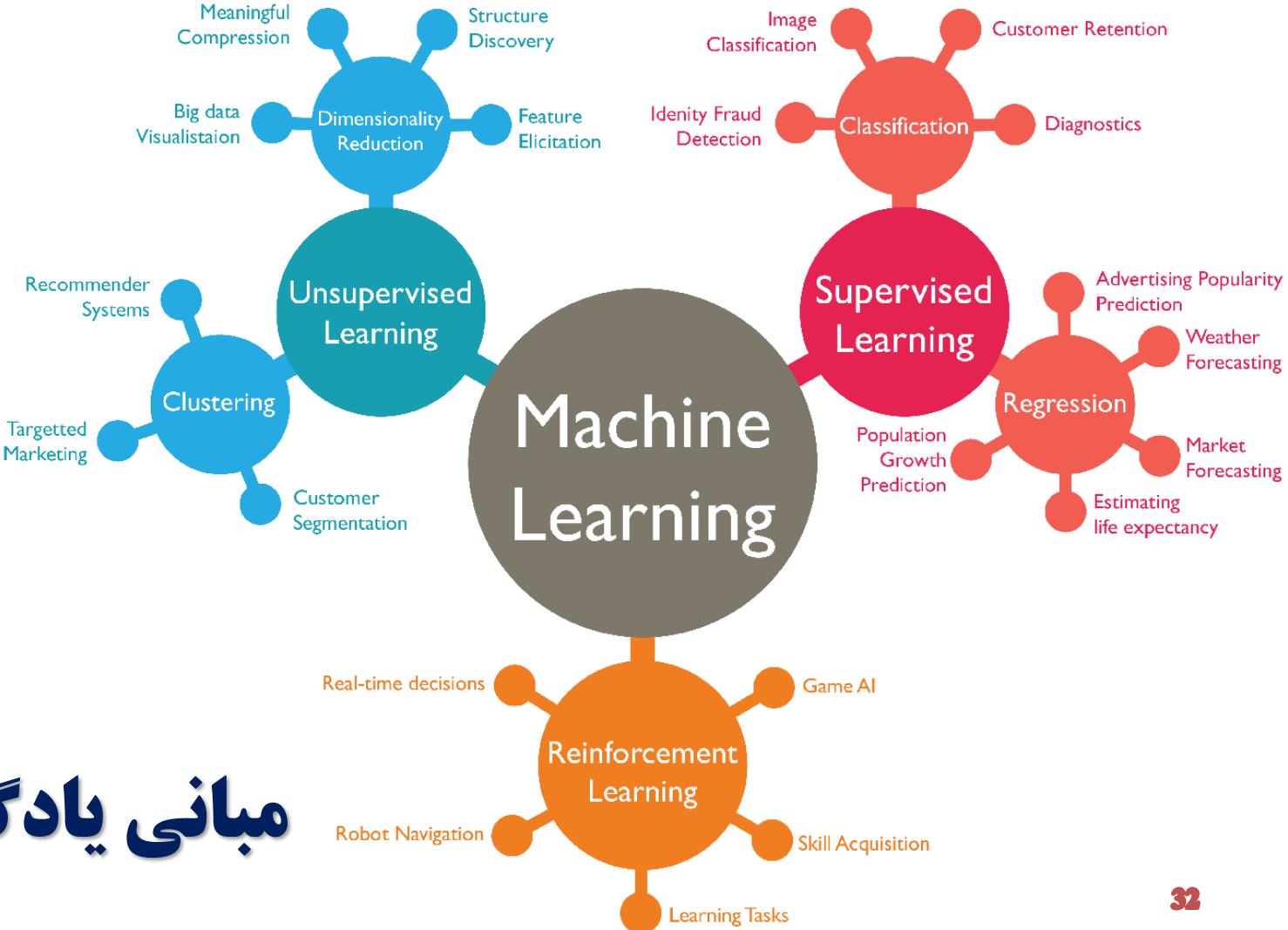
# استراحة (١٠ دقائق)



# یادگیری ماشین با scikit-learn



# مبانی یادگیری ماشین



# آشنایی با scikit-learn



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



## Classification

Identifying to which category an object belongs to.

**Applications:** Spam detection, Image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, ...

— Examples

## Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, ridge regression, Lasso, ...

— Examples

## Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, ...

— Examples

## Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency

**Algorithms:** PCA, feature selection, non-negative matrix factorization.

— Examples

## Model selection

Comparing, validating and choosing parameters and models.

**Goal:** Improved accuracy via parameter tuning

**Modules:** grid search, cross validation, metrics.

— Examples

## Preprocessing

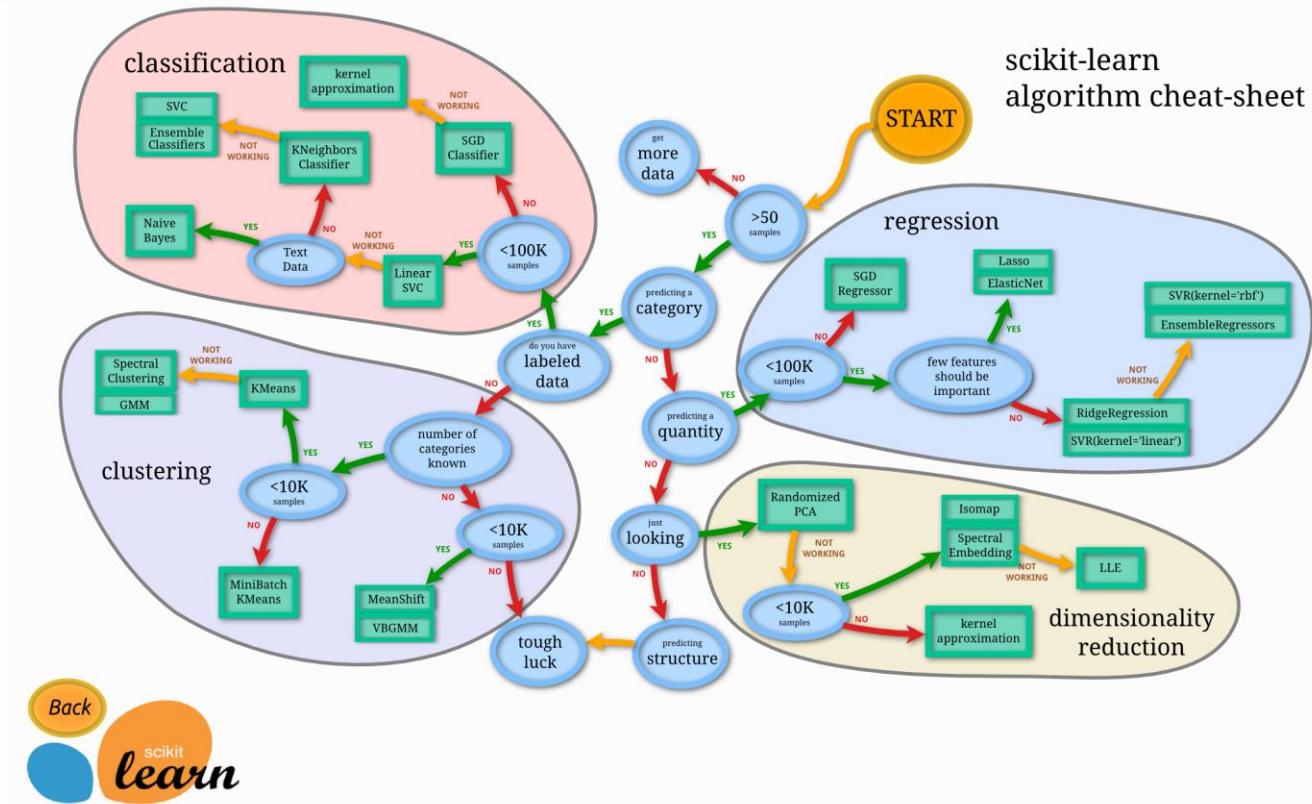
Feature extraction and normalization.

**Application:** Transforming input data such as text for use with machine learning algorithms.

**Modules:** preprocessing, feature extraction.

— Examples

# آشنایی با scikit-learn



# مثال عملی: رگرسیون خطی

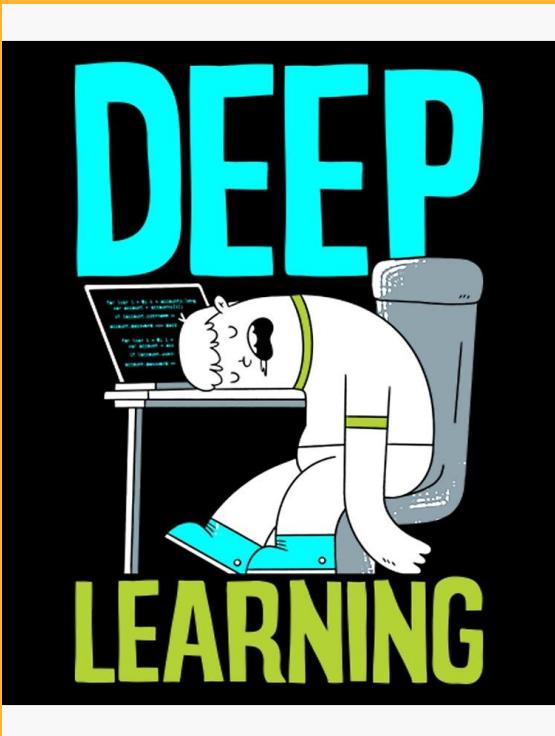


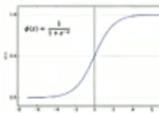
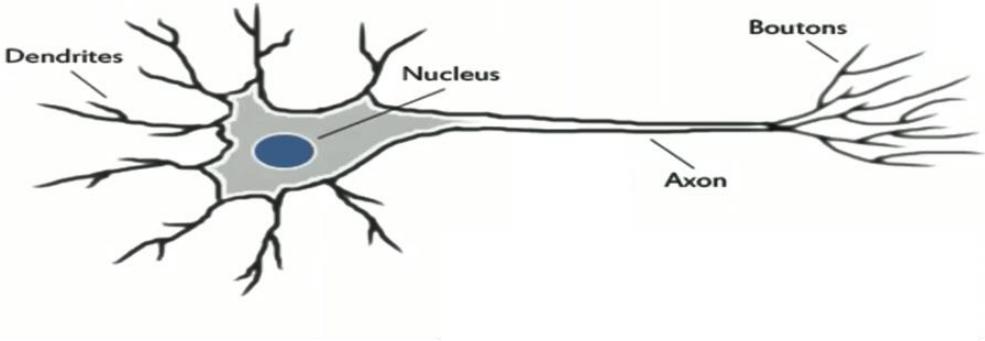
# مثال عملی: طبقه‌بندی

Google Colab for Python 

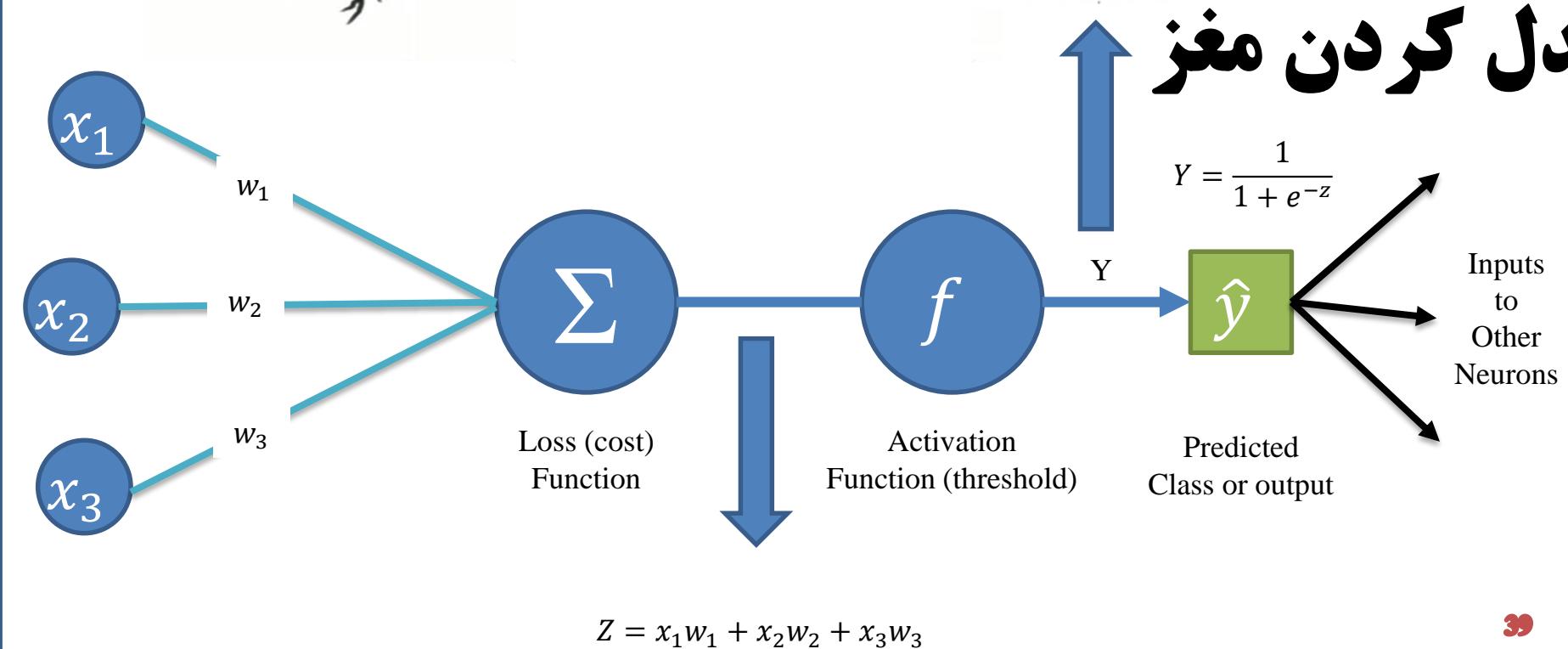


# یادگیری عمیق

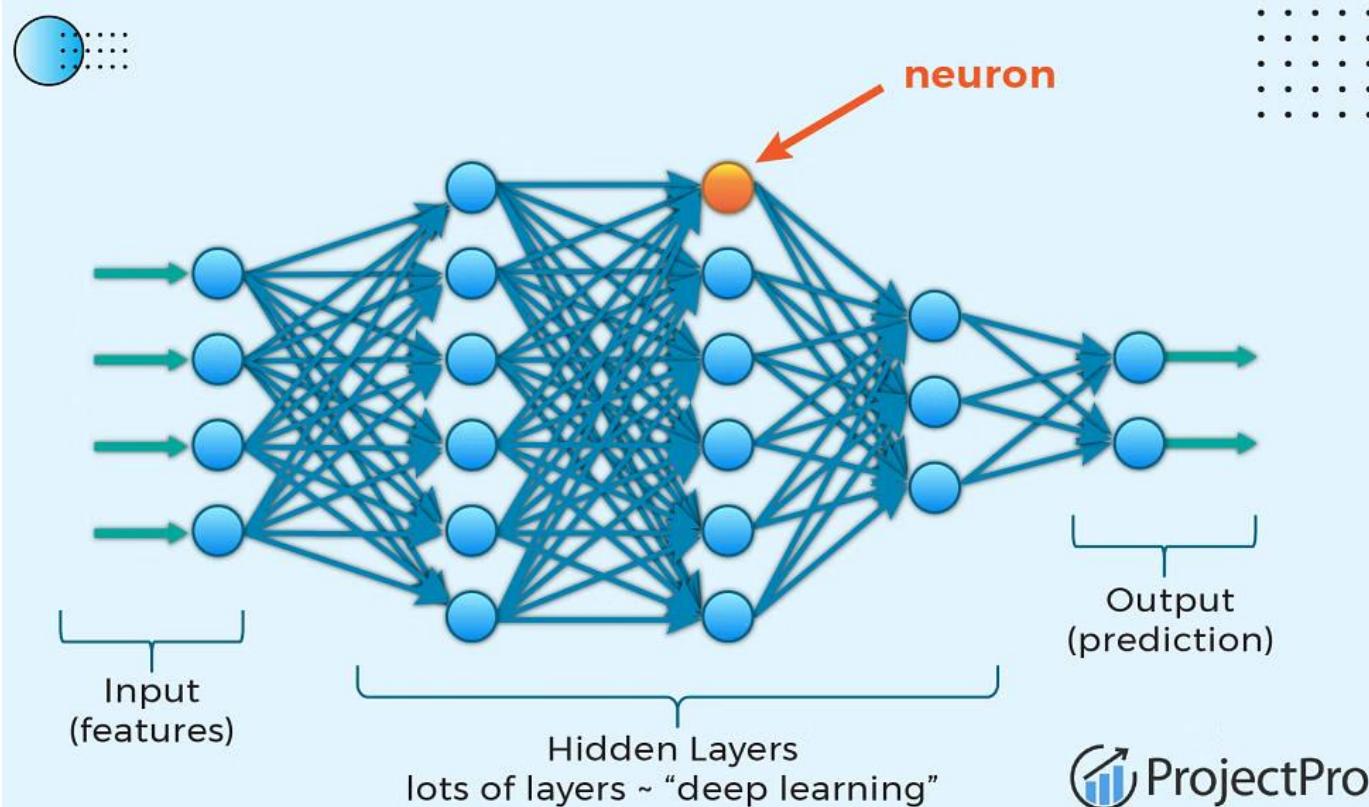




# مدل کردن مغز



# مبانی یادگیری عمیق



# کتابخانه‌های یادگیری عمیق

(Keras ، PyTorch ، TensorFlow)



PYTHON

## BEST PYTHON LIBRARIES FOR MACHINE LEARNING



TensorFlow



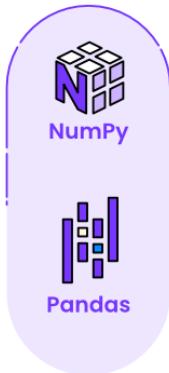
theano

# مثال ساده شبکه عصبی

Google Colab for Python 



Data Manipulation  
and Analysis



Visualization



Machine Learning



Natural Language  
Processing



Deep  
Learning



Computer  
Vision



The Best Python  
Libraries for Machine  
Learning and AI

# آینده شغلی و کاربردهای عملی



- **پزشکی و سلامت**
  - بخش‌بندی مشتریان
  - تشخیص اشیا، چهره‌ها و حالات
  - کاربرد در پزشکی، خودروهای خودران، امنیت تصاویر پزشکی
  - کشف داروهای جدید
  - پیش‌بینی شیوع بیماری‌ها
- **روباتیک و اتوماسیون**
  - روبات‌های صنعتی هوشمند
  - پهپادها و خودروهای خودران
- **پردازش تصویر و بینایی ماشین**
  - تشخیص اشیا، چهره‌ها و حالات
  - کاربرد در پزشکی، خودروهای خودران، امنیت
- **پردازش زبان طبیعی**
  - چتبات‌ها و دستیاران هوشمند
  - ترجمه ماشینی، تحلیل احساسات، خلاصه‌سازی متن
- **تحلیل داده‌های کسب و کار**
  - پیش‌بینی فروش و روندها
  - سیستم‌های توصیه‌گر

# Fastest growing and declining jobs by 2030

Top fastest growing jobs	Top fastest declining jobs
1 Big data specialists	1 Postal service clerks
2 FinTech engineers	2 Bank tellers and related clerks
3 AI and machine learning specialists	3 Data entry clerks
4 Software and applications developers	4 Cashiers and ticket clerks
5 Security management specialists	5 Administrative assistants and executive secretaries
6 Data warehousing specialists	6 Printing and related trades workers
7 Autonomous and electric vehicle specialists	7 Accounting, bookkeeping and payroll clerks
8 UI and UX designers	8 Material-recording and stock-keeping clerks
9 Light truck or delivery services drivers	9 Transportation attendants and conductors
10 Internet of things specialists	10 Door-to-door sales workers, news and street vendors, and related workers
11 Data analysts and scientists	11 Graphic designers
12 Environmental engineers	12 Claims adjusters, examiners and investigators
13 Information security analysts	13 Legal officials
14 DevOps engineers	14 Legal secretaries
15 Renewable energy engineers	15 Telemarketers

Note: The jobs that survey respondents report the highest and lowest net growth (%) by 2030.

Source: World Economic Forum. (2025). Future of Jobs Report 2025.



## Future of Jobs Report 2025: These are the fastest growing and declining jobs

<https://www.weforum.org/stories/2025/01/future-of-jobs-report-2025-the-fastest-growing-and-declining-jobs/>

# بازار کار هوش مصنوعی



## توسعه‌دهنده‌ی نرم‌افزار

برنامه نویسان دارای آینده شغلی در ایران و جهان هستند، شما می‌دانید مهندسان نرم‌افزار و توسعه‌دهنگان در این لیست هستند، اما بهزودی فقط دانش ساده‌ی کدنویسی کافی نخواهد بود. یادگیری ماشین (Machine Learning) و هوش مصنوعی بهزودی جای برنامه‌نویسان تازه‌کار را خواهند گرفت؛ بنابراین، مهندسان نرم‌افزار برای ادامه‌ی کار به افزودن یادگیری ماشین به مهارت‌های خود نیاز دارند.

مهندس نرم‌افزار مطلوب در آینده دارای مهارت‌های سطح بالا مانند ساختمان داده‌ها و درک هوش مصنوعی است؛ توسعه‌دهنگان بیشتر وقت خود را برای ادغام شدن با سایر سرویس‌ها صرف خواهند کرد و زمان کمتری برای نوشتن کدی خاص می‌گذارند. کدنویسان خودآموز بخش بزرگی از نیروی کار خواهند بود.

# حوزه‌های کاربردی هوش مصنوعی



ماشین

یادگیری  
مهندس  
(Machine Learning Engineer)  
(Data Scientist)  
دانشمند داده

ماشین

بینایی  
متخصص  
(Computer Vision Specialist)

طبیعی

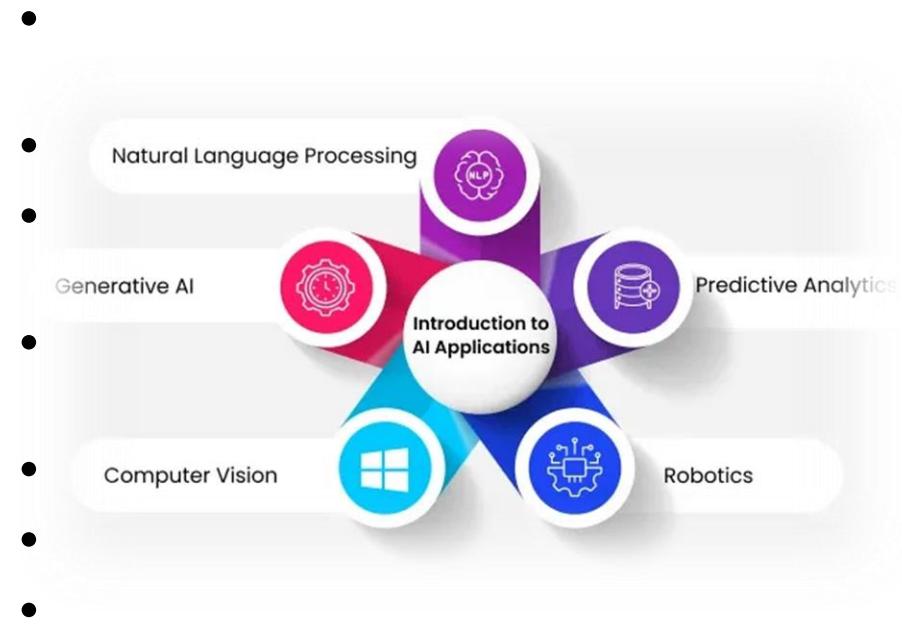
پردازش زبان  
متخصص  
(NLP Specialist)

مهندس داده  
(Data Engineer)

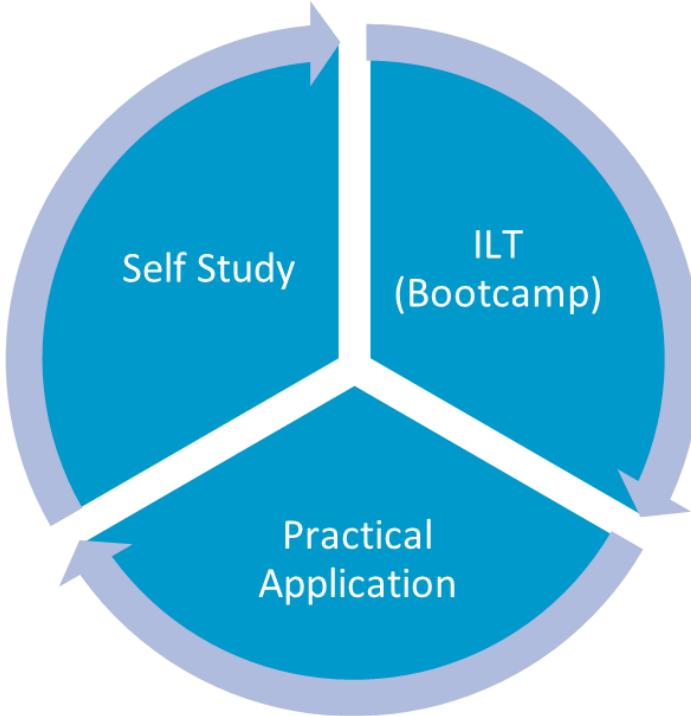
تحقیق هوش مصنوعی  
(AI Researcher)

مصنوعی

اخلاق هوش  
متخصص  
(AI Ethics Specialist)



# مسیر یادگیری پیشنهادی پس از این دوره



- تقویت مبانی پایتون و آمار (NumPy, Pandas)
- تسلط بر کتابخانه‌های تحلیل داده (scikit-learn, TensorFlow, PyTorch)
- انتخاب یک کتابخانه یادگیری عمیق (TensorFlow, PyTorch) و تسلط بر آن
- انجام پروژه‌های عملی در یک حوزه خاص
- مشارکت در پروژه‌های متن باز یا چالش‌های kaggle

با شکر از شما

