

# Strojové Učenie

## Python Scikit-learn



Machine Learning

# Ako Začneme?

1. Pridajte si ma na **LinkedIn**

[www.linkedin.com/in/miroslav-reiter](http://www.linkedin.com/in/miroslav-reiter)

2. Stiahnite si cvičenia

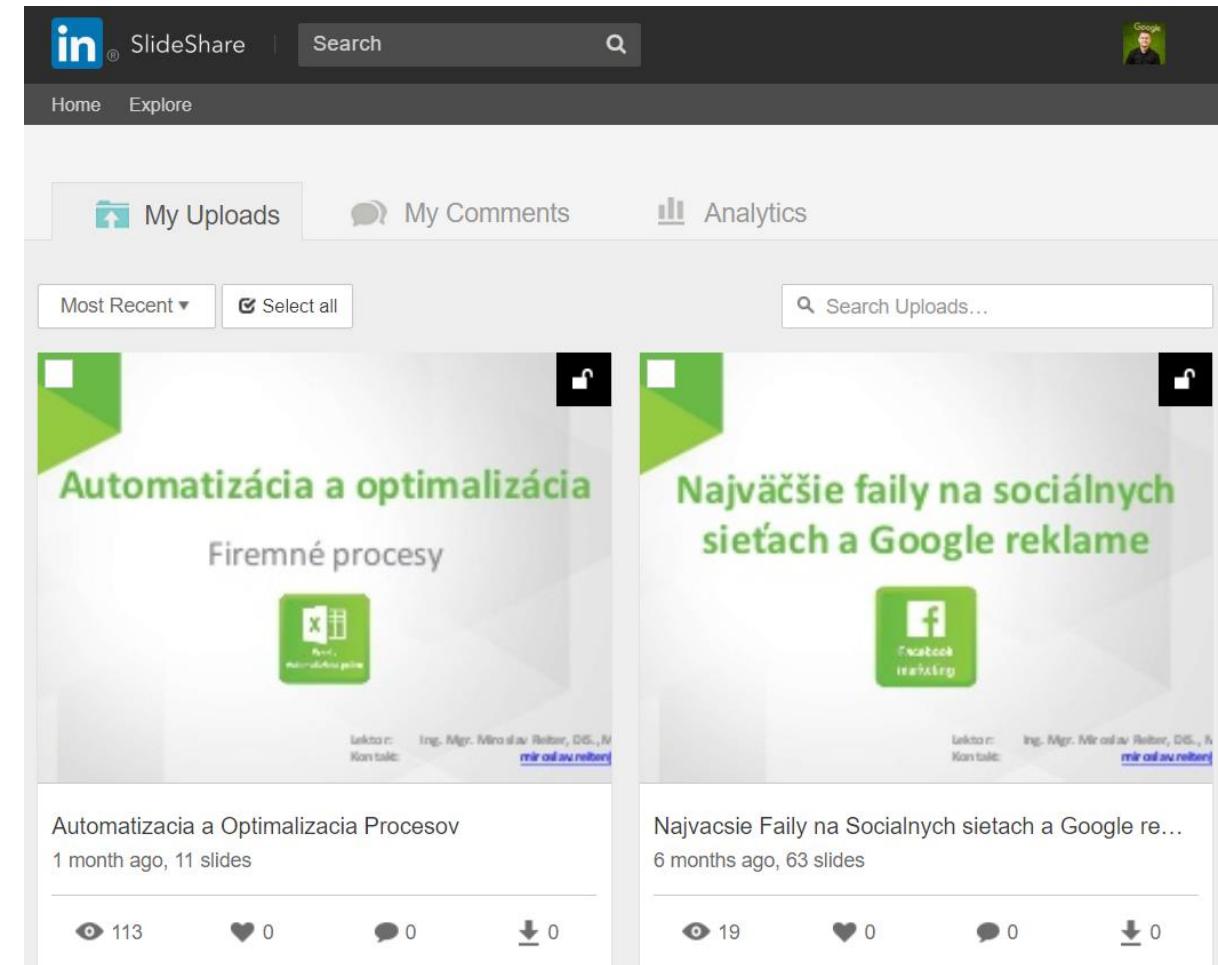
Súbor alebo Cloud

2. Prezentácia po prednáške

<https://github.com/miroslav-reiter/>

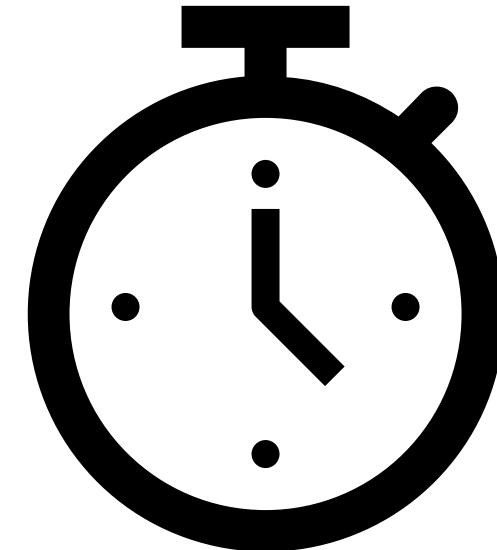
3. Videá na YouTube

[www.youtube.com/@IT-Academy](http://www.youtube.com/@IT-Academy)



# Úvodné Informácie

- Časový rozvrh (9:00-13:30)
  - Programujeme (50 min)
  - Prestávky (10 min)
  - Obedová prestávka
  - Mobilné telefóny a zariadenia
- 
- Priprav si otázky a rovno sa pýtaj
  - Interaktívna forma



# O mne - Miroslav Reiter

4

40000+  
klientov a  
1000+ firiem

IT Architekt  
Programátor  
Manažér

Microsoft  
Google  
ISTQB tréner

134  
certifikácií

151 príručiek  
a publikácií

13 škôl

62 projektov

Vlastná firma



---

**MOTIVÁCIA**

# Študuje 5 odborov a absolvoval už 12 univerzít. Ako zvláda stres a manažovanie času?



Foto: Jakub Kovalík pre FMK UCM | Miroslav Reiter na prednáške Grow with Google na FMK UCM.

**Nikola Kotláriková**

19. júl 2022 · 8 min. čítania





# Miroslav Reiter

1. PhDr. VŠM (Podnikovný manažment)
2. Ing. STU FEI (**Aplikovaná informatika**)
3. Mgr. UK FM (**Strategický manažment a marketing**)
4. Mgr. VŠM (**Manažérstvo kvality**)
5. Mgr. VŠEMVŠ (Verejná správa)
6. Mgr. DTI (**Učiteľstvo ekonomických predmetov**)
7. DiS. AMOS (Cestovný ruch)
8. **MBA LIGS (Executive management)**
9. **DBA Humanum (IT manažment)**
10. MPA IES (Verejná správa a samospráva krajov)
11. MSc. Humanum (**Bezpečnosť inf. systémov**)
12. Ing. Paed. IGIP STU
13. Mgr. PEVŠ (**Bezpečnosť informačných systémov**)
14. RNDR. PEVŠ (**Bezpečnosť informačných systémov**)



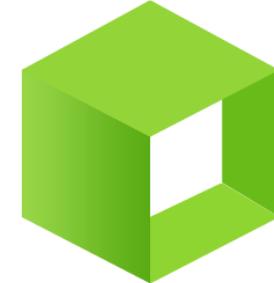
**FAKULTA MANAGEMENTU**  
Univerzita Komenského  
v Bratislave

**DIGITÁLNA  
UNIVERZITA**

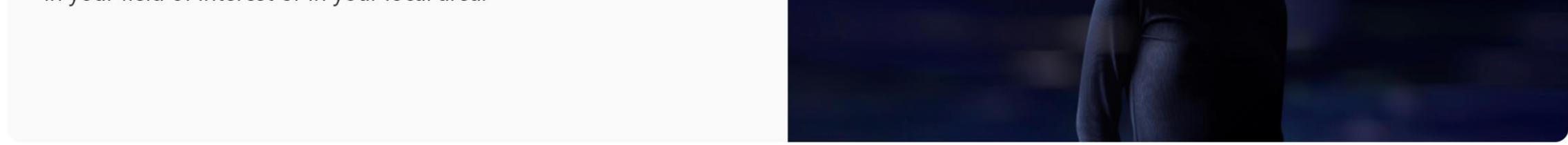
STU  
FIIT



 **VITA**  
ACADEMY



**IT ACADEMY**



## Filter

[People](#)    [Events](#)

[Search by filter](#)

- > Program
- > Country/Region
- > State/Province
- > Language
- > Award Category
- > Technology Area

# Search Profiles

[Search Profiles](#)

[Search](#)

[Most Valuable Professionals](#)

[Country/Region: Slovakia](#)

## Most Valuable Professionals



[Miroslav Reiter](#)

IT Architect and Programmer Hard worker Lecturer and Certified Trainer

Slovakia

## Most Valuable Professionals



[Peter Belko](#)

Content Developer, Technical Writer, Consultant, Trainer for M365 + Office apps

Slovakia

## Most Valuable Professionals

[About](#) [Events](#) [Find an MVP](#)[Profile](#) [Events](#)

### Headline

🌟 IT Architect and Programmer 📚 Hard worker 🎓  
Lecturer and Certified Trainer

### Biography

👉 I'm a hard worker, intellectual and joker. I love learning and teaching as well. My main objective is to teach people and improve their IT knowledge. To create truly practical knowledge necessary for life and present it in an interesting way. I don't like snake charmers and people who cannot...

[▼ Read more](#)

## Miroslav Reiter

 Slovakia IT Academy s.r.o.

### Most Valuable Professionals

## High Impact Activities

### Award Category

M365

### Technology Area

Visio, Excel

### Languages

English, Slovak

### Social



This community leader has not added a high impact activity yet.

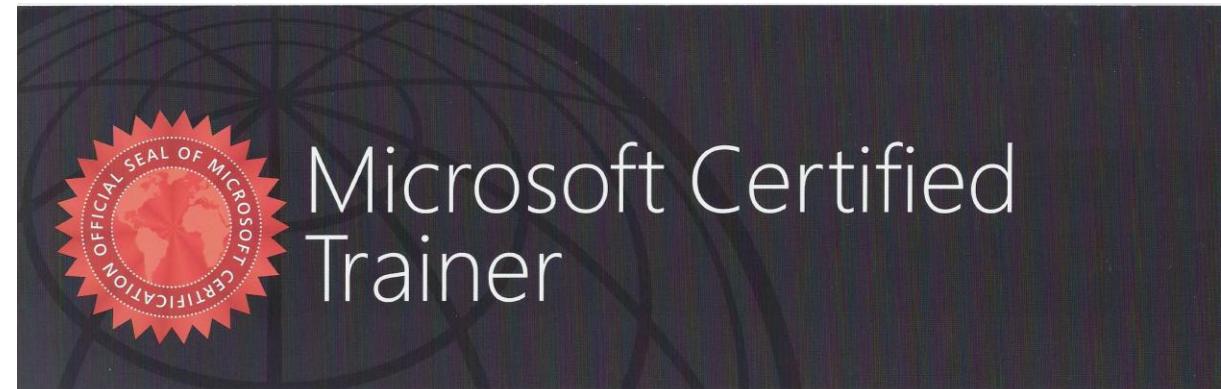
# Miroslav Reiter

získava status

## Google Certified Trainer

Automation

Google



Microsoft Certified  
Trainer

MIROSLAV REITER

Has successfully completed the requirements to be recognized as a Microsoft Certified Trainer

N. S.  
Satya Nadella  
Chief Executive Officer

Microsoft  
CERTIFIED  
Trainer

Znalec

# PhDr. Ing. Miroslav Reiter

Evidenčné číslo: 915864

## Miesto výkonu činnosti

Tomášikova 50  
83104 Bratislava  
Slovenská republika  
[Zobrazit na mape](#)

## Kontaktné údaje

Mobil: +421 908 163 084  
E-mail: [znapec@it-academy.sk](mailto:znapec@it-academy.sk)

## Odbory a odvetvia

Odbor / Odvetvie	História	Stav
<b>100000 - Elektrotechnika</b>		
100400 - Riadiaca technika, výpočtová technika (hardware)	14.02.2023 - <b>Zápis</b>	<b>AKTÍVNY</b>
100800 - Nosiče zvukových a zvukovoobrazových záznamov	14.02.2023 - <b>Zápis</b>	<b>AKTÍVNY</b>
100900 - Počítačové programy (software)	14.02.2023 - <b>Zápis</b>	<b>AKTÍVNY</b>
101000 - Bezpečnosť a ochrana informačných systémov	14.02.2023 - <b>Zápis</b>	<b>AKTÍVNY</b>

# Vyberte si online kurz

Naučte sa programovať, tvoriť webstránky a grafiku, manažovať alebo sa zamerajte na osobný rozvoj. Všetko jednoducho vďaka našim online kurzom z pohodlia domova.

## Ročné Predplatné na všetky Online Kurzy

2290 €

490 €

Pristup pre Vás do všetkých Aktuálnych aj Pripravovaných Online Kurzov

12 mesačná platnosť

Kúpiť teraz

Získať viac



541 kurzov v ponuke



Zábavné online lekcie



Akreditované kurzy



12 rokov skúseností



Certifikovaní profesionálni lektori

## Top kurzy Najobľúbenejšie kurzy

**BESTSELLER**

Online kurz SAP I.  
Začiatočník  
Miroslav Reiter  
 294 € 390 €

**BESTSELLER**

Online kurz Python a Jupyter I. Začiatočník  
Miroslav Reiter  
 152 € 214 €

**BESTSELLER**

Online kurz JavaScript I.  
Začiatočník  
Miroslav Reiter  
 168 € 226 €

**BESTSELLER**

Online kurz Java I.  
Začiatočník  
Miroslav Reiter  
 186 € 254 €

**BESTSELLER**

Online kurz SQL I.  
Začiatočník  
Miroslav Reiter  
 168 € 226 €

**BESTSELLER**

Online kurz Microsoft Excel III. Pokročilý  
Miroslav Reiter  
 196 € 223 €

[Všetky kurzy kategórie Najobľúbenejšie kurzy →](#)



Luigi, Mário  
a Yoshi

# Čo Robíte?

1. Študent/učiteľ'

2. Zamestnanec

3. Podnikateľ'

4. Nezamestnaný/materská

5. Dievča pre všetko



# Vaše Ciele a Očakávania

15

1. Doplniť si znalosti z jazyka Python

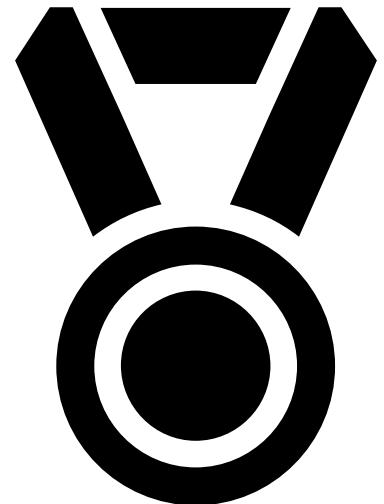
2. Základy strojového učenia

3. Základy knižnice scikit-learn

4. Dátové čistenie

5. Clustering

Zábava je v zaručená v každom bode :-)



# ČO JE STROJOVÉ UČENIE?

AKREDITOVANÝ KURZ



# Čo je Machine Learning?

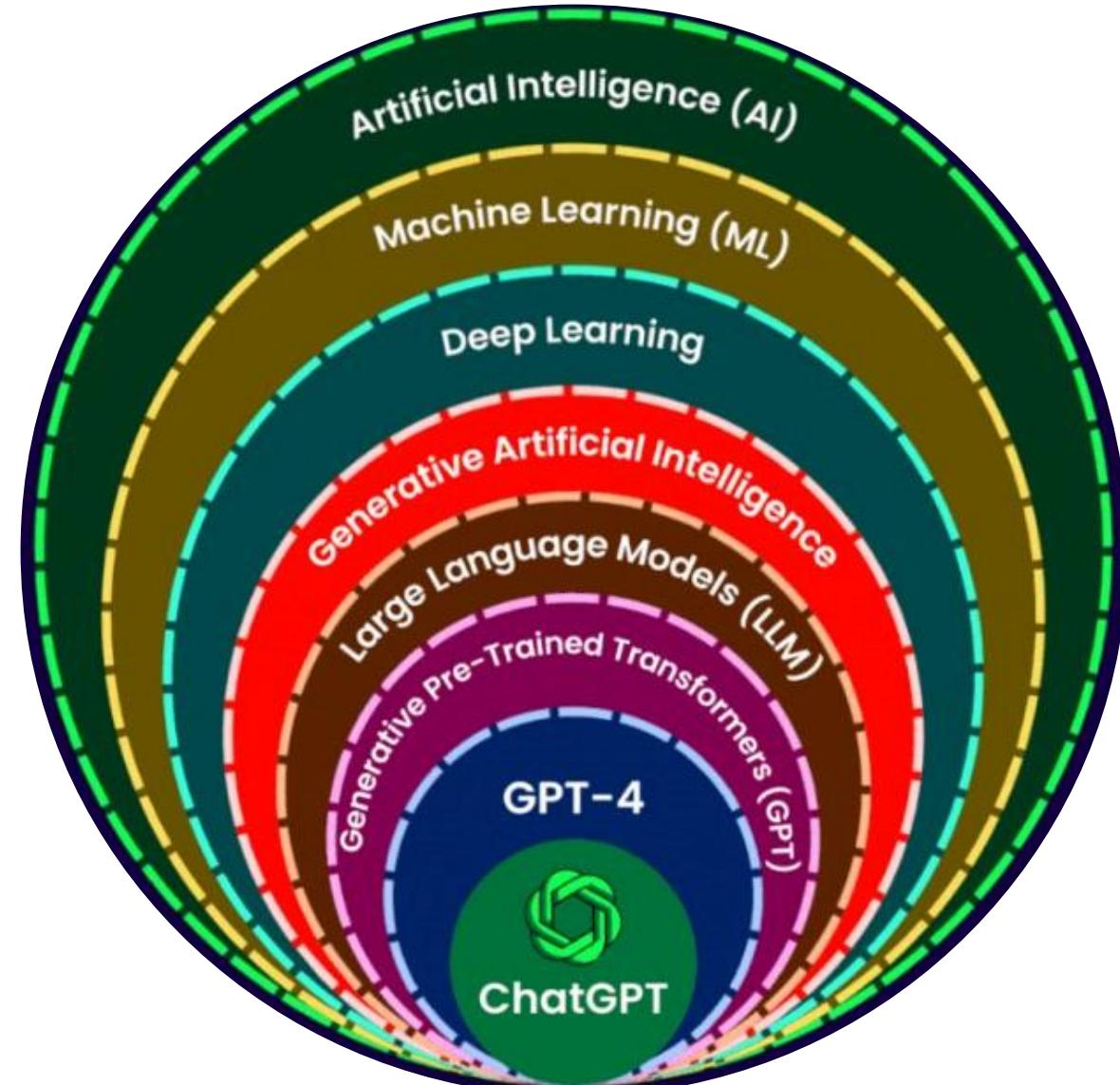
- Machine learning algorithms build **a mathematical model of sample data**, known as "training data", in order to make **predictions** or **decisions without** being **explicitly programmed to perform the task**.

-- Arthur L. Samuel, 1959



# Čo je Machine Learning?

- Oblast' umelej inteligencie, ktorá sa zaoberá vývojom algoritmov a modelov, ktoré umožňujú počítačom učiť sa z údajov a zlepšovať svoje výkony bez explicitného programovania
- Modely sa trénujú na historických dátach, aby dokázali predpovedať alebo klasifikovať nové, neznáme údaje



Kedy použiť strojové učenie	Kedy nepoužiť strojové učenie
Veľké množstvo dát dostupných na analýzu	Malé množstvo dát alebo nedostatok kvalitných údajov
Zložité vzory alebo vzťahy, ktoré je ťažké modelovať	Jednoduché problémy s jasnými pravidlami
Potreba predikcie na základe historických dát	Situácie s minimálnymi alebo žiadnymi historickými dátami
Adaptácia na nové vzory alebo zmeny v údajoch	Stabilné prostredie s nemennými pravidlami
Automatizácia rozhodovacích procesov	Situácie, kde je potrebná ľudská intuícia alebo hodnotenie

**big data**

Hľadaný výraz

**machine learning**

Hľadaný výraz

**data analysis**

Hľadaný výraz

+ Pridať porovnanie

SEO metrics for the Toolbar are available with one of the [paid plans](#)

SEO metrics for the Toolbar are available with one of the [paid plans](#)

SEO metrics for the Toolbar are available with one of the [paid plans](#)

Na celom svete ▾

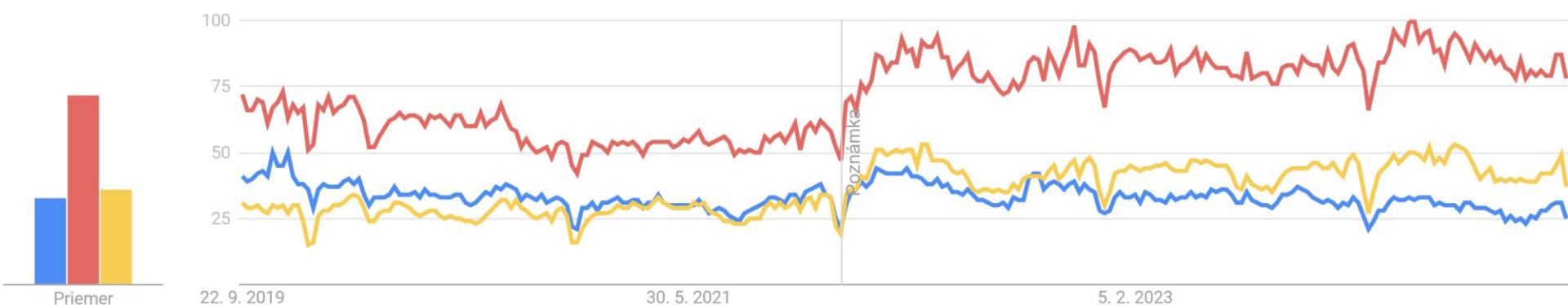
Posledných 5 rokov ▾

Všetky kategórie ▾

Webové vyhľadávanie ▾

Get Trend Data in Bulk

## Záujem v priebehu času



Priemer

22. 9. 2019

30. 5. 2021

5. 2. 2023

Export CSV

★ Add All Keywors



● big data ● machine learning ● data analysis

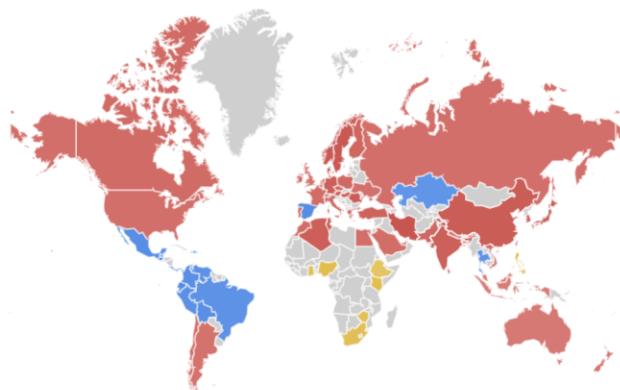
Na celom svete, Posledných 5 rokov

### Porovnanie podľa oblasti

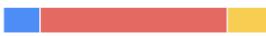
Oblast' ▾



● big data ● machine learning ● data analysis



Zoradiť: Záujem o výraz machine learning ▾

1	Irán	
2	Švajčiarsko	
3	Izrael	
4	Nórsko	
5	Švédsko	

Intenzita farby znázorňuje percento vyhľadávaní [ĎALŠIE INFORMÁCIE](#)

◀ Zobrazené oblasti 1 – 5 z 71 ▶

Zahrnúť oblasti s nízkym objemom vyhľadávania

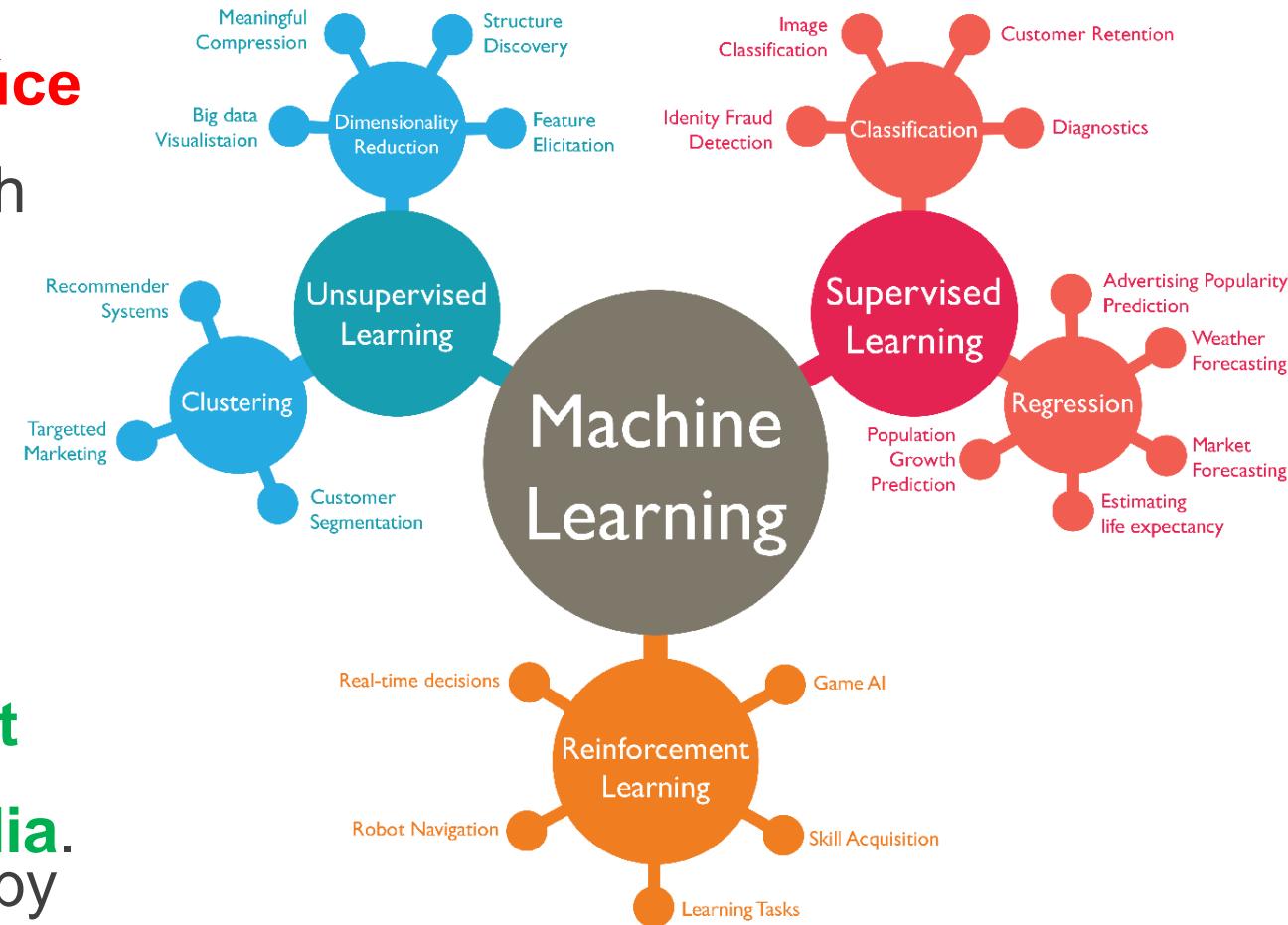
Export CSV

★ Add All Keywors



# Typy Strojového Učenia

- A. **Učenie s učiteľom (Supervised Learning)**: Model sa trénuje na dátach, ktoré **obsahujú vstupy a zodpovedajúce výstupy**. Cieľom je naučiť model predpovedať výstupy na základe nových vstupov. Príkladmi sú **regresia** a **klasifikácia**.
- B. **Učenie bez učiteľa (Unsupervised Learning)**: Model sa trénuje na dátach **bez explicitných výstupov**. Cieľom je objaviť vzory alebo štruktúry v údajoch, ako je **zhlukovanie alebo redukcia dimenzií**.
- C. **Učenie so zosilnením (Reinforcement Learning)**: Model sa učí na základe skúseností a **spätnej väzby z prostredia**. Cieľom je **optimalizovať sériu akcií**, aby sa dosiahol maximálny zisk.



# Učenie s učiteľom



- **Regresia:**  
**Predpovedanie cien** nehnuteľností  
na základe faktorov ako plocha,  
počet izieb a lokalita
- **Klasifikácia:**  
**Rozpoznávanie** e-mailov ako spam  
alebo nie spam na **základe obsahu**  
**a metadát**

NETFLIX

 Zillow

facebook

# Učenie bez učiteľa



- **Zhlukovanie:**  
**Segmentácia zákazníkov** na základe **nákupného správania** na **identifikáciu rôznych skupín**  
(napr. verní zákazníci, príležitostní kupujúci)
- **Redukcia dimenzií:**  
**Použitie PCA** (Principal Component Analysis) na **zjednodušenie** a **vizualizáciu** komplexných **dátových súborov**



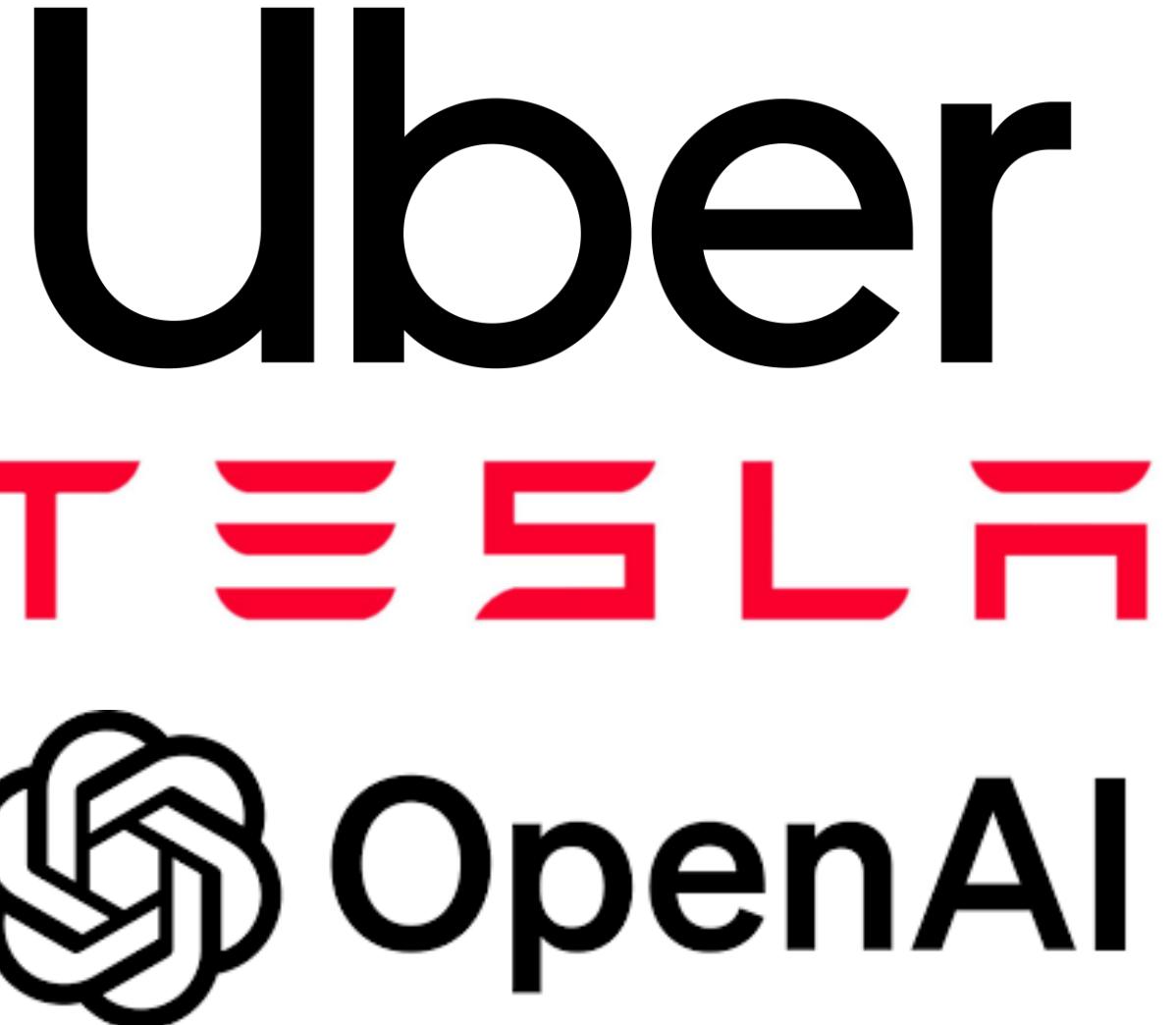
**Spotify**

**Google News**

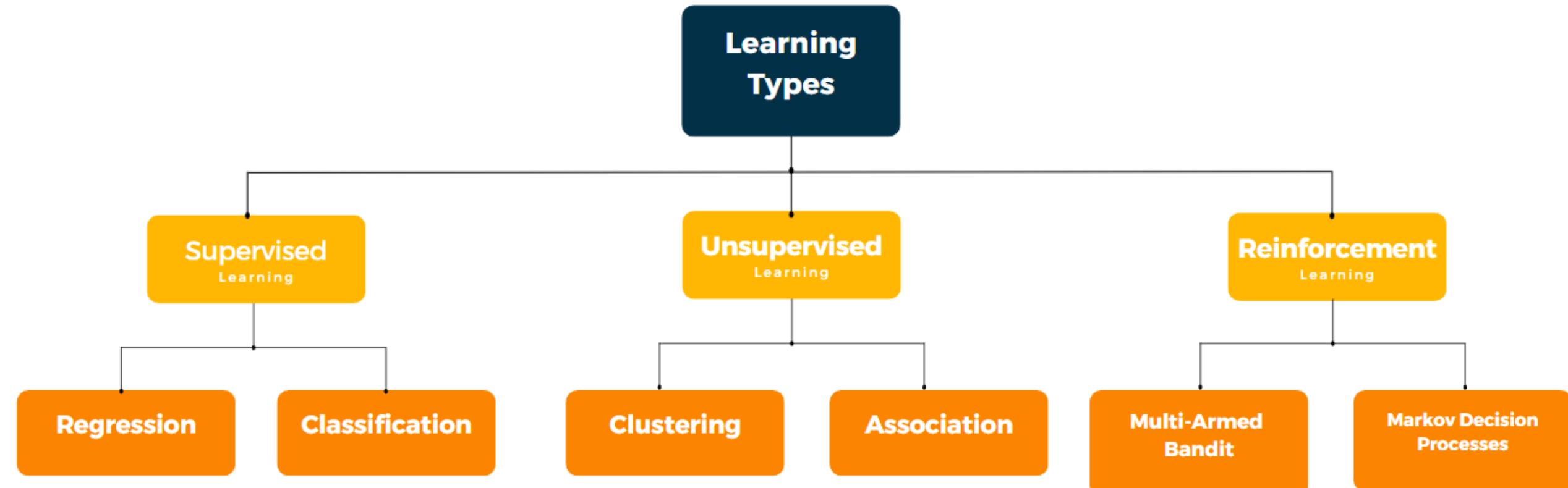
**LinkedIn**

# Učenie so zosilnením

- **Hry:**  
**Tréning agentov na hranie hier**, ako je Go alebo šach, kde sa učia strategicky hrať na základe výsledkov svojich tăahov
- **Roboti:**  
**Učenie robota**, ako sa **pohybovať** v prostredí, **optimalizovať cestu** a **vyhýbať sa prekážkam** na základe **skúseností a odmien**



# Typy Strojového Učenia



# Čo Treba na ML?



- 1. Dáta:** **Kvalitné** a **relevantné** dátá sú základom. Môžu byť štruktúrované (tabuľkové) alebo neštruktúrované (text, obrázky, zvuk).
- 2. Algoritmy:** Rôzne algoritmy strojového učenia (napr. **regresia**, **klasifikácia**, **zhľukovanie**) na analýzu a modelovanie dát.
- 3. Výpočtová kapacita:** Dostatočné hardvérové prostriedky (**CPU**, **GPU**) na spracovanie a trénovanie modelov, najmä pri veľkých dátach.
- 4. Programovací jazyk a knižnice:** Jazyk ako **Python**, **R** alebo **Java** a knižnice ako **TensorFlow**, **scikit-learn**, **Keras**, alebo **PyTorch** na implementáciu algoritmov.
- 5. Odborné znalosti:** Znalosti z oblasti **štatistiky**, **matematiky** a **doménové znalosti** sú potrebné na pochopenie problémov a interpretáciu výsledkov.
- 6. Nástroje na vizualizáciu:** Nástroje ako **Matplotlib**, **Seaborn** alebo **Tableau** na analýzu a prezentáciu výsledkov.
- 7. Infrastruktúra na ukladanie dát:** Systémy ako **databázy** alebo **cloudové služby** (napr. **AWS**, **Google Cloud**) na ukladanie a sprístupnenie dát.
- 8. Prototypovanie a testovanie:** Postupy na vytváranie a testovanie prototypov modelov na validáciu ich výkonnosti.

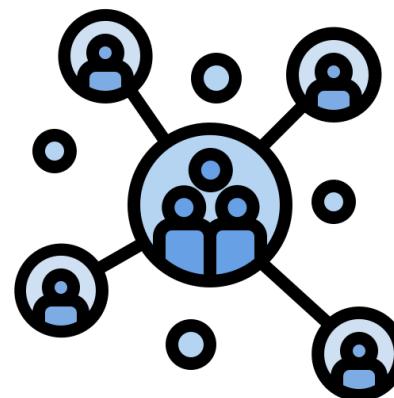


python



kaggle

matplotlib



Čo Treba  
na ML?



 Datalore for Teams

Your team's private platform for collaborative Python analytics and interactive BI apps

Create your Datalore Cloud account

 Email Password [Try 14 days for free](#)

Already have an account? [Log in](#)

Or sign up with:

[Google](#)[JetBrains Account](#)

#### On-Premises

Contact product expert for platform demo and pricing

[Schedule demo](#)



The easiest way to run Python, Spyder with SciPy and friends out of the box on any Windows PC, without installing anything!

Project Home is on [Github](#), downloads pages are on [Sourceforge](#) and [Github](#), [md5-sha](#) , [Discussion Group](#)

## Recent Releases

Release [2024-04](#) of September 14th, 2024

Highlights (\*): Python-3.11.9, Python-3.12.6, Python-3.13.0rc2, Numpy-2.0.1, Scipy-1.14.1, Torch-2.4.1, a slim version

WinPython **3.11** Downloads (\*\*) via [SourceForge](#) and [Github](#)

- WinPython64-**3.11**.9.0dot = Python 3.11.9 64bit only : [Changelog](#), [Packages](#)

WinPython **3.12** Downloads (\*\*) via [SourceForge](#) and [Github](#)

- WinPython64-**3.12**.6.0dot = Python 3.12.6 64bit only : [Changelog](#), [Packages](#)

- WinPython64-**3.12**.6.0slim = Python 3.12.6 64bit with PyQt5 + Spyder - Torch: [Changelog](#), [Packages](#)

- WinPython64-**3.12**.6.0 = Python 3.12.6 64bit with PyQt5 + Spyder : [Changelog](#), [Packages](#)

WinPython **3.13** Downloads (\*\*) via [SourceForge](#) and [Github](#)

- WinPython64-**3.13**.0.1rc2 = Python 3.13.1rc2 64bit only : [Changelog](#), [Packages](#)

- WinPython64-**3.13**.0.1rc2 = Python 3.13.1rc2 64bit only : [Changelog](#), [Packages](#)

Release [2024-03](#) of June 22nd, 2024

Highlights (\*): Python-3.12.4, Python-3.13.0b2 free-threading, Jupyterlab-4.2.2, Scipy-1.13.1, scikit\_learn-1.5.0, Duckdb-1.0.0

WinPython **3.12** Downloads (\*\*) via [SourceForge](#) and [Github](#)

- WinPython64-**3.12**.4.1dot = Python 3.12.4 64bit only : [Changelog](#), [Packages](#)

- WinPython64-**3.12**.4.1 = Python 3.12.4 64bit with PyQt5 + Spyder : [Changelog](#), [Packages](#)

WinPython **3.13** Downloads (\*\*) via [SourceForge](#) and [Github](#)

- WinPython64-**3.13**.0.0dotb2b = Python 3.13.0b2 64bit only : [Changelog](#), [Packages](#)

- WinPython64-**3.13**.0.0freeb2b = Python 3.13.0b2 64bit only : [Changelog](#), [Packages](#)

Release [2024-02](#) of April 19th, 2024

Highlights (\*): Python-3.12.3, Jupyterlab-4.1.5, Spyder-5.5.4, Numpy-1.26.4, Duckdb-0.10.1, Keras-3.1.1

WinPython **3.12** Downloads (\*\*) via [SourceForge](#) and [Github](#)

- WinPython64-**3.12**.3.0dot = Python 3.12.3 64bit only : [Changelog](#), [Packages](#)

- WinPython64-**3.12**.3.0 = Python 3.12.3 64bit with PyQt5 + Spyder : [Changelog](#), [Packages](#)

- WinPython64-**3.12**.3.0dot = Python 3.12.3 32bit only : [Changelog](#), [Packages](#)

Release [2024-01](#) of February 16th, 2024

Highlights (\*): Python-3.11.8, Python-3.12.2, ortools-9.8.3296, Pandas-2.2.0, Scipy-1.12.0, Torch-2.2.0

WinPython **3.11** Downloads (\*\*) via [SourceForge](#) and [Github](#)





# Tinker With a Neural Network Right Here in Your Browser.

## Don't Worry, You Can't Break It. We Promise.

Epoch  
000,000Learning rate  
0.03Activation  
TanhRegularization  
NoneRegularization rate  
0Problem type  
Classification

## DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

## FEATURES

Which properties do you want to feed in?



## 2 HIDDEN LAYERS



4 neurons



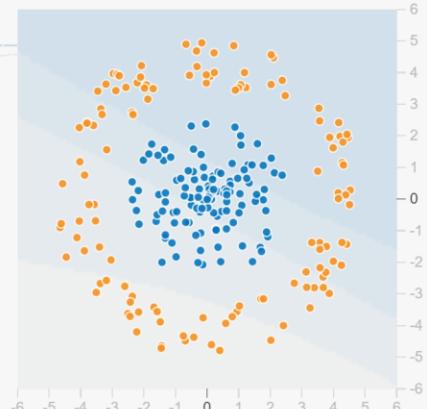
2 neurons



This is the output from one neuron.  
Hover to see it larger.

The outputs are mixed with varying weights, shown by the thickness of the lines.

## OUTPUT

Test loss 0.513  
Training loss 0.507

Colors shows data, neuron and weight values.

Show test data Discretize output



LINEAR REGRESSION SIMULATOR

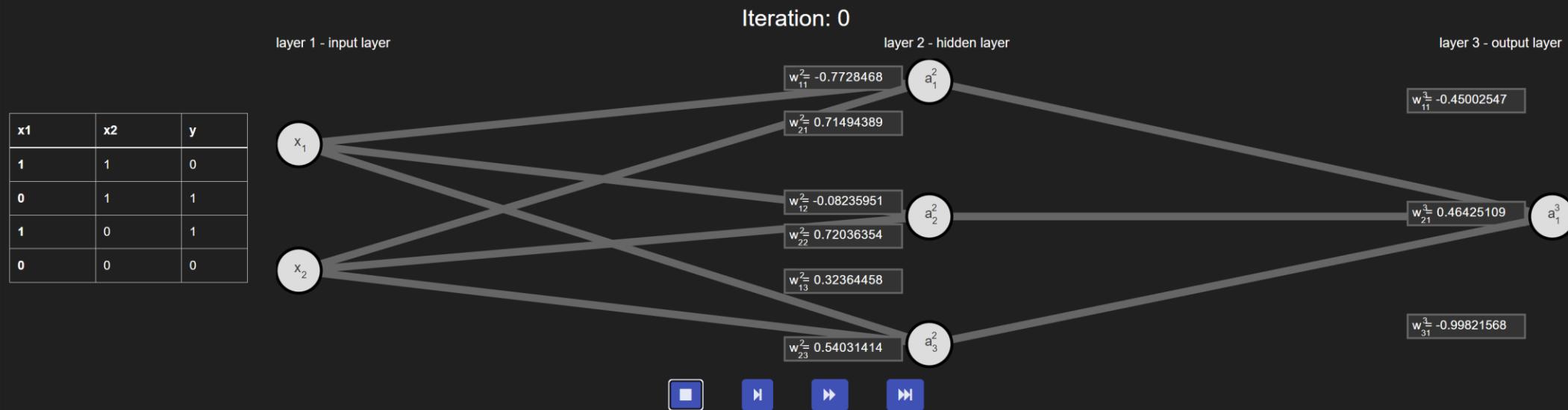
NEURAL NETWORK SIMULATOR

ELMAN RECURRENT NETWORK

Q-LEARNING SIMULATOR

**NEURAL NETWORK SIMULATOR**

Neural Network Simulator is a real feedforward neural network running in your browser. The simulator will help you understand how artificial neural network works. The network is trained using backpropagation algorithm, and the goal of the training is to learn the XOR function. One forward and the backward pass of single training example is called iteration, each iteration consists of 10 steps.

**Step 0) Initialization**

The weights are randomly initialized to the range (-1, 1).

**Forward pass****Step 1) Input layer**

$$x_1 =$$

$$x_2 =$$

**Step 2) Hidden layer**

$$a_1^{(2)} = \sigma(w_{11}^{(2)}x_1 + w_{21}^{(2)}x_2) =$$

$$a_2^{(2)} = \sigma(w_{12}^{(2)}x_1 + w_{22}^{(2)}x_2) =$$

$$a_3^{(2)} = \sigma(w_{13}^{(2)}x_1 + w_{23}^{(2)}x_2) =$$

**Step 3) Output layer**

$$a_1^{(3)} = \sigma(w_{11}^{(3)}a_1^{(2)} + w_{21}^{(3)}a_2^{(2)} + w_{31}^{(3)}a_3^{(2)}) =$$

**Step 4) Calculate the cost**

$$E = \frac{1}{2}(y - a_1^{(3)})^2 =$$





Search



Datasets



Tasks

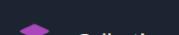


Flows



Runs

10.1M



Collections



Benchmarks

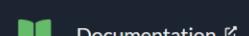


Task Types

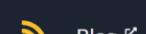


Measures

Learn



Documentation ↗



Blog ↗



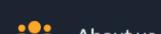
API's



Contribute



Meet up



About us



Terms &amp; Citation

10103733 runs found

**❗ torch.1c39b0d3ec8abb7(1) on Meta\_Album\_PNU\_Micro**

torch.nn.Net.1c39b0d3ec8abb7(1) on Meta\_Album\_PNU\_Micro by Taniya D

10595146 3 weeks ago

**❗ torch.5a03bdc898bc4471(1) on Meta\_Album\_PNU\_Micro**

torch.nn.Net.5a03bdc898bc4471(1) on Meta\_Album\_PNU\_Micro by Taniya D

10595145 3 weeks ago

**❗ torch.2ffa3de8228eefa8(1) on Meta\_Album\_PNU\_Micro**

torch.nn.Net.2ffa3de8228eefa8(1) on Meta\_Album\_PNU\_Micro by Taniya D

10595144 3 weeks ago

**❗ sklearn.DecisionTreeClass... on mux6**

sklearn.pipeline.Pipeline(preprocessor=p... on mux6 by Karim Belaid

10595143 ACC 1 AUC 1 RMSE 0.0605 3 weeks ago

**❗ sklearn.RandomForestClass... on diabetes**

sklearn.ensemble.\_forest.RandomForestCla... on diabetes by Hari Mallik

10595142 ACC 0.7799 AUC 0.8331 RMSE 0.3955 3 weeks ago

**❗ sklearn.RandomForestClass... on diabetes**

sklearn.ensemble.\_forest.RandomForestCla... on diabetes by Kamal Sapkota

10595141 ACC 0.7734 AUC 0.8278 RMSE 0.3988 3 weeks ago

**❗ sklearn.RandomForestClass... on diabetes**

sklearn.ensemble.\_forest.RandomForestCla... on diabetes by Madhav Prasad Ghimire

10595140 ACC 0.7656 AUC 0.8216 RMSE 0.4016 3 weeks ago

**❗ sklearn.RandomForestClass... on diabetes**

sklearn.ensemble.\_forest.RandomForestCla... on diabetes by Niraj Dhital



# Top ML Knižnice

scikit-learn

TensorFlow

Keras

PyTorch

Fastai

XGBoost

LightGBM

CatBoost

H2O.ai

MLflow



# scikit-learn

## Machine Learning in Python

Getting Started

Release Highlights for 1.5

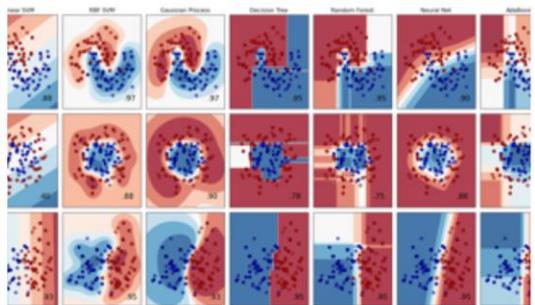
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

### Classification

Identifying which category an object belongs to.

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

**Algorithms:** [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [logistic regression](#), and [more...](#)



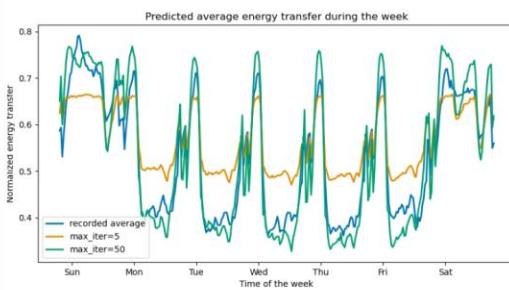
Examples

### 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...](#)



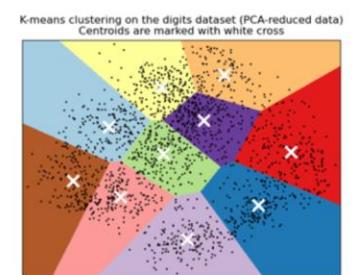
Examples

### Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, grouping experiment outcomes.

**Algorithms:** [k-Means](#), [HDBSCAN](#), [hierarchical clustering](#), and [more...](#)



Examples

### Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, increased efficiency.

**Algorithms:** [PCA](#), [feature selection](#), [non-negative matrix factorization](#), and [more...](#)

### Model selection

Comparing, validating and choosing parameters and models.

**Applications:** Improved accuracy via parameter tuning.

**Algorithms:** [Grid search](#), [cross validation](#), [metrics](#), and [more...](#)

### Preprocessing

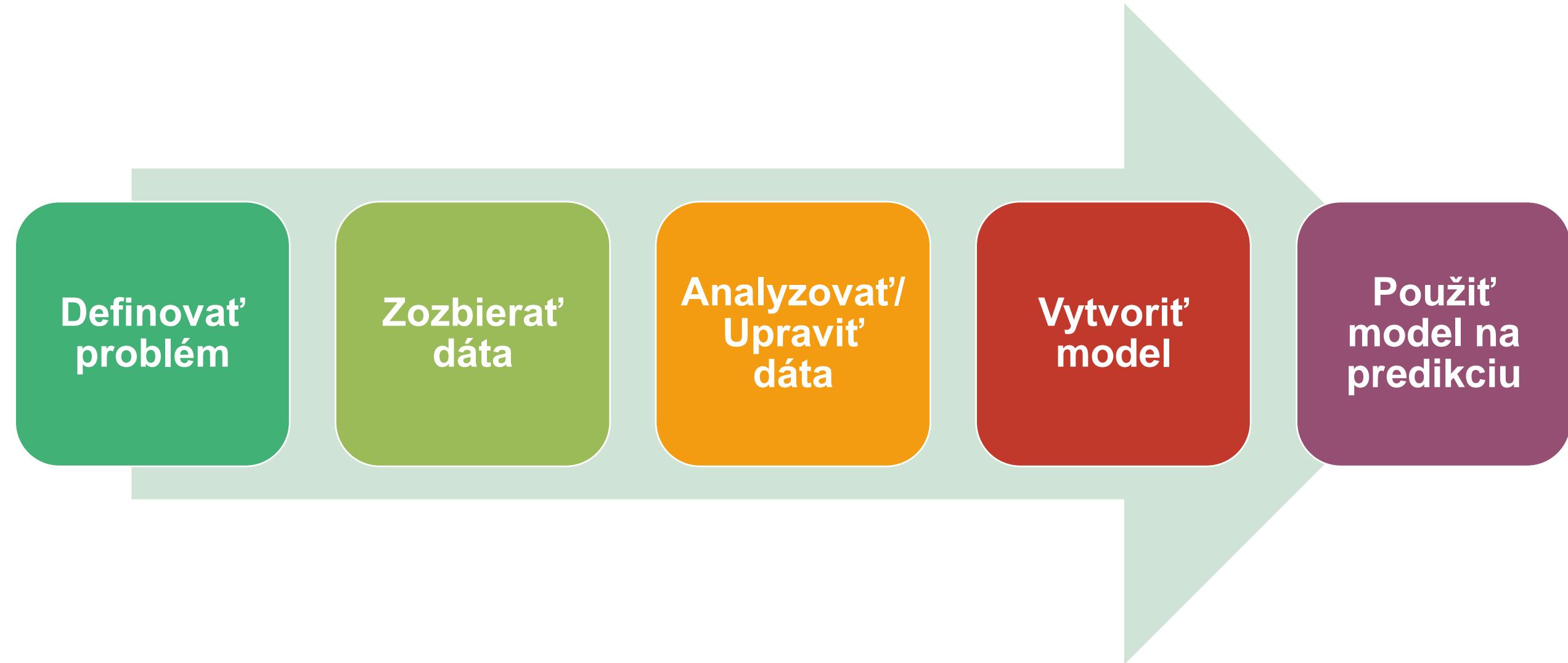
Feature extraction and normalization.

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

**Algorithms:** [Preprocessing](#), [feature extraction](#), and [more...](#)

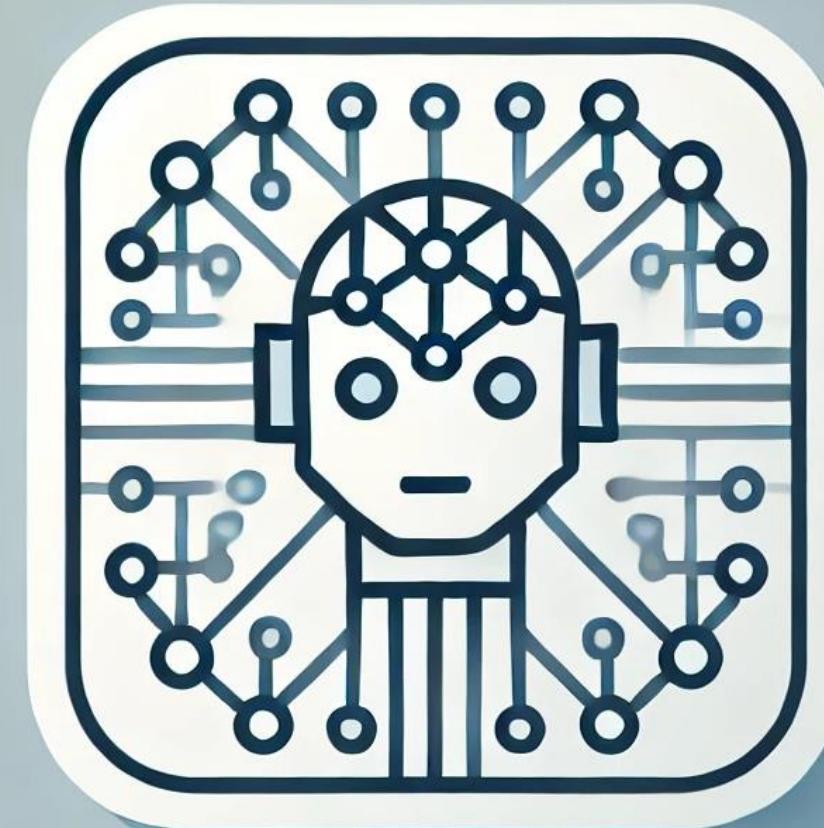


# Ako na ML Riešenie?



# Úlohy Strojové Učenie

1. Pochopenie čo je strojové učenie
2. Prehľad medzi typmi ML
3. Prehľad praktických aplikácií
4. Prehľad top ML knižníc
5. Prehľad potrebných vecí k ML



# ČO JE STROJOVÉ UČENIE?

AKREDITOVANÝ KURZ



# Knižnica Scikit-learn



# Čo je Scikit-learn?



- Python knižnica
- Používa sa na **strojové učenie**
- Obsahuje nástroje na **klasifikáciu, regresiu a klastrovanie**
- Poskytuje **metódy na predspracovanie údajov**
- Je známa **jednoduchým a flexibilným používaním**
- Obsahuje **implementácie algoritmov** ako rozhodovacie stromy, SVM a náhodné lesy
- Podporuje **vyhodnocovanie modelov**



# Použitie Skikit-learn

**1. Klasifikácia  
(Classification)**

**2. Regresia  
(Regression)**

**3. Klastrovanie  
(Clustering)**

**4. Redukcia  
dimenzií  
(Dimensionality  
reduction)**

**5. Výber modelu  
(Model  
selection)**

**6.  
Predspracovanie  
(Preprocessing)**

# Klasifikácia (diskrétny výstup)



---

**Klasifikácia e-mailov:** Rozdelenie e-mailov na spam a ne-spam

---

**Diagnóza chorôb:** Určenie, či pacient má určité ochorenie na základe symptómov a testov

---

**Detekcia podvodov:** Identifikácia podvodných transakcií na základe vzorcov správania

---

**Rozpoznávanie obrázkov:** Klasifikácia obrázkov do kategórií, ako sú zvieratá, rastliny, alebo objekty

---

**Analýza sentimentu:** Určenie, či je text pozitívny, negatívny alebo neutrálny

---

**Klasifikácia zákazníkov:** Rozdelenie zákazníkov do segmentov na základe ich nákupného správania

# Regresia (spojitý výstup)



1. **Predpovedanie cien nehnuteľností**: Určenie ceny domu na základe jeho charakteristík (rozloha, počet izieb, lokalita).
2. **Predikcia predaja**: Odhadovanie predaja výrobkov na základe historických údajov a marketingových kampaní.
3. **Predpovedanie teplôt**: Odhadovanie teploty vzduchu na základe meteorologických údajov a historických trendov.
4. **Analýza výkonnosti**: Predpovedanie výkonu zamestnancov na základe rôznych faktorov, ako sú skúsenosti a vzdelanie.
5. **Finančné analýzy**: Odhadovanie budúcich ziskov spoločnosti na základe minulých výsledkov a trendov v odvetví.
6. **Predikcia výdavkov**: Odhadovanie budúcich výdavkov domácnosti na základe predchádzajúcich nákupných zvyklostí a trendov.

# Klastrovanie (Učenie bez učiteľa)



1. **Segmentácia zákazníkov:** Rozdelenie zákazníkov do skupín na základe podobných nákupných zvyklostí a preferencií.
2. **Organizácia dokumentov:** Automatické zoskupovanie podobných dokumentov alebo textov do tém alebo kategórií.
3. **Analýza trhu:** Identifikácia rôznych segmentov trhu na základe demografických a behaviorálnych údajov.
4. **Detekcia anomálií:** Identifikácia nezvyčajných vzorov alebo udalostí v dátach, ako sú podvodné transakcie.
5. **Klasifikácia obrázkov:** Skupinovanie obrázkov s podobnými vlastnosťami, ako sú farby, tvary alebo objekty.
6. **Zoskupovanie biologických vzorov:** Identifikácia podobných genómov alebo proteínov v biologických dátach.

# scikit-learn

## Machine Learning in Python

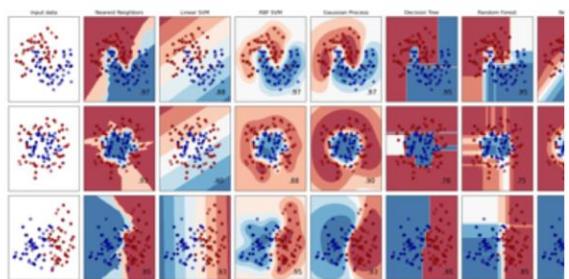
[Getting Started](#)[Release Highlights for 1.4](#)[GitHub](#)

- 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...](#)

[Examples](#)

### Dimensionality reduction

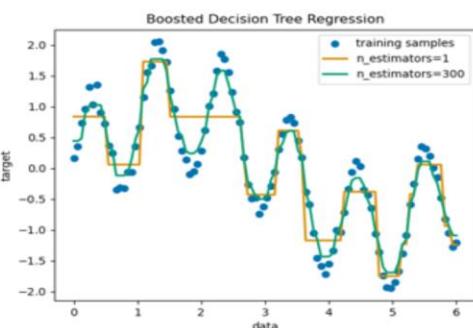
Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency  
**Algorithms:** PCA, feature selection, non-negative matrix factorization, and [more...](#)

### Regression

Predicting a continuous-valued attribute associated with an object.

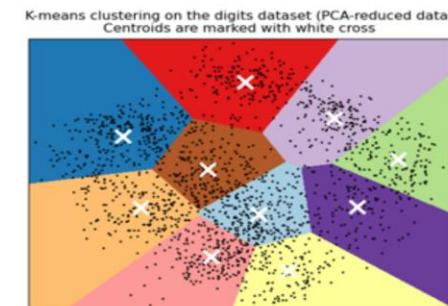
**Applications:** Drug response, Stock prices.  
**Algorithms:** Gradient boosting, nearest neighbors, random forest, ridge, and [more...](#)

[Examples](#)

### Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes  
**Algorithms:** k-Means, HDBSCAN, hierarchical clustering, and [more...](#)

[Examples](#)

### Model selection

Comparing, validating and choosing parameters and models.

**Applications:** Improved accuracy via parameter tuning  
**Algorithms:** grid search, cross validation, metrics,

### Preprocessing

Feature extraction and normalization.

**Applications:** Transforming input data such as text for use with machine learning algorithms.  
**Algorithms:** preprocessing, feature extraction, and [more...](#)



Release Highlights

Biclustering

Calibration

Classification

Clustering

Covariance estimation

Cross decomposition

Dataset examples

Decision Trees

Decomposition

Developing Estimators

Ensemble methods

**Examples based on real world datasets**

Compressive sensing: tomography reconstruction with L1 prior (Lasso)

Faces recognition example using eigenfaces and SVMs

Image denoising using kernel PCA

Lagged features for time series forecasting

Model Complexity Influence

Out-of-core classification of text documents

Outlier detection on a real data set

Prediction Latency

Species distribution modeling

Time-related feature engineering

Topic extraction with Non-negative Matrix Factorization and Latent Dirichlet Allocation

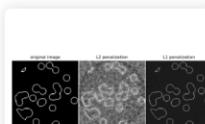
Visualizing the stock market structure

Wikipedia principal eigenvector

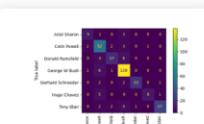
Feature Selection

# Examples based on real world datasets

Applications to real world problems with some medium sized datasets or interactive user interface.



Compressive sensing:  
tomography  
reconstruction with  
L1 prior (Lasso)



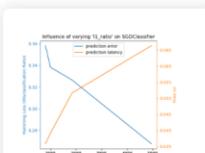
Faces recognition  
example using  
eigenfaces and  
SVMs



Image denoising  
using kernel PCA



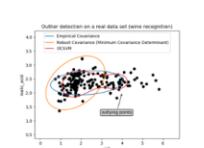
Lagged features for  
time series  
forecasting



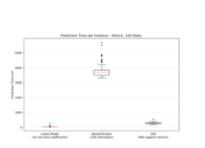
Model Complexity  
Influence



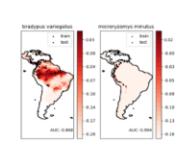
Out-of-core  
classification of text  
documents



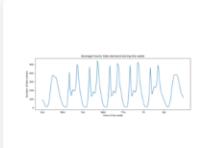
Outlier detection on  
a real data set



Prediction Latency



Species distribution  
modeling



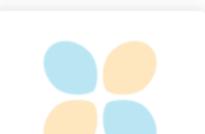
Time-related feature  
engineering



Topic extraction with  
Non-negative  
Matrix Factorization  
and Latent Dirichlet  
Allocation



Visualizing the stock  
market structure



Wikipedia principal  
eigenvector



# Evaluácia F1 Skóre

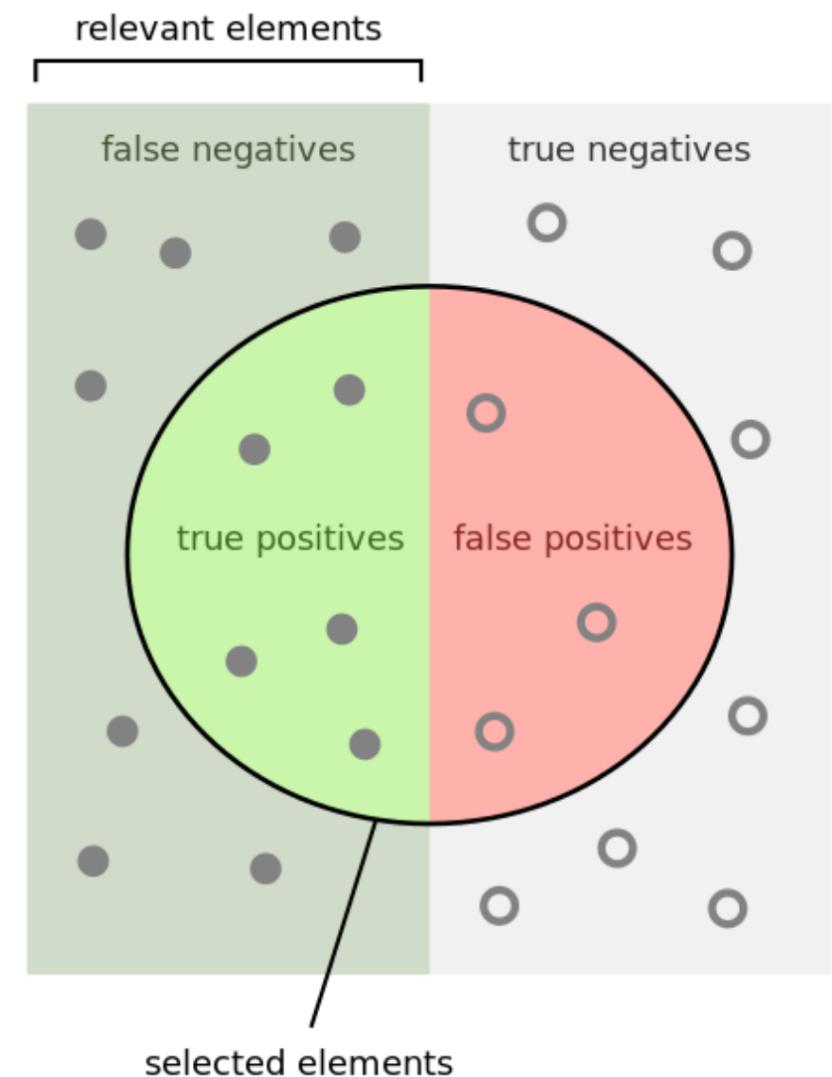
$$F_1 \text{ skóre} = \frac{2 * P * R}{P + R}$$

How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$



[For You](#)[Explore](#)[Following](#)[Friends](#)[LIVE](#)[Messages](#)[Profile](#)

## Following accounts

Accounts you follow will appear here

Create TikTok effects,  
get a reward

Company

Program

Terms &amp; Policies

© 2024 TikTok

**\_gcanale**  
Giuseppe Canale[Follow](#)

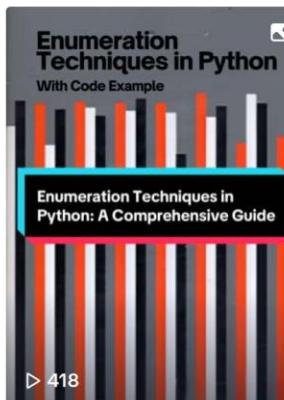
...

10 Following 8481 Followers 89.9K Likes

cybersecurity | AI | ML | CISSP | coding | database

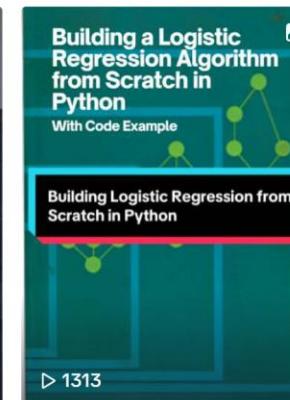
[gabeat](#)[Videos](#)[Reposts](#)[Liked](#)

Latest Popular Oldest

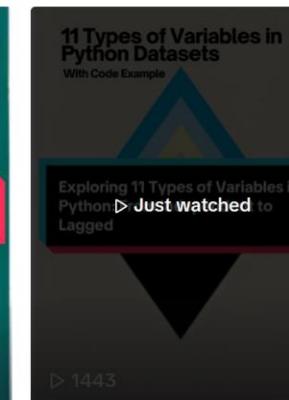


Enumeration Techniques in Python: A Comprehensive Guide

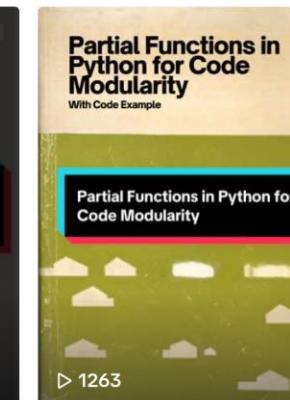
418

Building a Logistic Regression Algorithm from Scratch in Python  
With Code Example

1313

11 Types of Variables in Python Datasets  
With Code Example

1443

Partial Functions in Python for Code Modularity  
With Code Example

1263

Introduction to Random Forest in Python  
With Code Example

1103

FastAPI with Python  
With Code Example

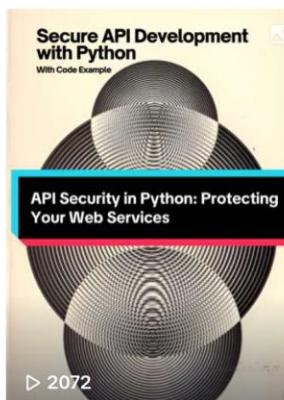
FastAPI in Python: Building High-Performance Web APIs

4713

Mastering QLoRA  
With Code Example

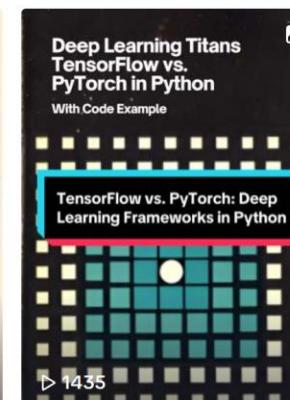
QLoRA: Efficient Fine-tuning of Large Language Models in Python

894

Secure API Development with Python  
With Code Example

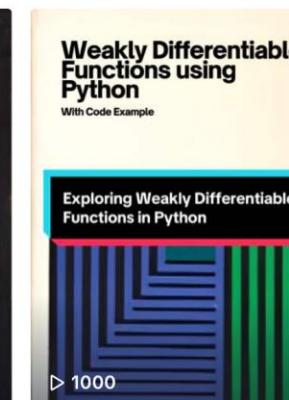
API Security in Python: Protecting Your Web Services

2072

Deep Learning Titans TensorFlow vs. PyTorch in Python  
With Code Example

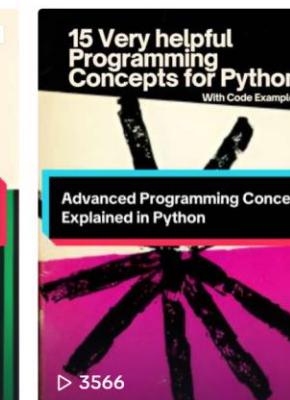
TensorFlow vs. PyTorch: Deep Learning Frameworks in Python

1435

Weakly Differentiable Functions using Python  
With Code Example

Exploring Weakly Differentiable Functions in Python

1000

15 Very helpful Programming Concepts for Python  
With Code Example

Advanced Programming Concepts Explained in Python

3566

Time Series Anomaly Detection with Spatial-Temporal Normality Learning  
With Code Example

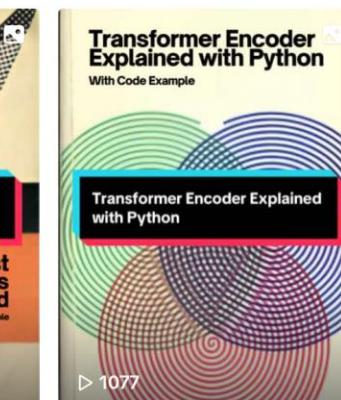
Time Series Anomaly Detection with Spatial-Temporal Normality Learning

1197

Python List Methods: Essential Operations Explained  
With Code Example

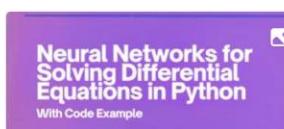
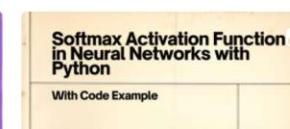
Python List Methods Illustrated

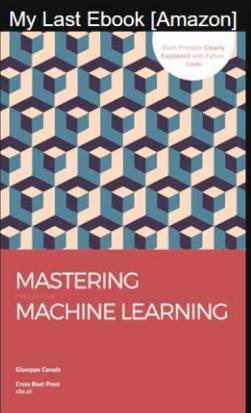
1967

Transformer Encoder Explained with Python  
With Code Example

Transformer Encoder Explained with Python

1077

Neural Networks for Solving Differential Equations in Python  
With Code ExampleSoftmax Activation Function in Neural Networks with Python  
With Code ExampleOptimizers for Deep Learning Momentum, Nesterov, Adagrad, RMSProp, Adam  
With Code ExampleVisualizing Embeddings and Outliers in Fine-Tuning with Python  
With Code ExampleUnleashing Word2Vec for NLP in Python  
With Code Example

[Support using PayPal](#)[Subscription using PayPal](#)[Sponsor Me! \[Github\]](#)

<a href="#">11 Types of Variables in Python Datasets</a> ↓	<a href="#">15 Most Commonly Used ML Algorithms in Python</a> ↓	<a href="#">15 Very helpful Programming Concepts for Python</a> ↓	<a href="#">2-Stage Backpropagation in Python</a> ↓	<a href="#">A Simple Neural Network Module for Relational Reasoning</a> ↓	<a href="#">AI vs Machine Learning Exploring the Differences</a> ↓	<a href="#">AI-Powered Academic Research Assistance with Python and ArXiv</a> ↓
<a href="#">Accelerated Generation Techniques in Large Language Models</a> ↓	<a href="#">Accelerating Python with Numba</a> ↓	<a href="#">Active Learning in Machine Learning with Python</a> ↓	<a href="#">Active Learning vs Cooperative Learning in Python</a> ↓	<a href="#">Active Shape Model for Face Image Generation</a> ↓	<a href="#">AdaBoost Model in Python</a> ↓	<a href="#">Adam Optimizer in Python</a> ↓
<a href="#">Advanced Clustering Techniques in Machine Learning in Python</a> ↓	<a href="#">Advanced Generative Adversarial Networks with Python</a> ↓	<a href="#">Advanced Handling Missing Data in Python</a> ↓	<a href="#">Advanced Integral Calculus in Machine Learning with Python</a> ↓	<a href="#">Advanced Machine Learning Models in Python</a> ↓	<a href="#">Advanced Pandas Techniques for Machine Learning</a> ↓	<a href="#">Advanced SQL Techniques CTEs, Subqueries, and More</a> ↓
<a href="#">Advanced Techniques for Many-to-One Relationships in Pandas</a> ↓	<a href="#">Advantages of Random Forest Algorithm in Python</a> ↓	<a href="#">Advantages of Weighted Averaging in Ensemble Models</a> ↓	<a href="#">Adversarial Attacks on Machine Learning Models in Python</a> ↓	<a href="#">Aggregation of Reasoning in Python</a> ↓	<a href="#">Algebraic Functions and Projective Curves in Python</a> ↓	<a href="#">Algebraic Topology Using Python</a> ↓
<a href="#">Analyzing User Engagement and Churn with Python</a> ↓	<a href="#">Analyzing the \$21 Million Lifetime Airline Ticket</a> ↓	<a href="#">Anomaly Detection in Graph-Based Data Using Latent Space Diffusion Models</a> ↓	<a href="#">Apache Spark Concepts and Python Examples</a> ↓	<a href="#">Applying Geometric Deep Learning with Python</a> ↓	<a href="#">Architectural Design Patterns in Python</a> ↓	<a href="#">Aspect-Based Sentiment Analysis with Python</a> ↓



# Úlohy sklearn

1. Pochopenie čo je scikit-learn
2. Prehľad použitia scikit-learn
3. Praktické príklady

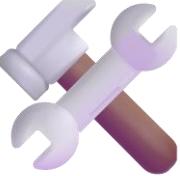


# Čistenie

# Dát



# Predspracovanie Dát



Načítanie dát a ich  
základná analýza

Čistenie a úprava dát

Normalizácia dát

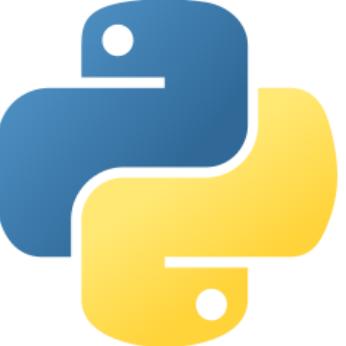
# Python Knižnice



# Čistenie a Úprava Dát

- Nahradíť chýbajúce alebo nevyhovujúce hodnoty: N/A vs. 0, alebo doplniť priemerom/mediánom
- Kartézsky súčin: produkty, farby - biele\_tricko, zelene\_tricko, zelene\_nohavice
- Nelineárne transformácie: priradiť číselné hodnoty do kategórii (napr. vek zákazníkov)
- Domain-specific features: dĺžka, šírka, výška - pridať súčin; váha, výška – BMI
- Variable-specific features: parsovanie stringov
- Normalizácia vstupov: škálovanie a normalizácia podľa strednej hodnoty

# Normalizácia Ret'azcov

 python



# Normalizácia Znakov

- Proces úpravy alebo transformácie znakov tak, aby boli reprezentované konzistentným spôsobom, aj keď môžu byť zapísané rôznymi formami
- Cieľom je zabezpečiť, aby rôzne formy toho istého znaku alebo kombinácie znakov boli rovnako reprezentované v pamäti a v dátach

# Kedy sa používa Normalizácia

1. Pri porovnávaní textov:  
Na zabezpečenie, že rôzne reprezentácie rovnakého textu sú rovnocenné
2. Pri ukladaní textov:  
Na zachovanie konzistentnej formy zápisu
3. Pri spracovaní textu:  
Napríklad pri odstraňovaní diakritiky, kde je potrebné najskôr text rozložiť pomocou NFD a následne odstrániť diakriticke znaky

# Typy Normalizácie

1. **NFC (Normalization Form C) – Kompozitná forma:** Znaky sa skombinujú do jednej formy, ak to je možné. Napríklad "a" + "́" (diakritické znamienko) sa skombinuje do "á".
2. **NFD (Normalization Form D) – Rozložená forma:** Znaky sa rozložia na základné písmeno a diakritické znamienko. Napríklad "á" sa rozloží na "a" + "́".
3. **NFKC (Normalization Form KC) – Kompatibilná kompozitná forma:** Podobná ako NFC, ale navyše zjednocuje aj znaky, ktoré vyzerajú alebo fungujú rovnako, ale nie sú identické (napr. latinské písmeno "VIII" a arabská číslica "8").
4. **NFKD (Normalization Form KD) – Kompatibilná rozložená forma:** Podobná ako NFD, ale navyše zjednocuje znaky na ich kompatibilné ekvivalenty.

[Previous topic](#)[textwrap](#) — Text wrapping  
and filling[Next topic](#)[stringprep](#) — Internet String  
Preparation[This Page](#)[Report a Bug](#)  
[Show Source](#)

# unicodedata — Unicode Database

This module provides access to the Unicode Character Database (UCD) which defines character properties for all Unicode characters. The data contained in this database is compiled from the [UCD version 15.0.0](#).

The module uses the same names and symbols as defined by Unicode Standard Annex #44, [“Unicode Character Database”](#). It defines the following functions:

## `unicodedata.lookup(name)`

Look up character by name. If a character with the given name is found, return the corresponding character. If not found, `KeyError` is raised.

*Changed in version 3.3:* Support for name aliases [1] and named sequences [2] has been added.

## `unicodedata.name(chr[, default])`

Returns the name assigned to the character `chr` as a string. If no name is defined, `default` is returned, or, if not given, `ValueError` is raised.

## `unicodedata.decimal(chr[, default])`

Returns the decimal value assigned to the character `chr` as integer. If no such value is defined, `default` is returned, or, if not given, `ValueError` is raised.

## `unicodedata.digit(chr[, default])`

Returns the digit value assigned to the character `chr` as integer. If no such value is defined, `default` is returned, or, if not given, `ValueError` is raised.

## `unicodedata.numeric(chr[, default])`

Returns the numeric value assigned to the character `chr` as float. If no such value is defined, `default` is returned, or, if not given, `ValueError` is raised.

## `unicodedata.category(chr)`

Returns the general category assigned to the character `chr` as string.

## `unicodedata.bidirectional(chr)`

Returns the bidirectional class assigned to the character `chr` as string. If no such value is defined, an empty string is returned.

## `unicodedata.combining(chr)`

Returns the canonical combining class assigned to the character `chr` as integer. Returns `0` if no combining class is defined.

## `unicodedata.east_asian_width(chr)`

Returns the east asian width assigned to the character `chr` as string.



# Klastrovanie Clustering



 python

The Python logo consists of two interlocking snakes, one blue and one yellow, followed by the word "python" in a lowercase sans-serif font.

# Oblíbené Jedlá

- Bryndzové halušky, pizza, burger, kung pao kurča, tacos, lasagne, guláš, ramen, enchilada, risotto, fried chicken, quesadilla, kapustnica, spaghetti, pad thai, zemiaková placka, pekinská kačica, burrito, pirohy, steak, nachos, tiramisu, chili, dim sum, guacamole, tempura, hot dog, sladkokyslá omáčka, strapačky, jiaozi (knedle), mac and cheese, musaka, mäsové guľky, pho, zemiakový šalát, korma, ceviche, jollof rice, borsch, kimchi, barbecue ribs

# Kategórie podľa Krajiny



- 
- A. Slovenské: bryndzové halušky, kapustnica, zemiaková placka, lokše, strapačky, pirohy, guláš, klobása
  - B. Talianske: pizza, lasagne, risotto, gnocchi, spaghetti, pesto, tiramisu, bruschetta
  - C. Americké: burger, hot dog, steak, barbecue ribs, fried chicken, macaroni and cheese, pancakes, chili
  - D. Čínske: kung pao kurča, jiaozi, chow mein, mapo tofu, pekinská kačica, wonton, sladkokyslá omáčka, pad thai
  - E. Mexické: tacos, burrito, enchilada, guacamole, quesadilla, nachos, mole, ceviche

# Oblíbené Jedlá



1. bryndzové halušky

2. pizza

3. lasagne

4. spaghetti

5. burger

6. hot dog

7. pekinská kačica

8. quesadilla

# Klastrovanie



## Hlavná ingredencia: (mäso, cestoviny, ryža)

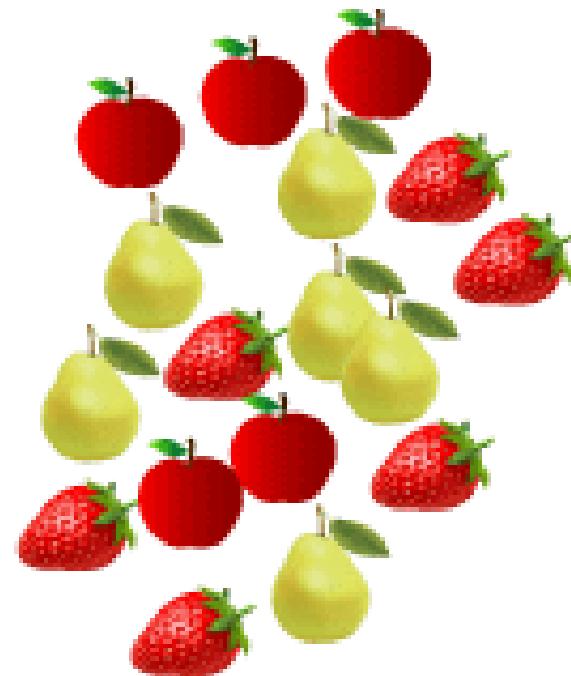
- Skupina 1: Burger, Fried Chicken, Steak, Barbecue Ribs, Pekinská kačica
- Skupina 2: Pizza, Lasagne, Spaghetti, Mac and Cheese
- Skupina 3: Risotto, Kung Pao Kurča, Korma, Pad Thai

## Spôsob prípravy: (vyprážané, dusené, grilované)

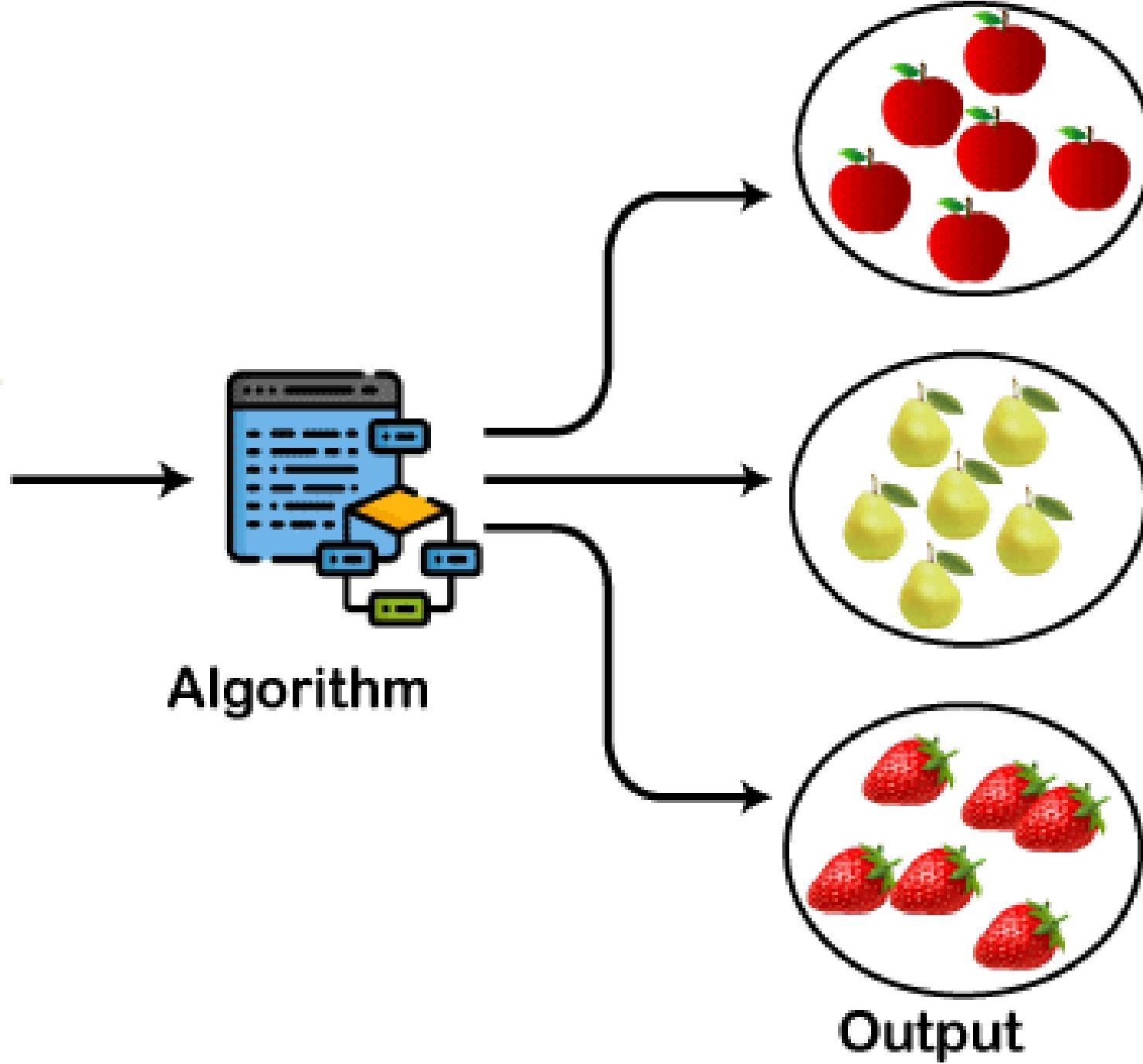
- Skupina 1: Fried Chicken, Tempura, Zemiaková Placka
- Skupina 2: Guláš, Borscht, Kapustnica
- Skupina 3: Barbecue Ribs, Steak

# Občerstvenie

- Bryndzové halušky, pivo, pizza, burger, kung pao kurča, tacos, lasagne, guláš, ramen, enchilada, risotto, fried chicken, quesadilla, džús, kapustnica, spaghetti, pad thai, voda, zemiaková placka, pekinská kačica, burrito, pirohy, steak, nachos, tiramisu, chili, dim sum, guacamole, tempura, víno, hot dog, sladkokyslá omáčka, strapačky, jiaozi (knedle), mac and cheese, musaka, mäsové guľky, pho, zemiakový šalát, korma, ceviche, jollof rice, borscht, kimchi, barbecue ribs, coca cola



Raw Data



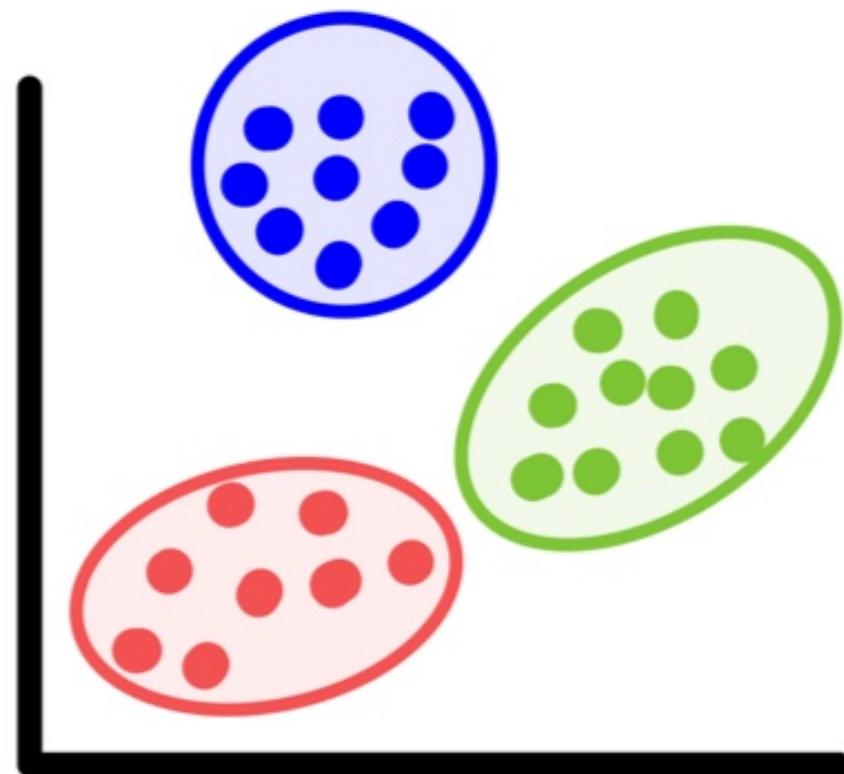
# Klastrovanie

- **Technika strojového učenia**
- Používa na **zoskupenie podobných objektov do skupín (klastre)**
- Knižnica **Scikit-learn**
- Metódy klastrovania
  1. K-means klastrovanie
  2. Hierarchické klastrovanie
  3. DBSCAN
  4. Mean Shift

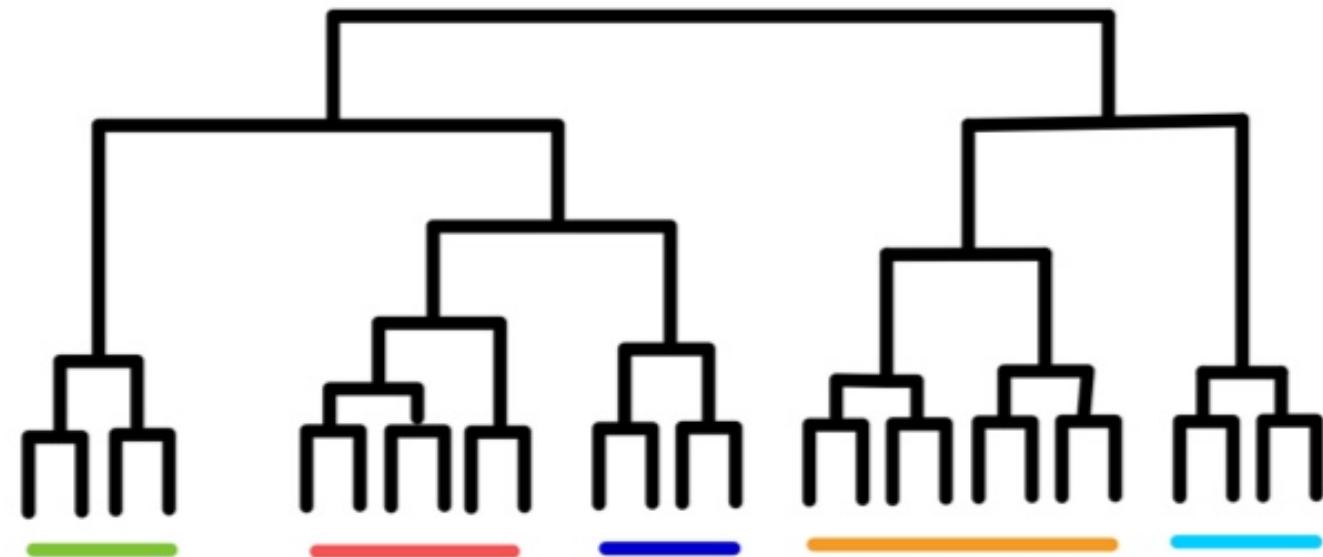


# Klastrovanie

Partitioning clustering

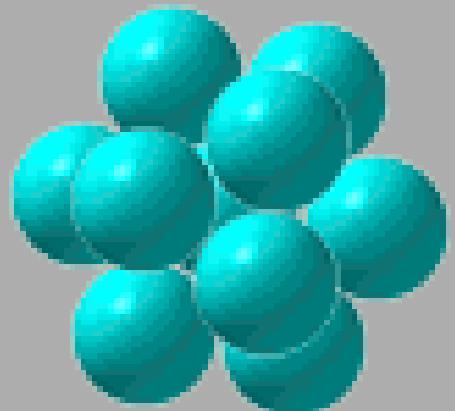


Hierarchical clustering

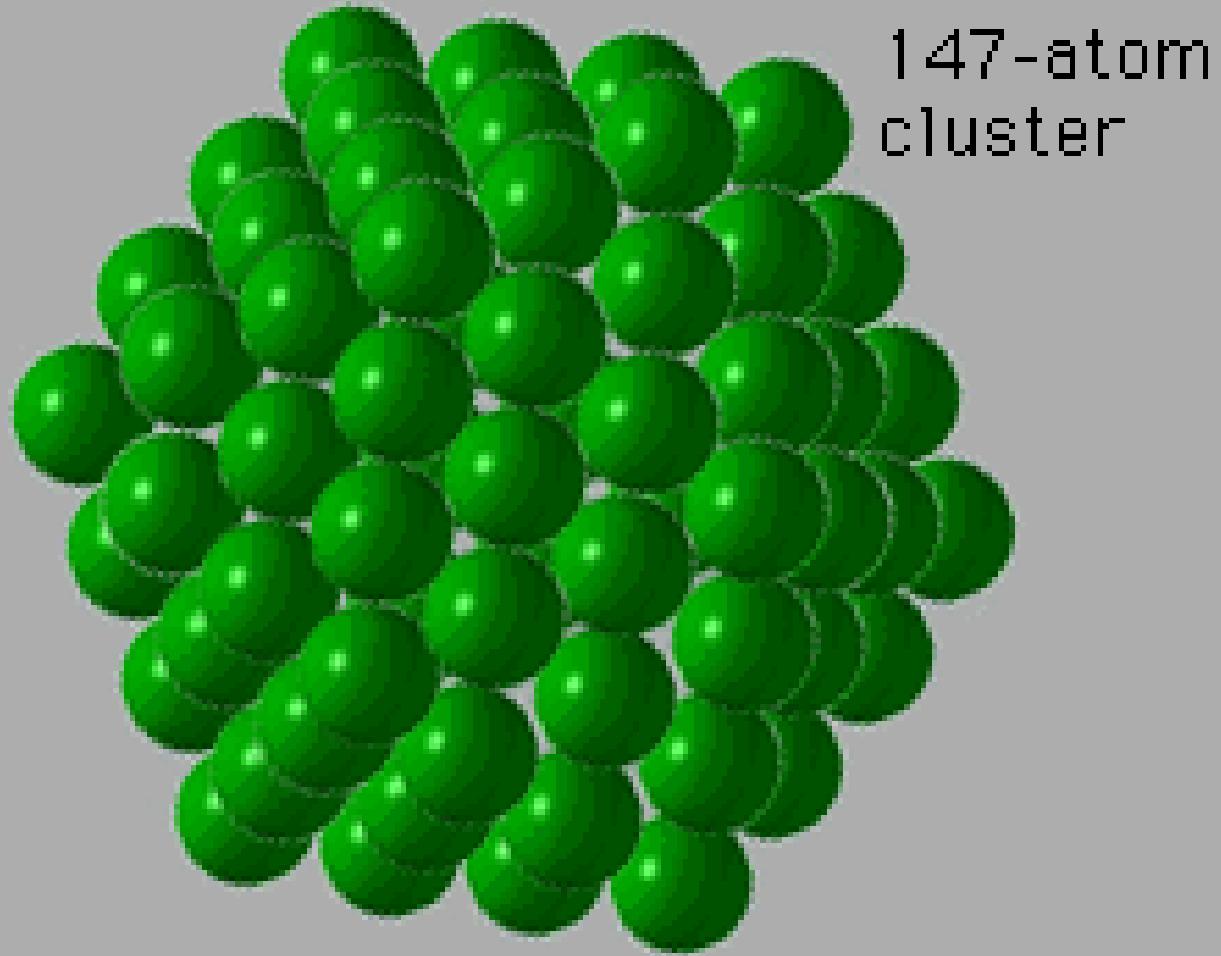


# Biznis Klastrovanie

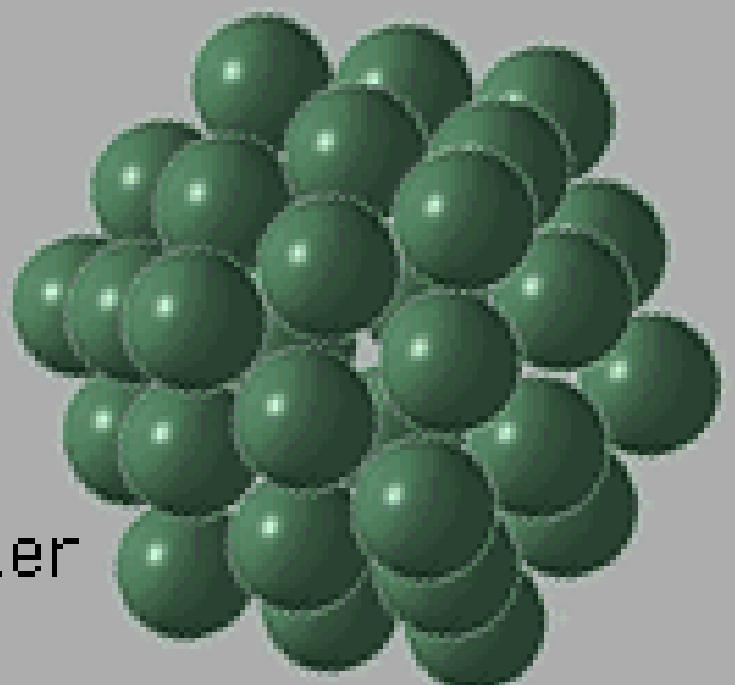




13-atom cluster



147-atom cluster



55-atom cluster

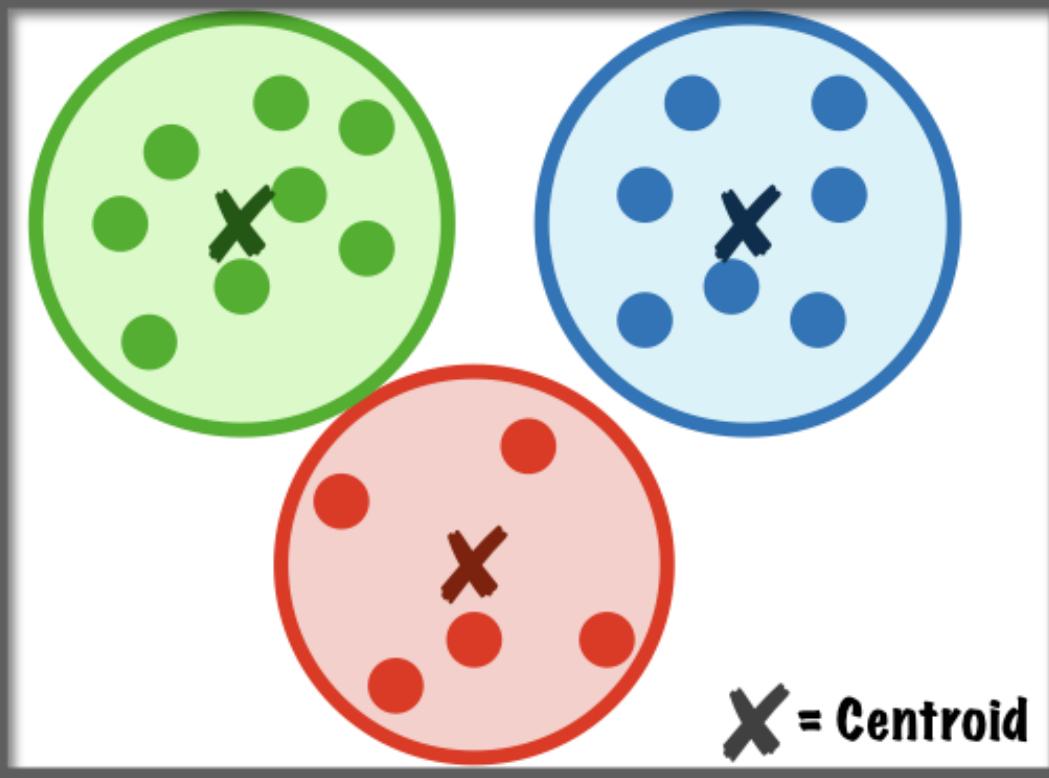


Unlabelled Data



K-means  
→

Labelled Clusters



**X** = Centroid



Knižnice  
Klastre

## TOOLS

# FAISS

FAISS (Facebook AI Similarity Search) is a library that allows developers to quickly search for embeddings of multimedia documents that are similar to each other. It solves limitations of traditional query search engines that are optimized for hash-based searches, and provides more scalable similarity search functions.

## Efficient similarity search

With FAISS, developers can search multimedia documents in ways that are inefficient or impossible with standard database engines (SQL). It includes nearest-neighbor search implementations for million-to-billion-scale datasets that optimize the memory-speed-accuracy tradeoff. FAISS aims to offer state-of-the-art performance for all operating points.

FAISS contains algorithms that search in sets of vectors of any size, and also contains supporting code for evaluation and parameter tuning. Some of its most useful algorithms are implemented on the GPU. FAISS is implemented in C++, with an optional Python interface and GPU support via CUDA.

GitHub

Join Community

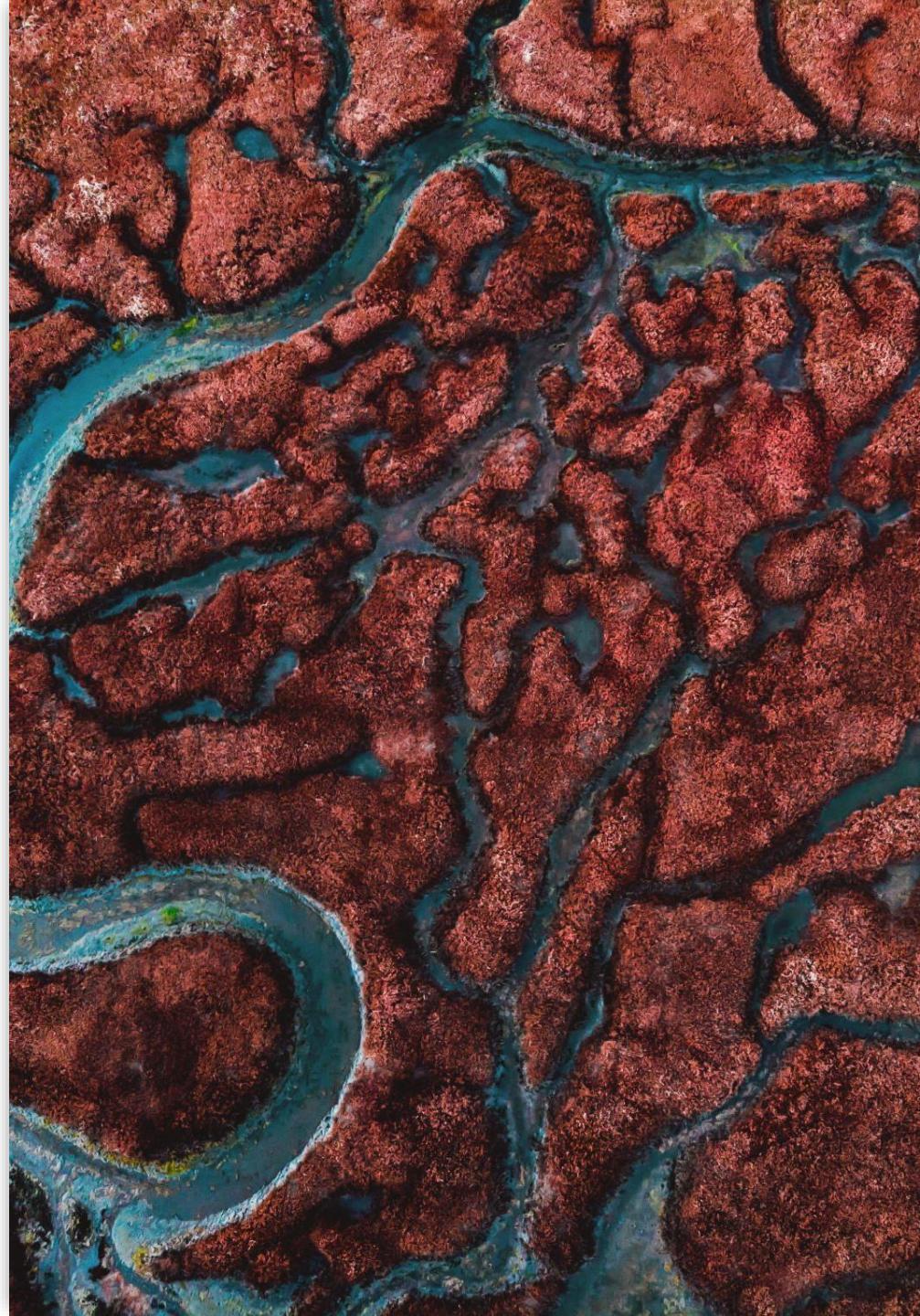
# Metódy Klastrovania

**1. K-means klastrovanie:** delí dátu na K klastrov, pričom K je vopred určený počet. Algoritmus sa snaží minimalizovať súčet štvorcov vzdialenosí medzi bodmi a ich najbližším centrom klastra

**2. Hierarchické klastrovanie:** vytvára strom podobnosti medzi jednotlivými objektami, čo umožňuje analyzovať dátu na rôznych úrovniach detailu. Môže byť aglomeratívna (zdola nahor) alebo deliaca (zhora nadol)

**3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** zoskupuje spolu body, ktoré sú tesne susediace a označuje body v šumových oblastiach ako výnimky. DBSCAN je obzvlášť užitočný, keď sú dátu nerovnomerne rozložené alebo keď existujú šumové body

**4. Mean Shift:** nachádza klastre tak, že posúva kandidátske centroidy smerom k najvyššej hustote dátových bodov



# Metódy Klastrovania

## K-means

- Najvhodnejšie pre dáta, ktoré sú dobre oddeliteľné a majú približne guľatý alebo sférický tvar klastrov
- Je efektívne pre veľké dátové sady, ale vyžaduje, aby bol počet klastrov K známy vopred
- Často sa používa v marketingovej segmentácii, organizácii dokumentov a obrazových analýzach, kde sú klastre relatívne rovnomerne rozložené

## Hierarchické

- Ked' je potrebné analyzovať vzťahy medzi objektami na rôznych úrovniach podrobnosti
- Je vhodná pre dáta, kde je dôležité vizuálne posúdiť a interpretovať štruktúru klastrov, napríklad v genetických alebo sociologických štúdiách
- Užitočné aj ked' nevieme vopred počet klastrov

# Metódy Klastrovania

## DBSCAN

- Najlepšie použiteľný na dátu s komplexnými geometrickými rozloženiami alebo dátu, kde sú prítomné šumové body
- Ideálne pre geopriestorové dátu, identifikáciu skupín bodov v šumovom prostredí a akékoľvek situácie, kde klastry nie sú sférického tvaru
- Môže automaticky detegovať počet klastrov na základe hustoty

## Mean Shift

- Vhodný pre dátu, kde klastre nemajú jasný alebo uniformný tvar
- Je účinný pri vizuálnej segmentácii obrazu a analýze dát, kde klastry nemusia mať rovnakú veľkosť alebo hustotu
- Je dobrý na prácu s obrazovými dátami a situáciami, kde je hustota dát kľúčovým faktorom pri tvorbe klastrov

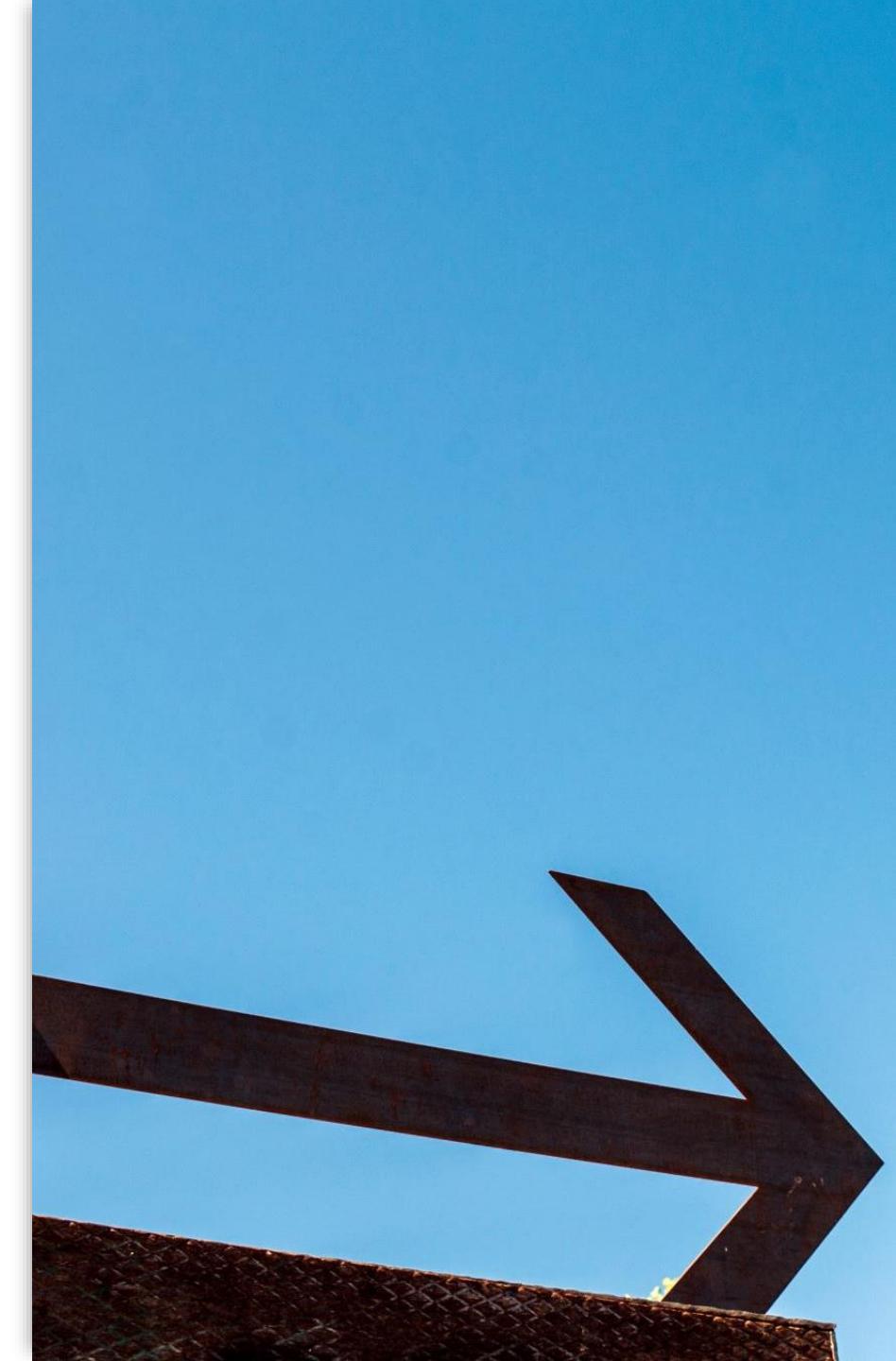
# Klastrovanie K-means

- Populárna metóda klastrovania
- Používa na **rozdelenie súboru dát na skupiny (klustre)** na základe ich **podobnosti**
- Cieľom je **zoskupiť dátá tak**, aby body v rámci jedného klustra boli čo **najviac podobné**
- Zatiaľ čo **body v rôznych klustroch boli čo najviac odlišné**

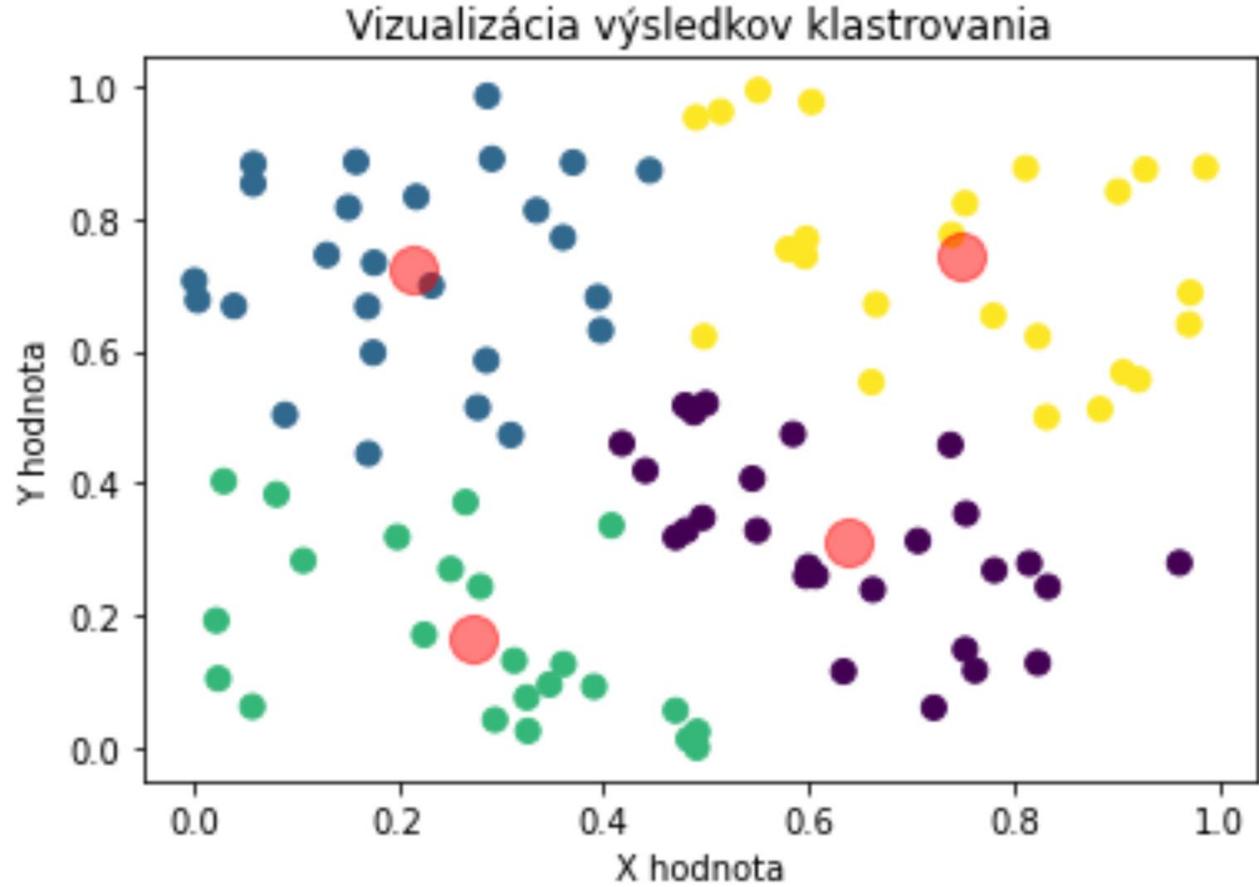
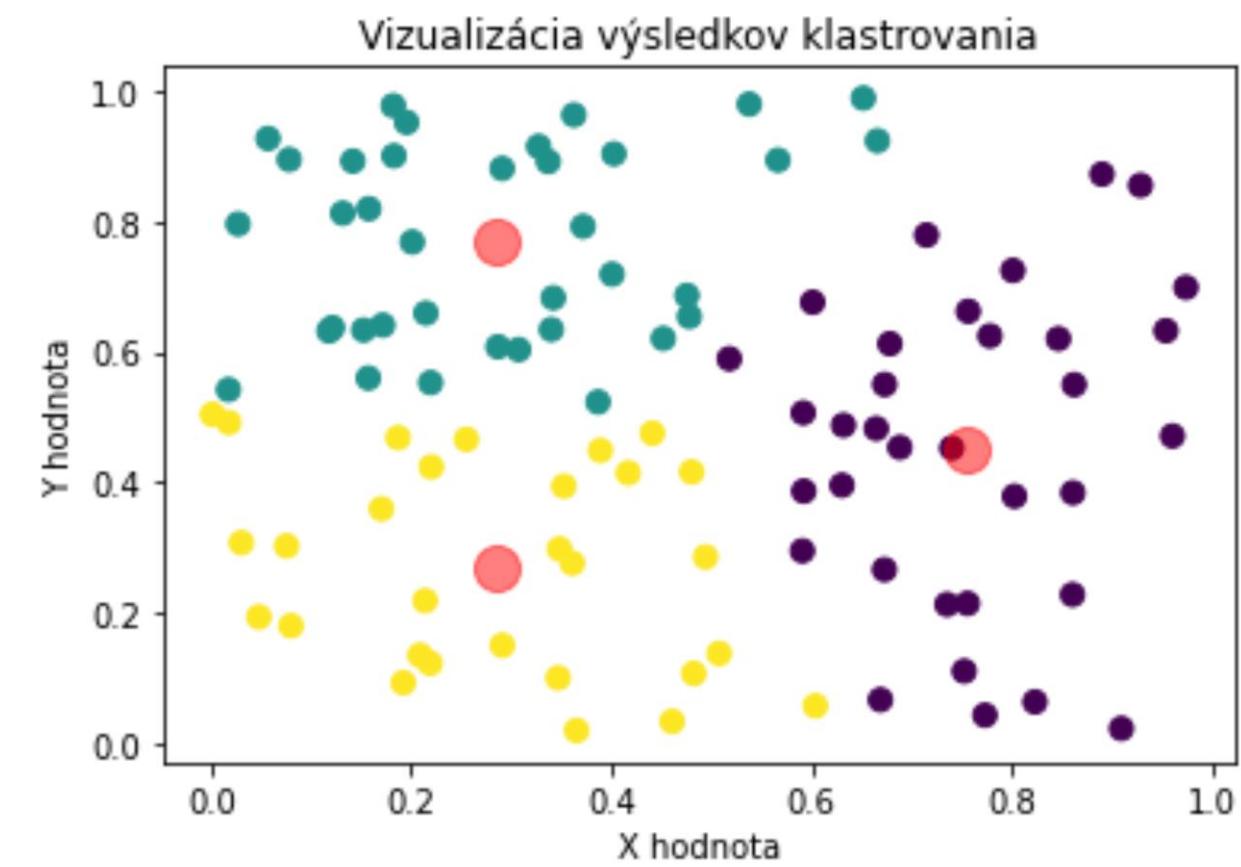


# Klastrovanie K-means

- **Segmentácia zákazníkov:** Rozdelenie zákazníkov do klastrov podľa ich nákupných zvyklostí alebo preferencií, čo môže pomôcť pri cielenom marketingu a optimalizácii produktových ponúk
- **Analytika sociálnych médií:** Zoskupenie používateľov alebo príspevkov na základe podobnosti obsahu alebo správania na sociálnych sietiach
- **Biologické vedy a medicína:** Klastrovanie genetických dát na identifikáciu skupín podobných variantov alebo pacientov s podobnými profilmami ochorení
- **Finančná analýza:** Identifikácia skupín podobných finančných produktov alebo trhových správaní, čo môže byť užitočné pri správe rizika alebo investičnom rozhodovaní

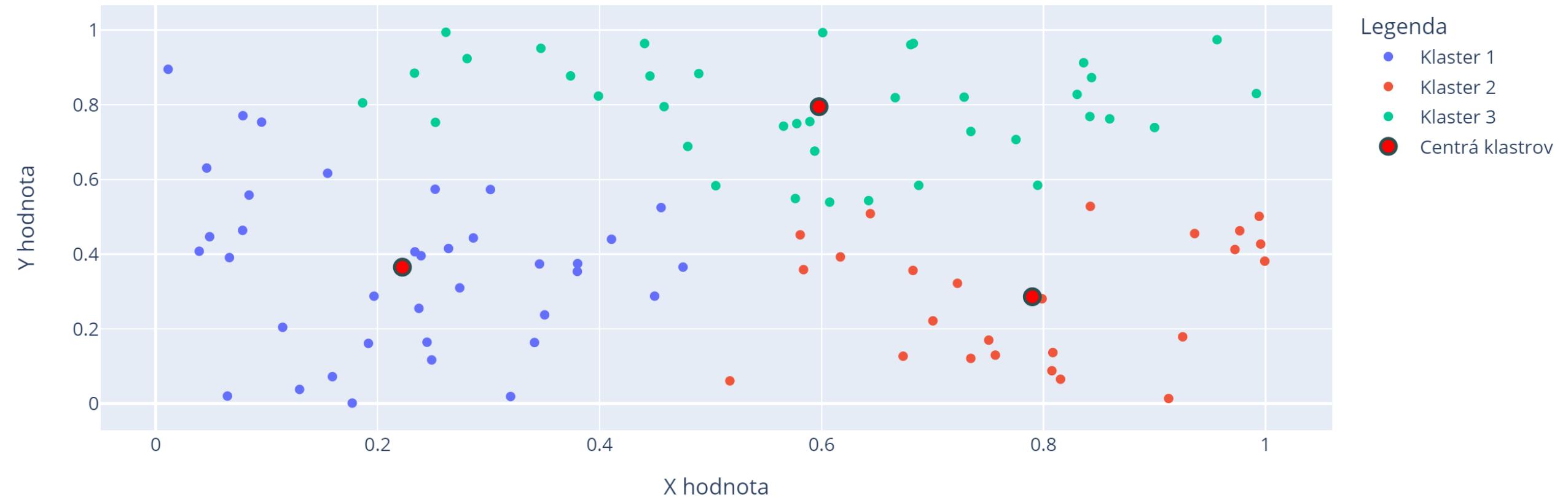


# Vizualizácia Matplotlib



# Vizualizácia Plotly

Vizualizácia výsledkov klastrovania s použitím Plotly



# Obmedzenia K-means

- **Musí byť vopred zvolený počet klastrov K:**  
Toto môže byť problematické, ak nevieme, kol'ko klastrov by malo byť v dátach
- **Citlivosť na výber počiatočných centroidov:**  
Náhodný výber môže viest' k rôznym výsledkom, pre konzistentnejšie výsledky sa niekedy používa metóda ako K-means++
- **Nie je vhodný pre klastre nepravidelných tvarov alebo rôznych veľkostí:** K-means najlepšie funguje, keď sú klastre približne rovnakej veľkosti a sférickej forme



# scikit-learn

## Machine Learning in Python

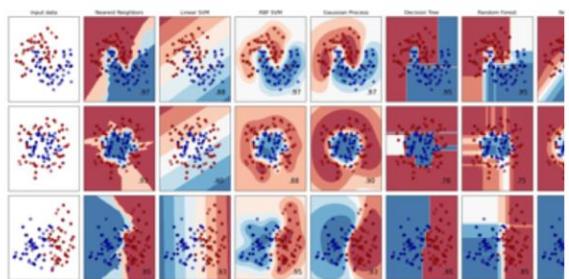
[Getting Started](#)[Release Highlights for 1.4](#)[GitHub](#)

- 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...](#)

[Examples](#)

### Dimensionality reduction

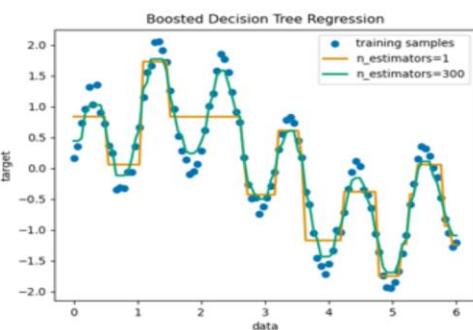
Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency  
**Algorithms:** PCA, feature selection, non-negative matrix factorization, and [more...](#)

### Regression

Predicting a continuous-valued attribute associated with an object.

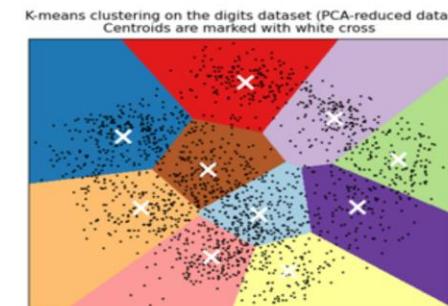
**Applications:** Drug response, Stock prices.  
**Algorithms:** Gradient boosting, nearest neighbors, random forest, ridge, and [more...](#)

[Examples](#)

### Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes  
**Algorithms:** k-Means, HDBSCAN, hierarchical clustering, and [more...](#)

[Examples](#)

### Model selection

Comparing, validating and choosing parameters and models.

**Applications:** Improved accuracy via parameter tuning  
**Algorithms:** grid search, cross validation, metrics,

### Preprocessing

Feature extraction and normalization.

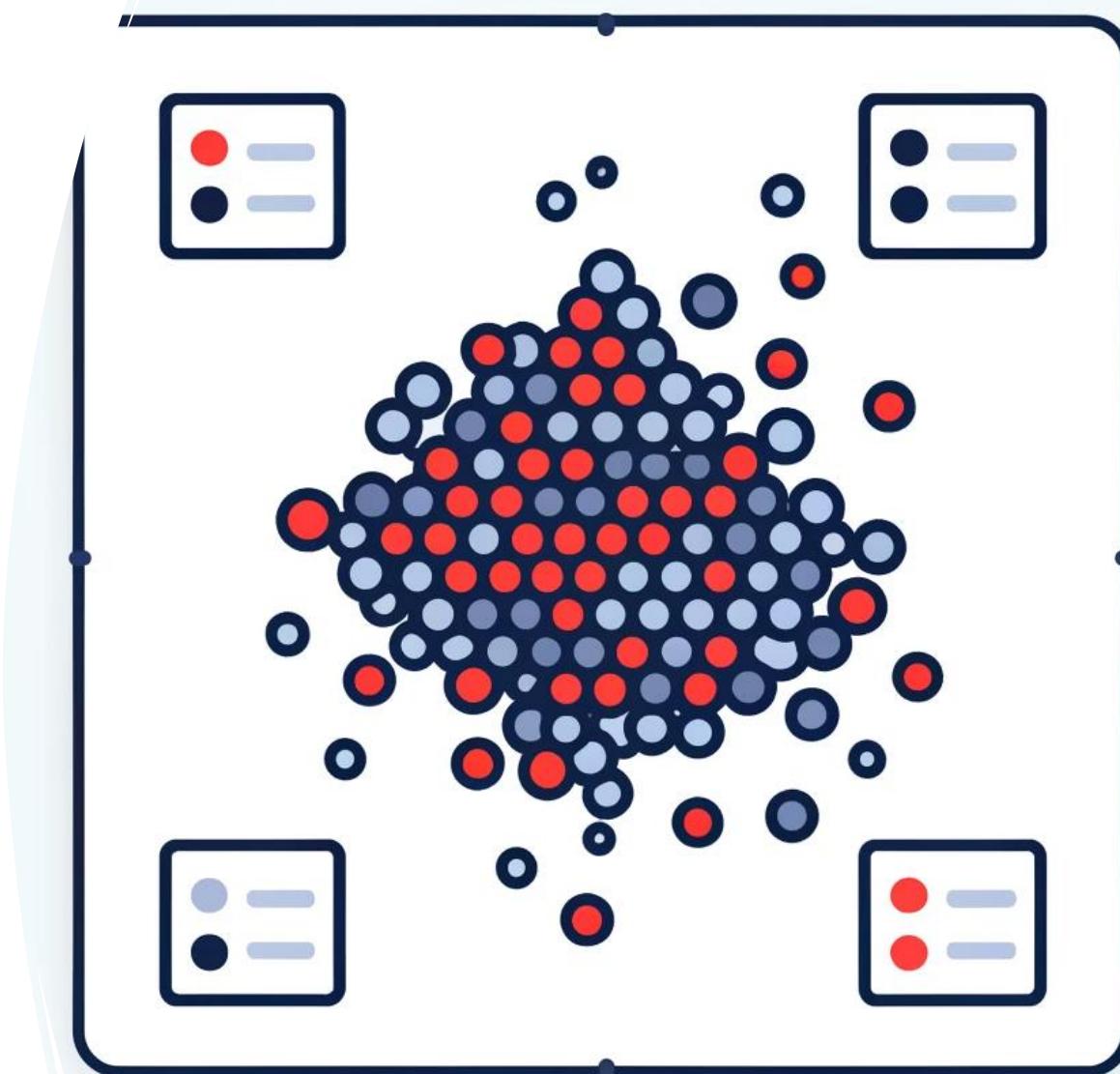
**Applications:** Transforming input data such as text for use with machine learning algorithms.  
**Algorithms:** preprocessing, feature extraction, and [more...](#)

# Algoritmus K-means

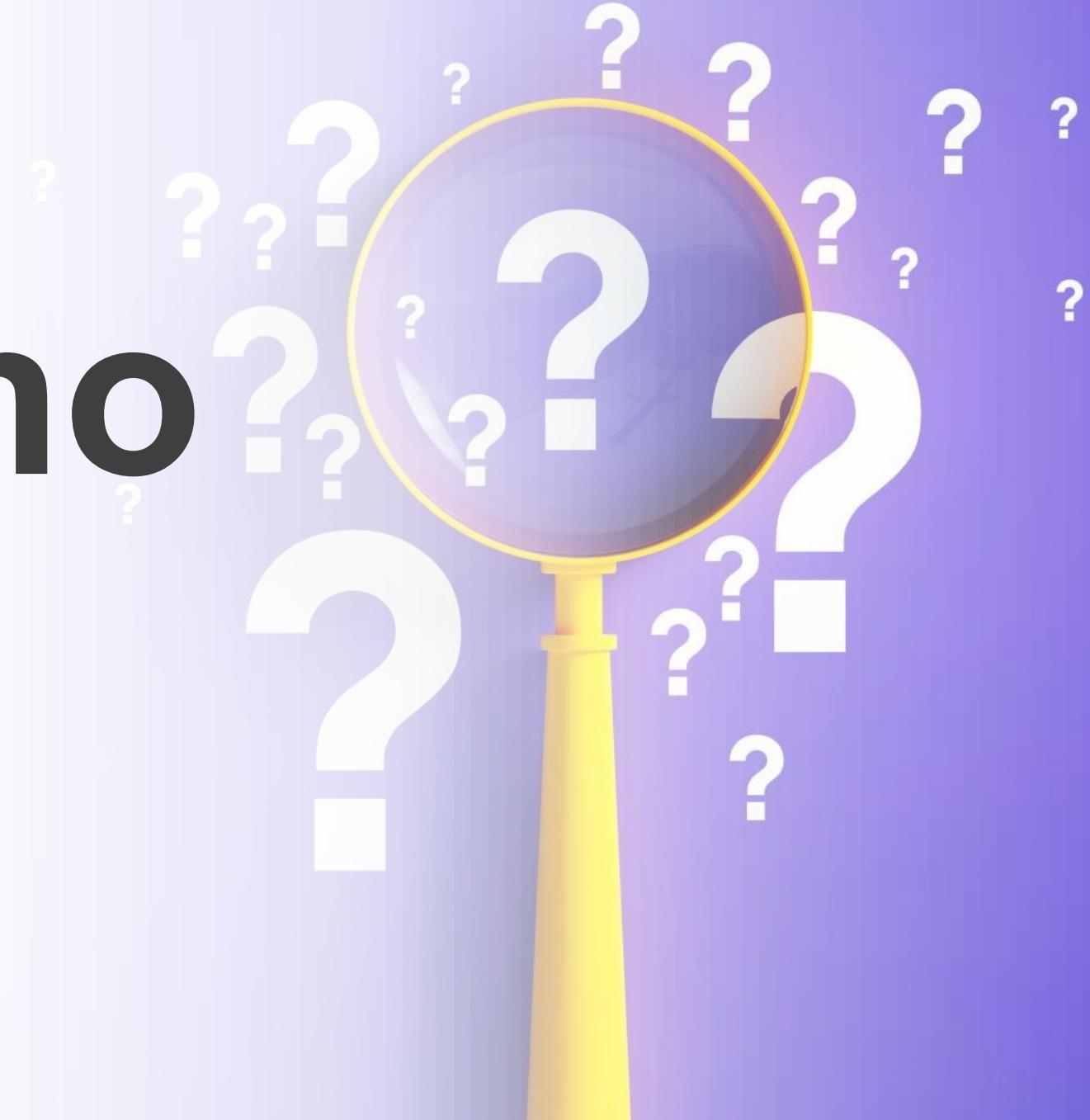
- 1. Inicializácia:** Na začiatku si vyberieme počet klustrov  $K$ , ktoré chceme nájsť. Následne náhodne vyberieme  $K$  bodov z našich dát, ktoré budú slúžiť ako **počiatočné stredy klustrov (centroidy)**.
- 2. Priradenie bodov do klustrov:** **Každý bod v našom datasete priradíme do klustra tak, že najdeme najbližší centroid.** Vypočítame vzdialenosť bodu od všetkých centroidov a priradíme ho k tomu centroidu, ktorý je k bodu najbližšie. Na výpočet vzdialenosťi sa často používa euklidovská vzdialenosť.
- 3. Aktualizácia centroidov:** Po priradení všetkých bodov do klustrov prepočítame polohu centroidov. To urobíme tak, že pre každý kluster vypočítame priemerné súradnice všetkých bodov, ktoré do tohto klustra patria. Tieto priemerné hodnoty stanú sa novými centroidmi.
- 4. Opakovanie:** Kroky 2 a 3 opakujeme iteratívne, až kým sa **poloha centroidov stabilizuje**, čiže sa už **nezmení**, alebo **zmena je menšia než určitá vopred stanovená prahová hodnota**. Taktiež môžeme proces zastaviť po dosiahnutí maximálneho počtu iterácií, ak je to potrebné.

# Úlohy Klastre

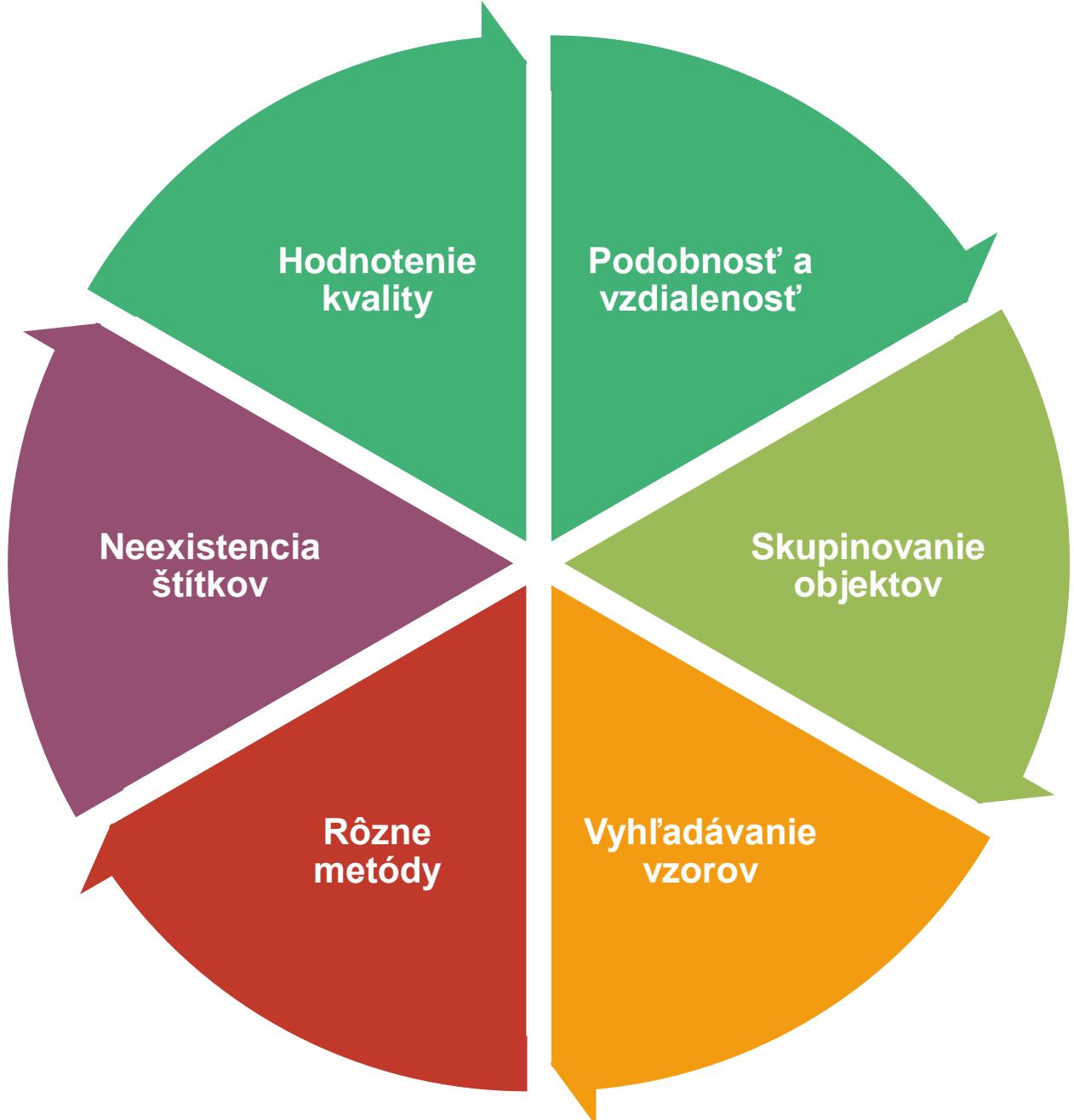
1. Pochopenie čo je klastrovanie
2. Prehľad medzi metódy klastrovania
3. Používanie klastrovania v Pythone
4. Pochopenie algoritmu k-means



# Princípy Strojového Učenia



# Základné Princípy



# Clustering Vzdialenosť

## Euklides

- Definícia:  
Meria **priamu vzdialosť** medzi 2 bodmi  
V 2 rozmeroch je definovaná ako:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Vlastnosti:  
Je **intuitívna** a **zodpovedá „priamke“** medzi 2 bodmi. Vhodná je pre **dáta**, kde má význam táto **priama vzdialosť**

## Manhattan

- Definícia:  
Meria vzdialosť medzi 2 bodmi ako **súčet absolútnych rozdielov ich súradníc**  
V 2 rozmeroch sa vypočíta ako:

$$d = |x_2 - x_1| + |y_2 - y_1|$$

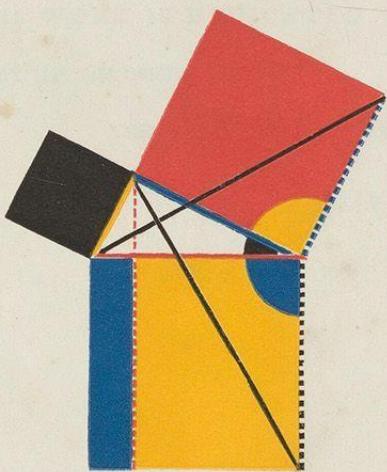
- Vlastnosti:  
Je známa aj ako „**vzdialosť v meste**“ (city block distance), pretože **zohľadňuje len pohyb v horizontálnych a vertikálnych smeroch**, nie **diagonálne**. Vhodná je pre dátá, kde sú **pohyby obmedzené**, ako v **mestských prostrediach**

THE FIRST SIX BOOKS OF  
**THE ELEMENTS OF EUCLID**  
 IN WHICH COLOURED DIAGRAMS AND SYMBOLS  
 ARE USED INSTEAD OF LETTERS FOR THE  
 GREATER EASE OF LEARNERS



BY OLIVER BYRNE

SURVEYOR OF HER MAJESTY'S SETTLEMENTS IN THE FALKLAND ISLANDS  
 AND AUTHOR OF NUMEROUS MATHEMATICAL WORKS



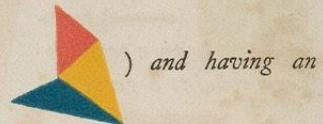
LONDON  
 WILLIAM PICKERING  
 1847

46

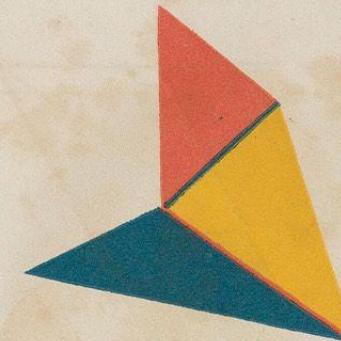
BOOK I. PROP. XLV. PROB.



O construct a parallelogram equal  
 to a given rectilinear figure



( + ) and having an  
 angle equal to a given rectilinear angle  
 ().



Draw —— and —— dividing  
 the rectilinear figure into triangles.



Construct =

having = (pr. 42.)

to —— apply =

having = (pr. 44.)

to —— apply =

having = (pr. 44.)

∴ + =

and + is a parallelogram. (prs. 29, 14, 30.)

having =

Q. E. D.

# Clustering Vzdialenosť

## Euklides

1. Geografické umiestnenia
2. Robotika
3. Počítačová grafika
4. Strojové učenie
5. Biológia
6. Fyzika

## Manhattan

1. Mestská navigácia
2. Optimalizácia logistiky
3. Analýza dát
4. Hranie hier
5. Dopravné modelovanie
6. Strojové učenie

# Clustering Vzdialenosť

## Euklides

- **Príklad:**  
Predpokladajme, že máme 2 body v 2D priestore: A(1, 2) a B(4, 6).

## Výpočet:

$$d = \sqrt{(4 - 1)^2 + (6 - 2)^2} =$$

$$\sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$$

- **Interpretácia:**  
Priama vzdialenosť medzi bodmi A a B je 5 jednotiek.

## Manhattan

- **Príklad:** Zoberme rovnaké body A(1, 2) a B(4, 6).

## Výpočet:

$$d = |4 - 1| + |6 - 2| = 3 + 4 = 7$$

- **Interpretácia:**  
Vzdialenosť medzi bodmi A a B, ak sa pohybujeme len horizontálne a vertikálne, je 7 jednotiek.



## Step-by-Step Solutions with Pro

Get a step ahead with your homework

[Go Pro Now](#)

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA

 WolframAlpha[NATURAL LANGUAGE](#)[MATH INPUT](#)[EXTENDED KEYBOARD](#)[EXAMPLES](#)[UPLOAD](#)[RANDOM](#)

Input

distance between

{1, 2}

{4, 6}

Result

5

[Download Page](#)

POWERED BY THE WOLFRAM LANGUAGE

Related Queries:

[multiplication table](#)  
[integer arithmetic](#)

[sqrt\(144\), cos\(0\), sin\(pi/2\), ln\(e^2\), e^0](#)  
[where the heck is Matt?](#)





## Step-by-Step Solutions with Pro

Get a step ahead with your homework

[Go Pro Now](#)

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA

[NATURAL LANGUAGE](#)[MATH INPUT](#)[EXTENDED KEYBOARD](#)[EXAMPLES](#)[UPLOAD](#)[RANDOM](#)

Input

ManhattanDistance[{1, 2}, {4, 6}]

Result

7

[Download Page](#)

POWERED BY THE WOLFRAM LANGUAGE

Related Queries:

- = sqrt(144), cos(0), sin(pi/2), ln(e^2), e^0
- = multiplication table
- = scientific calculators

- = integer arithmetic
- = calories burned while studying vs sleeping vs doing n...





points (0,0), (1, 2), (4, 6)



NATURAL LANGUAGE

MATH INPUT

EXTENDED KEYBOARD

EXAMPLES

UPLOAD

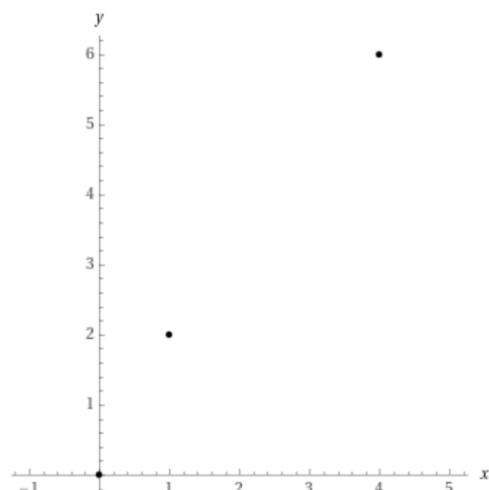
RANDOM

Assuming "points" is referring to geometry | Use as a unit instead

## Input interpretation

point	coordinates (0, 0)		point	coordinates (1, 2)	
point	coordinates (4, 6)				

## Visual representation



## Distances

	(0, 0)	(1, 2)	(4, 6)
(0, 0)	0	$\sqrt{5} \approx 2.23607$	$2\sqrt{13} \approx 7.2111$
(1, 2)	$\sqrt{5} \approx 2.23607$	0	5
(4, 6)	$2\sqrt{13} \approx 7.2111$	5	0

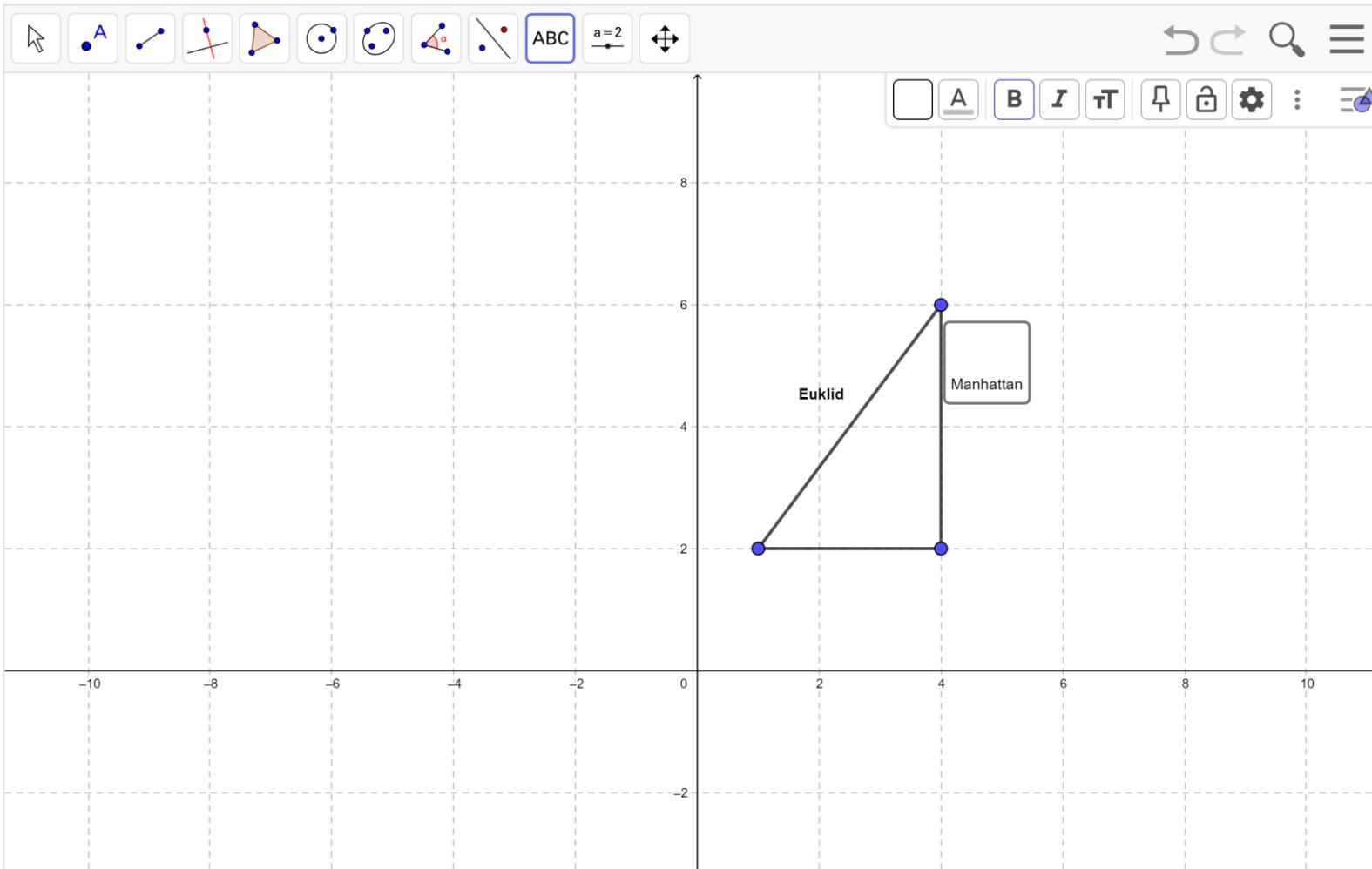


## Coordinate Plane

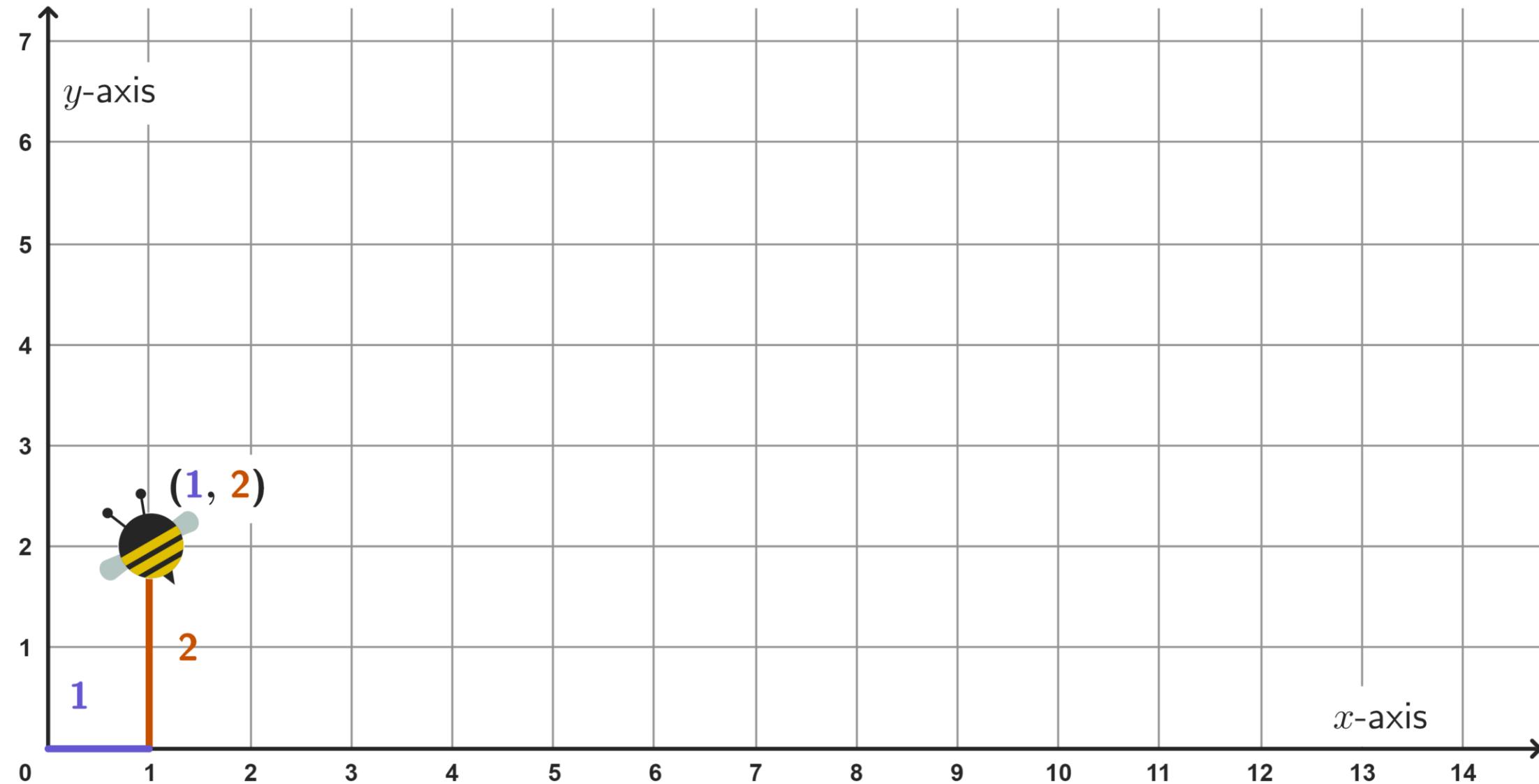
Author: Celia Jimenez

⋮

Coordinate Plane

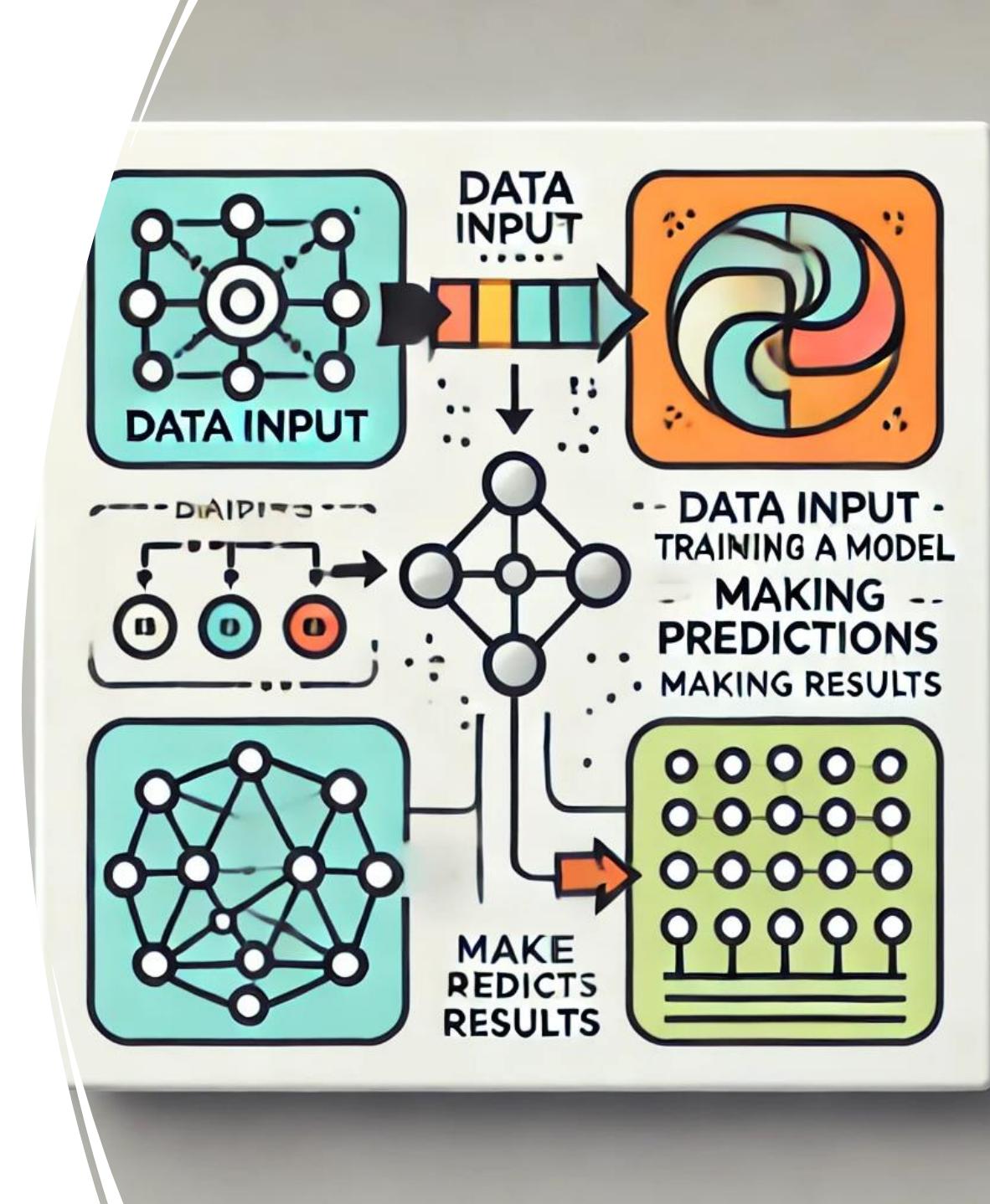


Move the bee and observe how the coordinates change!



# Úlohy ML Princípy

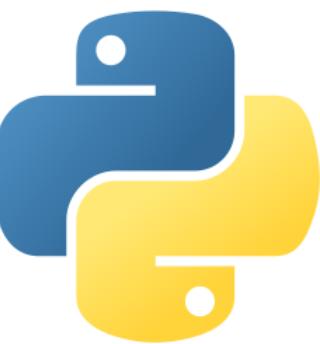
1. Pochopenie princípov ML
2. Praktické príklady euklides a manhattan



# Klastrovanie

## Typy

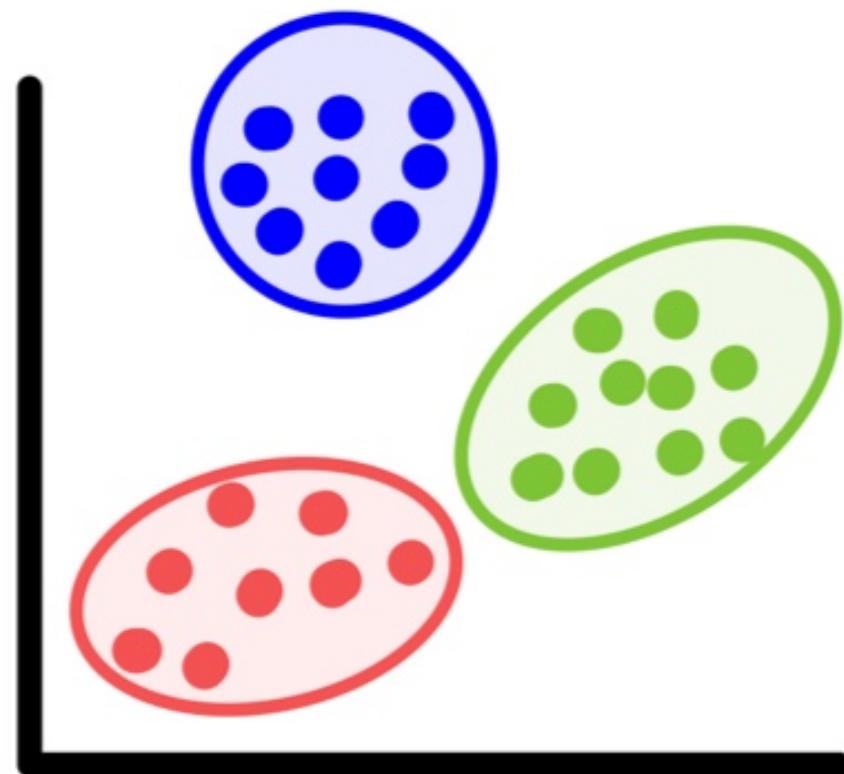


 python

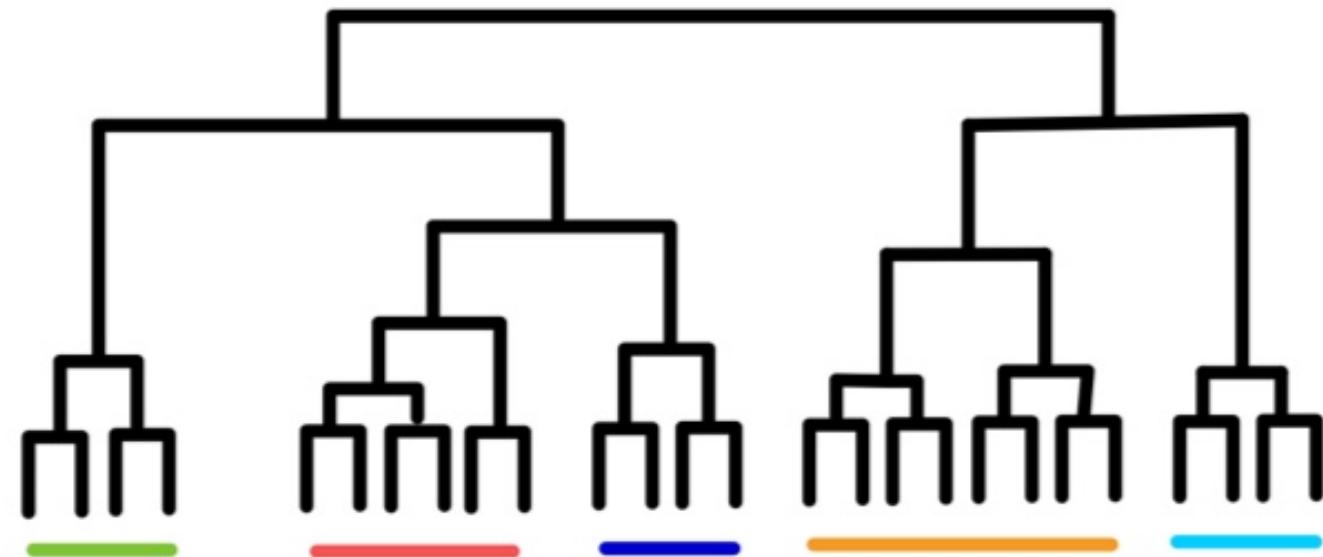


# Klastrovanie

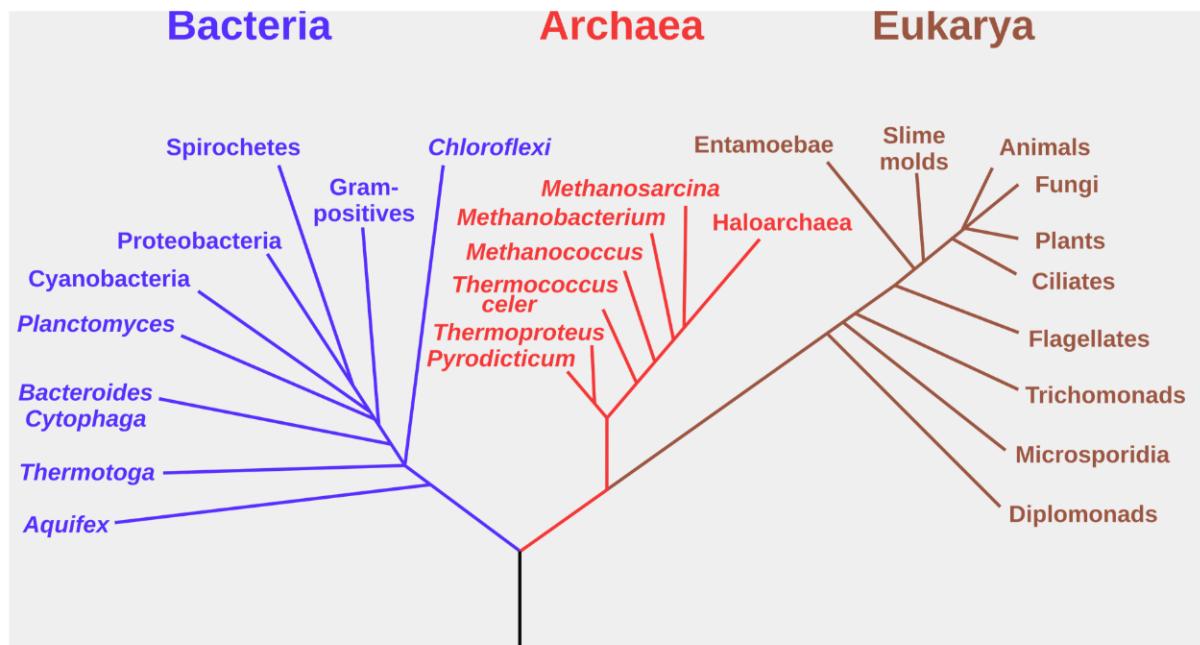
Partitioning clustering



Hierarchical clustering



# Čo je Dendrogram



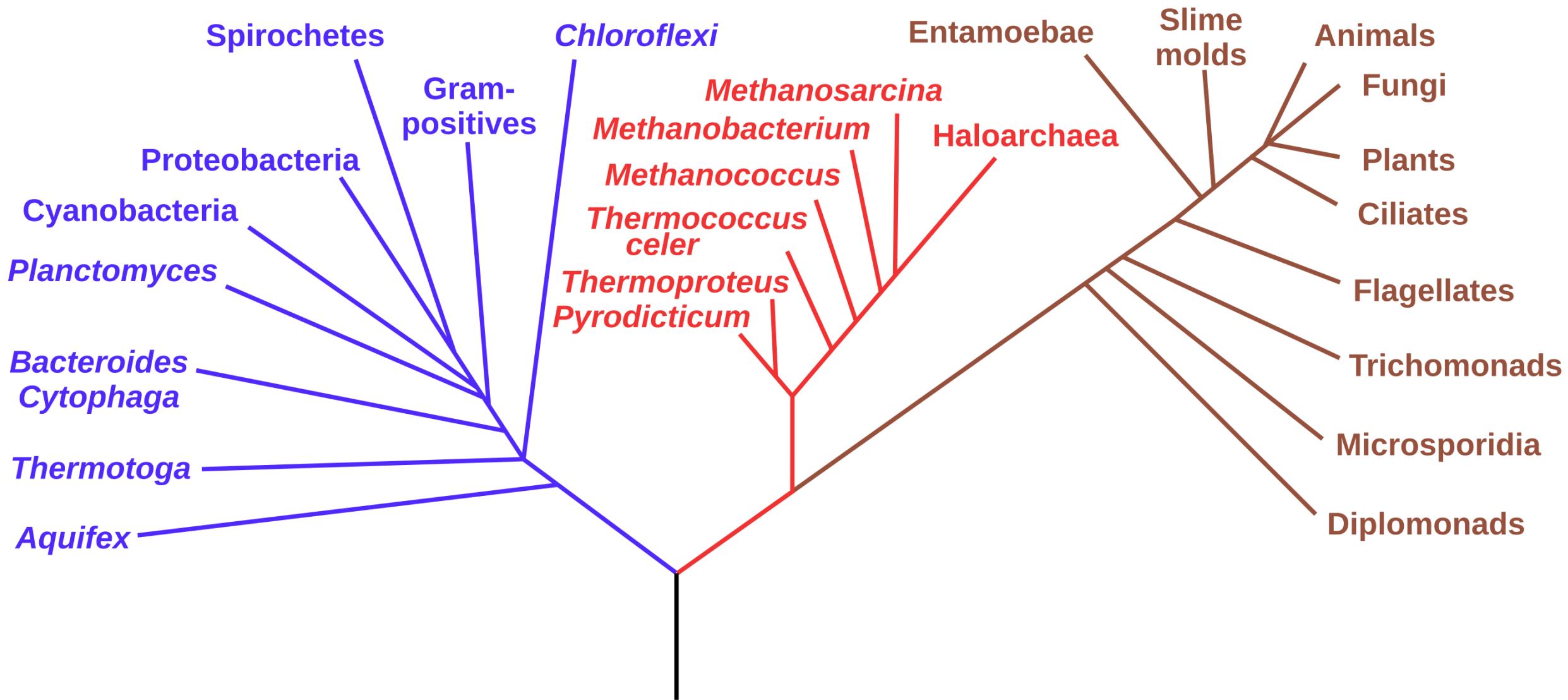
- **Druh diagramu**, používaný na znázornenie jednotlivých krokov **klastrovej analýzy**
- Pri postupe výpočtu metódou aglomeratívneho dendrogramu **vyjadruje každý prvok samostatne na zvislej osi**
- **Horizontálna os** vyjadruje **vzdialenosť** medzi **jednotlivými zhľukmi**
- Klastre sa zjednocujú podľa najkratšej vzdialnosti (či už merané **Euklidovskou vzdialenosťou** alebo **inou**, alebo použitím akejkoľvek metódy počítania vzdialostí (najbližšieho suseda, najvzdialenejšieho, centroidné (tažisko) atď.).)

# Bacteria

# Archaea

# Eukarya

101



# Hierarchické Klastrovanie



- Metóda zoskupovania dátových bodov, ktorá funguje na princípe postupného spájania jednotlivých bodov do väčších a väčších skupín (klastrov)
- Tento proces sa nazýva **zdola nahor** (anglicky **bottom-up**)
- Začíname tým, že každý dátový bod je samostatným klastrom, a postupne ich spájame do väčších klastrov, až kým neostane len 1 veľký klaster alebo dosiahneme vopred definovaný počet klastrov

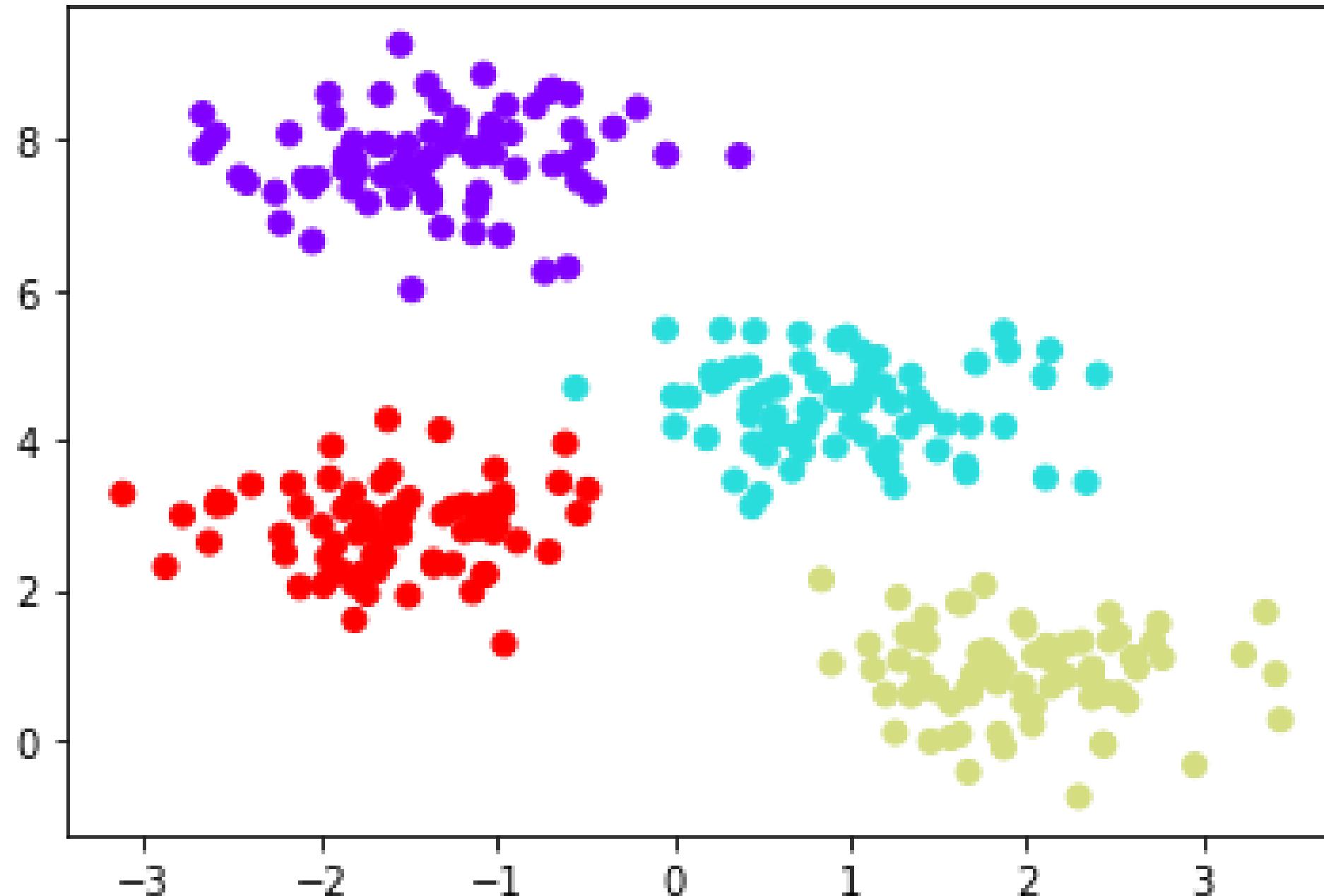
# Ako to funguje?

1. **Začiatok:** Na začiatku je každý bod v dátach považovaný za svoj vlastný klaster.
2. **Výpočet vzdialenosťí:** Algoritmus vypočíta vzdialenosťi medzi všetkými párami klastrov (na začiatku sú to jednotlivé body) pomocou rôznych metód, ako je napríklad euklidovská vzdialenosť.
3. **Spájanie klastrov:** V každom kroku algoritmus spojí dva najbližšie klastre do jedného nového klastru.
4. **Opakovanie:** Tento proces sa opakuje, pričom sa vždy spoja dva najbližšie klastre, až kým sa nevytvorí vopred určený počet klastrov alebo nevznikne jeden veľký klaster, ktorý obsahuje všetky body.
5. **Dendrogram:** Hierarchické klastrovanie sa často vizualizuje pomocou dendrogramu, čo je graf, ktorý ukazuje, ako sa jednotlivé body spájali do väčších klastrov v priebehu algoritmu.

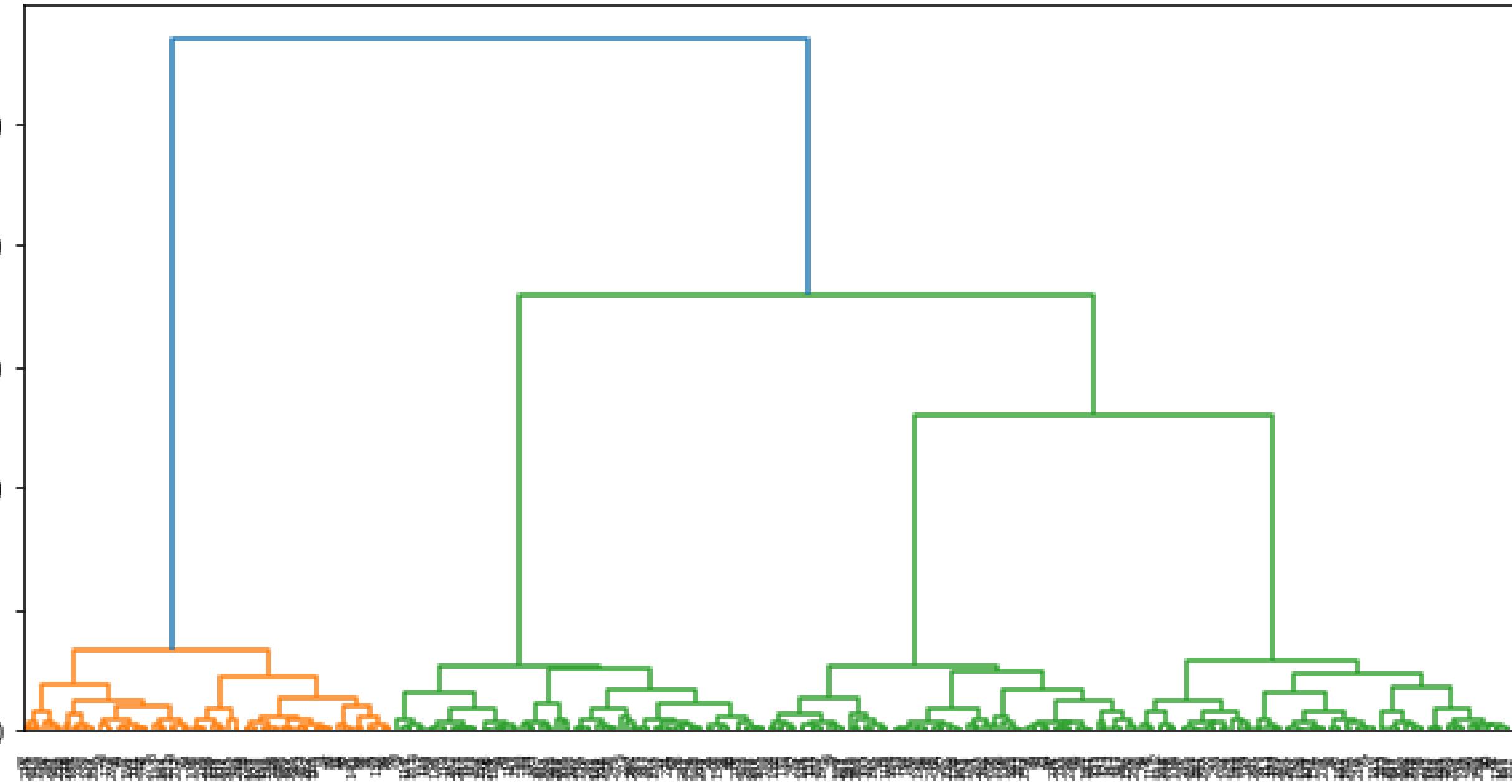
# Hierarchické Klastrovanie

- **Neurčujeme počet klastrov dopredu:** Na rozdiel od algoritmov ako K-Means, nemusíme vopred vedieť, koľko klastrov chceme vytvoriť.
- **Flexibilita pri analýze dát:** Môžeme si pozrieť rôzne úrovne klastrovania a vybrať si počet klastrov, ktorý najlepšie vyhovuje našim potrebám.
- **Vhodné pre malé dataset-y:** Funguje dobre, keď máme menší počet dátových bodov, pretože výpočty môžu byť náročné pre veľké množstvo dát.
- **Počítačová náročnosť:** Pri veľkých datasetoch sa hierarchické klastrovanie stáva veľmi pomalým, pretože algoritmus musí neustále počítať vzdialenosť medzi všetkými klastrami.
- **Citlivosť na šum a odľahlé body:** Ak sú v dátach odľahlé body (outliery), môžu negatívne ovplyvniť výsledky klastrovania.

## Výsledky agglomeratívneho klastrovania



## Dendrogram - hierarchické klastrovanie

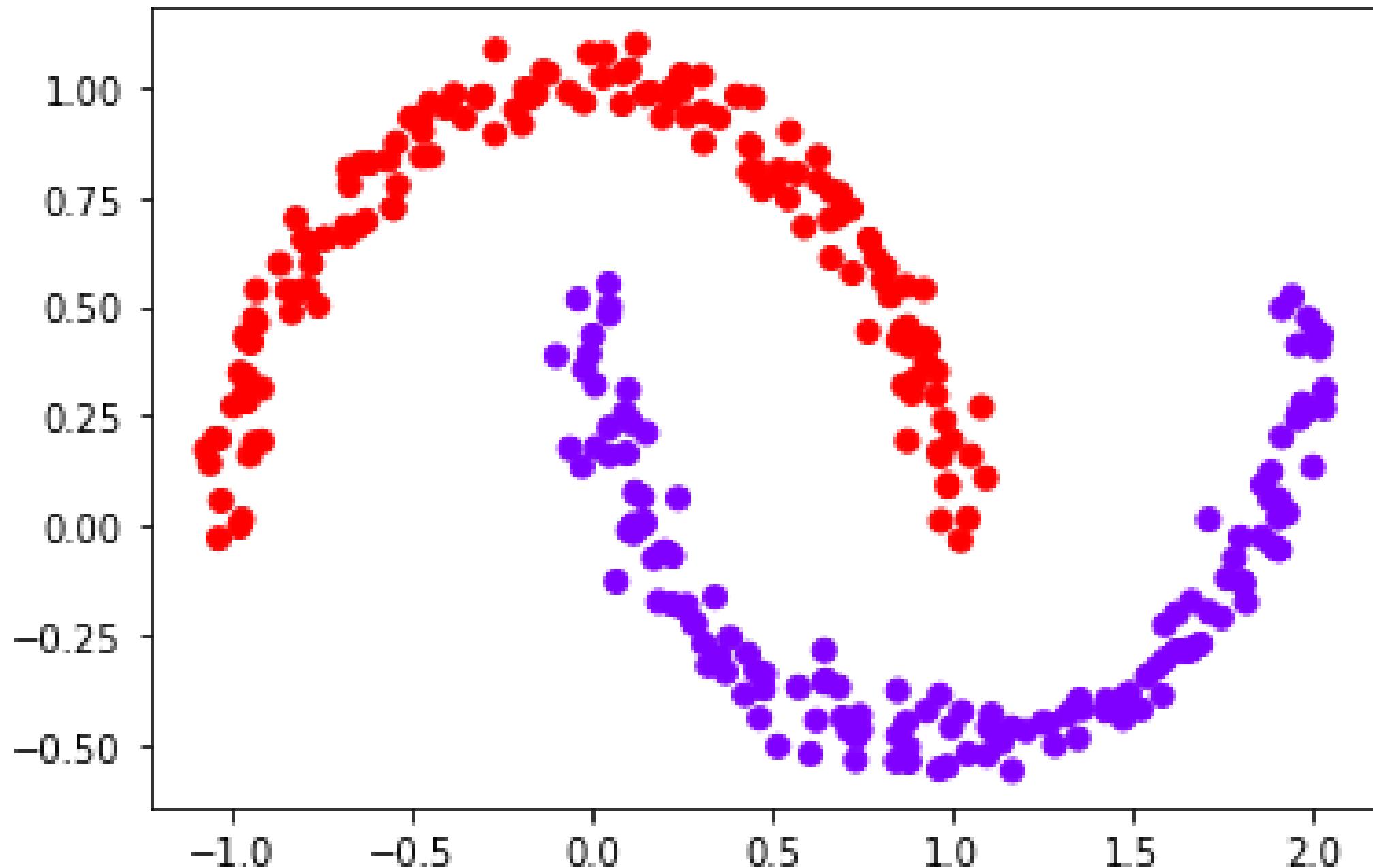


# DBSCAN



- **Density-Based (založené na hustote):** funguje na základe hustoty bodov. To znamená, že **klastre sa tvoria tam**, kde je **dostatok bodov blízko pri sebe (vysoká hustota)**. Ak sú body od seba príliš ďaleko, nepatria do rovnakého klastru.
- **Spatial Clustering (priestorové klastrovanie):** Algoritmus **hľadá klastre v priestore**, v ktorom sú **uložené dáta**. To znamená, že sa pozera na to, kde sú body v priestore, **a snaží sa ich rozdeliť** na **skupiny na základe** toho, kde sa **nachádzajú** (ich **súradnice alebo pozície**).
- **of Applications with Noise (aplikácie so šumom):** je špeciálny tým, že **dokáže ignorovať** šum (anglicky noise), čo znamená **body**, ktoré **nezapadajú do žiadneho klastru**. Tieto body sú označené ako "šumové" a sú považované za izolované alebo odľahlé dáta, ktoré sa nedajú priradiť k žiadnej skupine.

## Výsledky DBSCAN klastrovania

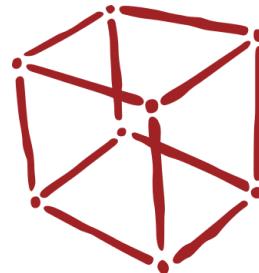




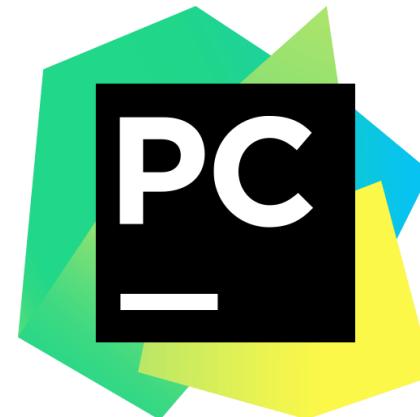
Klastrovací Algoritmus	Príklady použitia	Kedy sa hodí?
Hierarchické klastrovanie	<ul style="list-style-type: none"> <li>- Analýza genetických údajov (napr. podobnosť DNA sekvencii)</li> <li>- Klastrovanie dokumentov podľa obsahu</li> <li>- Skupinovanie obrazov v medicíne (napr. röntgenové snímky)</li> <li>- Segmentácia zákazníkov na základe podobnosti správania</li> <li>- Klastrovanie druhov zvierat podľa evolučných vzťahov</li> </ul>	<ul style="list-style-type: none"> <li>- Keď chceme vidieť hierarchickú štruktúru dát (ako sa malé skupiny zhlučujú do väčších)</li> <li>- Keď počet klastrov nie je dopredu známy</li> <li>- Pre menšie a stredne veľké dataset-y</li> <li>- Keď chceme analyzovať vzťahy medzi skupinami na rôznych úrovniach</li> <li>- Keď potrebujeme vizualizovať spojenia medzi jednotlivými klastrami pomocou dendrogramu</li> </ul>
DBSCAN	<ul style="list-style-type: none"> <li>- Klastrovanie dát s nepravidelnými tvarmi (napr. hviezdné galaxie v astronómii)</li> <li>- Detekcia anomalií v bankových transakciách (podvodné transakcie)</li> <li>- Skupinovanie miest na mape podľa blízkosti (napr. geografické dátá)</li> <li>- Rozpoznávanie pohybových vzorcov z GPS dát (sledovanie dopravných tokov)</li> <li>- Identifikácia hlavných tém v textoch, kde je šum (napr. analýza komentárov na sociálnych sietach)</li> </ul>	<ul style="list-style-type: none"> <li>- Keď máte dátá s nepravidelnými tvarmi klastrov (napr. kruhové alebo zakrivené skupiny)</li> <li>- Keď sú v dátach outliers (šumové body), ktoré chcete ignorovať</li> <li>- Keď nemáte predstavu o počte klastrov</li> <li>- Pre veľké dataset-y, kde potrebujete efektívne nájsť husté oblasti dát</li> <li>- Keď potrebujete algoritmus, ktorý dokáže ignorovať odľahlé body</li> </ul>
K-Means klastrovanie	<ul style="list-style-type: none"> <li>- Segmentácia zákazníkov na základe nákupného správania</li> <li>- Skupinovanie produktov podľa vlastností (napr. cenové skupiny)</li> <li>- Klastrovanie obrázkov na základe farieb (zjednodušenie obrazu do menšieho počtu farieb)</li> <li>- Skupinovanie oblastí na mape podľa priemernej teploty (napr. v klimatológii)</li> <li>- Rozdelenie užívateľov podľa správania na webovej stránke (webová analytika)</li> </ul>	<ul style="list-style-type: none"> <li>- Keď máte predbežnú predstavu o počte klastrov</li> <li>- Keď klastre majú približne guľovitý tvar</li> <li>- Pre rýchle a efektívne klastrovanie veľkých datasetov</li> <li>- Keď je dôležité, aby klastrovanie bolo jednoduché a interpretovateľné</li> <li>- Pre dátá, ktoré nemajú výrazné outliers alebo šumové body</li> </ul>

# Aké IDE mám použiť?

110



NetBeans



*wxPython*



Visual Studio



# Chceme úplne všetko!

---





Flexibility



Code  
Completion





```
print("Sorry, we are down for maintenance")
print("We'll be back shortly")
```

Individual Edition is now

# ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

### Anaconda Distribution

[Download !\[\]\(b1bbf497c6223285237576cdc19879f1\_img.jpg\)](#)

For Windows  
Python 3.9 • 64-Bit Graphical Installer • 510 MB

Get Additional Installers





## Open Source

Access the open-source software you need for projects in any field, from data visualization to robotics.



## User-friendly

With our intuitive platform, you can easily search and install packages and create, load, and switch between environments.



## Trusted

Our securely hosted packages and artifacts are methodically tested and regularly updated.

 ANACONDA NAVIGATOR[Sign in to Anaconda Cloud](#) Home

Applications on

base (root)

Channels

Refresh

 Environments Learning Community

## Documentation

## Developer Blog



Icon	Name	Version	Description	Action
	CMD.exe Prompt	0.1.1	Run a cmd.exe terminal with your current environment from Navigator activated	<a href="#">Launch</a>
	JupyterLab	1.2.6	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	<a href="#">Launch</a>
	Notebook	6.0.3	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	<a href="#">Launch</a>
	Powershell Prompt	0.0.1	Run a Powershell terminal with your current environment from Navigator activated	<a href="#">Launch</a>
	Qt Console	4.6.0	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	<a href="#">Launch</a>
	Spyder	4.0.1	Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features	<a href="#">Launch</a>
	VS Code	1.52.1	Streamlined code editor with support for development operations like debugging, task running and version control.	<a href="#">Launch</a>
	Glueviz	0.15.2	Multidimensional data visualization across files. Explore relationships within and among related datasets.	<a href="#">Install</a>
	Orange 3	3.23.1	Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.	<a href="#">Install</a>
	RStudio	1.1.456	A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.	<a href="#">Install</a>

Microsoft Store

python

Filtre

# "python"

Všetky oddelenia Aplikácie Hry

Applikácie Hranie

Nainštalované Bezplatné Bezplatné Bezplatné Bezplatné 0,99 € Bezplatné

Python 3.9 Python 3.8 Python 3.10 Python 3.7 Learn Django and Python by GoLearningBus WiBit.Net :: Programming in Python

Aplikácie • Developer tools Aplikácie • Developer tools Aplikácie • Developer tools Aplikácie • Books & reference Aplikácie • Books & reference

5★ 4

Python 3.9 Python 3.8 Python 3.10 Python 3.7 Learn Django and Python by GoLearningBus WiBit.Net :: Programming in Python

Aplikácie • Books & reference Aplikácie • Books & reference

Nainštalované Bezplatné Bezplatné Bezplatné Bezplatné Bezplatné

Python DATA SCIENCE Python Learn Python Introduction to Python Programming by... Python Playground Python Programming Manual For Absolute...

Aplikácie • Education Aplikácie • Education Aplikácie • Books & reference Aplikácie • Books & reference Aplikácie • Developer tools Aplikácie • Utilities & tools

Data Science with Python Python Programs Learn Python Introduction to Python Programming by... Python Playground Python Programming Manual For Absolute...

Aplikácie • Education Aplikácie • Education Aplikácie • Books & reference Aplikácie • Books & reference Aplikácie • Developer tools Aplikácie • Utilities & tools

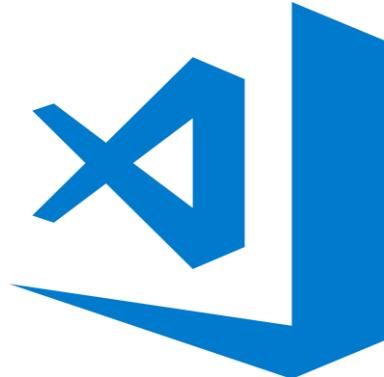
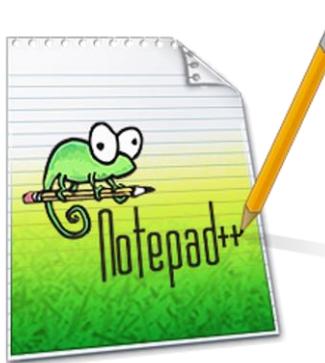
Py Tutorial

计算二级 Python 考试库

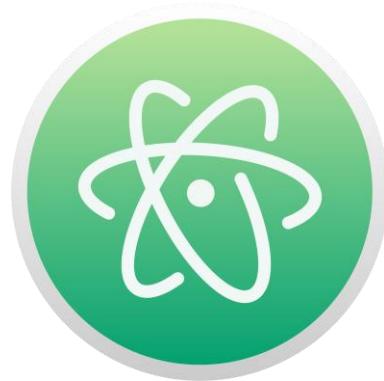
计算二级 Python 考试库

计算二级 Python 考试库

# Aký editor mám použiť?



```
:::  
iLE880j. :jD888880j:  
.LGitE888D.f8GjjjL8888E;  
iE :888Et. .G888.  
;i E888, ,8888,  
D888, :8888:  
D888, :8888:  
D888, :8888:  
D888, :8888:  
888W, :8888:  
W88W, :8888:  
W88W: :8888:  
DGGD: :8888:  
:8888:  
:W888:  
:8888:  
E888i  
tW88D
```



IDE ≠ editor



# Dashboard

Welcome, [itacademysk](#)**CPU Usage:** 0% used – 0.00s of 100s. Resets in 1 hour, 56 minutes [More Info](#)**File storage:** 0% full – 100.0 KB of your 512.0 MB quota [More Info](#)[Upgrade Account](#)

## Recent Consoles

[+ 5 -](#)*You have no recent consoles.*

## New console:

[\\$ Bash](#)[>>> Python ▾](#)[More...](#)

### Version

[2.7](#)[3.7](#)[3.8](#)

## Recent Files

[+ 5 -](#)

/home/itacademysk/.bashrc  
/home/itacademysk/.gitconfig  
/home/itacademysk/.profile  
/home/itacademysk/.pythonstartup.py  
/home/itacademysk/.vimrc

[+ Open another file](#)[Browse files](#)

## Recent Notebooks

[+ 5 -](#)

Your account does not support Jupyter Notebooks. [Upgrade your account](#) to get access!

## All Web apps

*You don't have any web apps.*[Open Web tab](#)

[MySQL](#)[Postgres](#)

## Initialize MySQL

Let's get started! The first thing to do is to initialize a MySQL server:

Enter a new password in the form below, and note it down: you'll need it to access the databases once you've created them. You will only need to do this once.

**New password:**

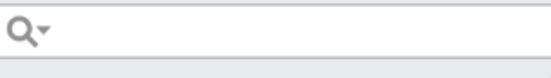
A password input field with a small circular icon containing a question mark in the top right corner.

**Confirm password:**

A password input field with a small circular icon containing a question mark in the top right corner.

**Initialize MySQL**

This should be different to your main PythonAnywhere password, because it is likely to appear in plain text in any web applications you write.



## Editor > Live Templates

By default expand with **Tab** ▼

### Python

- `compd` (Dict comprehension)
- `compdi` (Dict comprehension with 'if')
- `compg` (Generator comprehension)
- `compgi` (Generator comprehension with 'if')
- `compl` (List comprehension)
- `compli` (List comprehension with 'if')
- `comps` (Set comprehension)
- `compsi` (Set comprehension with 'if')
- `iter` (Iterate (for ... in ...))
- `itere` (Iterate (for ... in enumerate))
- `main` (if `_name_ == '__main__'`)
- `prop` (Property getter)
- `props` (Property getter/setter)
- `propsd` (Property getter/setter/deleter)
- `super` ('super(...)' call)

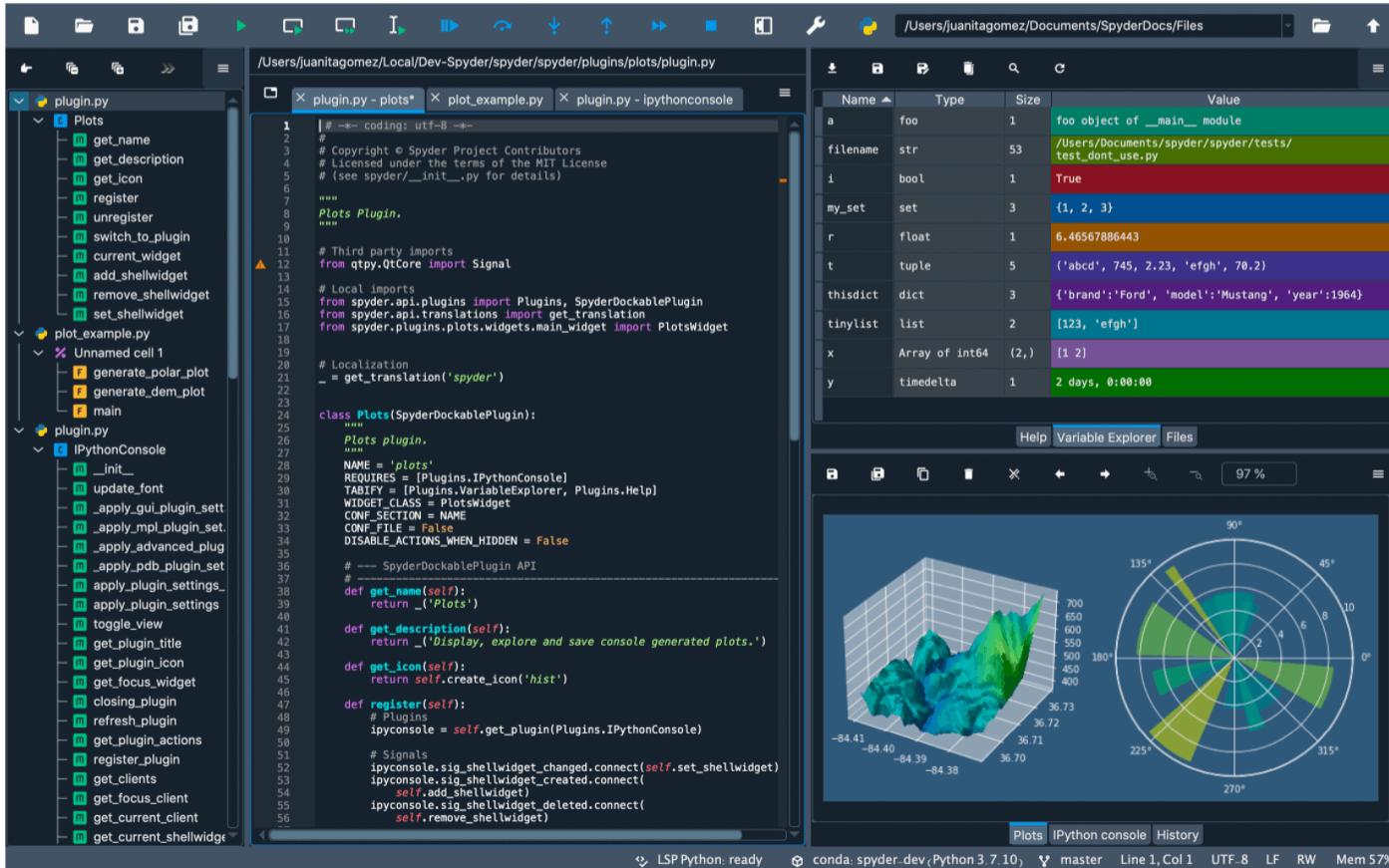
### R

### React

No live templates are selected



# The Scientific Python Development Environment



The screenshot displays the Spyder IDE interface with the following components:

- Code Editor:** Shows the file `/Users/juanitagomez/Local/Dev-Spyder/spyder/spyder/plugins/plots/plugin.py`. The code defines a `Plots` plugin for Spyder. It includes imports for `QtCore`, `Signal`, and various Spyder API modules. The class `Plots` inherits from `SpyderDockablePlugin` and implements methods like `get_name`, `get_description`, `get_icon`, `register`, etc.
- Variable Explorer:** A table showing the current state of variables:

Name	Type	Size	Value
a	foo	1	foo object of __main__ module
filename	str	53	/Users/Documents/spyder/spyder/tests/test_dont_use.py
i	bool	1	True
my_set	set	3	{1, 2, 3}
r	float	1	6.46567886443
t	tuple	5	('abcd', 745, 2.23, 'efgh', 78.2)
thisdict	dict	3	{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
tinylist	list	2	[123, 'efgh']
x	Array of int64	(2,)	[1 2]
y	timedelta	1	2 days, 0:00:00
- Plots:** Two plots are shown in the bottom right:
  - A 3D surface plot of a function over a grid.
  - A polar plot with radial and angular axes.
- Status Bar:** Shows "LSP Python: ready", "conda: spyder.dev:Python 3.7.10", and memory usage "Mem 57%".

VERSION

Spyder 5

Search ...



WELCOME

QUICKSTART

INSTALLATION GUIDE

▶ INTRO VIDEOS

▶ PANES IN DEPTH

▶ SPYDER PLUGINS

▶ TROUBLESHOOTING

▶ WORKSHOPS

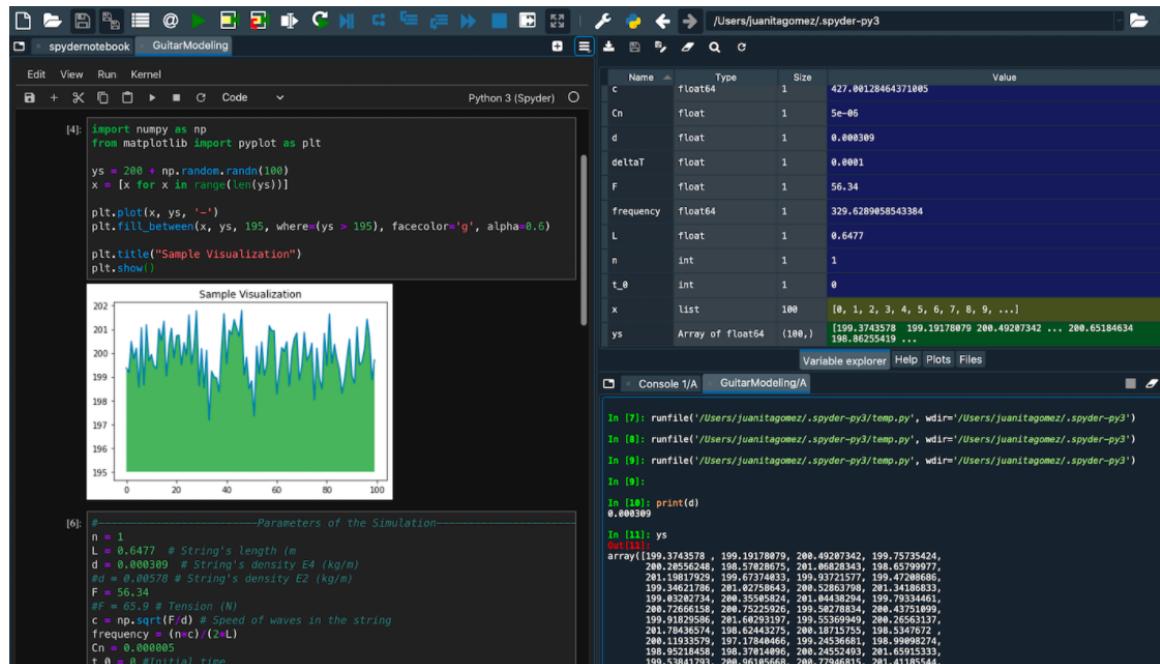
FAQ

# Spyder Notebook

## Warning

Currently, this plugin is still being ported to Spyder 5, and will likely not yet work or experience serious issues on this version of Spyder. A compatible version is expected soon. Thanks for your patience!

**Spyder-notebook** is a plugin that allows you to open, edit and interact with Jupyter Notebooks right inside Spyder.



conda install spyder-notebook -c spyder-ide

[Edit this page](#)[On this page](#)[Installing the Notebook](#)[Using the Notebook](#)[Connecting an IPython Console](#)[Additional Options](#)[OPEN CHAT](#)

C:\Users\miros

▼ ↑

Python\_I\_Za...

Edit View Run Kernel

Python 3 (Spyder)

Name Type Size Value

## Kurz Python - 1. deň

Miroslav Reiter | miroslav.reiter@it-academy.sk | <https://www.linkedin.com/in/miroslav-reiter/>

Kurz Python | <https://www.it-academy.sk/kurz/python/> | <https://github.com/miroslav-reiter>

## Komentáre, kódovanie, tlač a docstringy

```
[1]: # -*- coding: utf-8 -*-
# Toto je komentár (jednoriadkový)
"""Toto je docstring (document string)"""

# Pozor na nespravne zalomovanie riadku (Enter)
# SyntaxError: EOL while scanning string Literal
# Emotikony https://emojipedia.org/
print("Python je fajnovy jazyk!")
print(__doc__)
print("🎲 🎲 🎲")
print("Co bolo skorej? --> ", min(['\N{CHICKEN}', '\N{EGG}']))
```

Python je fajnovy jazyk!
Toto je docstring (document string)
🎲 🎲 🎲
Co bolo skorej? --> 🎲

## Premenné a typy

•••

Milujem Python
Milujem Python
Milujem Python

Nazov produktu je: Hypoteka pre mladych 2021
Splatka vasej hypoteky je: 600 Eur
Uroková sadzba je: 1.5 % p.a.

Variable explorer Help Plots Files

Console 1/A

Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]:

Editor Notebook

LSP Python: ready Kite: ready (no index) conda: base (Python 3.8.5) main [86] Line 8, Col 1 UTF-8 CRLF RW Mem 47% CPU 17% 09:26

## Útržky kódu

Filtrovať útržky kódu

Adding form fields →

Camera Capture →

Cross-output communication →

display.Javascript to execute Jav... →

Downloading files or importing da... →

Adding form fields

Vložit'

Forms example

Forms support multiple types of fields with type checking including sliders, date pickers, input fields, dropdown menus, and dropdown menus that allow input.

```
#@title Example form fields
#@markdown Forms support many types of fields with type checking including sliders, date pickers, input fields, dropdown menus, and dropdown menus that allow input.

no_type_checking = '' #@param
string_type = 'example' #@param
slider_value = 142 #@param {type: "number", min: 0, max: 200}
number = 102 #@param {type: "number", min: 0, max: 200}
date = '2010-11-05' #@param {type: "date", min: "2010-01-01", max: "2020-12-31"}
pick_me = "monday" #@param [ 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday' ]
select_or_input = "apples" #@param {type: "select", options: [ "apples", "oranges", "bananas" ]}
```

[Zobrazit zdrojový zápisník](#)

+ Kód + Text Kopírovať na Disk

```
tiene = True
# r g b, c m y k, w
farby_vlastne = ["black","pink", "b", "#CCCC00"]

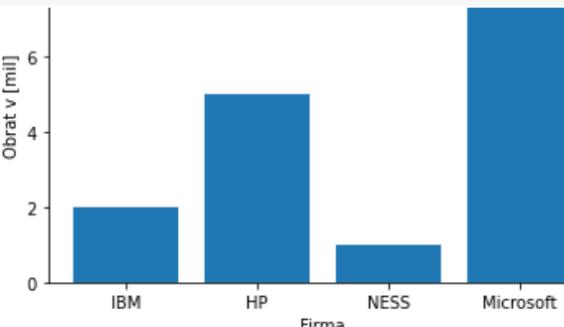
print(type(y))
print(y)

plt.pie(y, labels = menovky, startangle = 0, explode = vykrojenie, shadow = tiene, colors = farby_vlastne)
plt.legend(title = "Produkty ABC s.r.o.")
plt.title("Analyza predaja produktov za Q1-2022")
plt.show()

x1 = np.array(["IBM", "HP", "NESS", "Microsoft"])
y1 = np.array([2, 5, 1, 9])

plt.bar(x1, y1)
plt.title("Porovnanie IT firiem 2022")
plt.xlabel("Firma")
plt.ylabel("Obrat v [mil]")
plt.show()

nahoda = np.random.normal(100, 10, 200)
print(nahoda)
plt.hist(nahoda)
plt.show()
```



```
[ 79.76069196  99.4155264  114.29926387 101.33767141  88.49106384
 111.70288892  91.32702578 102.53587004 108.38846479 114.34889501
 98.79114202 117.40488367 89.26174251  94.12100639 101.96805716]
```



For Teams      Download

# Code Faster. Stay in Flow.

Kite adds AI powered code completions to your code editor, giving developers superpowers.



Download for Free

```
1 import os
2 import sys
3
4 def count_py_files_in_repos(dirname):
5     if os.path.exists(os.path.join(dirname, '.git')):
6         count = 0
7         for root, dirs, files in os.walk(dirname):
8             count += len([f for f in files if f.endswith('.py')])
9         print('{} has {} Python files'.format(dirname, count))
10        for name in os.listdir(di)
```



dirname  
dirs  
dict

kite.com

```
1 import os
2 import sys
3
4 def count_py_files_in_repos(dirname):
5     i|
```



C:\Users\miros\AppData\Local\Temp\kite\_tutorial.py

temp.py kite\_tutorial.py

```

1 # Welcome to...
2 #
3 #      `hmy+.      ://:
4 #      .mMMMMMNho: ` NMMm
5 #      :NMMMMMMMMMdS/.` NMMm      :ss:
6 #      +NMMMMMMMMMMMMmy+ NMMm      -MMM-   ---
7 #      `oMMMMMMMMMMMMMMMo NMMm      /ss/   :MM+
8 #      `yMMMMMMMMNshmNMMMN` NMMm      /MM+
9 #      .dMMMMMMMMm/hmhssydmMM+ NMMm      `/yhy. shhy ohmMMMhhhh. ./ydmmdho-
10 #     omMMMMMd/mMMMMMhsosy` NMMm      .omMMmo. mMMN odmMMMdsss. omMNsoshNMNy
11 #     .+dMMMy/mMMMMMMMMMd- NMMm-yNMMy/` mMMN /MM+ sMMN: `:NMMy
12 #     `ymo/NMMMMMMMMMd NMMmNMNMMN` mMMN :MM+ MMNdddNNMMN
13 #     `hMMMMMMMMMM: NMMm+mMMNs. mMMN :MM+ MMN///////////
14 #     `:yNMMMMMMMMh NMMm `/dMMNy- mMMN :MM+ ` sMMNo` `:
15 #     .+mMMMMMd- NMMm `/dMMNy- mMMN .MMNddNN/ +NMNdhydNNMs
16 #     `:yMMMy yhs      `/hhh shhs :ymmmdho: `/sdmmmmhs:`
17 #     `om.
18
19 """
20 Kite is your Python programming copilot. Kite will try to show you the
21 right information at the right time as you code to prevent you from context
22 switching out of your current line of thought.
23
24 This tutorial will teach you how to use all of Kite's core features. You
25 should be able to learn everything in 5 minutes.
26
27 If you get stuck at any point, please visit https://help.kite.com/ or file
28 an issue at https://github.com/kiteco/issue-tracker.
29 """
30
31
32
33
34 """ PART 0: BEFORE WE START =====
35
36 Spyder will by default try to start the Kite backend when the editor first
37 starts. You can change this behavior by opening settings, clicking on
38 "Completion and linting", "Advanced", and then changing Kite's "Start Kite
39 Engine on editor startup" setting.
40
41 Look for the Kite indicator in the bottom left corner of Spyder's status
42 bar – It will tell you if Kite is ready and working. If the indicator reads
43 "not running", then you'll have to start the Kite Engine manually before
44 proceeding with the rest of this tutorial.
45 """
46
47
48
49

```

↓ ☰ 🔍 C

Name	Type	Size	Value
a	int	1	5
b	str	1	Karol

Variable explorer Help Plots Files

Console 1/A

Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/miros/.spyder-py3/temp.py', wdir='C:/Users/miros/.spyder-py3')

In [2]: runfile('C:/Users/miros/.spyder-py3/temp.py', wdir='C:/Users/miros/.spyder-py3')  
5  
Karol

In [3]:

```
C:\Windows\System32\cmd.exe - pip install matlib
Microsoft Windows [Version 10.0.16299.785]
(c) 2017 Microsoft Corporation. Všetky práva vyhradené.

C:\Windows\System32>cd C:\Program Files (x86)\Python37-32\Scripts

C:\Program Files (x86)\Python37-32\Scripts>pip instal xlwt
ERROR: unknown command "instal" - maybe you meant "install"

C:\Program Files (x86)\Python37-32\Scripts>pip install xlwt
Collecting xlwt
  Downloading https://files.pythonhosted.org/packages/44/48/def306413b25c3d01753603b1a222a011b8621aed27cd7f89cbc27e6b0f4/xlwt-1.3.0-py2.py3-none-any.whl (99kB)
    100% |██████████| 102kB 826kB/s
Installing collected packages: xlwt
Could not install packages due to an EnvironmentError: [WinError 5] Access is denied: 'c:\\\\program files (x86)\\\\python37-32\\\\Lib\\\\site-packages\\\\xlwt'
Consider using the `--user` option or check the permissions.

You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Program Files (x86)\Python37-32\Scripts>pip install matlib
Collecting matlib
```

PIP a easy install



**Seems like I've  
installed wrong version  
of Python...**

sys Variables		String Methods		Datetime Methods					
argv	Command line args	capitalize() *	lstrip()	today()	fromordinal(ordinal)				
builtin_module_names	Linked C modules	center(width)	partition(sep)	now(timezoneinfo)	combine(date, time)				
byteorder	Native byte order	count(sub, start, end)	replace(old, new)	utcnow()	strftime(date, format)				
check_interval	Signal check frequency	decode()	rfind(sub, start ,end)	fromtimestamp(timestamp)	utcfromtimestamp(timestamp)				
exec_prefix	Root directory	encode()	rindex(sub, start, end)						
executable	Name of executable	endswith(sub)	rjust(width)						
exitfunc	Exit function name	expandtabs()	rpartition(sep)						
modules	Loaded modules	find(sub, start, end)	rsplit(sep)						
path	Search path	index(sub, start, end)	rstrip()						
platform	Current platform	isalnum() *	split(sep)						
stdin, stdout, stderr	File objects for I/O	isalpha() *	splittlines()						
version_info	Python version info	isdigit() *	startswith(sub)						
winver	Version number	islower() *	strip()						
<b>sys.argv</b> for \$ python foo.py bar -c qux --h		isspace() *	swapcase() *						
sys.argv[0]	foo.py	istitle() *	title() *						
sys.argv[1]	bar	isupper() *	translate(table)						
sys.argv[2]	-c	join()	upper() *						
sys.argv[3]	qux	ljust(width)	zfill(width)						
sys.argv[4]	--h	lower() *							
<b>os Variables</b>		<b>Note</b> Methods marked * are locale dependant for 8-bit strings.							
altsep	Alternative sep								
curdir	Current dir string								
defpath	Default search path								
devnull	Path of null device								
extsep	Extension separator								
linesep	Line separator								
name	Name of OS								
pardir	Parent dir string								
pathsep	Patch separator								
sep	Path separator								
<b>Note</b> Registered OS names: "posix", "nt", "mac", "os2", "ce", "java", "riscos"									
<b>Class Special Methods</b>									
__new__(cls)	__lt__(self, other)								
__init__(self, args)	__le__(self, other)								
__del__(self)	__gt__(self, other)								
__repr__(self)	__ge__(self, other)								
__str__(self)	__eq__(self, other)								
__cmp__(self, other)	__ne__(self, other)								
__index__(self)	__nonzero__(self)								
__hash__(self)									
__getattr__(self, name)									
__getattribute__(self, name)									
__setattr__(self, name, attr)									
__delattr__(self, name)									
__call__(self, args, kwargs)									
<b>Indexes and Slices (of a=[0,1,2,3,4,5])</b>									
	len(a)	6							
	a[0]	0							
	a[5]	5							
	a[-1]	5							
	a[-2]	4							
	a[1:]	[1,2,3,4,5]							
	a[:5]	[0,1,2,3,4]							
	a[:-2]	[0,1,2,3]							
	a[1:3]	[1,2]							
	a[1:-1]	[1,2,3,4]							
	b=a[:]	Shallow copy of a							
<b>Time Methods</b>									
	replace()	utcoffset()							
	isoformat()	dst()							
	__str__( )	tzname()							
	strftime(format)								
<b>Date Formatting (strftime and strptime)</b>									
	%a	Abbreviated weekday (Sun)							
	%A	Weekday (Sunday)							
	%b	Abbreviated month name (Jan)							
	%B	Month name (January)							
	%c	Date and time							
	%d	Day (leading zeros) (01 to 31)							
	%H	24 hour (leading zeros) (00 to 23)							
	%I	12 hour (leading zeros) (01 to 12)							
	%j	Day of year (001 to 366)							
	%m	Month (01 to 12)							
	%M	Minute (00 to 59)							
	%p	AM or PM							
	%S	Second (00 to 61 <sup>4</sup> )							
	%U	Week number <sup>1</sup> (00 to 53)							
	%w	Weekday <sup>2</sup> (0 to 6)							
	%W	Week number <sup>3</sup> (00 to 53)							
	%x	Date							
	%X	Time							
	%y	Year without century (00 to 99)							
	%Y	Year (2008)							
	%Z	Time zone (GMT)							
	%%	A literal "%" character (%)							
1. Sunday as start of week. All days in a new year preceding the first Sunday are considered to be in week 0.									
2. 0 is Sunday, 6 is Saturday.									
3. Monday as start of week. All days in a new year preceding the first Monday are considered to be in week 0.									
4. This is not a mistake. Range takes account of leap and double-leap seconds.									
Available free from <a href="http://AddedBytes.com">AddedBytes.com</a>									

PC Settings X

Search

> Appearance & Behavior

Keymap

Editor

- > General
- Font
- > Color Scheme
- > Code Style
- Inspections
- File and Code Templates
- File Encodings
- Live Templates**
- File Types
- > Emmet
- Images
- Intentions
- Language Injections
- Spelling
- TextMate Bundles
- TODO

Plugins

> Version Control

> Project: test1

> Build, Execution, Deployment

> Languages & Frameworks

> Tools

Editor > Live Templates

By default expand with **Tab** ▼

>  **Python**

- compd (Dict comprehension)
- compdi (Dict comprehension with 'if')
- compg (Generator comprehension)
- compgi (Generator comprehension with 'if')
- compl (List comprehension)
- compli (List comprehension with 'if')
- comps (Set comprehension)
- compsi (Set comprehension with 'if')
- iter (Iterate (for ... in ...))
- itere (Iterate (for ... in enumerate))
- main (if \_\_name\_\_ == '\_\_main\_\_')
- prop (Property getter)
- props (Property getter/setter)
- propsd (Property getter/setter/deleter)
- super ('super(...)' call)

>  **R**

>  **React**

No live templates are selected

OK Cancel Apply

# Dôležité Zdroje

## Slovensko a Česko

- Albatrosmedia
- Kopp
- Grada
- Wolters Kluwer
- BEN
- Veda

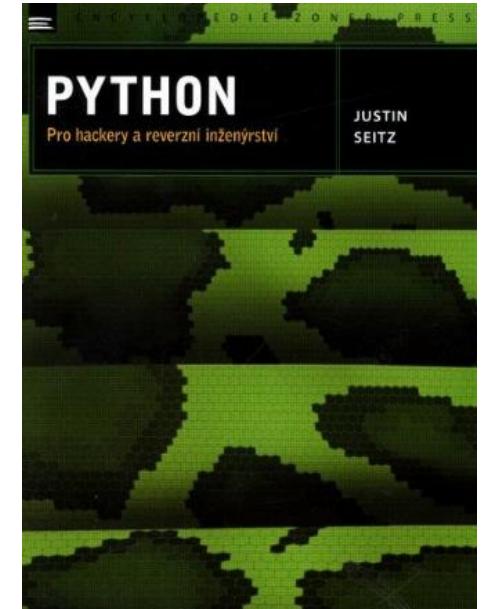
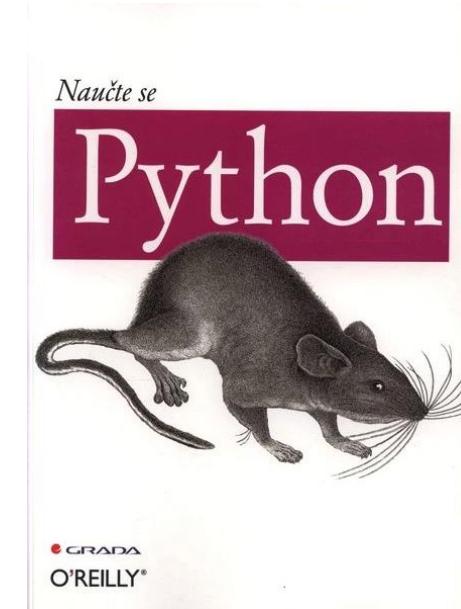
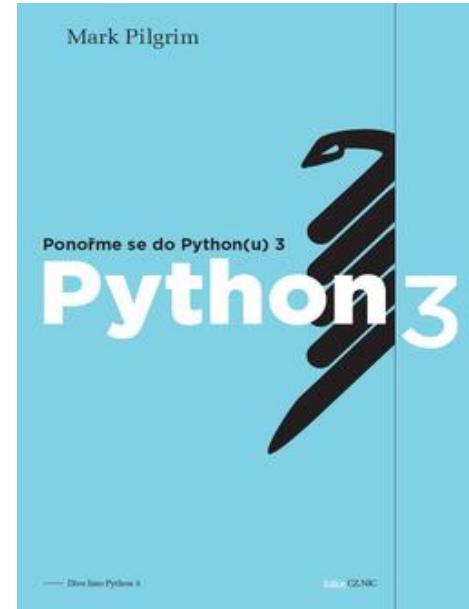
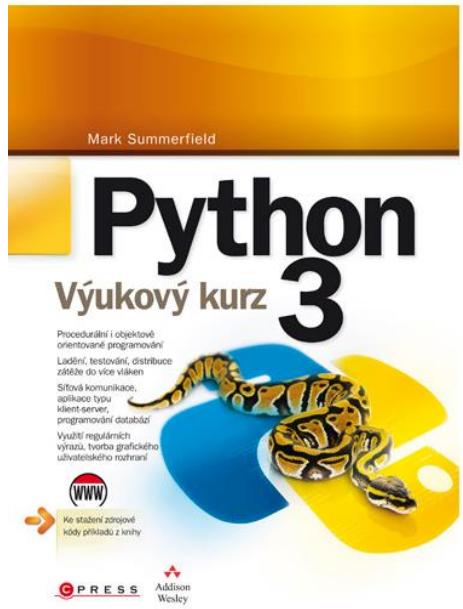
## Zahraničie

- O'Reilly
- Manning
- Packt
- Apress
- Wiley
- No Starch Press

## YouTube Tutoriály

- [IT Academy](#)

IT Academy



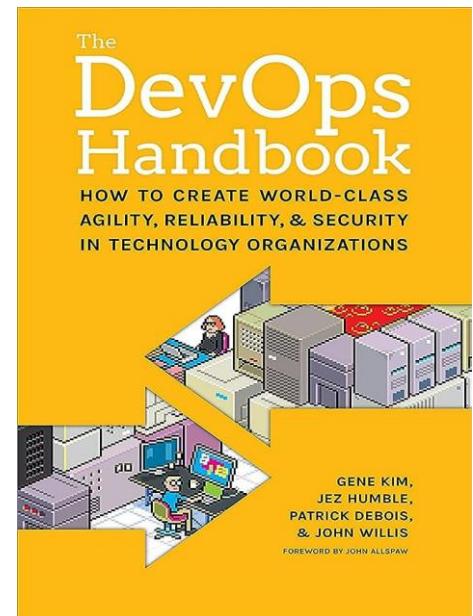
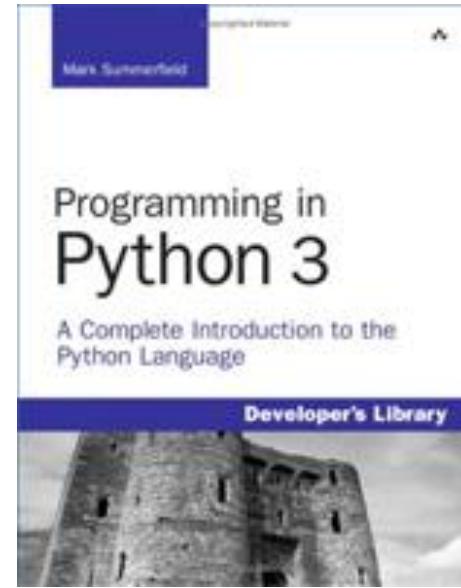
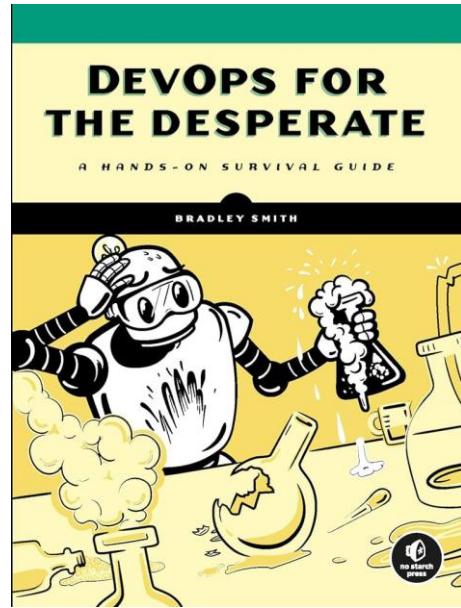
# Čo sa oplatí/neoplatí prečítať SK/CZ?



# Effective DevOps

BUILDING A CULTURE OF COLLABORATION,  
AFFINITY, AND TOOLING AT SCALE

Jennifer Davis & Ryn Daniels



# Čo sa oplatí/neoplatí prečítať EN?

# Čo Odporúčam si Pozriet’?

1. <https://docs.python.org/3/>
2. <https://realpython.com/tutorials/best-practices/>
3. <https://google.github.io/styleguide/pyguide.html>
4. <https://docs.python.org/3/>
5. <http://python2013.input.sk/19prednaska>
6. <https://realpython.com/python3-object-oriented-programming/>
7. <https://jeffknupp.com/blog/2014/06/18/improve-your-python-python-classes-and-object-oriented-programming/>
8. <https://overiq.com/python-101/inheritance-and-polymorphism-in-python/>
9. <https://www.javatpoint.com/python-oops-concepts>
10. <https://www.programiz.com/python-programming/object-oriented-programming>



# TOP Klávesové Skratky

# TOP Klávesové Skratky

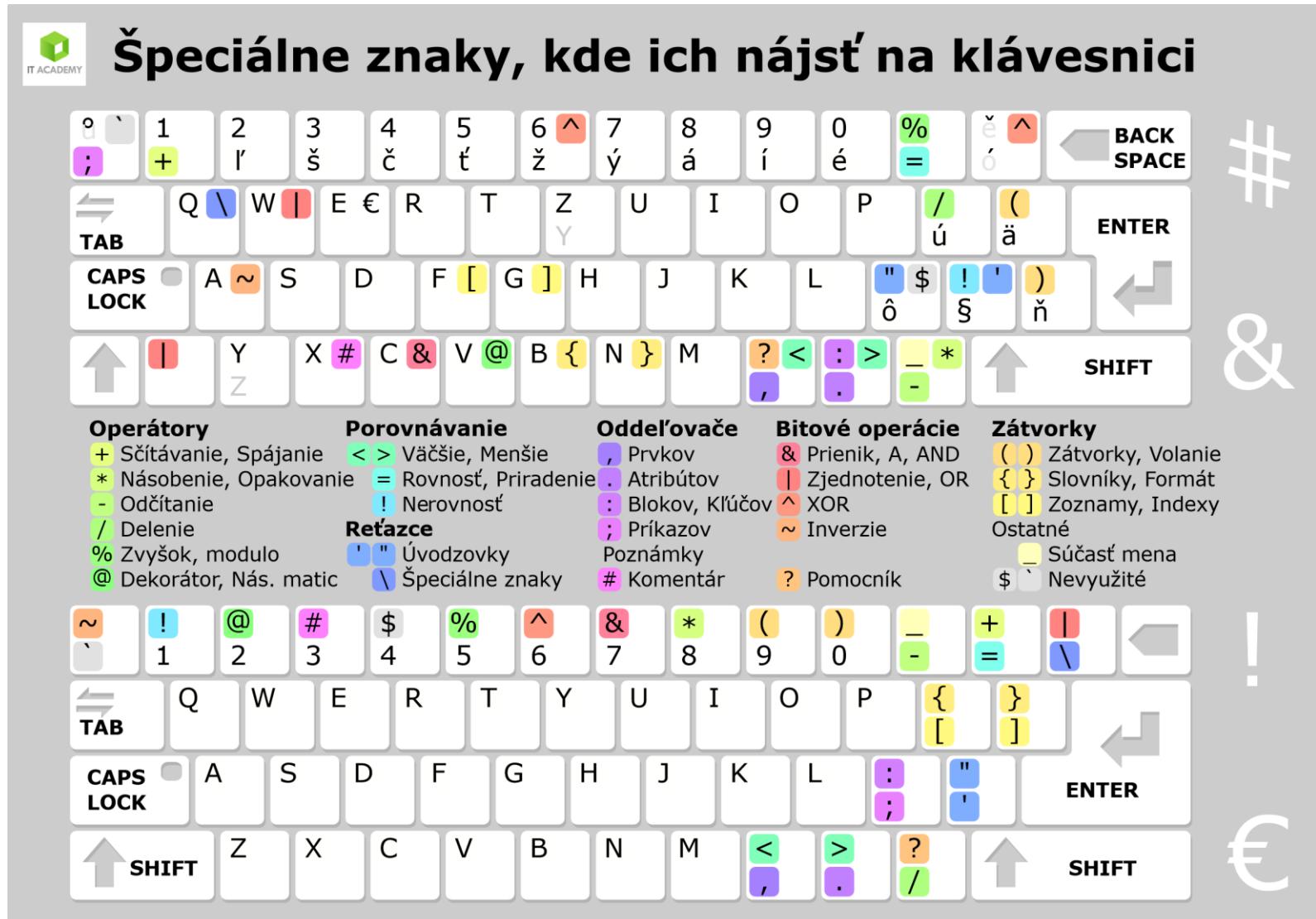
## Práca s IDE

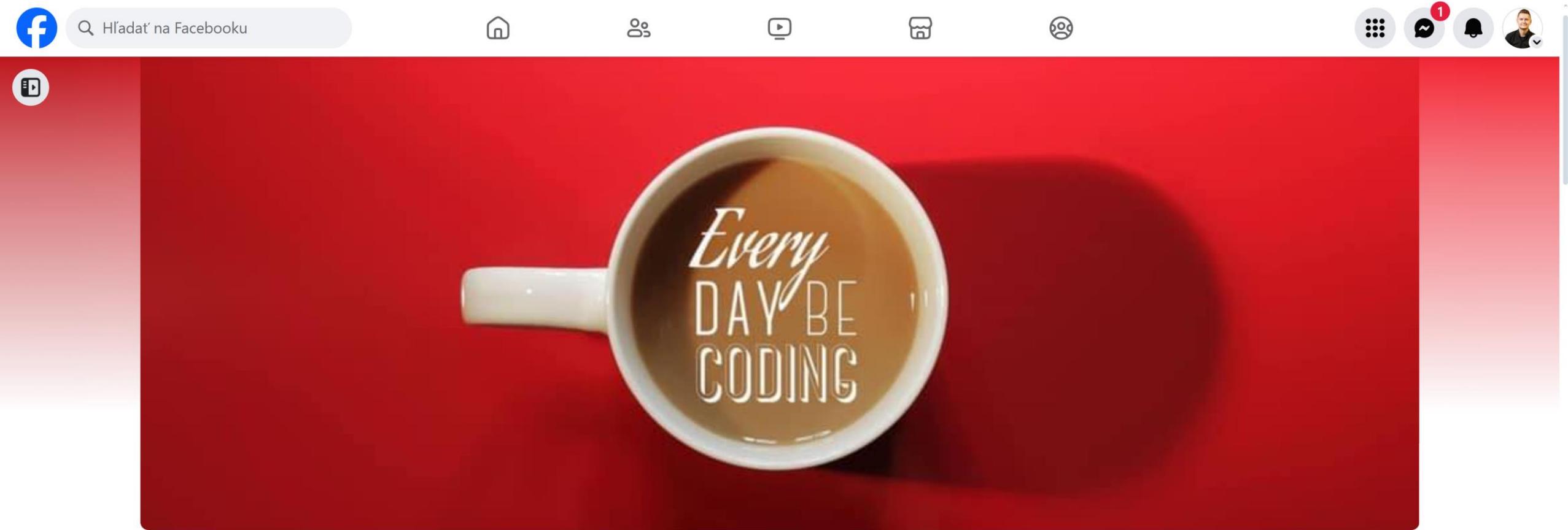
- Ctrl + D Delete zmaž riadok
- **Ctrl + Space** Asistent kódu
- **Ctrl + /** Komentáre
- Ctrl + A Označ všetko
- **Alt + /** Dokonči slovo
- Ctrl + F Hľadanie a náhrady
- Ctrl + Shift + F Kompakt režim
- Ctrl + Shift + S Ulož všetko

## Práca s browserom

- Ctrl + T Vytvor nový tab
- Ctrl + W Zatvor aktuálny tab
- Ctrl + Shift + W Zatvor všetky taby
- **Ctrl + Shift + T** Otvor posledný tab
- Ctrl + Shift + J/F12 Web console
- **F11** Fullscreen

# Efektívne Používanie Klávesnice





## Vývojári

Verejná skupina · 7,4 tis. členov

+ Pozvať

↗ Zdieľať

👤 Člen ▾

▼

Diskusia

Ludia

Podujatia

Médiá

Súbory



...



Napíšte niečo...



Reel



Fotka/video



Anketa

### Informácie

Skupina softvérových vývojárov. Táto skupina by mala byť miestom, kde sa môžu českoslovení vývojári vzájomne spoznať, vyžiadať si konštrukívnu... [Zobrazit viac](#)

### 👤 Verejná

Členov skupiny a ich príspevky bude vidieť ktokoľvek.





Hľadať na Facebooku



```
52 </div>
53 </body>
54 <script type="text/javascript">
55 <!--
56 var currentImage = "bigImage1";
57 var pages = Math.ceil.photos.length / 9);
58 updatePages();
59 updateAllImages();
60 // document.getElementById('bigImage0').src = 'images/wieksza' + photos[page * 9];
61 // document.getElementById('bigImage0').style.display = '';
62 changePhotoDescription('1');
63
64
65
66
67
68
69
70
71
72
73
```

## Programátori

Verejná skupina · 11,1 tis. členov

+ Pozvat

Člen ▾



Diskusia

Vybrané

Ludia

Podujatia

Média

Súbory



Napíšte niečo...



Reel



Fotka/video



Anketa

### Informácie

Táto skupina slúži na dohadzovanie si kšeftíkov a pre hľadačov programátorov / vývojárov.

#### Verejná

Členov skupiny a ich príspevky bude vidieť ktokoľvek.

#### Viditeľná

Príspevky sú viditeľní iba členom skupiny.



# I am programmer



I have Life



I have  
stackoverflow



IT ACADEMY

Home

PUBLIC

Questions

**Tags**

Users

COLLECTIVES

Explore Collectives

FIND A JOB

Jobs

Companies

TEAMS

Create free Team

# Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

[Show all tag synonyms](#)

python

**python**

Python is a multi-paradigm, dynamically typed, multi-purpose programming language. It is designed to be quick to learn, understand, and...

1870168 questions 695 asked today, 6387 this week

**python-3.x**

USE ONLY IF YOUR QUESTION IS VERSION-SPECIFIC. For questions about Python programming that are specific to version 3+ of the language...

303562 questions 93 asked today, 836 this week

**python-2.7**

Python 2.7 is the last major version in the 2.x series, and is no longer maintained since January 1st 2020. Use the generic [python] tag on all Python...

94965 questions 24 asked this week, 106 this month

**python-requests**

USE ONLY FOR THE PYTHON REQUESTS LIBRARY. Requests is a full-featured Python HTTP library with an easy-to-use, logical API.

18697 questions 8 asked today, 57 this week

**python-imaging-library**

The Python Imaging Library (PIL) provides the Python language with a de-facto standard foundation for image work. PIL's API is lightweight but...

7883 questions 5 asked today, 38 this week

**wxpython**

wxPython is a Python wrapper for the cross-platform C++ GUI API wxWidgets.

7047 questions 7 asked this week, 14 this month

**ipython**

IPython is a feature-rich interactive shell for Python, and provides a kernel for frontends such as IPython Notebook and Jupyter Notebook.

6886 questions 5 asked this week, 26 this month

**python-3.6**

Version of the Python programming language released in December 2016. For issues specific to Python 3.6. Use more generic [python] and [python-3....

5602 questions 11 asked this week, 24 this month

**python-asyncio**

to be used for the asyncio Python package which provides mechanisms for writing single-threaded concurrent code. The asyncio package provides...

5492 questions 29 asked this week, 125 this month

**python-import**

For questions about importing modules in Python

5119 questions 11 asked this week, 47 this month

**python-multiprocessing**

multiprocessing is a package that supports spawning processes using an API similar to the threading module in python programming language.

4036 questions 12 asked this week, 46 this month

**python-3.7**

Version of the Python programming language released in June 27, 2018. For issues that are specific to Python 3.7. Use the more generic [python] and...

4034 questions 5 asked this week, 21 this month

[Popular](#) [Name](#) [New](#)

[PROGRAM](#)[SPONZORI](#)

2%

[O NÁS](#)[KÚP SI LÍSTOK](#)

# PYCON SK 2024

15.3.2024 - 17.3.2024 | Bratislava

23 : 12 : 54 : 57

days

hours

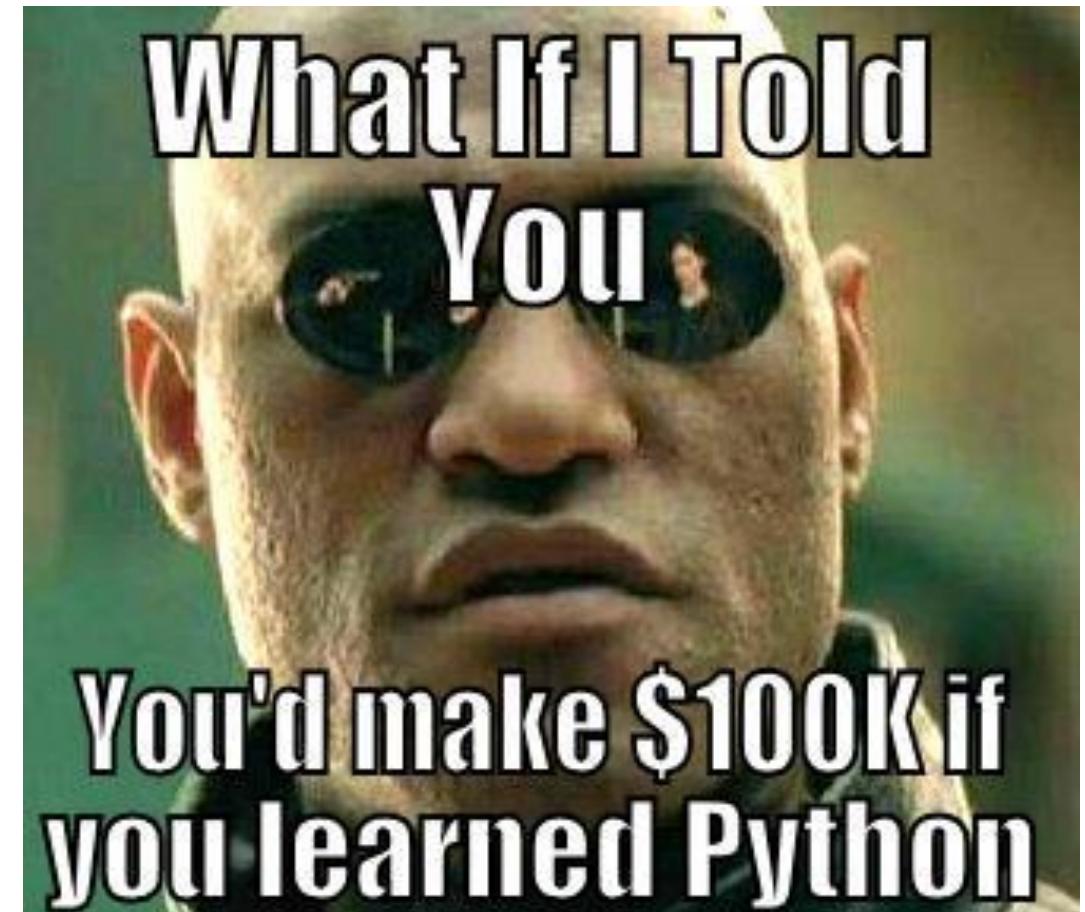
minutes

seconds

[KÚP SI LÍSTOK](#)

# Zen filozofia Pythonu

1. Krásny je lepší než škaredý
2. Explicitný je lepší ako implicitný
3. Jednoduchý je lepší ako zložitý
4. Zložitý je lepší ako komplikovaný
5. Plochý je lepší ako vnorený
6. Riedky je lepší ako hustý
7. Na čitateľnosti záleží
8. Praktickosť vyhráva nad čistotou



import this

# Čaká nás krásna budúcnosť

```
>>> from __future__ import braces  
SyntaxError: not a chance (<pyshell#13>, line 2)  
>>> |
```

No future {} a ;



# Inšpirácia projekty

# Python Project Ideas

Easy



Quote Gener.



Number guessing



Dice Simulation



YT downloader

Mid



Password Manag.



Mario Party

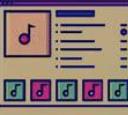


Web Crawler



Email Autom.

Pro



Music Player



Face Detection



Twitter Clone



Twitter Bot

@TheInsaneApp

# Ako Skončíme?

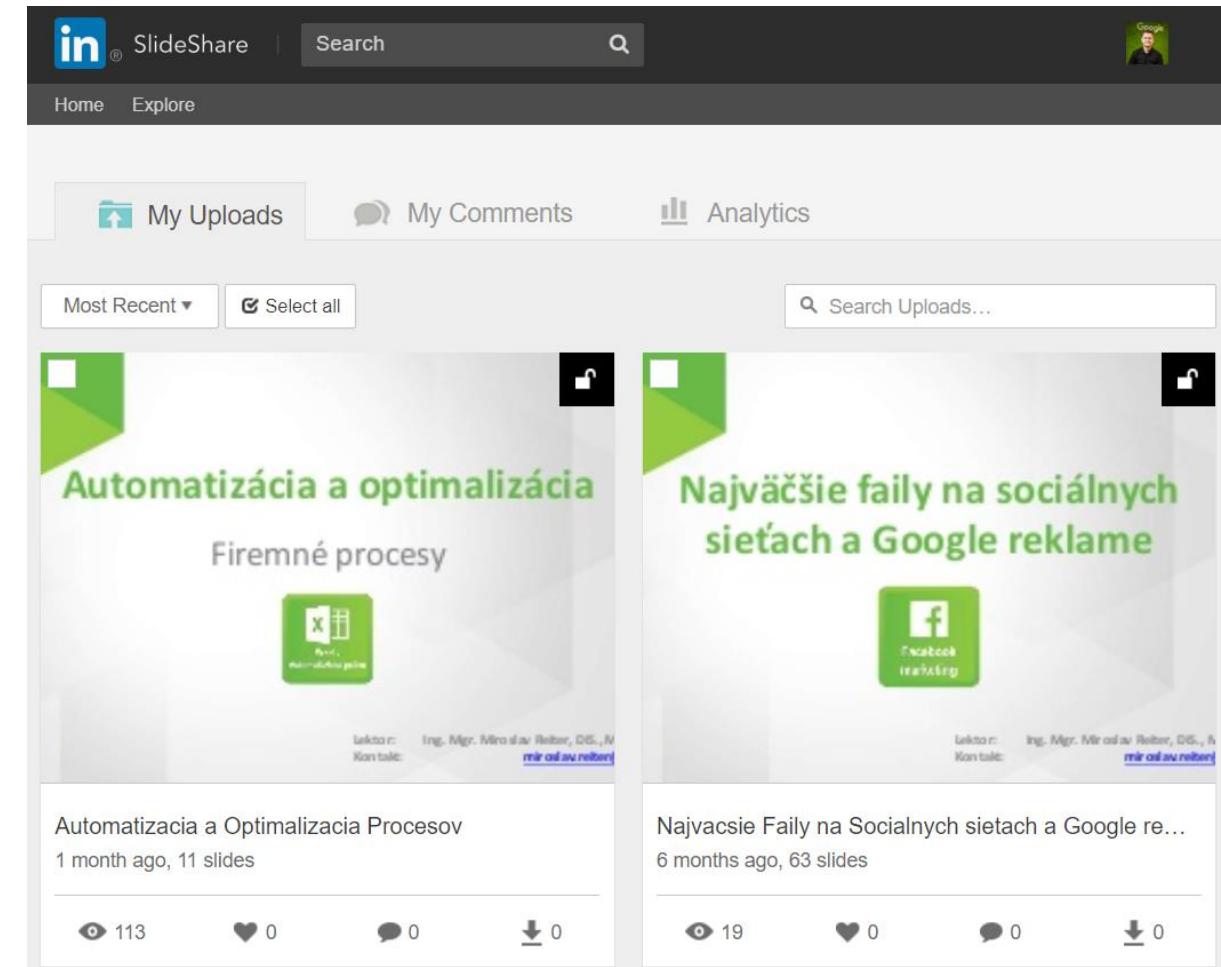
## 1. Stiahnite si Súbory

- [https://github.com/miroslav-reiter/Kurzy\\_SAV\\_Analytika\\_Python\\_R](https://github.com/miroslav-reiter/Kurzy_SAV_Analytika_Python_R)

## 2. Pridajte si ma na LinkedIn

- [www.linkedin.com/in/miroslav-reiter/](http://www.linkedin.com/in/miroslav-reiter/)

## 3. Prezentácia a materiály po prednáške



# Vyber si online kurz

Nauč sa programovať, tvoriť webstránky a grafiku, manažovať alebo sa zameraj na osobný rozvoj. Všetko jednoducho vďaka našim online kurzom z pohodlia tvojho domova.

Ročné  
predplatné na  
všetky online  
kurzy

~~2299.99€~~

399.99€

Priístup pre Teba do všetkých aktuálnych aj pripravovaných online kurzov

12 mesačná platnosť

🛒 Kúpiť teraz

## Zadarmo

1. Kurzy SAV

2. YouTube kanál IT Academy

<https://www.youtube.com/c/IT-AcademySK>

## Platené

Moje kurzy na [www.vita.sk](http://www.vita.sk)

# Online Kurzy Zdarma



Úrady práce,  
sociálnych  
vecí a rodiny



[info@vita.sk](mailto:info@vita.sk)



Práca a  
zamestnanosť



Rodina a sociálna  
pomoc



Sociálne poistenie  
a dôchodkový  
systém



Ministerstvo

INŠPEKCIJA V SOCIÁLNYCH  
VECIACH

Aktívne starnutie

Pomoc pre odídencov z Ukrajiny

Informácie pre médiá

Aktuality

Informácie pre verejnosť

Európsky orgán práce ELA

Pracovná mobilita cudzincov

Elektronické služby, formuláre a  
žiadosti

Legislatíva

Organizácie ministerstva

Programy Európskej únie

Plán obnovy a odolnosti

Potravinová a základná  
materiálna pomoc

Národné kompetenčné centrum  
pre sociálne inovácie

Neprehliadnite

Podcasty o vývoji na trhu práce

Naučné kurzy

19.03.2024

## Úrady práce sociálnych vecí a rodiny podporia rozvoj zručnosti

Nový národný projekt Zručnosti pre trh práce prinesie už od 8.apríla možnosť bezplatne získať finančnú podporu na vzdelávanie či rekvalifikáciu. Je to prvý projekt svojho druhu, ktorý je určený pre ľudí bez práce, ale aj pre zamestnaných.

„Podobné projekty boli v minulosti obľúbené a fungovali, no vlane skončili. Jednou z našich priorit hned po nástupe na ministerstvo bolo opäťovne a v čo najkratšom čase podobný projekt opäť rozbehnuť. A tak sme pripravili národný projekt Zručnosti pre trh práce. Vzdelávanie je klúčovým nástrojom na podporu nezamestnaných pri ich ceste späť na trh práce, ako aj pre ľudí v ohrození stratou zamestnania. Tak ako pri ostatných nástrojoch, aj v tomto prípade sme urobili viac a priniesli inováciu s cieľom zvýšiť efektivitu a kvalitu služieb pre občanov a v jednom projekte spájame podporu pre ľudí bez práce aj zamestnaných. Okrem toho majú ľudia možnosť využiť tento projekt opakovane,“ povedal minister práce, sociálnych vecí a rodiny Erik Tomáš.

Cieľom nového národného projektu Zručnosti pre trh práce je podporiť uchádzačov o zamestnanie v snahе nájsť si prácu prostredníctvom rekvalifikácie. Projekt bude pôsobiť aj preventívne. Možnosť získať bezplatné vzdelávanie ponúkne aj zamestnaným záujemcom o zamestnanie, a to s cieľom ochrániť a zlepšiť ich postavenie na trhu práce. Projekt nie je určený pre samostatne zárobkovo činné osoby.

„Tento projekt sme nastavovali citlivо, na základe potrieb ľudí, ale aj údajov o trhu práce tak, aby bola pomoc čo najúčinnejšia. Väčší priestor a viac podpory sme v rámci neho dali ľuďom, ktorí čelia komplikovanejšej životnej situácii a lepšie vzdelanie u nich má väčší potenciál podstatne im zlepšiť kvalitu života. Aj preto sme kládli dôraz na znevýhodnených uchádzačov o zamestnanie a mladých ľudí do 30 rokov bez práce,“ dodal generálny riaditeľ a generálny tajomník služobného úradu Ústredia práce, sociálnych vecí a rodiny Peter Ormandy.

Žiadatelia si sami vyberú vzdelávanie a aj jeho poskytovateľa. Úrady práce, sociálnych vecí a rodiny budú akceptovať iba poskytovateľov pôsobiacich v oblasti vzdelávania najmenej jeden rok. O príspevok na vzdelávanie bude možné žiadať od 8.apríla na príslušnom úrade práce, sociálnych vecí a rodiny, ktorý každú žiadosť individuálne posúdi. Žiadosť o poskytnutie príspevku je potrebné podať najneskôr 30 kalendárnych dní pred začiatkom kurzu. Pri scievaní žiadostí o poskytnutie príspevku sa žiadateľ a jeho rodina prihládajú na individuálne predpoklady a poskytnutie príspevku nie je právny nárok, môže byť odmietnuté. Úrady práce, sociálnych vecí a rodiny žiadost schvália a poskytnú príspevok, keďže žiadateľ a jeho rodina zaplatí vzdelávací kurz. V rámci tohto projektu sa žiadateľ a jeho rodina môžu získať finančnú podporu na vzdelávanie či rekvalifikáciu.

info@vita.sk

# Ako sa s nami Spojit'?



**ADRESA:** IT Academy, s. r. o.

Budova KOLOSEO prízemie  
Tomášikova 50/A  
831 04 Bratislava



**WEB:** [www.it-academy.sk](http://www.it-academy.sk)



**E-MAIL:** [info@it-academy.sk](mailto:info@it-academy.sk)



**TELEFÓN:** +421 917 095 406



# Ako Vieme Pomôcť?

#Copywriting

#Školenia

#Zamestnanci

#Pomáhame

#Rast

#Projekty

#Certifikácie

#Kurzy

#Tréningy

#Vzdelávanie

#PPC Kampane

#Elearning

#Mentoring

#Konzultácie

#Online

#Programovanie

#Vývoj

#Marketing

#Reklama

#Prenájom Techniky

- [Domov](#)
- [Shorts](#)
- [Odbory](#)
- [Moje](#)
- [História](#)

Ak chcete k videám pridať označenie páči sa mi, komentovať alebo sa prihlásiť na odber, musíte sa prihlásiť.

[Prihlásiť sa](#)

#### Preskúmať

- [Trendy](#)
- [Hudba](#)
- [Filmy](#)
- [Hry](#)
- [Šport](#)
- [Prehliadat kanály](#)

#### Viac zo služby YouTube

- [YouTube Premium](#)
- [YouTube Music](#)
- [YouTube Kids](#)

# Naštartuj sa s nami v IT!



**PREZENČNÉ KURZY**  
[www.it-academy.sk](http://www.it-academy.sk)



**ONLINE KURZY**  
[www.vita.sk](http://www.vita.sk)



**Daj Odber**

## IT Academy


**IT ACADEMY**

@IT-Academy · 7,02 tis. odberateľov · 938 videí

[Online certifikované kurzy a školenia IT, marketingu a manažmentu.](#) >

[vita.sk a 8 ďalších odkazov](#)
[Odoberať](#)
[Domov](#) [Videá](#) [Shorts](#) [Naživo](#) [Zoznamy](#) [Komunita](#) [Hľadať](#)

### Videá

[Prehrať všetko](#)

[Online kurz Data Science a Jazyk R - Jemný Úvod do...](#)

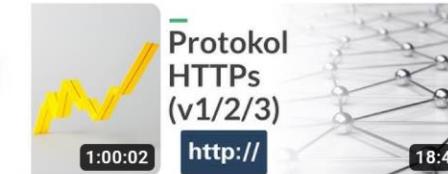
7 zhliadnutí · pred 1 hodinou


[Online kurz Python - Ako programuje Google v Python?](#)

51 zhliadnutí · pred 1 dňom


[Online Kurz Microsoft Excel Grafy a Typy Grafov](#)

61 zhliadnutí · pred 2 dňami


[Online Kurz Počítačové Siete - Protokol HTTP\(s\), Stavové...](#)

166 zhliadnutí · pred 3 dňami


[Online Kurz Efektívna Komunikácia - Názory a...](#)

60 zhliadnutí · pred 3 dňami


[Online kurz Microsoft Outlook - Ako na...](#)

142 zhliadnutí · pred 5 dňami

### Oblíbené videá

[Prehrať všetko](#)

[SQL I. Začiatčník](#)

1:51:23


[Microsoft Excel - Kontingenčné Tabuľky](#)

1:51:58


[Siete I. Základy Siete](#)

1:40:33


[AKO SA STAŤ TESTEROM](#)

2:45:09


[Microsoft Access I. Začiatčník](#)

1:46:27


[Ako efektívne pracovať v Microsoft Excel?](#)

2:07:28

# Dajte odber na IT Academy



[www.YOUTUBE.COM/C/IT-ACADEMYSK](https://www.youtube.com/c/IT-ACADEMYSK)



IT ACADEMY