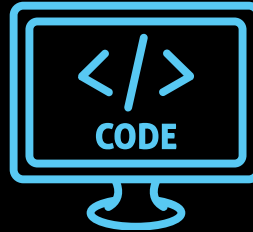






Projekty  
matematyczne



Projekty  
programistyczne



Aplikacje  
webowe



Artykuły  
popularnonaukowe



Wydarzenia



Integracje

# Czym jest AI?



# Czym jest AI?

ARTIFICIAL INTELLIGENCE (AI)

W JĘZYKU POLSKIM: SZTUCZNA INTELIGENCJA (SI)

to programy komputerowe oparte na uczeniu maszynowym potrafiące przyjmować dane od użytkowników i uczyć się na nich. Nauczone programy potrafią transformować lub generować od nowa dane na podstawie zapamiętanych parametrów zależnie od potrzeb wykorzystania.



Supervised Learning



Machine Learning

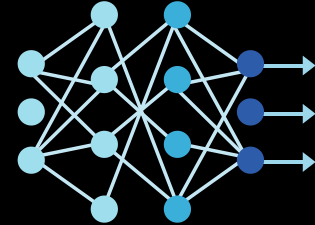


Unsupervised Learning

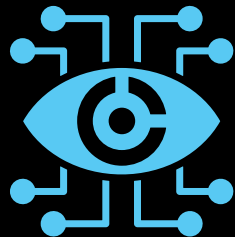


Reinforcement Learning

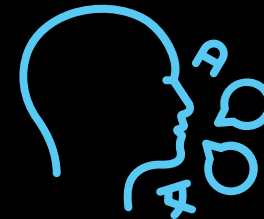
Typy AI



Deep Learning



Computer Vision

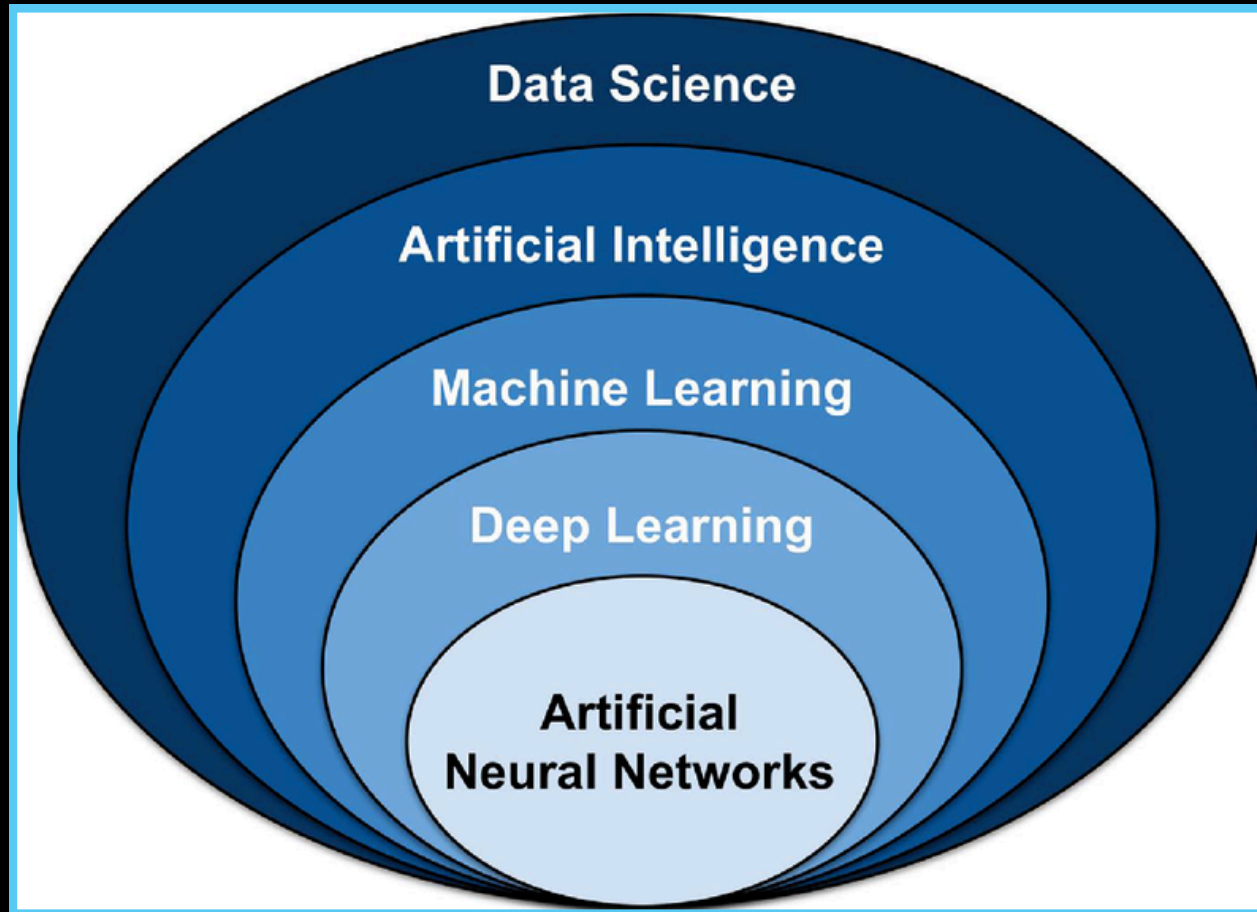


Natural Language Processing

# Typy AI

- Uczenie Nadzorowane (Supervised Learning)
- Uczenie Nienadzorowane (Unsupervised Learning)
- Uczenie Maszynowe (Machine Learning - ML)
- Uczenie Głębokie (Deep Learning - DL)
- Uczenie przez wzmocnienie (Reinforcement Learning - RL)
- Wizja Komputerowa (Computer Vision - CV)
- Przetwarzania języka naturalnego (Natural Language Processing - NLP)

# Typy AI



<https://viso.ai/deep-learning/ml-ai-models/>

# Klasyfikacja





# Klasyfikacja

Klasyfikacja, jak sama nazwa wskazuje, polega na przypisaniu obiektom o konkretnych parametrach jakieś etykiety najlepiej je opisujące.

Często używa się jej w Wizji Komputerowej, a głównymi metodami które pozwalają ją uzyskać jest Uczenie Nadzorowane i Uczenie Głębokie.

# Klasyfikacja

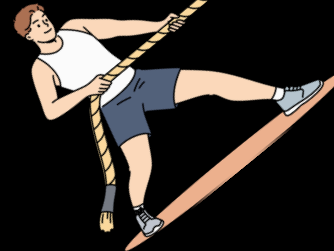
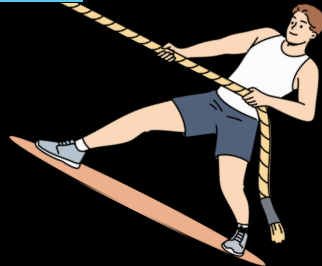


<https://viso.ai/deep-learning/ml-ai-models/>

# Regresja

zmienna zależna  
(docelowa)

zmienne niezależne  
(predykcyjne)



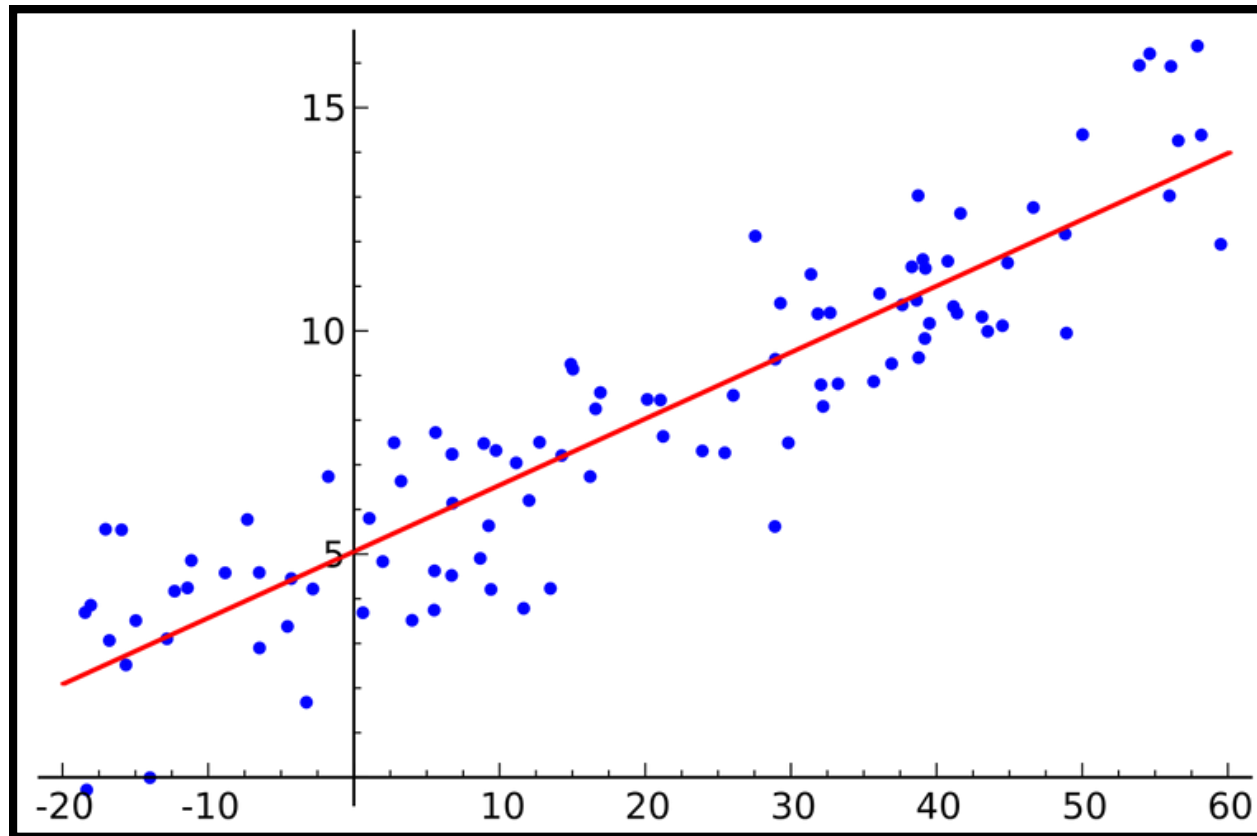
funkcja

# Regresja

Regresja to podejście statystyczne stosowane do analizy relacji między zmienną zależną (zmienną docelową) a jedną lub większą liczbą zmiennych niezależnych (zmiennymi predykcyjnymi).

Celem jest określenie najbardziej odpowiedniej funkcji charakteryzującej związek między tymi zmiennymi.

# Regresja



# AI w języku Python

W języku programowania Python istnieją dwa główne moduły do tworzenia modeli AI. Jednym z nich jest TensorFlow, który został stworzony przez Google i obsługuje też języki Java, JavaScript, C++. Drugim bardzo popularnym w świecie AI modulem do Pythona i skonstruowanym szczególnie pod ten język jest PyTorch.





znacznie szybszy, bo bazuje na działaniach bezpośrednio na tensorach

problem z niektórymi danymi -> konieczność dodatkowej obróbki przed ich konwersją na tensory

składnia typowa dla Pythona

proste tworzenie przez dziedziczenie klas

wspiera typy danych z biblioteki NumPy

również używa tensorów, ale działania na nich są abstrahowane przez moduł

# PyTorch vs TensorFlow

PyTorch posługuje się składnią typową dla Pythona i pozwala w prosty sposób tworzyć sieci neuronowe czy inne konstrukty przez dziedziczenie klas, oraz wspiera typy danych z biblioteki NumPy, umożliwiając szeroką adaptację modeli pod różny typ danych. Podobnie jak TensorFlow, używa tensorów, ale działania na nich są abstrahowane przez moduł.



# PyTorch vs TensorFlow

TensorFlow bazuje na działaniach bezpośrednio na tensorach i jest przez to znacznie szybszy, lecz może sprawiać problemy podczas tworzenia modeli ze względu na to, że niektóre dane nie mogą być przekonwertowane bezpośrednio na tensory i potrzebują dodatkowej, specjalistycznej do zastosowania obróbki.

Co to tensor?



# Co to tensor?

Jeśli miało się doświadczenie z algebrą liniową, tensor można porównać do wektora lub macierzy. Tensory są obiektami matematycznymi niezależnymi od układu współrzędnych, w którym się znajdują.

Zestawy obliczeń i działań na tensorach nazywa się równaniami tensorowymi. Ponadto, takie równania jak i same obiekty będąc słuszne w jednym układzie współrzędnych będą słuszne w każdym innym układzie współrzędnych.

# Co to tensor?

Scalar

Vector

Matrix

Tensor

1

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$
$$\begin{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 4 \end{bmatrix} \end{bmatrix}$$

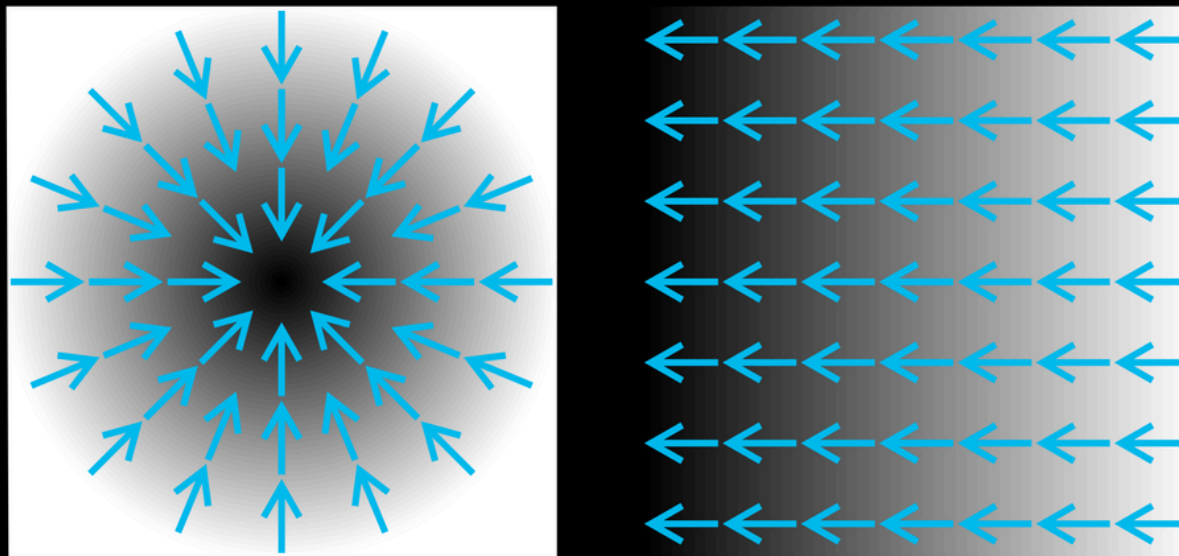
Co to gradient?



# Gradienty

Gradientem określa się pole wektorowe, w którym każdy wektor określa kierunek najszybszego wzrostu dla danej funkcji, a ich długości określają szybkość wzrostu w tym kierunku.

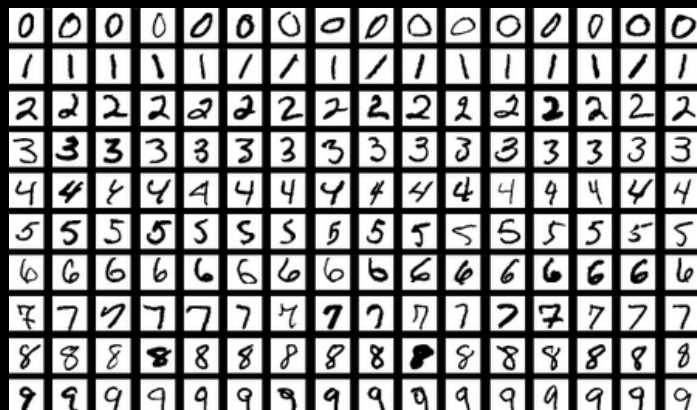
Gradienty często używane są w modelach AI przy ich trenowaniu, pomagając określić w jaki sposób wagi i biasy neuronów powinny być dostosowane tak, aby ich kolejne przewidywania były lepsze.



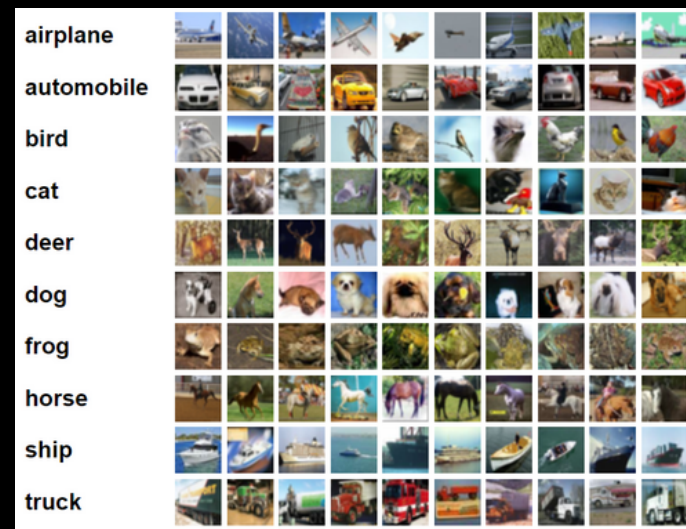
# Zestawy danych

Zestawy danych, wymagane do wytrenowania i testowania sieci neuronowych mogą być pobrane z ogólnodostępnych i gotowych zestawień pod konkretne zastosowania (np. dla klasyfikacji: CIFAR, MNIST, CALTECH).

PyTorch pozwala też tworzyć własne zestawy danych i ładować je w łatwy sposób dla sieci.



Przykład MNIST



Przykład CIFAR

# Zestawy danych

```
class CustomImageDataset(Dataset): 2 usages
    def __init__(self, annotations_file, img_dir, transform=None, target_transform=None, test=False):
        self.img_labels = pd.read_csv(annotations_file)
        self.img_dir = f"{img_dir}/test" if test else f"{img_dir}/train"
        self.transform = transform
        self.target_transform = target_transform

    def __len__(self):
        return len(self.img_labels)

    def __getitem__(self, idx):
        img_path = os.path.join(self.img_dir, self.img_labels.iloc[idx, 0])
        image = read_image(img_path)
        label = self.img_labels.iloc[idx, 1]
        if self.transform:
            image = self.transform(image)
        if self.target_transform:
            label = self.target_transform(label)
        return image, label
```

Przykład definiowania własnego zestawu danych, Jakub Susoł



# Gdzie znaleźć dodatkowe materiały o AI?

The Kaggle logo, featuring the word "kaggle" in a blue, lowercase, sans-serif font with a small trademark symbol (TM) to the upper right.The logo for Sebastian Raschka, consisting of the name "Sebastian" in black and "Raschka" in red, both in a sans-serif font, enclosed within a light blue rectangular border.The GeeksforGeeks logo, featuring a green stylized "GG" symbol above the text "GeeksforGeeks" in a black sans-serif font, all enclosed within a light blue rectangular border.

# Gdzie znaleźć dodatkowe materiały o AI?

Dodatkową wiedzę o AI, tworzeniu modeli, metodach trenowania i testowania oraz sposobach działania modeli można znaleźć w naprawdę dużej ilości źródeł. Dla samego PyTorch wszystkie metody z przykładami ich zastosowania są dobrze udokumentowane na głównej stronie modułu.

Ponadto polecanymi źródłami informacji są:

- Sebastian Rashka ([sebastianraschka.com](http://sebastianraschka.com))
- Kaggle ([kaggle.com](http://kaggle.com))
- Geeks For Geeks ([geeksforgeeks.org](http://geeksforgeeks.org))

# A gdzie liczyć?



CPU

vs

GPU

