

Wprowadzenie do uczenia maszynowego

dr inż. Marcin Nowak

marcin.nowak@poznan.merito.pl

Literatura

1. Geron, A. (2023). Uczenie maszynowe z użyciem Scikit-Learn, Keras i TensorFlow, Helion
2. Gallatin, K. (2024). Uczenie maszynowe w Pythonie. Receptury. Od przygotowania danych do deep learningu, Helion
3. Hurbans, R. (2020) Algorytmy sztucznej inteligencji. Ilustrowany przewodnik, Helion
4. Raschka, S., Mirjalili, V. (2019). Python. Uczenie maszynowe. Wydanie II, Helion

Historia i koncepcje uczenia maszynowego

Sztuczna inteligencja - to szeroka dziedzina technologii, która obejmuje tworzenie maszyn naśladujących ludzką inteligencję i funkcje poznawcze, takie jak rozwiązywanie problemów i uczenie się. AI wykorzystuje prognozy i automatyzację do optymalizacji i rozwiązywania złożonych zadań, które historycznie były wykonywane przez ludzi (IBM).

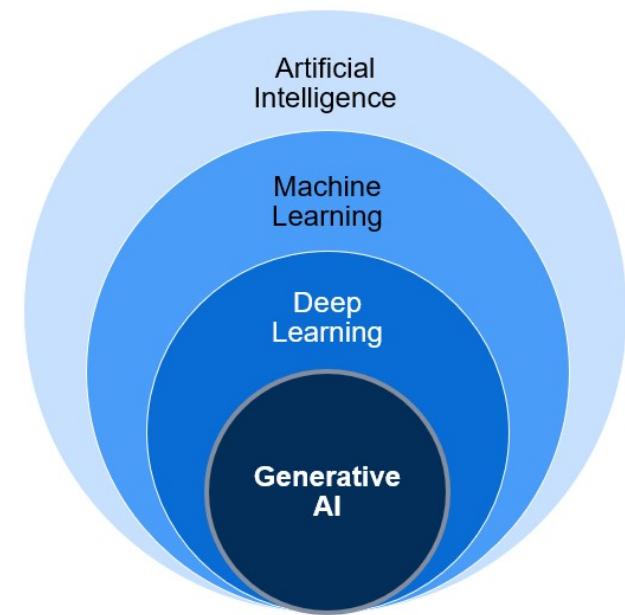
Sztuczna inteligencja to dziedzina badań i rodzaj technologii charakteryzującej się rozwojem i wykorzystaniem maszyn zdolnych do wykonywania zadań, które zazwyczaj wymagałyby ludzkiej inteligencji (World Economic Forum).



Historia i koncepcje uczenia maszynowego

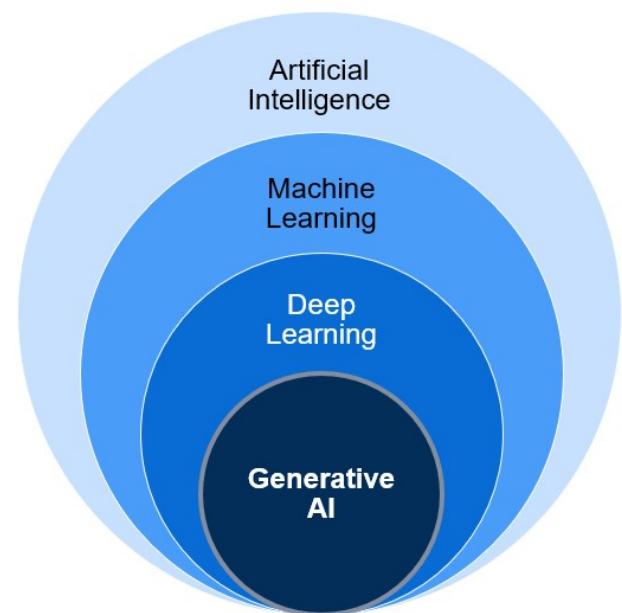
"Uczenie maszynowe (ML) to dziedzina sztucznej inteligencji, która umożliwia systemom komputerowym uczenie się i poprawianie swoich zdolności na podstawie danych, bez konieczności jawnego programowania"
– Arthur Samuel (1959).

„Uczeniem maszynowym nazywa się proces, w którym program komputerowy uczy się na podstawie doświadczeń E rozwiązania problemów T względem kryterium oceny P , o ile jakość Q , z jaką rozwiązuje on problemy T rośnie wraz z doświadczeniem E ” (Mitchel, 1997).



Historia i koncepcje uczenia maszynowego

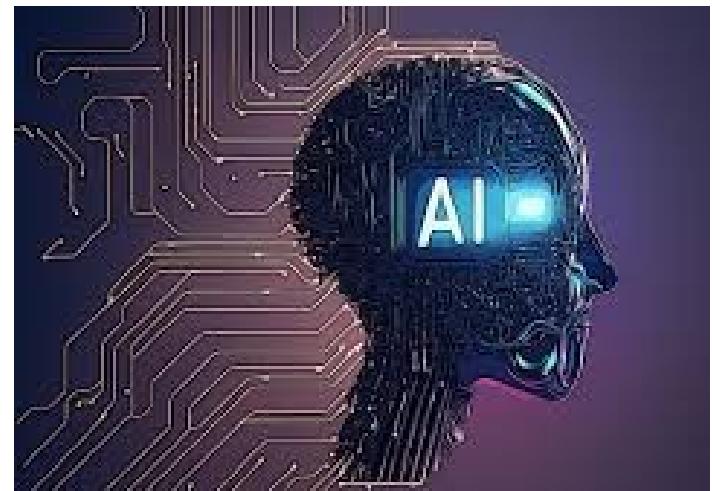
- Uczenie maszynowe jest szczególnym przypadkiem AI
- Szczególnym przypadkiem uczenia maszynowego są sztuczne sieci neuronowe
- Szczególnym przypadkiem SSN są głębokie SSN
- Szczególnym przypadkiem głębokich SSN są modele generatywne
- Szczególnym przypadkiem modeli generatywnych jest czat GPT



Podstawowe pojęcia

Przykładowa AI niebędąca uczeniem maszynowym:

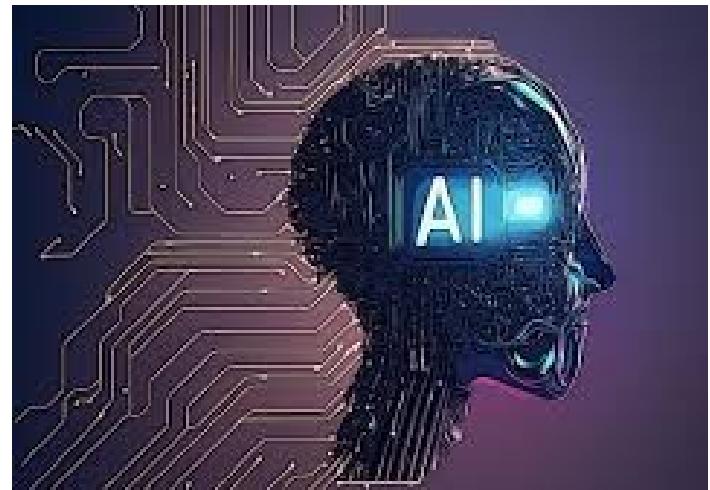
- **Systemy eksperckie** - Systemy eksperckie to programy komputerowe, które naśladują zdolności podejmowania decyzji przez ludzkiego eksperta. Wykorzystują one bazy wiedzy zawierające fakty i reguły dotyczące określonej dziedziny, aby wyciągać wnioski i rozwiązywać problemy. Przykładem może być system diagnostyczny w medycynie, który pomaga lekarzom w diagnozowaniu chorób na podstawie objawów pacjentów.
- **Algorytmy wyszukiwania i optymalizacji** - Algorytmy takie jak A* (A-star) czy algorytmy przeszukiwania heurystycznego są używane do rozwiązywania problemów z wyszukiwaniem ścieżek, optymalizacją i planowaniem. Na przykład, algorytmy te są stosowane w grach komputerowych do znajdowania najkrótszej ścieżki postaci z punktu A do punktu B.



Podstawowe pojęcia

Przykładowa AI niebędąca uczeniem maszynowym:

- **Algorytmy ewolucyjne i genetyczne** - Algorytmy te naśladują procesy biologiczne, takie jak selekcja naturalna i mutacja, aby znaleźć optymalne rozwiązania problemów. Są one wykorzystywane w optymalizacji i projektowaniu, np. do tworzenia efektywnych układów elektronicznych.
- **Robotyka** - koncentruje się na tworzeniu i programowaniu robotów, które są zdolne do wykonywania zadań fizycznych w rzeczywistym świecie, często na podstawie wcześniej zaprogramowanych instrukcji.



Uczenie maszynowe



Uczenie maszynowe

- filtrowanie spamu mailowego,
- systemy rekomendacyjne (Amazon, Netflix),
- wybór postów do pokazania w portalu Facebook,
- wybór twitów na portalu X,
- translator Google,
- system rekomendacyjny Spotify oraz You Tube,
- nawigacja Google Maps,
- sprawdzania gramatyki,
- monitoring z kamer przemysłowych,
- wykrywanie anomalii w systemach informatycznych,
- automatyczne odczytywanie adresów z listów,
- odszukiwanie niezdecydowanych wyborców,
- interpretacja zdjęć,
- szeroko zakrojona diagnostyka zdrowotna,
- przewidywanie cen nieruchomości,
- serwisy randkowe,
- autonomiczne pojazdy,
- osobista asystentka Siri.

Historia i koncepcje uczenia maszynowego

Krótką historią uczenia maszynowego

- Lata 50.–60. - Początki – Arthur Samuel i program do warcabów
- Lata 80. - Rozwój sieci neuronowych i backpropagation
- Lata 90. - Wzrost popularności maszyn wektorów nośnych (SVM) i metod ensemble
- XXI wiek - Przełom w głębokim uczeniu (Deep Learning) dzięki big data i GPU
- Współczesność - ML w biznesie, autonomiczne systemy, generatywna AI (np. modele językowe)

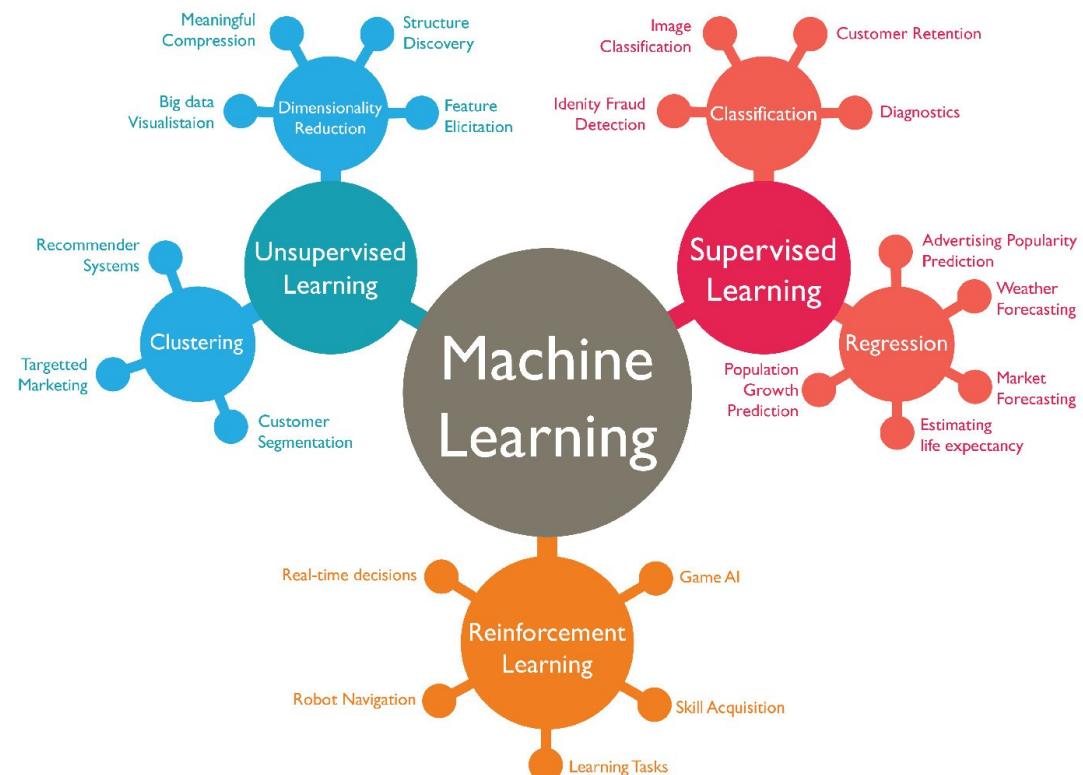
Historia i koncepcje uczenia maszynowego

Kluczowe daty w historii ML

- 1950 - Alan Turing publikuje "Computing Machinery and Intelligence" – pytanie o możliwość myślenia maszyn
- 1959 - Arthur Samuel definiuje uczenie maszynowe i tworzy program do warcabów
- 1986 - Backpropagation w sieciach neuronowych (Rumelhart, Hinton, Williams)
- 1995 - Wprowadzenie maszyn wektorów nośnych (Vapnik i Cortes)
- 2012 - AlexNet wygrywa ImageNet, rewolucjonizując głębokie uczenie
- 2020+ - Modele językowe (np. GPT-3, BERT) i generatywna AI

Praca Turinga z 1950 roku zarysowała teoretyczne podstawy sztucznej inteligencji, pytając, czy maszyny mogą myśleć. Program Samuela do warcabów był jednym z pierwszych przykładów systemu uczącego się. Backpropagation umożliwiło efektywne trenowanie sieci neuronowych, a AlexNet w 2012 roku pokazał siłę głębokiego uczenia w rozpoznawaniu obrazów. Współczesne modele językowe, takie jak GPT-3, ilustrują, jak ML ewoluowało w kierunku bardziej zaawansowanych i wszechstronnych zastosowań.

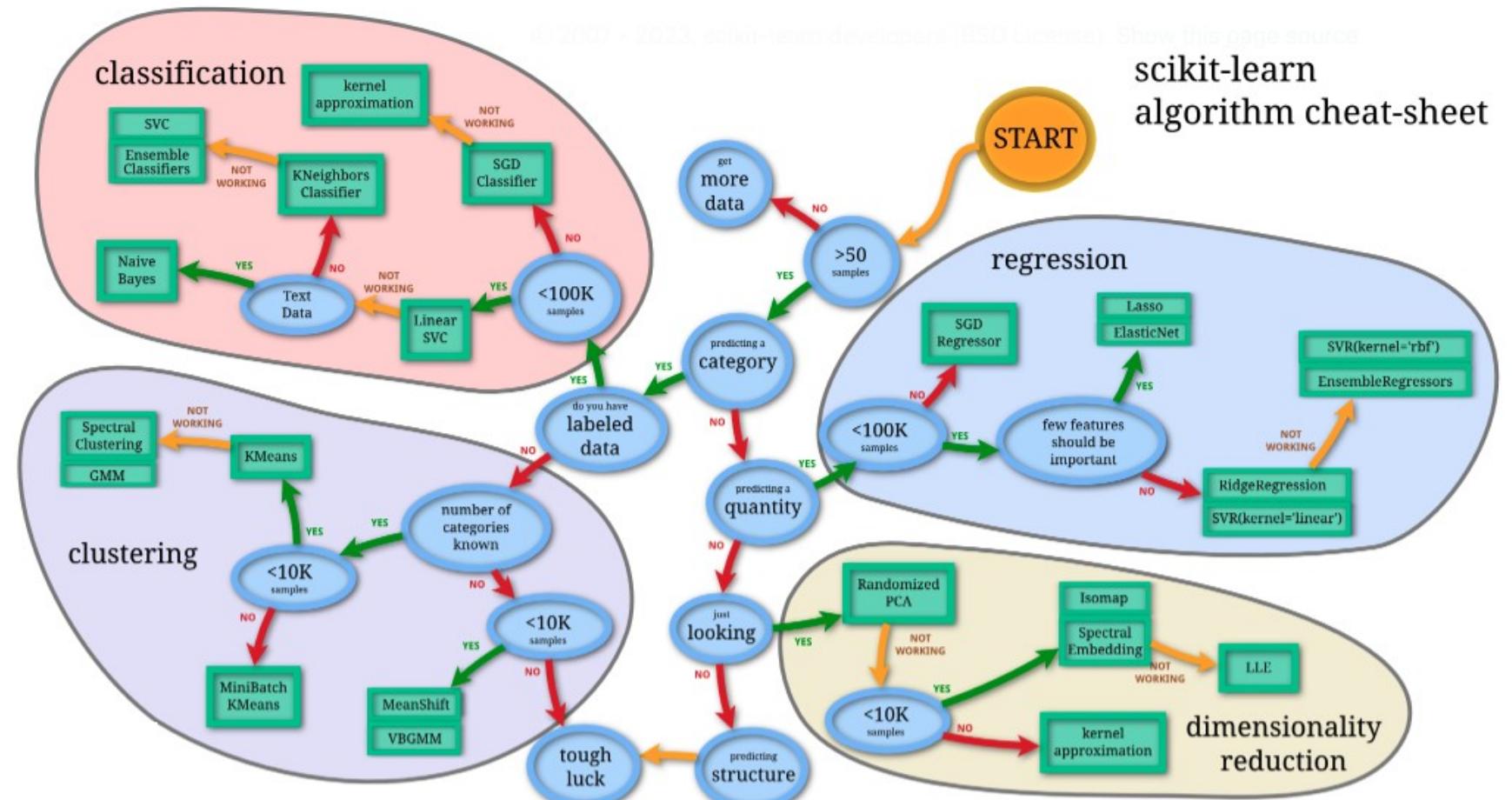
Historia i koncepcje uczenia maszynowego



Główne paradygmaty uczenia maszynowego

- Uczenie nadzorowane** - dane z etykietami, np. przewidywanie cen
- Uczenie nienadzorowane** - dane bez etykiet, np. grupowanie klientów
- Uczenie ze wzmocnieniem** - uczenie przez nagrody, np. autonomiczne pojazdy
- Uczenie półnadzorowane** - połączenie danych oznaczonych i nieoznaczonych
- Uczenie transferowe** - wykorzystanie wiedzy z jednego zadania do innego

Historia i koncepcje uczenia maszynowego



Historia i koncepcje uczenia maszynowego

Uczenie nadzorowane

Definicja: Algorytmy uczą się na podstawie danych oznaczonych (wejście + etykieta)

Przykłady zastosowań:

- Przewidywanie cen mieszkań
- Rozpoznawanie spamu w e-mailach
- Klasyfikacja obrazów (np. kot vs pies)

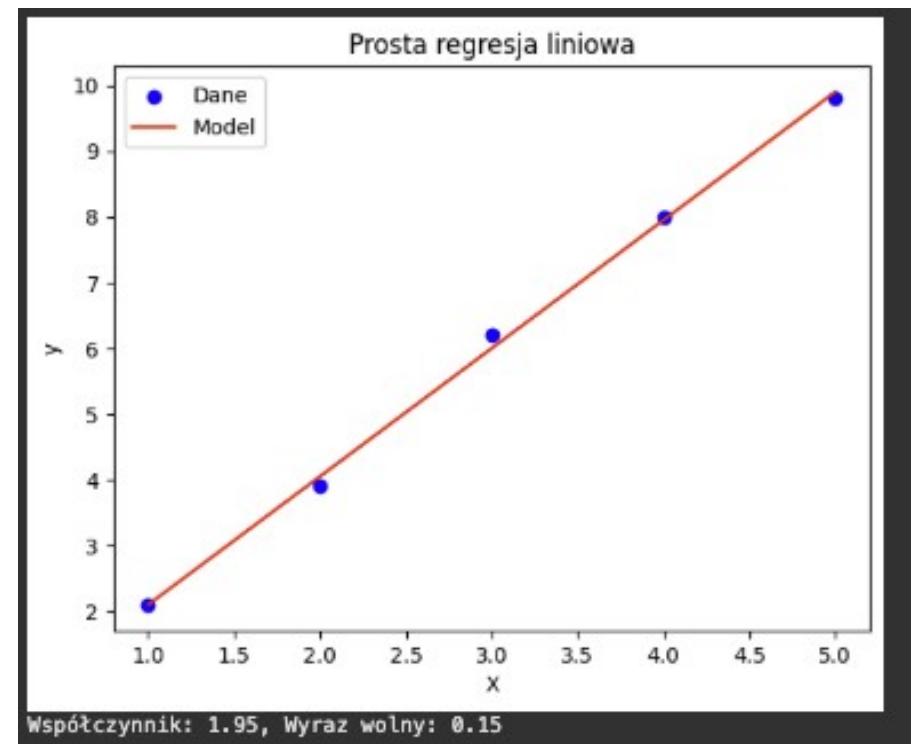
Popularne algorytmy: Regresja liniowa, regresja logistyczna, drzewa decyzyjne

Schemat: Dane → Model → Predykcja

Historia i koncepcje uczenia maszynowego

Przykład kodu – regresja liniowa

```
[1] 1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.linear_model import LinearRegression
4
5 # Dane syntetyczne
6 X = np.array([[1], [2], [3], [4], [5]])
7 y = np.array([2.1, 3.9, 6.2, 8.0, 9.8])
8
9 # Trenowanie modelu
10 model = LinearRegression()
11 model.fit(X, y)
12
13 # Predykcje
14 y_pred = model.predict(X)
15
16 # Wizualizacja
17 plt.scatter(X, y, color='blue', label='Dane')
18 plt.plot(X, y_pred, color='red', label='Model')
19 plt.xlabel('X')
20 plt.ylabel('y')
21 plt.title('Prosta regresja liniowa')
22 plt.legend()
23 plt.show()
24
25 # Wyświetlenie parametrów
26 print(f'Współczynnik: {model.coef_[0]:.2f}, Wyraz wolny: {model.intercept_:.2f}')
```



Historia i koncepcje uczenia maszynowego

Uczenie nienadzorowane – definicja i przykłady

Definicja: Algorytmy analizują dane bez etykiet, szukając wzorców lub struktur

Przykłady zastosowań:

- Segmentacja klientów w marketingu
- Kompresja danych (np. PCA)
- Wykrywanie anomalii w danych

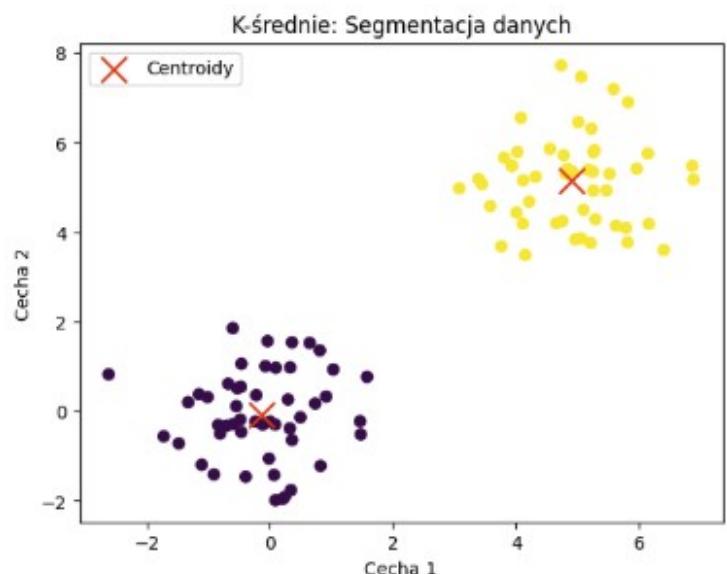
Popularne algorytmy: K-średnie, DBSCAN, PCA

Schemat: Dane → Wzorce (np. grupy, klastry)

Historia i koncepcje uczenia maszynowego

Przykład kodu – k-średnie

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.cluster import KMeans
4
5 # Dane syntetyczne
6 np.random.seed(42)
7 X = np.concatenate([np.random.normal(0, 1, (50, 2)), np.random.normal(5, 1, (50, 2))])
8
9 # K-średnie
10 kmeans = KMeans(n_clusters=2, random_state=42)
11 kmeans.fit(X)
12
13 # Wizualizacja
14 plt.scatter(X[:, 0], X[:, 1], c=kmeans.labels_, cmap='viridis')
15 plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], c='red', marker='x', s=200, label='Centroidy')
16 plt.title('K-średnie: Segmentacja danych')
17 plt.xlabel('Cecha 1')
18 plt.ylabel('Cecha 2')
19 plt.legend()
20 plt.show()
```



Historia i koncepcje uczenia maszynowego

Uczenie ze wzmocnieniem – definicja i przykłady

Definicja: Algorytmy uczą się przez interakcję ze środowiskiem, maksymalizując nagrodę

Przykłady zastosowań:

- Gry komputerowe (np. AlphaGo)
- Sterowanie robotami
- Optymalizacja procesów biznesowych

Kluczowe elementy: Agent, środowisko, akcje, nagrody

Schemat: Agent → Akcja → Środowisko → Nagroda

Przykład AlphaGo pokazuje, jak RL osiągnął przełomowe wyniki w grach, a inne zastosowania (np. robotyka, optymalizacja procesów) wskazują na jego wszechstronność.

Historia i koncepcje uczenia maszynowego

Uczenie półnadzorowane

Definicja: Połączenie danych oznaczonych i nieoznaczonych w celu poprawy wydajności modelu

Przykłady zastosowań:

- Rozpoznawanie obrazów z ograniczoną liczbą etykiet
- Analiza sentymentu w mediach społecznościowych

Zalety: Redukcja kosztów oznaczania danych

Schemat: Dane oznakowane + nieoznakowane → Model

Przykłady, takie jak rozpoznawanie obrazów, pokazują, jak można wykorzystać małą liczbę oznakowanych danych w połączeniu z dużą ilością danych nieoznaczonych. Zalety (redukcja kosztów) są istotne w biznesie, np. w analizie dużych zbiorów danych z mediów społecznościowych.

Historia i koncepcje uczenia maszynowego

Uczenie transferowe

Definicja: Wykorzystanie modelu wytrenowanego na jednym zadaniu do rozwiązania innego, pokrewnego zadania

Przykłady zastosowań:

- Pretrenowane modele językowe (np. BERT)
- Transfer modeli w rozpoznawaniu obrazów

Zalety: Szybsze trenowanie, mniejsze zapotrzebowanie na dane

Schemat: Model pretrenowany → Nowy zbiór danych → Dostosowany model

Przykłady, takie jak BERT, pokazują, jak pretrenowane modele mogą być dostosowane do specyficznych zadań (np. analiza sentymentu). Zalety uczenia transferowego, takie jak oszczędność czasu i danych, są istotne w biznesie, gdzie zasoby mogą być ograniczone.

Historia i koncepcje uczenia maszynowego

Uczenie transferowe – przykład z modelem obrazowym

```
1 from tensorflow.keras.applications import VGG16
2 from tensorflow.keras.models import Model
3 from tensorflow.keras.layers import Dense, Flatten
4
5 # Ładowanie pretrenowanego modelu VGG16
6 base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
7
8 # Dodanie nowych warstw
9 x = Flatten()(base_model.output)
10 x = Dense(128, activation='relu')(x)
11 predictions = Dense(2, activation='softmax')(x)
12
13 # Tworzenie modelu
14 model = Model(inputs=base_model.input, outputs=predictions)
15
16 # Zamrożenie warstw bazowych
17 for layer in base_model.layers:
18     layer.trainable = False
19
20 # Kompilacja modelu
21 model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
22 print("Model gotowy do treningu na nowym zbiorze danych!")

→ Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 0s 0us/step
Model gotowy do treningu na nowym zbiorze danych!
```

Przykład użycia uczenia transferowego z pretrenowanym modelem VGG16 (siecią konwolucyjną) w TensorFlow/Keras. Kod ładuje model przeszkolony na zbiorze ImageNet, dodaje nowe warstwy do klasyfikacji binarnej (np. kot vs pies) i zamraża warstwy bazowe, aby dostosować model do nowego zadania. Wyjaśnienie podkreśla prostotę uczenia transferowego i jego praktyczne zastosowanie w oszczędzaniu zasobów

Historia i koncepcje uczenia maszynowego

Rola danych w uczeniu maszynowym

Dane jako paliwo ML: Jakość i ilość danych decydują o skuteczności modelu

Rodzaje danych:

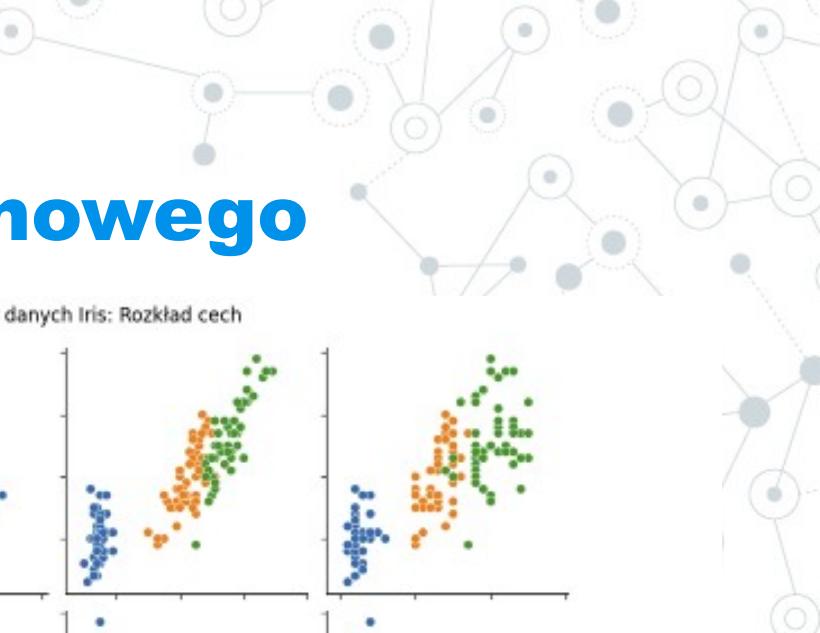
- Strukturalne (np. tabele)
- Niestrukturalne (np. obrazy, tekst)

Wyzwania:

- Braki w danych
- Szum
- Niezbalsowane zbiory danych

Przykład: Zbiór danych Iris (klasyfikacja kwiatów)

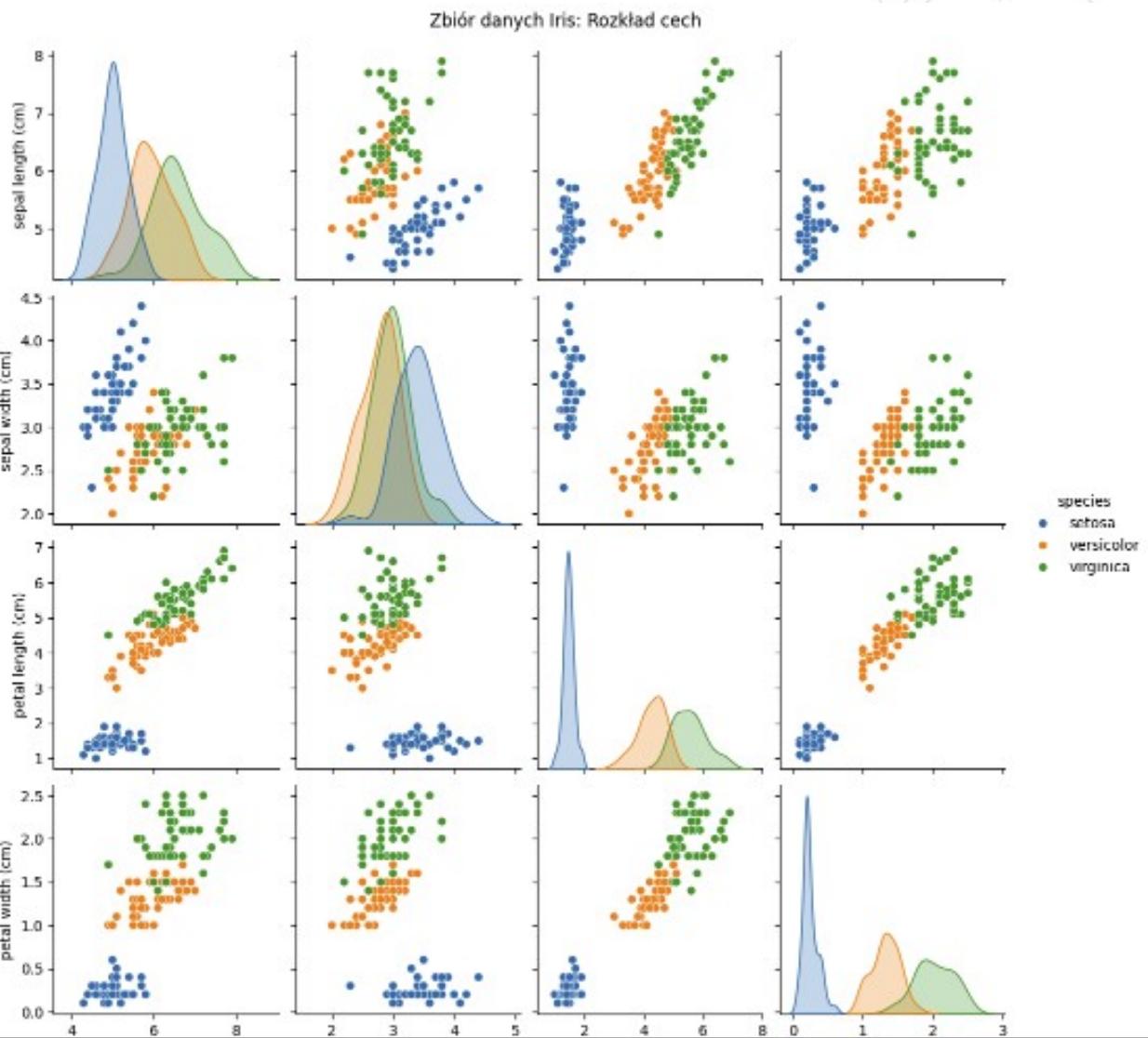
Historia i koncepcje uczenia maszynowego



Przykład kodu – eksploracja zbioru Iris

```
1 import pandas as pd
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4 from sklearn.datasets import load_iris
5
6 # Ładowanie danych Iris
7 iris = load_iris()
8 df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
9 df['species'] = iris.target_names[iris.target]
10
11 # Wizualizacja
12 sns.pairplot(df, hue='species')
13 plt.suptitle('Zbiór danych Iris: Rozkład cech', y=1.02)
14 plt.show()
15
16 # Statystyki opisowe
17 print(df.describe())
```

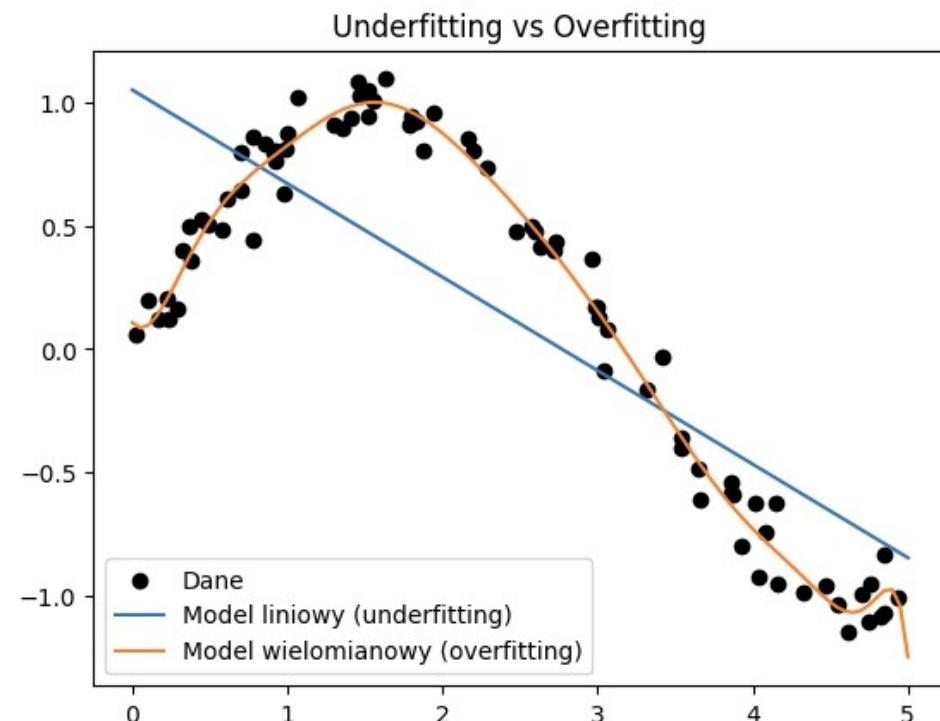
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000



Historia i koncepcje uczenia maszynowego

Podział danych na treningowe i testowe:

- Istota podziału,
- Przeuczenie i niedouczenie,
- Typowe proporcje,



Historia i koncepcje uczenia maszynowego

Zastosowania uczenia maszynowego w biznesie

Marketing:

- Segmentacja klientów
- Systemy rekomendacji

Finanse:

- Wykrywanie oszustw
- Ocena ryzyka kredytowego

Logistyka:

- Optymalizacja tras
- Prognozowanie popytu

Przykłady firm: Netflix, Amazon, Google

Historia i koncepcje uczenia maszynowego

Wyzwania w uczeniu maszynowym

- **Jakość danych:** Braki, szum, niebalansowanie
- **Obliczenia:** Duże zbiory danych wymagają dużej mocy obliczeniowej
- **Interpretowalność:** "Czarna skrzynka" w modelach ML
- **Etyka:** Uprzedzenia w danych, prywatność
- **Przykłady:** Uprzedzenia w rekrutacji, problemy z prywatnością danych

Historia i koncepcje uczenia maszynowego

Rola obliczeń w ML

Ewolucja mocy obliczeniowej:

CPU → GPU → TPU

Big Data: ML potrzebuje dużych zbiorów danych

Chmura obliczeniowa: AWS, Google Cloud, Azure

Przykłady: Trenowanie modeli głębokiego uczenia na GPU

▫ **CPU** (Central Processing Unit) – uniwersalny procesor ogólnego zastosowania.

▫ **GPU** (Graphics Processing Unit) – procesor graficzny, dobrze nadający się do obliczeń równoległych, dlatego znalazł zastosowanie w AI.

▫ **TPU** (Tensor Processing Unit) – wyspecjalizowany układ stworzony wyłącznie pod macierze i tensory (podstawowe struktury danych w sieciach neuronowych).

Historia i koncepcje uczenia maszynowego

Wprowadzenie do narzędzi ML

Biblioteki Python:

- Scikit-learn: Proste modele ML
- TensorFlow/Keras: Głębokie uczenie
- Pandas: Analiza danych

Środowiska: Jupyter Notebook, Google Colab

Przykłady zastosowań: Analiza danych, trenowanie modeli

Zalety: Łatwość użycia, społeczność open-source

Historia i koncepcje uczenia maszynowego

Przykład kodu – prosta klasyfikacja z wykorzystaniem regresji logistycznej

```
 1 from sklearn.datasets import make_classification
 2 from sklearn.model_selection import train_test_split
 3 from sklearn.linear_model import LogisticRegression
 4 from sklearn.metrics import accuracy_score
 5
 6 # Generowanie danych z 2 cechami informacyjnymi i 0 redundantnymi
 7 X, y = make_classification(
 8     n_samples=100,
 9     n_features=2,
10     n_informative=2,
11     n_redundant=0,
12     n_classes=2,
13     random_state=42
14 )
15
16 # Podział na zbiór treningowy i testowy
17 X_train, X_test, y_train, y_test = train_test_split(
18     X, y,
19     test_size=0.2,
20     random_state=42
21 )
22
23 # Trenowanie modelu
24 model = LogisticRegression()
25 model.fit(X_train, y_train)
26
27 # Predykcje i ocena
28 y_pred = model.predict(X_test)
29 print(f'Dokładność: {accuracy_score(y_test, y_pred):.2f}')
30
```

Dokładność: 0.95