Proces uczenia się SI

Michał Kasprowicz
michal.kasprowicz@wroclaw.merito.pl

Plan zajęć

O przedmiocie

1. Wymagania wstępne
1. 2. Osiągnięcie efektów wynikających z realizacji kursów: -
2. Założenia i cele uczenia się
Celem przedmiotu jest nabycie zaawansowane wiedzy oraz praktycznych umiejętności z zakresu uczenia maszynowego i sztucznej inteligencji. Kurs koncentruje się na implementacji, optymalizacji i wdrażaniu algorytmów uczenia maszynowego do rozwiązywania rzeczywistych problemów. Studenci nauczą się wybierać odpowiednie metody, przygotowywać dane, trenować modele, oceniać ich jakość oraz interpretować wyniki w kontekście praktycznych zastosowań.

UMIEJĘTNOŚCI

Kod przedmiotowego efektu uczenia się	Opis przedmiotowego efektu uczenia się	Odniesienie do kierunkowych efektów uczenia się
PUS_U01	Analizuje i optymalizuje proces uczenia się w systemach SI.	
PUS_U02	Implementuje różne strategie uczenia maszynowego.	
PUS_U03	Ocenia efektywność procesów uczenia się.	

O przedmiocie

9. Sposób zaliczenia	
Na ocenę 5,0	***91% - 100%***
Na ocenę 4,5	***81% - 90%***
Na ocenę 4,0	***71% - 80%***
Na ocenę 3,5	***61% - 70%***
Na ocenę 3,0	***51% - 60%***
Na ocenę 2,0	***Poniżej 51%***
Obecność i aktywny udział w zajęciach, dopuszczalna jest jedna nieobecność. Należy ustalić z prowadzącym formę zaliczenia materiału przerabianego na zajęciach, które student opuścił; uzyskanie co najmniej oceny dostatecznej ze sprawozdań dotyczących zadań realizowanych podczas zajęć (karty pracy)	

10. Opis sposobu sprawdzenia efektów uczenia się			
Przedmiotowy efekt uczenia się	Egzamin/Zaliczenie	Projekt	Aktywności
PUS_U01			+
PUS_U02			+
PUS_U03			+
PUS_K01			+

Metody sprawdzające:	
Zadania praktyczne	90 %
Prezentacja przeprowadzonych badań/analiz i ich wyników	10 %
	100/100

Harmonogram

- 11.10. - 2 godziny
- 25.10. - 2 godziny
- 15.11. - 2 godziny
- 29.11. - 2 godziny
- 13.12. - 1 godzina
- 17.01. - 1 godzina
- 31.01. - 1 godzina
- 14.02. - 1 godzina

4. Szczegółowy program zajęć

Nr	*Treści programowe*	*Przedmiotowe efekty uczenia się*
1.	Praktyczne ćwiczenia w środowisku Anaconda. -	PUS_U01, PUS_U02, PUS_U03, PUS_K01
2.	Eksperymenty z różnymi sposobami reprezentacji danych. -	PUS_U01, PUS_U02, PUS_U03, PUS_K01
3.	Eksperymenty z różnymi strategiami uczenia. -	PUS_U01, PUS_U02, PUS_U03, PUS_K01
4.	Eksperymenty z różnymi modelami w celu zrozumienia problemu overfittingu. -	PUS_U01, PUS_U02, PUS_U03, PUS_K01
5.	Implementacja transfer learning z użyciem pretrenowanych modeli. -	PUS_U01, PUS_U02, PUS_U03, PUS_K01
6.	Praktyczne zastosowanie narzędzi AutoML do automatyzacji procesu uczenia. -	PUS_U01, PUS_U02, PUS_U03, PUS_K01
7.	Implementacja algorytmów uczenia przyrostowego. -	PUS_U01, PUS_U02, PUS_U03, PUS_K01
8.	Techniki oceny modeli i interpretacji wyników. Wizualizacja wyników i analiza błędów. -	PUS_U01, PUS_U02, PUS_U03, PUS_K01

5. Metody dydaktyczne

Wykład tradycyjny, Ćwiczenia,

Obszary tematyczne

- Praktycznie ćwiczenia w środowisku Anaconda.
  - **Cel**: zapewnienie spójnego środowiska pracy, zarządzanie środowiskami, instalacja pakietów, obsługa notebooków.
- Eksperymenty z różnymi sposobami reprezentacji danych.
  - **Cel**: normalizacja, metody kodowania danych, redukcja wymiarowości, tworzenie cech. Generalnie, dane i ich format mogą mieć większy wpływ na wynik, niż wybór modelu.
- Eksperymenty z różnymi strategiami uczenia.
  - **Cel**: zrozumienie i sprawdzenie różnych paradygmatów (uczenie nadzorowane, nienadzorowane, przez wzmocnienie).
- Eksperymenty z różnymi modelami w celu zrozumienia poblemu overfittingu.
  - **Cel**: zrozumienie czym jest bias-variance tradeoff.
- Implementacja transfer learningu z użyciem pretrenowanych modeli.
  - **Cel**: wykorzystanie dużych, pretrenowanych modeli do małych problemów.

Obszary tematyczne

- Praktycznie ćwiczenia w środowisku Anaconda.
  - **Cel**: zapewnienie spójnego środowiska pracy, zarządzanie środowiskami, instalacja pakietów, obsługa notebooków
- Eksperymenty z różnymi sposobami reprezentacji danych.
  - **Cel**: normalizacja, metody kodowania danych, redukcja wymiarowości, tworzenie cech. Generalnie, dane i ich format mogą mieć większy wpływ na wynik, niż wybór modelu.
- Eksperymenty z różnymi strategiami uczenia.
  - **Cel**: zrozumienie i sprawdzenie różnych paradygmatów (uczenie nadzorowane, nienadzorowane, przez wzmocnienie)
- Eksperymenty z różnymi modelami w celu zrozumienia poblemu overfittingu.
  - **Cel**: zrozumienie czym jest bias-variance tradeoff
- Implementacja transfer learningu z użyciem pretrenowanych modeli.
  - **Cel**: wykorzystanie dużych, pretrenowanych modeli do małych problemów.
  - Uwaga: część opcjonalna dla projektu.

Obszary tematyczne

- Praktyczne zastosowanie narzędzi AutoML do automatyzacji procesu uczenia.
  - **Cel**: automatyzacja etapów budowy modelu, które zazwyczaj wykonuje się ręcznie.
- Implementacja algorytmów uczenia przyrostowego.
  - **Cel**: trenowanie modelu dla scenariusza w którym dane regularnie napływają i nie ma możliwości ponownego trenowania modelu od nowa.
  - Uwaga: część opcjonalna dla projektu.
- Techniki oceny modeli, interpretacja i wizualizacja wyników.
  - **Cel**: głębsze zrozumienie błędów modelu.

Zaliczenie

Zaliczenie

- Do wyboru dwie drogi:
  - Egzamin na ostatnich zajęciach
    - Dużo nauki i przygotowań
    - Wynik: zakuj, zdaj, zapomnij
  - Projekt zespołowy
    - Praktyczne zastosowanie wiedzy i prawdziwe rozwiązanie problemu
    - Wynik: hands-on project, który może być inspiracją do kolejnych

Projekt

- **Idea**: od danych do wytrenowanego i wyjaśnionego modelu
- **Definicja:** Wybór realnego problemu, który da się rozwiązać za pomocą uczenia maszynowego (klasyfikacja, regresja, grupowanie) i realizacja kompletnego procesu wraz z uzasadnieniem.
- **Dane:** zebrane samodzielnie (ankieta, scraping, pomary, dane publiczne, własnoręcznie wykonane zdjęcia).
- **Pipeline:**
  - Przygotowanie i analiza danych – analiza eksploracyjna, czyszczenie, transformacja
  - Wybór modelu i trening – porównanie paru modeli/strategii (**bonus:** dodatkowy eksperyment wykazujący zjawisko przeuczenia)
  - Ocena i interpretacja – wybór metryk, analiza błędów, omówienie wyników
  - Wizualizacja i prezentacja – 5-10 minutowa prezentacja na ostatnich zajęciach (**wymagany:** działający kod, **bonus:** profesjonalna, ciekawa prezentacja, efekt wow!)
- **Warunki techniczne:**
  - Inkrementacyjnie (po każdych zajęciach) uzupełniane repozytorium (git).
  - Główny notebook działa po „Restart & Run All“.
  - Oprócz repozytorium – prezentacja (PDF/ppt/odp/etc) na ostatnich zajęciach.
  - Zestaw danych powinien być umieszczony na zewnętrznym dysku/chmurze (google drive, one drive, dropbox). Nie wolno dodawać do repozytorium na gitea (!!!!).

Projekt

- Zespoły **2** (max 3) osobowe.
- Zespół **buduje** realny projekt uczenia maszynowego, krok po kroku, na podstawie zajęć laboratoryjnych, dodając do niego kolejne elementy.
- Zalecane jest **uzgodnienie** które tematy są już zajęte przez inny zespół, aby nie dochodziło do powielania się tematów.
- Każdy zespół **zbiera samodzielnie** dane. Pobranie gotowego zbioru z Kaggle (lub innego źródła) **nie wystarczy** – przynajmniej połowa zbioru testowego **musi zawierać własne dane**.
- **Obszary**: klasyfikacjia tabularna, regresja, obraz/dźwięk, clustering (nienadzorowane).
- Zespół rozbudowuje projekt, aktualizując repozytorium po każdych zajęciach.

Projekt

Dane:

- Projekt musi zawierać kartę danych (1 strona prezentacji):
  - Co było zbierane, czym, kiedy i gdzie,
  - Opis rozkładu w klasach oraz każdej fali,
  - Jak dane były oznaczane,
  - Jakie są problemy oraz ograniczenia zbioru (potencjalnie lub jawnie).

Wymaganie	Klasyfikacja	Regresja	Obrazy	Clustering
Rozmiar zbioru	Min. 400 obserwacji	Min. 400 obserwacji	Min. 60 zdjęć na klasę	Min. 400 obserwacji
Złożoność problemu	Min. 3 klasy	Min. 6 cech	Min. 3 klasy	Min. 3 klasy
Pozyskanie danych	Min. 2 źródła	Min. 2 źródła, cel nie może być przewidywalny.	Zdjęcia własne lub min. 2 źródła + augmentacja	Min. 2 źródła, scraping lub ankieta + czyszczenie
Czego unikać	Danych z publicznych źródeł w stanie surowym	Celu stworzonego ze wzoru danych	Jedej sesji zdjęciowej (data leak)	Klastrowania bez pytania badawczego

Przykładowe obszary tematyczne

- **Rozpoznawanie gestów dłoni** – klasyfikacja obrazów, dane zbierane za pomocą telefonu, kamery w laptopie, itp.
- **Rozpoznawanie aktywności fizycznej** (chodzenie, bieganie, aktywność fizyczna) – szeregi czasowe, dane z telefonu/zegarka.
- **Klasyfikacja dźwięków otoczenia** (koncert, autobus, ulica, uczelnia, sklep) – dyktafon w telefonie
- **Rozpoznawanie odręcznie napisanych tekstów z polskimi znakami** – obraz, arkusze wypełnione przez członków zespołu + znajomi
- **Dojrzałość owoców** – codziennie robione zdjęcie tych samych owoców, ręcznie etykietowanie
- **Rozpoznawanie gatunków drzew** – spacery w parku i stworzenie bazy zdjęć liści
- **Rozpoznawanie produktów** (np. spożywczych) – zdjęcia na różnym tle, w różnym świetle, różne perspektywy
- **Przewidywany czas przejazdów** (np. dojazd do pracy w różne dni tygodnia i o różnych porach)
- **System rekomendacji** (literatura, film, muzyka) – na podstawie danych publicznych oraz własnych badań/ankiet
- **Klasyfikacja jakości zdjęć** – rozmyte, poruszone, dzień, noc

Dzisiejsze zajęcia

Czym jest uczenie maszynowe?

- **Uczenie maszynowe** to dziedzina sztucznej inteligencji, w której systemy uczą się wzorców z danych bez jawnego programowania reguł.
- **Uczenie maszynowe** (ang. machine learning) – obszar sztucznej inteligencji poświęcony algorytmom, które poprawiają się automatycznie poprzez doświadczenie, czyli ekspozycję na dane. Algorytmy uczenia maszynowego budują model matematyczny na podstawie przykładowych danych (zbiór uczący), w celu prognozowania lub podejmowania decyzji bez bycia zaprogramowanym do tego celu. *Wikipedia (https://pl.wikipedia.org/wiki/Uczenie_maszynowe)*
- Systemy oparte na algorytmach uczenia maszynowego są wykorzystywane w zastosowaniach, takich jak ochrona przed spamem lub rozpoznawanie obrazów, w których opracowanie konwencjonalnych algorytmów do wykonywania potrzebnych zadań jest **trudne** lub **niewykonalne**.
- Kluczowe cechy:
  - Automatyzacja procesu uczenia
  - Wykorzystuje się dane do tworzenia modeli
  - Zdolność do adaptacji i poprawy wydajności

Czym jest uczenie maszynowe?

- Współczesny rozwój uczenia maszynowego
  - Wzrost **mocy obliczeniowej** komputerów
  - Dostępność ogromnych zbiorów danych (**Big Data**)
  - Nowe algorytmy i modele
  - Zastosowanie we wszystkich dziedzinach:
    - Rozpoznanie obrazów
    - Rozpoznanie mowy, dźwięków
    - Motoryzacja – automatyczne pojazdy
    - Systemy rekomendacji (netflix, youtube)
    - Medycyna – diagnostyka medyczna
- Przyszłość?
  - Rozwój modeli językowych i generatywnej AI
  - Iternet of Things (IoT)
  - Asystent osobisty
  - Wyzwania etyczne i prawne
  - Możliwość rozwiązywania globalnych problemów (łańcuchy dostaw, opieka zdrowotna, zmiany klimatyczne)

Czym jest uczenie maszynowe?

- Algorytmy uczenia maszynowego są wykorzystywane tam, gdzie reguły są:
  - Zbyt złożone do ręcznego sterowania
  - Zmienne w czasie
  - Nieznane z góry
- Nowy paradygmat:
  - Programowanie klasyczne
    Dane + Reguły → Program → Odpowiedzi
  - Uczenie maszynowe
    Dane + Odpowiedzi → Trenig → Reguły (Model)
- Model jest wyuczonym zestawem reguł wyrażonych matematycznie.
- **Konsekwencja:**

Czym jest uczenie maszynowe?

- Algorytmy uczenia maszynowego są wykorzystywane tam, gdzie reguły są:
  - Zbyt złożone do ręcznego sterowania
  - Zmienne w czasie
  - Nieznane z góry
- Nowy paradygmat:
  - Programowanie klasyczne
    Dane + Reguły → Program → Odpowiedzi
  - Uczenie maszynowe
    Dane + Odpowiedzi → Trenig → Reguły (Model)
- Model jest wyuczonym zestawem reguł wyrażonych matematycznie.
- **Konsekwencja:**

  jakość modelu jest bezpośrednio powiązana z jakością danych (garbage in, garbage out)

Czym jest uczenie maszynowe?

- Gdzie mamy do czynienia ze zjawiskiem uczenia maszynowego?
  - **Serwisy streamingowe** – podpowiedzi oparte na indywidualnych preferencjach użytkownika.
  - **Filtry antyspamowe** – klasyfikator tekstu.
  - **Rozpoznawanie twarzy w telefonie** – neuronowa sieć konwolucyjna.
  - **Algorytmy wspomagania podejmowania decyzji** – np. scoring konsumencki w banku.
  - **Tłumaczenie** tekstu.
  - **Rozpoznawanie anomalii** (dane finansowe, medyczne)
  - **Systemy bezpieczeństwa** – analiza obrazu lub video.

Uczenie maszynowe

- Trzy główne kategorie:
  - **Uczenie nadzorowane** (*ang. supervised learning*)
    - Wykorzystanie etykietowanych danych (wejście x + poprawna odpowiedź y)
    - Klasyfikacja spamu, predykcja cen mieszkań (klasyfikacja / regresja)
  - **Uczenie nienadzorowane** (*ang. unsupervised learning*)
    - Dane bez etykiet (znajdowanie podobieństw)
    - Grupowanie klientów, detekcja anomalii, kompresja danych (klasteryzacja / redukcja wymiarowości)
  - **Uczenie ze wzmocnieniem** (*ang. reinforcement learning*)
    - Zestaw nagród i kar za podejmowane przez model decyzje
    - Robotyka, gry (AlphaGo)
- **Uczenie transferowe** (*ang. transfer learning*)
  - Wykorzystanie wstępnie wytrenowanego modelu i dotrenowanie na własnych danych
- **Uczenie półnadzorowane** (*ang. semi-supervised learning*)
  - Wykorzystanie niewielkiego zbioru danych z etykietami z dużym zbiorem danych bez etykiet. Punkty danych blisko siebie są podobne (ta sama etykieta), dane tworzą skupiska a granica decyzyjna powinna przechodzić przez obszary o niskiej gęstości

Uczenie maszynowe

- **Uczenie nadzorowane** – metoda, w której model uczony jest na oznaczonych danych, zawierających zarówno wejścia, jak i odpowiadające im poprawne wyjścia.
- Celem jest nauczenie (**wykrycie wzorców**) modelu przewidywania wejść dla nowych, nieznanych przez model danych.
- **Wymaga ludzkiego nadzoru** w procesie uczenia.
- Wykorzystuje zestaw danych treningowych:
  - Dane wejściowe - wektor cech
  - Pożądane odpowiedzi (etykiety)
- Rodzaje problemów:
  - Regresja – przewidywanie wartości liczbowych
  - Klasyfikacja – rozpoznanie zjawiska (np. chory vs. zdrowy, spam vs. nie spam)
- **Zalety**: wysoka dokładność przedwidywań, możliwość rozwiązania złożonych problemów
- **Wyzwania**: konieczność posiadania dużej ilości oznakowanych danych (wysoki koszt), może dochodzić do przeuczenia modelu.

Uczenie maszynowe

- **Algorytmy liniowe**
  - Regresja liniowa
  - Regresja logistyczna
- **Algorytmy oparte na drzewach**
  - Drzewa decyzyjne
  - Lasy losowe
- **Algorytmy oparte na podobieństwie i geometrii**
  - k-NN (k-nearest neighbors), leniwy model, nic nie trenuje jedynie liczy odległości w predykcji
  - SVM (maszyna wektorów nośnych) – poszukiwanie granicy decyzyjnej z maksymalnym marginesem, kernel trick
- **Sieci neuronowe i uczenie głębokie**
  - Neurony ułożone w sieć
  - MLP – dane tabelaryczne
  - CNN – obraz, dźwięk
  - RNN – dane sekwencyjne, szeregi czasowe
  - Transformery – tekst, sekwencje, obraz, dźwięk

Uczenie maszynowe

- **Regresja liniowa** – podstawowa metoda statystyczna służąca do modelowania zależności między zmiennymi.
- Opisuje zależności między zmienną zależną a jedną lub wieloma zmiennymi niezależnymi.
- Celem jest znalezienie funkcji liniowej, która najlepiej opisuje badaną zależność

  Model matematyczny:

  Y = ax + b

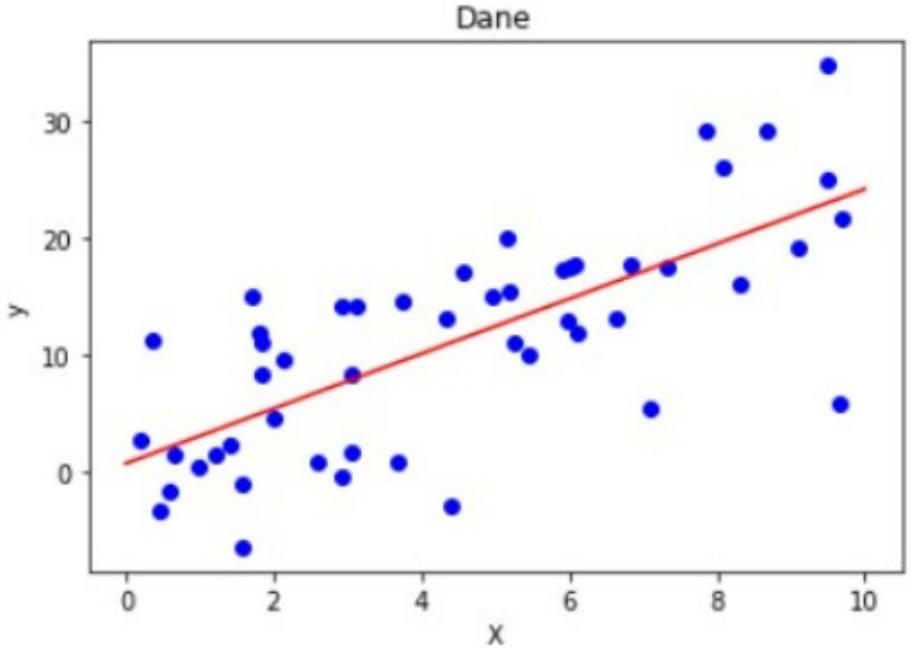
  gdzie

  Y - zmienna zależna

  x – zmienna niezależna

  a – współczynnik kierunkowy

  b – wyraz wolny (punkt przecięcia z osią Y)



Uczenie maszynowe

- **Regresja wielomianowa** – metoda statystyczna służąca do modelowania złożonych, nieliniowych zależności między zmiennymi.
- Opisuje nieliniowe zmiany w zależnościach
- Analiza trendów, prognozowanie, modelowanie zjawisk fizycznych, optymalizacja procesów produkcyjnych.

  Model matematyczny:

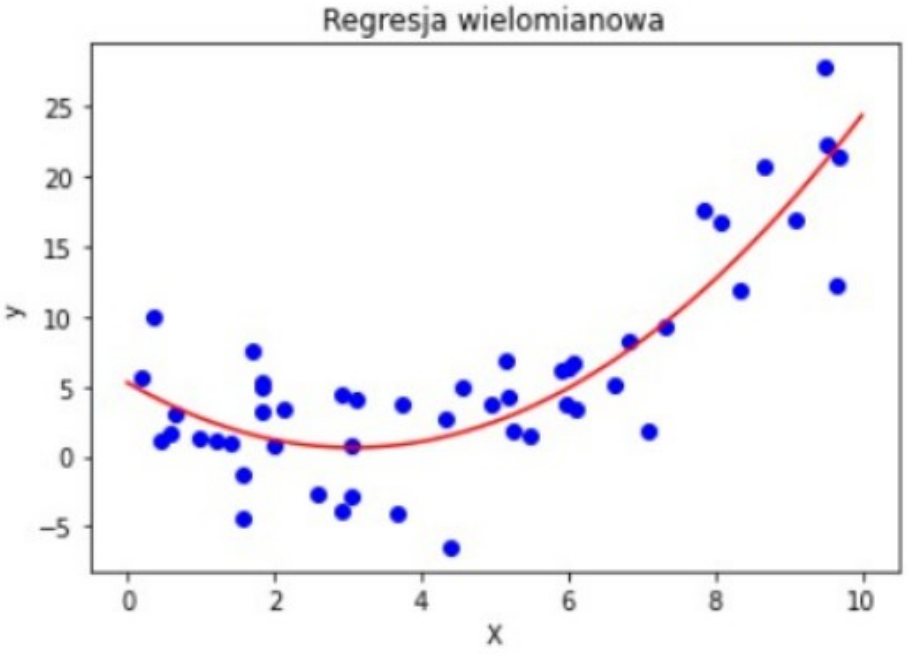
  $$Y = a_0 + a_1x + a_2x^2 + .. + a_nx^n$$

  gdzie

  Y - zmienna zależna

  x – zmienna niezależna

  a0, a1 – współczynniki wielomianu



Uczenie maszynowe

- **Regresja prosta** - jedna zmienna niezależna i jedna zależna
- **Regresja wieloraka** (wielozmiennowa) – jedna zmienna zależna, ale wiele zmiennych niezależnych.
- Uwaga: „*correlation does not imply causation*“ – korelacja nie oznacza przyczyny

https://www.buzzfeednews.com/article/kjh2110/the-10-most-bizarre-correlations

Uczenie maszynowe

- **Metody oceny**:
  - **Błąd średniokwadratowy** (Mean Square Error) – oblicza średnią wartość kwadratów różnic między rzeczywistymi wartościami docelowymi a przewidywanymi wartościami generowanymi przez model.
  - Przydatny do optymalizacji modelu

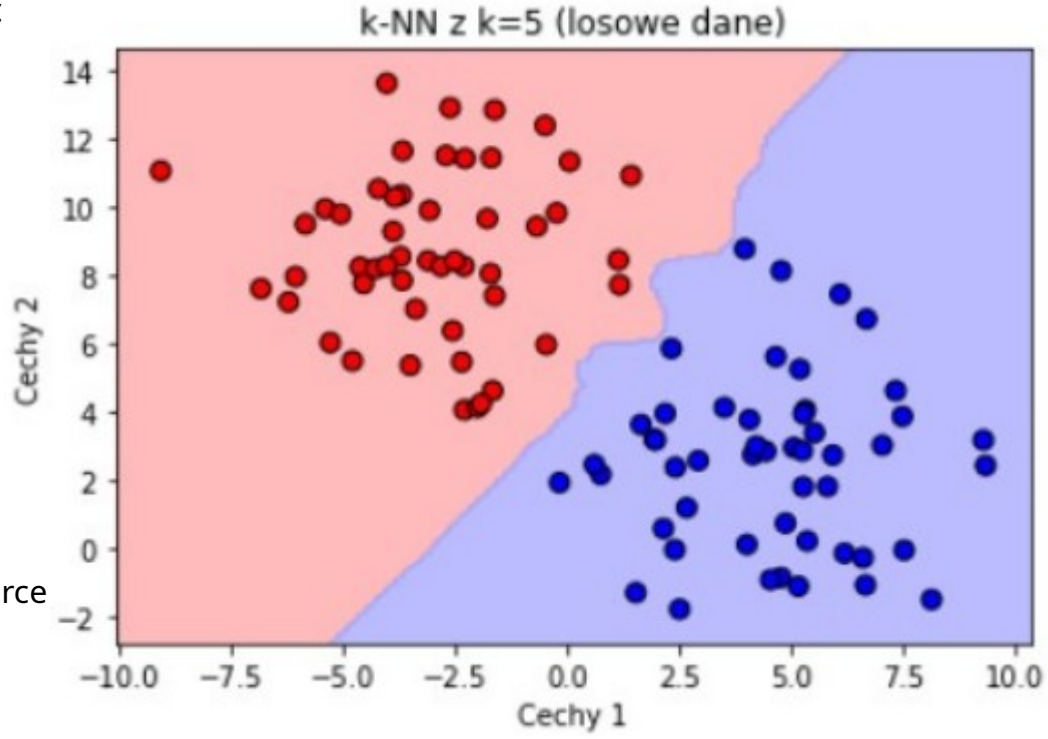
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_{\mathrm{true},i} - y_{\mathrm{pred},i})^2$$

  - **Pierwiastek błędu średniokwadratowego** (Root Mean Square Error) – przekształca wartość MSE z powrotem do tych samych jednostek, co dane wejściowe, dzięki czemu jest bardziej intuicyjny i łatwiejszy do zinterpretowania.
  - Przydatny do interpretacji oraz raportowania wyników

$$RMSE = \sqrt{\mathrm{MSE}}$$

Klasyfikacja

- **Metoda k-najbliższych sąsiadów** (*ang. k-Nearest Neighbors*) – algorytm używany zarówno do klasyfikacji jak i regresji.
- Klasyfikacja obiektu odbywa się na podstawie głosowania *k* najbliższych sąsiadów z zestawu treningowego.
- W przypadku problemów regresji, przewidywana wartość jest średnią wartości *k* najbliższych sąsiadów.
- Działanie algorytmu:
  - Wybór parametru k (liczba sąsiadów do uwzględnienia)
  - Obliczenie odległości między nowym obiektem a wszystkimi obiektami w zbiorze
  - Wybór k najbliższych sąsiadów
- Prosty w implementacji i zrozumieniu
- Stosowany do klasyfikacji wieloklasowej
- Optymalizacja k
  - Małe k – model elastyczny, ale podatny na szum
  - Wysokie k – model stabilny, ale możw tracić lokalne wzorce



Klasyfikacja

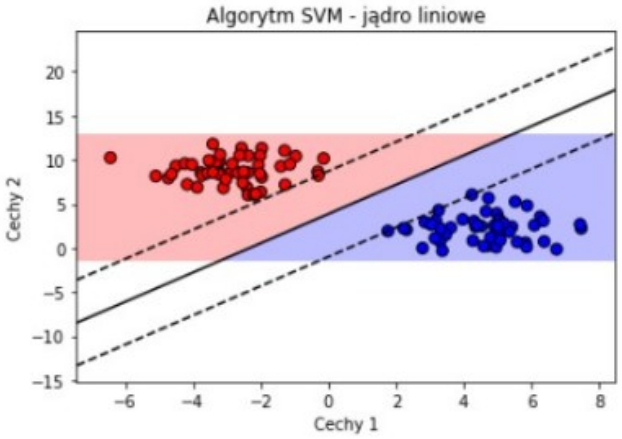
- **Maszyny wektorów nośnych** (ang. Support Vector Machines) – algorytm, który dąży do znalezienia hiperpłaszczyzny, która najlepiej rozdziela klasy w przestrzeni cech.

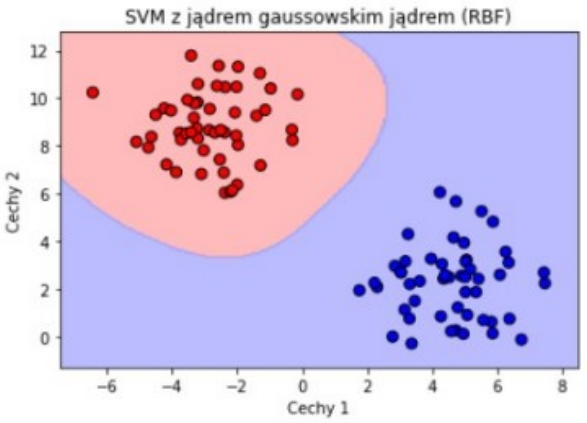
  Maksymalizuje margines między klasami.

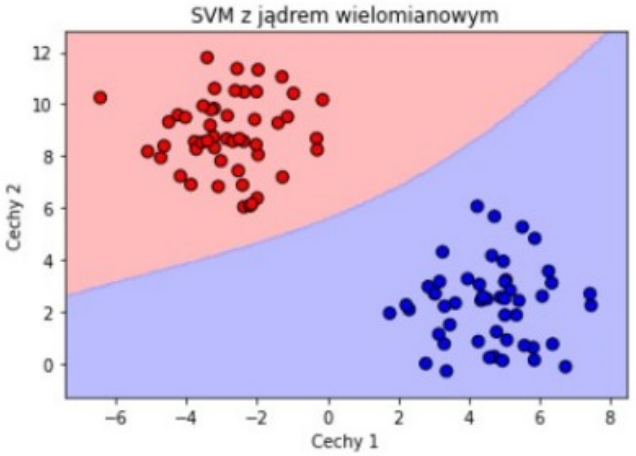
  Algorytm szczególnie skuteczny w klasyfikacji binarnej, gdy granica decyzyjna jest bardzo złożona.

  - **Hiperpłaszczyzna** – linia decyzyjna rozdzielająca klasy.
  - **Wektory nośne** – punkty danych najbliższe hiperpłaszczyźnie.
  - **Margines** – odległość między hiperpłaszczyzną a najbliższymi punktami.
  - **Funkcja jądra** – decyduje o sposobie rozdzielenia danych (liniowe, wielomianowe, radialne).
- Skuterczne w wielowymiarowych zestawach danych.

Klasyfikacja





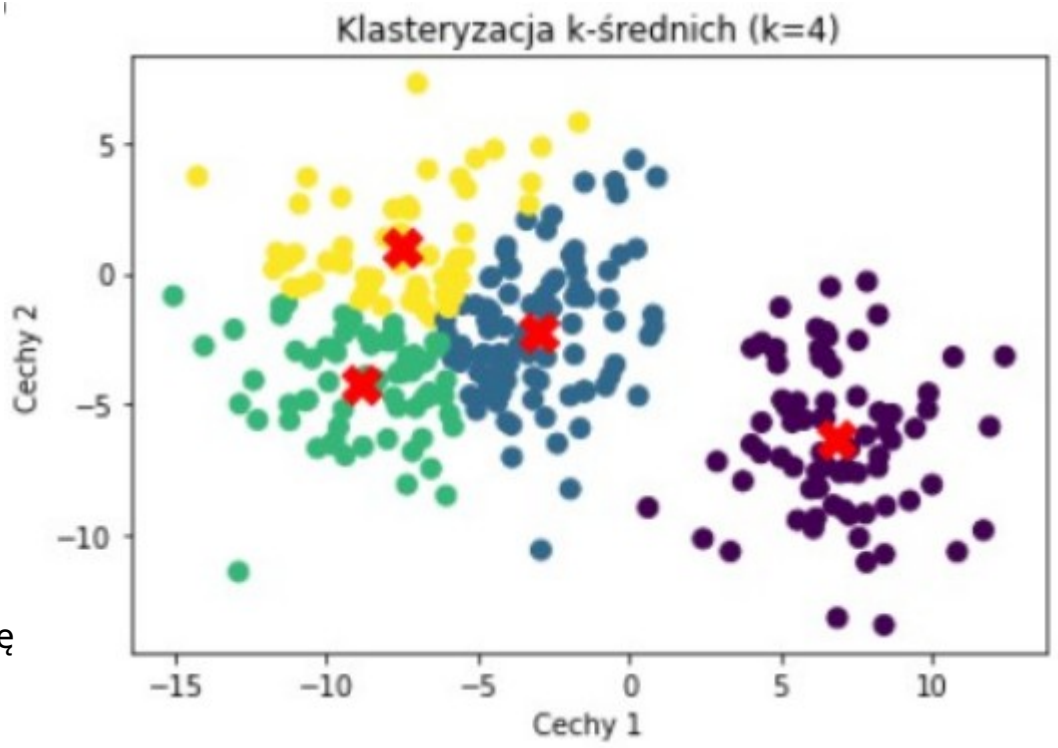


Uczenie maszynowe

- **Uczenie nienadzorowane** – metoda, w której algorytm analizuje nieoznakowane dane, próbując znaleźć ukryte struktury lub wzorce bez zdefiniowanych etykiet lub wyników.
- Nie wymaga ludzkiego nadzoru w procesie uczenia
- Główne zadania:
  - **Klasteryzacja** (grupowanie) – na podstawie podobnych cech
    - Algorytm k-średnich
    - Algorytm k-medoidów
  - **Redukcja wymiarowości** – zmniejszenie liczby cech przy zachowaniu jak największej ilości informacji
    - Analiza głównych składowych (PCA)
- Przydatne do odkrywania nieznanych wcześniej struktur i ukrytych cech w danych.
- Przykłady zastosowań
  - Segmentacja klientów na potrzeby marketingowe
  - Grupowanie artykułów w systemach rekomendacji

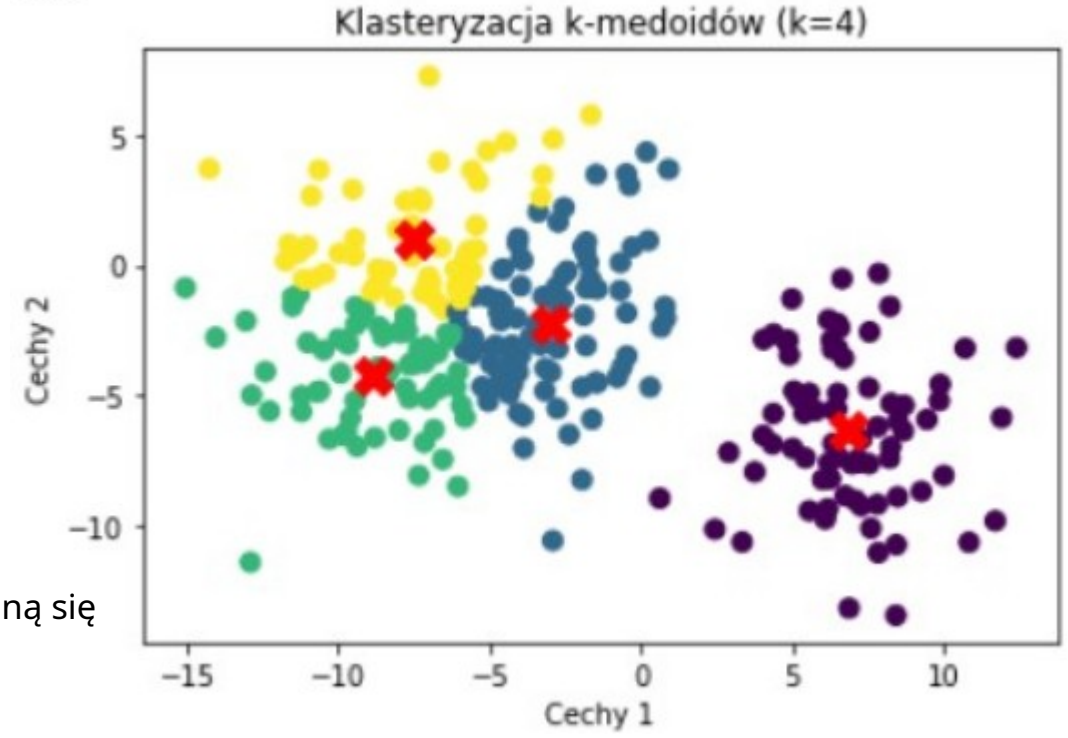
Klasteryzacja

- **Algorytm k-średnich** (*ang. k-means*) – rozdziela dane na k rozłącznych klastrów, gdzie każdy klaster jest reprezentowany przez jego **centroid** (średni punkt).
- Jest algorytmem Iteracyjnym
- Prosty i szybki, efektywny dla dużych zbiorów danych.
- Dobry dla dużych danych, jednak może mieć problemy z klastrami o nieregularnych kształtach
- Wrażliwy na wartości odstające
- Zasada działania:
  - Wybór k centroidów (np. Losowo)
  - Każdy punkt jest przypisywany do najbliższego centroidu
  - Centroidy są przeliczane jako średnia punktów w klastrze
  - Kroki 2 i 3 są powtarzane, aż centroidy przestaną się zmieniać lub osiągniemy maksymalną ilość iteracji



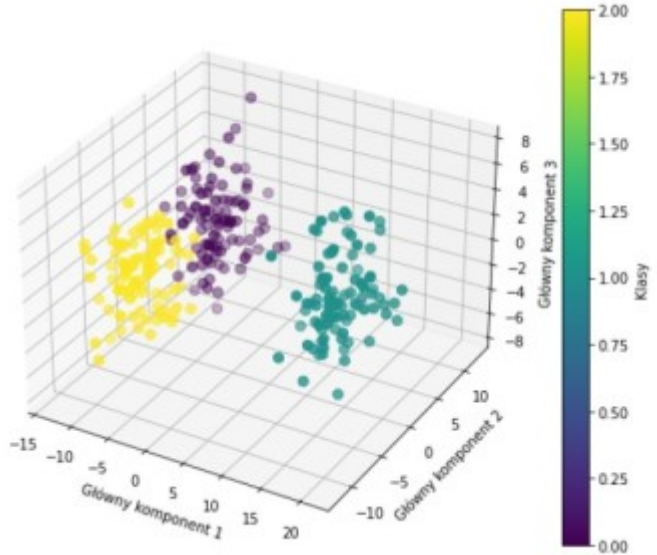
Klasteryzacja

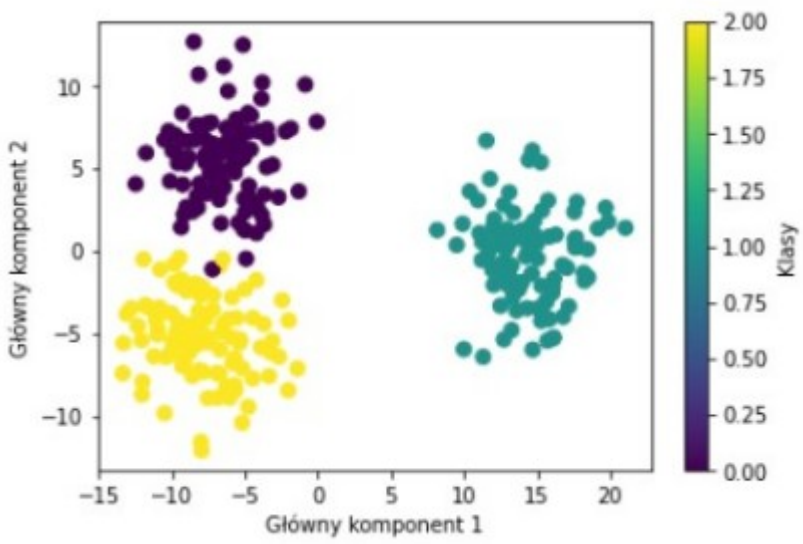
- **Algorytm k-medoidów** (*ang. k-medoids*) – alternatywny do k-średnich algorytm do rozdzielania danych na k rozłącznych klastrów.
- Różnica polega na wyborze **medoidów** (rzeczywisty punkt danych) zamiast centroidów (średnia arytmetyczna punktów)
- Odporniejszy na zbiór danych z wartościami odstającymi
- Zasada działania:
  - Losowy wybór k punktów (medoidów)
  - Każdy punkt jest przypisywany do najbliższego medoidu
  - Obliczenie kosztu dla wybranego medoidu
  - Obliczenie kosztu dla każdego punktu w klastrze
  - Jeżeli jakiś punkt ma niższy koszt, to staje się on medoidem
  - Kroki 2, 3, 4 i 5 są powtarzane, aż medoidy przestaną się zmieniać lub osiągniemy maksymalną ilość iteracji



Redukcja wymiarowości

- **Analiza Głównych Składowych** (*ang. principal component analysis*) – algorytm do redukcji wymiarowości danych przy jednoczesnym zachowaniu jak największej liczby informacji.
- Przekształca oryginalny zestaw danych na nowy zestaw nieskorelowanych zmiennych (głównych składowych)
- Wykorzystuje wektory własne (*ang. eigenvector*) oraz odpowiadające im wartości własne (*ang. eigenvalues*) macierzy kowariancji.
- Wektory własne określają kierunki głównych komponentów a wartości własne ich „ważność“.





Uczenie maszynowe

- Uczenie przez wzmacnianie (ang. Reinforcement learning) – metoda, w której agent uczy się podejmować decyzje poprzez interakcjie ze środowiskiem.
- Pojęcia:
  - Agent – jednostka podejmująca decyzje
  - Środowisko – otoczenie działania agenta
  - Stan – reprezentacja danej sytuacji w środowisku
  - Akcja – decyzja podejmowana przez agenta
  - Nagroda – sygnał ze środowiska informujący o jakości podjętej akcji
  - Polityka – strategia podejmowania decyzji przez agenta
- Proces uczenia:
  - Agent obserwuje stan środowiską
  - Na podstawie stanu, podejmuje działanie
  - Środowisko zmienia się pod wpływem działania
  - Agent otrzymuje nagrodę i uczy się maksymalizować (lub minimalizować) nagrodę

Uczenie maszynowe

- **Funkcja wartości** (*ang. value function*) – określa oczekiwaną sumę nagród dla danego stanu. Przykład: Gra planszowa, gdzie agent musi dotrzeć z punktu (S) do mety (M)

```
[S] [ ] [ ]
[ ] [X] [ ]
[ ] [ ] [M]
```

```
[0.8] [0.9] [0.95]
[0.7] [0.0] [0.99]
[0.6] [0.8] [1.00]
```

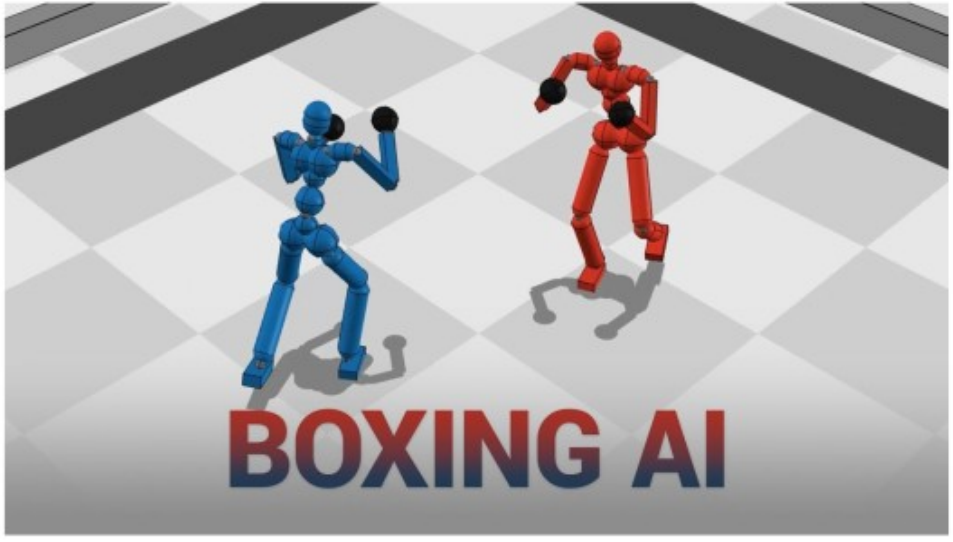
- **Funkcja polityki** (*ang. policy function*) – definiuje zachowanie agenta w środowisku, określa prawdopodobieństwo wyboru danej akcji w określonym stanie.
  - Deterministyczna – dla każdego stanu określana jest jedna akcja.
  - Stochastyczna – funkcja zwraca rozkład prawdopodobieństwa wykonania poszczególnej akcji w stanie s.
- Agent korzystając z polityki eksploruje środowisko, otrzymuje nagrody za swoje działania i uszy się optymalizować swoje decyzje, aby maksymalizować nagrodę.
- **Funkcja nagrody** – określa natychmiastową nagrodę za podjęcie akcji w danym stanie, innymi słowy: przypisuje wartość liczbową dla pary stan-akcja.

$$R(s, a, s')$$

gdzie s to obecny stan, a – wykonana akcja, s’ to nowy stan po wykonaniu akcji a.

Uczenie maszynowe

- Zastosowania uczenia przez wzmocnienie:
  - Robotyka i systemy sterowania.
  - Gry – od prostych planszówek po złożone gry strategiczne.
    (https://www.nature.com/articles/nature24270)
  - Optymalizacja strategii handlowych (uczenie na podstawie warunków rynkowych i danych historycznych).
  - Pojazdy autonomiczne – podejmowanie decyzji w złożonych warunkach drogowych.
- Uczenie przez wzmocnienie pozwala **uczyć się** i **optymalizować** zachowanie w dynamicznych i nieprzewidywalnych środowiskach (metoda prób i błędów).
- Umożliwia ciągłe **doskonalenie się** agenta w miarę gromadzenia doświadczeń.



Uczenie maszynowe

- **Uczenie transferowe** (*ang. transfer learning*) – technika polegająca na wykorzystaniu zdobytej wiedzy podczas rozwiązywania jednego problemu do rozwiązania innego.
- Umożliwia dotrenowanie modelu do nowych zadań (np. rozpoznawania nowych twarzy).
  - Wykorzystanie wstępnie wytrenowanego modelu.
  - Fine-tuning – proces dostrajania wstępnie wytrenowanego modelu do nowego, ale powiązanego zadania.
  - Dostrojenie, dodanie (lub nawet wymiana) warstwy klasyfikacyjnej przy zachowaniu warstw konwolucyjnych (np. w przypadku systemu do rozpoznawania twarzy).
  - Niższe warstwy konwolucyjne wychwytują uniwersalne cechy a wyższe warstwy oraz część klasyfikacyjna są specyficzne dla danego zadania.
- Zalety – skrócenie czasu treninigu (powszechna dostępność różnych modeli)
- Możliwość uzyskania dobrych wyników nawet przy małym zbiorze danych
- Zastosowania: rozpoznanie obrazu, klasyfikacja obiektów, moderacja treści, przetwarzanie języka naturalnego.

Trenowanie modelu

- Anatomia treningu nadzorowanego
  - Zbieranie i przygotowanie **danych**
  - **Czyszczenie i normalizacja danych**
  - **Inicjalizacja** – start z losowymi wagami
  - **Predykcja** – model zwraca predykcję odpowiedzi dla danych treningowych
  - **Funkcja kosztu** (loss) – pomiar jak bardzo model się myli
  - **Aktualizacja parametrów** – korekta wag
  - **Powtórka** – tak długo, aż błąd przestanie maleć
- Podstawowe pojęcia:
  - Epoka (epoch) – jedno pełne przejście przez cały zbiór treningowy
  - Wsad (batch) – porcja danych, która jest przetwarzana między aktualizacją wag

Przygotowanie danych

- Czyszczenie danych:
  - Usuwanie braków (lub uśrednianie, np. średnia, mediana, moda)
  - Duplikaty i wartości odstające (outliers)
- Transformacja:
  - Normalizacja / standaryzacja
  - Kodowanie kategorii (one-hot, labels)
- Inżynieria cech (feature engineering):
  - Tworzenie nowych zmiennych z istniejących (np. BMI z wagi i wzrostu)
  - Często może mieć większy wpływ na wynik niż wybór modelu

Problemy z danymi

- Dane niezbalansowane
  - 99% zdrowych, 1% chorych
  - Sposoby zapobiegania
- Bias w danych
  - Model się uczy uprzedzeń a nie faktów
  - Sposoby zapobiegania
- Drift danych
  - Świat się zmienia, model pozostaje w niezaktualizowanym świecie
  - Sposoby zapobiegania: uczenie przyrostowe

Hiperparametry

- Parametry – to czego uczy się model
- Hiperparametry – sposób ustawienia modelu przed treningiem
- Przykady:
  - Learning rate
  - Liczba drzew w lesie losowym
  - Głębokość drzewa
  - Wartość k w k-NN
  - Wielkość wsadu
- Problem: nie ma uniwersalnej wartości hiperparametrów, wszystko zależy od danych

Dostęp do prezentacji

https://git.michalkasprowicz.com/administrator/Proces_uczenia_SI