softserve

Proces budowy modelu: od przygotowania danych do trenowania

Natalia Cheba

Senior Data Scientist

softserve

O mnie

Doświadczenie:

Senior Data Scientist z ponad pięcioletnim doświadczeniem w projektowaniu i wdrażaniu rozwiązań opartych na sztucznej inteligencji w różnych branżach, m.in. retail, insurance, energy i manufacturing. Główne obszary kompetencji to machine learning, deep learning, prognozowanie szeregów czasowych oraz przetwarzanie języka naturalnego, ze szczególnym naciskiem na zastosowania biznesowe. Realizacja projektów end-to-end – od przygotowania danych i budowy modeli po ich wdrożenie – we współpracy z zespołami biznesowymi i technicznymi.

Natalia Cheba

Edukacja:

- **MSc Data Science (Politechnika Śląska)**
- **BSc Business Analytics (Politechnika Śląska)**
- **Engineering in Computer Science (Politechnika Śląska)**

Linkedin profile

softserve

Etapy tworzenia modelu

Jak wygląda proces tworzenia modelu?

Pozyskiwanie danych

5%

- Zbieranie danych z różnych źródeł, takich jak bazy danych, API, systemy transakcyjne czy zbiory zewnętrzne.

Data Engineering

5%

- Projektowanie i utrzymanie potoków danych (data pipelines), których celem jest przekształcenie surowych danych w ustrukturyzowaną, spójną i skalowalną postać, gotową do dalszej analizy i modelowania.

Eksploracyjna analiza danych

15%

- Analiza danych w celu zrozumienia ich struktury, zależności pomiędzy zmiennymi oraz identyfikacji potencjalnych problemów wpływających na jakość modelu.

Data Preprocessing

15%

- Czyszczenie danych, obsługa braków danych, usuwanie obserwacji odstających, kodowanie zmiennych kategorycznych oraz normalizacja lub standaryzacja danych numerycznych.

softserve

Etapy tworzenia modelu

Jak wygląda proces tworzenia modelu?

Feature Engineering

30%

- Tworzenie, selekcja i transformacja cech w celu zwiększenia zdolności predykcyjnej modelu.

Wybór modelu

15%

- Dobór odpowiedniego algorytmu w zależności od rodzaju problemu, charakterystyki danych oraz przyjętych kryteriów oceny.

Trenowanie i ocena modelu

10%

- Trenowanie modelu na danych historycznych oraz optymalizacja jego parametrów. Ocena jakości modelu z wykorzystaniem odpowiednich metryk oraz technik walidacyjnych.

Monitorowanie i utrzymanie

5%

- Monitorowanie jakości predykcji w czasie oraz ponowne trenowanie modelu w przypadku zmian danych lub zjawiska *data drift*.

softserve

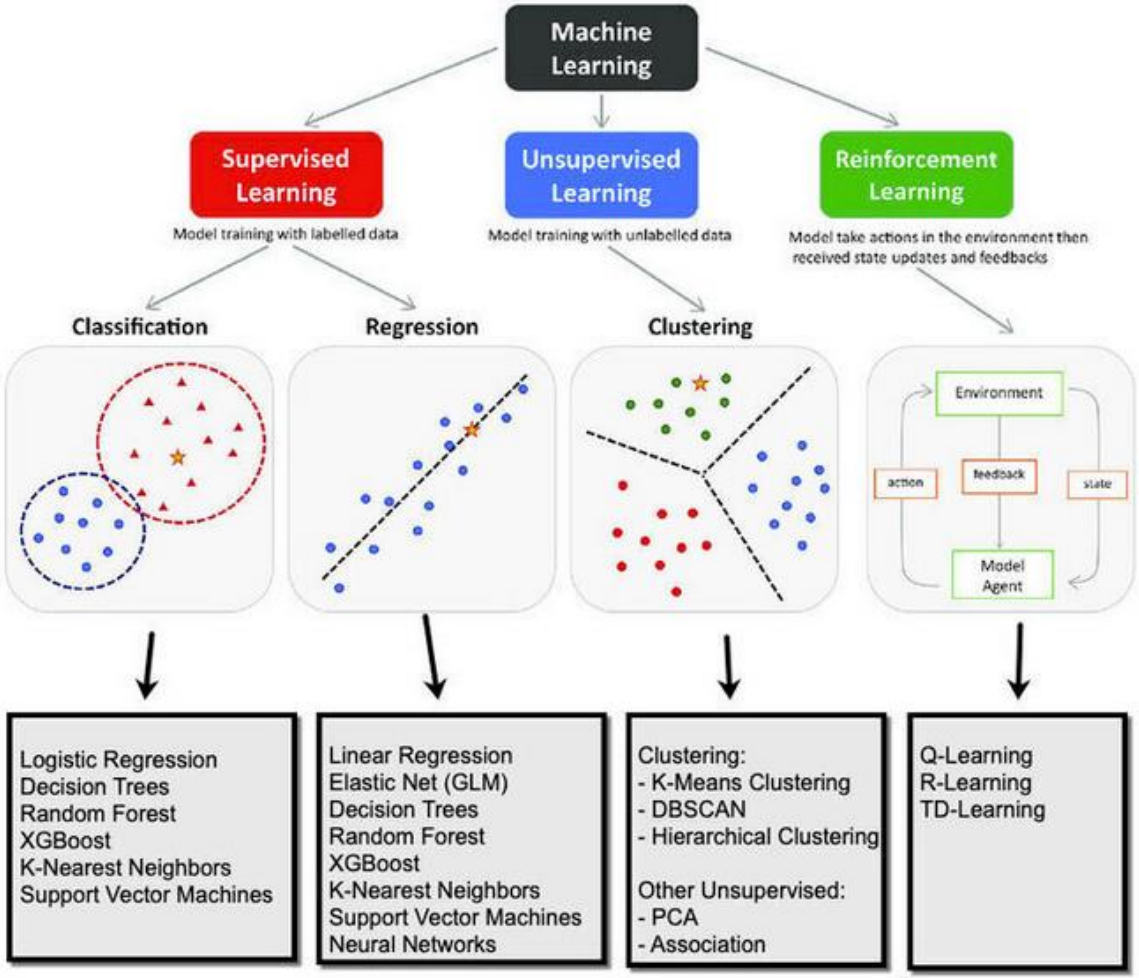
Lekcja 1: Poznaj Swój Problem

Zanim zaczniemy budować model, musimy dokładnie zrozumieć **cel projektu** oraz **dane**, które będą go wspierać.

W tym etapie określamy:

- **Co chcemy przewidzieć lub zoptymalizować** (np. sprzedaż, ryzyko, churn)
- **Jaki jest kontekst biznesowy / domenowy – czy jest to branża Heathcare/Automotive?**
- **Jakie dane są dostępne i czy są wystarczające?**

Dopiero po jasnym zdefiniowaniu problemu możemy przejść do przygotowania danych i budowy modelu.



softserve

Lekcja 1: Poznaj Swój Problem

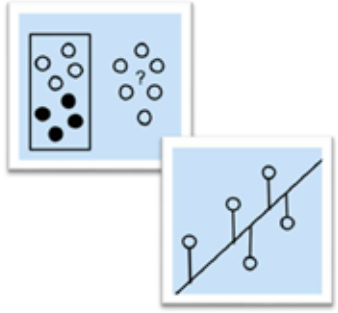
Uczenie nadzorowane:

Celem tych modeli jest przewidywanie nieznanych wartości Y (zmiennej docelowej) na podstawie znanych wartości X (cech).

Model jest trenowany na danych treningowych, gdzie wartości **Y są znane** (tzw. dane oznaczone). Dlatego nazywa się to **uczeniem nadzorowanym** – podczas tworzenia modelu wiemy, kiedy się myli.

Dwie klasy modeli:

- **Klasyfikacja** – przewidywanie zmiennej kategorycznej
- **Regresja** – przewidywanie zmiennej ciągłej



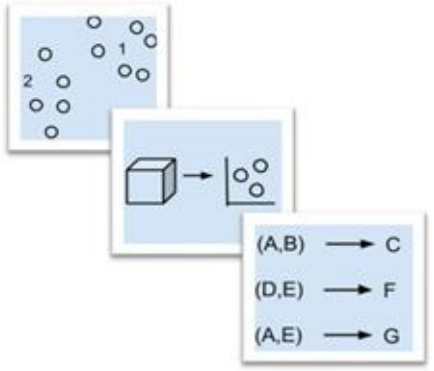
Uczenie nienadzorowane:

Celem tych modeli jest znalezienie ogólnych wzorców w znanych wartościach X, bez zmiennej docelowej.

Model jest trenowany na danych **bez wyróżnionej zmiennej Y** (dane nieoznaczone). Dlatego nazywa się to **uczeniem nienadzorowanym** – nie wiemy, czy model jest poprawny, czy nie.

Najczęściej stosowane klasy modeli:

- **Klasteryzacja** – identyfikacja względnie jednorodnych grup, np. segmentacja klientów
- **Redukcja wymiarowości –** zmniejszanie liczby zmiennych przy minimalnych stratach informacji
- **Reguły asocjacyjne** – wyszukiwanie zależności między zmiennymi kategorycznymi, np. analiza koszyka zakupowego (np. częste występowanie piwa i chipsów).



softserve

Lekcja 2: Poznaj Swoje Dane

Zanim przejdziesz do modelowania, upewnij się, że **rozumiesz każdą zmienną** i że **ma ona sens w kontekście problemu**.

W tym kroku pytamy:

- Czy dana zmienna rzeczywiście wpływa na wynik?
- Czy jest poprawnie zdefiniowana i mierzona?
- Czy nie wprowadza „przecieków informacji” (data leakage)?
- Czy jest wiarygodna i dostępna w przyszłych danych?

Dobór właściwych zmiennych to podstawa dobrego modelu - bez tego nawet najlepszy algorytm nie da wartościowych wyników.

Korelacja a związek przyczynowy

⚠ **Correlation =/= causation**

Przykład lodów i ataku rekina

softserve

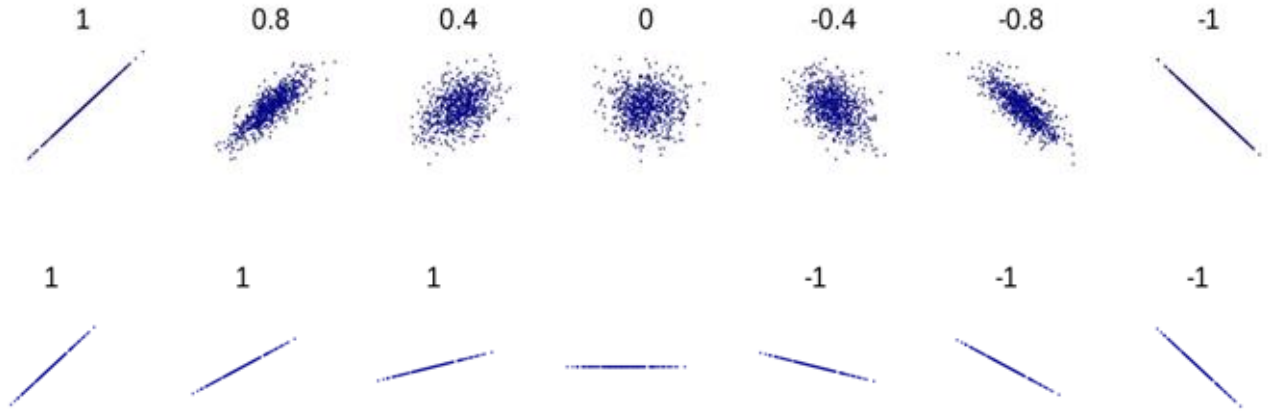
Lekcja 2: Poznaj Swoje Dane

Analiza korelacji – sposoby oceny

- **wartość chi-kwadrat** i miar na niej opartych (Yule'a, V – Cramera, T – Czuprowa)
- **współczynnik korelacji liniowej Pearsona**
- **współczynnik korelacji rang Spearmana**

Do oceny zależności korelacyjnej można dokonać oceny w oparciu o umowną ocenę siły zależności:

- **< 0.3** - korelacja jest słaba
- **0.3 > x > 0.5** - korelacja umiarkowana
- **0.5 < x** - korelacja silna



Nawet wysoki współczynnik korelacji nie oznacza, że zmienna X wpływa na zmienną Y.

Korelacja pokazuje jedynie **zależność i kierunek związku**, ale nie mówi nic o **nachyleniu (siła wpływu)** ani o przyczynowości.

Garbage in garbage out.

Jakość modelu **nigdy nie będzie lepsza niż jakość danych**, które do niego trafiają. Nawet najbardziej zaawansowany algorytm nie uratuje źle przygotowanych danych - dlatego preprocessing jest kluczowym etapem budowy modelu.

softserve

Lekcja 3: Data Quality Issues

Zła jakość danych prowadzi do:

- błędnych wniosków,
- niestabilnych modeli,
- „dobrych” wyników tylko na papierze

Typ problemu	Przykład
Brakujące dane	Brak 30% wartości w kolumnie
Niepoprawny format	Data zapisana jako YYYY-MM-DD i DD-MM-YYYY
Błędy literowe	Warszwa, Warzsawa
Duplikaty	łosoś, losoś, losoś jako różne produkty
Przestarzałe dane	Nieaktualna lokalizacja sklepu
Nieprawidłowe wartości	203% zawartości białka
Nieadekwatne dane	kolor = Golden Retriever

Lepiej poświęcić czas na czyszczenie i zrozumienie danych niż na strojenie modelu, który uczy się błędów.

Feature engineering ma większy wpływ niż wybór modelu.

Często sprawdzenie wielu różnych modeli nie przynosi tak dużej poprawy jak **odpowiednie przygotowanie cech**. Dobrze zaprojektowane cechy potrafią znacząco zwiększyć jakość predykcji, nawet przy prostszych algorytmach.

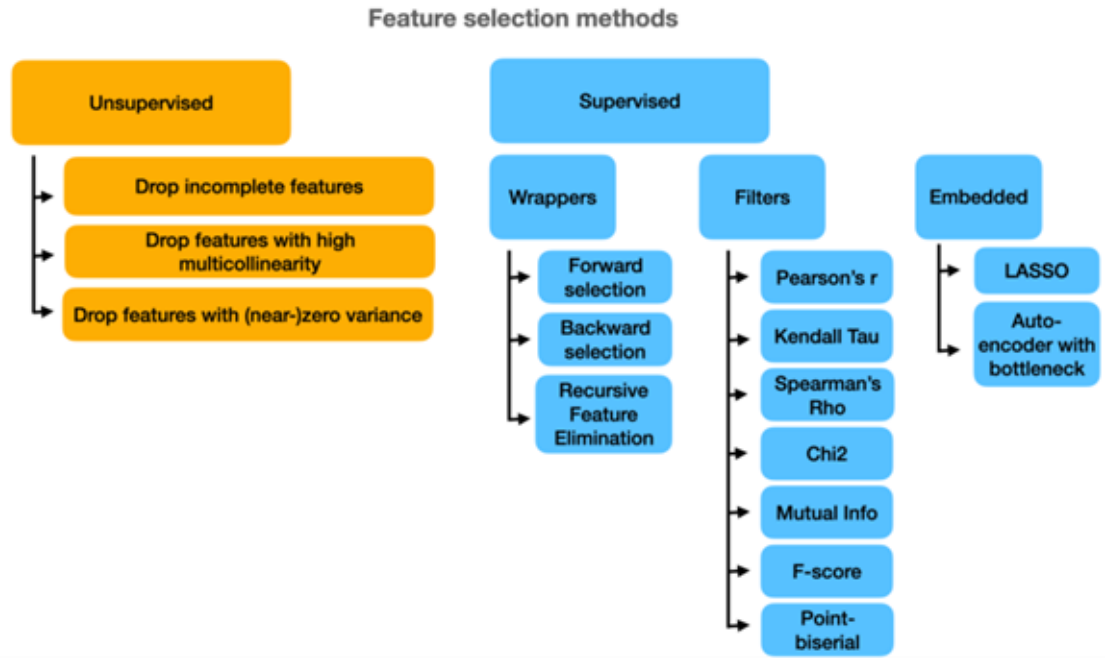
softserve

Lekcja 4: Wybór zmiennych – jak je wybrać?

Wybór cech **(feature selection)** zależy od problemu i danych, ale zawsze warto kierować się **kontekstem biznesowym**.
Dobry wybór zmiennych wpływa na jakość modelu, jego stabilność i interpretowalność.

Praktyczne wskazówki:

- **Zacznij od biznesu**: wybierz zmienne, które mają sens w kontekście celu modelu.
- **Sprawdź korelacje**: usuń lub połącz zmienne silnie skorelowane (redukcja redundancji).
- **Testuj różne kombinacje**: sprawdź, które zestawy cech dają najlepsze wyniki (np. poprzez walidację).
- **Wykorzystaj wcześniejsze ustalenia**: np. wyniki EDA, istotność cech, wyniki modeli.
- **Wybierz cechy stabilne w czasie**: unikaj zmiennych, które będą niedostępne w przyszłości.
- **Uważaj na zmienne o dużej liczbie braków**: mogą wprowadzać szum i błędy.
- **Zastanów się nad cechami pochodnymi**: tworzenie nowych cech z kombinacji istniejących (np. ratio, różnice, logarytmy).



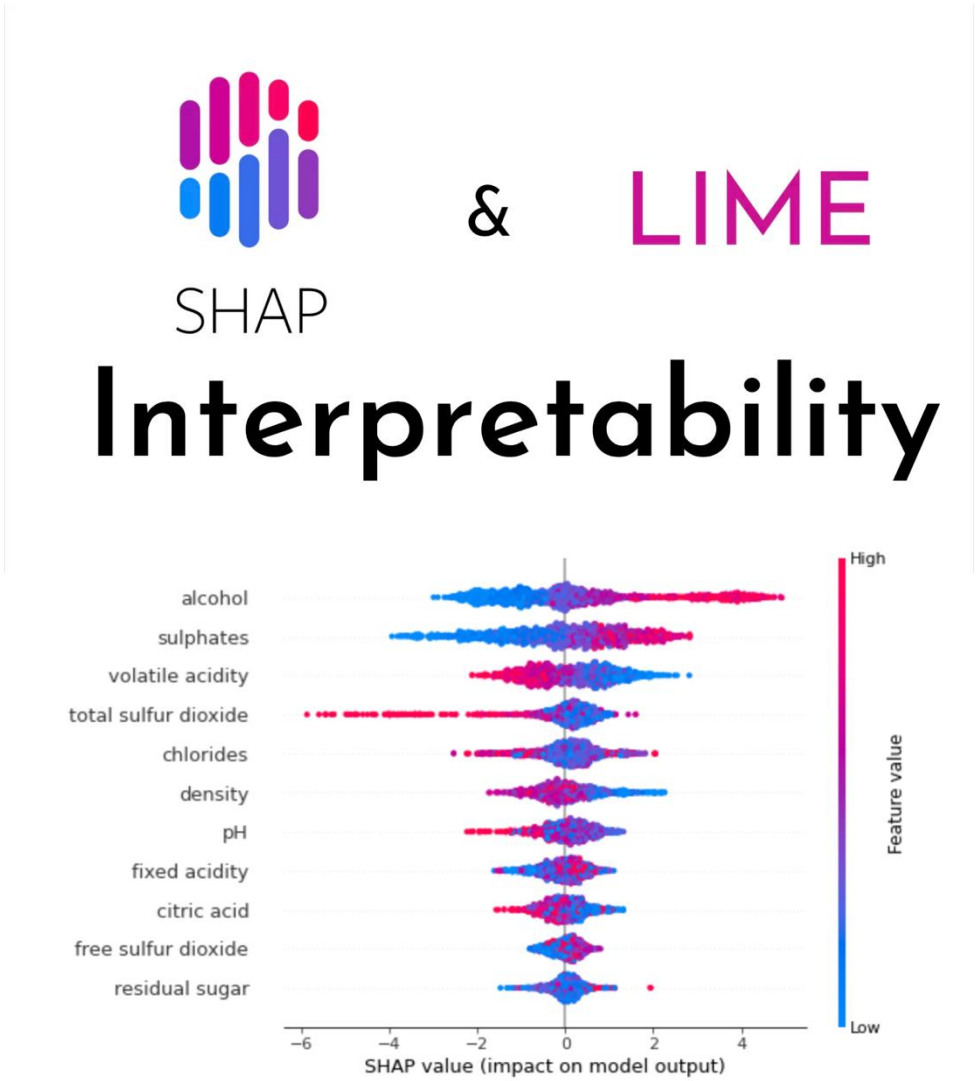
softserve

Lekcja 4: Wybór zmiennych – jak je wybrać?

Wybór cech **(feature selection)** zależy od problemu i danych, ale zawsze warto kierować się **kontekstem biznesowym**.
Dobry wybór zmiennych wpływa na jakość modelu, jego stabilność i interpretowalność.

SHAP vs LIME – A Practical Comparison

Feature	SHAP	LIME
Based On	Game Theory (Shapley values)	Local surrogate models
Scope	Global + Local	Local only
Output Type	Exact Shapley Attribution	Linear Approximation
Accuracy of Logic	High (theoretically sound)	Medium (approximation-based)
Model Compatibility	Tree-based models optimized	Model-agnostic (text/image/tabular
Speed	Slower (resource-intensive)	Faster (especially on prototypes)
Best Use Case	Regulatory, fairness, precision	Quick insights, early prototyping



softserve

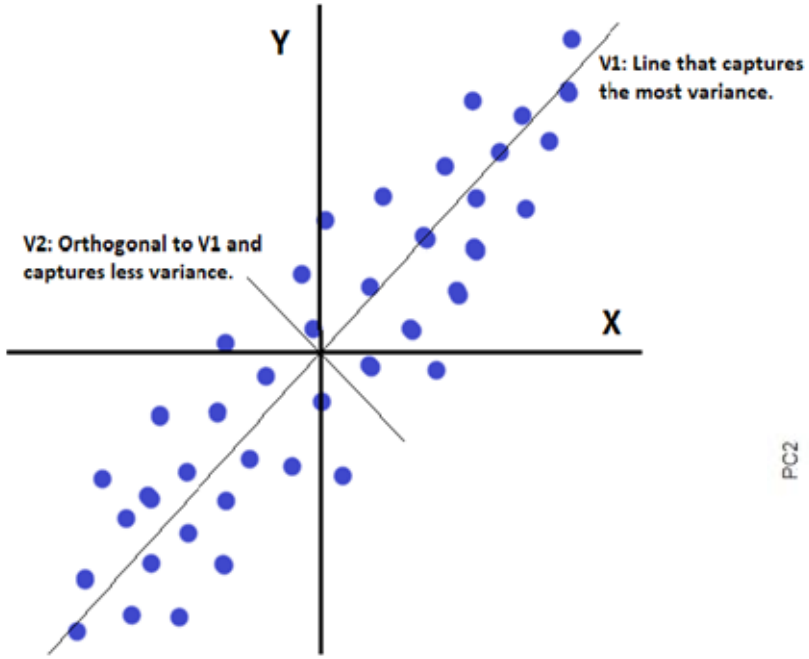
Lekcja 5: Redukcja zmiennych jako insight

Principal Component Analysis (PCA)

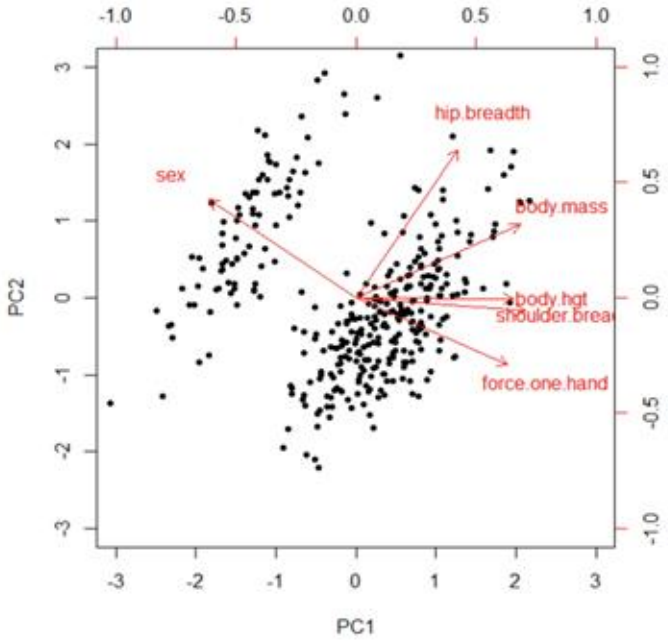
- służy do redukcji wymiaru danych, czyli uzyskania podobnej informacji przy mniejszej liczbie zmiennych
- daje też możliwość **wizualizacji związków między zmiennymi**
- może być wykorzystane jako **wejście do modelu predykcyjnego**.

Tworzy nowe zmienne (składowe główne), które wyznaczają kierunki największej zmienności danych. Często kilka pierwszych składowych wyjaśnia większość zmienności.

Jak interpretować wynik?
- pojedyncza zmienna z oryginalnego zbioru wyjaśnia 1/N wariancji
- pierwsza składowa może wyjaśniać np. 80% wariancji, druga 15%, itd. wtedy 2–3 składowe mogą wystarczyć do opisania większości informacji



Wyznaczenie składowych głównych sprowadza się do analizy **macierzy korelacji**. Z tej macierzy oblicza się **wartości własne** i **wektory własne**.

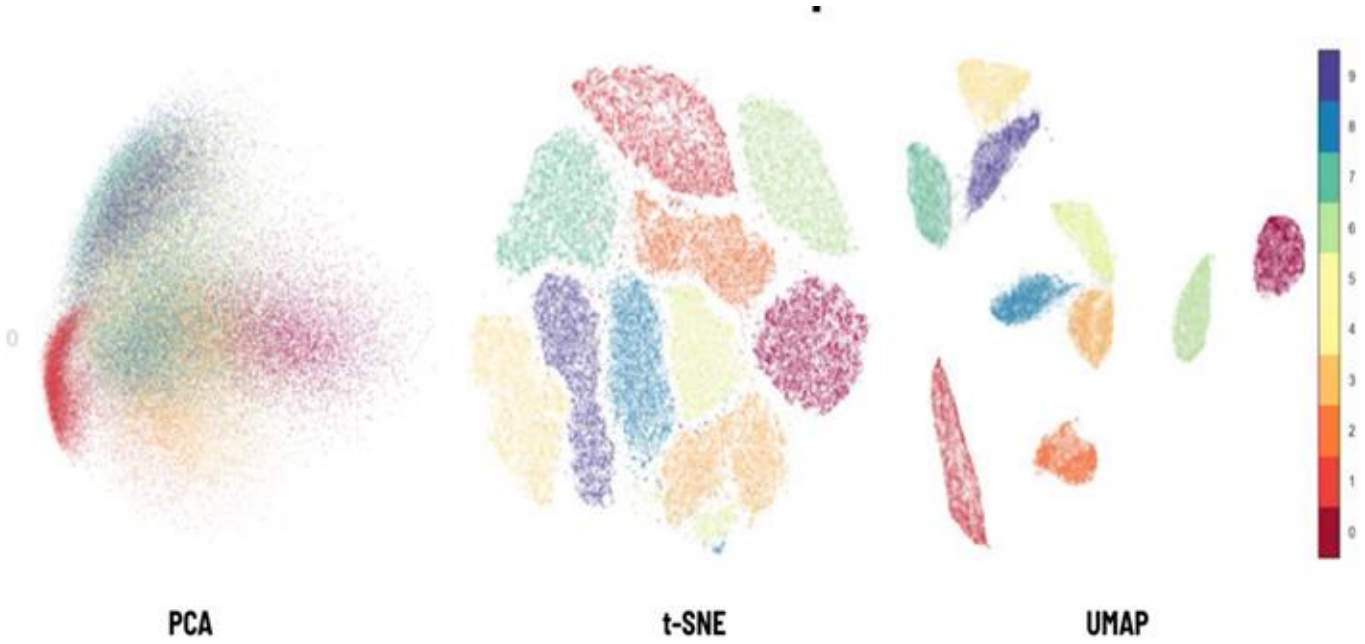


softserve

Lekcja 5: Redukcja zmiennych jako insight

PCA | T-SNE | UMAP

- **PCA:** deterministyczne, łatwe do interpretacji, opiera się na redukcji liniowej.
- **t-SNE:** stochastyczne, koncentruje się na strukturach lokalnych, doskonałe do wizualizacji danych o złożonych zależnościach.
- **UMAP:** stochastyczne, zachowuje lokalne i globalne struktury, działa szybciej i jest bardziej efektywne dla dużych zbiorów danych.

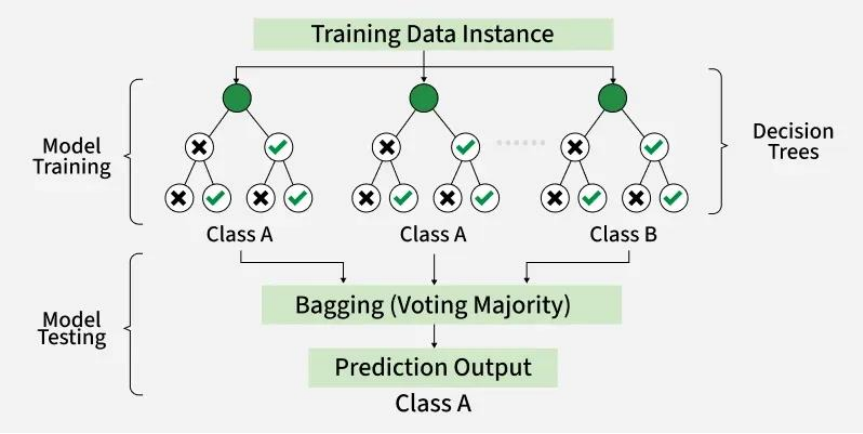


softserve

Lekcja 6: XGBoost vs Random Forest - co wybrać?

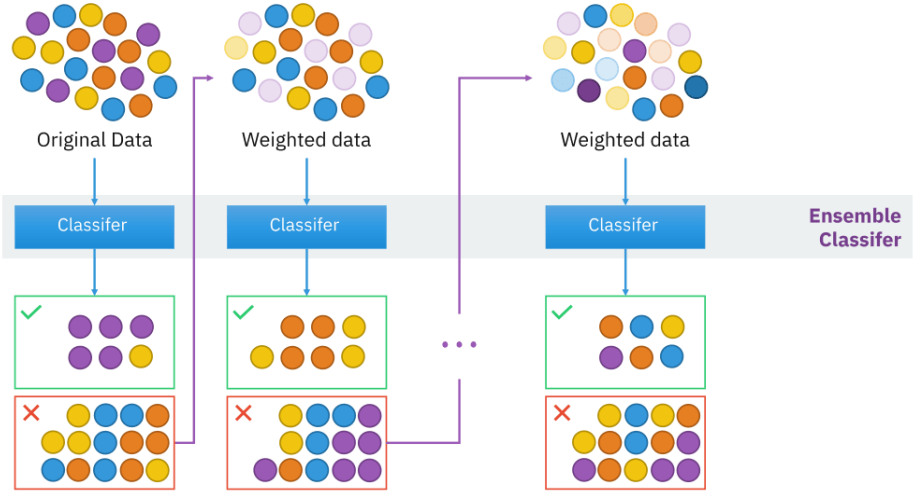
Random Forest

Random Forest to metoda uczenia zespołowego (**ensemble learning**), która łączy **wiele drzew decyzyjnych**. Każde drzewo jest trenowane na innej próbce danych (**bootstrap sample**), a ostateczna decyzja powstaje przez głosowanie wszystkich drzew. Aby zwiększyć różnorodność drzew, w Random Forest przy każdym rozgałęzieniu **losowo wybierany jest podzbiór cech** (predictors).



XGBoost

XGBoost to jedna z najpopularniejszych implementacji algorytmu **boosting.** W boosting kolejno buduje się wiele modeli, gdzie każdy kolejny **model uczy się na błędach poprzednich,** przydzielając większe wagi obserwacjom, które zostały źle przewidziane.

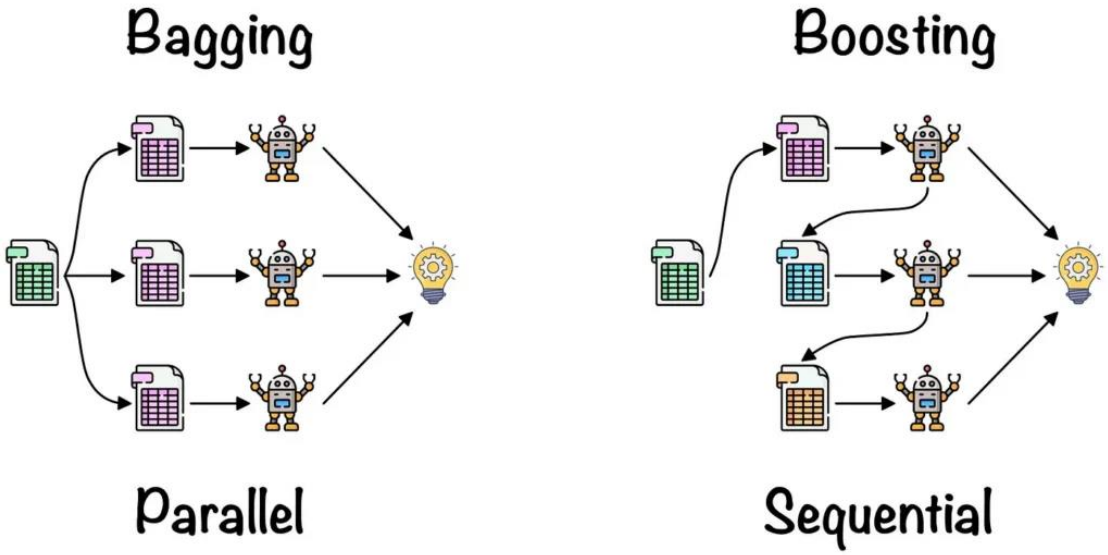


softserve

Lekcja 6: XGBoost vs Random Forest - co wybrać?

Który model będzie najlepszy?

Oba algorytmy to **dobry punkt startowy** w projektach ML.
Random Forest jest łatwiejszy i szybszy, XGBoost często daje lepsze wyniki, ale jest bardziej wymagający.



Jeśli początkowy model wykazuje niską dokładność (np. 50%), może to wskazywać, że problem jest z natury trudny i można zaoszczędzić czas poprzez wczesną identyfikację niewykonalnych rozwiązań.

softserve

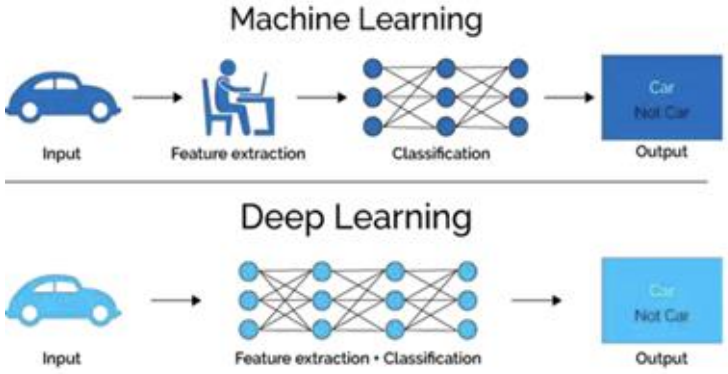
Lekcja 7: Kiedy warto sięgnąć po sieci neuronowe?

Sieci neuronowe

Sieć neuronowa - transformuje dane wejściowe poprzez obliczenie sumy ważonej wejść i zastosowanie nieliniowej funkcji do tej transformacji w celu uzyskania stanu pośredniego.

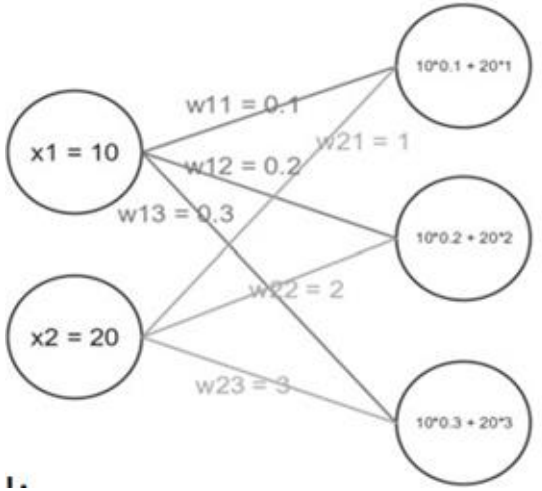
Sieci neuronowe są potężnym narzędziem i często dają bardzo dobre wyniki, ale ich trenowanie i strojenie jest bardziej czasochłonne niż w przypadku klasycznych modeli.

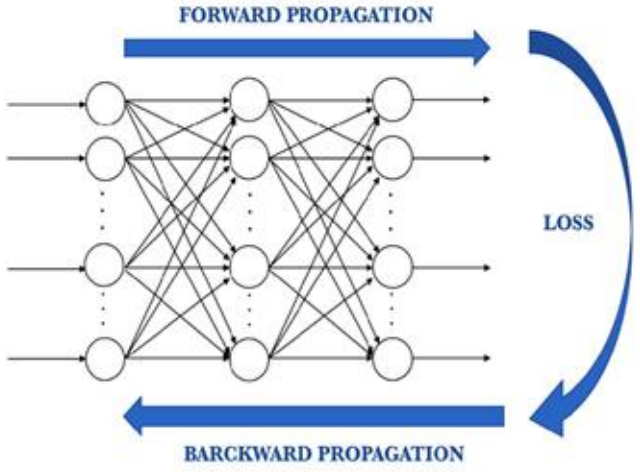
W praktyce sieci neuronowe są szczególnie popularne w zadaniach takich jak **computer vision.** W danych tabularycznych częściej stosuje się **prostsze modele ML**, które są szybsze, łatwiejsze w interpretacji i często równie skuteczne.



Końcowa suma czyli z to przesunięcie b i suma wszystkich **wag** pomnożona przez **wartości neuronów**.

$$z = b + \sum_i w_i x_i$$





softserve

Lekcja 8: **Podział Zbioru Ma Znaczenie**

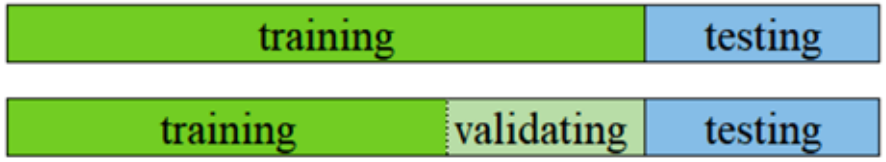
Podział zbioru danych

Dostępny zbiór danych jest losowo dzielony na części:

- **Zbiór treningowy** (50–80%) – służy do **trenowania modelu**
- **Zbiór walidacyjny** – służy do **doboru hiperparametrów**
- **Zbiór testowy** – służy do **ostatecznej oceny.**

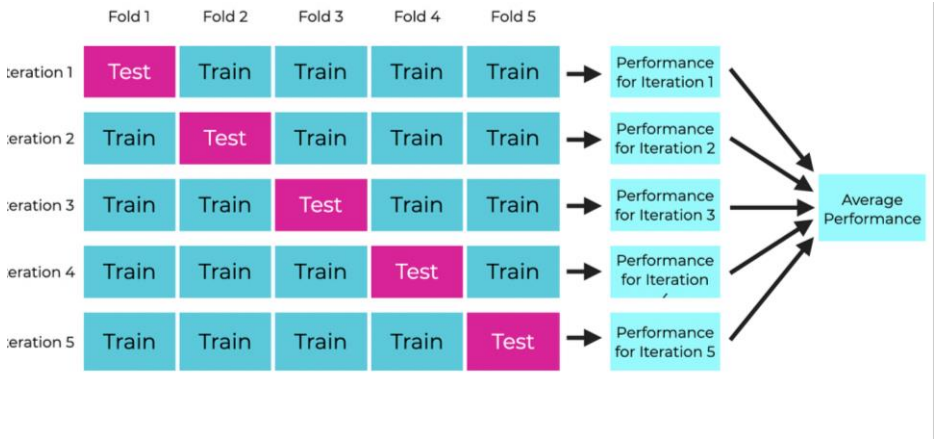
Metoda holdout

- Model jest trenowany na zbiorze treningowym, a następnie oceniany na zbiorze testowym.
- Pojedynczy podział może dawać niepewne wyniki.



Cross-Validation

Cross-walidacja daje bardziej **wiarygodną ocenę modelu**, ponieważ testuje go na wielu różnych podziałach danych, a nie tylko na jednym losowym zbiorze testowym.



softserve

Lekcja 9: **Ewaluacja – czy mój model działa?**

Ocena modeli uczenia nadzorowanego

Dokładność modeli uczenia nadzorowanego można oceniać za pomocą różnych miar. W zależności od typu zadania stosuje się inne miary:

- **dla klasyfikacji –** najczęściej używana miara to dokładność (**accuracy**), czyli stosunek liczby poprawnych przewidywań do liczby wszystkich przewidywań
- **dla regresji -** stosuje się inne miary (np. MSE, MAE, R^2)

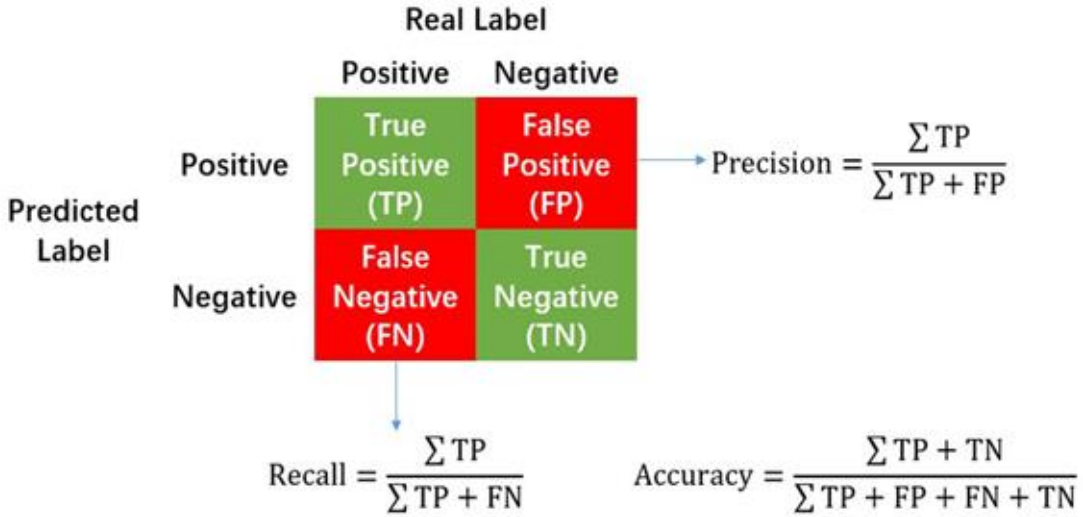
n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

N = 1000	Pred. NO	Pred. YES
Actual NO	900	0
Actual Yes	100	0

Accuracy = 900 / 1000 = **90%**

W wielu zastosowaniach (np. medycyna) **znacznie ważniejsze jest, aby wykryć chorobę, nawet jeśli czasem pojawi się fałszywy alarm**, niż ryzykować sytuację, w której choroba zostanie przeoczona.

Confusion Matrix



Nigdy dla niezbalansowanego zbioru nie wybieraj miary accuracy.

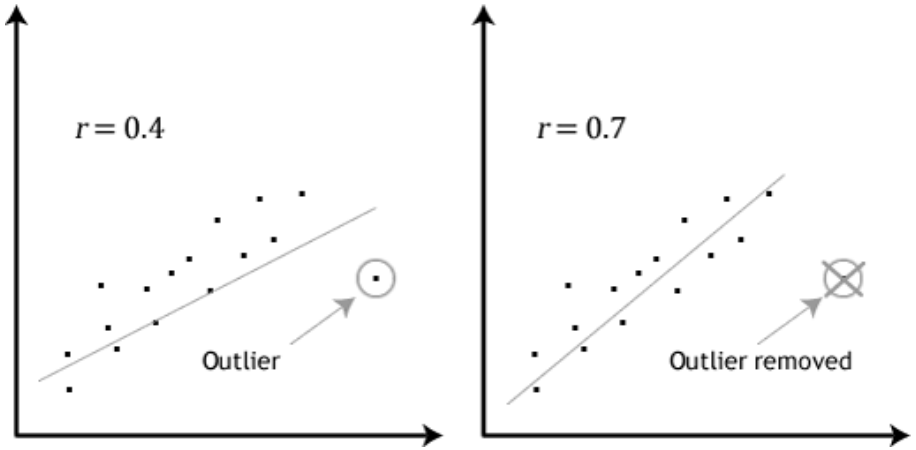
softserve

Lekcja 9: **Ewaluacja – czy mój model działa?**

Ocena modeli uczenia nadzorowanego

Dokładność modeli uczenia nadzorowanego można oceniać za pomocą różnych miar. W zależności od typu zadania stosuje się inne miary:

- **dla klasyfikacji –** najczęściej używana miara to dokładność (**accuracy**), czyli stosunek liczby poprawnych przewidywań do liczby wszystkich przewidywań
- **dla regresji -** stosuje się inne miary (np. MSE, MAE, R^2)



Regresja jest zdecydowanie trudniejsza niż klasyfikacja

Żadna miara nie mówi całej prawdy. Najczęściej stosuje się:

- **MAE (Mean Absolute Error) -** średni błąd bezwzględny (łatwy do interpretacji)
- **MSE / RMSE -** karze większe błędy bardziej niż MAE
- **R^2** (współczynnik determinacji) - pokazuje, ile wariancji wyjaśnia model

softserve

Lekcja 10: Bias/Variance Trade-Off

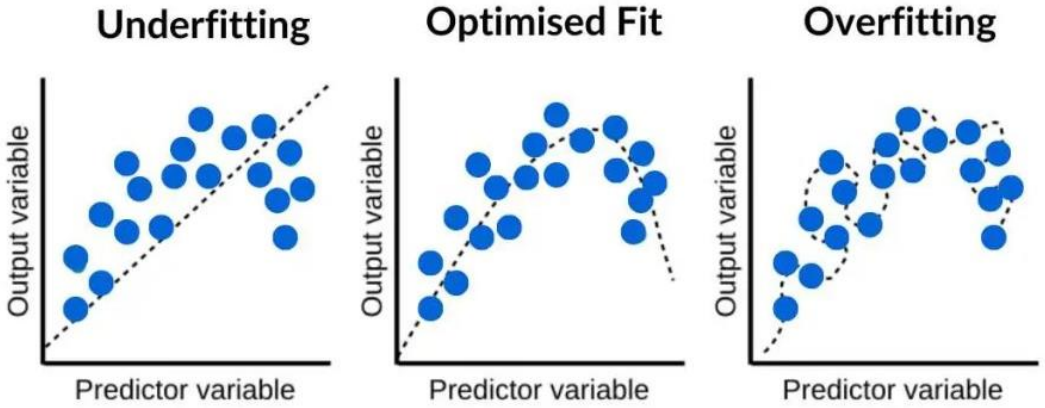
Overfitting czy underfitting?

Underfitting (niedouczenie)

- Model jest zbyt prosty i nie potrafi uchwycić wzorców w danych.
  Efekt: **słabe dopasowanie zarówno do danych treningowych, jak i testowych**.
- Model ma **wysoki bias (obciążenie)** i niską złożoność.

Overfitting (przeuczenie)

- Model jest zbyt dopasowany do danych treningowych, w tym do szumów i wyjątków.
  Efekt: **bardzo dobre wyniki na treningu, ale słabe na nowych danych**.
- Model ma **wysoką wariancję** i zbyt dużą złożoność.



softserve

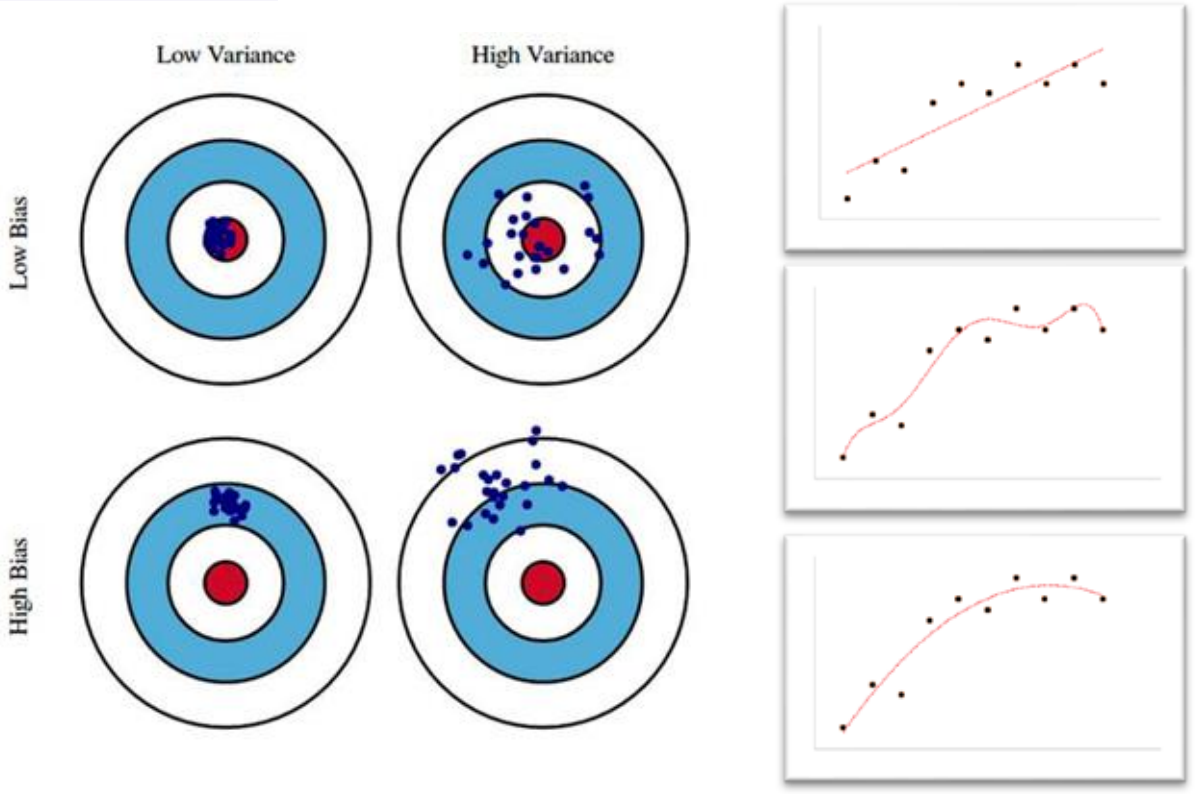
Lekcja 10: Bias/Variance Trade-Off

Model = Variance + Bias + Randomness

- **Bias (obciążenie):** model uczy się uproszczonych wzorców, co prowadzi do systematycznych błędów.
- **Variance (wariancja):** model zbyt mocno dopasowuje się do danych treningowych, przez co słabo działa na nowych danych.
- **Losowość:** błędy wynikające z czynników niezależnych od modelu lub danych treningowych.

Overfitting
- Zmniejsz złożoność modelu (płytsze drzewa, mniejsze max_depth)
- Zastosuj regularizację (L1 / L2)
- Ogranicz liczbę zmiennych lub zastosuj feature selection

Underfitting
- Zwiększ złożoność modelu
- Dodaj nowe, bardziej informatywne zmienne (feature engineering)
- Zastosuj bardziej zaawansowany model (np. XGBoost zamiast regresji liniowej)



Obciążenie i wariancja są składnikami błędu modelu.
Redukcja błędu wymaga odpowiedniej równowagi między nimi

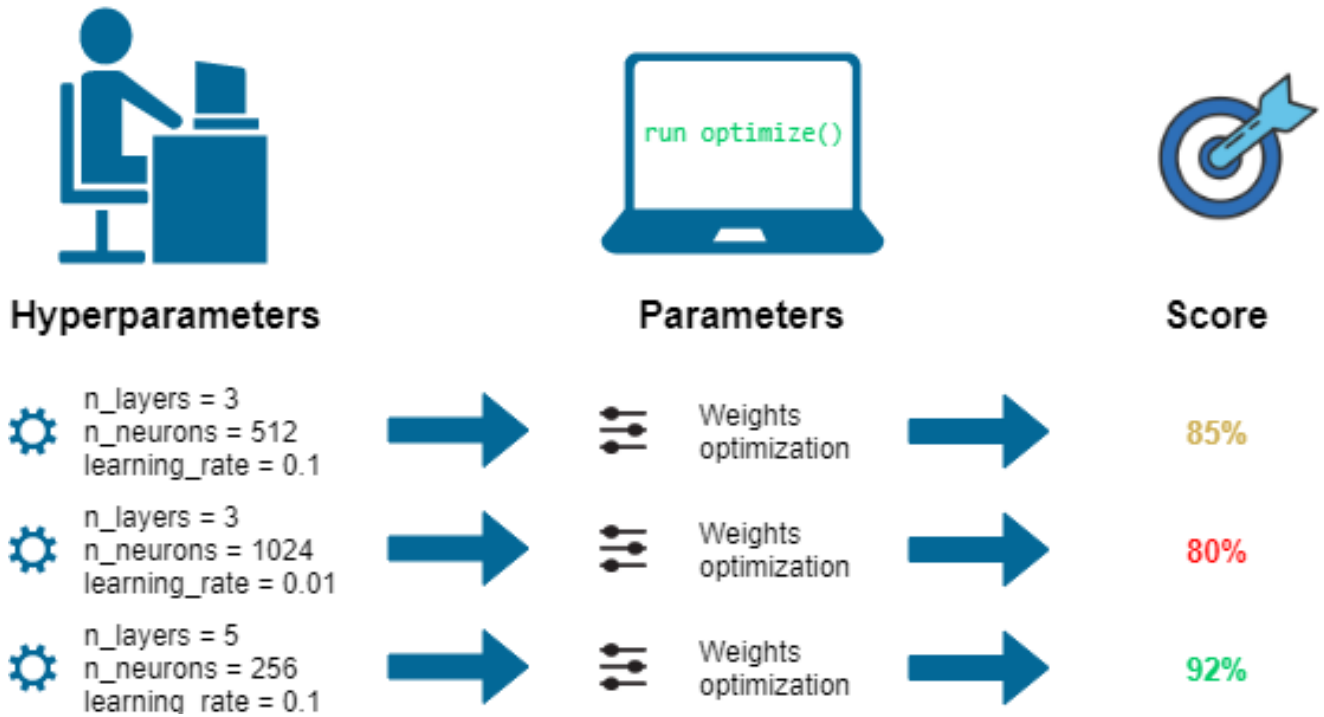
softserve

Lekcja 11: Hyperparameter Tuning wisienka na torcie

Dostosowywanie hiperparametrów może poprawić wyniki modelu, ale **to zwykle jest dopiero ostatni etap procesu**. Największy wpływ na jakość predykcji ma **feature engineering i przygotowanie cech** – bez tego nawet najlepsze strojenie nie przyniesie znaczącej poprawy. Jeśli decydujemy się na tuning, warto używać narzędzi takich jak:

- **Optuna**
- **Hyperopt**

Przykładowe hiperparametry XGBoost
- *learning_rate (eta)* – tempo uczenia
- *max_depth* – maksymalna głębokość drzewa
- *n_estimators* – liczba drzew
- *lambda / alpha* – regularizacja (L2 / L1)



softserve

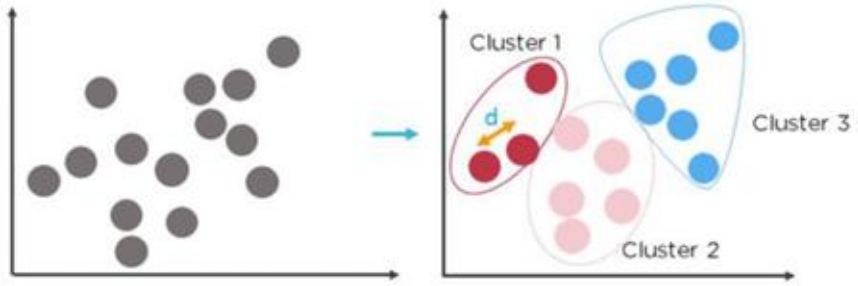
Lekcja 12: Grupowanie: odkrywanie ukrytych struktur

Grupowanie

Grupowanie polega na identyfikacji w danych w miarę jednorodnych grup obserwacji.

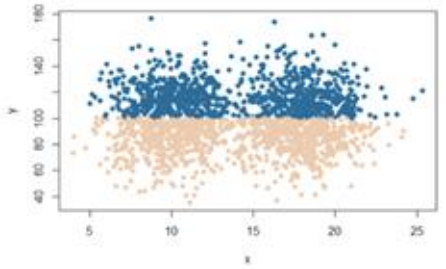
W klasteryzacji wyznacza się odległości pomiędzy punktami. Na podstawie tych odległości wyznaczane są grupy (klastry, clusters):

Przynależność obserwacji do jakiejś grupy (cluster) oznacza, że ta obserwacja jest bardziej podobna do tej grupy niż do innej grupy.

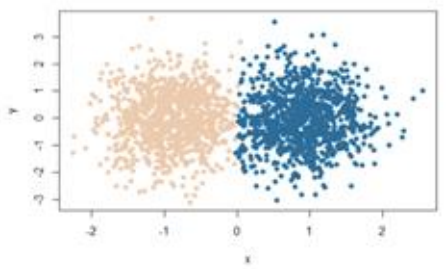


https://www.quora.com/What-does-the-distance-in-a-cluster-mean

Różne skale zmiennych mogą mieć wpływ na działanie algorytmu grupującego, dlatego przed zastosowaniem algorytmów grupowania należy przeprowadzać standaryzację danych.

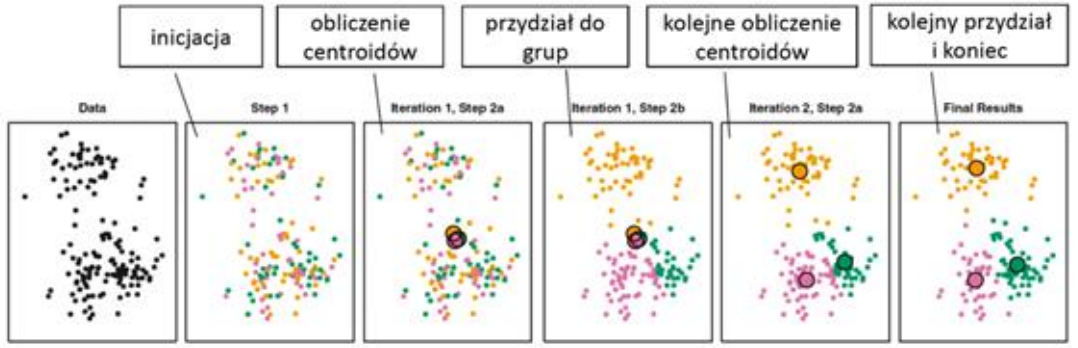


wynik działania algorytmu k-średnich na danych oryginalnych



wynik działania algorytmu k-średnich na danych ustandaryzowanych

K-means

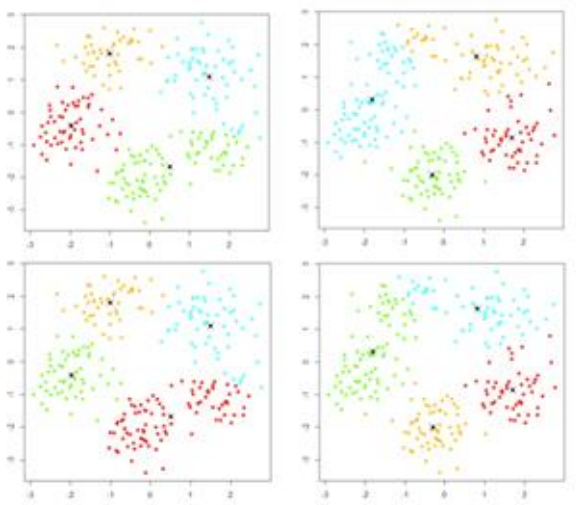


softserve

Lekcja 12: Grupowanie: odkrywanie ukrytych struktur

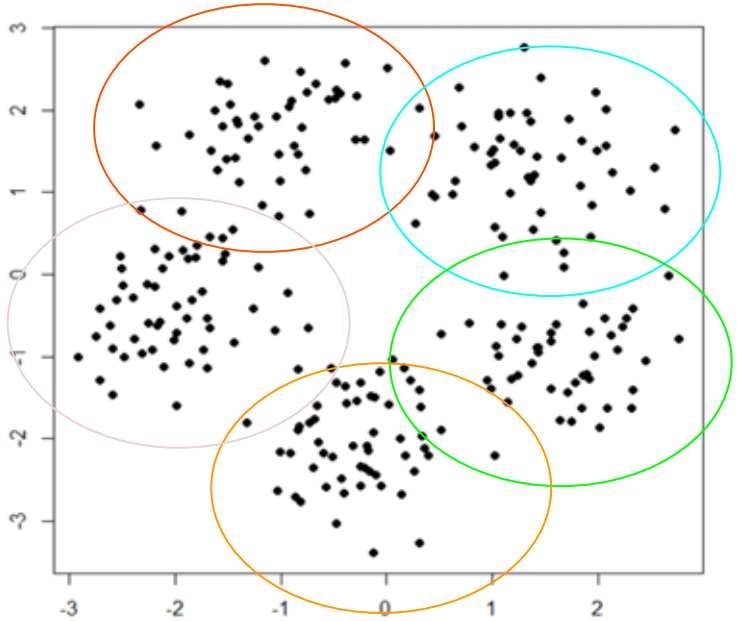
Problem: Losowa inicjalizacja

Minimalizacja błędu WSS
Algorytm w każdej iteracji zmniejsza sumę kwadratów odległości wewnątrzgrupowych (WSS), czyli sumę kwadratów odległości między punktami, a ich centroidami i zatrzymuje się, gdy nie może jej już poprawić. Zaleca się uruchomić go kilkukrotnie i wybrać wynik o najmniejszym WSS.

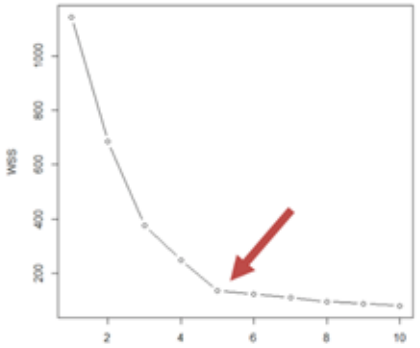


Problem: Wybór liczby skupień

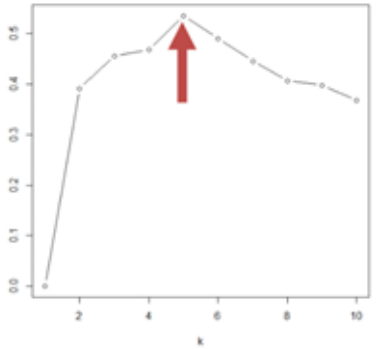
- Wykres osypiska wariancji wewnątrzgrupowej
- Kryterium Calińskiego-Harabasza
- Kryterium współczynnika Silhouette



Elbow method



Silhouette score



$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

softserve

Dziękuję za uwagę

Natalia Cheba

Senior Data Scientist

Wrocław, Politechnika Wrocławska 2026

softserve

KAHOOT QUIZ