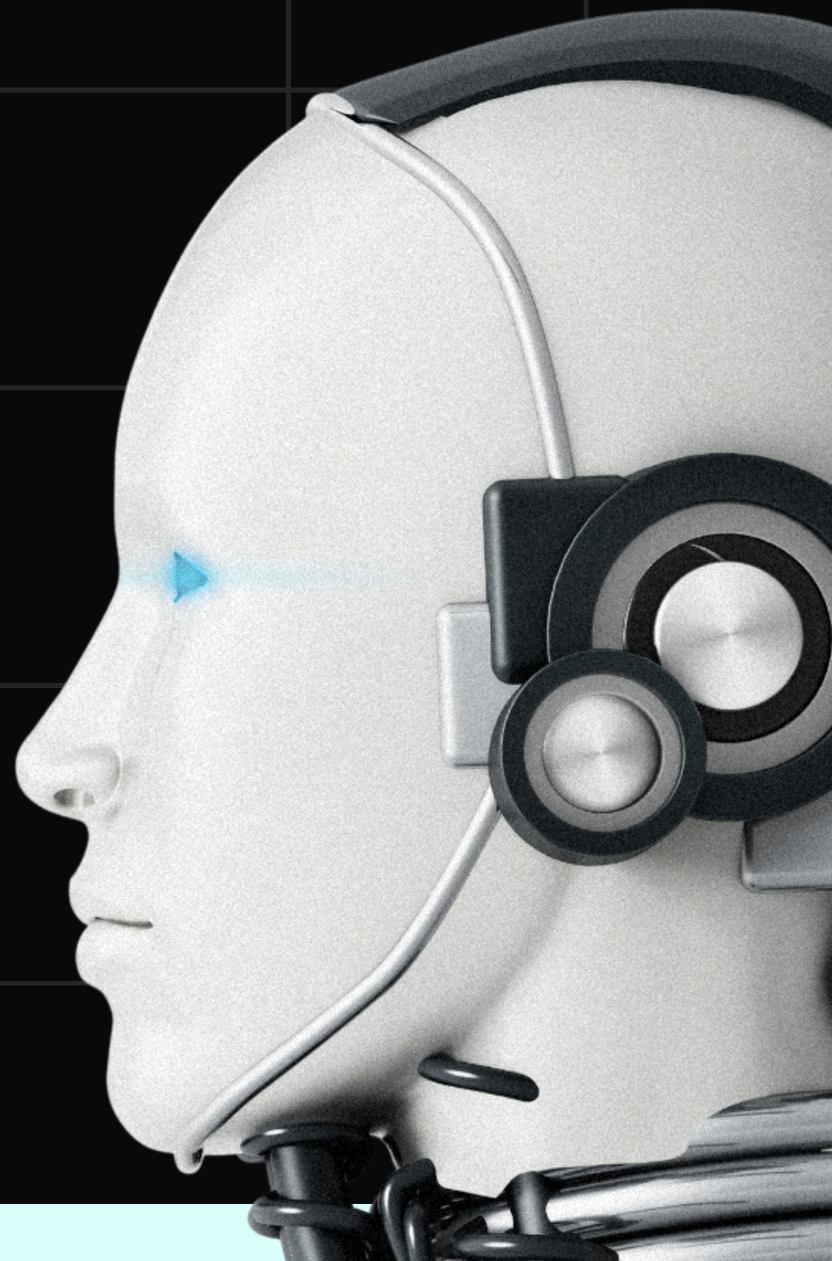
Analiza danych kardiologicznych i predykcji chorób serca

Olha Yakymenko

Celem projektu jest stworzenie zaawansowanego systemu wspomagania decyzji medycznych w diagnozowaniu chorób serca przy użyciu narzędzi sztucznej inteligencji i uczenia maszynowego. Projekt wykorzystuje dane medyczne pacjentów, przetwarza je, analizuje i trenuje różne algorytmy klasyfikacyjne w celu stworzenia systemu zdolnego do przewidywania ryzyka choroby serca. W prezentacji jest szczegółowo opisany każdy element systemu, aby nawet osoby bez specjalistycznej wiedzy mogły go zrozumieć.

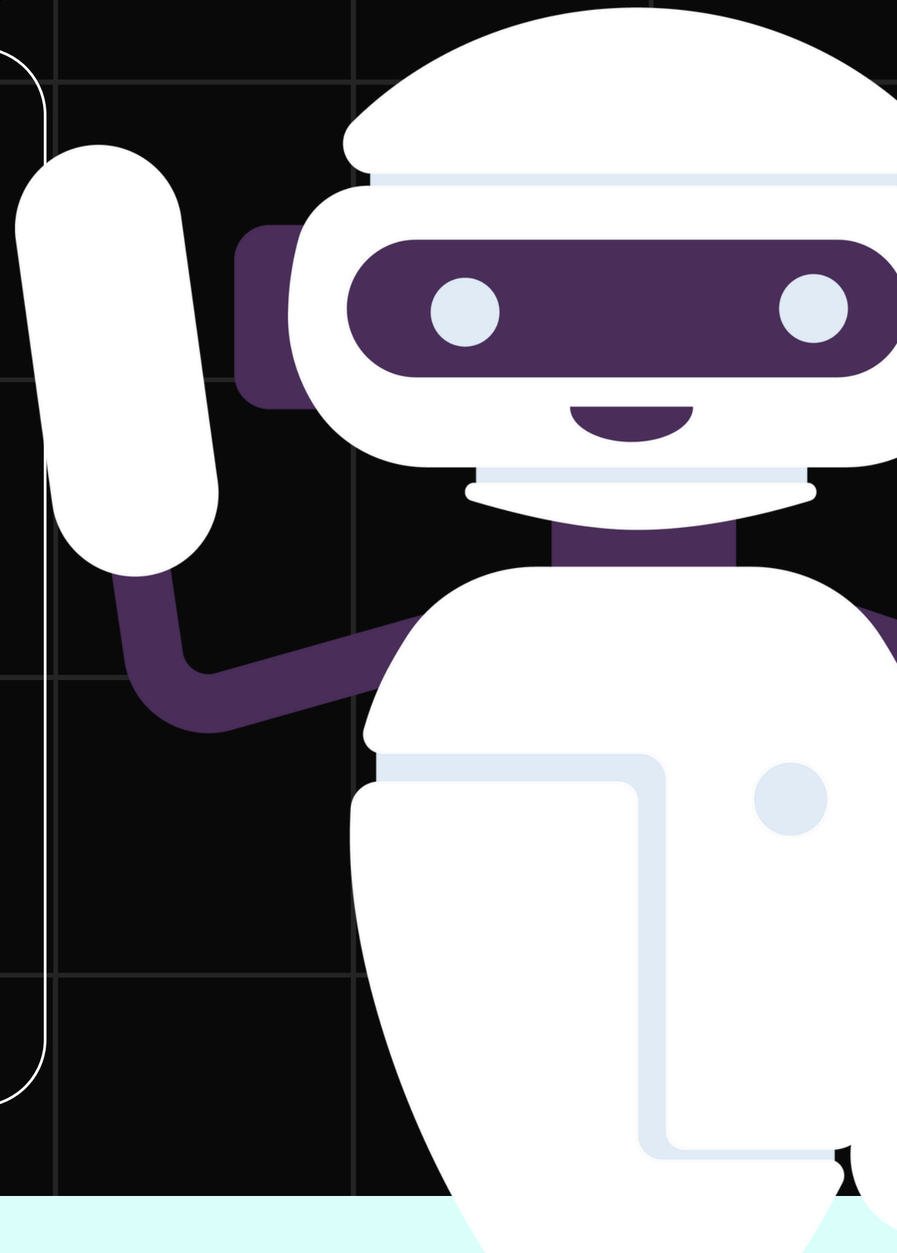
Dane wejściowe: cardio_train.csv
Zbiór danych zawiera informacje medyczne 70 000 pacjentów. Jest to bardzo duży zbiór, co umożliwia trenowanie skomplikowanych modeli, które lepiej generalizują wyniki.



Cechy zawarte w zbiorze:

- age: wiek pacjenta w dniach (przekształcany do lat)
- gender: płeć (1 - mężczyzna, 2 - kobieta)
- height, weight: wzrost i waga (służą też do obliczania BMI)
- ap_hi, ap_lo: ciśnienie krwi (skurczowe i rozkurczowe)
- cholesterol, gluc: poziomy cholesterolu i glukozy w trzech poziomach (normalny, podwyższony, wysoki)
- smoke, alco, active: informacje o stylu życia
- cardio: wartość docelowa (0 - zdrowy, 1 - chory)

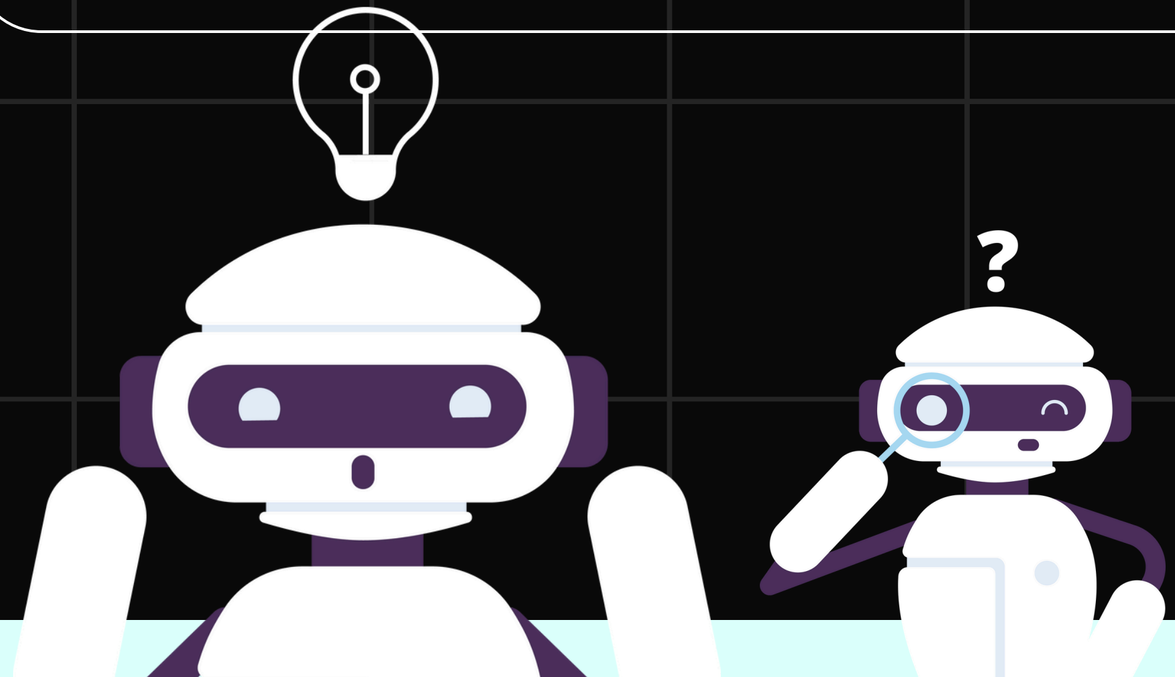
- Wiek przekształcany na lata dla lepszej zrozumiałości.
- Kodowanie zmiennych kategorycznych: np. płeć, cholesterol – za pomocą OneHotEncoder.
- Skalowanie cech liczbowych:
  - StandardScaler: do modeli liniowych i sieci neuronowych
  - MinMaxScaler: do modeli opartych na odległości (KNN)
  - RobustScaler: odporny na wartości odstające



Balansowanie danych

Dane często zawierają nierówne liczby przykładów klas – tzn. może być dużo więcej zdrowych niż chorych pacjentów. To prowadzi do tego, że model preferuje klasy większościowe. Zastosowane metody:

- SMOTE (Synthetic Minority Oversampling Technique): tworzy nowe próbki danych klasy mniejszościowej przez interpolację.
- ADASYN (Adaptive Synthetic): rozszerza SMOTE – generuje więcej danych tam, gdzie klasy są trudniejsze do rozróżnienia.
- TomekLinks: oczyszcza dane przez usunięcie prób o wątpliwej przynależności klasowej.



Modele uczenia maszynowego

Uczenie maszynowe to proces trenowania modelu na danych historycznych, by mógł przewidywać przyszłe przypadki.

MLPClassifier – Sieć neuronowa (MLP)

- Opis: Model inspirowany mózgiem – posiada warstwy „neuronów” przetwarzających dane.
- Zastosowanie: Gdy dane mają złożoną strukturę i wiele wzajemnych zależności.
- Zalety: Bardzo elastyczny, uczy się skomplikowanych zależności.
- Wady: Długi czas treningu, duże zapotrzebowanie na dane.

RandomForestClassifier – Las losowy

- Opis: Składa się z wielu drzew decyzyjnych. Każde drzewo klasyfikuje, a wynik to głos większości.
- Zastosowanie: Dobry „domyślny” model – radzi sobie w większości przypadków.
- Zalety: Radzi sobie z różnymi typami danych.
- Wady: Może być wolny przy bardzo dużej liczbie drzew.

Modele uczenia maszynowego

Uczenie maszynowe to proces trenowania modelu na danych historycznych, by mógł przewidywać przyszłe przypadki.

XGBoost / LightGBM / CatBoost – Nowoczesne boostery

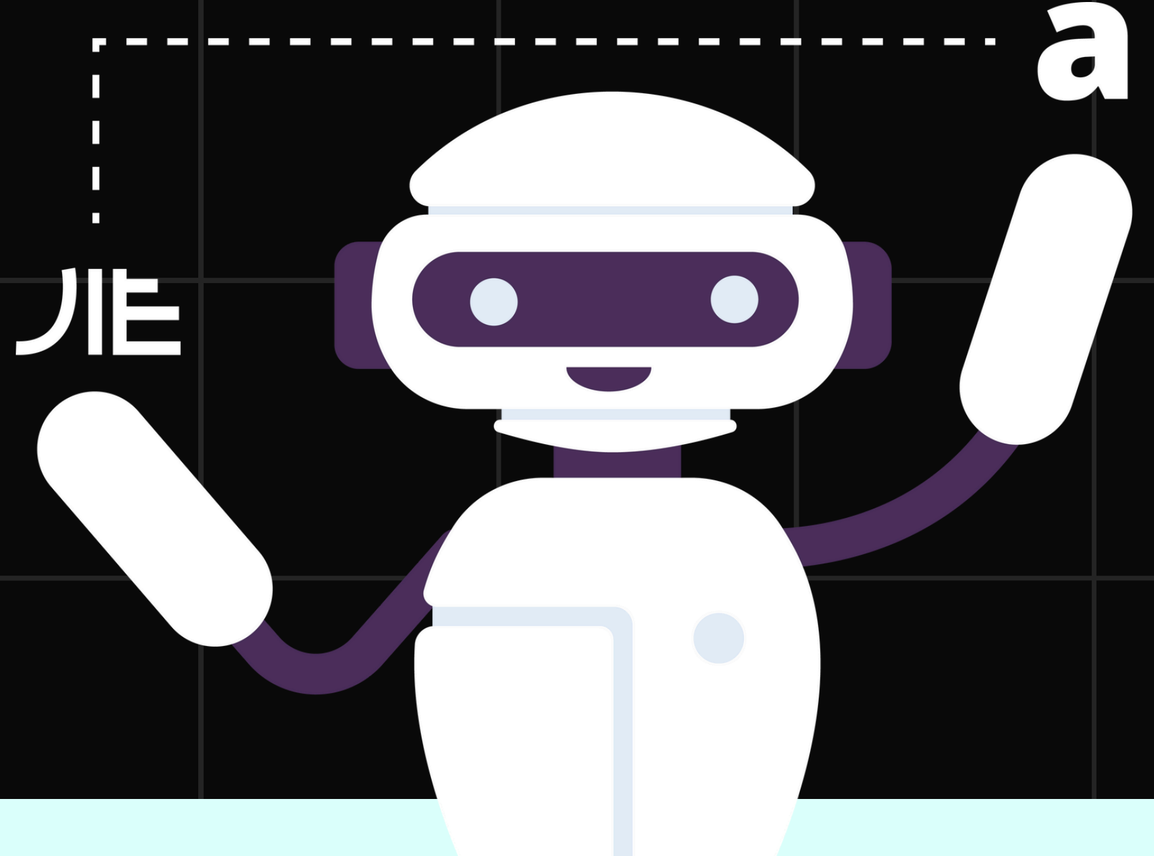
- XGBoost: Wydajny, obsługuje brakujące dane, ma mechanizmy zapobiegające przeuczeniu.
- LightGBM: Ekstremalnie szybki, bardzo dobry przy dużych zbiorach.
- CatBoost: Idealny dla danych kategorycznych – nie wymaga ich kodowania.

GradientBoostingClassifier – Gradient Boosting

- Opis: Buduje kolejne modele poprawiające błędy poprzednich.
- Zalety: Bardzo dokładny.
- Wady: Długi czas uczenia, wymaga dobrego strojenia.

Inne modele

- **LogisticRegression**: Szybki, prosty model liniowy.
- **KNeighborsClassifier**: Klasyfikuje na podstawie odległości do sąsiadów. Powolny przy dużych zbiorach.
- **GaussianNB**: Opiera się na rozkładzie normalnym – szybki, ale nie zawsze trafny.
- **SVC**: Działa dobrze w wielu wymiarach, ale bardzo wolny przy dużych danych.
- **MLP**: Warstwy neuronów uczą się nieliniowych reprezentacji. Może modelować złożone zależności



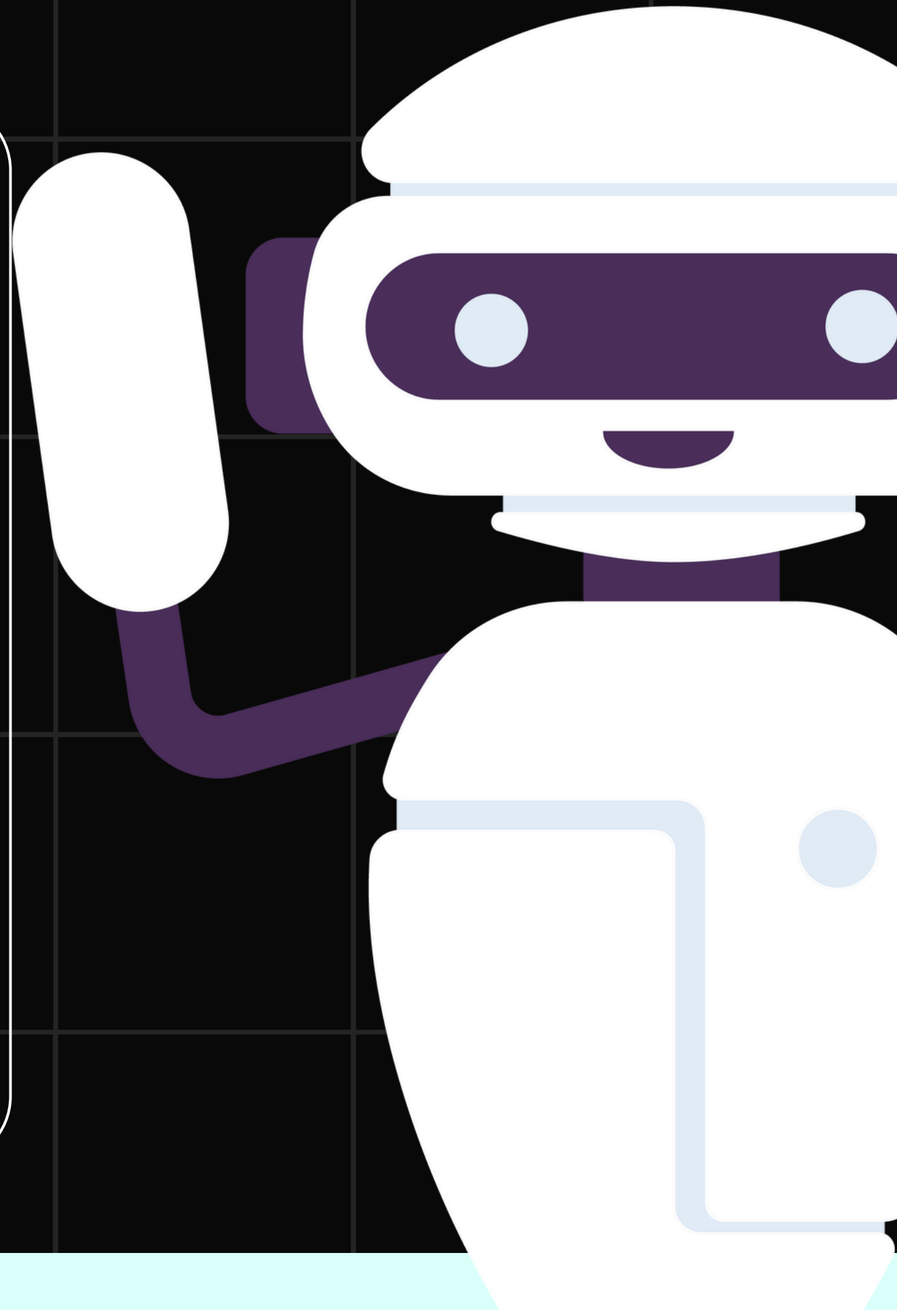
Ewaluacja modeli

Aby wybrać najlepszy model, mierzymy jego skuteczność za pomocą:

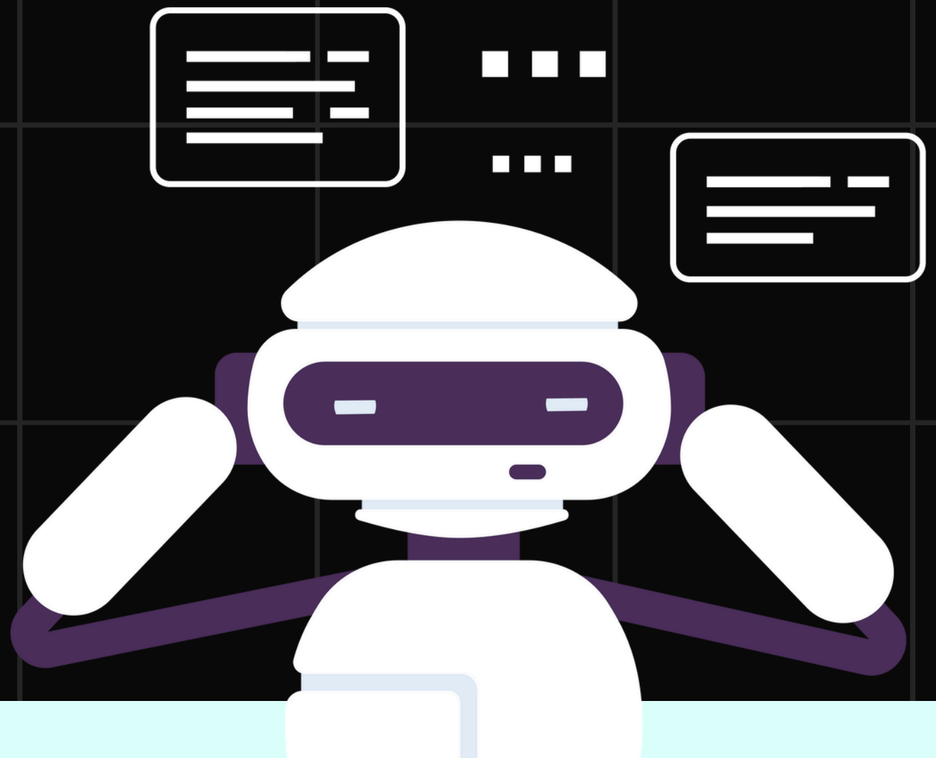
- Accuracy: % poprawnych predykcji
- Precision: precyzja przewidywania klasy pozytywnej (Z ilu przypadków, które model oznaczył jako pozytywne, rzeczywiście są pozytywne?)
- Recall: wykrycie wszystkich przypadków choroby
- F1-score: średnia harmoniczna precyzji i recall.
- Confusion Matrix: szczegółowa analiza trafień i pomyłek
- ROC AUC: wykres pokazujący zdolność rozróżniania klas (im bliżej 1 – tym lepiej)

Dodatkowo:

- Krzywe uczenia
- Krzywe ROC
- Analiza błędów klasyfikacji



Interpretowalność modelu



Model musi być nie tylko dokładny, ale też zrozumiały dla lekarza. Zastosowane techniki:

- Permutation Importance: zmiana kolejności cech i sprawdzenie wpływu na dokładność
- LIME: lokalna interpretacja modelu – pokazuje, dlaczego dany przypadek został sklasyfikowany w określony sposób
- Calibration Curve: sprawdza, czy przewidywane prawdopodobieństwa są realistyczne

Reguły asocjacyjne (Apriori)

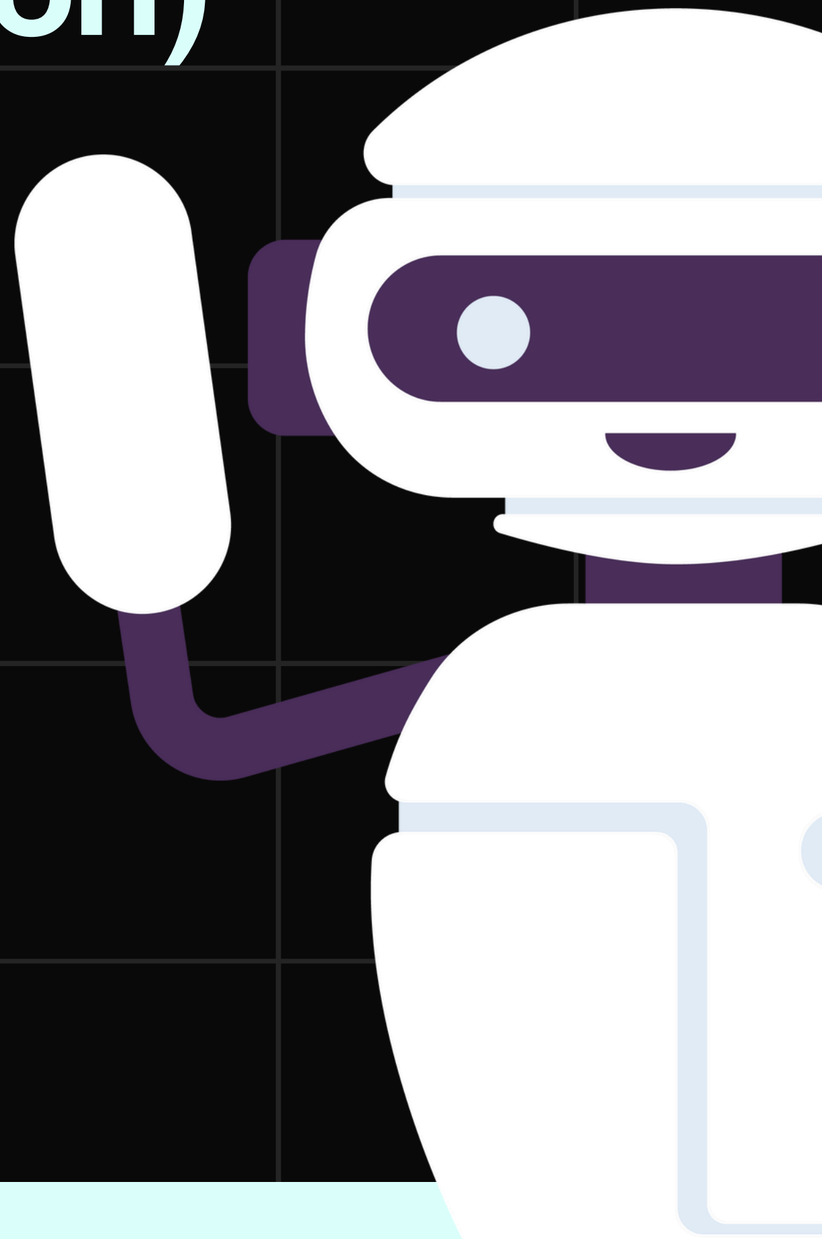
Analiza asocjacyjna służy do wykrywania zależności między cechami pacjentów:

- Apriori: algorytm, który znajduje często współwystępujące zestawy cech (np. wysoki cholesterol + palenie)
- association_rules: na ich podstawie tworzy reguły typu: „jeśli A i B, to często występuje C”

Przykład:

- Jeśli pacjent pali i ma wysokie ciśnienie → ma większe ryzyko choroby serca

To narzędzie jest przydatne do odkrywania ukrytych relacji między czynnikami ryzyka.



Aplikacja Streamlit – Interfejs użytkownika

Aplikacja zbudowana przy użyciu frameworka Streamlit:

- Interaktywny panel użytkownika
- Przegląd danych: statystyki, wykresy, korelacje
- Panel predykcji: użytkownik wpisuje dane pacjenta i otrzymuje informację o ryzyku choroby serca
  - Interfejs zawiera pola do wpisania wieku, ciśnienia, poziomu cholesterolu itp.
  - Po kliknięciu przycisku, aplikacja pokazuje, czy pacjent jest zagrożony
- Możliwość porównania skuteczności różnych modeli w czasie rzeczywistym