**Aplikacja do wykrywania anomalii w odczytach czujników temperatur komputera**

Daniel Skowron 13K2 K02

Mateusz Socha 13K2 K02

Daniel Siembab 13K2 K02

Opis działania:

Aplikacja to kompleksowe narzędzie Machine Learning do automatycznego wykrywania anomalii w odczytach temperatur komponentów komputerowych (CPU, GPU, płyta główna). System składa się z trzech głównych modułów:

1. Generator danych syntetycznych – tworzy realistyczne dane temperaturowe z zaimplementowanymi anomaliami.
2. Moduł treningu modelu – trenuje algorytm Isolation Forest do wykrywania nietypowych wzorców.
3. Detektor anomalii – aplikuje wytrenowany model na nowe dane i wizualizuje wyniki.

Aplikacja automatycznie wykrywa zarówno duże wzrosty temperatury (potencjalne przegrzanie), jak i nagłe spadki (awarie czujników), co widać na załączonym wykresie z różnymi markerami dla każdego typu czujnika.

Użyte technologie Machine Learning:

Isolation Forest:

* Typ: Algorytm uczenia nienadzorowanego (unsupervised learning)
* Metoda: Izolacja anomalii poprzez losowe podziały przestrzeni cech
* Zalety: Nie wymaga etykietowanych anomalii, skuteczny dla danych wielowymiarowych
* Parametry kluczowe:
  + contamination=0.05 (oczekiwany 5% anomalii)
  + n\_estimators=100 (100 drzew decyzyjnych)

StandardScaler:

* Cel: Normalizacja danych (średnia=0, odchylenie standardowe=1)
* Znaczenie: Kluczowe dla algorytmów opartych na odległościach

Feature Engineering:

* Ekstrakcja cech czasowych (godzina, minuta, dzień tygodnia)
* Uwzględnienie cykliczności dobowej temperatur

Kroki wykonywania i opis kodów:

1. generate\_data.py – Generator danych syntetycznych

Funkcja: Tworzenie realistycznego zbioru danych treningowych i testowych

Kluczowe operacje:

* Generowanie sinusoidalnych cykli dobowych temperatur
* Dodawanie szumu Gaussowskiego dla realizmu
* Losowe wstawianie anomalii (wzrosty 80-100°C, spadki 10-30°C)
* Oznaczanie typu anomalii (CPU/GPU/MB) i kierunku (up/down)

Wyjście: Pliki CSV z danymi + wizualizacja

Uruchomienie:  
python generate\_data.py

1. train\_model.py – Trening modelu ML

Funkcja: Trenowanie algorytmu Isolation Forest na danych syntetycznych

Kluczowe operacje:

* Wczytanie danych treningowych
* Ekstrakcja cech czasowych i temperaturowych
* Standaryzacja danych (StandardScaler)
* Trening modelu Isolation Forest
* Walidacja na danych treningowych
* Zapis modelu, skalera i statystyk

Wyjście:

* Wytrenowany model (.joblib)
* Wykres porównujący rzeczywiste vs wykryte anomalie
* Metryki skuteczności

Uruchomienie:  
python train\_model.py

1. detect\_anomalies.py – Detekcja anomalii

Funkcja: Aplikacja wytrenowanego modelu na nowe dane

Kluczowe operacje:

* Wczytanie nowych danych i zapisanego modelu
* Preprocessing danych (analogicznie jak w treningu)
* Predykcja anomalii używając Isolation Forest
* Filtrowanie wyników względem progów statystycznych
* Wizualizacja z zaznaczonymi anomaliami

Wyjście:

* Plik CSV z wynikami detekcji
* Wykres z zaznaczonymi anomaliami (jak na załączonym obrazie)
* Lista wykrytych anomalii w konsoli

Uruchomienie:  
python detect\_anomalies.py data/new\_temperatures.csv

Wizualizacja wyników:

Na załączonym wykresie widać:

* Linie ciągłe: Normalne odczyty temperatur trzech czujników
* Czerwone markery: Wykryte anomalie
  + ❌ (X) – Anomalie CPU
  + ⬛ (kwadraty) – Anomalie GPU
  + 🔺 (trójkąty) – Anomalie płyty głównej
* Wykrycie: Zarówno skoki w górę, jak i nagłe spadki temperatur

Zastosowania praktyczne:

1. Monitoring serwerowni – wykrywanie przegrzewania sprzętu
2. Diagnostyka komputerów – identyfikacja uszkodzonych czujników
3. Prewencyjna konserwacja – wczesne ostrzeganie przed awariami
4. Analiza wydajności – optymalizacja systemów chłodzenia

Aplikacja demonstruje praktyczne zastosowanie uczenia maszynowego w monitoringu infrastruktury IT, oferując automatyzację procesu wykrywania anomalii bez konieczności ręcznego ustawiania progów temperaturowych.