Chmurowe usługi AI

Prognozowanie i wykrywanie anomalii - ćwiczenia

Piotr JANKOWSKI

1 Wykrywanie anomalii

1.1 Wymagania

Wymagania poniżej dotyczą wszystkich zadań (patrz niżej).

1. Klucze uwierzytelniające nie mogą w żadnym momencie być umieszczone w repozytorium. Jeśli wynikowe pliki zawierają takie dane, usuń je ręcznie przed skomitowaniem.
2. Klucze uwierzytelniające uzyskasz od prowadzącego podczas zajęć.

1.2 Cel zadania

Celem zadania jest wykrycie anomalii w seriach danych.

1.2.1 Zapoznanie się z danymi testowymi

Dane testowe znajdują się w `src/test/resources`.

1. Zapoznaj się z plikami w `src/test/resources`.
2. Otwórz je za pomocą edytora tekstowego oraz (jeśli masz taką możliwość) arkusza kalkulacyjnego.

Dane testowe pochodzą z serwisu Google Trends https://trends.google.com/trends/explore.

1.2.2 Wykrycie anomalii za pomocą aplikacji webowej

Odwiedź stronę https://algoevaluation.azurewebsites.net/#/

Zapoznaj się z interfejsem graficznym (konfiguracja, zakładki, przykładowe dane, uruchamianie analizy, parametryzacja).

Wprowadź URL endpointa usługi Anomaly Detection (od prowadzącego).

Wprowadź API Key usługi Anomaly Detection (od prowadzącego).

W zakładce **Univariate Anomaly Detector Entire API** załaduj przykładowe dane **Sample 1**, **Sample 2** lub **Sample 3 with seasonability** i wykonaj analizę. Pola **Current request** i **Current response** powinny zawierać wartości.

Załaduj dane z pliku `wakacje_5lat.csv`. Wykonaj analizę w trybie *univariate, entire* - czyli *batch* z jedną zmienną.

Powtórz dla pliku `kaszel_5lat.csv`.

1.2.3 Wykrycie anomalii za pomocą aplikacji Java i REST API

Zapoznaj się z dokumentacją usługi Anomaly Detection w wariancie *univariate* https://learn.microsoft.com/en-us/azure/cognitive-services/anomaly-detector/quickstarts/client-libraries?tabs=bash&pivots=rest-api.

Zwróć uwagę na format (model) danych wejściowych wymaganych przez tę usługę https://westus2.dev.cognitive.microsoft.com/docs/services/AnomalyDetector/operations/post-timeseries-entire-detect oraz format zwracany w odpowiedzi.

Zaimplementuj metody:

1. `parseCSVSeries` - metoda czytająca zawartość pliku CSV i zwracająca obiekty javowe
2. `prepareJsonData` - metoda "składająca" mapę (`java.util.Map`) reprezentującą zapytanie i serializująca ją do JSONa.

Zapoznaj się z kodem testu `ai.cloud.AzureAnomalyDetectionTest`.

Skonfiguruj zmienne środowiskowe wymagane do uruchomienia testu.

Uruchom test i upewnij się, że przechodzi poprawnie.

1.2.4 Analiza anomalii w wybranym przez siebie zapytaniu do wyszukiwarki Google

Odwiedź stronę serwisu Google Trends (link powyżej).

Wprowadź własne zapytanie, ustaw zakres czasu na 5 lat.

Pobierz plik CSV.

W edytorze tekstowym usuń dwie pierwsze linijki z pliku CSV, aby zostały tylko: linia z etykietami, linie z seriami danych.

Załaduj zestaw danych do aplikacji testowej dla Anomaly Detection (link powyżej). Wykonaj analizę. Jeśli potrzeba, dobierz odpowiednią wartość `sensitivity` i `max anomaly ratio`. Wyeksportuj uzyskany wykres (jako plik graficzny). Umieść go w repo (szczegóły poniżej).

Dopisz nową metodę testową do klasy testowej. Niech będzie to test prosty (nie parametryzowany), który wykona wyszukiwanie anomalii w wybranych przez ciebie danych.

1.2.5 Rezultaty

Napisany przez siebie kod wkomituj do swojego repo (i wypuszuj).

Umieść w repo również wyeksportowany wykres z wykrytymi anomaliami. Nazwij go `echarts-[TWOJA WYSZUKIWANA FRAZA].png`.

Odpowiedz na pytania:

1. Jaka biblioteka została użyta do operacji na plikach CSV?
2. Jaka biblioteka została użyta do operacji na formacie JSON?
3. Jaką własną frazę analizowałaś/-eś?
4. Ile anomalii zostało wykrytych i przy jakich parametrach (`sensitivity`, `max anomaly ratio`)?

Odpowiedzi umieść w pliku `responses.txt` lub `response.md` i umieść w głównym katalogu swojego repo.