

TEMATY UZUPEŁNIAJĄCE

REGUŁY ASOCJACYJNE, GENERATYWNE AI

📁 ZADANIE 1: TITANIC I REGUŁY ASOCJACYJNE

Ponad 100 lat temu zatonął Titanic. Na pokładzie znajdowały się tysiące ludzi. Niektórzy przeżyli, inni nie. Czy były jakieś zależności pomiędzy płcią, zamożnością i przeżywalnością? Twoim zadaniem jest znalezienie reguł asocjacyjnych, które odpowiedzą na te pytania. Wczytaj plik `titanic.csv` ze strony lub z Internetu i upewnij się, że ma tylko 4 kolumny: "Class", "Sex", "Age", "Survived".

Rozwiązanie powinno zawierać:

1. Uruchomienie algorytmu Apriori na bazie danych.
2. Wyszukanie reguł o sensownych parametrach (minimum 0.005, i ufności 0.8) i posortowanie ich wg ufności.
3. Wyszukanie najciekawszych reguł. W szczególności wskazujących, kto przeżywał a kto nie. Można tutaj próbować odfiltrować niepotrzebne reguły.
4. Zobrazowanie reguł na pomocniczych wykresach i interpretacja ich. Zaprezentuj przynajmniej jeden.

Skorzystaj z jednego z samouczków:

<https://medium.com/analytics-vidhya/association-analysis-in-python-2b955d0180c>

<https://www.codespeedy.com/apriori-algorithm-in-python/>

<https://pbpython.com/market-basket-analysis.html>

<https://towardsdatascience.com/mba-for-breakfast-4c18164ef82b>



📁 ZADANIE 2: GENEROWANIE OBRAZKÓW

Na wykładzie były prezentowane programy `gan01.py`, ..., `gan07.py`. Zostały stworzone na podstawie samouczka, do którego warto zajrzeć

<https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-an-mnist-handwritten-digits-from-scratch-in-keras/>

Wygenerujemy obrazki pisanych ręcznie cyfr (MNIST).

- a) Ściągnij pliki pythonowe z wykładu i przejrzyj je. Możesz je uruchomić.
- b) Szczególnie ważny jest plik `gan07.py`. Uruchom go ustawiając liczbę epok na 3 (trenowanie powinno zająć około 10 minut).
- c) Sprawdź jak wyglądają wygenerowane obrazki po każdej epoce (w `generated_plot_eXXX.png`).
- d) Czy model GAN da się zapisywać (wagi sieci w pliku)? Czy taki zapisany model da się potem załadować i dotrenować na kolejnych próbkach danych? Spróbuj znaleźć sposób na to.
- e) Wytrenuj model na większej liczbie epok (np. kilka godzin trenowania). Jak będą wyglądały wygenerowane obrazki? Pokaż je.

📁 ZADANIE 3: LSTM I GENEROWANIE TEKSTU

Wykonaj następujące polecenia związane z sieciami rekurencyjnymi.

- a) Uruchom i przeanalizuj pliki z wykładu o sieciach rekurencyjnych:
 - `rnn01.py` (prosta demonstracja)
 - `rnn02.py` (Badanie jak działa sieć rekurencyjna)
 - `rnn03.py` (Przewidywanie liczby plam na słońcu w danym miesiącu -RNN)

- lstm01.py (Przewidywanie liczby plam na słońcu w danym miesiącu - LSTM)
- lstm02.py (Uczenie generowania tekstu przez LSTM litera po literze)
- lstm03.py (Uruchomienie generatora LSTM z poprzedniego zadania)
- lstm04.py (Uczenie generowania tekstu przez LSTM słowo po słowie)
- lstm05.py (Uruchomienie generatora LSTM z poprzedniego zadania)

Uwaga: w zadaniach lstm02.py i lstm04.py uczenie może trwać wiele godzin.

- b) Znacznie zmniejsz liczbę epok, by trenowanie trwało parę minut.
- c) Zadeemonstruj jak działają generatory tekstu (lstm03 i lstm05) po Twoim krótkim treningu. **Wybierz inną książkę niż Alice in Wonderland.**
- d) Dotrenuj model z lstm02 i lstm04 o parę epok, wykorzystując jako bazę startową już wytrenowany przez Ciebie model zapisany w pliku hdf5. Następnie pokaż, czy generowany tekst jest trochę lepszy.