

TEMATY UZUPEŁNIAJĄCE

REGUŁY ASOCJACYJNE, TRANSFER LEARNING, OPENCV, ANOTACJA DANYCH I DETEKCJA OBIEKTÓW

📖 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: OPENCV I DOMINUJĄCY KOLOR ZIELONY

Na wykładzie zastanawialiśmy się nad następującym problemem. Jeśli mamy kolorowy obrazek z kanałami RGB (czy BGR) i chcemy go skonwertować na obrazek w szali szarości. To jako należy obliczyć wartość szarego piksela.

Pierwsza intuicja mówi nam, że należy obliczyć średnią

$$\text{Gray} = \text{Round}\left(\frac{R + G + B}{3}\right)$$

Okazało się jednak, że lepsze dla ludzkiego oka jest stosowanie wzoru

$$\text{Gray} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

Przetestuj oba sposoby dla paru zdjęć i porównaj efekty.

ZADANIE 3: OPENCV I LICZNIK PTAKÓW

Do folderu z materiałami dołączono folder z miniaturami obrazów ptaków.



Zadaniem jest policzenie ptaków na obrazie i zwrócenie liczby. Na przykład dla ostatniego obrazu powyżej powinno być 2.

Jak rozwiązać ten problem?

Możesz spróbować następujących kroków:

- Wczytaj obrazy. Możesz spróbować znaleźć funkcję, która iteruje przez wszystkie obrazy w danym folderze.
- Konwertuj je na skalę szarości (są to obrazy kolorowe, chociaż wyglądają na szare!).
- Pobaw się filtrami, aby spróbować znaleźć optymalne progi dla wykrywania obiektów. Idealnie byłoby, gdyby tło było białe, a ptaki czarnymi plamami pikseli. Być może znajdziesz lepsze filtry adaptujące się do jasności obrazu.
- Napisz funkcję, która liczy liczbę czarnych plam na obrazie.
- Użyj funkcji do liczenia ptaków na wszystkich obrazach. Wydrukuj listę: nazwa obrazu + liczba ptaków na tym obrazie.

Jeśli masz problemy z tym zadaniem, możesz użyć ChatGPT lub Google Gemini, lub innych narzędzi do pomocy.

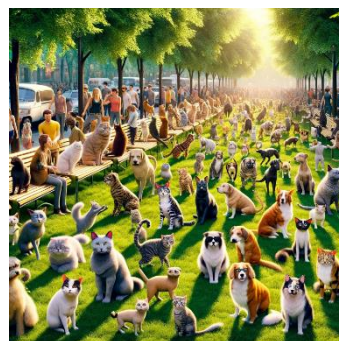
ZADANIE 4: TRANSFER LEARNING DLA PSÓW I KOTÓW

Poprzednio klasyfikowaliśmy psy i koty za pomocą sieci konwolucyjnej.

Pytanie jest następujące: czy można wykorzystać jakiś pre-trained model, dotrenować go, by osiągnąć lepsze rozpoznawanie psów i kotów z obrazków?

Jaki jest plan działania:

- Wybierz jakiś model (np. dostępnych w obrębie keras: <https://keras.io/api/applications/>)
- załaduj model i odpowiednio go zmodyfikuj, doklejając do niego warstwy.
- Przetrenuj tak stworzony model na bazie danych z psami i kotami.
- Porównaj wyniki z wynikami z poprzednich laboratoriów.



ZADANIE 5: 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).

- Ściągnij pliki pythonowe z wykładu i przejrzyj je. Możesz je uruchomić.
- Szczególnie ważny jest plik gan07.py. Uruchom go ustawiając liczbę epok na 3 (trenowanie powinno zająć około 10 minut).
- Sprawdź jak wyglądają wygenerowane obrazki po każdej epoce (w generated_plot_eXXX.png).
- 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 6: DETEKCJA OBIEKTÓW

Zajmij się tematem detekcji obiektów na obrazach („object detection”).

Należy zrobić dwie rzeczy:

- Wybrać jakiś algorytm detekcji obiektów na obrazkach (np. YOLO lub Fast R-CNN)
- Zademonstrować działanie tego algorytmu w Pythonie na jakichś wyuczonych modelach, na przykładowych kilku zdjęciach. Można też wytrenować własny.

Można się inspirować tutorialami z Internetu lub podeprzeć pomocą od AI. Kilka linków, które wpadły mi w oko:

- <https://www.geeksforgeeks.org/detect-an-object-with-opencv-python/>
- <https://www.javatpoint.com/object-recognition-using-python>
- <https://www.analyticsvidhya.com/blog/2018/06/understanding-building-object-detection-model-python/>
- <https://www.hitechbpo.com/blog/top-object-detection-models.php>
- <https://dagshub.com/blog/best-object-detection-models/>

Są pewnie lepsze. Poszukaj! 😊