

# 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.

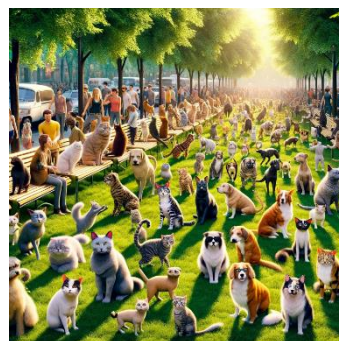
### ZADANIE 4: TRANSFER LEARNING DLA PSÓW I KOTÓW

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: ANOTACJA DANYCH

Anotacja (etykietowanie, oznaczanie) danych to jeden z pierwszych etapów uczenia maszynowego nadzorowanego. Każda obserwacja musi mieć przypisaną etykietkę (klasę). Niestety przypisanie wymaga ingerencji człowieka, najlepiej eksperta w danej dziedzinie i jest to praca dość żmudna i czasochłonna.

W tym zadaniu zmierzmy się z zadaniem dotyczącym oznaczania danych. Będziemy z tego mieli następujące korzyści:

- Oznaczone dane zostaną zebrane i udostępnione jako baza danych do uczenia. Będzie można z niej skorzystać np. w Projekcie 1.
- Zmierzymy się z prawdziwymi danymi zebranymi z kamer/sensorów wykorzystywanych w przemyśle. Będzie to unikalna okazja by skorzystać z danych, których nie znajdziemy nigdzie w Internecie. Unikalność tych danych sprawia, że jest to dobry temat badawczy (praca naukowa, magisterska, licencjacka).

Firma Bioseco (<https://bioseco.com/>) tworzy systemy monitorujące ptaki wokół turbin i lotnisk. Gdy ptaki zbliżają się do turbiny wiatrowej, są odstraszone sygnałami świetlnymi i dźwiękowymi, lub turbina jest spowalniana lub zatrzymywana. Dzięki temu ptaki nie giną w zderzeniach z łopatami turbiny.

System musi jednak pomyślnie rozróżniać ptaki od innych obiektów (samolotów, chmur). Naszym zadaniem będzie przejrzanie zdjęć z systemu BPS (Bird Protection System by Bioseco) i oznaczanie miniaturk obiektów. Zrobimy to za pomocą serwisu online:

<https://am.bioseco.com/>

(login i hasło zostaną przekazane przez prowadzących zajęcia)

Co należy zrobić?

- Zaloguj się w serwisie i pracuj w nim odpowiedzialnie 😊 Twoja praca będzie wykorzystywana przez inne osoby i sprawdzona przez administratora serwisu.
- Wybierz task „**cloud confirm**”, w którym trzeba zgadywać czy obiekty na miniaturkach to chmury czy nie. Mogą zdarzyć się inne obiekty (łopatki turbiny, samoloty ze smugą kondensacyjną).
- Większość zdjęć będzie z chmurami, ale niektóre mogą mieć inne obiekty. Twoim zadaniem jest kliknąć na obiekty niebędące chmurami i oznaczyć je jako „**not cloud**”. Gdy w zestawie miniatur oznaczysz wszystkie zdjęcia, kliknij przycisk „**Send**” w prawym górnym rogu.
- W każdym zestawie miniaturek jest po parędziesiąt zdjęć (zależy od wielkości monitora). Oznaczenie 100 miniaturek powinno więc zająć kilka sekund, 1000 około minuty. Spróbuj oznaczyć 10 000 miniaturek (zajmie kilka minut).
- Przejdźmy następnie do drugiego zadania: „**planes assign**”, które jest ciekawsze i nieco trudniejsze. Jest ono też naszym głównym celem w potencjalnych projektach.
- Zanim przejdiesz do tego zadania, przeczytaj instrukcję obsługi znajdującą się pod guzikiem „**help**” w lewym górnym rogu. Instrukcja opisuje: jak pracować z aplikacją (skrótów klawiszowe, myszka, zaznaczanie) oraz jakie są kategorie obiektów.
- Gdy zapoznasz się z instrukcją przejdź do oznaczeń. Oznacz miniaturki jedną z ośmiu kategorii (zaznaczenie obrazków, i guzik „**assign**” lub „**r**” lub numer). Gdy oznaczysz cały zestaw wyślij go „**Send**” lub litera „**d**”.
- Postaraj się oznaczyć też minimum 10 000 miniatur. Mile widziane i większe numery. Osoby z największą liczbą oznaczeń mogą dostać bonusy punktowe do egzaminu.

Skompletowana baza danych będzie udostępniana na bieżąco wraz z postępem oznaczeń 😊

## ZADANIE 6: DETEKCJA OBIEKTÓW

Co prawda nie był to temat na wykładzie, ale spróbuj zgłębić temat detekcji obiektów na obrazach („object detection”).

Należy zrobić dwie rzeczy:

- Wybrać jakiś algorytm detekcji obiektów na obrazkach (np. YOLO lub R-CNN) i krótko go streścić np. w komentarzu w programie.
- 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! 😊