

Algorytmiczna Analiza Danych 2026/2027

Lista zadań na laboratoria

1 Python + Jupyter Notebook (deadline: 2gie laboratoria)

Zakładam, że znasz podstawy Pythona. Zadania w tej sekcji przeprowadzą Cię przez proces tworzenia wygodnego środowiska programistycznego. Na końcu tej sekcji powinieneś być w stanie pisać i uruchamiać kod Python w Jupyter Notebook oraz znać podstawowe polecenia Jupyter Notebook. Powinieneś również wiedzieć, jak instalować nowe pakiety i przełączać środowiska.

Zadanie 1 — (0p) Pobierz i zainstaluj [Anacondę](#). Jeśli masz już zainstalowanego Pythona i kluczowe w analizie danych pakiety, zainstalowanie Jupyter Notebook jest wystarczające. Przejrzyj [przewodnik użytkownika](#) oraz [ściągę conda](#), aby nauczyć się, jak tworzyć i przełączać środowiska oraz instalować nowe pakiety.

Zadanie 2 — (0p) Uruchom Jupyter Notebook. Naucz się tworzyć, przesuwać i uruchamiać komórki. Naucz się uzyskiwać informacje o obiektach i metodach oraz jak wyświetlać ich kod źródłowy.

Zadanie 3 — (0p) Przypomnij sobie, jakie [struktury danych](#) są dostępne w Pythonie. Zwróć uwagę na list comprehensions - często pozwalają pisać krótszy i czytelniejszy kod.

Zadanie 4 — (0p) Zainstaluj i zapoznaj się z pakietami `numpy`, `pandas`, `matplotlib`.

- Przeczytaj [oficjalny przewodnik użytkownika NumPy](#).
- Przeczytaj [oficjalny przewodnik użytkownika pandas](#).
- Przeczytaj [oficjalny przewodnik użytkownika matplotlib](#).

Zadanie 5 — (1p) Znajdź źródło swojej ulubionej książki i zapisz je w formacie UTF-8. Wczytaj książkę i podziel ją na pojedyncze słowa. Zamień wszystkie litery na małe litery, usuń znaki interpunkcyjne i usuń słowa nieistotne (ang. stop words). Listę słów nieistotnych możesz znaleźć w Internecie (np. [tutaj](#)). Możesz również spróbować użyć [stemmingu](#) lub [lematyzacji](#), aby zredukować różne formy danego słowa do wspólnej formy:

```
from stemming.porter2 import stem
filtered_words = [stem(word) for word in filtered_words]
```

Następnie przekształć uzyskaną listę słów w listę par `(word,1)` typu `(String,Int)`:

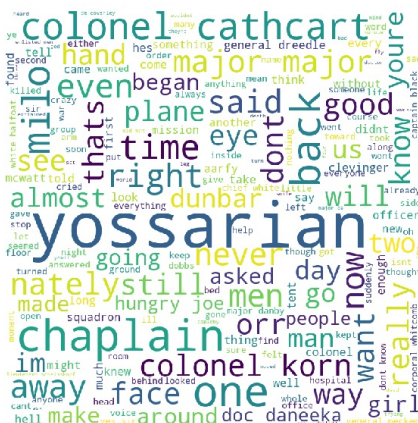
```
pairs = [(w,1) for w in filtered_words]
```

Pogrupuj listę par według różnych słów i policz łączną liczbę wystąpień każdego słowa, aby uzyskać pary `(word, occurrences)`. Na przykład możesz użyć metody `groupby`. Jednak zauważ, że metoda `groupby` wymaga, aby lista wejściowa była posortowana według kluczy,

a takie sortowanie może być kosztowne obliczeniowo dla dużych list. Spróbuj wymyślić i zaimplementować bardziej efektywny sposób agregacji wystąpień, który nie wymaga sortowania.

[illegible]

Posortuj uzyskaną listę par według drugiego elementu w kolejności malejącej. Usuń kilka początkowych elementów posortowanej list (mało istotne, najczęściej pojawiające się słowa) i zapisz wynik do pliku tekstowego. Zbuduj chmurę słów z uzyskanej listy. Możesz użyć biblioteki Pythona lub serwisu <http://www.wordclouds.com/>.



Zadanie 6 — (1p) To jest kontynuacja poprzedniego zadania.

1. Podziel swoją książkę na rozdziały. Traktujemy każdy rozdział jako osobny dokument.
2. Podziel każdy dokument na słowa (używając małych liter, lematyzacji itp.).
3. Wyznacz wagi **tf-idf** dla wszystkich słów we wszystkich dokumentach:

$$tf-idf(t, d, D) = tf(t, d) \times idf(t, D),$$

gdzie t oznacza termin (słowo), d oznacza dokument, a D oznacza zbiór wszystkich dokumentów. Częstotliwość słowa $tf(t, d)$ to liczba wystąpień słowa t w dokumencie d . Odwrotna częstotliwość dokumentu $idf(t, D)$ jest często definiowana jako

$$idf(t, D) = \log \frac{|D|}{1 + |\{d \in D : t \in d\}|}.$$

Uzasadnij sensowność powyższej formuły. Istnieją pakiety, które ułatwiają obliczanie wag tf-idf, ale spróbuj zaimplementować odpowiednią procedurę samodzielnie.

4. Dla każdego dokumentu osobno zbuduj chmurę słów używając uzyskanych wag tf-idf.

Zadanie 7 — (1p) Napisz funkcję, która jako wejście przyjmuje słowo i używa wag tf-idf do stworzenia listy rozdziałów twojej książki najbardziej pasujących do tego słowa (tj. powinna zwrócić listę rozdziałów posortowanych według odpowiednich wag tf-idf).

Zadanie 8 — (1p) Dla każdego słowa w swojej książce utwórz listę pięciu najczęstszych słów, które pojawiają się bezpośrednio po rozważanym słowie (ale pominię słowa stopu). Następnie wygeneruj losowy akapit, wybierając każde kolejne słowo spośród tych pięciu - tym częściej, im częściej występowało ono po danym słowie w książce. Pozwól użytkownikowi podać pierwsze słowo. Porównaj wynik z wersją, w której słowa stopu nie są pomijane.

