

## LAB 5 – Poszukiwanie bibliotek o określonej funkcjonalności

Przetwarzanie i analiza danych - NumPy - stanowi fundamentalną bibliotekę dla obliczeń naukowych w Pythonie. Zapewnia wsparcie dla dużych, wielowymiarowych tablic i macierzy wraz z szerokim zestawem funkcji matematycznych. Jest szczególnie użyteczny w operacjach wymagających wysokiej wydajności obliczeniowej.

### Zalety:

Wydajność i efektywność - NumPy wykorzystuje mniej pamięci i przestrzeni dyskowej w porównaniu do standardowych list Python. NumPy oferuje czytelną i intuicyjną składnię, która czyni go przyjaznym dla użytkownika. NumPy jest narzędziem open source, które można używać całkowicie bezpłatnie.

### Wady:

Skupienie na numerycznych i homogenicznych typach danych jest kluczowe dla wydajności NumPy, ale może również ograniczać elastyczność w porównaniu do innych mechanizmów przechowywania tablic. Wstawianie lub dodawanie wpisów do tablicy nie jest tak proste jak w przypadku list Python. Operacje takie jak `np.concatenate()` tworzą nowe tablice zamiast łączyć istniejące.

Wizualizacja danych - Matplotlib - to najpopularniejsza biblioteka do tworzenia wykresów w Pythonie. Słynie z możliwości tworzenia wysokiej jakości diagramów i oferuje szerokie możliwości customizacji. Podstawowy wykres liniowy można utworzyć za pomocą funkcji `plt.plot()`.

### Zalety:

Matplotlib zapewnia użytkownikom szeroką gamę typów wykresów, w tym wykresy liniowe, punktowe, słupkowe, histogramy, wykresy kołowe, kontury, wykresy 3D i wiele innych. Oferuje wysokiej jakości dokumentację z galeriami przykładów i tutorialami.

### Wady:

API biblioteki może być uciążliwe w użyciu i nie zawsze wydaje się bardzo pythoniczne. Próba przeniesienia niezgrabnego API MATLAB do Python sprawiła, że całość stała się podwójnie niezgrabna.

Przetwarzanie języka naturalnego (NLP) – SpaCy - to nowoczesna i szybka biblioteka do NLP, zaprojektowana z myślą o produkcji. Oferuje doskonałe wsparcie dla wielu języków, w tym polskiego. Umożliwia wykonywanie zaawansowanych operacji jak lematyzacja, analiza zależności czy rozpoznawanie nazwanych jednostek.

Zalety:

Wyjątkowa wydajność - SpaCy jest zbudowany z Cython, co zapewnia błyskawiczną wydajność, nawet dla złożonych zadań. Jest zoptymalizowany pod kątem szybkości i przyjazny dla pamięci. Gotowe modele - SpaCy oferuje różnorodne wstępnie wytrenowane modele, które można dostosować do konkretnych przypadków użycia, oszczędzając czas i wysiłek.

Wady:

Wymagania pamięciowe - Dla bardzo dużych dokumentów (100K+ tokenów) mogą wystąpić problemy z wyczerpaniem pamięci. Kompatybilność wersji - Wymaga Python  $\geq 3.7$ ,  $< 3.13$  (tylko 64 bit).

NumPy:

Dokumentacja oficjalna: <https://numpy.org/doc/>

Repozytorium GitHub: <https://github.com/numpy/numpy>

PyPI: <https://pypi.org/project/numpy/>

Matplotlib:

Dokumentacja oficjalna: <https://matplotlib.org/>

Repozytorium GitHub: <https://github.com/matplotlib/matplotlib>

PyPI: <https://pypi.org/project/matplotlib/>

SpaCy:

Dokumentacja oficjalna: <https://spacy.io/>

Repozytorium GitHub: <https://github.com/explosion/spaCy>

PyPI: <https://pypi.org/project/spacy/>

Modele językowe: <https://spacy.io/models>