Pracownia rotacyjna - Chemia Kwantowa

Jan Wolski

Marzec 2025

1 Wstęp

Celem tego projektu będzie stworzenie modelu sztucznej inteligencji, który na podstawie deskryptorów substancji chemicznych będzie przewidywała powinowactwo elektronowe podanego związku.

2 Dane

Dane pozyskane za pomocą scrapera internetowego [1] napisanego na potrzeby projektu. Został napisany w języku Go i używa biblioteki gorod [2], a same informacje pobiera ze strony: webbook.nist.gov [3]. Program przechodzi w wyszukiwarce wszystkie możliwe zakresy powinowactwa elektronowego i zapisuje je do pliku csv o strukturze:

	Związek	EA (eV)	Wzór sumaryczny
Opis pola	nazwa związku	wartość powinowactwa elektronowego	Wzór sumaryczny
Typ zmiennej	string	float	string
Format	-	6 cyfr precyzji	indeksy dolne są w formie liczby
Przykład	cycloheptanide anion	-0.156000	C7H13-

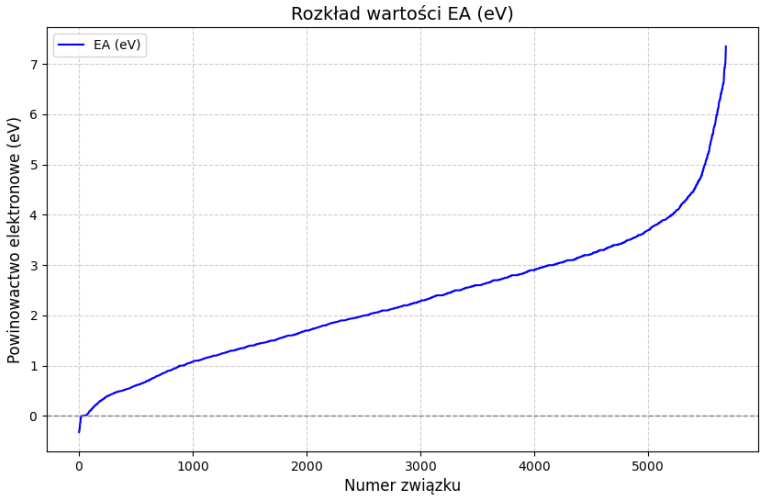
2.1 Rozkład danych

Wartości wahają się miedzy wartościami -0,32 a 7,35 eV (Rysunek 1). Wartości rozkładu mają tendencje do skupiania się wokół 2. Rozkład jest prawoskośny, czyli istnieje niewiele wartości, ale są one duże. Histogram na Rysunku 2.

2.2 Inne wartości statystyczne danych

Na podstawie danych wyliczone inne wartości statystyczne:

- Średnia: 2.2859
- Mediana: 2.1900
- Odchylenie standardowe: 1.2816



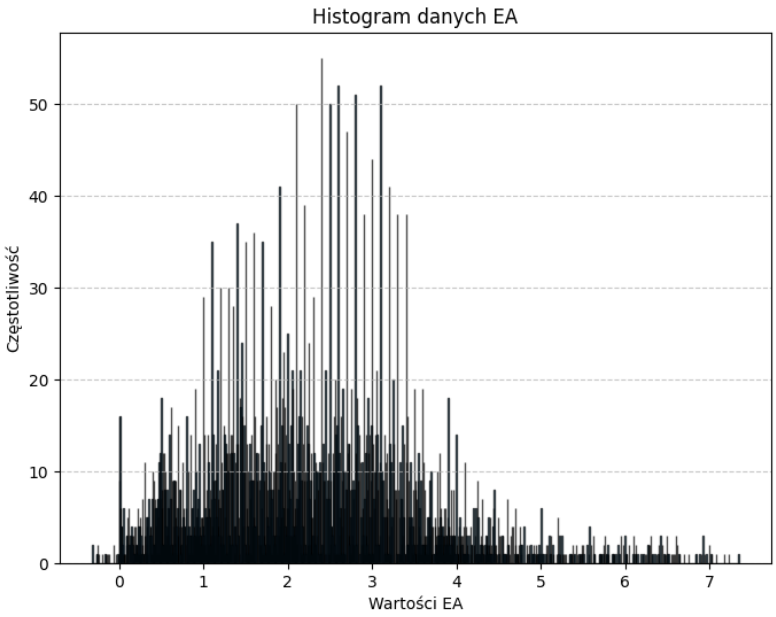
Rysunek 1: Rozkład wartości danych

- Wariancja: 1.6424
- Min: -0.3200
- Max: 7.3500
- Rozstęp: 7.6700
- Q1 (25%): 1.3450
- Q3 (75%): 3.0700
- IQR (Q3-Q1): 1.7250
- Skośność: 0.6672
- Kurtoza: 0.6957

3 Użyte deskryptory

Poniżej przedstawiono deskryptory, wraz z ich skrótami używanymi dalej w pracy, które zostały użyte do trenowania sztucznej inteligencji:

1. Liczba **atomów** w cząsteczce (number of atoms, **nAT**)
2. Liczba atomów w cząsteczce niebędących atomami **wodoru** (number of non-H atoms, **nSK**)



Rysunek 2: Histogram danych - ilość koszyczków = $\frac{ndanych}{4}$

3. Liczba atomów **wodoru** w cząsteczce (number of Hydrogen atoms, **nH**)
4. Liczba atomów **węgla** w cząsteczce (number of Carbon atoms, **nC**)
5. Liczba atomów **azotu** w cząsteczce (number of Nitrogen atoms, **nN**)
6. Liczba atomów **tlenu** w cząsteczce (number of Oxygen atoms, **nO**)
7. Liczba atomów **fosforu** w cząsteczce (number of Phosphorous, **nP**)
8. Liczba atomów **siarki** w cząsteczce (number of Sulfur atoms, **nS**)
9. Liczba atomów **fluoru** w cząsteczce (number of Fluorine atoms, **nF**)
10. Liczba atomów **chloru** w cząsteczce (number of Chlorine atoms, **nCl**)
11. Liczba atomów **bromu** w cząsteczce (number of Bromine atoms, **nBr**)
12. Liczba atomów **boru** w cząsteczce (number of Boron atoms, **nB**)
13. Liczba **heteroatomów** w cząsteczce (number of heteroatoms, **nHet**, czyli atomów niebędących atomami węgla i wodoru)
14. Liczba atomów **halogenu** (tzn. suma atomów fluoru, chloru, bromu i jodu, number of halogen atoms, **nX**)

15. Jaki % całkowitej liczby atomów stanowią atomy **wodoru** (percentage of H atoms, **H%**)

16. Jaki % całkowitej liczby atomów stanowią atomy **węgla** (percentage of C atoms, **C%**)

17. Jaki % całkowitej liczby atomów stanowią atomy azotu (percentage of N atoms, **N%**)

18. Jaki % całkowitej liczby atomów stanowią atomy tlenu (percentage of O atoms, **O%**)

19. Jaki % całkowitej liczby atomów stanowią atomy halogenu (percentage of halogen atoms, **X%**)

20. **Masa cząsteczkow**a (molecular weight, **MW**)

$$MW = \sum m_i,$$

gdzie m_i to masa poszczególnych atomów

21. Średnia masa molowa (Average molecular weight, **AMW**)

$$AMW = \frac{MW}{nAT}$$

22. Suma promieni jonowych atomów tworzących cząsteczkę (sum of atomic vradious, **Sr**)

23. **Suma elektroujemności** Paulinga atomów tworzących cząsteczkę (sum of atomic Pauling electronegativities, **Se**)

24. Suma **polaryzowalności** wszystkich atomów tworzących cząsteczkę (sum of atomic polarizabilities, **Sp**)

25. Suma **energii jonizacji** wszystkich atomów tworzących cząsteczkę (sum of first ionization potentials, **Si**)

26. Średni promień jonowy atomów tworzących cząsteczkę (mean atomic radius, **Mv**)

$$Mv = \frac{Sr}{nAT}$$

27. Średnia elektroujemność atomów tworzących cząsteczkę (mean atomic Pauling electronegativity, **Me**)

$$Me = \frac{Se}{nAT}$$

28. Średnia polaryzowalność atomów tworzących cząsteczkę (mean atomic polarizability, **Mp**)

$$Mp = \frac{Sp}{nAT}$$

29. Średnia energia jonizacji atomów tworzących cząsteczkę (mean first ionization potential, **Mi**)

$$Mi = \frac{Si}{nAT}$$

30. Wskaźnik stopnia nienasycenia (Hydrogen Deficiency Index, HDI)

$$HDI = \frac{2nC + 2 + nN - nH - nX}{2}$$

31. Proporcja węgla do wodoru (Hydrogen/Carbon ratio, H/C)

$$H/C = \frac{nH}{nC}$$

32. Średnia % zawartości pierwiastków (IDK)

$$IDK = \frac{\sum \frac{nE}{nAT}}{N},$$

gdzie:
nE - to ilości kolejnych pierwiastków zawartych w związku,
N - ilość wszystkich unikalnych atomów w związku

Za ich pomocą bedę próbował przewidzieć wartość powinowactwa elektronowego dla zadanego pierwiastka. W pliku *descryptors.py* napisano funkcje o nazwie *descX* gdzie X to numer deskryptora zgodnie z powyższą listą. W tym samym pliku jest również funkcja *get_ descryptors* która zwraca wszystkie wartości. Dokładne omówienie funkcji w sekcji 5.

4 Sieć neuronowa

4.1 Projekt i architektura sieci neuronowej

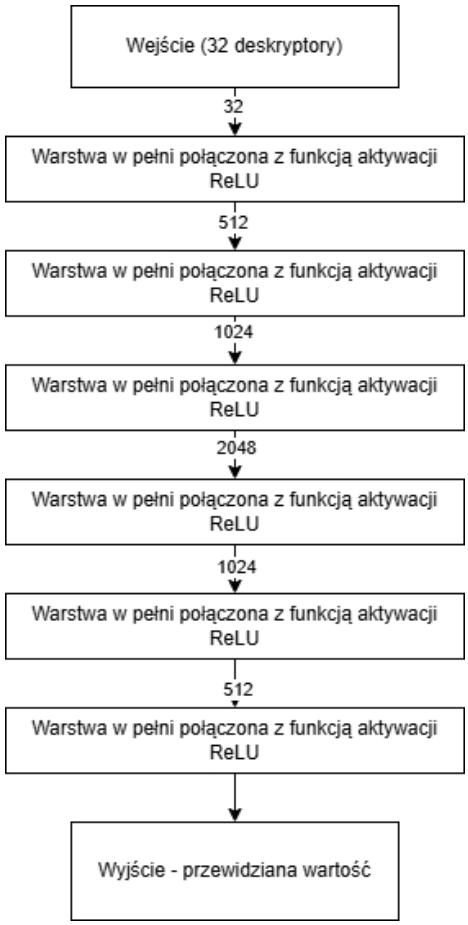
4.1.1 Projekt 1

Pierwszy projekt sieci neuronowej do tego zadania był prosty - gęsta sieć w pełni połączona (Deep Fully Connected Network, DFCN) z jednym neuronem wyjściowych bez funkcji aktywacji (3). Jednak z powodu małej ilości danych nie udało się osiągnąć zadowalających wyników. Zaimplementowano mechanizm *early stopping* - mechanizm, który zapobiega przeuczeniu, gdy sieć nie ulepsza wyników przez określoną liczbę epok (w tym przypadku przez 10).

Wyniki ewaluacji

Ilość epok: 39

Test Loss (MAE): 0.5178



Rysunek 3: Schemat architektury modelu nr. 1 - na strzałkach wielkości przekazywanych macierzy

Mean Squared Error (MSE): 0.6749

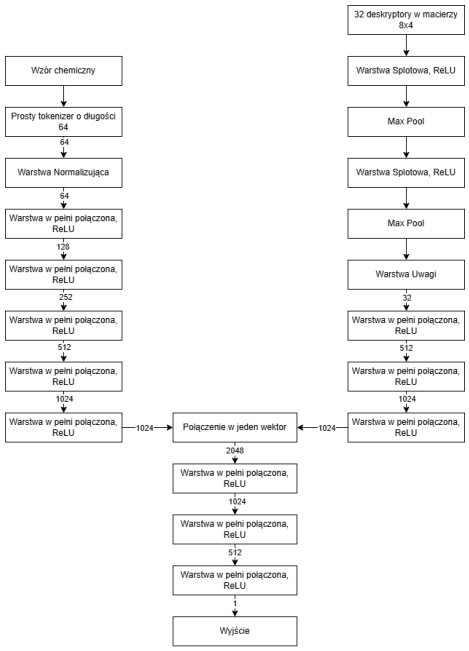
Root Mean Squared Error (RMSE): 0.8215

R^2 Score: 0.6138

Mean Absolute Percentage Error (MAPE): 493.74%

Symmetric Mean Absolute Percentage Error (SMAPE): 30.12%

RMSE i MAE nie są bardzo wysokie to znaczy że model działa poprawni, ale nie bardzo dobrze. R^2 wskazuje na umiarkowaną skuteczność modelu (pokrywa ponad 60% wariancji) - potrafi uchwycić część zależności, ale nie wszystkie.



Rysunek 4: Schemat architektury modelu nr. 2 - na strzałkach wielkości przekazywanych macierzy

Wysokie MAPE sugeruje, że model ma poważne problemy z niektórymi przypadkami, szczególnie z tymi bliskimi 0. SMMAPE pokazuje, że model średnio myli się o 30% rzeczywistej wartości - co nie jest zbyt dobre.

Nawet pomimo użycia augmentacji danych wyniki nie poprawiły się znacząco, więc wiadome było, że przy tej architekturze modelu nie uda się osiągnąć lepszych wyników.

4.1.2 Projekt 2

Tak jak widać na Rysunku 4 drugi projekt modelu podzielony jest na 3 części. Pierwsza jest odpowiedzialna za przetwarzanie informacji o 32 deskryptorach, jednak tym razem zostały one przedstawione za pomocą macierzy 8x4, co pozawala na ich dalszą analizę w warstwie splotowej. Po dwóch warstwach splotowych jest jedna Warstwa Uwagi, a następnie wnioski z nich są przekazywane dalej z pomocą warstw w pełni połączonych (Dense).

Druga część modelu zajmuje się przetwarzaniem informacji zawartych we wzorze chemicznym. Najpierw normalizuje wektor przekazany przez tokenizer, a następnie przepuszcza go przez 5 warstw Dense.

Trzecia część zajmuje się połączeniem wyjść z dwóch pierwszych i wyciągnięciem z nich wniosków z pomocą kilku warstw Dense. Po dwóch pierwszych zastosowano Dropout z parametrem 0.2 aby zmniejszyć szansę przeuczenia. Zastosowano, również augmentację danych (szczegóły w sekcji 5) jak i inicjalizację wag modelu, co pozwoliło znacznie podnieść jakość uczenia. Tak jak w modelu 1 zaimplementowano mechanizm *early stopping*.

Wyniki ewaluacji

Ilość epok: 98

Test Loss (MAE): 0.2108

Mean Squared Error (MSE): 0.1070

Root Mean Squared Error (RMSE): 0.3271

R^2 Score: 0.9348

Mean Absolute Percentage Error (MAPE): 149.76%

Symmetric Mean Absolute Percentage Error (SMAPE): 14.78%

Model jest dobrze dopasowany, co sugeruje R^2 i niskie RMSE, ale wysokie MAPE sugeruje problemy z niektórymi danymi, prawdopodobnie tymi bliskimi 0. W porównaniu do modelu 1 SMAPE jest mniejsze o połowę.

4.1.3 Konkluzja

Po przeanalizowaniu obu prototypów zdecydowano się na kontynuację prac nad modelem nr. 2 przez lepsze wyniki pod każdym względem.

5 Omówienie implementacji

5.1 Liczenie deskryptorów

Wszystkie operacje na deskryptorach umieszczono w pliku *descryptors.py*. Funkcje przyjmujące wzór chemiczny i zwracające wartość danego deskryptora zawarto w funkcjach *descX*, gdzie X to nr deskryptora np.:

```
def desc30(formula: str) -> float:
    """HDI"""
    return (2 * desc4(formula) + 2 + desc5(formula) - desc3(formula
    ) - desc14(formula))/2
```

Listing 1: Funkcja licząca wartość HDI

Na początku pliku znajduje się tablica wszystkich pierwiastków - służy ona *filter_ unknown_ elements* do zidentyfikowania błędów we wzorach chemicznych. *parse_ formula* zwraca słownik symbol pierwiastka: ilość elementów. *extract_ elements* zwraca unikalne symbole pierwiastków we wzorze. Główna funkcja pliku: *get_ descriptors* pobiera dataset z dysku i oblicza wszystkie deskryptory dla wszystkich związków chemicznych w danych - wyrzucając te które zawierają nieznane dla algorytmu pierwiastki. Deskryptory 1 - 21 obliczane są po klei, a 22-29 obliczane są współbieżnie. Za każdym razem, gdy uruchomimy plik zostanią pobrane wszystkie potrzebne wartości stałych z biblioteki mendeleev [4].

Aby wyliczyć deskryptory i zapisać je na dysku, należy umieścić plik *dataset.csv* [1] i uruchomić plik. Kopia datasetu wraz z obliczonymi deskryptorami zapisze się jako *descriptors.csv*.

5.2 Model

Implementacja modelu znajduje się w pliku *models.py*.

SelfAttention To klasyczna implementacja warstwy uwagi, która przypisuje wagi każdemu elementowi na podstawie jego relacji z innymi elementami.

1. Obliczamy macierze Kluczy (K), Zapytań (Q) i wartości (V) przez transformacje wag uczonych w trakcie treningu.
2. Obliczamy iloczyn skalarny Q i K i stosujemy funkcję softmax, aby wydobyć współczynnik uwagi.
3. Mnożymy uzyskany wynik przez V i tworzymy reprezentacje wyjściową.

FormulaEncoder Część modelu zajmująca się ztokenizowanym wzorem chemicznym

DescryptorEncoder Część modelu zajmująca się deskryptorami w formie 8x4

ElectronAffinityRegressor Główny model łączący ze sobą dwie części. Wartą uwagi jest część gdzie rozdzielamy i łączymy dane z danych wejściowych (Figura 2). W niej bierzemy pierwsze 64 pozycje w wektorze (zakładamy że właśnie tam są tokeny z formuły) i przekazujemy do FormulaEncoder a pozostałe wartości przekształcamy tak aby nadawały sie do warstwy konwolucyjnej (x.reshape(b, 1, 8, 4), b to wielkość batchu, 1 kanał, macierz 8x4) i przekazujemy do DescryptorEncoder. gdy otrzymamy już oba wyniki (każdy z nich o wymiarze 1024) spłaszczamy je i dodajemy do siebie tworząc jeden wektor od długości 2048. Po próbach zdecydowano się powiększyć i pogłębić ostatnią część modelu do 7 warstw Dense.

```
y = inp[:, :64]
y = self.formula(y)

```

```
x = inp[:, 64:]
x = x.reshape(b, 1, 8, 4)
x = self.descryptor(x)
x = x.reshape(b, -1)
out = torch.cat((x, y), dim=1)
```

Listing 2: sposób rozdzielania i łączenia danych w modelu

5.3 Funkcje pomocnicze

Funkcje pomocnicze znajdują się w pliku *utils.py*.

augment_dataframe - funkcja, która dodaje do naszego zbioru jego kopię i dodaje do niej losowy szum, przez co mamy ich więcej. Jest to jeden ze sposobów augmentacji danych.

get_token_trans, tokenize_formula - są to funkcje tokenizujące wzory chemiczne z podanego zbioru danych. Opierają się na *token_dic* jako zbiorze tokenów, których pozycja w liście odpowiada ich kluczowi+10, ponieważ cyfry są zakodowane swoimi wartościami. Jedyny problem token dict jest taki, że trzeba go przenosić między wywołaniami co utrudnia z nim pracę.

init_weights Funkcja, która inicjalizuje wagi podanego modelu. Usprawnia i polepsza jakość uczenia modelu.

save_dic, load_dic rozwiazują problem z nieprzenoszeniem słownika miedzy wywołaniami. Zapisują je jako plik typu json i zapisują w miejscu gdzie wskazuje zmienna *SAVE_DIC_LOC*

5.4 Hiperparametry modelu

Ostateczne hiperparametry modelu:

1. learning rate = 0.00005 - funkcja nie wydawała się utykać w minimach lokalnych, więc zmniejszałem lr dopóki tego nie zrobiła
2. batch size = 256 - przy 512 dokładność modelu zaczęła spadać
3. optimizer = AdamW - klasyczny wybór do różnych sieci neuronowych, przy próbowaniu alternatyw dokładność spadała
4. scheduler = ReduceLROnPlateau - zmniejsza lr tylko wtedy kiedy model przestaje się uczyć co pomaga lepiej dostosować lr do danych
5. funkcja straty MSE - klasyczny wybór

5.5 Użyte biblioteki

pytorch [5] - jak sami o sobie piszą "*Naszą misją jest napędzanie wdrażania sztucznej inteligencji i głębokiego uczenia się poprzez wspieranie otwartego, neutralnego dla dostawców ekosystemu zbudowanego wokół PyTorch. Udostępniając wszystkim najnowocześniejsze narzędzia i biblioteki, dążymy do demokratyzacji innowacji w AI i ML*". Biblioteka użyta została do implementacji sieci neuronowej jak i procesu jej uczenia.

scikit-learn [6] - biblioteka do uczenia maszynowego w pythonie. W tym projekcie zastosowana została do normalizacji danych.

mendeleev [4] - biblioteka do analizy i pracy z układem okresowym pierwiastków chemicznych. Umożliwia dostęp do różnych właściwości pierwiastków, takich jak masa atomowa, elektroujemność czy promień atomowy.

pandas [7] - szybkie, wydajne, elastyczne i łatwe w użyciu narzędzie open source do analizy i manipulacji danymi, zbudowane na bazie języka programowania Python.

6 Wyniki

Wyniki najlepszego z modeli:

1. Test Loss (MAE) = 0.1415
   Średni błąd względny oznacza, że model przeciętnie myli się o te wartość w eV.
2. Mean Squared Error (MSE) = 0.0486
3. Root Mean Squared Error (RMSE) = 0.2205
4. R^2 Score: 0.9704
   Współczynnik determinacji bliski 1 sugeruje, że model bardzo dobrze wyjaśnia zmienność danych - w tym przypadku model wyjaśnia 97,04% wariancji danych co jest doskonałym wynikiem.
5. Mean Absolute Percentage Error (MAPE): 112.07%
   Wysoka wartość MAPE sugeruje, że model ma trudności z dokładnymi przewidywaniami w odniesieniu do wartości rzeczywistych. Wartości powyżej 100% oznaczają, że model znacząco się myli w stosunku do rzeczywistych wartości. Jednak w naszym przypadku wartość tą mocno zawyżają wartości bliskie 0 ($\lim_{x\to\infty} \frac{0}{x} = \infty$), w porównaniu do poprzednich wyników i tak się bardzo poprawiła.
6. Symmetric Mean Absolute Percentage Error (SMAPE): 12.96% Oznacza błąd względny modelu - 12,96% jest akceptowalnym wynikiem.

Model jest bardzo dobrze dopasowany - pokazuje to wysoki współczynnik determinacji R^2. Błędy bezwzględne są dosyć niskie, co oznacza, że modelem można się z dużym spokojem sugerować.

7 Wnioski

Z powodu małej ilości danych bardzo trudno będzie osiągnąć lepszy wynik. Dodatkowym pomysłem będzie też zastosowanie większej ilości deskryptorów - szczególnie takich, które często przybierają wartości różne od zera, albo dodać augmentacje danych nie tylko na ich ilość, ale też na ilość ich deskryptorów.

Dodatkowym pomysłem będzie zastosowanie kolejnej części modelu w formie autoencodera, który będzie próbował odtworzyć deskryptory na podstawie danych, co da dodatkowe informacje do nauki.

Podsumowując, model jest skuteczny w wyjaśnianiu ogólnych trendów i zmienności danych, ale szczegółowe wartości mogą być obarczone błędem. Zaleca się traktowanie wyników modelu jako punktu wyjścia do dalszych badań, a nie jako dokładnych prognoz.

Bibliografia

[1] Jan Wolski. *Electro affinity scrapping*. https://github.com/s26435/Electro_affinity_scrapping. Accessed: 10.03.2025.

[2] Yad Smood. *Biblioteka gorod*. https://github.com/go-rod/rod. Accessed: 10.03.2025.

[3] NIST. *NIST Chemistry WebBook*. https://github.com/go-rod/rod. Accessed: 10.03.2025.

[4] mendeleev. *mendeleev 0.20.1*. https://pypi.org/project/mendeleev/. Accessed: 14.03.2025.

[5] PyTorch Foundation. https://pytorch.org/. Accessed: 14.03.2025.

[6] David Cournapeau. *scikit-learn, Machine Learning in Python*. https://scikit-learn.org/stable/. Accessed: 14.03.2025.

[7] *pandas*. https://pandas.pydata.org/. Accessed: 14.03.2025.