ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики
(повне найменування назва факультету)

Кафедра інформаційних систем
(повна назва кафедри)

КУРСОВА РОБОТА

на тему:

Розробка системи кластеризації антитіл на основі коефіцієнту перехресного зв'язування

Студента 1 курсу, магістратури групи ПМІМ-12,
спеціальності 122 Комп'ютерні науки
Зелінського Олександра
(прізвище та ініціали)
Керівник доцент, канд. фіз.-мат. наук, Горлач В.М
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____
Кількість балів: _____Оцінка: ECTS _____

Львів – 2022

ЗМІСТ

ВСТУП

В сучасному світі комп’ютери та інша потужна обчислювальна техніка набули величезної популярності в житті кожної людини. Саме тому зручним та важливим є використання комп’ютерних алгоритмів та моделей для виконання завдань пов’язаних з дослідженнями у різних сферах науки та техніки.

Використання математичних моделей суттєво зменшує кількість рутинної тривалої та дорогої роботи в лабораторіях для хіміків, фізиків, біологів та інших науковців. Тому з часом математичні та комп’ютерні моделі почали зменшувати кількість експериментів.

В умовах нещодавньої пандемії вірусу Covid-19 надзвичайно важливими були швидкі дослідження, які несли безпосередню користь для виявлення, запобігання та лікування вірусних захворювань, в тому числі вірусу Covid-19 або SARS-CoV-2. Саме тому в цій роботі буде розглянуто задачу, яка допоможе в знаходженні двох оптимальних антитіл до будь-якого вірусу (для прикладу взято SARS-CoV-2) та можливий спосіб її вирішення за допомогою алгоритму кластеризації.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ

Вірус (*virus* – отрута) – неклітинний інфекційний агент, який може відтворюватися лише всередині живих клітин.

Антитіла, або імуноглобуліни (Ig) – білкові сполуки, які організм хребетних тварин (в тому числі людей) виробляє у відповідь на антигени, чужорідні речовини, що потрапляють до крові, лімфи або тканин організму, з метою знищити або нейтралізувати потенційно небезпечні з них – бактерії, віруси, отрути та деякі інші речовини.

У поставленій задачі дано молекулу вірусу SARS-CoV-2, до якої приєднуються два антитіла (вони можуть бути як різними так і однаковими), для того щоб можна було їх відрізнити одне з них помічається *.

Для простоти вважатимемо, що все відбувається на площині, антитіла це два круги однакового розміру, що мають невеликий „дзьоб” для взаємодії з вірусом. Антитіла в свою чергу приєднуються до меншого круга який представляє молекулу вірусу. Схематичне зображення цього процесу можна побачити на рисунку 1.

У нашому випадку конкуренція йде між антитілами з молярною вагою180 кД за зв’язування з вірусом з молярною вагою 45 кД. З цієї причини відбувається запекла конкуренція.

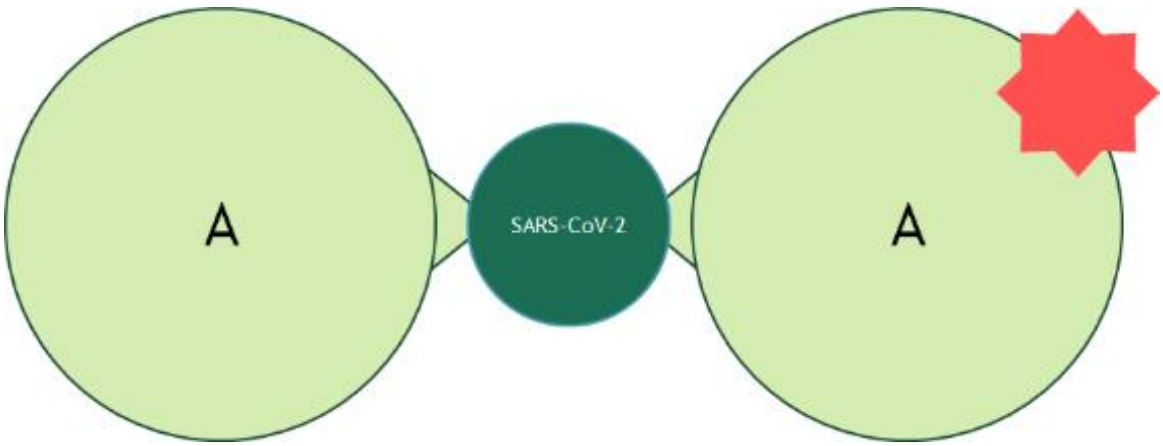

Рис. 1. Модель приєднання антитіл до вірусної молекули

Основне завдання полягає у знаходженні двох антитіл (можуть бути однаковими), таких що знаходяться на оптимальній відстані одне від одного тобто не перетинаються та не знаходяться занадто близько один до одного, щоб почати конкурувати між собою.

Зручним теоретичним методом вирішення цієї проблеми є розбиття списку антитіл на групи за ознакою того наскільки вони заважають один одному або іншими словами чи приєднуються вони до вірусу в одній і тій же області. Те що антитіла належать до різних груп означає, що вони прив’язуються в різних областях та взаємодіють краще ніж якби вони були з однієї.

В сучасних комп’ютерних науках популярним способом для розбиття даних на групи є група алгоритмів, яка називається кластерним аналізом.

Кластерний аналіз (англ. *Data clustering*) – задача розбиття заданої вибірки об'єктів (ситуацій) на підмножини, які називаються кластерами, так, щоб кожен кластер складався з схожих об’єктів, а об’єкти різних кластерів істотно відрізнялися. Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу завдань навчання без вчителя.

РОЗДІЛ 2. ОПИС ДАНИХ ТА АЛГОРИТМУ

Дані з експерименту подані у вигляді таблиці, де кожна комірка це коефіцієнт перехресного зв'язування міченого антитіла (зі стовпця) та не міченого (з рядка). В рядку позначеному як "blank" надані максимальні значення коефіцієнтів перехресного зв'язування для відповідного міченого антитіла.

Labelled	NP1501*	NP1502*	NP1503*	NP1508*	NP1510*	NP1514*	NP1516*	NP1517*	NP1518*	NP1520*	NP1521*
blank	1.089	1.067	1.3664	1.412	1.67	1.07	1.1704	1.11	1.1616	1.007	1.084
NP1501	0.449	0.715	1.0664	1.0248	1.136	0.26	0.4172	0.268	0.8512	0.614	0.596
NP1502	0.893	0.425	0.336	0.196	0.349	0.3625	0.665	0.45	1	0.188	0.535
NP1503	0.768	0.309	0.068	0.052	0.095	0.305	0.7126	0.408	0.9888	0.075	0.57
NP1507	0.422	0.856	0.7216	0.6088	0.785	0.1825	0.4256	0.154	1.0352	0.482	0.583
NP1508	0.732	0.388	0.1456	0.0848	0.19	0.345	0.8456	0.411	1.1344	0.144	0.55
NP1510	0.781	0.382	0.2152	0.1056	0.233	0.495	0.7826	0.527	1	0.227	0.661
NP1512	0.79	0.789	0.876	0.7504	0.979	0.735	1.015	0.737	0.816	0.853	0.638
NP1514	0.448	0.822	1.0968	1.0088	1.189	0.2475	0.4858	0.253	1.0208	1.04	0.626
NP1516	0.385	1.034	0.9832	0.9304	1.053	0.1775	0.3052	0.152	0.9504	0.782	0.455
NP1517	0.425	0.644	0.7952	0.7304	0.885	0.155	0.2758	0.079	1.1504	0.636	0.414
NP1518	0.517	0.636	1.0272	0.9424	1.107	0.2425	0.5502	0.34	0.2736	0.757	0.394
NP1520	0.669	0.503	0.0856	0.0536	0.106	0.315	0.5866	0.406	0.8352	0.095	0.431
NP1521	0.538	0.629	0.9264	0.9016	1.057	0.4425	0.6342	0.574	1.2048	0.851	0.107

Рис. 2. Фрагмент початкових даних

Для подальшої роботи з даними їх позначають за наступним алгоритмом:

- Цифрою 3 (темно-зеленим кольором) антитіла з поганим зв'язуванням, якщо

$$\frac{-(cell[i][j] - blank[j])}{blank[j]} > 0.75$$

- Цифрою 2 (світло-зеленим кольором) антитіла з середнім зв'язуванням, якщо

$$0.5 < \frac{-(cell[i][j] - blank[j])}{blank[j]} \leq 0.75$$

- Цифрою 1 (білим кольором) антитіла з хорошим зв'язування, у всіх інших випадках.

label	NP1501*	NP1502*	NP1503*	NP1508*	NP1510*	NP1514*	NP1516*	NP1517*
NP1501	2	1	1	1	1	3	2	3
NP1502	1	2	3	3	3	2	1	2
NP1503	1	2	3	3	3	2	1	2
NP1507	2	1	1	2	2	3	2	3
NP1508	1	2	3	3	3	2	1	2
NP1510	1	2	3	3	3	2	1	2
NP1512	1	1	1	1	1	1	1	1
NP1514	2	1	1	1	1	3	2	3
NP1516	2	1	1	1	1	3	2	3
NP1517	2	1	1	1	1	3	3	3

Рис. 3. Фрагмент позначених даних

Тепер задача полягає у розбитті матриці перехресного зв'язування антитіл на групи за ознакою подібності раніше створеного показника зв'язування для полегшення виявлення оптимальних пар та приблизної локалізації місця зв'язування. Для цього використовують методи кластеризації, а саме модифікація методу k-means для якісних змінних що називається k-modes.

k-means (k-середніх) – це популярний алгоритм кластеризації на основі центроїдів, який розділяє дані представлені у вигляді точок на k кластерів, кожен із яких має майже рівну кількість цих точок. Ідея цього алгоритму кластеризації полягає в тому, щоб знайти k центроїд, де кожна точка з набору даних буде належати будь-якій з k-множин з найближчим (найчастіше мінімальна евклідова відстань) до нього середнім значенням.

k-modes метод, розроблений Хуагном в 1998 році, що визначає кластери на основі відповідності категорій між точками даних. В цьому алгоритмі:

а) відстань між двома точками (проста міра не схожості) даних X та Y описується як сума не схожих елементів:

$$d_1(X,Y) = \sum_{i=1}^{n} \delta(x_i, y_i),$$

де

$$\delta(x_i, y_i) = \begin{cases} 0, & x_i = y_i \\ 1, & x_i \neq y_i \end{cases}$$

б) Середнє змінюється на моду

в) Мода знаходиться на основі частоти

Нижче наведено кроки для кластеризації на основі k-modes:

1. Виберіть k початкових мод
2. Розділіть елементи на кластери на основі простої міри несхожості відносно початкових мод. Оновлюйте моду кожного з кластерів після додавання нового елементу.
3. Після того, як усі елементи були віднесені до кластера, перевірте значення несхожості кожного спостереження з модою. Якщо виявляється, що найближча мода знаходиться в іншому кластері, перемістіть елемент у відповідний кластер і оновіть моду обох кластерів.
4. Повторюйте крок 3, доки жоден із елементів не змінить кластер на інший

Метод k-modes працює лише при початкові відомій кількості кластерів, оскільки вона нам не відома, то для визначення оптимальної кількості кластерів використовується ліктьовий метод (elbow method), який для різних значень k буде вибирати значення k у тій точці, де значення істотно не зменшується зі збільшенням значення k. Метод модифікований, для того щоб використовувати різницю всередині кластера (within-cluster difference).

$$WCD = \sum_{j=1}^{k} \sum_{i=1}^{m} d_1(x_i, y_c),$$

де WCD – різниця всередині кластера, k – кількість кластерів, m – кількість спостережень у кожному кластері, c – центроїд кластера, а d_1 – проста міра несхожості.

У якості початкового розбиття на кластери використовується алгоритм розроблений Fuyuan Cao в 2009 році. Цей метод ініціалізації для категоріальних даних, у якому враховують відстань між об’єктами та щільність об’єкта, що визначається на основі частоти значень атрибутів.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

Веб застосунок розроблений в результаті виконання курсової роботи складається з трьох частин:

1. Інтерфейс користувача, написаний на мові програмування TypeScript з використанням бібліотеки React, який реалізує функції:
    a. Вивантаження файлу з початковими даними
    b. Перегляд результатів
    c. Завантаження .xlsx файлу з результатами
2. Прикладний програмний інтерфейс (API), який відповідає за кластеризацію написаний на мові програмування Python 3. Для завантаження та обробки даних використовувалась бібліотека pandas, для візуалізації бібліотека matplotlib, для кластеризації kmodes та kneed для знаходження оптимальної кількості кластерів.
3. Прикладний програмний інтерфейс, що поєднує між собою два попередні пункти та реалізує перетворення даних для відображення та збереження результатів. Цей інтерфейс написаний на мові NET 5, та ASP.NET Core 5.0.

Архітектура веб застосунку зображена на рисунку 4.

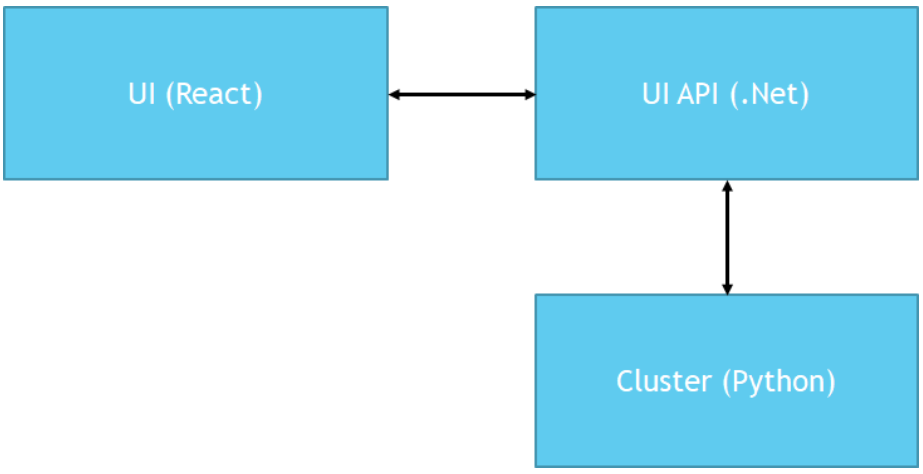

Рис. 4. Архітектура веб застосунку

РОЗДІЛ 4. РЕЗУЛЬТАТИ

Розроблений веб застосунок має декілька сторінок на рисунку 5 зображено початкову сторінку на якій можна завантажити файл з даними про перехресне зв’язування антитіл.

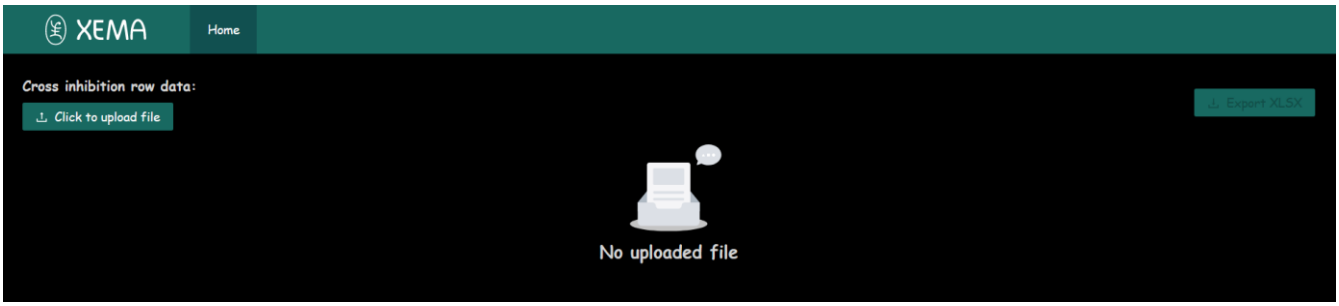

Рис. 5. Початкова сторінка веб застосунку

Після завантаження файлу відбувається його відправка на API яке відповідає за кластеризацію. Спочатку дані маркуються кольорами з гідно з алгоритмом наведеним в розділі 2. Після цього знаходиться відстань в середині кластера для k – кількість кластерів від 0 до розмір вибірки та на основі цих даних будується графік. Для наших даних видно, що оптимальна кількість кластерів буде 9.

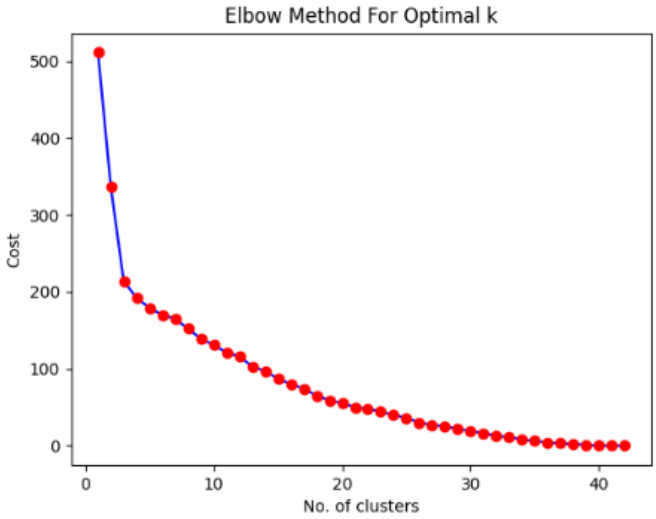

Рис. 6. Графік кількості кластерів до відстані в середині кластера

Коли оптимальна кількість кластерів знайдена, відбувається безпосередня кластеризація методом k-modes та відправка результатів на інтерфейс користувача. На інтерфейсі користувача дані відображаються у два способи:

1. У вигляді карток де заголовок вказує на групу, а в тілі міститься посортований список антитіл (рис. 7).
2. У вигляді початкових даних, позначеними кольорами (рис. 8). Співпадіння кольорів в тій чи іншій колонці в середині певного кластеру підтверджує, що вони справді належать до однієї групи.

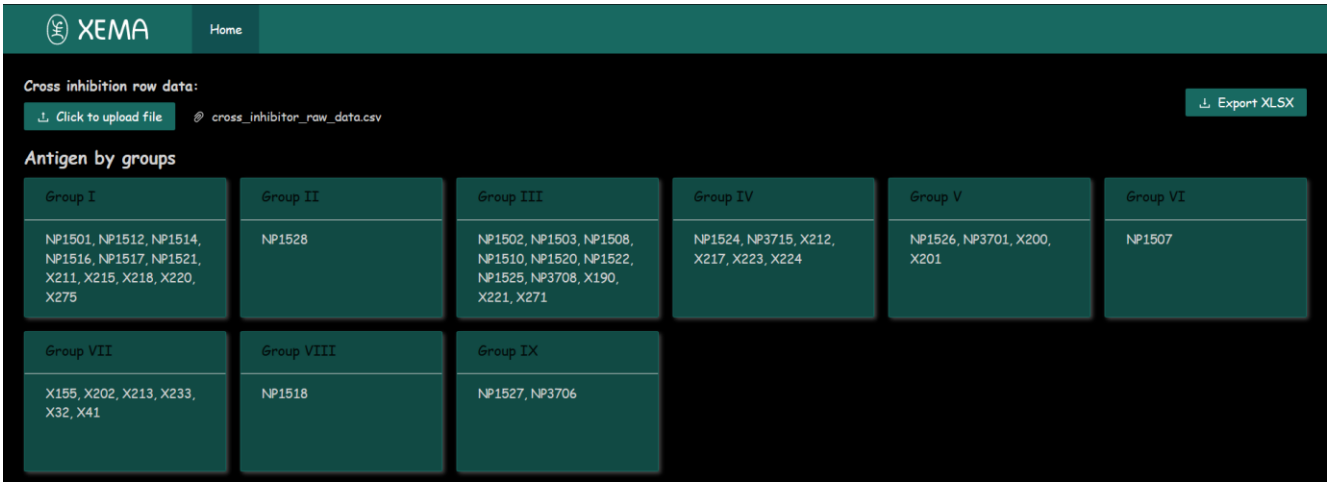

Рис. 7. Результат кластеризації у вигляді карток

Рис. 8. Фрагмент початкових даних розбитих на групи

Групи	Під групи	Елементи
1	A	NP1501, NP1514, NP1516, NP1517, NP1507
	B	X190, NP1526, X200, X201
1B/2		NP1512, NP1521
2		NP1502, NP1503, NP1508, NP1510, NP1520, NP1522, NP1525, X221, X271, NP3701, NP3708
2B/3		NP1528
3	A	X202, X218, NP1518, NP1527
	B	X32, X155, X41, X212, X213, X217, X223, X224, X233, NP1524, NP3715
4	A	NP3706
	B	X211
	C	X215
5		X220
6		X275

Табл. 1. Очікуваний результат

Кластери	Елементи
1	NP1501, NP1512, NP1514, NP1516, NP1517, NP1521, X211, X215, X218, X220, X275
2	NP1528
3	NP1502, NP1503, NP1508, NP1510, NP1520, NP1522, NP1525, NP3708, X190, X221, X271
4	NP1524, NP3715, X212, X217, X223, X224
5	NP1526, NP3701, X200, X201
6	NP1507
7	X155, X202, X213, X233, X32, X41
8	NP1518
9	NP1527, NP3706

Табл. 2. Отриманий в результаті кластеризації результати

В результаті експериментальних досліджень доведено, що для даного вірусу найкращий результат дає пара де одне антитіло з групи 3B (X155, X41, X213 та X32), а інше з групи 4A (NP3706). Можемо побачити що розглянутий в цій роботі алгоритм кластеризації добре, розділив саме ці антитіла на групи.

Також реалізовано збереження результатів у форматі .xlsx. Ці результати подані у вигляді таблиці схожі на результати подані на веб сторінці та складаються з:

1. Таблиці де заголовок вказує на групу, а в тілі міститься посортований список антитіл (рис. 9).
2. Таблиця з початковими даними розбита на групи та позначена кольорами (рис. 10).

	1	2	3	4	5	6	7	8	9	10
1										
2		Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
3		NP1501	NP1528	NP1502	NP1524	NP1526	NP1507	X155	NP1518	NP1527
4		NP1512		NP1503	NP3715	NP3701		X202		NP3706
5		NP1514		NP1508	X212	X200		X213		
6		NP1516		NP1510	X217	X201		X233		
7		NP1517		NP1520	X223			X32		
8		NP1521		NP1522	X224			X41		
9		X211		NP1525						
10		X215		NP3708						
11		X218		X190						
12		X220		X221						
13		X275		X271						

Рис. 9. Результат кластеризації записаний у файл

		NP1501*	NP1502*	NP1503*	NP1508*	NP1510*	NP1514*	NP1516*	NP1517*	NP1518*	NP1520*	NP1521*	NP1522*	NP1524*	NP1525*	NP1527*	NP3706*
Group 1	NP1501	0.449	0.715	1.0664	1.0248	1.136	0.26	0.4172	0.268	0.8512	0.614	0.596	0.7326	1.1392	1.061	0.852	0.9114
	NP1512	0.79	0.789	0.876	0.7504	0.979	0.735	1.015	0.737	0.816	0.853	0.638	0.7101	1.1096	0.836	0.841	0.7623
	NP1514	0.448	0.822	1.0968	1.0088	1.189	0.2475	0.4858	0.253	1.0208	1.04	0.626	0.7821	1.2512	1.136	1.024	0.84
	NP1516	0.385	1.034	0.9832	0.9304	1.053	0.1775	0.3052	0.152	0.9504	0.782	0.455	0.6399	1.0936	0.99	0.752	0.7833
	NP1517	0.425	0.644	0.7952	0.7304	0.885	0.155	0.2758	0.079	1.1504	0.636	0.414	0.5427	1.0624	0.92	0.71	0.8673
	NP1521	0.538	0.629	0.9264	0.9016	1.057	0.4425	0.6342	0.574	1.2048	0.851	0.107	0.729	1.0872	1.058	0.754	1.071
	X211	0.64	0.861	1.2072	1.1376	1.365	0.765	0.8428	0.836	0.7184	1.241	0.598	0.9387	0.7984	1.333	0.215	0.8043
	X215	0.615	0.824	1.1512	1.1752	1.344	0.605	0.4928	0.665	0.9296	1.017	0.339	0.9045	1.0648	1.332	1.23	0.9345
	X218	0.424	0.729	1.1096	1.1384	1.228	0.5475	0.4956	0.653	0.6256	0.968	0.39	0.9072	0.4816	1.219	0.947	0.6825
	X220	0.508	0.696	0.9744	0.9952	1.131	0.4375	0.4844	0.59	0.9872	0.828	0.393	0.8703	0.7072	1.173	0.616	0.8337
	X275	0.437	0.575	0.7456	0.6936	0.85	0.5575	0.3836	0.708	1.0864	0.718	0.378	0.5904	0.9064	0.915	0.511	0.8526
Group 2	NP1528	1.103	0.865	0.2632	0.1912	0.231	0.3075	0.5474	0.445	0.6592	0.258	0.273	0.2736	0.316	0.257	1.075	0.7728
Group 3	NP1502	0.893	0.425	0.336	0.196	0.349	0.3625	0.665	0.45	1	0.188	0.535	0.1719	1.1584	0.223	0.802	0.9303
	NP1503	0.768	0.309	0.068	0.052	0.095	0.305	0.7126	0.408	0.9888	0.075	0.57	0.063	1.1224	0.076	0.858	0.8169
	NP1508	0.732	0.388	0.1456	0.0848	0.19	0.345	0.8456	0.411	1.1344	0.144	0.55	0.1017	1.1632	0.135	0.806	0.8736
	NP1510	0.781	0.382	0.2152	0.1056	0.233	0.495	0.7826	0.527	1	0.227	0.661	0.1332	1.1216	0.146	0.836	0.8505
	NP1520	0.669	0.503	0.0856	0.0536	0.106	0.315	0.5866	0.406	0.8352	0.095	0.431	0.0702	1.028	0.069	0.765	0.8274
	NP1522	0.669	0.384	0.1608	0.0912	0.17	0.4725	0.7476	0.574	1.0064	0.205	0.441	0.1143	1.0672	0.128	0.795	0.9093
	NP1525	1.003	0.389	0.1272	0.0872	0.148	0.765	0.9506	0.67	0.9728	0.166	0.672	0.126	0.892	0.123	0.872	0.9429
	NP3708	0.676	0.378	0.2016	0.1336	0.254	0.185	0.392	0.175	0.9568	0.237	0.624	0.1476	0.9704	0.189	0.735	0.8484
	X190	0.442	0.386	0.0576	0.0448	0.068	0.3275	0.4452	0.239	0.904	0.077	0.349	0.063	0.9232	0.068	0.82	0.7581
	X221	0.646	0.343	0.2072	0.1576	0.245	0.56	0.6818	0.603	0.9632	0.237	0.484	0.1719	0.9912	0.209	1.348	0.8274
	X271	0.487	0.378	0.0808	0.0552	0.104	0.5375	0.4648	0.538	0.9472	0.109	0.399	0.0747	0.8736	0.084	0.81	0.8967
Group 4	NP1524	0.604	0.948	1.1128	1.0584	1.194	0.7425	0.9926	0.833	0.456	1.185	0.675	0.8163	0.0728	1.123	0.712	0.6027
	NP3715	0.605	0.551	0.7296	0.6112	0.848	0.41	0.7686	0.554	0.2864	0.855	0.612	0.522	0.0712	0.725	0.768	0.6762
	X212	0.712	0.805	1.2536	1.2192	1.414	0.87	0.9436	0.872	0.2224	1.268	0.664	0.972	0.0728	1.406	1.94	0.8085
	X217	0.476	0.733	0.9376	0.956	1.072	0.5325	0.4494	0.609	0.2096	0.844	0.305	0.7965	0.0568	1.124	0.738	0.5775
	X223	0.694	0.79	1.0064	1.0096	1.103	0.745	0.7616	0.849	0.2016	1.113	0.539	0.7848	0.06	1.14	1.618	0.7665
	X224	0.895	0.939	1.0288	0.9728	1.169	0.8475	0.9408	0.882	0.2304	0.825	0.562	0.7947	0.0656	1.154	2.033	0.8295

Рис. 10. Фрагмент початкових даних розбитих на групи та записаних у файл

ВИСНОВКИ

В результаті виконання цієї курсової було розроблено веб застосунок, який полегшує задачу розбиття списку антитіл на групи на основі індексу перехресного зв’язування. Цей застосунок представляє з себе три мікросервіси:

1. Інтерфейс користувача для вивантаження файлів, перегляду та завантаження результатів.
2. Прикладний програмний інтерфейс, який відповідає за кластеризацію
3. API, що поєднує інтерфейс для кластеризацію та інтерфейс користувача, а також реалізує завантаження результатів.

Для розробки системи було використано NET 5 з ASP.NET, Python 3 з Flask та Type Script з React.

Також було розділено антитіла для конкретного вірусу. Зважаючи на те, які антитіла опинились в однакових групах порівнюючи очікуваний та отриманий результат можна сказати, що правильність розбиття на кластери склала 70%.

Важливим аспектом, є те, що попри невеликий об’єм даних (матриця 40x30) розроблений застосунок за невеликий час (1-2 хвилини) робить об’єм роботи, для виконання якого людині потрібно було б декілька днів. Це є дуже важливо, оскільки при збільшенні вибірки кількість часу, що потрібен на виконання алгоритму комп’ютером буде залишатись невеликим в порівнянні з часом який потрібен буде людині для виконання такого ж завдання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Зелінський О. Розробка системи кластеризації антитіл на основі коефіцієнту перехресного зв’язування / О. Зелінський, В. Горлач, Ю. Лебедін // Міжнародна студентська наукова конференція з питань прикладної математики та комп’ютерних наук (МСНКПМК-2022), 5-6 травня 2022 р. – Львів:2022. – С. 8-12. Режим доступу: https://ami.lnu.edu.ua/wp-content/uploads/2022/05/ISSCAMCS-2022.pdf
2. Satyam Kumar Clustering Algorithm for data with mixed Categorical and Numerical features [Electronic resource]. – 2021. – URL: https://towardsdatascience.com/clustering-algorithm-for-data-with-mixed-categorical-and-numerical-features-d4e3a48066a0
3. Z. Huang. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values (1998). Data Mining and Knowledge Discovery. 2(3): 283–304.
4. Python: K-modes explanation [Electronic resource]. – 2017. – URL: https://stackoverflow.com/questions/42639824/python-k-modes-explanation
5. Audhi Aprilliant The k-modes as Clustering Algorithm for Categorical Data Type [Electronic resource]. – 2021. – URL: https://medium.com/geekculture/the-k-modes-as-clustering-algorithm-for-categorical-data-type-bcde8f95efd7.
6. Fuyuan Cao A new initialization method for categorical data clustering, Fuyuan Cao, Jiye Liang, Liang Bai [Electronic resource]. – 2009. – URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.474.8181&rep=rep1&type=pdf

ДОДАТКИ

Додаток А. Код який відповідає за кластеризацію

```
import matplotlib.pyplot as plt
import numpy as np
import numpy.typing as npt
import pandas as pd
from kmodes.kmodes import KModes  # pip install kmodes
from kneed import KneeLocator  # pip install kneed

from constants import cluster_column_name, labelled_column_name

# 1 to show and 0 to hide
verbose = 1


def setup_kmodes(num_clusters: int, data: pd.DataFrame) -> KModes:
    """
    Setup KModes
    :param num_clusters: amount of clusters
    :return: instance of KModes
    """
    return KModes(n_clusters=num_clusters, init="Cao", n_init=15,
verbose=verbose)


def build_elbow_curve(data: pd.DataFrame) -> tuple[range, list[int]]:
    """
    Build Elbow curve to find optimal cluster amount
    :param data: data to be clustered
    :return: range from 1 to max amount of clusters, cost of each clustered
model
    """
    cost = []
    cluster_amount_range = range(1, data.shape[0])
    for num_clusters in list(cluster_amount_range):
        kmodes = setup_kmodes(num_clusters, data)
        kmodes.fit_predict(data)
        cost.append(kmodes.cost_)

    if verbose == 1:
        build_elbow_plot(cluster_amount_range, cost)
    return cluster_amount_range, cost


def build_elbow_plot(cluster_amount_range: range, cost: list[int]):
    """
    Build and show Elbow curve plot
    :param cluster_amount_range: range from 1 to max amount of clusters
    :param cost: cost of clustering
    """
    plt.plot(cluster_amount_range, cost, 'o-', color="blue",
markerfacecolor='red', markeredgecolor='red')
    plt.xlabel('No. of clusters')
    plt.ylabel('Cost')
    plt.title('Elbow Method For Optimal k')
    plt.show()


def get_optimal_cluster_amount(data: pd.DataFrame) -> int:
    """
```

```
    Get exact cluster amount
    :param data: data to be clustered
    :return: optimal cluster amount base on elbow method
    """
    cluster_amount_range, cost = build_elbow_curve(data)
    kl = KneeLocator(cluster_amount_range, cost, curve="convex",
direction="decreasing")
    exact_cluster_amount = kl.elbow
    return exact_cluster_amount


def get_clusters_for_optimal_model(data: pd.DataFrame, cluster_amount: int) ->
npt.NDArray[np.uint16]:
    """
    Build optimal model with 'cluster_amount' clusters
    :param cluster_amount: cluster amount
    :param data: data to be clustered
    :return: list of cluster number for each data row
    """
    kmodes = setup_kmodes(cluster_amount, data)
    clusters = kmodes.fit_predict(data)
    return clusters


def format_response(data: pd.DataFrame, clusters: npt.NDArray[np.uint16]) ->
pd.DataFrame:
    """
    Format response data
    :param data: data to be clustered
    :param clusters: list of cluster number for each data row
    :return: json with clustered data
    """
    data.insert(0, cluster_column_name, clusters, True)
    data = data.reset_index()
    data = data.sort_values(by=[cluster_column_name, labelled_column_name])
    data = data.set_index(labelled_column_name)
    return data


def clustering(data: pd.DataFrame) -> pd.DataFrame:
    """
    Clustering data
    :param data: data to be clustered
    :return: json with clustered data
    """
    cluster_amount = get_optimal_cluster_amount(data)
    print(f"Optimal cluster amount {cluster_amount}")
    clusters = get_clusters_for_optimal_model(data, cluster_amount)
    response = format_response(data, clusters)
    return response
```

Додаток Б. Код який відповідає за збереження результатів в Excel

```
public class ExcelService : IExcelService
    {
        private readonly IStyleProvider _styleProvider;

        public ExcelService(IStyleProvider styleProvider)
        {
            _styleProvider = styleProvider;
        }

        public MemoryStream GetFileStream(CrossInhibitorRawDataModel dataModel)
        {
            var sl = new SLDocument();

            var rowIndex = 2;
            var columnIndex = 2;

            var groupHeaderStyle = _styleProvider.GetGroupHeaderStyle();
            var groupCellStyle = _styleProvider.GetGroupCellStyle();
            var lastGroupCellStyle = _styleProvider.GetGroupLastCellStyle();
            var darkGreenStyle =
_styleProvider.GetFilledCellStyle(System.Drawing.Color.FromArgb(0, 176, 79));
            var lightGreenStyle =
_styleProvider.GetFilledCellStyle(System.Drawing.Color.FromArgb(146, 208, 80));

            // Add clusters to file
            foreach (var keyValue in dataModel.Clusters)
            {
                sl.SetCellValue(rowIndex, columnIndex, $"Group {keyValue.Key + 1}");
                sl.SetCellStyle(rowIndex, columnIndex, groupHeaderStyle);

                rowIndex++;
                for (var i = 0; i < keyValue.Value.Count; i++)
                {
                    var item = keyValue.Value[i];
                    sl.SetCellValue(rowIndex, columnIndex, item);

                    if (i == keyValue.Value.Count - 1)
                    {
                        // Add border in the bottom to close table
                        sl.SetCellStyle(rowIndex, columnIndex, lastGroupCellStyle);
                    }
                    else
                    {
                        sl.SetCellStyle(rowIndex, columnIndex, groupCellStyle);
                    }

                    rowIndex++;
                }

                columnIndex++;
                rowIndex = 2;
            }

            var intialRawDataRow = rowIndex + dataModel.Clusters.Values.ToList().Max(x
=> x.Count) + 2;

            // Add marked antigen labels to file
            rowIndex = intialRawDataRow;
            columnIndex = 3;
            foreach (var label in dataModel.MarkedAntigenLabels)
            {
                sl.SetCellValue(rowIndex, columnIndex, label);
```

```
                columnIndex++;
            }

            // Add antigen labels to file
            rowIndex++;
            columnIndex = 1;
            for (var i = 0; i < dataModel.AntigenLabels.Count; i++)
            {
                sl.SetCellValue(rowIndex, columnIndex, $"Group {i + 1}");
                columnIndex++;

                var labelGroup = dataModel.AntigenLabels[i];
                foreach (var label in labelGroup)
                {
                    sl.SetCellValue(rowIndex, columnIndex, label);
                    rowIndex++;
                }

                rowIndex++;
                columnIndex = 1;
            }

            // Add cross inhibition indexes to file
            rowIndex = intialRawDataRow + 1;
            columnIndex = 3;
            foreach (var clusterGroup in dataModel.CrossInhibitionIndexes)
            {
                foreach (var row in clusterGroup)
                {
                    foreach (var cell in row)
                    {
                        sl.SetCellValue(rowIndex, columnIndex, cell.Value);
                        switch (cell.MarkerColor)
                        {
                            case InhibitionColors.DarkGreen:
                                sl.SetCellStyle(rowIndex, columnIndex,
darkGreenStyle);
                                break;
                            case InhibitionColors.LightGreen:
                                sl.SetCellStyle(rowIndex, columnIndex,
lightGreenStyle);
                                break;
                            default:
                                break;
                        }

                        columnIndex++;
                    }

                    rowIndex++;
                    columnIndex = 3;
                }

                rowIndex++;
            }

            using var stream = new MemoryStream();
            sl.SaveAs(stream);

            return stream;
        }
    }
```

```
public class StyleProvider : IStyleProvider
    {
        public SLStyle GetGroupHeaderStyle()
        {
            var style = new SLStyle();
            style.SetTopBorder(BorderStyleValues.Thin,
SLThemeColorIndexValues.Dark1Color);
            style.SetLeftBorder(BorderStyleValues.Thin,
SLThemeColorIndexValues.Dark1Color);
            style.SetRightBorder(BorderStyleValues.Thin,
SLThemeColorIndexValues.Dark1Color);
            style.SetBottomBorder(BorderStyleValues.Thin,
SLThemeColorIndexValues.Dark1Color);
            style.SetHorizontalAlignment(HorizontalAlignmentValues.Center);

            return style;
        }

        public SLStyle GetGroupCellStyle()
        {
            var style = new SLStyle();
            style.SetLeftBorder(BorderStyleValues.Thin,
SLThemeColorIndexValues.Dark1Color);
            style.SetRightBorder(BorderStyleValues.Thin,
SLThemeColorIndexValues.Dark1Color);

            return style;
        }

        public SLStyle GetGroupLastCellStyle()
        {
            var style = new SLStyle();
            style.SetLeftBorder(BorderStyleValues.Thin,
SLThemeColorIndexValues.Dark1Color);
            style.SetRightBorder(BorderStyleValues.Thin,
SLThemeColorIndexValues.Dark1Color);
            style.SetBottomBorder(BorderStyleValues.Thin,
SLThemeColorIndexValues.Dark1Color);

            return style;
        }

        public SLStyle GetFilledCellStyle(System.Drawing.Color color)
        {
            var style = new SLStyle();
            style.Fill.SetPattern(PatternValues.Solid, color, color);

            return style;
        }
    }
```