НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

**КУРСОВА РОБОТА**

з дисципліни «Аналіз даних в інформаційних системах»

на тему: «Кластеризація даних про ціни на житло у Каліфорнії. Методи Kmeans та AgglomerativeClustering»

Студента 2 курсу групи ІП-11

Спеціальності: 121

«Інженерія програмного забезпечення»

Головні Олександра Ростиславовича

«ПРИЙНЯВ» з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

доц. Ліхоузова Т.А. / доц. Олійник Ю.О.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Підпис                    Дата

Київ - 2023 рік

Національний технічний університет України “КПІ ім. Ігоря Сікорського”

Кафедра інформатики та програмної інженерії

Дисципліна Аналіз даних в інформаційно-управляючих системах

Спеціальність 121 "Інженерія програмного забезпечення"

Курс 2 Група ІП-11 Семестр 4

**ЗАВДАННЯ**

**на курсову роботу студента**

|  |
| --- |
| Головні Олександра Ростиславовича |

|  |  |
| --- | --- |
| 1.Тема роботи | Кластеризація даних про ціни на житло у Каліфорнії. Методи |
| Kmeans та AgglomerativeClustering | |
|  | |

|  |  |
| --- | --- |
| 2.Строк здачі студентом закінченої роботи |  |

|  |  |
| --- | --- |
| 3. Вхідні дані до роботи | методичні вказівки до курсової роботи, обрані дані з сайту |
| https://www.kaggle.com/datasets/camnugent/california-housing-prices | |
|  | |

4.Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

|  |
| --- |
| 1.Постановка задачі |
| 2.Аналіз предметної області |
| 3.Розробка сховища даних |
| 4.Інтелектуальний аналіз даних |

5.Перелік графічного матеріалу ( з точним зазначенням обов’язкових креслень )

|  |  |
| --- | --- |
|  | |
|  | |
|  | |
|  | |
| 6.Дата видачі завдання | 16.04.2022 |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Назва етапів курсової роботи | Термін виконання етапів роботи | Підписи керівника, студента |
| 1. | Отримання теми курсової роботи | 16.04.2022 |  |
| 2. | Визначення зовнішніх джерел даних | … |  |
| 3. | Пошук та вивчення літератури з питань курсової роботи |  |  |
| 4. | Розробка моделі сховища даних |  |  |
| 4. | Розробка ETL процесів |  |  |
| 5. | Обґрунтування методів інтелектуального аналізу даних |  |  |
| 6. | Застосування та порівняння ефективності методів інтелектуального аналізу даних |  |  |
| 7. | Підготовка пояснювальної записки |  |  |
| 8. | Здача курсової роботи на перевірку |  |  |
| 9. | Захист курсової роботи | 19.06.2022 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  |  | Головня О.Р. |
|  | (підпис) |  | (прізвище, ім’я, по батькові) |

|  |  |  |  |
| --- | --- | --- | --- |
| Керівник |  |  | доц. Ліхоузова Т.А |
|  | (підпис) |  | (прізвище, ім’я, по батькові) |
| Керівник |  |  | доц. Олійник Ю.О. |
|  | (підпис) |  | (прізвище, ім’я, по батькові) |

"26" червня 2022 р.

**АНОТАЦІЯ**

Пояснювальна записка до курсової роботи: 33 сторінки, 17 рисунків, 7 посилань.

Предмет дослідження: створення програмного забезпечення, що проводить аналіз даних з їх кластеризацією та графічним відображенням результатів.

Мета роботи: проектування та реалізація сховища даних та ETL процесів, а також реалізація програмного забезпечення для отримання даних зі сховища та їх подальшого аналізу та прогнозування.

Дана курсова робота включає в себе: постановку задачі, аналіз предметної

області, роботу з даними, аналіз обраних методів для кластеризації та їх порівняння, їх графічного відображення та прогнозування за допомогою різних моделей.

ДАТАСЕТ, АНАЛІЗ ДАНИХ, ELBOW METHOD, K-MEANS CLUSTERING, AGGLOMERATIVE CLUSTERING

Оглавление

[**Вступ** 5](#_Toc133879995)

[**1.Постановка задачі** 6](#_Toc133879996)

[**2.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ** 7](#_Toc133879997)

[**3. РОБОТА З ДАНИМИ** 8](#_Toc133879998)

[3.1 Опис обраних даних 8](#_Toc133879999)

[K-means method 8](#_Toc133880000)

[3.2.1 Перевірка даних 8](#_Toc133880001)

[3.2.2 Попередня обробка даних 9](#_Toc133880002)

[3.2.3 Створення моделі 10](#_Toc133880003)

[3.2.4 Візуалізація 11](#_Toc133880004)

[AgglomerativeClustering method 14](#_Toc133880005)

[3.3.1 Візуалізація 14](#_Toc133880006)

[**ПОРІВНЯННЯ МЕТОДІВ** 18](#_Toc133880007)

[**ВИСНОВКИ** 20](#_Toc133880008)

[**ПЕРЕЛІК ПОСИЛАНЬ** 21](#_Toc133880009)

[**Додаток А Тексти програмного коду** 22](#_Toc133880010)

**Вступ**

Кластеризація даних є потужним інструментом для організації великих обсягів інформації у логічні групи, які можуть бути подальше використані для виявлення закономірностей та зроблення узагальнень. У даній курсовій роботі я розгляну процес кластеризації даних про ціни на житло у Каліфорнії. Дані про ціни на житло у Каліфорнії є дуже різноманітними, оскільки ціни можуть значно відрізнятися в залежності від різних факторів, таких як географічне розташування, розмір будинку, вік будівлі тощо.

Отже, використання кластеризації даних може допомогти зробити ці дані більш зрозумілими та використовувати їх для прийняття рішень в галузі нерухомості. У даній роботі ми описуємо процес виконання кластеризації даних, а також вивчаємо результати, щоб зрозуміти, як краще використовувати цю інформацію для прийняття рішень.

Для реалізації буде використано мову програмування Python3. з використанням бібліотек Pandas, Seaborn, Matplotlib, Sklearn та інші.

**1.Постановка задачі**

Виконання даної курсової роботи потребує виконання декількох задач:

1. Розуміння структури та характеристик набору даних.
2. Виконання попереднього аналізу даних та підготовка їх для кластеризації.
3. Вибір методу кластерного аналізу, який найкраще підходить для розглянутих даних.
4. Виконання кластерного аналізу та групування даних за різними критеріями.
5. Вивчення результатів кластерного аналізу та їх інтерпретація.
6. Визначення взаємозв'язку між кластерами та їх характеристиками.
7. Розробка рекомендацій для подальшого використання кластеризації даних у галузі нерухомості.

Отже, мета курсової роботи полягає у застосуванні методів кластерного аналізу для групування даних про ціни на житло у Каліфорнії та вивченні результатів їх аналізу для прийняття рішень у галузі нерухомості.

Для кластеризації буде використано метод K-Means Clustering, The Elbow Method. Для методу проаналізувати результати та в кінці порівняти результати.

Використати мову програмування Python 3 для реалізації.

Курсовий проект здати до дедлайну (початок сесії) та виконати у єдиному стилі написання коду (coding style).

**2.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

Предметна область даної курсової роботи - ціни на житло в Каліфорнії. Нерухомість в Каліфорнії є дуже популярною та високо цінується через свій клімат, культурне спадщину та економічний потенціал. Однак, ціни на житло в різних районах Каліфорнії значно відрізняються, що може бути пов'язано з різними факторами, такими як географічне розташування, розмір будинку, вік будівлі, наявність комунікацій та ін. В сучасному світі, з використанням інформаційних технологій, аналіз цін на нерухомість стає все більш популярним. Кластерний аналіз є одним з методів, які використовуються для обробки та аналізу даних про ціни на житло. Цей метод дозволяє групувати подібні набори даних у кластери, що дозволяє зробити більш об'єктивні висновки про тенденції на ринку нерухомості.

У програмному забезпеченні буде реалізовано наступну функціональність, що включає в себе:

1. Виконання попереднього аналізу даних та підготовка їх для кластеризації.
2. Вибір методу кластерного аналізу, який найкраще підходить для розглянутих даних.
3. Виконання кластерного аналізу та групування даних за різними критеріями.
4. Вивчення результатів кластерного аналізу та їх інтерпретація.
5. Визначення взаємозв'язку між кластерами та їх характеристиками.

**3. РОБОТА З ДАНИМИ**

3.1 Опис обраних даних

Для вирішення поставленної перед нами задачі був обраний «California Housing Prices Datasets». Він містить середні ціни на житло для округів Каліфорнії. Тому цей набір даних дуже підходить для кластеризації цін на житло за допомогою k-means method and the elbow method.

**Стовпці наведені нижче, їхні назви досить зрозумілі:**

Longitude - довгота

Latitude - широта

housing\_median\_age – житло\_середній\_вік

total\_rooms – всього-кімнат

total\_bedrooms – всього спален

population - населення

households – домашніх господарств

median\_income – середній дохід

median\_house\_value -медіана буд.

ocean\_proximity – близькість океану

K-means method

3.2.1 Перевірка даних

Для роботи з даними на мові Python ми використовуємо бібліотеку «pandas».

Імпорт бібліотек і читання набору даних

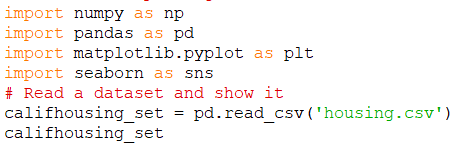


Рисунок 3. – Імпорт бібліотек і читання набору даних

3.2.2 Попередня обробка даних

Для початку ми зчитуємо дані з файлу та виводимо основну інформацію про наш датафрейм

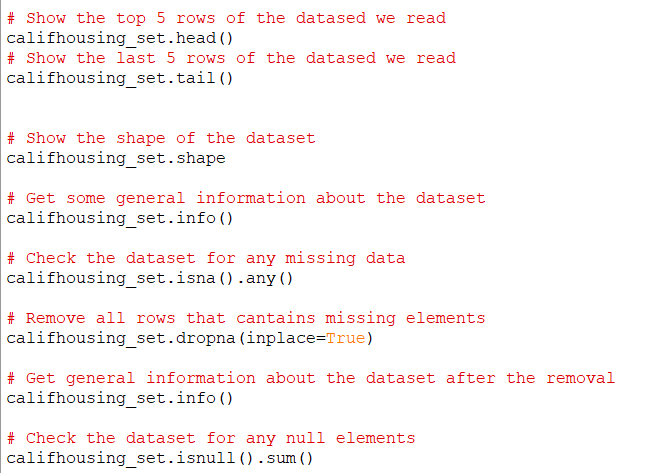
****

Рисунок 3.1 – Обробка даних

Отримаємо загальну інформацію про набір даних після видалення

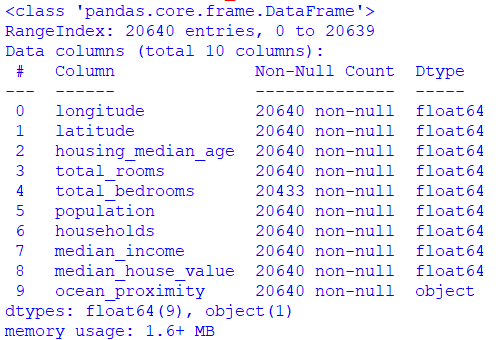


Рисунок 3.2 – Загальна інформація про датафрейм

Створюємо DataFrame зі стовпців "total\_rooms" і "median\_house\_value", які ми будемо використовувати як features  


Рисунок 3.3 – DataFrame зі стовпців "total\_rooms" і "median\_house\_value"

3.2.3 Створення моделі

Імпортуємо Kmeans. inertia\_ — це метод, який поділяє точки даних на кластери. Знаходимо оптимальну кількість кластерів

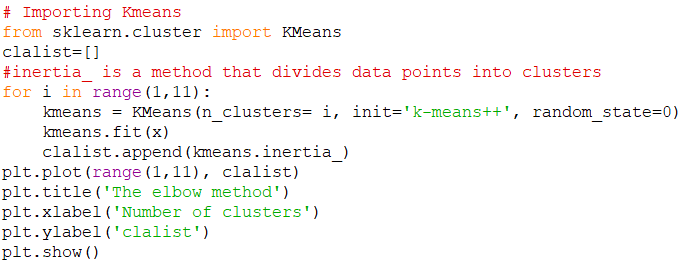


Рисунок 3.4 – Код виконання пошуку оптимальної кількості кластерів

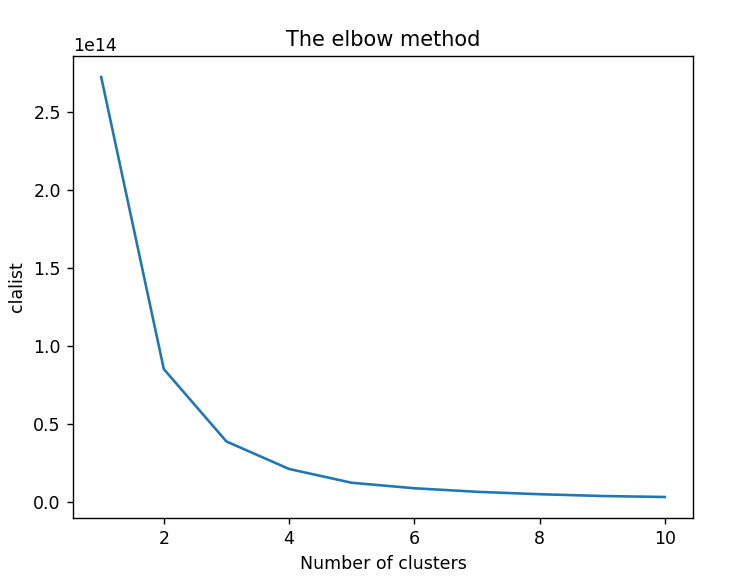


Рисунок 3.5 – Графік пошуку оптимальної кількості кластерів  
Таким чином, як видно з діаграми вище, оптимальна кількість кластерів становить 4

3.2.4 Візуалізація

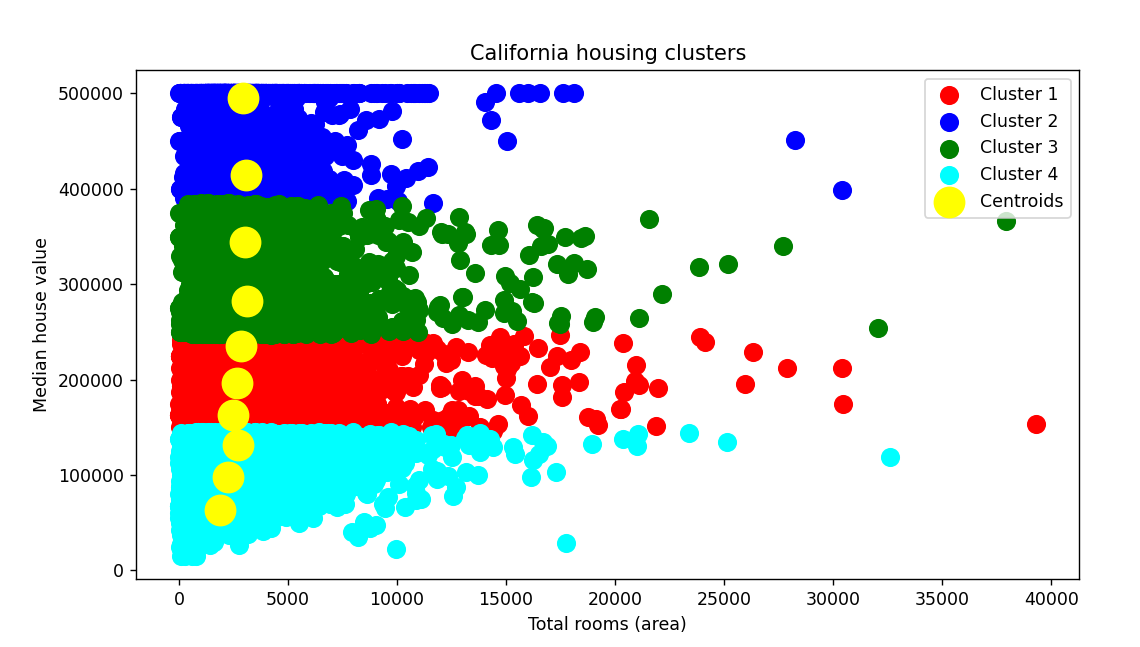


Рисунок 3.6 – Графік кластеризації цін будинків(від кімнат, 4кластера)

Як і слід було очікувати, наведений вище графік показує, що більші будинки, як правило, коштують дорожче. Але також положення центроїдів говорить нам про те, що дуже-дуже великих будинків не так вже й багато, де-факто всі центроїди стоять на майже прямій осі. Але значення дуже різняться. Таким чином, можна сказати, що в Каліфорнії площа будинку не є вирішальним фактором його вартості

Зробимо ще один тест з іншим стовпчиком аргументів, який дасть нам більш лінійну залежність

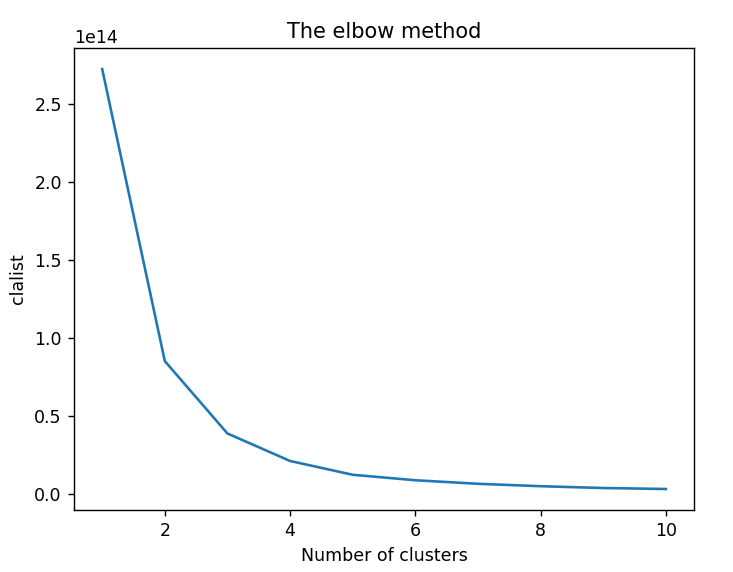


Рисунок 3.6 – Графік пошуку оптимальної кількості кластерів для іншого датфрейму

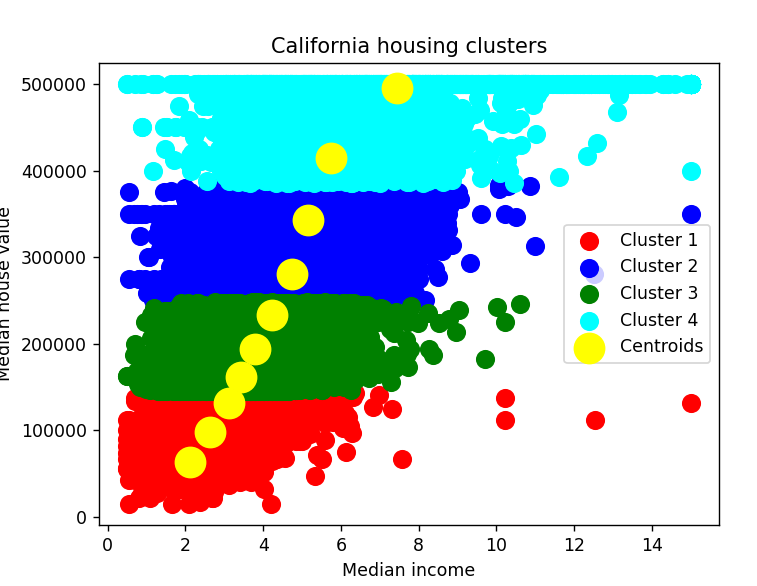
****

Рисунок 3.6 – Графік кластеризації цін будинків(від доходу, 4кластера)

Тут ми маємо сильну залежність між доходом, який приносить той чи інший будинок його власнику, та вартістю цього будинку. Але, як бачимо, зелений і синій кластери дуже близькі, тому зробимо останній тест - зменшимо кількість кластерів.

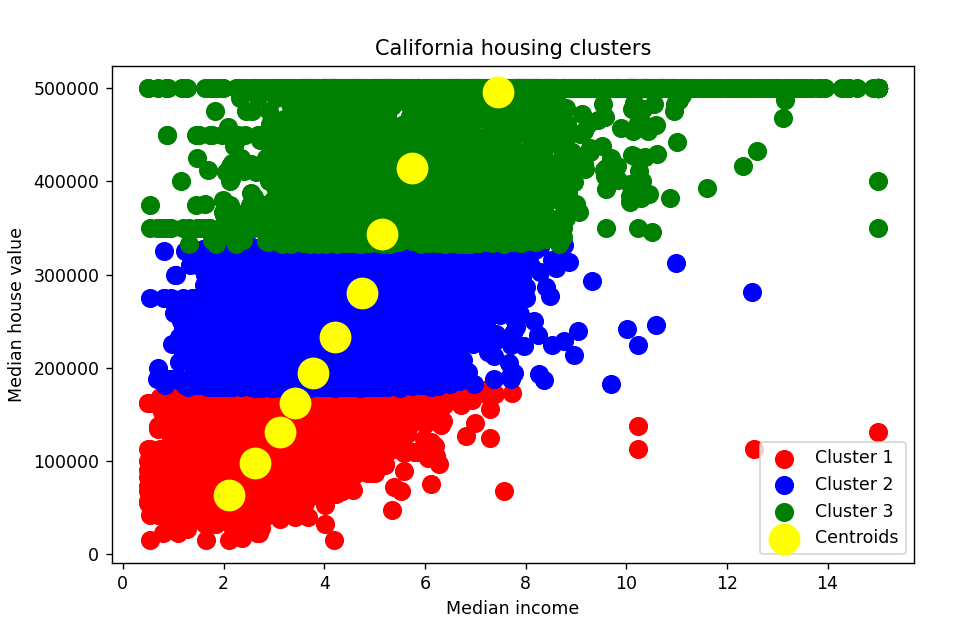


Рисунок 3.6 – Графік кластеризації цін будинків(від доходу, 3кластера)

Це можна пояснити тим, що на графіку кількості кластерів видно, що різниця між 3 і 4 також не дуже велика, хоча вона більш суттєва, ніж між 4 і 5, 6 і т.д. Тому, якщо ми виберемо, наприклад, 6 кластерів, то побачимо гіршу ситуацію, кластери дуже маленькі і між будь-якими двома сусідніми кластерами практично немає відмінностей

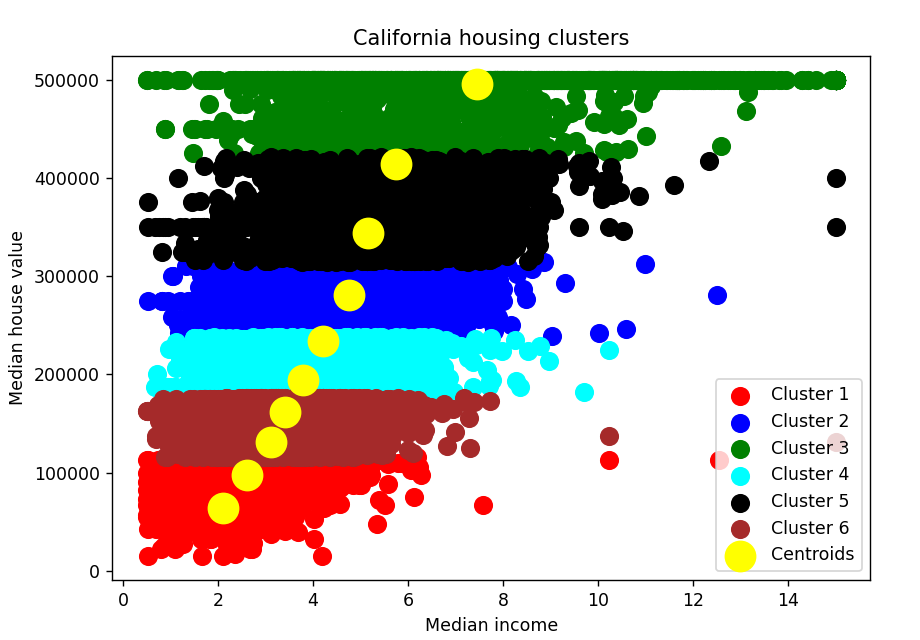


Рисунок 3.7 – Графік кластеризації цін будинків(від доходу, 6 кластерів)

AgglomerativeClustering method

3.3.1 Візуалізація

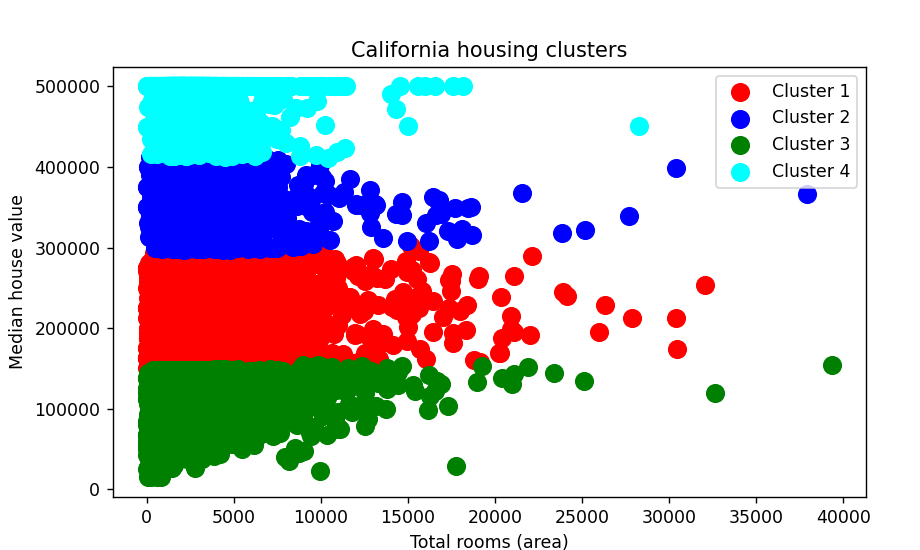


Рисунок 3.3.1 – Графік кластеризації цін будинків(від кімнат, 4кластера)

Як і слід було очікувати, графік схожий з попереднім методом, графік показує, що більші будинки, як правило, коштують дорожче

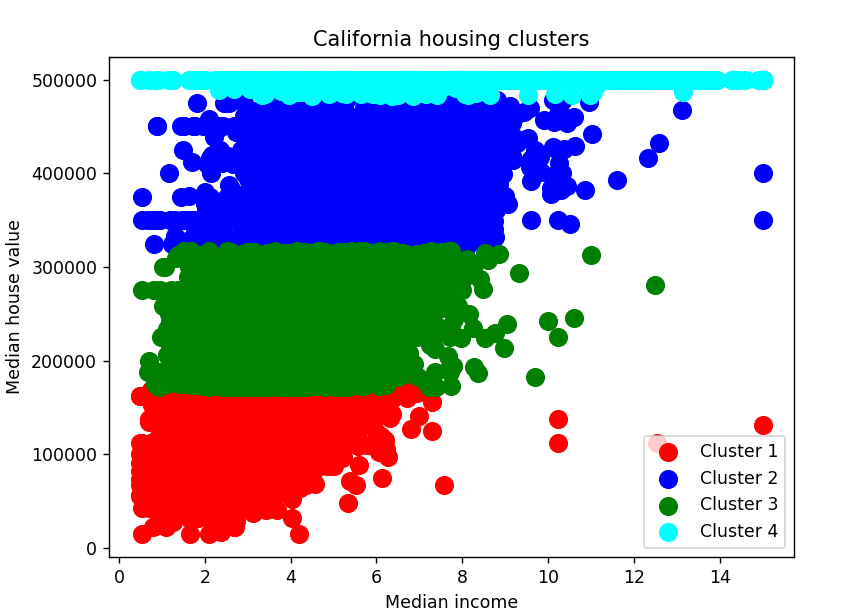
****

Рисунок 3.3.2 – Графік кластеризації цін будинків(від доходу, 4кластера)

Спостерігаємо сильну залежність між доходом, який приносить той чи інший будинок його власнику та вартістю цього будинку. Зелений і синій кластери дуже близькі, тому зменшимо кількість кластерів.

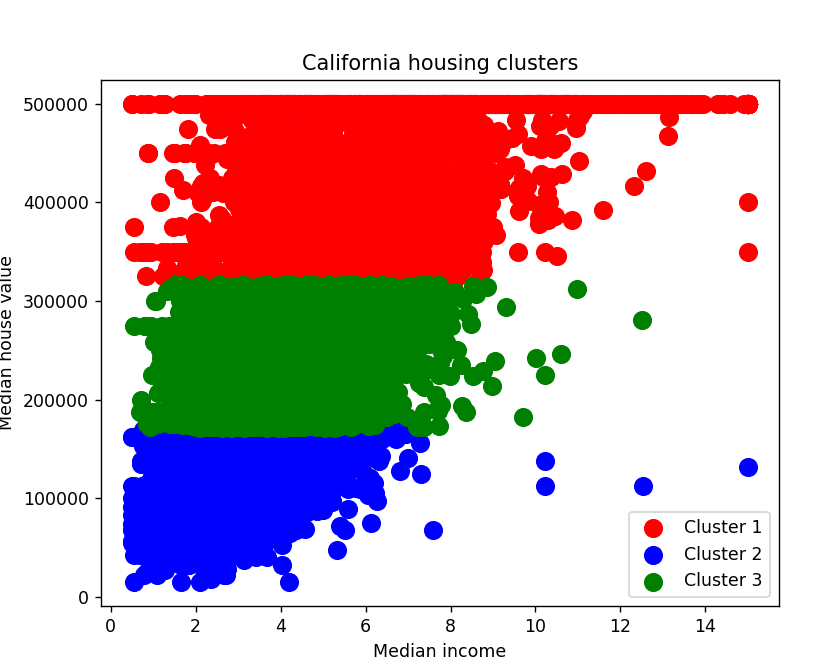


Рисунок 3.3.3 – Графік кластеризації цін будинків(від доходу, 3кластера)

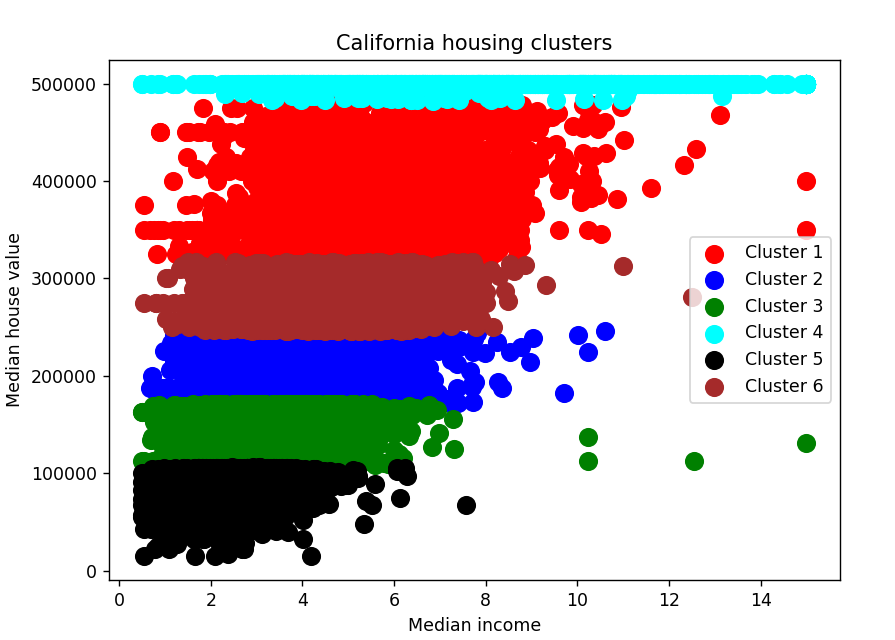


Рисунок 3.3.4 – Графік кластеризації цін будинків(від доходу, 6 кластерів)

Як бачимо, всі графіки схожі, тому щоб вияснити який же метод кращий – проведемо оцінку, аналіз та порівняння цих методів

**4.МЕТОДИ КЛАСТЕРИЗАЦІЇ**

The Elbow Method - це метод, який використовується для визначення оптимальної кількості кластерів у K-Means Clustering. Він базується на обчисленні значення суми квадратів відстаней між кожною точкою даних та центроїдом кластера. Зазвичай, при збільшенні кількості кластерів, сума квадратів відстаней буде зменшуватись. Однак, після певної кількості кластерів, зменшення величини суми квадратів відстаней стає менш і менш помітним. Точка, де зменшення стає менш помітним, називається "ліктем". Це означає, що оптимальна кількість кластерів відповідає кількості, яка знаходиться на лікті на графіку.

K-Means Clustering - це один з найбільш популярних методів кластеризації даних. Він дозволяє групувати набори даних у певну кількість кластерів, що має багато практичних застосувань в різних галузях, включаючи фінанси, маркетинг, медицину та інші.

K-Means Clustering заснований на розділенні набору даних на підмножини зі схожими характеристиками, такими як розмір, форма, кольори, тощо. Для визначення кількості кластерів, що слід використовувати, можна використовувати метод "The Elbow Method".

AgglomerativeClustering є одним з алгоритмів кластеризації, який належить до категорії ієрархічних методів кластеризації.

У методі AgglomerativeClustering, спочатку кожен об'єкт вважається окремим кластером. Далі, на кожній ітерації, два найближчих кластери об'єднуються в один кластер до тих пір, поки не буде досягнуто певний критерій зупинки. Цей процес продовжується до тих пір, поки не буде сформовано один загальний кластер, який містить усі об'єкти. Ієрархічний підхід здатний представляти кластери як дерево, де кожний листок це окремий об'єкт, а рівні вище це кластери, які об'єдналися в попередньому кроці. Цей результат можна візуалізувати у вигляді дендрограми.

AgglomerativeClustering може бути застосований до широкого спектру задач, таких як класифікація об'єктів за характеристиками, знаходження подібних груп користувачів у соціальних мережах, або виявлення кластерів на зображеннях.

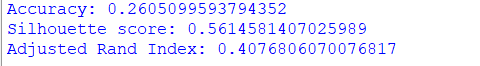
AgglomerativeClustering має параметр, який визначає, який метрику використовувати для визначення відстані між кластерами на кожній ітерації. Метрики можуть бути такими, як Евклідова відстань, Манхеттенська відстань, косинусна схожість тощо. Також, метод має параметр, який визначає кількість кластерів, які мають бути сформовані. Один з недоліків методу AgglomerativeClustering полягає в тому, що він може бути обчислювально вимогливим для великих даних

Отже, використання K-Means Clustering та методу "The Elbow Method" дозволяє зробити об'єктивний аналіз набору даних, зменшити кількість варіантів та допомогти зробити обґрунтовані висновки та прийняти рішення.

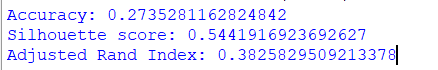
**ПОРІВНЯННЯ МЕТОДІВ**

У першому випадку, для порівняння ми обчислюємо точністьі друкуємо результати(Для AgglomerativeClustering на 1% краще, але слід зазначити, що в кластеризації точність не є основною метрикою оцінки якості, оскільки у кластеризації немає "правильної" відповіді. Оцінка якості кластеризації є більш складним процесом, і для цього використовуються різні метрики, наприклад як silhouette score

Для Kmeans

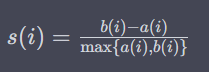
****

Для AgglomerativeClustering



Silhouette score - це метрика, яка використовується для оцінки якості кластеризації. Вона враховує як відстань між елементами в межах одного кластера, так і відстань між елементами різних кластерів. Значення Silhouette score лежить в інтервалі [-1, 1], де більші значення вказують на більш оптимальну кластеризацію. Значення 1 вказує на ідеальну кластеризацію, коли всі елементи в кожному кластері розташовані близько один до одного і далеко від елементів інших кластерів. Значення -1 вказує на те, що багато елементів були помилково додані до кластерів, і вони повинні були бути у іншому кластері або взагалі не належать до кластерів.

Silhouette score для кожного елемента обчислюється за наступною формулою:



де a(i) - середнє значення відстані між елементом i та всіма іншими елементами у тому ж кластері, b(i) - середнє значення відстані між елементом i та всіма елементами у сусідніх кластерах, b(i) - найменша середнє значення відстані між елементом i та всіма елементами у кластері, до якого він належить.

Adjusted Rand Index (ARI) - це міра подібності між двома кластеризаціями, яка враховує як схожість кластерів, так і схожість елементів в кластерах. Ця міра більш за все використовується для порівняння якості кластеризації, особливо в тих випадках, коли ми маємо ідеальну кількість кластерів.

ARI повертає значення в діапазоні від -1 до 1, де -1 вказує на абсолютну розбіжність між кластеризаціями, 0 вказує на випадкову кластеризацію, а 1 вказує на ідеальну подібність між кластеризаціями. Значення більше 0,5 вважається дуже хорошим.

ARI обчислюється шляхом порівняння матриці співпадінь між кластерами істинної кластеризації і кластеризацією, яку ми отримали. Оскільки ARI враховує випадковість, він корисний для порівняння кластеризації, що можуть мати різну кількість кластерів.

Як бачимо, по двом метрикам «перемагає» метод K-means, АЛЕ не можна однозначно стверджувати, що метод K-means краще Agglomerative Clustering або навпаки, оскільки кожен з них має свої переваги та недоліки і підходить для різних типів даних та завдань кластеризації.

Основні переваги методу K-means:

* Простота і швидкість роботи.
* Добре показує себе на даних з явно визначеними кластерами та коливаннями у межах кластерів.
* Працює добре з великими даними.

Основні переваги методу Agglomerative Clustering:

* Працює добре на даних з нечіткими границями між кластерами та змінними розмірами кластерів.
* Зазвичай має кращі результати на даних зі складними залежностями між ознаками.

Отже, вибір між методами K-means і Agglomerative Clustering залежить від характеру даних та постановки задачі кластеризації. Варто розглянути обидва методи та провести порівняльний аналіз їх результатів на конкретному датасеті для вибору найбільш підходящого методу.

**ВИСНОВКИ**

В результаті виконання курсової роботи був проведений аналіз даних про ціни на житло у Каліфорнії, що дозволяє зробити висновки про характеристики цін на житло в різних районах штату.

Кластеризація даних була проведена з використанням методів K-Means Clustering та Agglomerative Clustering та методу "The Elbow Method" для визначення оптимальної кількості кластерів.

Результати аналізу показали, що можна виділити чотири основні кластери цін на житло.

Кластер високих цін на житло складається, в основному, будинки з великою кількістю кімнат. Кластер низьких цін на житло складається з будинкі з низькими середніми доходами та невеликим за розміром житла відповідно. Кластер середніх цін на житло включає райони з середнім доходом та середнім за кількістю кімнат, але не настільки високим, як у класі високих цін, що по суті логічно.

Отримані результати можуть бути корисними для оцінки ринкової конкурентоспроможності та формування стратегії в галузі нерухомості, а також для прийняття рішень про інвестиції та розвиток відповідних районів. Таким чином, кластеризація даних є потужним інструментом аналізу даних, який може допомогти у зробленні об'єктивних висновків та прийнятті рішень на основі даних. Для реалізації поставленої задачі було використано мову програмування Python3. з використанням бібліотек Pandas, Seaborn, Matplotlib, Sklear, Kmeans та інші.

Отже, поставлені задачі були виконані.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Документація мови програмування Python. [Електронний ресурс] –

Режим доступу до ресурсу: <https://docs.python.org/3/>

1. Бібліотека Pandas. [Електронний ресурс] – Режим доступу до ресурсу: <https://pandas.pydata.org/docs/>
2. 3. Бібліотека Seaborn. [Електронний ресурс] – Режим доступу до ресурсу: <https://seaborn.pydata.org/introduction.html>
3. Бібліотека Matplotlib. [Електронний ресурс] – Режим доступу до ресурсу: <https://matplotlib.org/stable/>
4. Бібліотека Sklearn. [Електронний ресурс] – Режим доступу до ресурсу: <https://scikit-learn.org/stable/user_guide.html4>
5. Бібліотека KMeans. [Електронний ресурс] – Режим доступу до ресурсу: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
6. Elbow method (clustering) – <https://en.wikipedia.org/wiki/Elbow_method_(clustering)>

**Додаток А Тексти програмного коду**

*студента групи ІП-11 ІІ курсу*

*Головні О.Р.*

(Обсяг програми (документа), арк., Кб)

(Вид носія даних)

*Жорсткий диск*

(Найменування програми (документа))

*Тексти програмного коду кластеризації цін на житло у Каліфорнії. Метод K-Means та Elbow Method*

# Libraries importing

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import DBSCAN

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import silhouette\_score

from sklearn.metrics import accuracy\_score

from sklearn.metrics import adjusted\_rand\_score

# Read a dataset and show it

califhousing\_set = pd.read\_csv('housing.csv')

califhousing\_set

# Read a dataset and show it

califhousing\_set = pd.read\_csv('housing.csv')

califhousing\_set

# Show the top 5 rows of the datased we read

califhousing\_set.head()

# Show the last 5 rows of the datased we read

califhousing\_set.tail()

# Show the shape of the dataset

califhousing\_set.shape

# Get some general information about the dataset

califhousing\_set.info()

# Check the dataset for any missing data

califhousing\_set.isna().any()

# Remove all rows that cantains missing elements

califhousing\_set.dropna(inplace=True)

# Get general information about the dataset after the removal

califhousing\_set.info()

# Check the dataset for any null elements

califhousing\_set.isnull().sum()

# Build a DataFrame from columns "total\_rooms" and "median\_house\_value", which we will use as features

x = califhousing\_set[['total\_rooms', 'median\_house\_value']].values

x

print('---------------------------------------------------')

# Importing Kmeans

from sklearn.cluster import KMeans

clalist=[]

#inertia\_ is a method that divides data points into clusters

for i in range(1,11):

kmeans = KMeans(n\_clusters= i, init='k-means++', random\_state=0)

kmeans.fit(x)

clalist.append(kmeans.inertia\_)

plt.plot(range(1,11), clalist)

plt.title('The elbow method')

plt.xlabel('Number of clusters')

plt.ylabel('clalist')

plt.show()

kmeansmodel = KMeans(n\_clusters= 4, init='k-means++', random\_state=0)

y\_kmeans= kmeansmodel.fit\_predict(x)

# Get true labels

y = califhousing\_set['median\_house\_value'].apply(lambda x: 0 if x < 200000 else 1 if x < 350000 else 2).values

# Calculate accuracy

accuracy = accuracy\_score(y, y\_kmeans)

# Print result

print(f"Accuracy: {accuracy}")

score = silhouette\_score(x, y\_kmeans)

print("Silhouette score:", score)

ari = adjusted\_rand\_score(y, y\_kmeans)

print("Adjusted Rand Index:", ari)

plt.scatter(x[y\_kmeans == 0, 0], x[y\_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')

plt.scatter(x[y\_kmeans == 1, 0], x[y\_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')

plt.scatter(x[y\_kmeans == 2, 0], x[y\_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')

plt.scatter(x[y\_kmeans == 3, 0], x[y\_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')

plt.scatter(kmeans.cluster\_centers\_[0:10, 0], kmeans.cluster\_centers\_[0:10, 1], s = 300, c = 'yellow', label = 'Centroids')

plt.title('California housing clusters')

plt.xlabel('Total rooms (area)')

plt.ylabel('Median house value')

plt.legend()

plt.show()

# Build a DataFrame from columns "median\_income" and "median\_house\_value", which we will use as features

x = califhousing\_set[['median\_income', 'median\_house\_value']].values

x

clalist=[]

for i in range(1,11):

kmeans = KMeans(n\_clusters= i, init='k-means++', random\_state=0)

kmeans.fit(x)

clalist.append(kmeans.inertia\_)

plt.plot(range(1,11), clalist)

plt.title('The elbow method')

plt.xlabel('Number of clusters')

plt.ylabel('clalist')

plt.show()

kmeansmodel = KMeans(n\_clusters= 4, init='k-means++', random\_state=0)

y\_kmeans= kmeansmodel.fit\_predict(x)

plt.scatter(x[y\_kmeans == 0, 0], x[y\_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')

plt.scatter(x[y\_kmeans == 1, 0], x[y\_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')

plt.scatter(x[y\_kmeans == 2, 0], x[y\_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')

plt.scatter(x[y\_kmeans == 3, 0], x[y\_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')

plt.scatter(kmeans.cluster\_centers\_[0:10, 0], kmeans.cluster\_centers\_[0:10, 1], s = 300, c = 'yellow', label = 'Centroids')

plt.title('California housing clusters')

plt.xlabel('Median income')

plt.ylabel('Median house value')

plt.legend()

plt.show()

kmeansmodel = KMeans(n\_clusters= 3, init='k-means++', random\_state=0)

y\_kmeans= kmeansmodel.fit\_predict(x)

plt.scatter(x[y\_kmeans == 0, 0], x[y\_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')

plt.scatter(x[y\_kmeans == 1, 0], x[y\_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')

plt.scatter(x[y\_kmeans == 2, 0], x[y\_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')

plt.scatter(kmeans.cluster\_centers\_[0:10, 0], kmeans.cluster\_centers\_[0:10, 1], s = 300, c = 'yellow', label = 'Centroids')

plt.title('California housing clusters')

plt.xlabel('Median income')

plt.ylabel('Median house value')

plt.legend()

plt.show()

kmeansmodel = KMeans(n\_clusters= 6, init='k-means++', random\_state=0)

y\_kmeans= kmeansmodel.fit\_predict(x)

plt.scatter(x[y\_kmeans == 0, 0], x[y\_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')

plt.scatter(x[y\_kmeans == 1, 0], x[y\_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')

plt.scatter(x[y\_kmeans == 2, 0], x[y\_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')

plt.scatter(x[y\_kmeans == 3, 0], x[y\_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')

plt.scatter(x[y\_kmeans == 4, 0], x[y\_kmeans == 4, 1], s = 100, c = 'black', label = 'Cluster 5')

plt.scatter(x[y\_kmeans == 5, 0], x[y\_kmeans == 5, 1], s = 100, c = 'brown', label = 'Cluster 6')

plt.scatter(kmeans.cluster\_centers\_[0:10, 0], kmeans.cluster\_centers\_[0:10, 1], s = 300, c = 'yellow', label = 'Centroids')

plt.title('California housing clusters')

plt.xlabel('Median income')

plt.ylabel('Median house value')

plt.legend()

plt.show()

# Importing libraries

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.cluster import AgglomerativeClustering

from sklearn.preprocessing import StandardScaler

from scipy.cluster.hierarchy import linkage

from scipy.cluster.hierarchy import dendrogram

from sklearn.metrics import accuracy\_score

from sklearn.metrics import silhouette\_score

from sklearn.metrics import adjusted\_rand\_score

# Load the data

califhousing\_set = pd.read\_csv('housing.csv')

califhousing\_set

califhousing\_set.dropna(inplace=True)

# Build a DataFrame from columns "total\_rooms" and "median\_house\_value",

x\_std = califhousing\_set[['total\_rooms', 'median\_house\_value']].values

x\_std

n\_clusters = 4

model = AgglomerativeClustering(n\_clusters=n\_clusters, linkage='ward')

model.fit(x\_std)

# Get true labels

labels = model.labels\_

y = califhousing\_set['median\_house\_value'].apply(lambda x: 0 if x < 200000 else 1 if x < 350000 else 2).values

# Calculate accuracy

accuracy = accuracy\_score(y, labels)

# Print result

print(f"Accuracy: {accuracy}")

# Обчислення Silhouette score

score = silhouette\_score(x\_std, labels)

print("Silhouette score:", score)

ari = adjusted\_rand\_score(y, labels)

print("Adjusted Rand Index:", ari)

# Plot the clusters

plt.scatter(x\_std[model.labels\_==0, 0], x\_std[model.labels\_==0, 1], s=100, c = 'red', label='Cluster 1')

plt.scatter(x\_std[model.labels\_==1, 0], x\_std[model.labels\_==1, 1], s=100, c = 'blue', label='Cluster 2')

plt.scatter(x\_std[model.labels\_==2, 0], x\_std[model.labels\_==2, 1], s=100, c = 'green', label='Cluster 3')

plt.scatter(x\_std[model.labels\_==3, 0], x\_std[model.labels\_==3, 1], s=100, c = 'cyan', label='Cluster 4')

plt.title('California housing clusters')

plt.xlabel('Total rooms (area)')

plt.ylabel('Median house value')

plt.legend()

plt.show()

# Build a DataFrame from columns "median\_income" and "median\_house\_value", which we will use as features

x\_std = califhousing\_set[['median\_income', 'median\_house\_value']].values

x\_std

n\_clusters = 4

model = AgglomerativeClustering(n\_clusters=n\_clusters, linkage='ward')

model.fit(x\_std)

# Plot the clusters

plt.scatter(x\_std[model.labels\_==0, 0], x\_std[model.labels\_==0, 1], s=100, c = 'red', label='Cluster 1')

plt.scatter(x\_std[model.labels\_==1, 0], x\_std[model.labels\_==1, 1], s=100, c = 'blue', label='Cluster 2')

plt.scatter(x\_std[model.labels\_==2, 0], x\_std[model.labels\_==2, 1], s=100, c = 'green', label='Cluster 3')

plt.scatter(x\_std[model.labels\_==3, 0], x\_std[model.labels\_==3, 1], s=100, c = 'cyan', label='Cluster 4')

plt.title('California housing clusters')

plt.xlabel('Median income')

plt.ylabel('Median house value')

plt.legend()

plt.show()

n\_clusters = 3

model = AgglomerativeClustering(n\_clusters=n\_clusters, linkage='ward')

model.fit(x\_std)

# Plot the clusters

plt.scatter(x\_std[model.labels\_==0, 0], x\_std[model.labels\_==0, 1], s=100, c = 'red', label='Cluster 1')

plt.scatter(x\_std[model.labels\_==1, 0], x\_std[model.labels\_==1, 1], s=100, c = 'blue', label='Cluster 2')

plt.scatter(x\_std[model.labels\_==2, 0], x\_std[model.labels\_==2, 1], s=100, c = 'green', label='Cluster 3')

plt.title('California housing clusters')

plt.xlabel('Median income')

plt.ylabel('Median house value')

plt.legend()

plt.show()

n\_clusters = 6

model = AgglomerativeClustering(n\_clusters=n\_clusters, linkage='ward')

model.fit(x\_std)

# Plot the clusters

plt.scatter(x\_std[model.labels\_==0, 0], x\_std[model.labels\_==0, 1], s=100, c = 'red', label='Cluster 1')

plt.scatter(x\_std[model.labels\_==1, 0], x\_std[model.labels\_==1, 1], s=100, c = 'blue', label='Cluster 2')

plt.scatter(x\_std[model.labels\_==2, 0], x\_std[model.labels\_==2, 1], s=100, c = 'green', label='Cluster 3')

plt.scatter(x\_std[model.labels\_==3, 0], x\_std[model.labels\_==3, 1], s=100, c = 'cyan', label='Cluster 4')

plt.scatter(x\_std[model.labels\_==4, 0], x\_std[model.labels\_==4, 1], s=100, c = 'black', label='Cluster 5')

plt.scatter(x\_std[model.labels\_==5, 0], x\_std[model.labels\_==5, 1], s=100, c = 'brown', label='Cluster 6')

plt.title('California housing clusters')

plt.xlabel('Median income')

plt.ylabel('Median house value')

plt.legend()

plt.show()