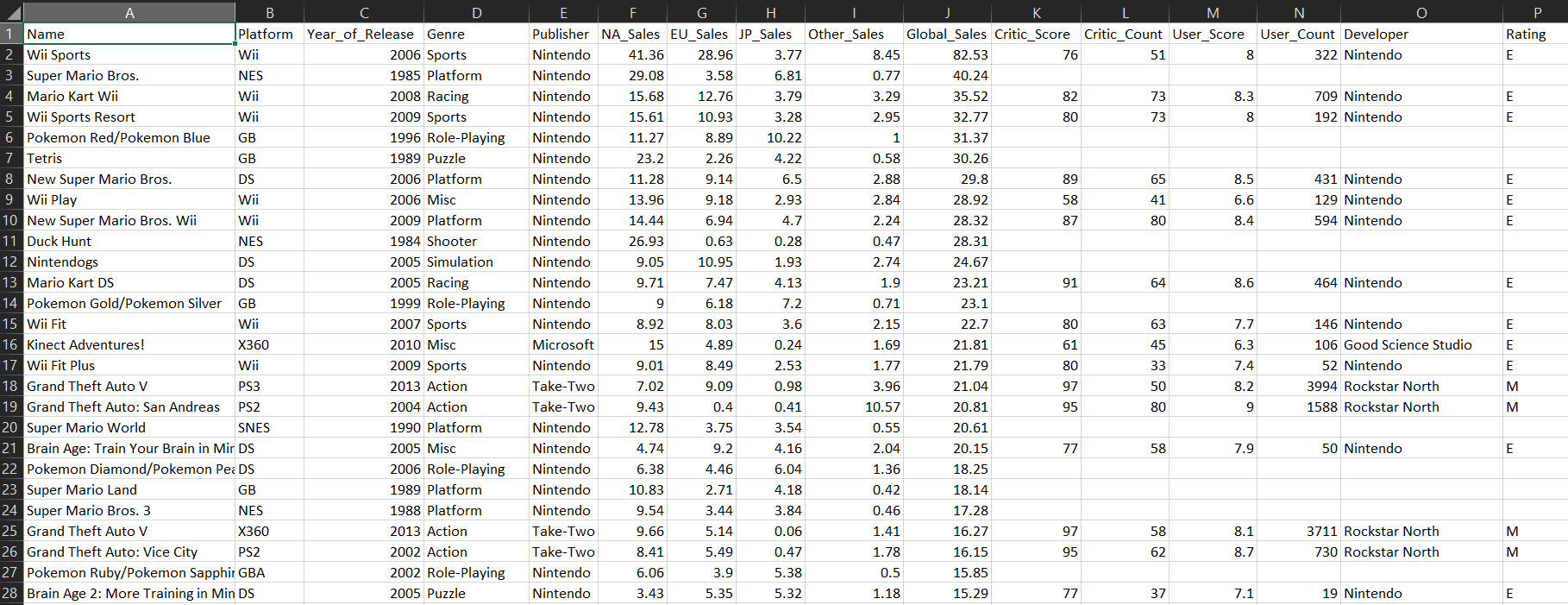
Застосування методів нечіткої кластеризації C-середніх (FCM algorithm) та GMM (Gaussian Mixture Models) для аналізу даних на мові Python.

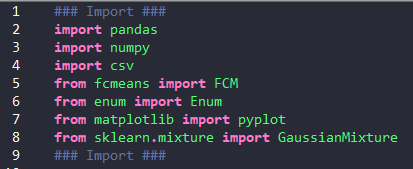
Файл вхідного набору даних про продажі відеоігор:



**Хід роботи**

Для використання алгоритму нечіткої кластеризації потрібна множина точок, що утворюють між собою скупчення в умовному просторі. Для того, щоб була можливість зобразити результати графічно та інтерпретувати їх, будемо виводити утворені кластери у двовимірному просторі.

1. Імпортую необхідні бібліотеки.

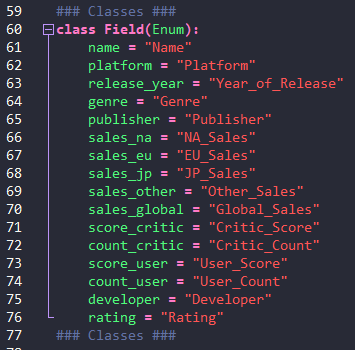


2. Реалізую допоміжні функції:

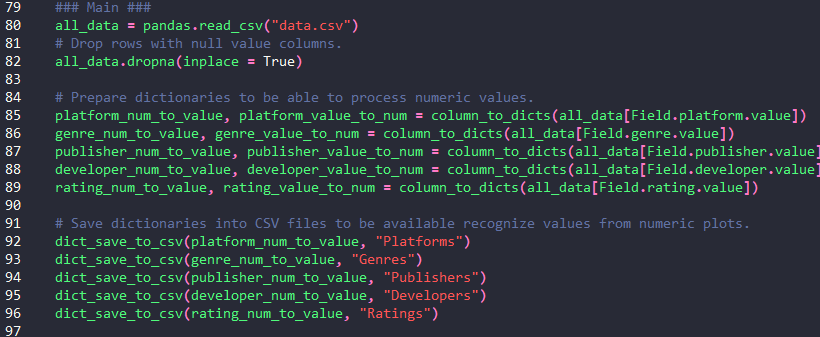


* column\_to\_dicts – отримує стовбець із фрейму даних бібліотеки *pandas*, і перетворює його в два словники (*{0: “value”}* та *{“value”: 0}*), далі це буде використано для представлення рядкових даних у числовому форматі з метою кластеризації
* precess\_with\_fcm – здійснює м’яку кластеризацію за методом С-середніх із бібліотеки *fcmeans* і виводить результат на двовимірному графіку. Також зберігає зображення графіку у файл
* dict\_save\_to\_csv – зберігає у CSV файл словник, отриманий на вхід. Використана бібліотека *csv*. Це зберігання необхідне для того, щоб очно зіставити числові дані графіку з відповідними їм рядковими значеннями
* process\_with\_gmm – здійснює обробку даних за методом Gaussian Mixture Models із бібліотеки *sklearn.mixture*

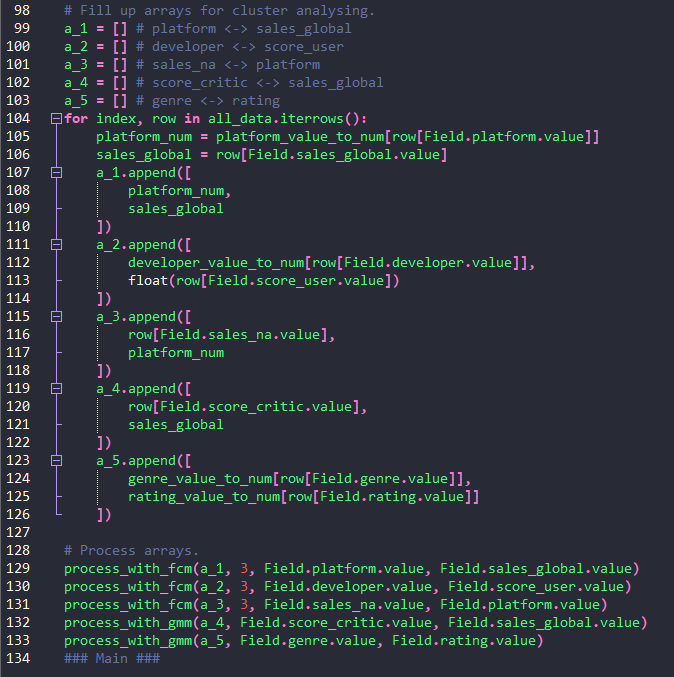
3. Оголошую клас-перелічення, щоб мінімізувати використання однакових рядкових значень для доступу до полів даних.



4. Засобами бібліотеки *pandas* зчитую набір даних з CSV файлу, відкидаю рядки що містять порожні значення. Готую словники відповідностей унікальних рядкових значень до їх числового індексу, та записую дані до CSV файлів.

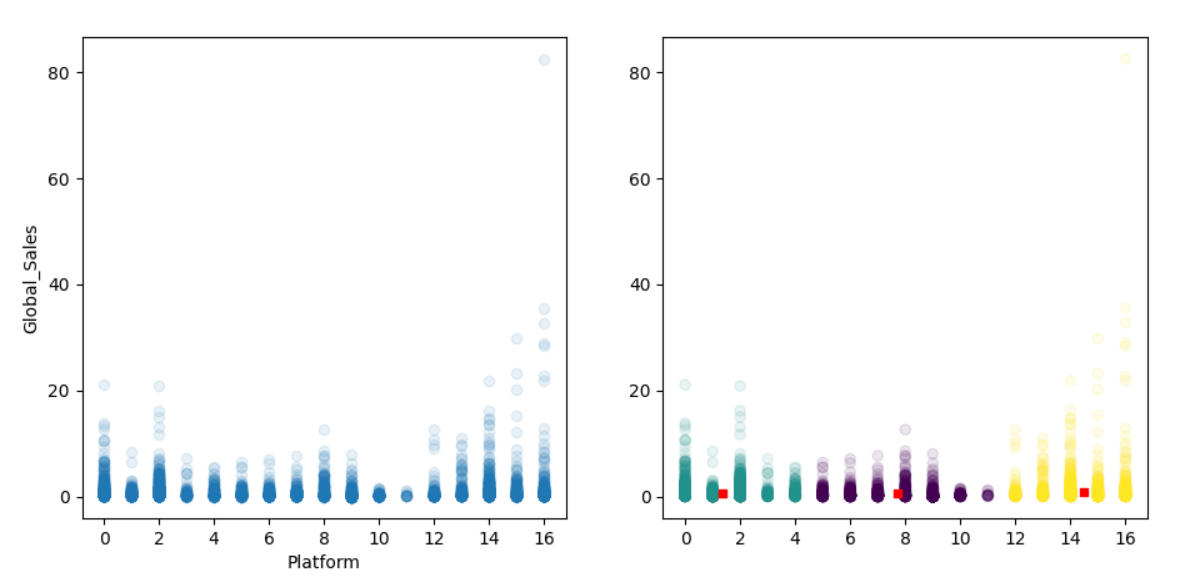


5. Оголошую масиви для пар атрибутів, що будуть аналізуватися на взаємозв’язок. Наповнюю масиви відповідними значеннями з рядків набору, та відправляю їх на обробку.

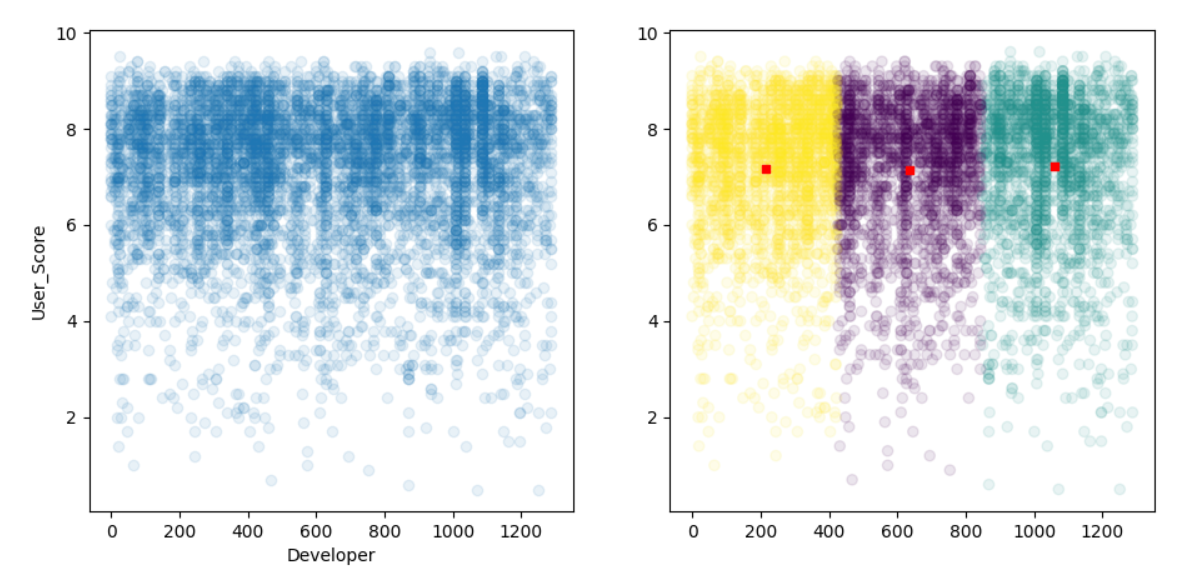


6. Далі аналізую графік першої кластеризацї. Реалізації розподілилися так, що є помітно низькі глобальні продажі відеоігор (у мільйонах одиниць) на платформах 11 (*DC*) та 10 (“*PSV” aka PlayStation Vita*). На провому графіку різні кольори позначають різні кластери, і червоні квадрати – їх центри. Усі три центри по висоті наближені до нижньої границі. Звідси роблю висновок:

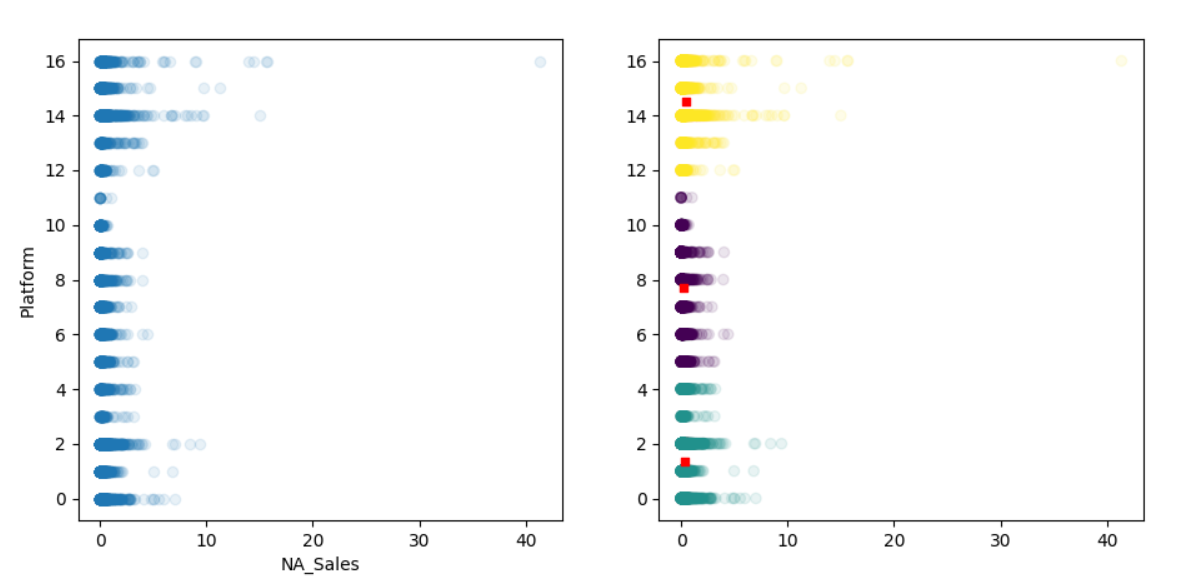
*Здебільшого всі ігрові платформи мають низьку чи невисоку кількість глобальних продажів. Однак, є такі рекорди як більше ніж 80 мільйонів копій у платформи 16 (Wii).*



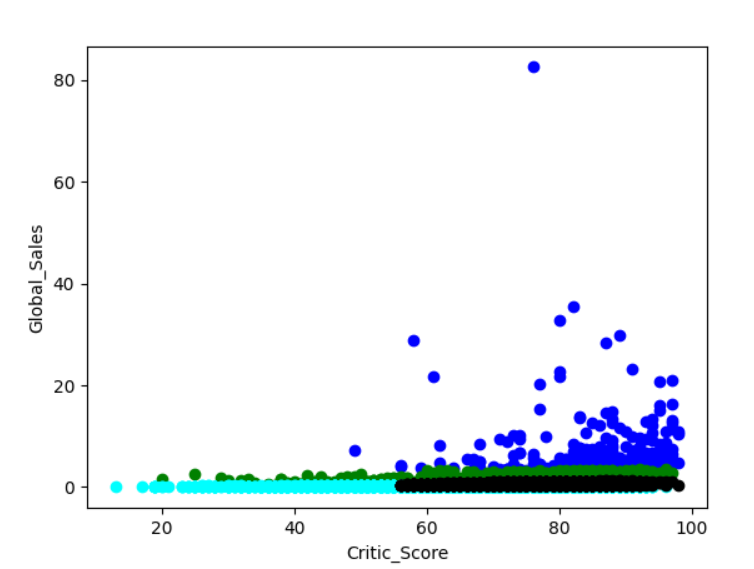
7. На графіках другої пари ознак можна побачити, що користувачі здебільшого ставлять високі оцінки (від 6 до 9), а також оцінок дещо більше у окремих розробників. За розміщенням центрів кластерів можна побачити, що густина розподілу реалізацій в них переважно однакова, отож робимо висновок, що оцінка користувача не пов’язана з конкретними розробниками відеоігор.



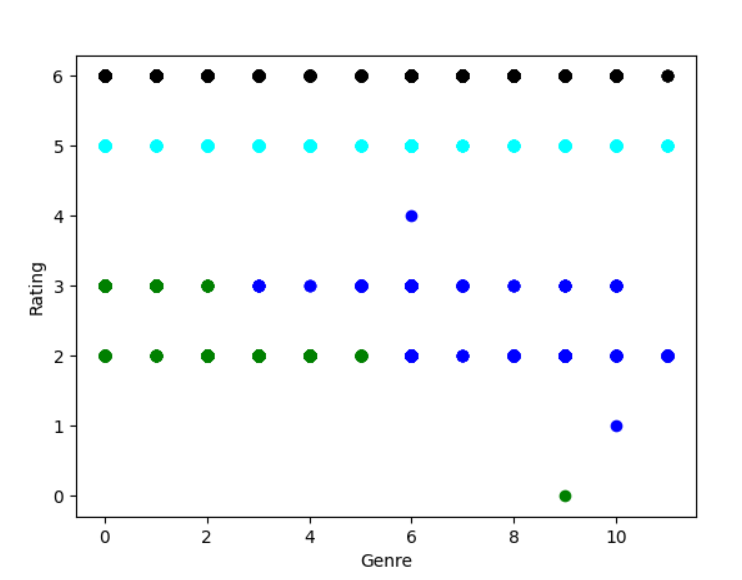
8. Наступна пара ознак – продажі відеоігор у Північній Америці та ігрова платформа. З розміщення центрів кластерів можна виокремити схильність доп платформ 1 (PS3), 8 (PSP), 14 (PS) та 15 (X360).



9. На графіках ознак оцінки критиків та глобальних продажів можна спостерігати високу залежність продажів від оцінки у третьому (фіолетовому) кластері.



10. З графіку пари ознак жанр-рейтинг можна виокремити, що існують конкретні жанри відеоігор, для яких рейтинг стабільно високий, а для деяких не вище середнього.



**Висновок**: в ході даної роботи було використано реалізацію методу кластеризації C-середніх Python-бібліотеки *fcmeans* та методу Gaussian Mixture Models бібліотеки *sklearn.mixture*, для аналізу даних з продажу відеоігор та представлення кластерів аналізу цих даних у графічному вигляді.