

СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

**КУРСОВ ПРОЕКТ**

**ПО СИСТЕМИ, ОСНОВАНИ НА ЗНАНИЯ**

**Тема:**

*Система за генериране на препоръка за закупуване на книги*

Студенти:

Александра Георгиева Бамбалдокова, III гр., ф.н 72088

Антон Мариов Янчев, III гр., ф.н 72086

Мария Димитрова Червенкова, III гр., ф.н. 72098

София, януари 2023 г.

1. Формулировка на задачата

Реализиране на система за генериране на препоръка за закупуване на книги. Системата трябва по дадена книга от използвания набор от данни да направи предложения на потребителя за подходящи книги, базирани на рейтинги от всички потребители.

1. Използвани алгоритми

Проучване и запознаване с темата: [[1]](https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-part-2-k-nearest-neighbors-and-matrix-c04b3c2ef55c) *How Did We Build Book Recommender Systems in An Hour Part 2 — k Nearest Neighbors and Matrix Factorization*.

Набор от данни: [[2]](http://www2.informatik.uni-freiburg.de/~cziegler/BX/) *Book-Crossing Dataset*.

За решение на задачата използваме к-най-близки съседи.

[[3]](https://learn.g2.com/k-nearest-neighbor) Алгоритъмът k-най-близки съседи (KNN) е метод за класифициране на данни за оценка на вероятността точка от данни да стане член на една или друга група въз основа на групата, към която принадлежат най-близките до нея точки от данни.

Алгоритъмът на k-най-близкия съсед е вид алгоритъм за контролирано машинно обучение, използван за решаване на проблеми с класификация и регресия. Въпреки това, той се използва главно за проблеми с класификацията. KNN е мързелив и непараметричен алгоритъм.

Нарича се алгоритъм за мързеливо обучение или мързелив учащ се, защото не извършва никакво обучение, когато предоставяте данните за обучение. Вместо това, той просто съхранява данните по време на обучението и не извършва никакви изчисления. Той не изгражда модел, докато не бъде извършено запитване към набора от данни. Това прави KNN идеален за извличане на данни.

Смята се за непараметричен метод, тъй като не прави никакви предположения относно основното разпределение на данните. Просто казано, KNN се опитва да определи към коя група принадлежи точка от данни, като разглежда точките от данни около нея.

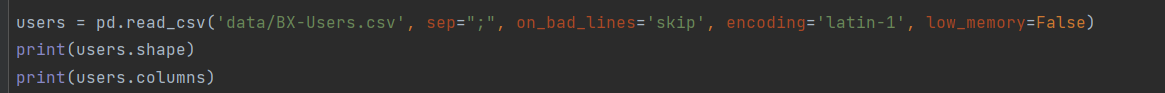
1. Описание на програмната реализация

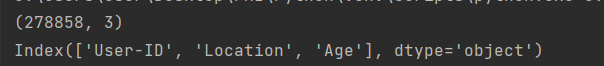
* Представяне на набора от данни

Наборът от данни *Book-Crossing* се състои от 3 таблици.

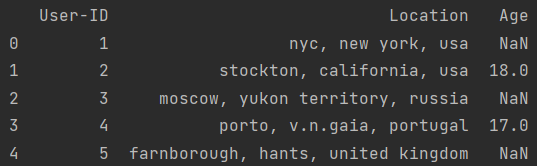
**BX-Users**

Съдържа потребителите, които се представят чрез User-ID, Location, Age. Потребителските идентификатори („User-ID“) са анонимизирани и се съпоставят с цели числа. Предоставят се демографски данни („Местоположение“, „Възраст“), ако са налични. В противен случай тези полета съдържат NULL-стойности.



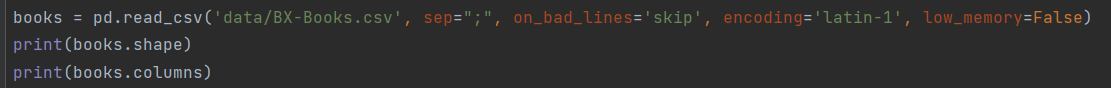


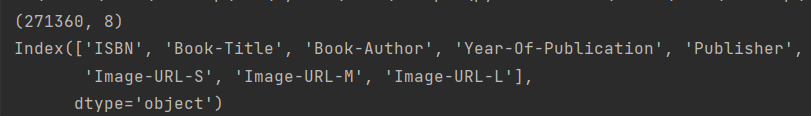
Таблицата изглежда по следния начин (първите 5 реда за пример):



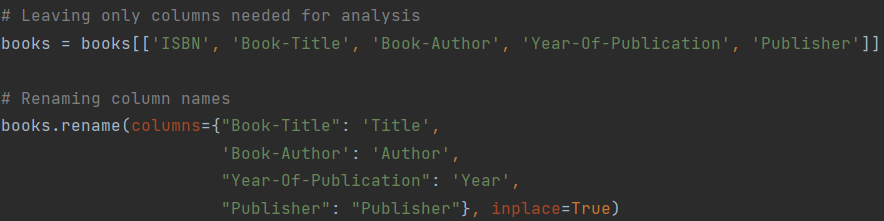
**BX-Books**

Книгите се идентифицират със съответния им ISBN номер. Невалидните ISBN номера вече са премахнати от набора от данни. Освен това е дадена информация, базирана на съдържание („Book-Title “, „Book-Author“, „Year-Of-Publication “, „Publisher“), получена от Amazon Web Services. Имайте предвид, че в случай на няколко автора се предоставя само първият. Дадени са и URL адреси, водещи към изображения на корицата, които се появяват в три различни вида („URL-Image-S“, „URL-Image-M“, „URL-Image-L“), т.е. малък, среден, голям. Тези URL адреси сочат към уеб сайта на Amazon.

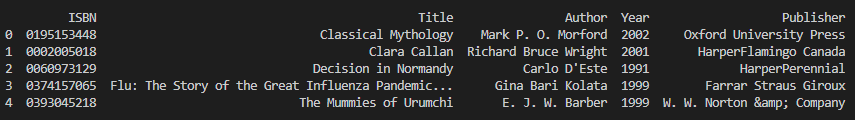




За целта на проекта премахваме колоните с Image-URL, понеже не са необходими за системата. Също така преименуваме колоните с по-кратки имена за прегледност.



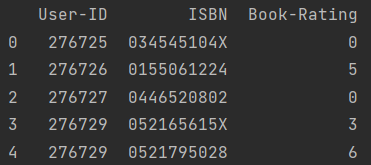
Преработената таблица изглежда по следния начин (първите 5 реда за пример):



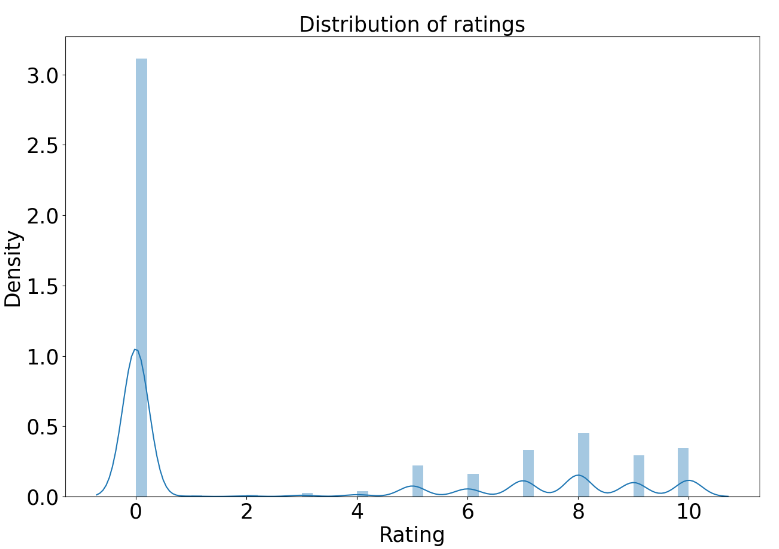
**BX-Book-Ratings**

Съдържа информация за отзив на книгата. Отзивите (`Book-Rating`) са или изрични, изразени по скала от 1-10 (по-високите стойности означават по-висок отзив), или имплицитни, изразени с 0.

Таблицата изглежда по следния начин (първите 5 реда за пример):



Разпределение на отзивите на всички потребители:



Забелязваме, че отзивите са разпределени неравномерно и мнозинството от тях са 0. По тази причина за по-добри резултати в системата премахваме всички потребители с по-малко от 300 отзива.



* Решение на задачата

1. Съединяваме таблиците Books и Ratings по ISBN на книгите.



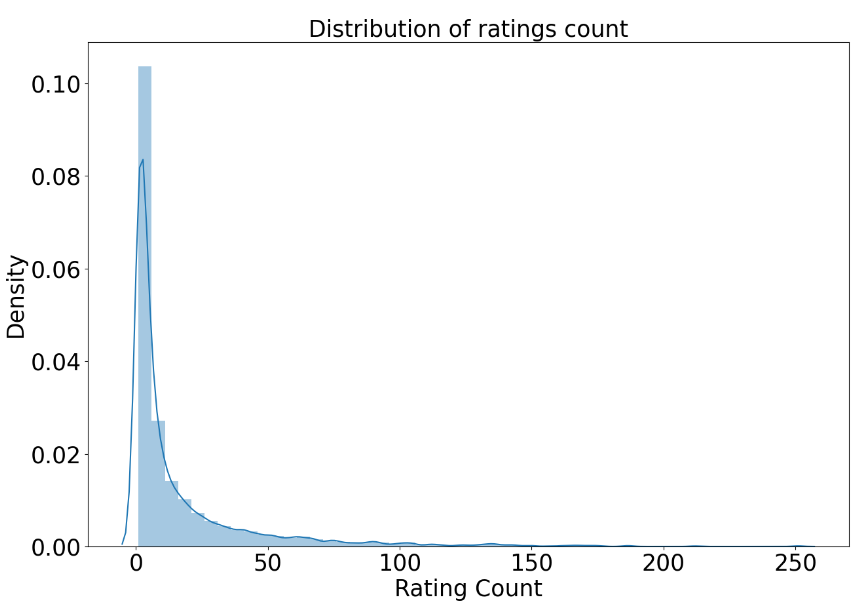
1. Създаваме таблица, която съпоставя име на книга с общ брой отзиви за нея.



1. Съединяваме новосъздадената таблица с предишната по заглавие на книгите.



Дистрибуция на броя на отзивите:



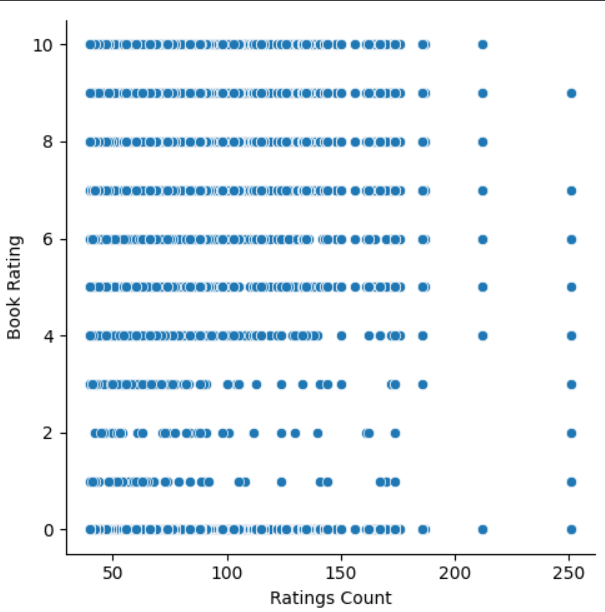
Забелязваме, че мнозинството книги имат по 0 отзива и за подобряване точността на системата премахваме всички книги с по-малко от 40 отзива.



1. Премахваме повтарящите се отзиви от един и същ потребител на една и съща книга.



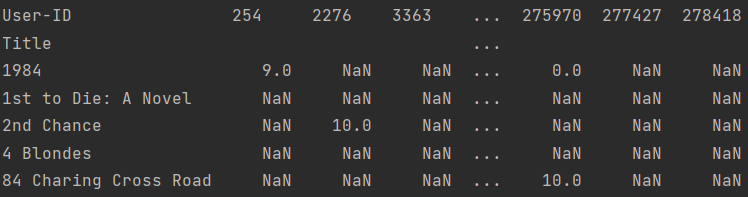
Връзка между брой отзиви и отзиви на книгите:



Забелязваме, че няма връзка между атрибутите Book Rating и Ratings Count.

1. Създаваме обобщена таблица с колони идентификаторите на потребителите, редове- заглавия на книгите. Стойностите на таблицата ще са отзивите, които съответният потребителя е дал на дадена книга.





Заменяме стойностите NaN (липса на отзив от даден потребител за книга) с 0, защото алгоритъмът, който ще използваме по-късно, пресмята разстояние между съседни стойности.

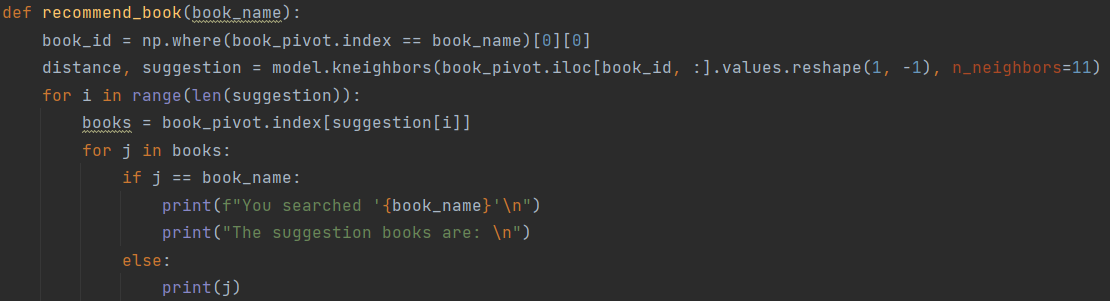
1. Трансформираме обобщената таблица в SciPy разредена матрица за по-ефикасни пресмятания.



1. Прилагаме алгоритъма k- най-близки съседи върху получената разредена матрица. Избираме алгоритъма, чрез който да намерим най-близките съседи, да е brute (алгоритъм с брутална сила) и напасваме модела.

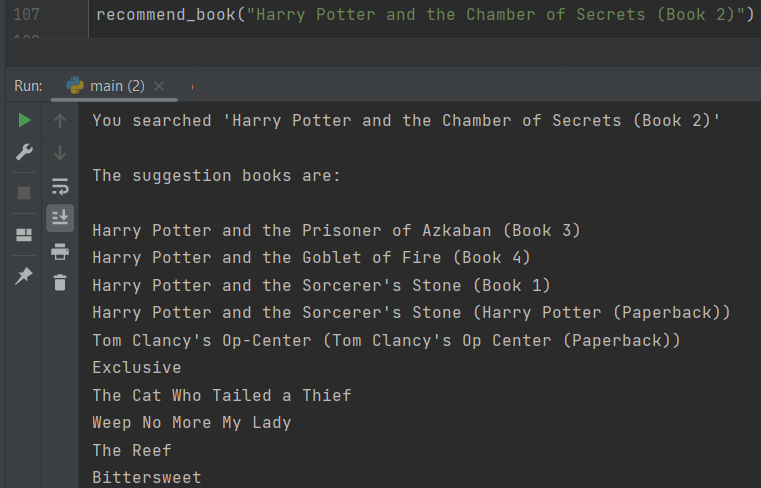


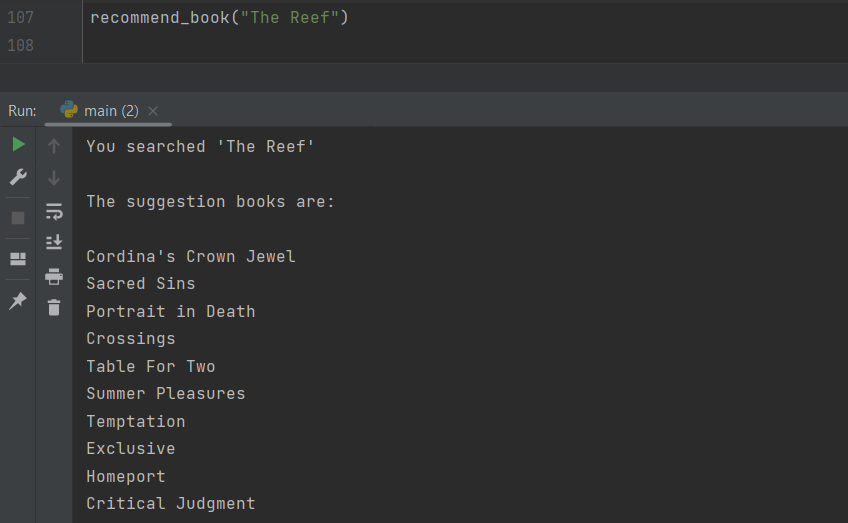
* Тестов модел на системата

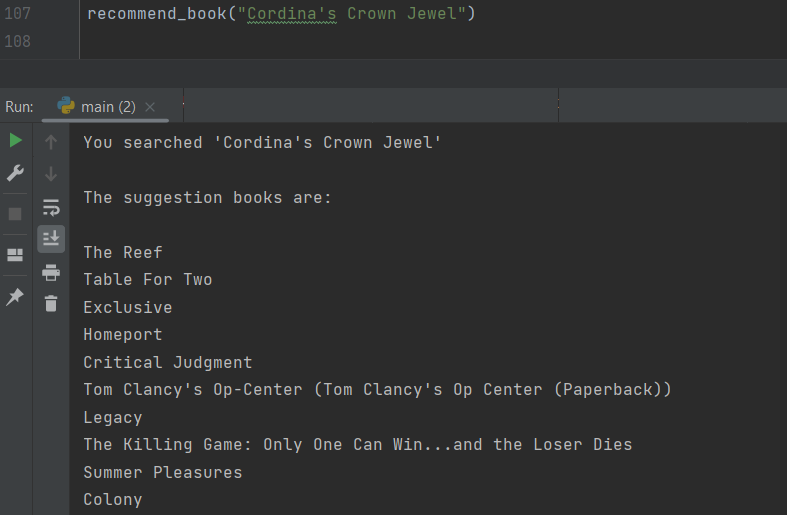


Създаваме тестов модел на системата, който приема име на книга и връща предложения за книги на потребителя.

1. Запазваме номера на реда, на който се намира подадената книгата като book\_id, за да го използваме за търсене в алгоритъма.
2. Прилагаме алгоритъма к- най-близки съседи за намерения индекс и избираме броя на книгите, които искаме системата да предлага на потребителите да е 10.
3. Извеждаме предложените книги.
4. Примери, илюстриращи работата на програмната система







1. Литература
2. [Susan Li, How Did We Build Book Recommender Systems in An Hour Part2 — k Nearest Neighbors and Matrix Factorization, Towards Data Science, 2017](https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-part-2-k-nearest-neighbors-and-matrix-c04b3c2ef55c), 28.12.2022г.
3. [Cai-Nicolas Ziegler, DBIS Freiburg, Book-Crossing Dataset, Institut für Informatik, Universität Freiburg, 2004](http://www2.informatik.uni-freiburg.de/~cziegler/BX/), 28.12.2022г.
4. [Аmal Joby, What is K-Nearest Neighbor? An ML Algorithm to Classify Data, 2021](https://learn.g2.com/k-nearest-neighbor), 28.12.2022г.