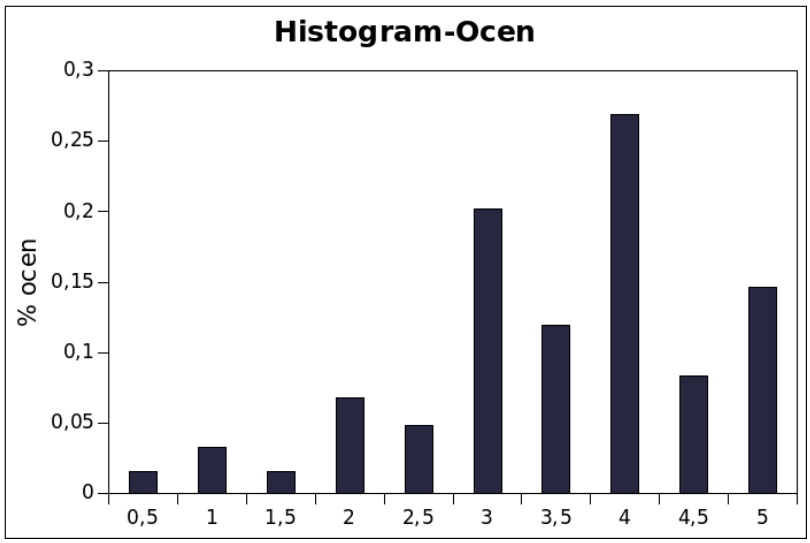
Mikołaj Koszowski 274392
Projekt FSiT Etap III

Temat: Dyfuzyjny system rekomendacji filmów.

Obróbka danych:
Zbiór danych zawierał:

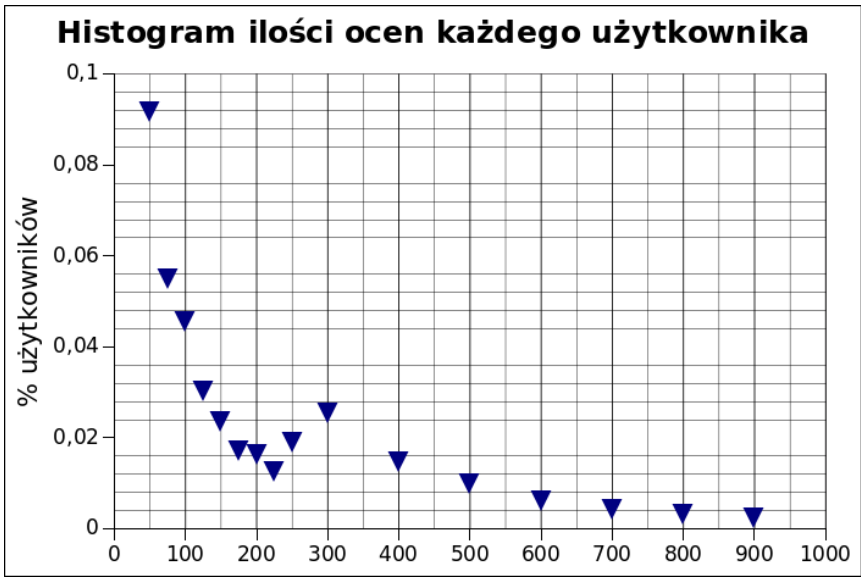
- 26mln ocen 270k unikalnych użytkowników
- 45 000 filmów z informacjami jak budżet, obsada i gatunek

Obróbkę danych zaczęto od stworzenia histogramu ocen użytkowników(Stats.py)
Strona z której zbierane były dane ma system oceny stworzony z 5 gwiazdek, z dokładnością oceny do połowy gwiazdki.



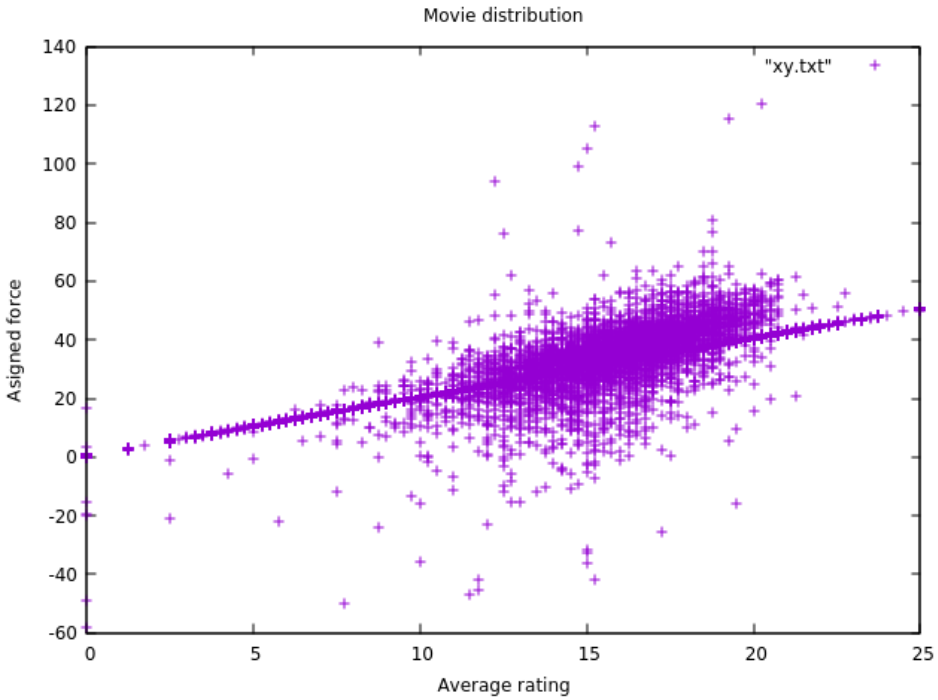
Widać, że użytkownicy zdecydowanie niechętnie dają ocenę połówkową.

Następnie postanowiono przeanalizować średnią liczbę ocen każdego użytkownika.(userStats.py)



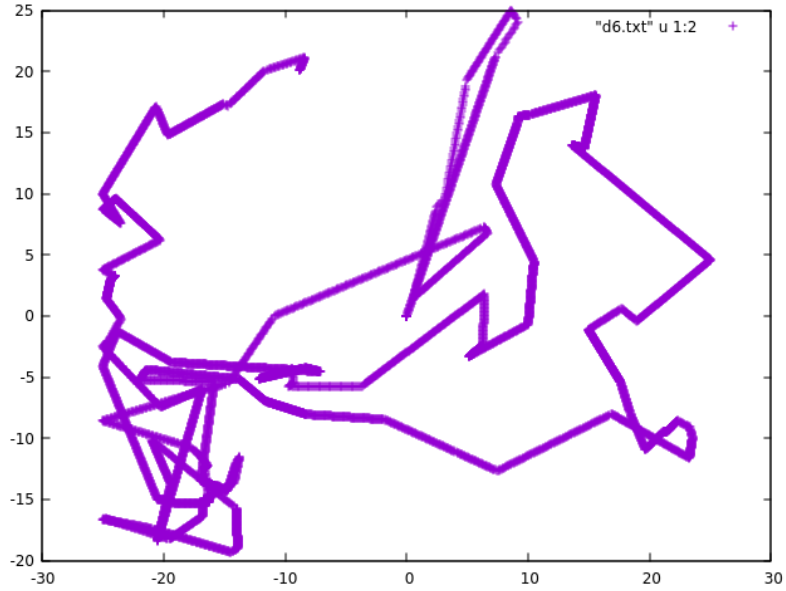
Na powyższym histogramie pominięto użytkowników, którzy ocenili mniej niż 25 filmów.
Stanowili oni ~45% całej bazy danych.
Skupiono się na użytkownikach, którzy mają ocenione przynajmniej 300 filmów, widać, że stanowią oni oddzielną grupę od poprzednich użytkowników, którą cechował wykładniczy spadek ilości obejrzanych filmów.

Obróbka danych bazy 45k filmów 0,15% wpisów było wadliwych i zostały porzucone, okazało się, że informacje o budżecie filmu jak i jego zyskach istnieją tylko 5500 filmów zatem zmniejszono znaczenie współczynnika B w stosunku do planowanego, a reszcie przypisano 0 w tym parametrze. Dodano również fluktuacje no poziomie 2%, żeby generowane cząsteczki filmów nie pokrywały się dla tych samych ocen co niwelowało potrzebę tworzenia bardziej skomplikowanych systemów umieszczania cząsteczek.
Przestrzeń wszystkich filmów, rozmieszczona na podstawie oceny i zysków. Oś pozioma proporcjonalna do średniej oceny danego filmu, oś pionowa proporcjonalna do siły oddziaływań między cząsteczką filmu, a cząsteczką użytkownika w modelu.

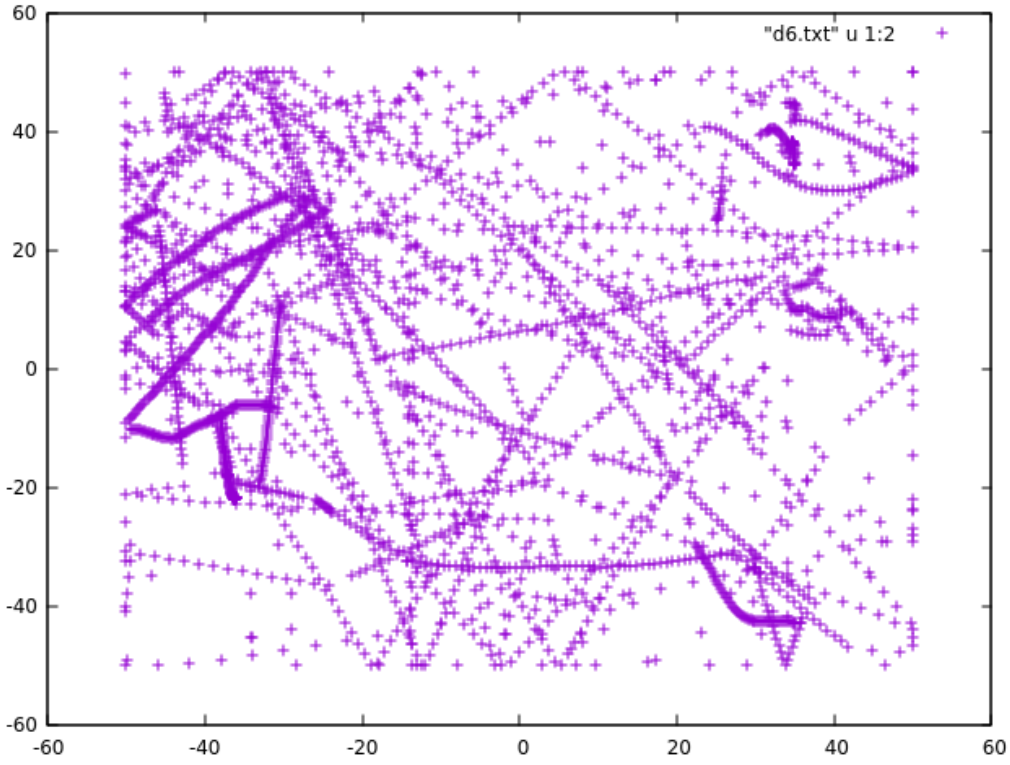


Symulacja
Ograniczenia sprzętowe i wydajność napisanej symulacji, wymagały ograniczenia liczby cząsteczek do kilkunastu tysięcy, ponieważ symulacja 10s samych zderzeń 10k filmów trwała 5,5h. Przestrzeń, w której zachodziła dyfuzja również została ograniczona kwadratową ramką, zależnie od maksymalnych położeń filmów na osiach. Zdecydowano się na ten krok ponieważ, bez niego cząsteczki zderzały się kilka razy i uciekały do nieskończoności.
Założenie o generowaniu dodatkowych cząsteczek na podstawie parametru związanego z popularnością zamieniono na, branie filmów z posortowanej listy pod tym względem, więc generowany gaz był złożony z najpopularniejszych filmów, czego wstępnie chciano uniknąć, ale dawało to najlepsze rezultaty, ponieważ popularne filmy mają znacznie skrupulatniej wypełnione dane, a co za tym idzie można dla większości z nich policzyć współczynnik związany z zyskami i ten związany z reżyserami. Dokładny wzór określający siłę oddziaływań znajduje się poniżej.

Poniżej ruch śledzonej cząsteczki jednego z filmów.



Poniżej ruch cząsteczki domyślnego użytkownika, poruszającego się w niespersonalizowanym gazie, generowanym tylko na podstawie średniej oceny i zysków.
Następnie z bazy danych reżyserów i filmów wygenerowano plik łączących reżyserów z danym filmem. Potem na podstawie ocen użytkownika przypisano do każdego reżysera współczynnik, będący różnicą między oceną użytkownika, a średnią oceną filmu, dodatni współczynnik świadczy, o tym, że użytkownikowi film zdecydowanie się podobał, analogicznie ujemny współczynnik świadczy o dużej niechęci do dzieła danego reżysera. Wygenerowane współczynniki zostały następnie przypisane do wszystkich filmów w bazie, które nakręcił każdy z reżyserów.



Ostatecznie wzór na siłę oddziaływań prezentuje się następująco:

$$F = A \cdot avg + B \cdot \ln\left(\frac{revenue}{budget}\right) + C \cdot \Delta ratio + fluctuations$$, gdzie A,B,C to dodatnie stałe

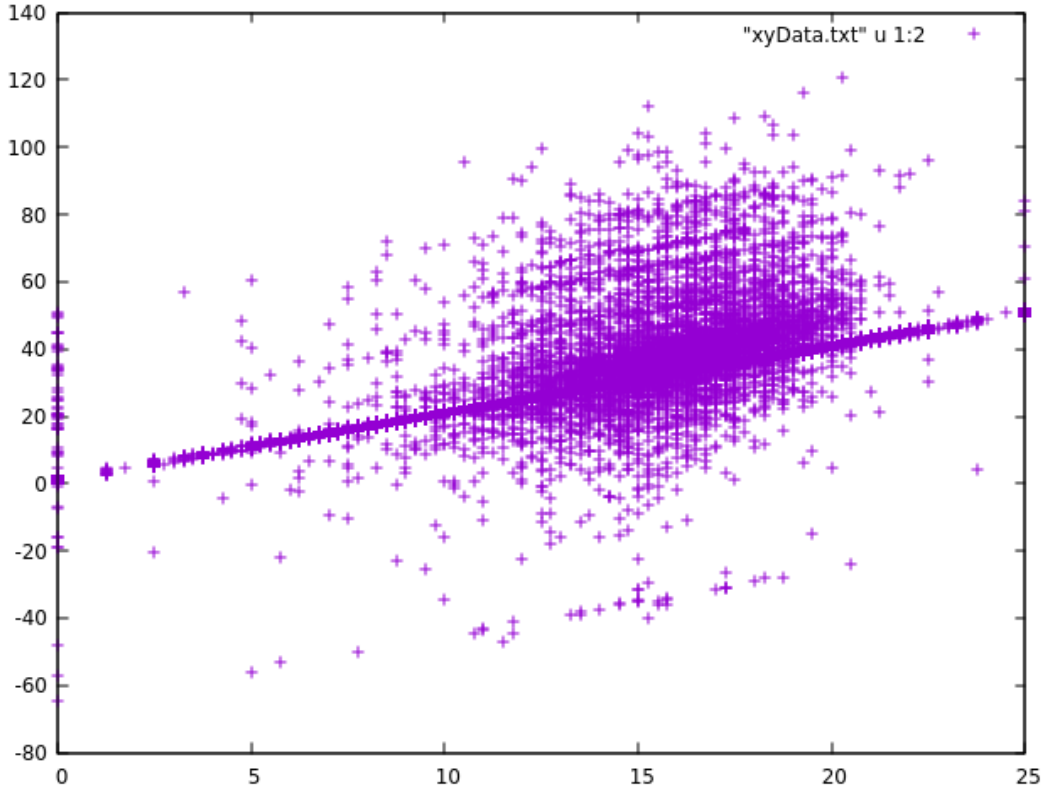
Przyjęte stałe określające znaczenie poszczególnych współczynników:
A=2; B=5; C=25

avg - średnia ocena filmu
revenue/budget – wielokrotność zysków danego filmu
Δratio – wyżej wspomniany współczynnik, generowany na podstawie oceny poprzednich filmów danego reżysera unikalny i przypisany tylko do niego. Wszystkie filmy w bazie danego reżysera mają ten sam współczynnik, na podstawie danej oceny. Jeżeli użytkownik obejrzał kilka to filmów tego samego reżysera to wszystkie jego filmy są duplikowane każdy z odpowiednim Δratio.

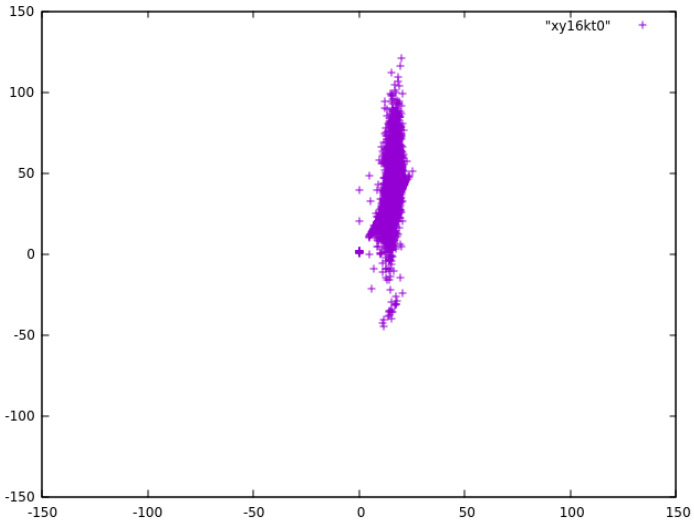
Na przykład

Oceny użytkownika				Metadata				Baza użyta do generacji gazu		
(title)	id	rating		id	avg		Δratio	id	avg	Δratio
Godfather	238	5		238	4,25		0,75	238	4,25	0,75
Godfather II	240	4,5		240	4,2		0,3	238	4,25	0,3
								240	4,2	0,75
								240	4,2	0,3

Wylosowano użytkownika 49, który ocenił 357 filmów.
Zatem po uwzględnieniu ocen użytkownika przestrzeń 45k filmów wygląda następująco:

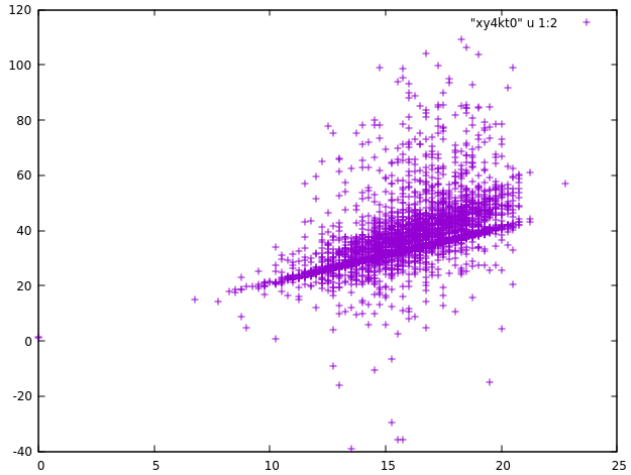


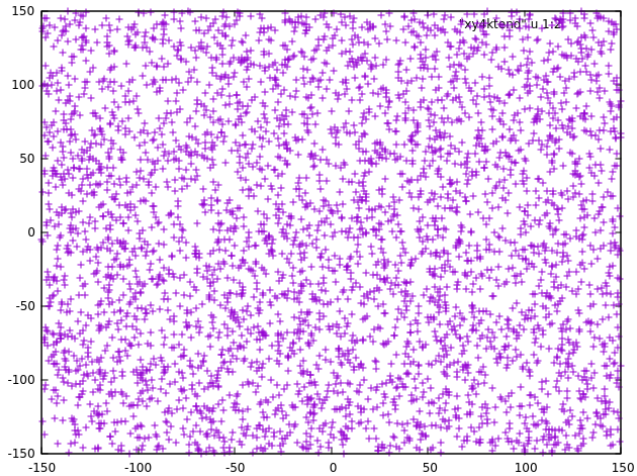
Widać znacznie większy rozrzut w porównaniu z przestrzenią generowaną tylko na podstawie ogólnych danych filmowych
Należy wspomnieć, że cząsteczki filmów były wzorowane na gazie doskonałym o dostatecznie dużej temperaturze, że rozkład Gaussa wystarczająco dobrze przybliżał rozkład Maxwella. Dla prędkości na osi X brano rozkład normalny o średniej 40 i odchyleniu standardowy równym 10, a dla prędkości na osi Y przesunięto prędkości o 30 jednostek w dół, w związku z początkowym rozmieszczeniem w górnym prawym rogu ograniczonego obszaru. Sens tego zabiegu dobrze obrazuje poniższy wykres, przedstawiający początkowe położenie w całym obszarze.



Poniżej dane z dwóch symulacji:
W obu przypadkach symulowano pierwsze 10 sekund i użyto jednej cząsteczki użytkownika. Masa cząsteczki filmu wynosiła 1, a promień 0.1, cząsteczka użytkownika miała podwojony promień i masę, w celu zwiększenia jej bezwładności, żeby nie zmieniała zbyt bardzo kierunku. W obu przypadkach również granica przestrzeni wynosiła ±150 jednostek, a siła oddziaływania była liczona na podstawie prostej analogii do siły odwrotnie proporcjonalnej do kwadratu odległości. Jeżeli użytkownik obejrzał kilka filmów danego reżysera to wszystkie jego filmy zostaną zwielokrotnione o tą ilość razy i każdy z współczynników zostanie uwzględniony.

- Symulacja 4k cząsteczek w chwili t=0s i t=10s (Czas symulowania 45min)





W związku z tym, że brano 4k najpopularniejszych filmów widać wspomnianą mniejszą ilość pustych rekordów, ponieważ zdecydowanie mniejsza ilość filmów układa się na linii

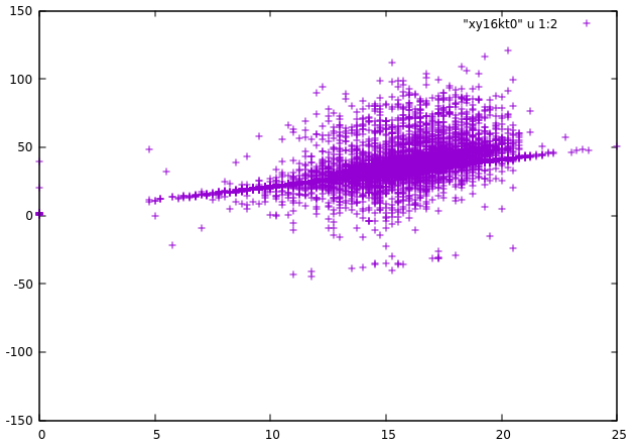
Otrzymane wyniki interakcji większe niż 1 (count -ilość interakcji)

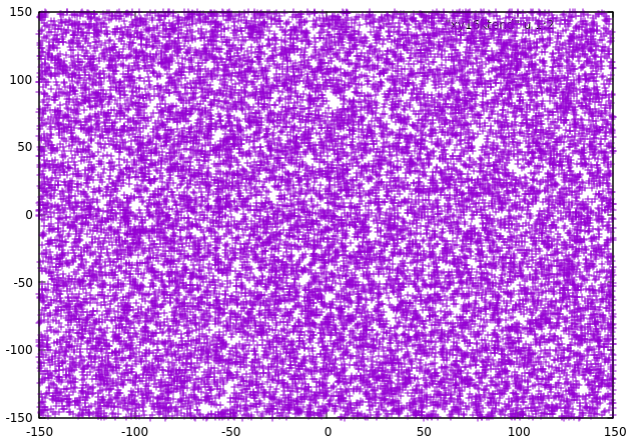
id	count	avg	popularity	profit	ratio	title
30596	3	3.25	2,39	−1,37	1,30	Bodyguards and Assassins
30596	3	3.25	2,39	−1,37	0,35	Bodyguards and Assassins
30596	3	3.25	2,39	−1,37	0,80	Bodyguards and Assassins
1688	2	3.05	2,55	1,74	0,00	Conquest of the Planet of the Apes
48988	2	2.70	2,21	0,84	0,00	Hall Pass

	Interakcje	4k cząsteczek
count	1,0362	-
avg	3,1256	3,19
popularity	2,4990	2,52
profit	0,2661	0,56
ratio	0,0342	0,06

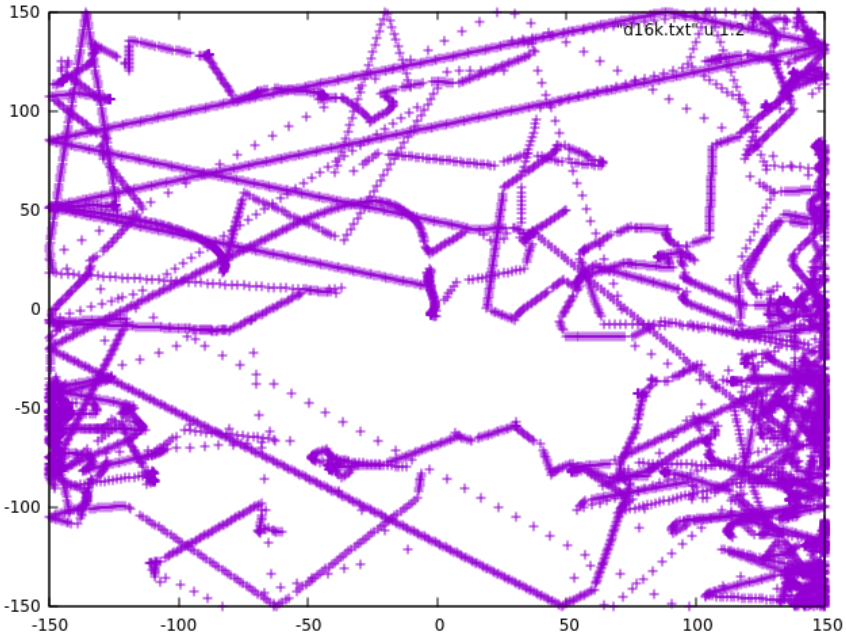
Zatem widać zakładane tendencje o tym, że cząstka użytkownika będzie wchodzić w interakcje z cząstkami o dodatnim zysku i współczynniku reżyserów. W przypadku pierwszego filmu mamy ujemny profit, ale współczynniki związane z reżyserami są dodatnie i mają 5 razy większą wagę przez co dominują w tym przypadku. Współczynnik związany z popularnością i średnią oceną jest niższy niż średnia, ale jest to związane z mniejszym procentowym wkładem do współczynnika związanego siłą.
Druga tabela zawiera średnią z danej kolumny, mimo, że wielokrotne interakcje wykazują zakładane tendencje to ogólnie wyniki są poniżej średniej. Dwie interakcje niewiele odstają od jednej, co świadczy o tym, że czas symulacji powinien być dłuższy, a samych cząsteczek filmów powinno być więcej.

- Symulacja 16k cząsteczek w chwili t=0s i t=10s (Czas symulowania 13h27min)





Położenie cząsteczki użytkownika podczas symulacji



Widać, że cząstka użytkownika spędza sporo czasu po obu stronach granicy obszaru, a nie jak planowano w górnym prawym rogu. Spowodowane jest to dużą ilością filmów o ujemnym parametrze siły, czyli odpychającym charakterze, tworzą one zwartą grupę przez co cząstka użytkownika jest spychana do brzegu i nie może się wydostać. Za to w górnej połowie przestrzeni widać, planowane przyciąganie do cząsteczek filmów i dużą ilość interakcji.

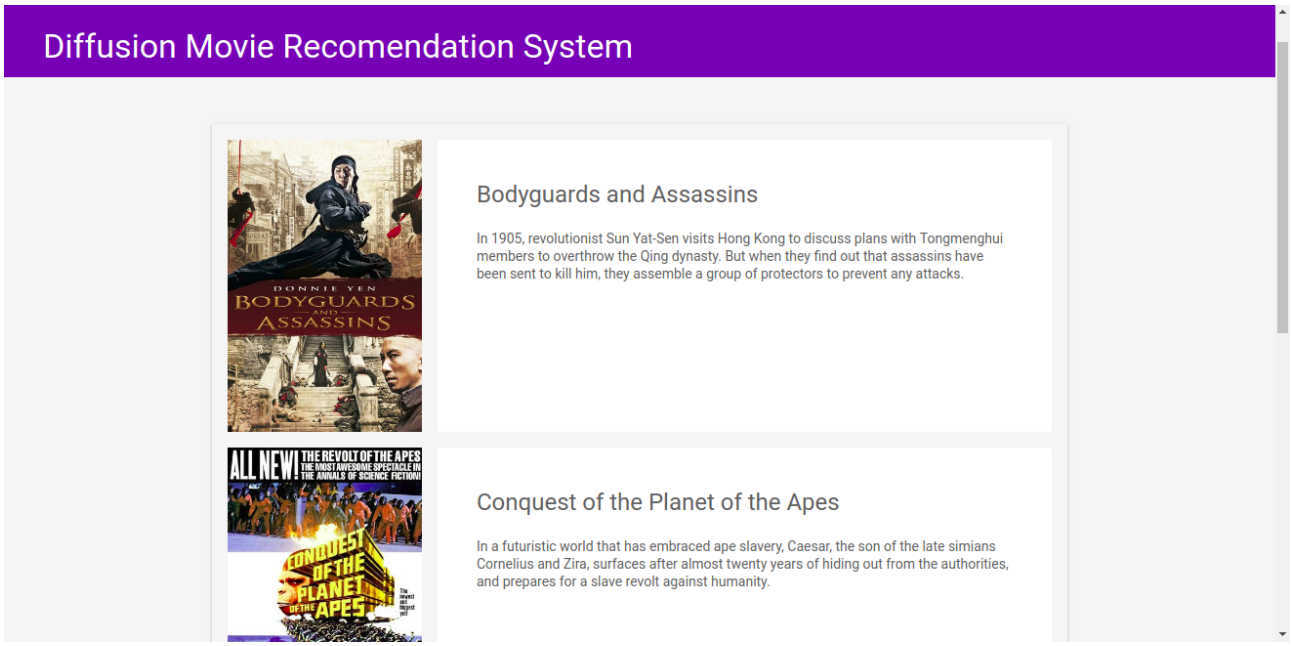
Otrzymane interakcje:

id	count	avg	popularity	profit	ratio	title
9993	2	2,95	1,4028	0,00	1,2998	Little Fish
9993	2	2,95	1,4028	0,00	0,7998	Little Fish
9993	2	2,95	1,4028	0,00	0,3501	Little Fish
265189	2	3,4	2,4986	0,00	0,0	Force Majeure
265189	2	3,4	2,4986	0,00	0,0	Force Majeure
25541	1	3,5	0,9509	0,00	0,0	Brotherhood
25541	1	3,5	0,9509	0,00	0,0	Brotherhood
25541	1	3,5	0,9509	0,00	0,0	Brotherhood
25541	1	3,5	0,9509	0,00	0,0	Brotherhood
4912	1	3,3	2,4275	0,0957	0,0	Confessions of a Dangerous Mind
4912	1	3,3	2,4275	0,0957	0,0	Confessions of a Dangerous Mind
4912	1	3,3	2,0342	0,0957	0,0	Confessions of a Dangerous Mind
4912	1	3,3	2,0342	0,0957	0,0	Confessions of a Dangerous Mind

	Interakcje	16k
count	1,095	-
avg	3,106	3,094
popularity	1,747	1,747
profit	0,108	0,173
ratio	0,053	0,046

Analizując powyższą tabelę widać, że otrzymaliśmy wyniki powyżej średniej dla całego zbioru interakcji, co świadczy, o tym, że model z większą ilością cząstek był dokładniejszy tak jak przewidywano.

Stworzono dokumenty html w celu prezentacji otrzymanych wyników, znajdują się w folderze Presentation.



Wnioski

Okazało się, że znaleziona baza danych jest bardziej dziurawa niż się z początku wydawało, przez co wymagana była modyfikacja kilku założeń z poprzednich etapów. Projekt zdecydowanie wymaga optymalizacji sprawdzania kolizji, takiej jak na przykład podzielenie przestrzeni i sprawdzanie zderzeń tylko w danych sektorach. Można by usprawnić również zliczanie, ponieważ jeśli chodzi o symulacje 16k filmów to cząsteczka użytkownika potrafiła wracać, do tej samej cząstki kilka razy w dużych odstępach czasowych i były to właśnie filmy, które chciałbym żeby algorytm polecał.

Opis skryptów i klas:
/facets-master – googlowski projekt do wizualizacji dużych zbiorów danych
/prog – zawiera klasę movie i user i pliki odpowiedzialne za symulację
/data – zawiera obrobione dane i skrypty służące do obróbki
directorStats.py – wyciąga z bazy reżyserów danego filmu
endStat.py – generuje plik potrzebny do stworzenia strony
interactionStat.py – generuje i sortuje listę interakcji na podstawie pliku z symulacji
movieStat.py – pobiera i obrabia metadane filmów z bazy
Stats.py – tworzy histogramy dotyczących użytkowników
userStats.py – łączy i personalizuję bazę filmów o współczynnik związany z reżyserami

Źródła

Baza danych 45 tysięcy filmów.
https://www.kaggle.com/rounakbanik/the-movies-dataset/data

Książki
Diffusion. *Mass Transfer in Fluid Systems*. E. L. Cussler.
Anomalous Transport: Foundations and Applications, Rainer Klages, Gunter Radons, Igor M. Sokolov.
Diffusion in Gases and Porous Media, R.E. Cunningham, R.J.J. Williams
Diffusion in Solids, A.S. Nowick, J.J.Burton
The Mathematics of Diffusion, J.Crank

https://en.wikipedia.org/wiki/Continuity_equation
https://en.wikipedia.org/wiki/Recommender_system
https://en.wikipedia.org/wiki/Netflix_Prize
http://www.filmweb.pl/help#I.6
http://home.agh.edu.pl/~dabrowa/files/4.Rownania-dyfuzji-wst-p-teoretyczny.pdf
http://ericleong.me/research/circle-circle/#dynamic-circle-circle-collision