Wat is de opdracht ?
(je kan stukje voor stukje laten aftekenen)

Programmeer in een OO taal (liefst Java of C#). PHP word took toegestaan
Het is een individuele opdracht !
Je moet alles zelf maken
Je moet al je eigen code kunnen uitleggen
Je moet OO dwz opdelen in deel stukken
Let op single responsibility
Je mag een geheel uitgewerkte opdracht ter beoordeling aanbieden.
Je mag maximaal 2 keer iets ter beoordeling aanbieden
Zorg dat je een goede REAME (installatie, uitleg opzet programma enz) file toevoegt
Elke opdracht heeft een eigen map
Zorg dat je printscreens van je werkende programma toevoegd

Opdracht 1: User-Item
Beoordeling
3,5 + 0,75 cijfer punt voor User Item
- 1,5 punt voor basis
- 1 punt voor nearest neigbour
- 1 grouplens groeplens dataset 100K
- 0,75 punt voor: hoe zorg je er voor dat het bepalen van uiteindelijke advies (niet het inlezen) snel gaat

Opdracht 2 : Item-Item
3,5 + 0,75 cijfer punt voor Item-Item
- 1 punt voor opstellen van item-item tabel
- 1 punt voor one slope
- 1 punt voor groeplens dataset 100K
- 0,75 punt voor hoe zorg je er voor dat het bepalen van uiteindelijke advies (niet het inlezen) snel gaat

Opdracht 3 : Apriori
1,5 cijfer punt voor Association rule
- 0,3 punt : voor dataset vinden
- 0,6 punt : gevonden Apriori programma werkend krijgen
- 0,6 punt : berekenen van lift , support, confidence

Deel 1 User-Item :

1) Start data set : userid, ranking
   a. Inlezen data
   b. Bepalen voor wie (welke userid; stel user X) een recommendation wordt gemaakt
   c. Vergelijk gekozen user X met alle andere users
      i. Mate van overeenkomst mbv similarity fomules = wegingscoëfficiënt
      ii. Welke extra producten (items) heeft een de andere aanbevolen
         1. Geen dan overslaan
         2. Zo ja welke hoe per extra product bij de ranking en het gewichts (wegingscoëfficiënt) bij
   d. Na dat alle andere user zijn nagelopen, bepaal je per item (aan de hand van de wegingscoëfficiënten in ranking) de ranking per product
   e. Sorteer de product (item) lijst op ranking
   f. Toon product informatie + ranking gesorteerd op ranking
2) Simalarity formules

Tav similarity formules
Vergelijkingsalgoritme (runtime veranderbaar dwz maak gebruik van een design pattern) dwz toepassen van strategy pattern. Invoer : double[] X en double[] Y

A. Pearson

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad \textit{(Eq.3)}$$

where:

- n is sample size
- x_i, y_i are the individual sample points indexed with i
- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (the sample mean); and analogously for $\bar{y}$

B. Euclidean

The **Euclidean distance** between points **p** and **q** is the length of the line segment connecting them ($\overline{\mathbf{pq}}$).

In Cartesian coordinates, if **p** = (p_1, p_2,..., p_n) and **q** = (q_1, q_2,..., q_n) are two points in Euclidean *n*-space, then the distance (d) from **p** to **q**, or from **q** to **p** is given by the Pythagorean formula:[1]

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

C. Cousine

The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\|\,\|\mathbf{B}\|\cos\theta$$

Given two vectors of attributes, *A* and *B*, the cosine similarity, cos(θ), is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

User – Item testen op basis data set

dataset :	Artikelen : 101 t/m 106	0:: entrykey=1 userid=1

userid, itemid, rating

```
1,101,2.5
1,102,3.5
1,105,2.5
3,104,3.5
2,102,3.5
1,106,3.0
2,101,3.0
5,103,2.0
2,103,1.5
2,104,5.0
6,105,3.5
2,106,3.0
3,101,2.5
6,104,5.0
3,102,3.0
5,106,3.0
3,106,4.0
4,102,3.5
6,106,3.0
4,104,4.0
7,104,4.0
4,105,2.5
1,104,3.5
1,103,3.0
2,105,3.5
4,106,4.5
5,101,3.0
5,102,4.0
6,102,4.0
5,104,3.0
4,103,3.0
5,105,2.0
6,101,3.0
7,102,4.5
7,105,1.0
```

Gebruikers : 1 t/m 7
Waardering : 1.0 t/m 5.0

	101	102	103	104	105	106
1	2.5	3.5	3.0	3.5	2.5	3.0
2	3.0	3.5	1.5	5.0	3.5	3.0
3	2.5	3.0	-	3.5	-	4.0
4	-	3.5	3.0	4.0	2.5	4.5
5	3.0	4.0	2.0	3.0	2.0	3.0
6	3.0	4.0	-	5.0	3.5	3.0
7	-	4.5	-	4.0	1.0	-

(101,2.5) (102,3.5) (103,3.0) (104,3.5) (105,2.5) (106,3.0)
1:: entrykey=2 userid=2
(101,3.0) (102,3.5) (103,1.5) (104,5.0) (105,3.5) (106,3.0)
2:: entrykey=3 userid=3
(101,2.5) (102,3.0) (104,3.5) (106,4.0)
3:: entrykey=4 userid=4
(102,3.5) (103,3.0) (104,4.0) (105,2.5) (106,4.5)
4:: entrykey=5 userid=5
(101,3.0) (102,4.0) (103,2.0) (104,3.0) (105,2.0) (106,3.0)
5:: entrykey=6 userid=6
(101,3.0) (102,4.0) (104,5.0) (105,3.5) (106,3.0)
6:: entrykey=7 userid=7
(102,4.5) (104,4.0) (105,1.0)

Opzoek naar medegebruikers die bij de gekozen gebruiker passen

Kies een gebruiker : bijvoorbeeld gebruiker nummer 7
Let op : de persoon 7(de gekozenen) heeft geen waardiging gegeven aan artikel 101, 103 en106
We gaan op zoek naar gebruikers die het beste passen bij gebruiker nummer 7
HOE : pearson correlation coefficient of euclidian distance

Pearson

Persoon nummer 7 vergeleken met de andere personen

user	pearson	euclidean distance similarity
1	0.9912407071619304	0.3483314773547883
2	0.38124642583151175	0.25824569976124334

3	-0.9999999999999998	0.38742588672279304
4	0.8934051474415644	0.3567891723253309
5	0.924473451641905	0,4
6	0.6628489803598702	0.2674788903885893

Opzoek naar artikelen die de gekozen persoon NIET heeft gewaardeerd en de anderen wel
Per persoon : userid, pearson , rating (persoon wel, gekozen NIET)
Let op : neem alleen die personen in de lijst op die minimaal 1 (extra) artikel hebben gewaardeerd

	101	**102**	**103**	**104**	**105**	**106**		**gelijk**	**verschillend**
1	2.5	3.5	3.0	3.5	2.5	3.0		(102,3.5) (104, 4.0) (105,2.5)	(101,2.5) (103,3.0) (106,3,0)
2	3.0	3.5	1.5	5.0	3.5	3.0		(102,3.5) (104, 5.0) (105,3.5)	(101,3.0) (103,1.5) (106,3,0)
3	2.5	3.0	-	3.5	-	4.0		(102,3.5) (104, 3.5)	(101,2.5) (106,4,0)
4	-	3.5	3.0	4.0	2.5	4.5		(102,3.5) (104, 4.0) (105,2.5)	(103,3.0) (106,4.5)
5	3.0	4.0	2.0	3.0	2.0	3.0		(102,4.0) (104, 3.0) (105,2.0)	(101,3.0) (103,2.0) (106,3,0)
6	3.0	4.0	-	5.0	3.5	3.0		(102,4.0) (104, 5.0) (105,3.5)	(101,3.0) (106,3,0)
7	-	4.5	-	4.0	1.0	-			

Totaal verschillende items : 101, 103, 106
userid=1 similarity =0.9912407071619304
(101,2.5) (103,3.0) (106,3.0)
userid=4 similarity =0.8934051474415644
(103,3.0) (106,4.5)
userid=5 similarity =0.924473451641905
(101,3.0) (103,2.0) (106,3.0)
voorbeeld: user 7: itemid
=101 sim=0.9912407071619304 rating_item_101=2,5 sim*rating=2.478101767904826

	similarity	**101**		**103**		**106**	
userid	**pearson**	**ranking**	**ranking * pearson**	**ranking**	**ranking * pearson**	**ranking**	**ranking * pearson**
1	0.9912407071619304	2,5	2.478101767904826	3.0	2.9737221214857 91	3	2.973722121485791
4	0.8934051474415644	-	-	3.0	2.6802154423246933	4,5	4.020323163487 04
5	0.9244734516419 05	3	2.773420354925715	2.0	1.8489469032838 1	3	2.773420354925715
		sumpears 101	sum 101	sumpears 106	sum 103	sumpears 106	sum 106
		1.915714	5.2515211	2.809119	7.502884	2.809119	9.767466
	predicted ranking	5.2515211/1.915714= 2,7412869		7.502884/2.809119= 2,670903		9.767466/2.809119= 3,477056	

Vervolgens de artikelen sorteren op RAKING en tonen !!

dwz **106, 101, 103**

itemid=106 predictedvalue=9.767466/2.8091192=3.4770563
itemid=101 predictedvalue=5.251522/1.9157141=2.741287
itemid=103 predictedvalue=7.5028844/2.8091192=2.6709027

3) **Nearest neigbbour** (**UserX** , parameters : 1) **N=minimaal aantal users**, T=drempel waarde (kiezen bijv 0.8), **SimilarityFomule** (bijv pearson) , **P= minimaal aantal aan te bevelen producten**)
    a. Doel je loopt de lijst met gebruikers af totdat je voldoende weet om een passende ranking te maken
    b. Algoritme:
    bepaal van elke (andere dan X) user (stel Y) de overeenkomst met user X
    Is deze waarde boven de drempel waarde
    Zo NEE volgende user
    Zo ja. Heeft de user Y andere artikelen aan bevolen
    Zo NEE dan volgende user
    Zo ja
    Zijn er al N gebruikers
    Nee neem gebruiker Y op
    Zo Ja : is vergelijking coëfficiënt beter dan diegene met de laagste. Neem deze op
    Is het aantal producten dat geadviseerd moet worden bereikt ?
    Zo NEE
    Volgende gebruiker
    Zo Ja return lijst met gebruikers (beter om de extra items met bijbehorende wegingsoëfficiënten door te geven)

4) **Test dataset** : Grote data set : grouplens.org (kies MovieLens 100K dataset)

Deel 2 Item-Item

1) Basis
   Cosinus adjustment

Cosinus adjustment formule
n items; m: users

$$s(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_u)^2}}$$

$$R_{u,i} = \textit{rating user u gives on item i}$$

$$\bar{R}_u \textit{ average user u for ALL items}$$

Stap 1: Opstellen van een tabel waarbij alle artikelen met elkaar worden verleken

We beginnen hier met het vergelijken van artikel 103 (eerste) met artikel 104 (tweede)
1.1 Allereerste wordt het gemiddelde van een gebruikers zijn aanbeveling uitgerekend (average rating R_u)
1.2 Vervolgens wordt er gebruik gemaakt van de cosinus adjustment formule om de twee artikelen met elkaar te vergelijken We nemen in de tabel ook op hoeveel users het artikel hebben aanbevolen

userid\itemid	**average rating** $\bar{R}_u$	**103**	**104**	**106**	**107**	**109**
1	3.25 (13/4)		3	5	4	1
2	3 (12/4)		3	4	4	1
3	2,75 (11/4)	4	3		3	1
4	3.2 (16/5)	4	4	4	3	1
5	4.25 (17/4)	5	4	5		3

User 3 gemiddelde = (4+3+3+1)/4=11/4=2,75
User 4 gemiddelde = (4+4+4+3+1)=16/5=3,2
User 5 gemiddelde = (5+4+5+3)=17/4=4,25

S(103,104) = (user 1 + user 2 + user 3+ user 4)/ (Item_103_waardering * item_104_waardering
User 1 : item 103 => 4 - 2,75 item 104 : 3 - 2,75
User 2 : item 103 => 4 – 3,2 item 104 : 4 – 3,2
enz

$$s(103,104) = \frac{\sum_{u \in Users}(R_{u,103} - \bar{R}_u)(R_{u,106} - \bar{R}_u)}{\sqrt{\sum_{u \in Users}(R_{u,103} - \bar{R}_u)^2}\sqrt{\sum_{u \in Users}(R_{u,106} - \bar{R}_u)^2}}$$

$$s(103,104) = \frac{(4-2.75)*(3-2.75)+(4-3.2)*(4-3.2)+(5-4.25)*(4-4.25)}{\sqrt{(4-2.75)^2+(4-3.2)^2+(5-4.25)^2}*\sqrt{(3-2.75)^2+(4-3.2)^2+(4-4.25)^2}}$$

$$\frac{0.7650}{\sqrt{2.765}*\sqrt{0.765}} = 0.525997$$

tabel cel: *similarity waarde(aantal users)*

-	109	107	106	104
103	-0.9549	0.3210	1.00	0.5260
104	0.3378	-0.2525	0.0075	
106	-0.9570	0.7841		
107	-0.6934			

tabel cel: *similarity waarde(aantal users)*

-	104	106	107	109
103	0.5260(3)	1.00(2)	0,321(2)	-0.9549(3)
104		0.0075(4)	-2.525(4)	0.3378(5)
106			0.7841(3)	-0.9570(4)
107				-0.6934(4)

Prediction

$$predicition(u,i) = p(u,i) = \frac{\sum_{j \in sim(i)} (S_{i,j} \times R_{u,j})}{\sum_{j \in sim(i)} (|S_{i,j}|)}$$

p(u,i)= predicted rating for user u of item i
j = all items for which there is a similarity value with i and which user u rated
- $S_{i,j}$ =sim(i,j) → similarity between items similarity between items i and j
  $R_{u,j}$= rating that user u gave to item j

For the formula to work best, $R_{u,j}$ should be between -1 to 1
The rating is needed to normalize :

	normaliseren	terug uit rekenenen
formules	$r_{normalized} = r_n = 2 * \frac{r - r_{min}}{r_{max} - r_{min}} - 1$	$r = (\frac{r_n + 1}{2}) \times (t_{max} - r_{min}) + r_{min}$
voorbeeld rating=2 range[1-5]	$r_n = 2 * \frac{2 - r_{min}}{5 - 1} - 1$ $r_n = -0,5$	$r = (\frac{-0.5 + 1}{2}) \times (5 - 1) + 1$ r=2

p(u,i)= predicted rating for user u of item i
j = all items for which there is a similarity value with i and which user u rated
- $S_{i,j}$ =sim(i,j) → similarity between items similarity between items i and j
  $R_{u,j}$= rating that user u gave to item j

For the formula to work best, $R_{u,j}$ should be between -1 to 1

Rating for user =1 Gevraagd : $Rating_{normalized}$

userid\itemid	**103**	**104**	**106**	**107**	**109**
1		3	5	4	1
$r_{normalized}$		0	1	0.5	-1

Set= {3,5,4,1} = > min=1 en max=5
stel r=3 rnorm= 2 * (3-1)/(5-1)

Gevraag : Rating for user =1 for item =1

-	104	106	107	109
103	0.5260(3)	1.00(2)	0,321(2)	-0.9549(3)
104		0.0075(4)	-2.525(4)	0.3378(5)
106			0.7841(3)	-0.9570(4)
107				-0.6934(4)

prediction (u=1,item=103)=?

$$p(1,103) = \frac{(r_n104 * sim(103,104)) + (r_n106 * sim(103,106)) + (r_n107 * sim(103,107)) + (r_n109 * sim(103,109))}{abs(sim(103,104)) + abs(sim(103,106)) + abs(sim(103,107)) + abs(sim(103,109))}$$

$$p(1,103) = \frac{(0 * 0.5260) + (1 * 1.000) + (0.5 * 0.321) + (-1 * -0.9549)}{0.5260 + 1.000 + 0.3210 + 0.9549}$$

$$p(1,103) = \frac{2.1154}{2.8019} = 0.754988$$

prediction(user=1,item=103) =0,755 (genormaliseerd)
rating user=1 item=103 =?

$$r = \frac{0.755 + 1}{2} \times (5 - 1) + 1 = 4.509975$$

2) Oneslope
======

1. Compute deviations between all pairs of items

$$dev_{i,j} = \frac{\sum_{u \in S_{i,j}} u_i - u_j}{card(S_{i,j})}$$

where - $S_{i,j}$= set of users which rated **both** items i and j
- $card(S_{i,j})$= number of users which rated both items
- u_i = rating of user ?u for item i (same for j)

example:

userid/itemid	103	106	109
1	4	3	4
2	5	2	?
3	?	3.5	4
4	5	?	3

$$verschil = dev_{103,106} = \frac{(4-3)+(5-2)}{2} = 2$$

$$verschil = dev_{106,109} = \frac{(3-4)+(3.5-4)}{2} = -0.75$$

$$verschil = dev_{103,109} = \frac{(4-4)+(3-5)}{2} = -1$$

2. Making predictions combining deviations and ratings

$$prediction(u,j) = p(u,i) = \frac{\sum_{j \in ratings(u)} (u_j + dev_{i,j}) \times card(S_{i,j})}{\sum_{j \in ratings(u)} card(S_{i,j})}$$

where = p(u,i)= predicted rating for user u for item i
are the items already rated by user u (u_j is the rating)
- $dev_{i,j}$= deviation between items i and j
- $card(S_{i,j})$= number of users which rated both i,j

example:

userid/itemid	103	106	109
2	5	2	?

Verschil= deviation matrix :
tabel cel : similarity (aantal users dat heeft bijgedragen = $card(S_{i,j})$)

itemid/itemid	103	106	109
103		2(2)	1(2)
106	-2		-0.75(2)

109	-1	0.75	

Gevraagd : voorspel de rating voor itemid=109 user= 2
Predict the reating itemid=109 for user=2
Hoe:
1 : kijk wat voor items user=2 heeft gewaardeerd
103 rating 5 en 106 met rating 2
2 : bepaald in de Oneslope tabel het verschil tenopzichte van het artikel waar van je een voorspelling wilt maken
Het gaat over itemid=109 (waarvan je wilt weten hoe deze persoon dit item zou waarderen
User 2 :
103 rating 5 tov 109 geeft een verschil van 1.0 (aantal users dat heeft bijgedragen =2)
106 rating 2 tov 109 geeft een verschil van -0.75 (aantal users dat heeft bijgedragen =2)

$$pred(user = 2, itemid = 109) = \frac{(5-1) \times 2 + (2+0.75) \times 2}{2+2} = \frac{13.5}{4} = 3.375$$

Denominator:
Dissecting the denominator we get something like for every item that user=2 has rated, sum the cardinalities of those musicians (how many people rated both that item and itemid=109).
So userid=2 has rated itemid=103 and the cardinality of itemid=103 and itemid=109 (that is, the total number of people that rated both of them) is 2.
Userid=2 has rated itemid=106 and his cardinality is also 2.
So the denominator is 4.

3) Grouplens dataset 100K
vergelijk performance verschil tussen User-Item en Item-Item

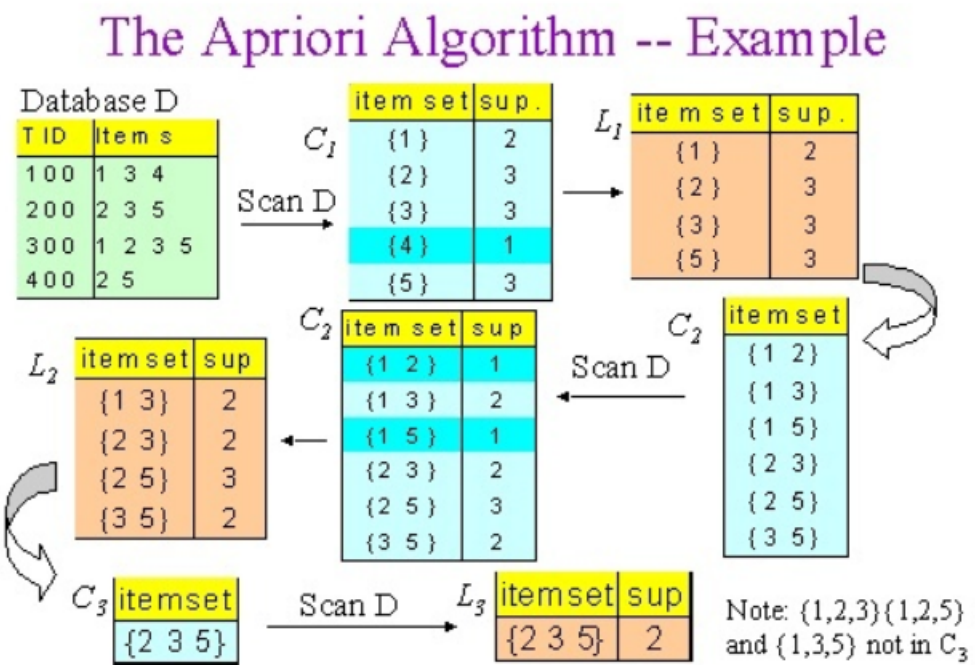
Deel 2 : Apriori (Association rule)

Association rule : Apriori
Hier gaat het niet om het zoveel mogelijke zelf coderen maar om maximaal gebruik te maken van bestaande oplossingen (die je op het internet kan vinden)

Maak gebruik van bestaande libraries
Java
http://www.philippe-fournier-viger.com/spmf/
https://gist.github.com/monperrus/7157717
Weka : https://www.youtube.com/watch?v=gdz9lc_6gTs

Je mag gebruik maken van code op internet (mist je maar aangeeft waar je het vandaan hebt)

Berekenen de support, confidence en de lift



Eisen aan het systeem

D. Vergelijkingsalgoritme (runtime veranderbaar dwz maak gebruik van een design pattern) dwz toepassen van stratepy pattern. Invoer : double[] X en double[] Y
   a. Pearson
   b. Euclidean
   c. cousine

E. User-Item
   a. Basis (ranking van (min) 5 artikelen op volgorde van ranking waarden)
   b. Het programma geeft prediction voor user Y de beste top X (P=2*X)

F. Item – Item
   a. Standaard (conform boek : http://guidetodatamining.com/)
   b. One-slope
   c. Prediction met bovenstaande technieken

G. Basis problemen oplossen
   a. Er is nog geen persoon die recommendation heeft (lege)
   b. Persoon heeft zoveel aan bevolen er is geen extra artikel is die kan worden aanbevolen
   c. (hoeft niet te programmeren) wel oplossing geven voor gegevens die je in de loop van de tijd steeds minder mee wilt tellen
   d. Probleem dat 80% van de mensen 20% van de artikelen kies (wel implementeren)

Code : OO , onderhoudbaar , vergelijkingsalgoritme niet vervuilen met context, design patterns, SOLID