Unsupervised Learning

What is unsupervised learning?

Unsupervised learning is a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabelled data.

Different unsupervised learning models

1. K-means clustering
2. K-NN (k nearest neighbours)
3. Topic modeling - LDA, NMF
4. Principal Component Analysis
5. Hierarchical clustering

Today's topics

1 Bag Of Words

- Count Vectoriser
- TF-IDF

2 Topic modeling

- Matrix decomposition
- NMF

3 Case study

1 Bag Of Words

- Count Vectoriser
- TF-IDF

1 Bag Of Words

- Count Vectoriser

- Count Vectoriser Matrix

Corpus

S1 Camera is good
S2 Camera resolution is good
S3 Battery is good. Battery is super
S4 Battery life is good

	Camera	is	good	resolution	battery	super	life
S1	1	1	1	0	0	0	0
S2	1	1	0	1	0	0	0
S3	0	2	1	0	2	1	0
S4	0	1	1	0	1	0	1

Individual representation of these samples can be treated as vectors

S1	1	1	1	0	0	0	0
S3	0	2	1	0	2	1	0

1 Bag Of Words

- TF-IDF

Corpus

S1 Camera is good
S2 Camera resolution is good
S3 Battery is good. Battery is super
S4 Battery life is good

- TF-IDF

	battery	camera	camers	good	is	life	resolution	super
1	0.000000	0.000000	0.804612	0.419880	0.419880	0.000000	0.000000	0.000000
2	0.000000	0.626884	0.000000	0.327134	0.327134	0.000000	0.626884	0.000000
3	0.716149	0.000000	0.000000	0.237006	0.474012	0.000000	0.000000	0.454172
4	0.25	0.000000	0.000000	0.354557	0.354557	0.679435	0.000000	0.000000

$$TFIDF\ score\ for\ term\ i\ in\ document\ j = TF(i,j) * IDF(i)$$

where

$IDF = Inverse\ Document\ Frequency$

$TF = Term\ Frequency$

$$TF(i,j) = \frac{\text{Term i frequency in document j}}{\text{Total words in document j}}$$

$$IDF(i) = \log_2\left(\frac{\text{Total documents}}{\text{documents with term i}}\right)$$

and

$t = Term$

$j = Document$

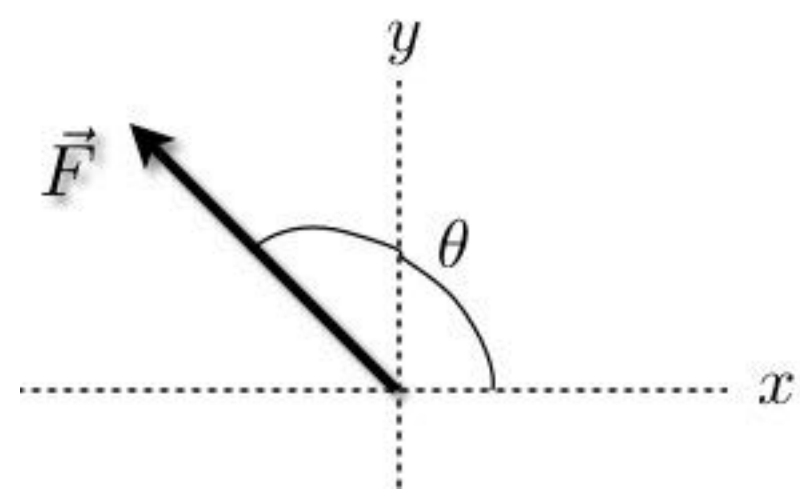
TF(battery,4) =
1/4 = 0.25

IDF(battery) =
= log2(4/2) = 1

TF-IDF(battery,4) =
= 0.25 * 1
=0.25

A document can be represented as a vector
Using

- Count Vectoriser
- TF-IDF
- Word embeddings
- Character embeddings

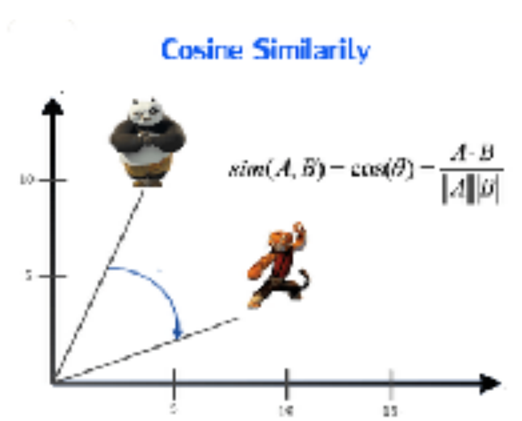


Let's say you represented vectors for 2 samples:

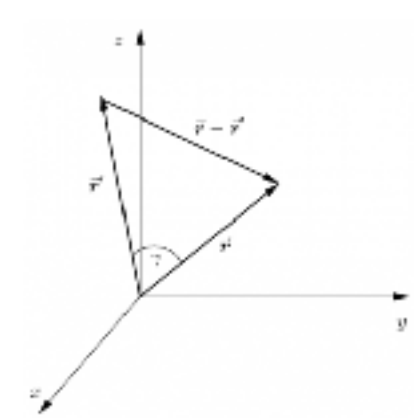
S1	1	1	1	0	0	0	0
S2	1	1	0	1	0	0	0

You can perform various operations with respect to those vectors

You can find similarity between these vectors

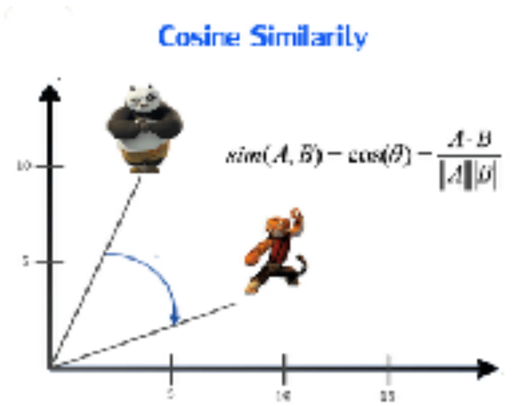


You can find distance between these vectors



S1	1	1	1	0	0	0	0
S2	1	1	0	1	0	0	0

How to find similarity between two vectors



S1 = [1, 1, 1, 0, 0, 0, 0]   S2 = [1, 1, 0, 1, 0, 0, 0]

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$$

$$\vec{S1} \cdot \vec{S2} = [1, 1, 1, 0, 0, 0, 0] \cdot [1, 1, 0, 1, 0, 0, 0]$$
$$= (1*1)+(1*1)+(1*0)+(0*1)$$
$$+(0*0)+(0*0)+(0*0)$$
$$= 2$$

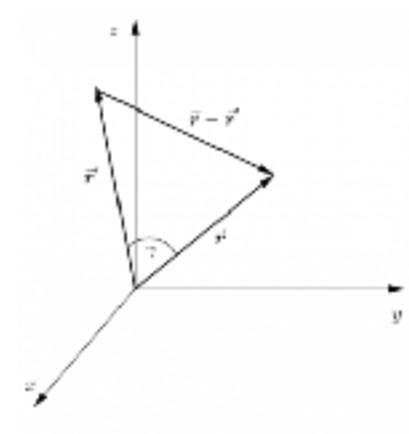
$$\|\vec{S1}\| = \sqrt{1^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2 + 0^2}$$

$$\|\vec{S2}\| = \sqrt{1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 0^2 + 0}$$

2

Sim = 2/3 = 0.666

How to find distance between two vectors



S1 = [1, 1, 1, 0, 0, 0, 0]   S2 = [1, 1, 0, 1, 0, 0, 0]

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$d = \sqrt{(1-1)^2 + (1-1)^2 + (1-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2}$$

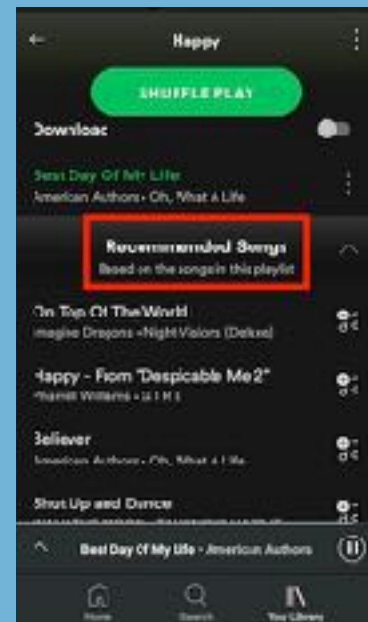
d =. 1.4

2 Topic modeling

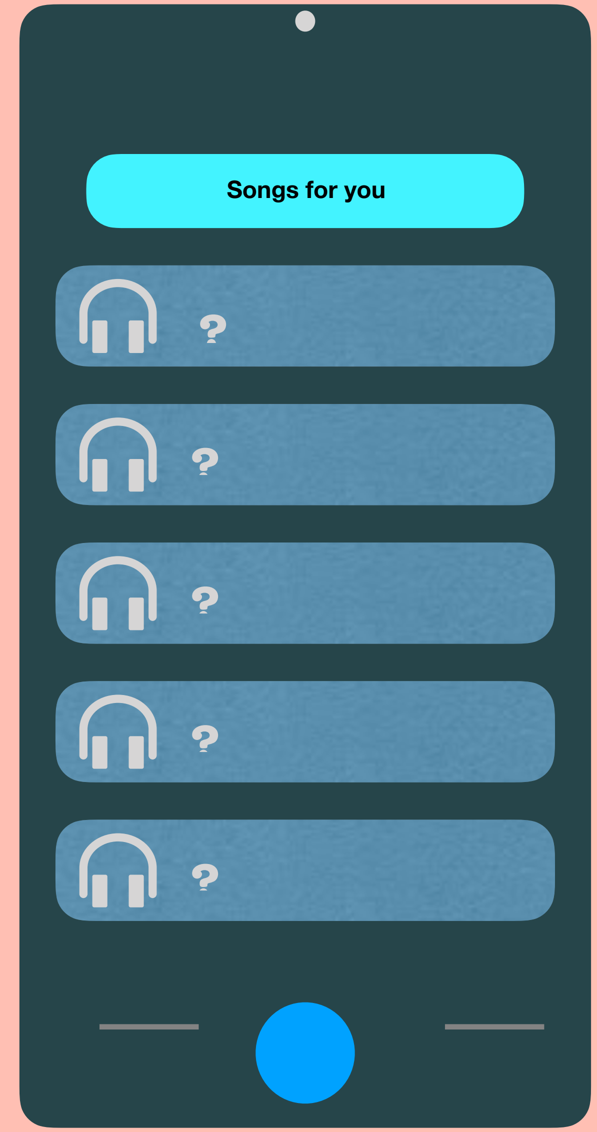
- Matrix decomposition
- NMF

3 Case study

Case study 1:



	user_id	song	score	Rank
3194	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Sehr kosmisch - Harmonia	37	1
4083	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Undo - Björk	27	2
931	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Dog Days Are Over (Radio Edit) - Florence + Th...	24	3
4443	4bd88bfb25263a75bbdd467e74018f4ae570e5df	You're The One - Dwight Yoakam	24	4
3034	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Revelry - Kings Of Leon	21	5
3189	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Secrets - OneRepublic	21	6
4112	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Use Somebody - Kings Of Leon	21	7
1207	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Fireflies - Charttraxx Karaoke	20	8
1577	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Hey_ Soul Sister - Train	19	9
1626	4bd88bfb25263a75bbdd467e74018f4ae570e5df	Horn Concerto No. 4 in E flat K495: II. Romanc...	19	10



spotify is a media-services provider founded in 2006. It is one of the biggest online platform of musics and songs. When they started the company, they wanted to build a small recommendation engine for there users. They have a dataset of users and the songs that they have listened to.

How would you help them to build a music recommendation system.

Matrix multiplication

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{pmatrix}$$

M * N **N * P**

Multiplication of two matrixes:

$$A * B = \begin{pmatrix} 1*5 + 2*8 & 1*6 + 2*9 & 1*7 + 2*10 \\ 3*5 + 4*8 & 3*6 + 4*9 & 3*7 + 4*10 \end{pmatrix}$$

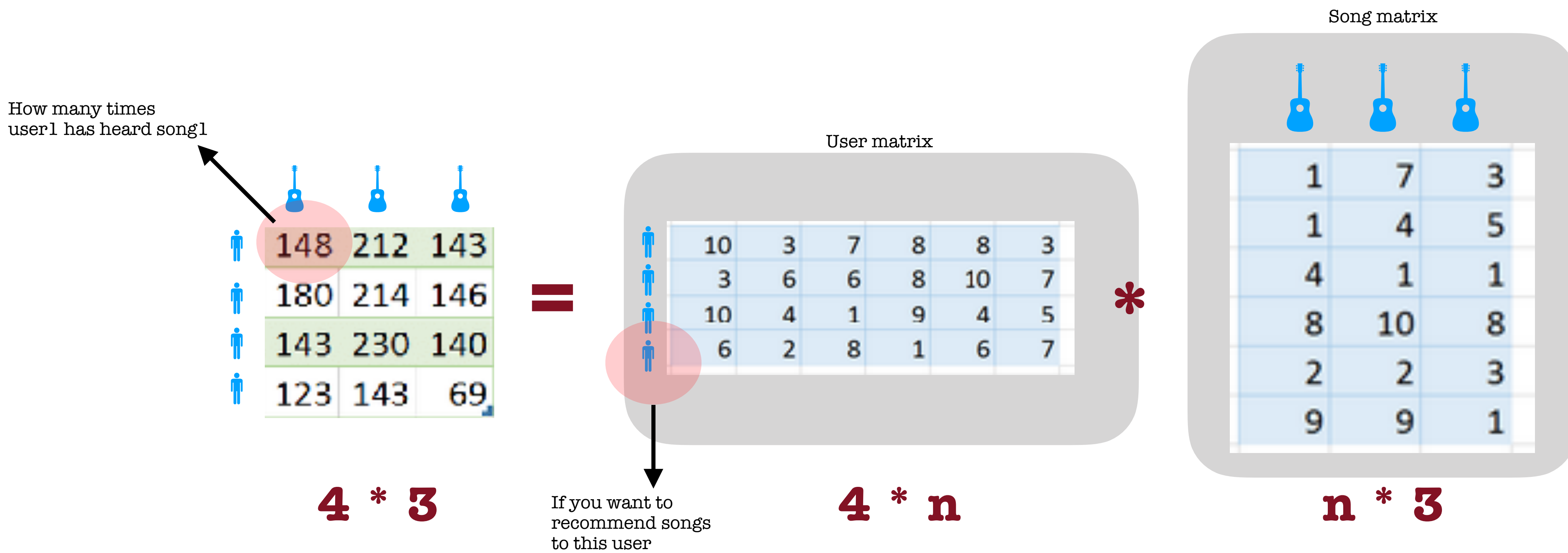
$$A * B = \begin{pmatrix} 21 & 24 & 27 \\ 47 & 54 & 61 \end{pmatrix}$$

M * P

[M * N] * [N * P] = [M * P]

$$[M * N] * [N * P] = [M * P]$$

$$[M * P] = [M * N] * [N * P]$$



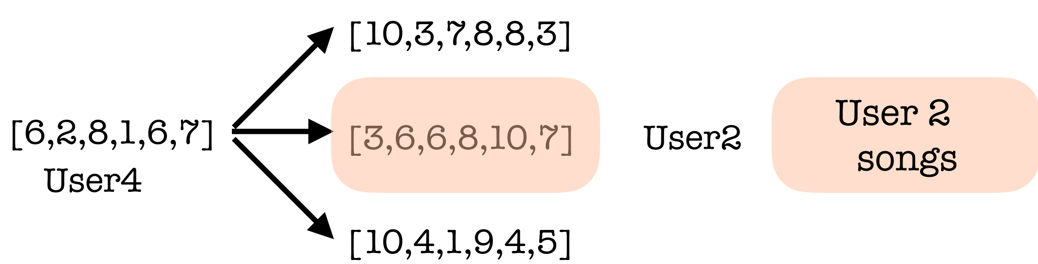
148	212	143
180	214	146
143	230	140
123	143	69

=

10	3	7	8	8	3
3	6	6	8	10	7
10	4	1	9	4	5
6	2	8	1	6	7

*

1	7	3
1	4	5
4	1	1
8	10	8
2	2	3
9	9	1



User matrix represent how well users are correlated
Song matrix represent how well songs are correlated

Where **n** can be any desired dimension

[Non-negative matrix factorization]

How does NMF decompose the matrix?

Inputs

$$V = \begin{array}{cc} & \text{6 columns} \downarrow \\ \text{3 rows} \rightarrow & \begin{bmatrix} 9 & -1 & 2 & 8 & -3 & 4 \\ 7 & -2 & 3 & 0 & 2 & -1 \\ 2 & -4 & 5 & 1 & 6 & 7 \end{bmatrix} \end{array}$$

Dimension d = 6

$$W = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \ldots\ldots & X_{1d} \\ X_{21} & X_{22} & X_{23} & \ldots\ldots & X_{2d} \\ X_{31} & X_{32} & X_{33} & \ldots\ldots & X_{3d} \end{bmatrix}$$

3 * d

$$H = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ X_{d1} & X_{d2} & X_{d3} & X_{d4} & X_{d5} & X_{d6} \end{bmatrix}$$

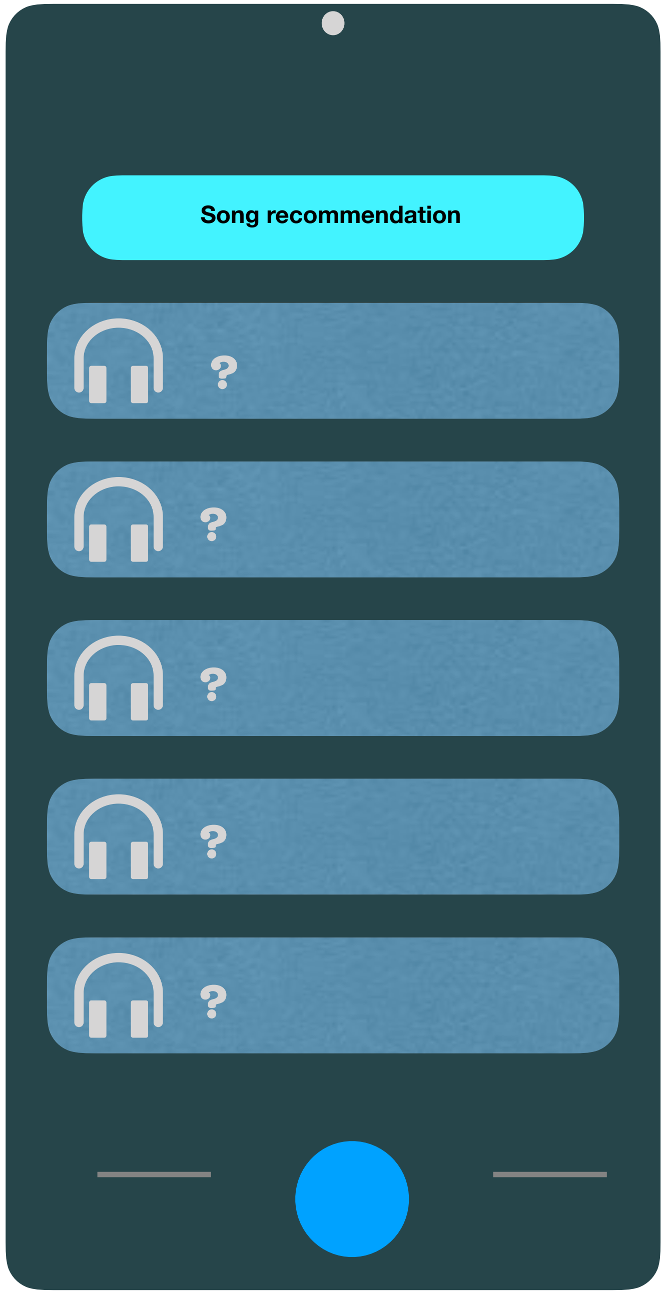
d * 6

$$H_{new} = H_{old} \frac{(W_{old}.T)\ V}{(W_{old}.T)\ W_{old}\ H_{old}}$$

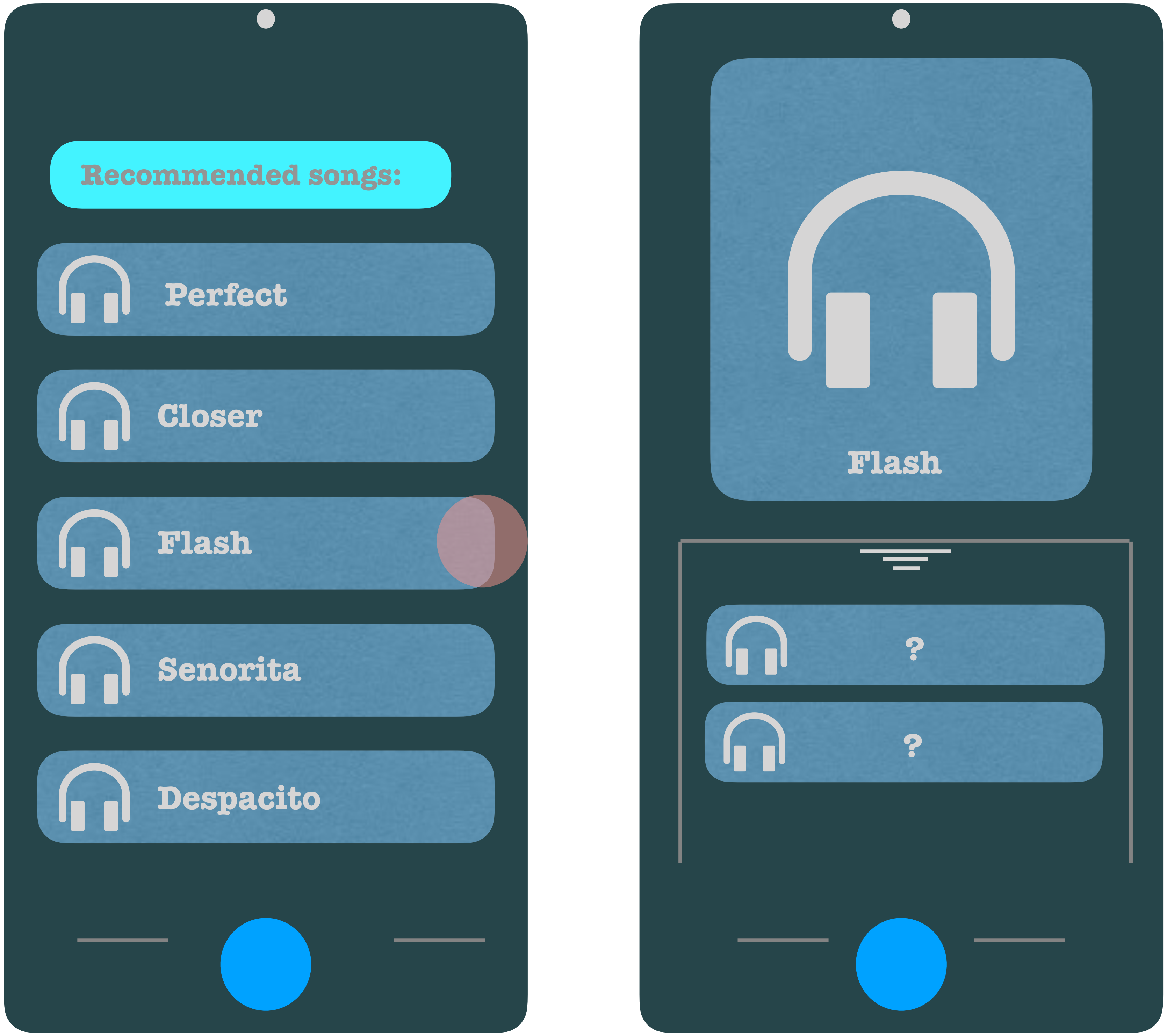
$$W_{new} = W_{old} \frac{V\ (W_{new}.T)}{W_{old}.\ H_{new}\ (H_{new}.T)}$$

Until H and W are stabilised

Case study 1
Solution



What's after creating recommendation list?



Will you decompose user song matrix?

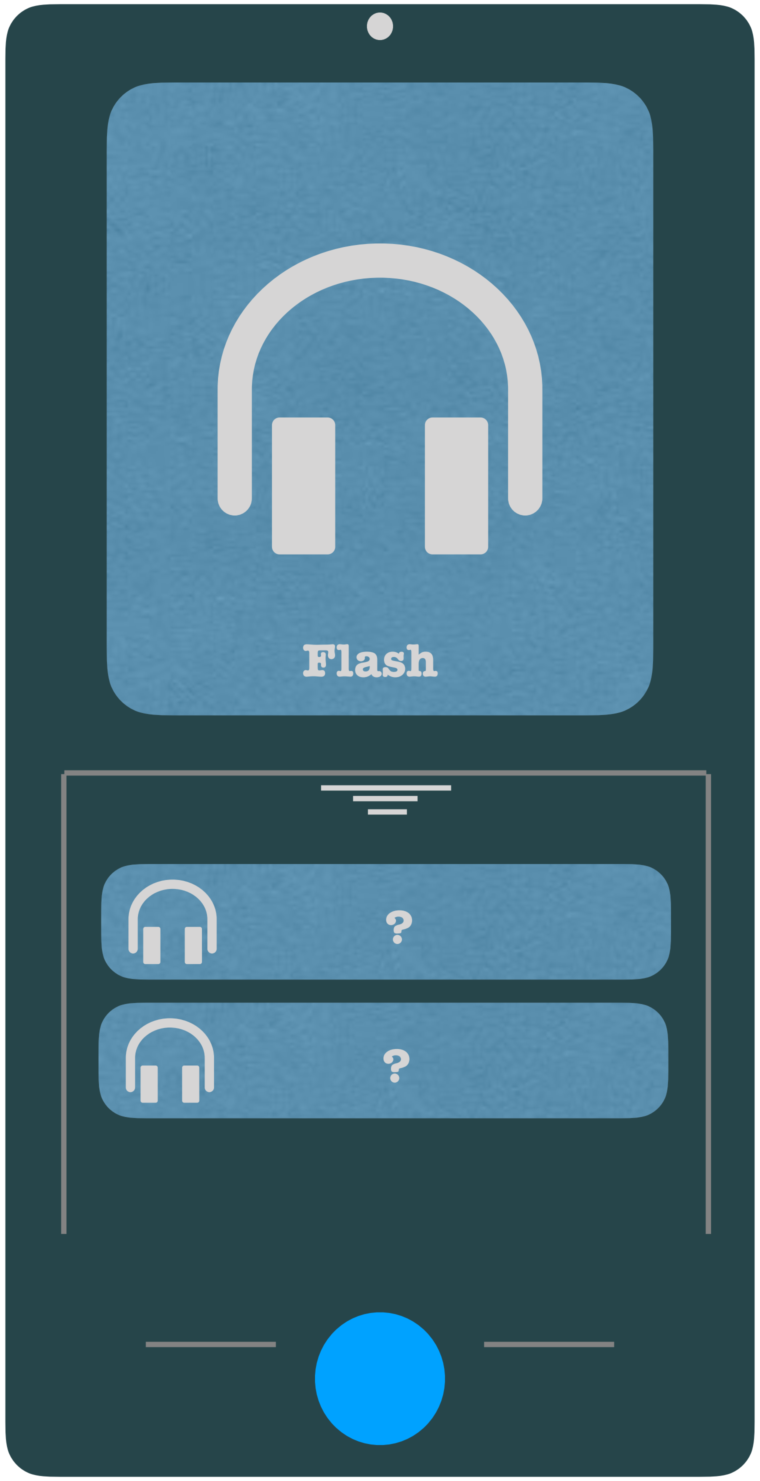
Can we use user matrix and do something?

Can we use song matrix and do something?

Can we use KNN and song matrix to give some recommendations?

Can we use KMeans, KNN and song matrix to give some recommendations?

Improvising
The recommendation



Introduction to
Case study 2:

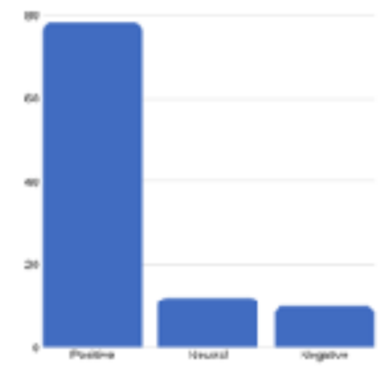
How is the camera of iPhone X?

Attribute model

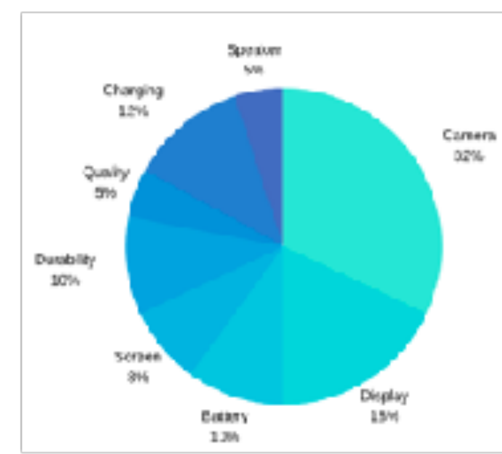
Attributes:

- Camera
- Display
- Battery
- Screen
- Durability
- Quality
- Charging
- Speaker

Product popularity



Product sentiment



Attribute popularity

Camera of this phone is very good and I fell in love with this phone...

URL: https://www.flipkart.com › ... › Apple iPhone 6s (Gold, 32 GB)

Summary: Camera is very good

positive

Camera of this phone is very good and I fell in love with this phone...

URL: https://www.amazon.com › ... › Apple iPhone 6s (Gold, 32 GB)

Summary: Camera is very good

positive

Camera of this phone is very good and I fell in love with this phone...

URL: https://www.snapdeal.com › ... › Apple iPhone 6s (Gold, 32 GB)

Summary: Camera is very good

negative

Camera of this phone is very good and I fell in love with this phone...

URL: https://www.amazon.com › ... › Apple iPhone 6s (Gold, 32 GB)

Summary: Camera is very good

neutral

Thank you