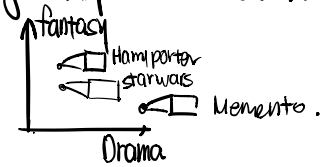
Building a simple Vector-based model

- content based filtering : uses item features to recommend new items that are similar to what the user has liked in the past.
- "learn How to Measure the similarity of elements in an embedding space"

Similarity Measures

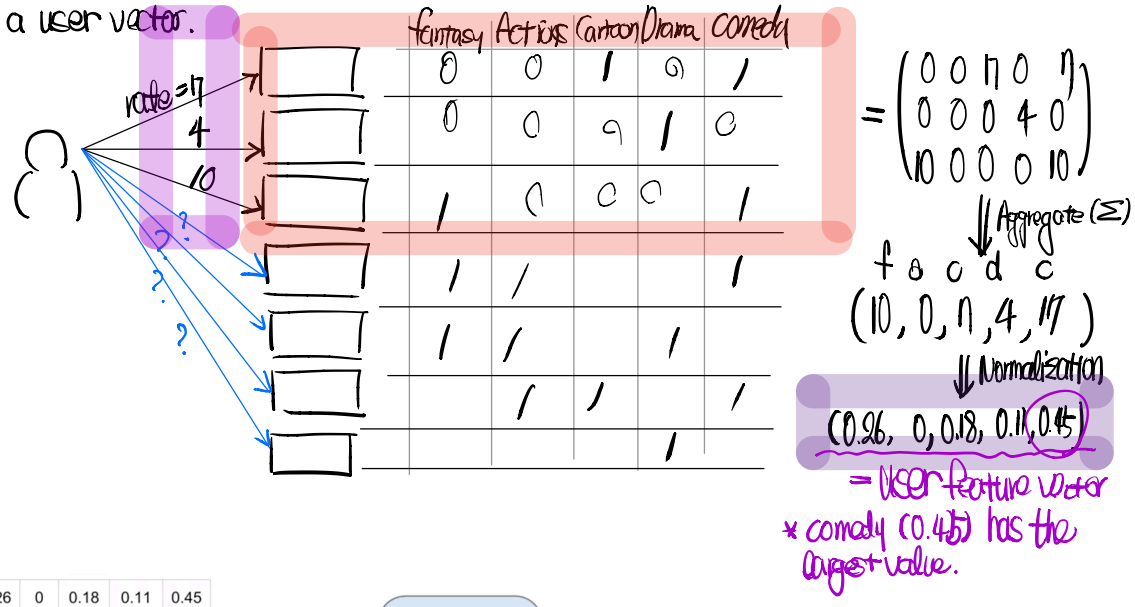
- embedding : map from our collection of items to some finite dimensional vector space.

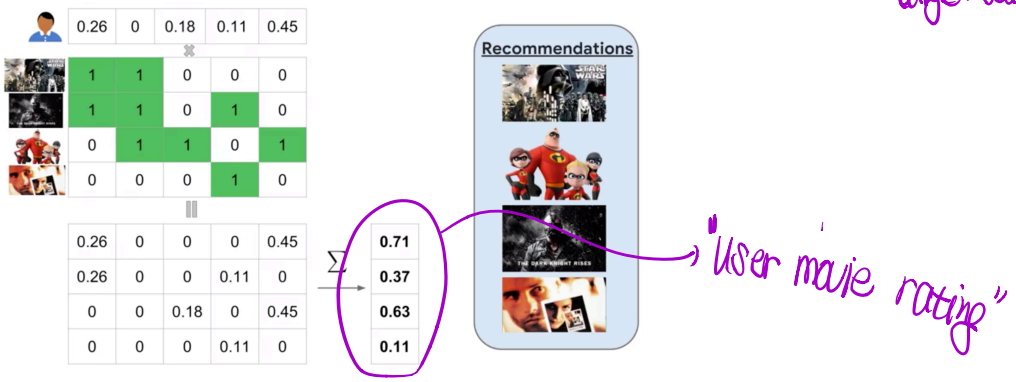
ex)



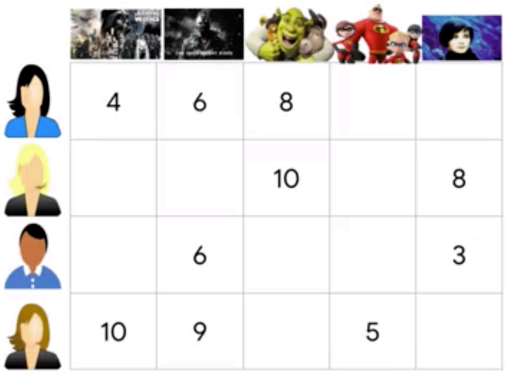
- Similarity measure : a metric for items in an embedding space
  - *) dot product.
  - *) cosine similarity

Building a user vector.



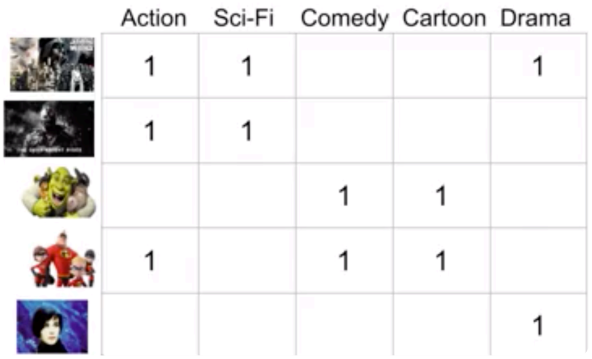


Making Recommendation for many users.



4	6	8		
		10		8
	6			3
10	9		5	

<user-item rating matrix>



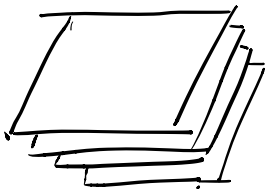
	Action	Sci-Fi	Comedy	Cartoon	Drama
	1	1			1
	1	1			
			1	1	
	1		1	1	
					1

<item-feature matrix>

- for first user.. "weighted feature matrix."

$$\begin{pmatrix} 4 & 4 & 0 & 0 & 4 \\ 6 & 6 & 0 & 0 & 9 \\ 0 & 0 & 8 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

... stack for every user.



tf.stack(value, axis=0, name=stack.)

- Quiz

What is the shape of the tensor resulting from stacking together all of the weighted feature matrices?

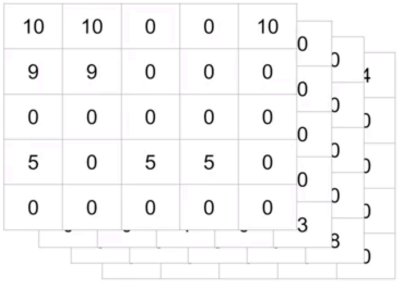
```
tf.stack(wgtd_feature_matrices, axis = 0)
```

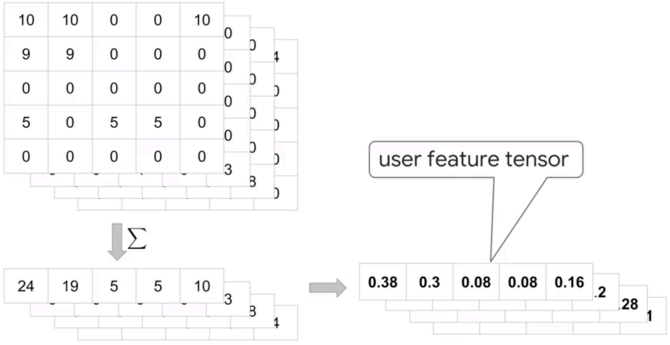
a) (# movies, # features)

b) (# users, # movies, # features)

c) (# features, #movies, #users)

d) (# movies, # features, # users)





Which TensorFlow operation could we use to mask the previously rated movies in our user-movie ranking matrix, so we only focus on previously unrated movies when providing recommendations?

a) tf.mask

b) tf.strided_slice

c) tf.tile

d) tf.sparse_slice

e) tf.where

- Content based recommender aren't good at expanding the interest of users to new domains.