**Title: Personalised recommendation for Scratch.**

What are we recommending?

* Projects to follow
* Users to follow
* Studios and Galleries to check

Data available - ?

Tasks -

* Selecting neighbourhood

Donot use too many neighbours

Low similarity neighbours can introduce noise to the data

Ex. 100 most similar neighbours

* Scoring Items from neighbourhood

Scoring can be **Average,** \***Weighted average** or a **Multiple linear regression**

* Normalizing Data

Compensating for different people giving different ratings.

Done by subtracting the average rating for each user in their ratings.

Or by converting to z-score

OR subtract item or item-user mean

* Computing similarities
  + Algorithms

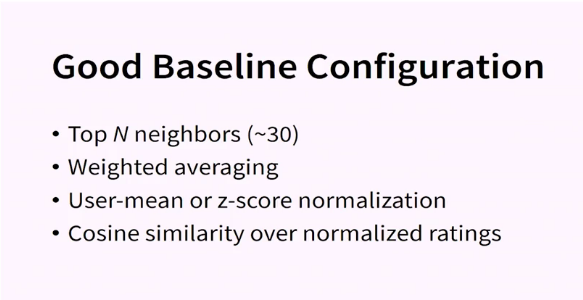
. Pearson correlation

. Spearman rank correlation

Weighting similarities.

* + Tweeks
* Good baseline configuration

For starting out do follows



Algorithms -

* Non Personalised Summary Statics ( External community data - Featured - Most liked -people following most these is already present on Scratch front page.)
* Content based Filtering
  + Information
  + Knowledge based
* Collaborative Filtering
  + User-user
  + Item-Item
  + Dimensionality reduction
* Others
  + Critique/Interview based recommendation
  + Hybrid techniques

Evaluation -

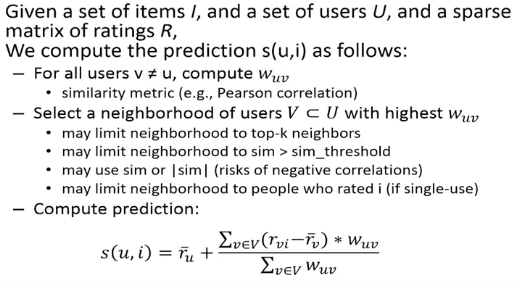
* Accuracy of prediction
* Usefulness of recommendation
  + Correctness
  + Non-obviousness
  + Diversity
* Computational Performance

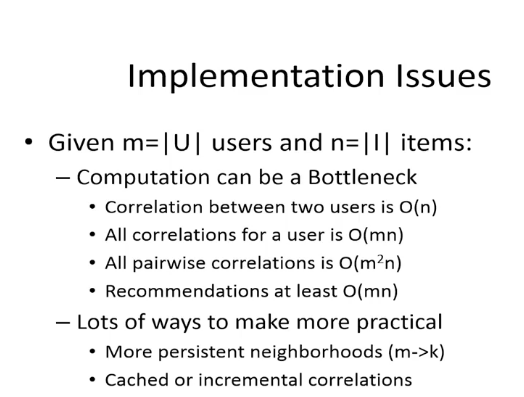
Recommendation questions idea -

1. What did I remix
2. Who I follow
3. What are my friends commenting on.
4. What did my friends recently remix

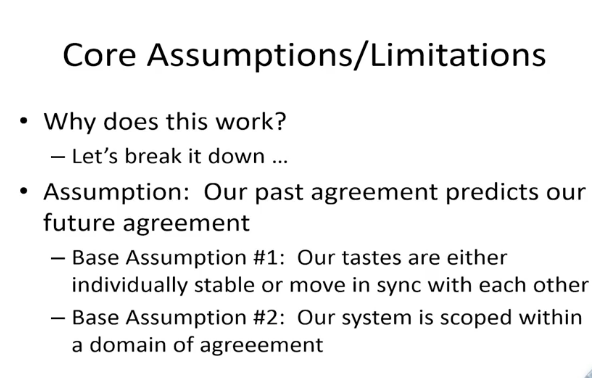
\*Scoring Function

\*Ordering function





To predict how much a target user will like an target item, the system first generates a neighborhood of other users who have agreed with that user on other items in the past. Then the system computes a weighted, normalized average of what those other users rated the target item.



Look at: Association rule mining

Future work may involve :

* Using machine learning
* Text mining on comments and code to find context and interest.
* TFIDF on tag data to identify context / content-based filtering

First, we identify the parameters in the

Identify the similarity between the projects based on the parameters.

Suggest **remixes** of the project

If they have the same origin

The same number of **block/block.id** suggest similar complexity of the project

Todo:

Assign weights to parameters

Recommend projects remixes

Calculate score +

Assign weights based on the correlation outputs( see graphs)

Limitations: the dataset was scraped on older version of scratch. sin

Implement Similarity matrix for projects liked by the user

what are you doing

context you understand the surrounding context

how you sat abot trying to answer the question - approch

technical work

what is the answer

Challenges: improve rating method

==============================================================

How are you assessing those, results

Evaluation of the quality of the recommendation.

In the because you can run this with real user, how are you evaluating how effective the recommender system is

How this could be done :

A/B testing.

Qualitative testing

Gather data through user surveys

Validated the results in the space of technical work.

You build the recommender in certen way

How could you know that that's any good? ---- You cant?

Design-build - confirmed -

If this project was six weeks longer the next thing i would argue would be did I get it right thats not something you can evaluate mathamatically, excatly.

Next step would be to actually use it with people in some ways there is lots of ways you could do it. You said some reasons why we would want a reocommneder.Are this this recommendations any good for any of those.

Making it clear,

You r design makes assumptions what people want in the recommendation

This is my second question,

I bought a toilet seat on amazon and now amazon ….

face book and hate speach.

The question for me is your recommender system is based on ,

You need a good justification for that. I am only

There are other ways of recommending.

Justifying why you build the way that you did it.

TODO:

Chap 4