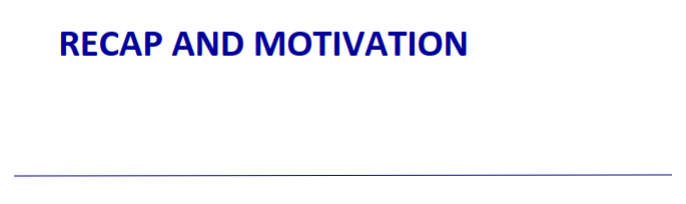
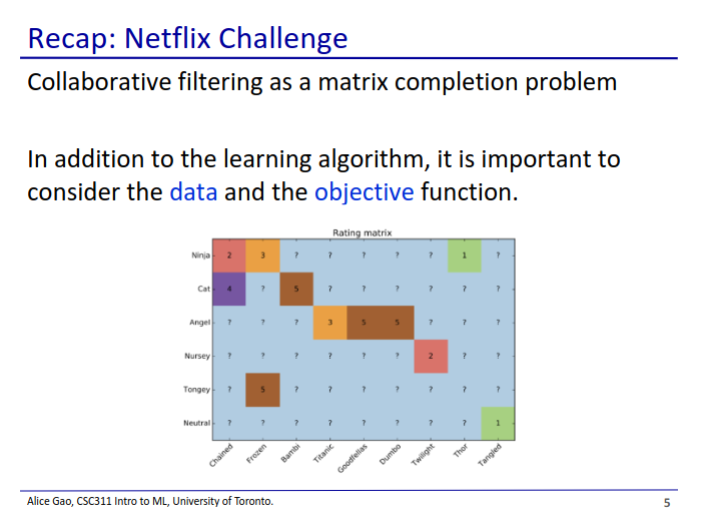
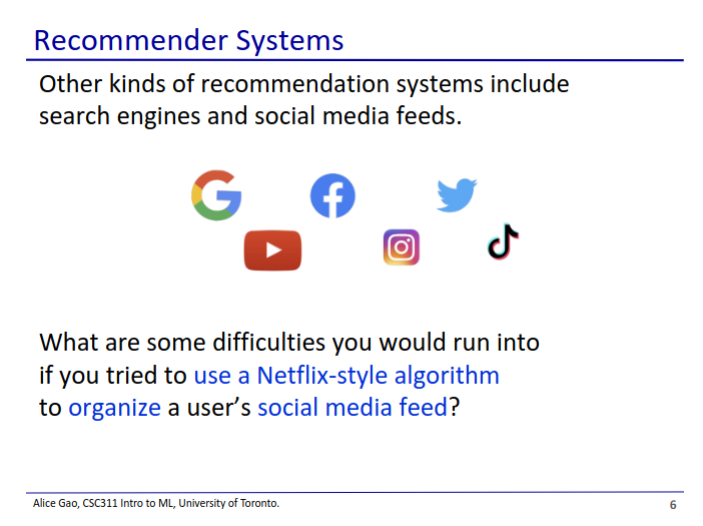


* For this week’s lectures, think more about the questions, there are no singular answers

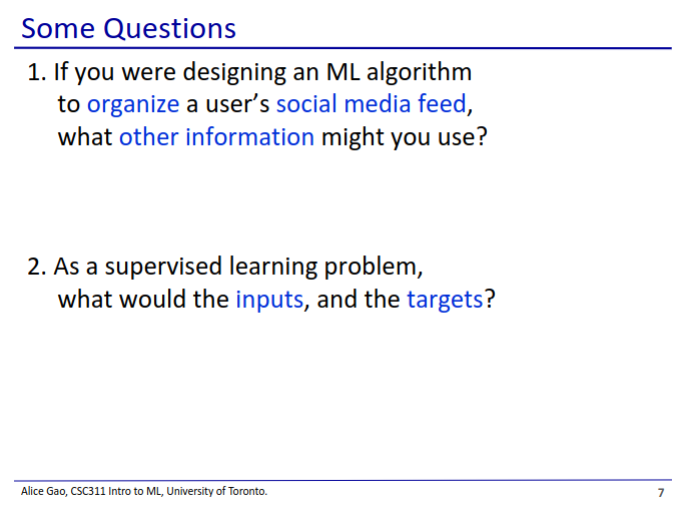




* Netflix challenge
  + Uses an algorithm to complete the matrix
* In this lecture we will consider other questions
  + How do we even define the problem?
  + What objectives are we trying to optimise? What may be the consequences of this?



* Netflix challenge is actually a very simple recommender system
* We interact with more complicated recommender systems every day
  + Search engines give an ordered list of results
  + Social media gives us an ordered list of posts
* Our solution for the netflix solution may not work for these problems



1. What information might you use?

* User’s friends
* User’s liked posts
* User’s demographics

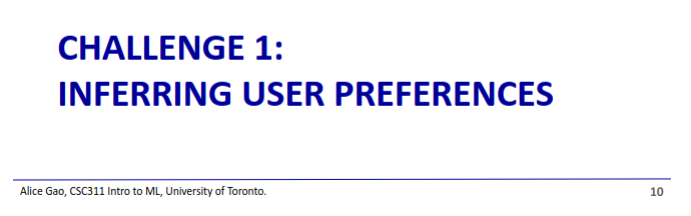
1. What would be the inputs and the targets?

* Inputs: all sorts of user and post data to generate the feed
* Output: generating an ordered list of posts



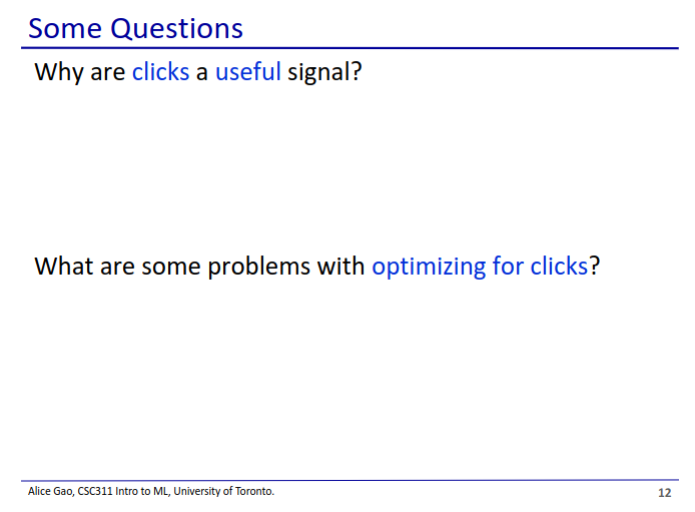


* The recommender algorithm has learned based on your activity which post you may like
* You click on a topic once and now its all that you see

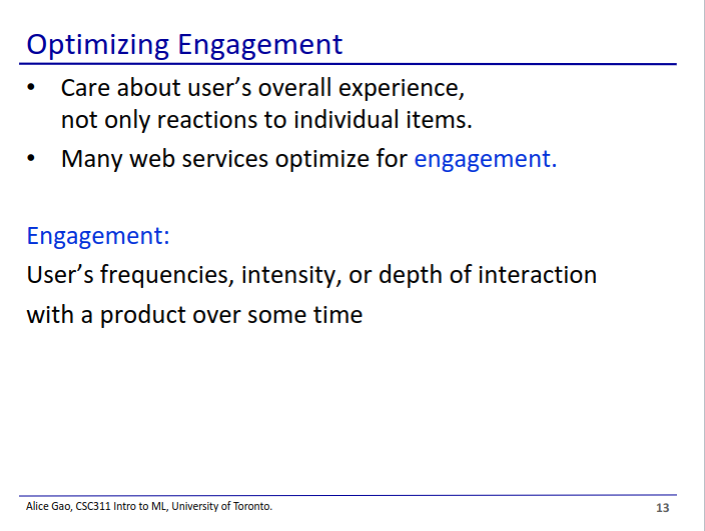


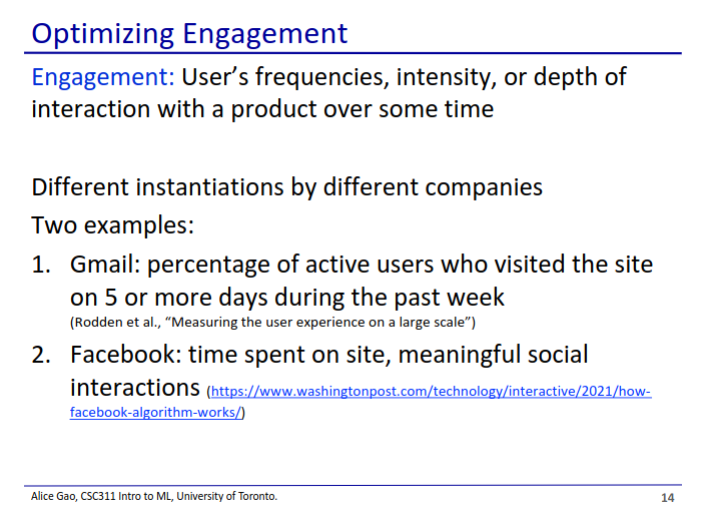


* Google news
  + A news feed service
  + Used an algorithm to try to predict which articles get clicked

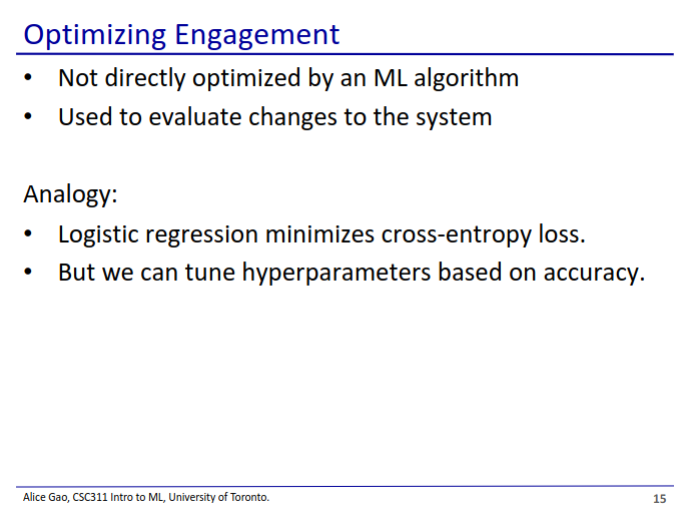


* Why are clicks a useful signal?
  + If a user clicks on an article, it is likely that they are engaged with the article
* What are some problems with optimising for clicks?
  + You might accidentally optimise for clickbait or brain-numbing content
  + A user clicking might not indicate they are actually interested in the item
  + An article may have an interesting title but may otherwise be uninteresting inside



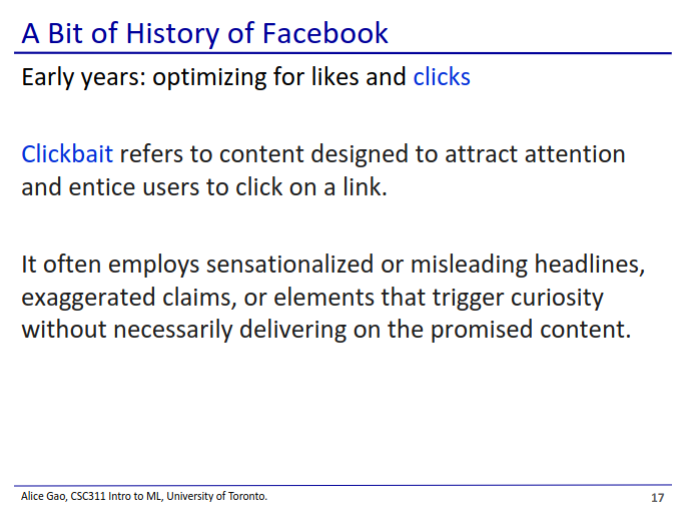


* Most companies will optimise for engagement, not just clicks
* How exactly they measure engagement depends on the company

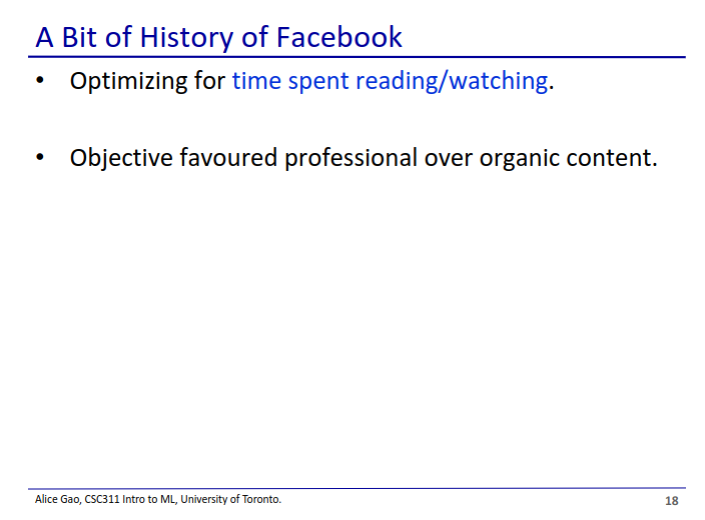


* User engagement is not used to optimise the model, but it is used to evaluate how the model changes the system
* Analogy
  + Logistic regression wants to minimise cross-entropy loss
  + We tune hyperparameters based on prediction accuracy, which is different than loss

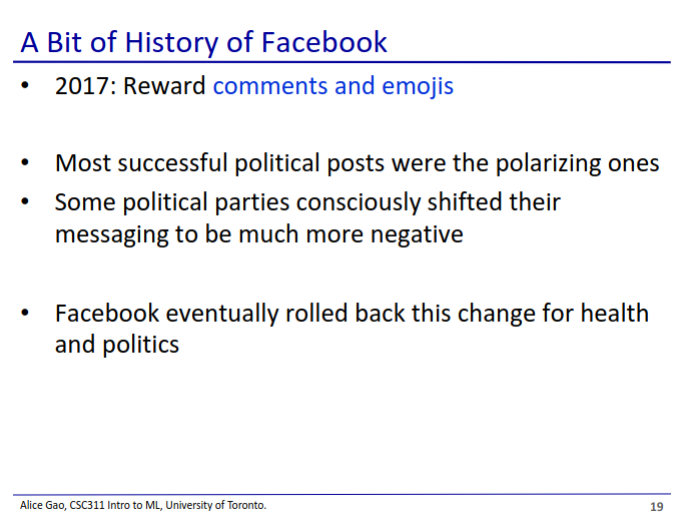




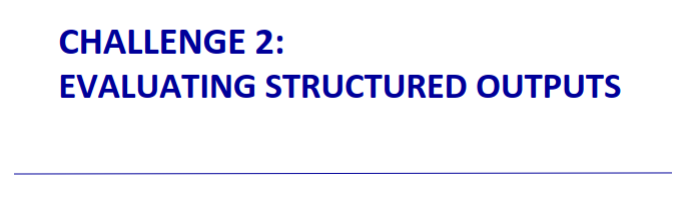
* Optimising for likes and clicks resulted in clickbait articles being promoted
* Clickbait - sensationalist titles meant to attract clicks to otherwise boring content

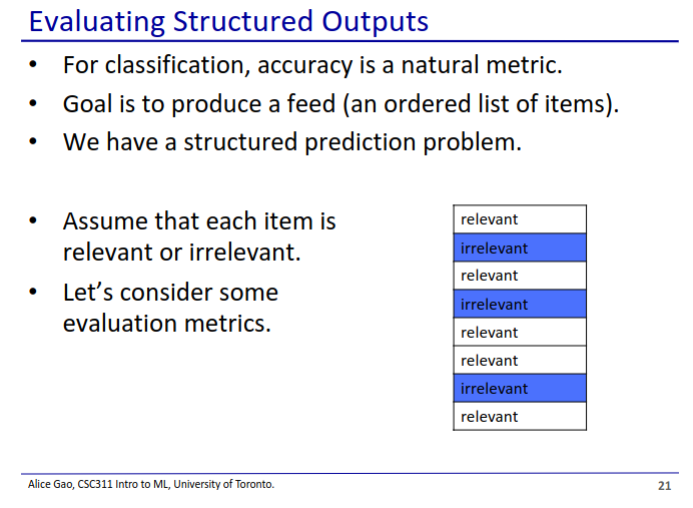


* Then they optimised for time spent reading/watching
* This metric favoured professional content
  + Professional content is more polished and can retain viewers for longer

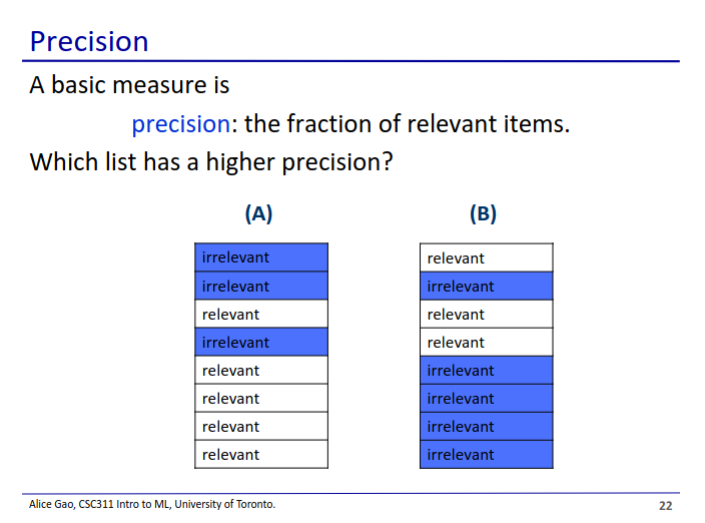


* The most effective emotion for encouraging comments is anger
  + Successful posts are now more polarising and elicit strong emotions

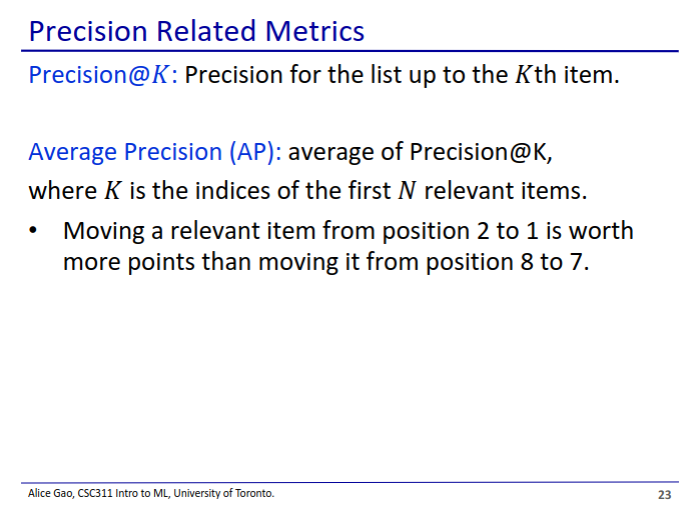




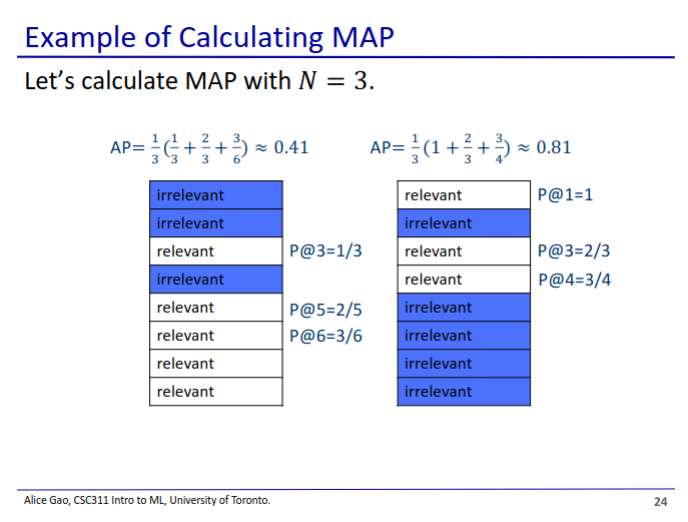
* Now that we have a social media feed, how do we evaluate how good it is?
* In this case, our output is an ordered list of items, which makes evaluating the output more complicated
  + This is a structured prediction problem - output is structured



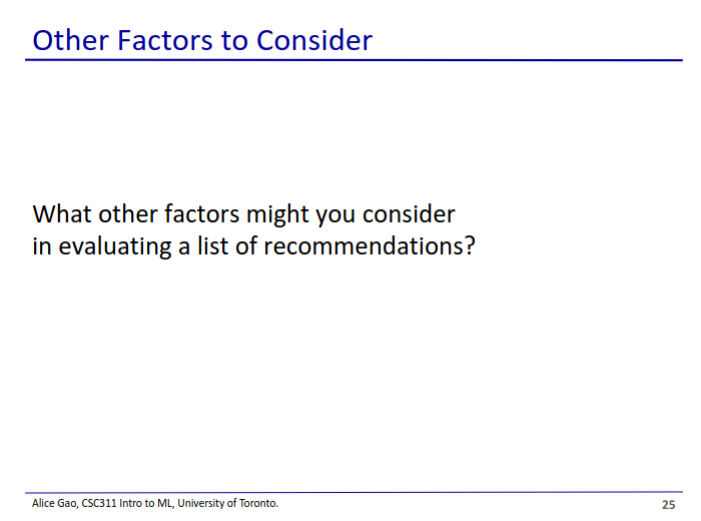
* A has a higher precision than B
  + A has more relevant results than B
* Precision is not necessarily a good measure of how good a feed is
  + A has worse results at the top of the feed, which may be undesirable



* Let’s refine our precision metric a little bit
  + Precision@K
    - Fraction of the list that’s relevant up to the K’th item
* Average precision
  + We can try many different cutoffs and take the average precision



* Each time we find a relevant item, we calculate the precision
* Note: typo in the top-left equation, should be ⅖ not ⅔



* User interaction with post
* How long a user stays on the feed



