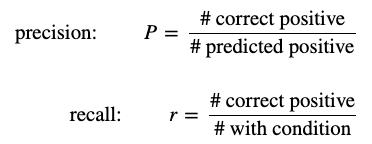
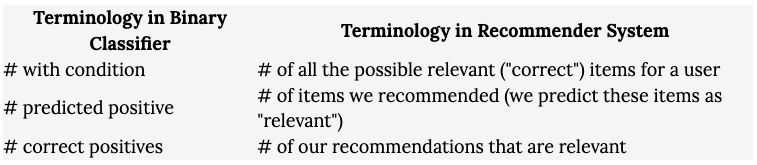
November 17, 2021 (Ross)

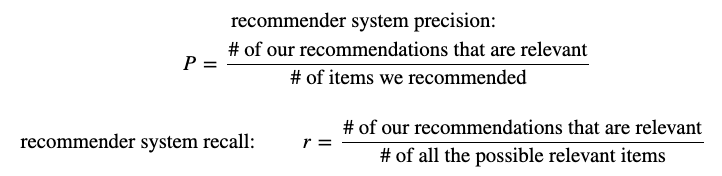
* We need a metric (or multiple) to evaluate how our recommendation engine is performing. I do not think that we can use accuracy because accuracy (in a ML sense) describes how many labels you predicted to be the true label.
* We do not have true labels for our recommendations (or at least not that I am aware of), so there is no way to compare the generated recommendations to our engine’s recommendations.
* I did some reading and there are other metrics that fit with recommendation systems, such as:
  + Mean average recall at K (MAR@K)
  + Coverage
  + Personnalization
  + Intra-list similarity
* Here is one link to an overview of the metrics above
  + <https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093>
* Another blog post (<http://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html>) goes in depth about mean average precision (MAP). Here are some notes from that blog post:
  + MAP implies that you are treating the recommendation like a ranking task (which movies will the user like the most)
  + MAP will show the top recommendations first. This should fit with both our content, user, and hybrid approaches
    - The content based approach ranks recommendations by cosine similarity, so higher cosine similarities = better recommendation, at least based on ratings only
* Precision and recall are very important in machine learning because accuracy can be skewed if the dataset is unbalanced (more samples belonging to class 1 than to class 0)



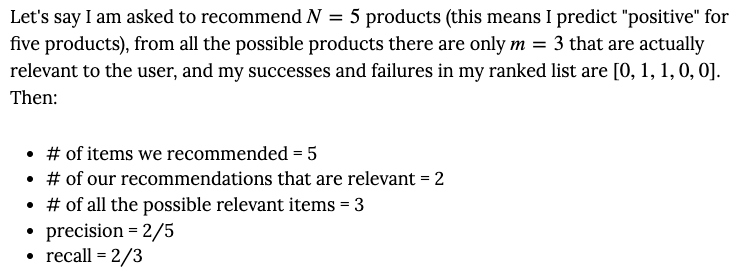
* Precision and recall are similar in recommendation systems:



* And then precision and recall are calculated as:



* Here is an example, which I am still a bit confused on. I think that you would have to pick an arbitrary value for determining what a relevant recommendation is



November 18, 2021 (Ross):

* I tried to loop over multiple values of K in the KNN clustering based engine, but I found that the number of neighbors doesn’t necessarily matter because the nearest neighbors will always be the best recommendation, second nearest will be second recommendation, etc. So from my understanding, altering the number of neighbors won’t change the recommendation. This is different then a more ML based approach where you try to guess the correct label of the data point rather than making a recommendation.