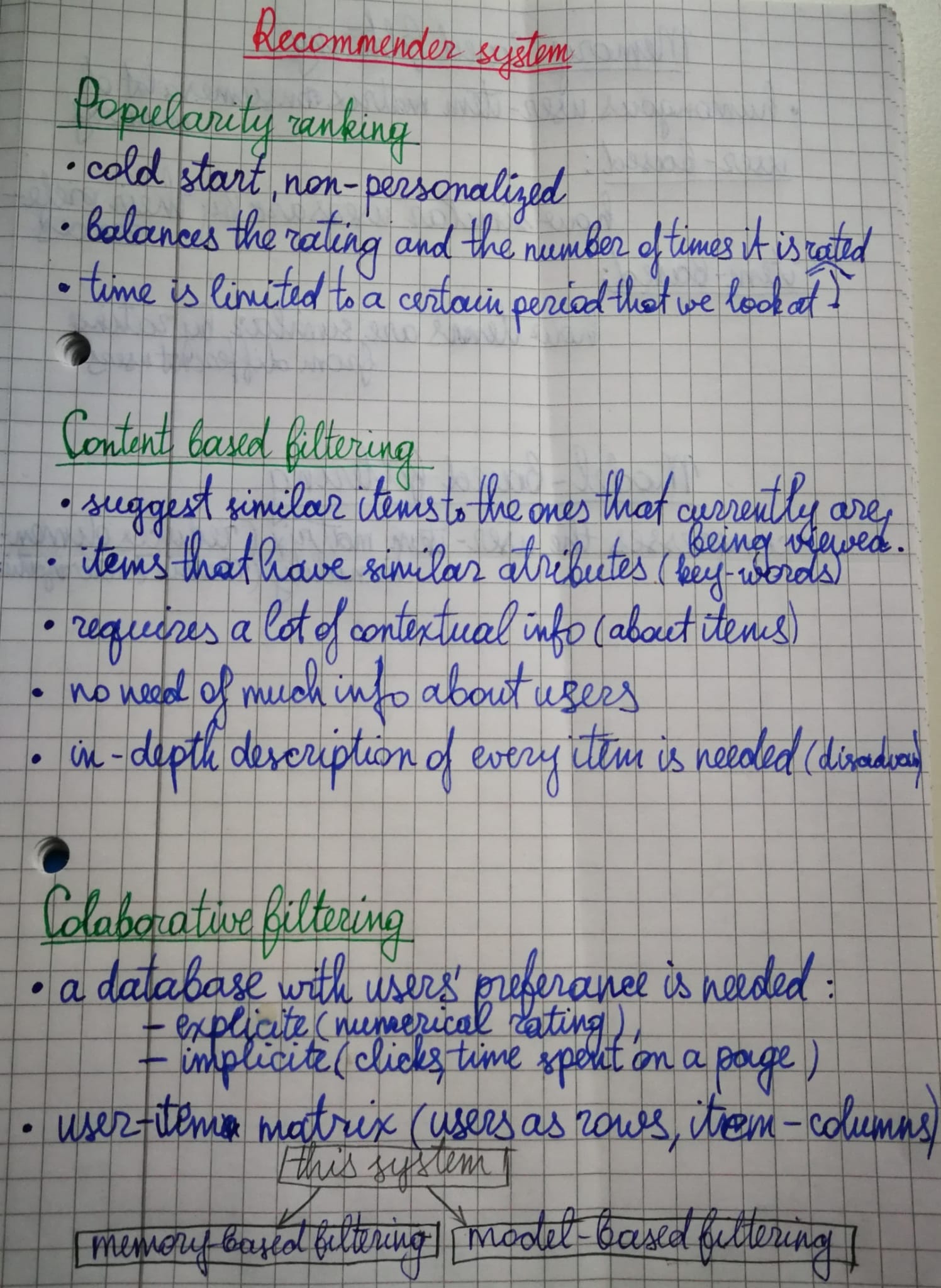
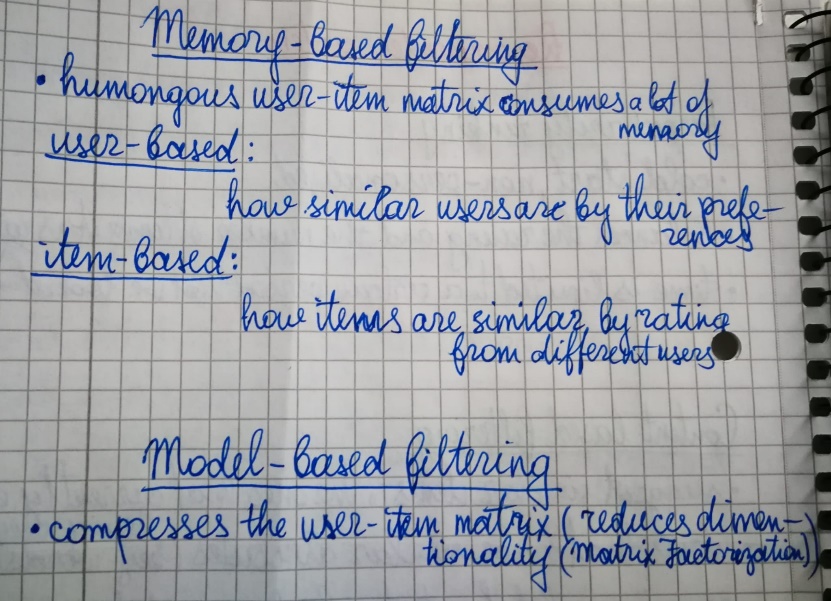
**Recommender/Filtering Systems**



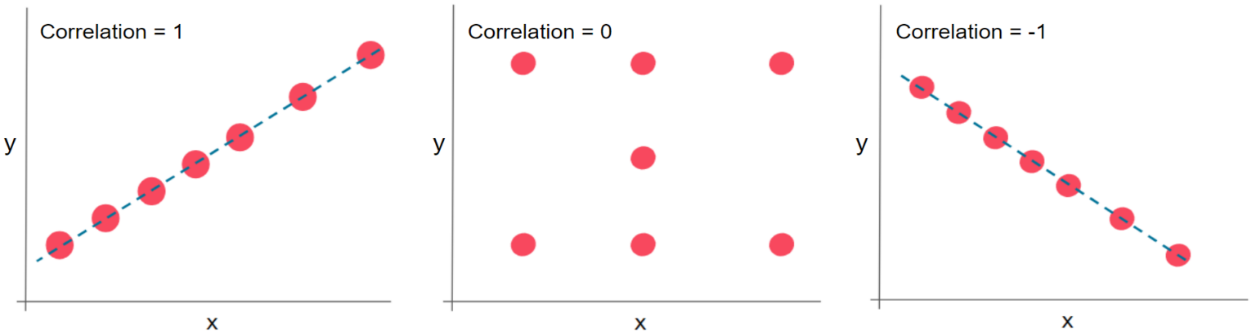
* **Item-based approach**: We recommend movies similar to a movie that the user rated highly recently (e.g. Pocahontas)
* **User-based approach**: We find other users similar to the user we’re recommending movies to. Then we recommend the movies those other users rated highly.

There are three main methods of calculating similarities in recommender systems: Jaccard(unary or binary data), Pearson, and Cosine(quantitative data).

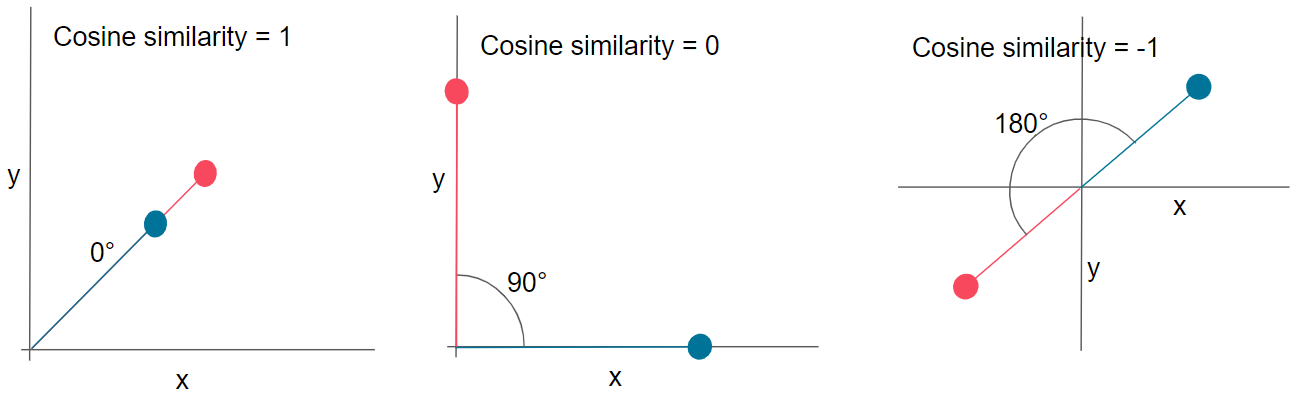
Pearson’s correlation

Pearson’s correlation coefficient tells us how strong the linear relationship between two variables is. The scores can range from -1 to 1.

There is a strong positive linear relationship between the two variables. As X increases Y increases.



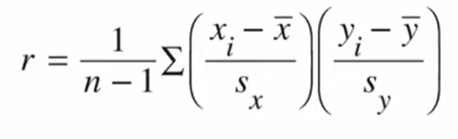
## Cosine similarity

The cosine similarity judges similarity based on the direction of vectors.

The **cosine\_similarity** function outputs the pairwise similarity for all samples in the index of the DataFrame. Since the function operates row-wise and not column-wise, we will transposed DataFrame (using .T) to compute the similarities between columns:

**from sklearn.metrics.pairwise import cosine\_similarity**

**cosine\_similarity(df.T)**

Calculating correlation coefficient r

r – shows to which extent a line can describe a relationship between X and Y, it is always >= -1 and <=1.

n- number of pairs to compare,

x**i** – every single value on an X axis,

x̄ – mean of x(sum of all points on the X axis divided by the number of all points on the X axis)

sx – standard deviation of x points

yi – every single value on an Y axis,

ȳ – mean of y

sy – standard deviation of y points