# Project-1 Report

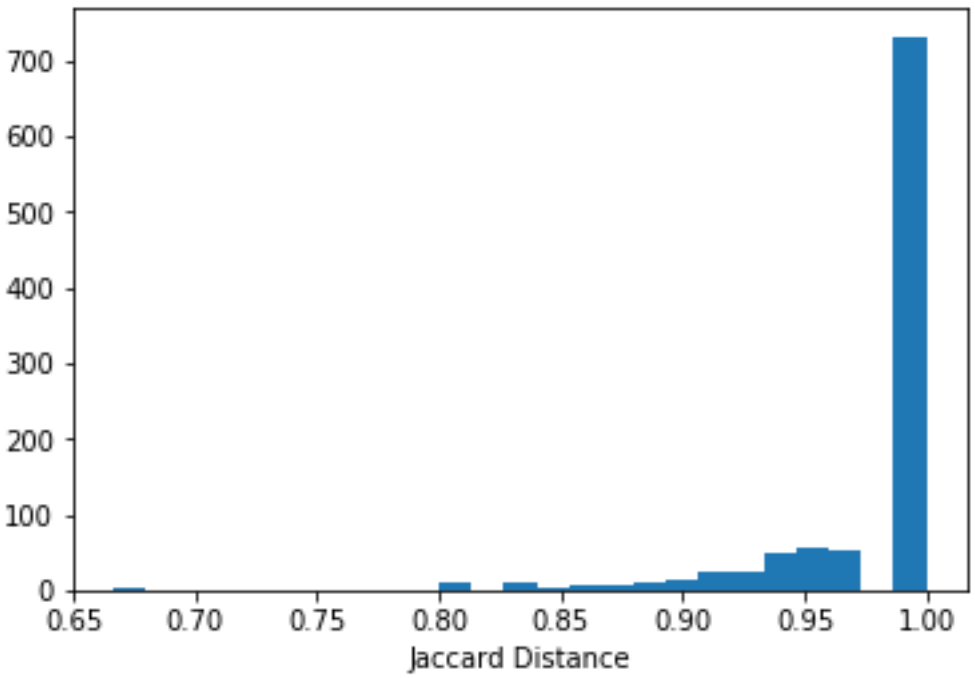
Aditya Kashyap

Karishma Jain

## Problem 1 and 3:

The M x N matrix that we get has 4499 unique movies (M) and 231424 unique users (N) who have rated between 1 and 20 movies with a rating greater than or equal to 3. This matrix is very sparse and storing this as an array not only increases the computational burden but also uses a significant chunk of the memory. Having recognized that, we created a sparse matrix which helps us store only the nonzero elements of the matrix along with their indices, thereby significantly reducing the computation time by eliminating operations on zero-valued elements.

## Problem 2:




Looking at the histogram, a major fraction of the 10,000 pairs that we selected at random are dissimilar, with a Jaccard Distance of ~1. The probability of any two random users having rated the same movies (out of 4499 movies) with rating 3 and above is very low. Hence, there are very few pairs with a Jaccard similarity of 0.65.

Each column vector represents a list of movie ratings of a particular user. Therefore, row indices of a column where the rating is equal to 1 will correspond to the movies rated higher than 3 for a particular user, which in turn will tell us the movies liked by the user.

Average Jaccard Distance = 0.98137

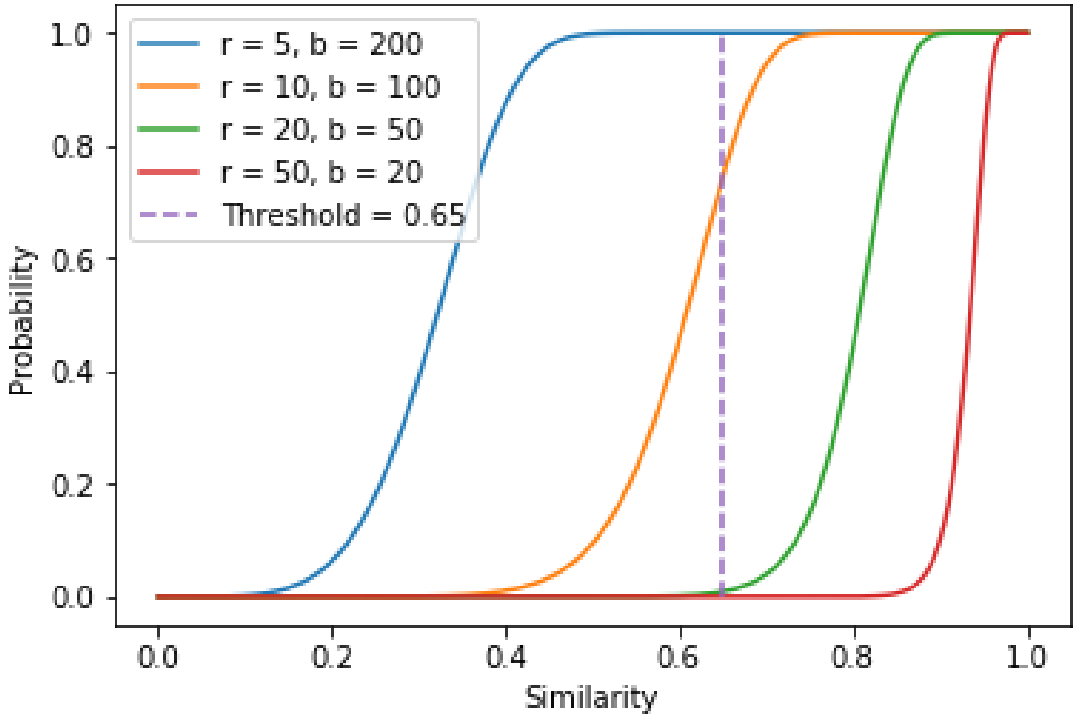
Minimum Jaccard Distance = 0.65

## Problem 4:

1000 hash functions were chosen to create the signature matrix. For dividing the signature matrix into b bands of r rows each, the value of r was decided as follows:

According to the problem definition, two users are close if their Jaccard distance is below 0.35. Therefore, the threshold (value of Jaccard similarity) is 0.65. An approximation to the threshold is given by the formula. $\left(\frac{1}{b}\right)^{\frac{1}{r}}$

We plotted the similarity vs probability curve for different values of r and b ( = 1000/r )




Since avoiding false negatives is important, we select b and r such that they produce a threshold value lower than 0.65.

We observe that for r=10, we get the value of $\left(\frac{1}{b}\right)^{\frac{1}{r}}$ as 0.63, which is close to 0.65.

So, we chose the value of r as 10 and b= 1000/10 =100.

We got 1,957,219 pairs that had Jaccard Similarity greater than 0.65 on a particular trial run. It ran within 8 minutes, once optimized in terms of both storage complexity and time complexity

## Problem 5:

A function was written that took in a user array of dimensions (4499,1) and returned the approximate nearest neighbor.