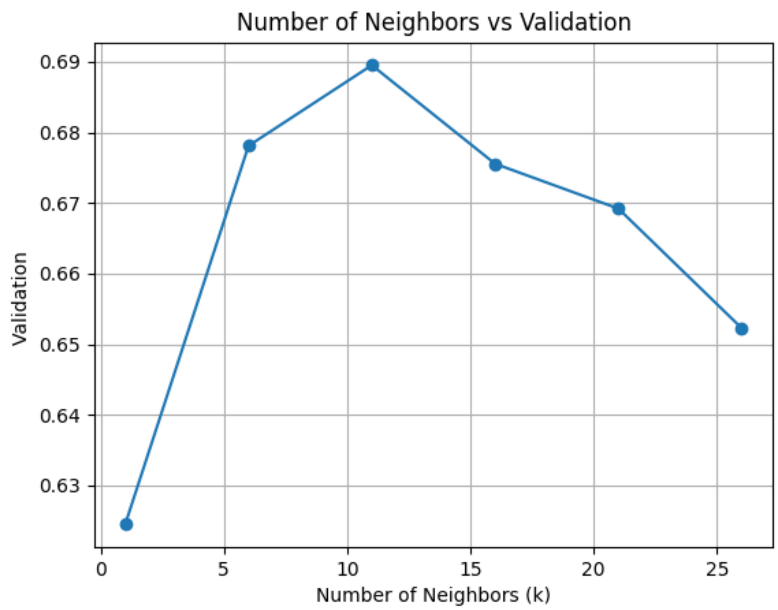
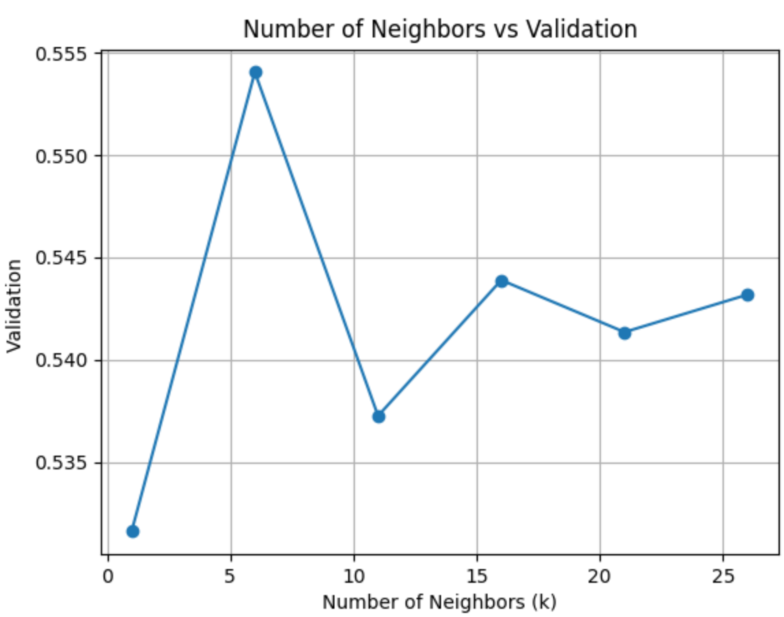
1. a.)
   Test Accuracy: 0.6841659610499576
   k: 11



   b.)
   The underlying assumption is that if other students consistently assign the same scores to both Question A and Question B, then both questions are considered equally correct or incorrect for those particular students.

   c.)
   Test Accuracy: 0.545582839401637
   k: 6



d.)
Item Based Filtering seems better for our use case.

Since user-based filtering is dependent on a students history, we need a student to answer a number of questions before we have reliable data, while item-based filtering just relies on other items.

A student may become better overtime, in which case, user-based filtering fails to consider a students improvement when training. On the other hand, questions don't change in difficulty overtime. We want to recommend harder questions if a student gets better and not penalize them for past incorrectness.

e.)
The effectiveness of KNN diminishes notably when dealing with sparse data. This is particularly evident in scenarios where the training set contains numerous missing values that require imputation. The presence of extensive empty spaces leads to limited overlap between data points, causing the nearest neighbors used for prediction to potentially be distant from the target point. For example,in the user-based filtering model, given we are trying to predict an answer for a student, if there are no students with closely matching diagnostic answers, the nearest neighbors may not accurately capture the local data structure.

KNN can be slow due to the need to compute distances between each and every data point, especially in this scenario, since we expect to have sparse matrices and a large number of missing values. Since we want to scale our application to get data from more students and increase the number of questions students can do, the large data set will increase computational cost.

2. a.)

We have,

$$p(C_{ij}=0|\theta_i,\beta_j)=1-\frac{exp(\theta_i-\beta_j)}{1+exp(\theta_i-\beta_j)}$$

$$p(C_{ij}=1|\theta_i,\beta_j)=\frac{exp(\theta_i-\beta_j)}{1+exp(\theta_i-\beta_j)}$$

$$p(C_{ij}|\theta_i,\beta_j)=p(C_{ij}=1|\theta_i,\beta_j)^{C_{ij}}p(C_{ij}=0|\theta_i,\beta_j)^{1-C_{ij}}$$

Then we derive the likelihood function.

$$p(C|\theta,\beta)=\prod_{i=0}\prod_{j=0}p(C_{ij}|\theta_i,\beta_j)$$

Now, we can get the log likelihood.

$$log(p(C|\theta,\beta))=\sum_{i=0}\sum_{j=0}log(p(C_{ij}|\theta_i,\beta_j))$$

$$=\sum_{i=0}\sum_{j=0}C_{ij}log(\frac{exp(\theta_i-\beta_j)}{1+exp(\theta_i-\beta_j)})+(1-C_{ij})log(1-\frac{exp(\theta_i-\beta_j)}{1+exp(\theta_i-\beta_j)})$$

$$=\sum_{i=0}\sum_{j=0}C_{ij}(\theta_i-\beta_j)-log(1+exp(\theta_i-\beta_j))$$

Next, we derive the derivative of log likelihood with respect to θ_i and β_j.

$$\frac{\partial p(C|\theta_i,\beta)}{\partial\theta_{i'}}=\frac{\partial\sum_{i=0}\sum_{j=0}C_{ij}(\theta_i-\beta_j)-log(1+exp(\theta_i-\beta_j))}{\partial\theta_i}$$

$$=\frac{\partial\sum_{j=0}C_{i'j}(\theta_{i'}-\beta_j)-log(1+exp(\theta_{i'}-\beta_j))}{\partial\theta_{i'}}$$

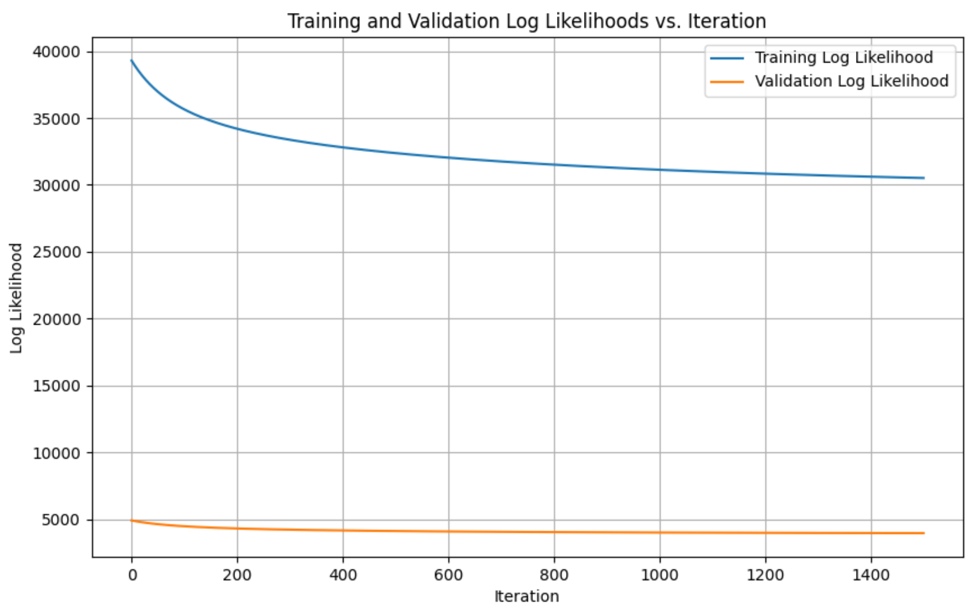
$$=\sum_{j=0}C_{i'j}-\frac{exp(\theta_{i'}-\beta_j)}{1+exp(\theta_{i'}-\beta_j)}$$

$$\frac{\partial p(C|\theta,\beta_j)}{\partial\beta_{j'}}=\frac{\partial\sum_{i=0}\sum_{j=0}C_{ij}(\theta_i-\beta_j)-log(1+exp(\theta_i-\beta_j))}{\partial\theta_i}$$

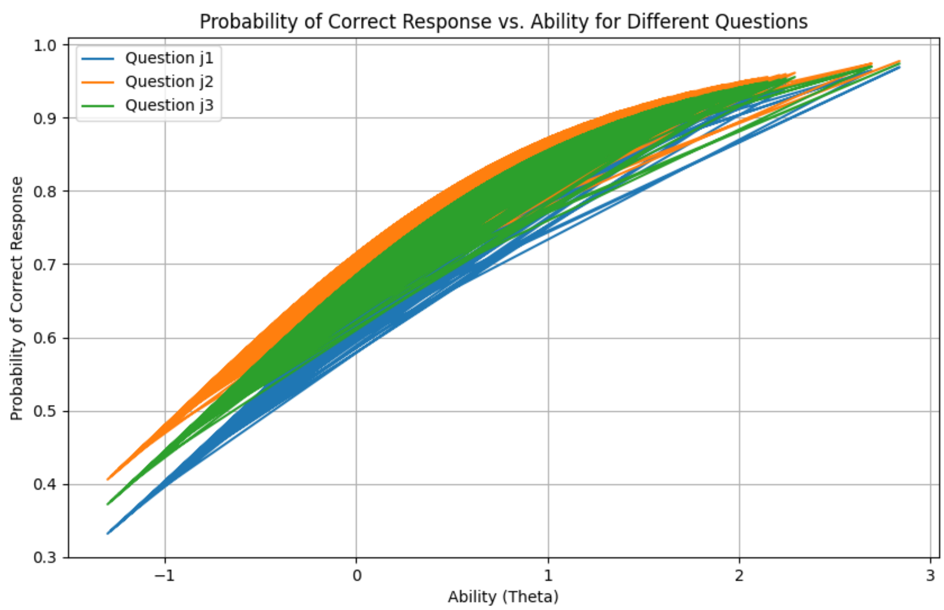
$$=\frac{\partial\sum_{i=0}C_{ij'}(\theta_i-\beta_{j'})-log(1+exp(\theta_i-\beta_{j'}))}{\partial\beta_{j'}}$$

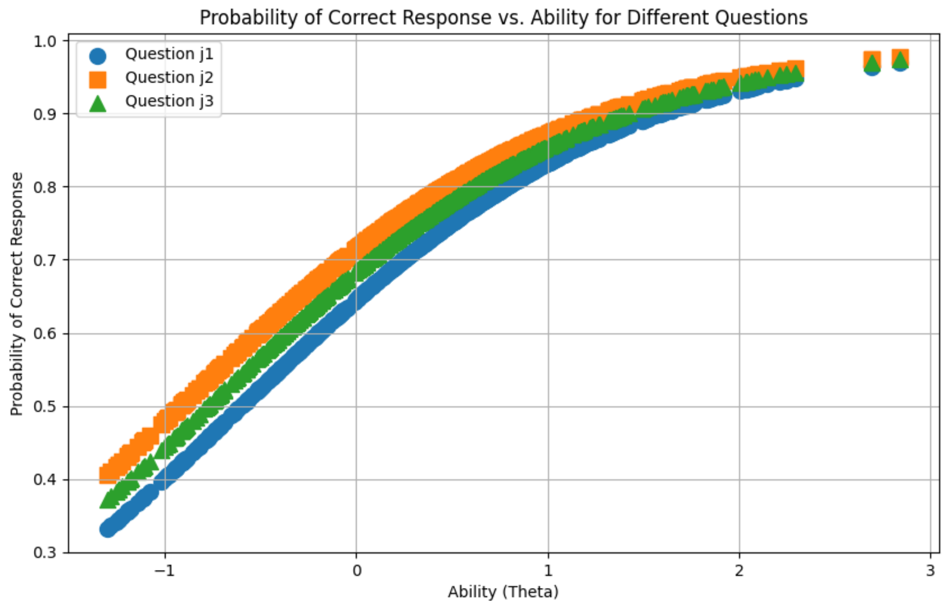
$$=\sum_{i=0}\frac{exp(\theta_i-\beta_{j'})}{1+exp(\theta_i-\beta_{j'})}-C_{ij'}$$

c.)
Learning rate: 0.0001
Iterations: 1500
Validation accuracy: 0.7075924357888794
Test accuracy: 0.7025119954840531



d.)





Questions used:
j1: 1525
j2: 1574
j3: 1030
Theta represents the ability of the student. It should be noted that as the ability of a student increases, they become more capable of solving a given question. This relationship is depicted in the graph. The curve illustrates that as Theta increases, the probability of a student solving question j1, j2, or j3 also increases. It is worth noting the separation between the curves: the orange curve has a higher probability of being solved than the green and blue curves. Therefore, we can infer that orange questions are easier than green questions, and green questions are easier than blue questions.