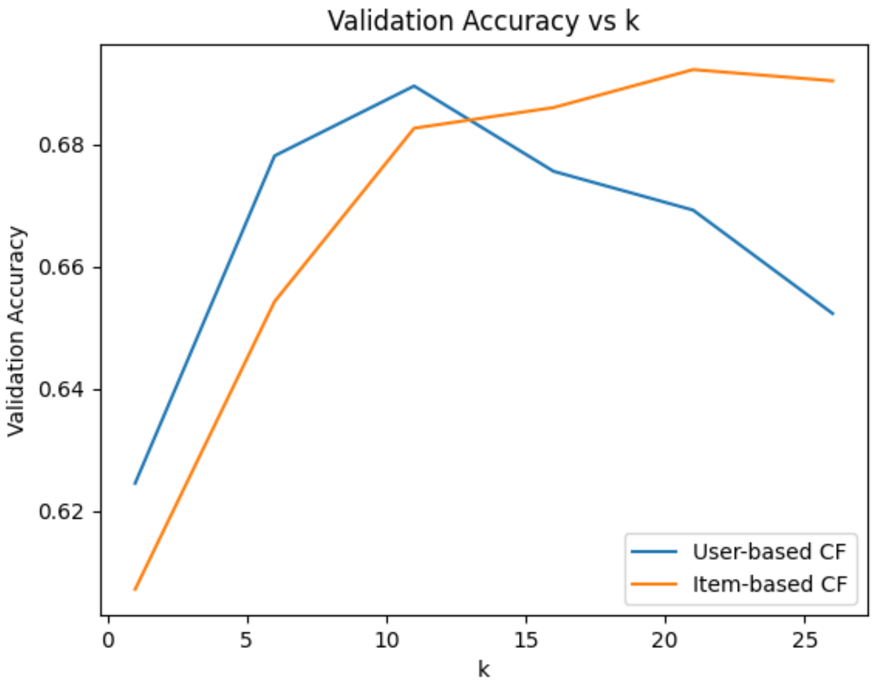
Question 1

(a) (b) (c) The accuracy on the validation data with $k \in \{1, 6, 11, 16, 21, 26\}$ on user-based and item-based collaborative filtering is as follows:



Test Accuracy on user-based CF with k* = 11: 0.6841659610499576

Test Accuracy on item-based CF with k* = 21: 0.6816257408975445

(d) The test on user-based CF is slightly better than item-based CF.

Additionally, the test accuracy on user-based CF cost less time than item-based CF.

Therefore, user-based CF is better than item-based CF in this case.

(e)
- The KNN algorithm is computational expensive for large datasets.
- The Curse of Dimensionality: In high dimensions, "most" points are approximately the same distance and the nearest neighbors are not very useful.

Question 2

(a) Given the probability that the question j is correctly answered by student i is:

$$p_{ij} = \frac{\exp(\theta_i - \beta_j)}{1 + \exp(\theta_i - \beta_j)}$$

The log-likelihood for all students is derived as follows:

$$\begin{aligned}
\log p(\mathbf{C}|\boldsymbol{\theta}, \boldsymbol{\beta}) &= \sum_{i,j} (c_{ij} \log p_{ij} + (1 - c_{ij}) \log(1 - p_{ij})) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} \left(c_{ij} \log \left(\frac{\exp(\theta_i - \beta_j)}{1 + \exp(\theta_i - \beta_j)} \right) + (1 - c_{ij}) \log \left(1 - \frac{\exp(\theta_i - \beta_j)}{1 + \exp(\theta_i - \beta_j)} \right) \right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} (c_{ij}(\theta_i - \beta_j) - \log(1 + \exp(\theta_i - \beta_j))),
\end{aligned}$$

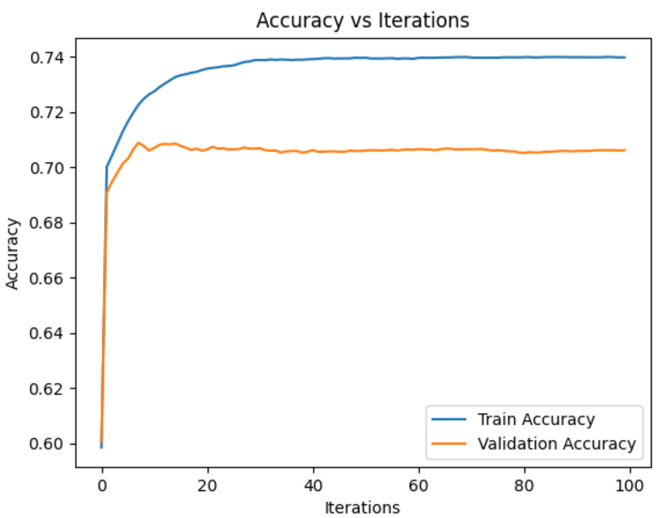
where c_{ij} is the binary response of student i to question j.
The log-likelihood with respect to θ_i is:

$$\begin{aligned}
\frac{\partial \log p(\mathbf{C}|\boldsymbol{\theta}, \boldsymbol{\beta})}{\partial \theta_i} &= \sum_{j=1}^{m} \left(c_{ij} - \frac{\exp(\theta_i - \beta_j)}{1 + \exp(\theta_i - \beta_j)} \right) \\
&= \sum_{j=1}^{m} (c_{ij} - p_{ij}).
\end{aligned}$$

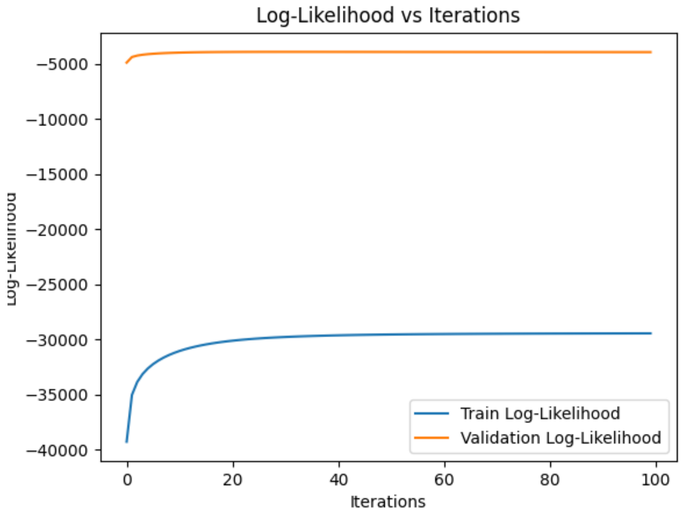
The log-likelihood with respect to β_j is:

$$\begin{aligned}
\frac{\partial \log p(\mathbf{C}|\boldsymbol{\theta}, \boldsymbol{\beta})}{\partial \beta_j} &= \sum_{i=1}^{n} \left(c_{ij} - \frac{\exp(\theta_i - \beta_j)}{1 + \exp(\theta_i - \beta_j)} \right) \\
&= \sum_{i=1}^{n} (c_{ij} - p_{ij}).
\end{aligned}$$

(b) The hyperparameters I selected are: learning rate $= 0.01$ and iterations $= 100$.
The training and validation accuracies vs iterations are in the graph below:



The log-likelihoods vs iterations are in the graph below:

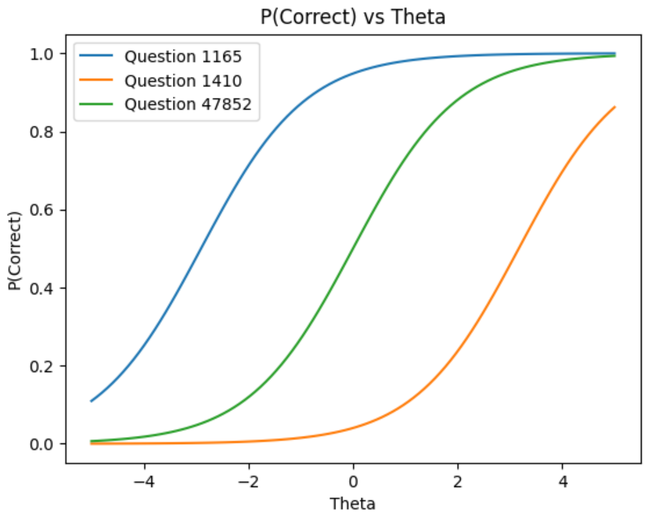


(c) The Final Validation Accuracy: 0.7063223257126728

The Final Test Accuracy: 0.707310189105278

(d) I select the lowest difficulty question j_1 (Question 1165), the highest difficulty question j_2 (Question 47852) and the average difficulty question j_3 (Question 1410).

The probability of the correct response is in the graph below:



(e) The shape of the curves are like the sigmoid function as expected.

Fix a question j. As θ_i increases, the probability of the correct response p_{ij} increases. This means if a student has a higher ability, the probability of the correct response increases.

Fix a student i. As β_j increases, the probability of the correct response p_{ij} decreases. This means if a question has a higher difficulty, the probability of the correct response decreases.

Question 3

We choose Option2

a)

1.ALS break down large matrix into lower-dimensional matrices, Neural network modeling non-linear relationship trough layers.
2.ALS is less flexible than Neural network since they are designed for matrix factorization where neural network can model non-linear relationship.
3.ALS is more computationally efficient than Neural network for sparse dataset, Neural network require significant computational resource.

b)

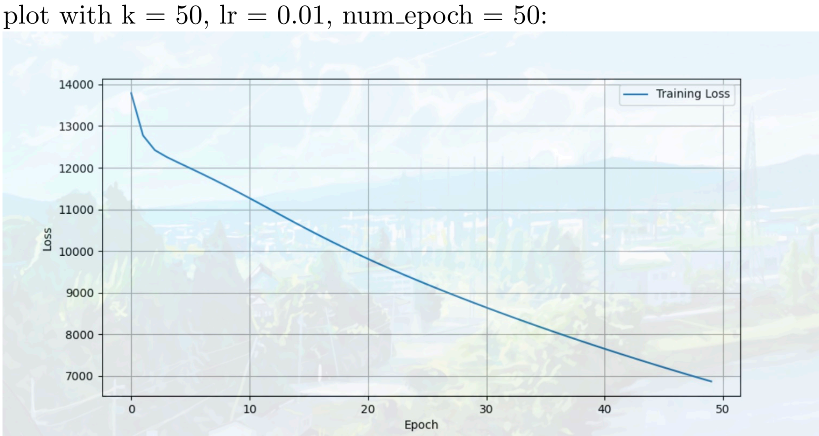
coding in neural_network.py

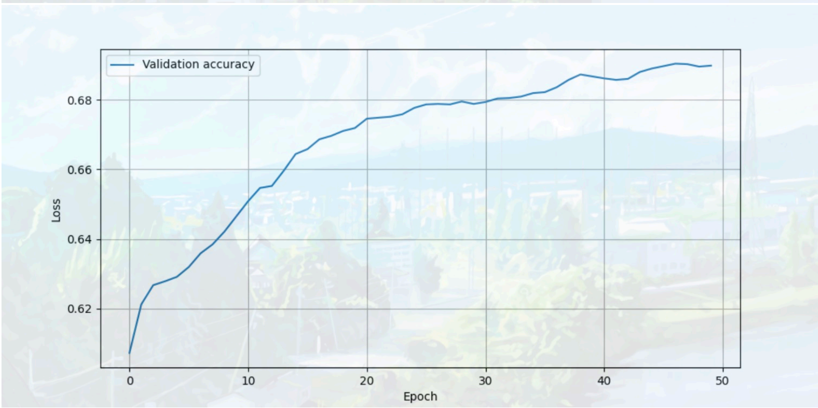
c)

the optimization hyperparameter we choose is:
k = 50, lr = 0.01, num_epoch = 50
We got Validation Accuracy of: 0.68981

d)

plot with k = 50, lr = 0.01, num_epoch = 50:





The Final Test Accuracy is: 0.68558

e)

the best regularization penalty is lamb = 0.01, with this lamb, we got:
Final Validation Accuracy: 0.67824
Final Test Accuracy: 0.68078
The model didn't perform better with the regularization penalty, this may because that our model already well-regularized and does not overfitting or only has negligible overfitting issues.

Question 4

The final validation accuracy is: 0.66286
The final test accuracy is: 0.66949

Ensemble process:
we use three neural network models to implemented bagging ensemble. We first randomly sample three sample with replacement from out training data set. Then we train three different neural network independently for each training sample. These three neural network are complete independent and can run individually. After all models are trained, we use them to make prediction separately, finally we take the average of each of their predictions as our final prediction.

Better or Not:
No, the bagging model is nearly the same performance as the single neural network model, so it doesn't improve the performance.

Reason:
Ensembling the same model which train on different data subset has lack model diversity, thus it does not always improve the model performance. Also small training subset could be another problem, when the training set is small, there could be a issue that training subset are even smaller so that each model are not well trained, which result in poor performance.