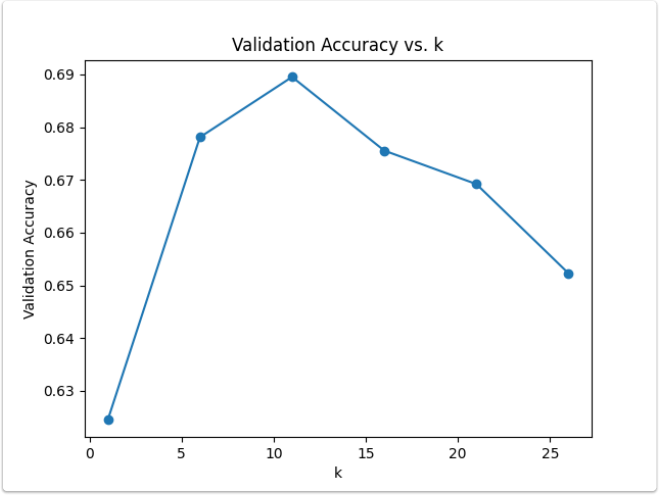
Option 1

Part A

Question 1

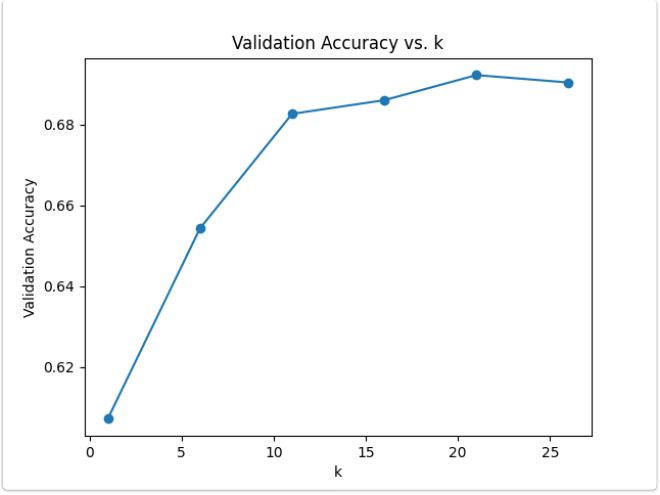
(a)



(b)

```
Validation Accuracy: 0.6244707874682472
Validation Accuracy: 0.6780976573525261
Validation Accuracy: 0.6895286480383855
Validation Accuracy: 0.6755574372001129
Validation Accuracy: 0.6692068868190799
Validation Accuracy: 0.6522720858029918
Validation accuracy with k* = 11: 0.6895
Validation Accuracy: 0.6841659610499576
Test accuracy with k* = 11: 0.6842
```

(c)




```
Validation Accuracy: 0.607112616426757
Validation Accuracy: 0.6542478125882021
Validation Accuracy: 0.6826136042901496
Validation Accuracy: 0.6860005644933672
Validation Accuracy: 0.6922099915325995
Validation Accuracy: 0.69037538808919
Validation accuracy with k* = 21: 0.6922
Validation Accuracy: 0.6816257408975445
Test accuracy with k* = 21: 0.6816
```

(d)

The highest validation accuracy is using the user-based collaborative filtering. To be precise, a k-value of 21 corresponds to a 68.16% test accuracy for a item-based collaborative filter; whereas in a user-based collaborative filter, the highest validation accuracy is with a k-value of 11 and a 68.42% test accuracy. This difference in accuracy is negligble; however, strictly speaking, user-based collaborative filtering performs better.

(e)

kNN suffers from being computationally expensive depending on the size of the data set. In this case, as the number of students and questions increases, the time and memory needed for imputation can drastically rise.
kNN also suffers from bad performance with sparse data. Since there are so many inputs of students and questions, there might not be enough information gathered to draw accurate conclusions on certain responses. As such, kNN will create biased or indecisive conclusions.

Question 2

(a)

The likelihood of observing c_{ij} is:

$$p(c_{ij} = 1|\theta_i, \beta_j) = \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}$$

Since c_{ij} is equal to either 0 or 1, each student-item response can be treated as a single Bernoulli trial.

$$p(c_{ij}|\theta_i, \beta_j) = (\frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}})^{c_{ij}}(1 - \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}})^{(1-c_{ij})}$$

Then, the log-likelihood for a single response c_{ij} for a student θ_i and question β_i is:

$$\log P(c_{ij}|\theta_i, \beta_j) = c_{ij} \log \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}} + (1 - c_{ij}) \log \left(1 - \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}\right)$$

The log-likelihood of observing response C for any student θ and any question β is the sum of the log-likelihoods of each individual response c_{ij}:

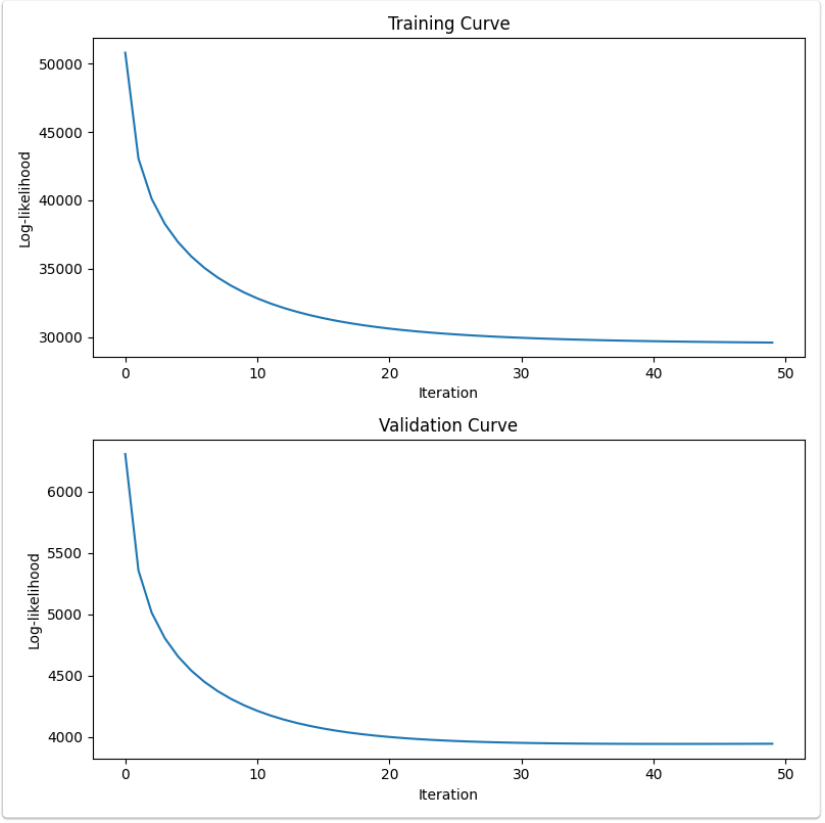
$$\begin{aligned}\log P(C|\theta, \beta) &= \sum_{i=1}^{I} \sum_{j=1}^{J} \left[c_{ij} \log \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}} + (1 - c_{ij}) \log \left(1 - \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}}\right)\right] \\ &= \sum_{i=1}^{I} \sum_{j=1}^{J} \left[c_{ij}(\theta_i - \beta_j) - \log(1 + e^{\theta_i - \beta_j})\right]\end{aligned}$$

Then, derive with respect to θ_i and β_j.

$$\frac{\partial}{\partial \theta_i} \log P(C|\theta, \beta) = \sum_{j=1}^{J} (c_{ij} - \sigma(\theta_i - \beta_j))$$

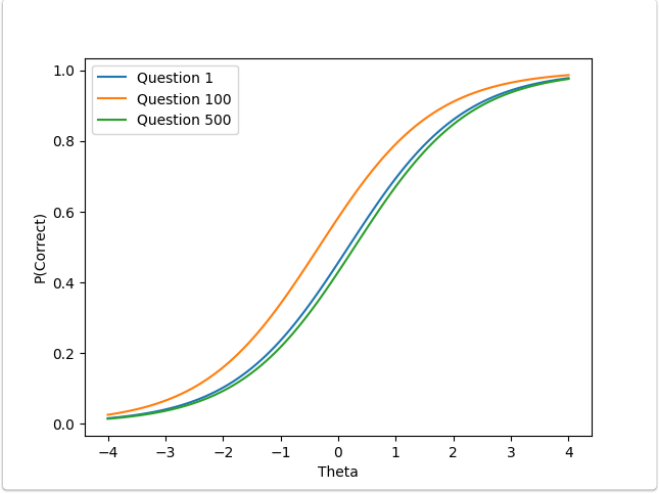
$$\frac{\partial}{\partial \beta_j} \log P(C|\theta, \beta) = \sum_{i=1}^{I} (-c_{ij} + \sigma(\theta_i - \beta_j))$$

(b), (c)




```
Best validation accuracy: 70.60%
Best iteration value: 100
Best learning rate: 0.01
Test accuracy: 70.76%
```

(d)



The curves take on the shape of the sigmoid function. They represent the probability of a student answering any of the three chosen questions correctly as a function of their own abilities (which is our Theta value).

Question 3

(a)

Training method: ALS uses iterative optimization algorithm that to estimate the original matrix by updating two lower dimensional matrices in alternating order. Neural network algorithm implements backpropogation and SGD to update weights based on the gradient of the loss function.
Optimization: ALS focuses on minimizing the difference between the original matrix and the two lower dimensional matrices. Neural networks focuses on minimizing the loss between actual outputs and algorithm-predicted outputs.
Computational and Memory cost: ALS has relatively small train time, computational time, and less memory. This is because ALS requires two lower dimensional matrices, and computations regarding two matrices can be done relatively easily. Neural networks require lots of memory, since there are usually many hidden layers (resulting in exponentially more weights needed). The computational time for neural networks correlates to the number of hidden layers and have a forward pass through the network (meaning computations for activation function of all layers).

(c)

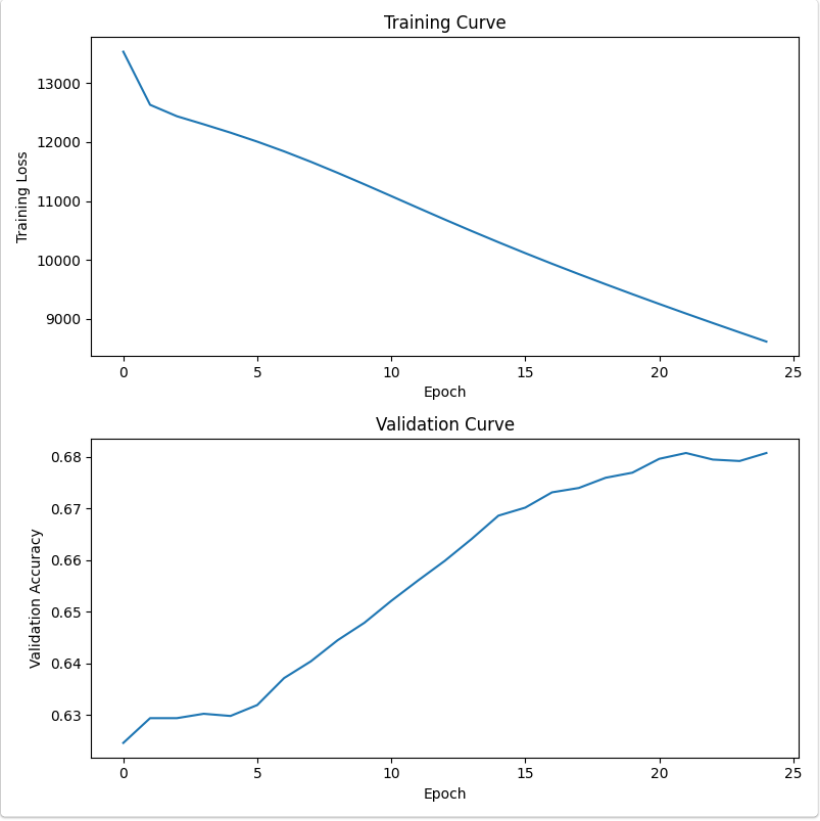
With the following parameters of:

```
# Set optimization hyperparameters.
lr = 0.01
num_epoch = 25
lamb = 0
```

We get:
$k = 10, \text{Valid Acc} = 66.08\%$
$k = 50, \text{Valid Acc} = 67.83\%$
$k = 100, \text{Valid Acc} = 68.00\%$
$k = 200, \text{Valid Acc} = 68.12\%$
$k = 500, \text{Valid Acc} = 67.23\%$
Then:
$k^* = 200$

(d)

With $k^* = 200$, we get:



and a test accuracy of: 67.37%

(e)

Based on the performance across the multiple λ values, $\lambda = 0.001$ was chosen. The final validation accuracy is 67.88% with a test accuracy of 67.94%. Based on raw numbers alone, the model performs better without the regularization penalty. However, the difference in accuracy is minimal, with a difference less than a quarter of a percent.

Question 4

The three models that we will be using is kNN, IRT, and NN. This is because we have all three models already implemented in previous questions. For each model, we will use the parameters that we have determined to produce the most accurate results. Thus, we will train the training data on the following models:

- kNN User-based collaborative filtering, with $k = 21$.
- IRT, with iterations $= 100$ and learning rate $= 0.01$.
- NN, with k latent dimensions $= 200$, learning rate $= 0.01$, epochs $= 25$, and $\lambda = 0$.
  For each model, a prediction of the answer made by a student on a question will be generated. Label the outcome of the prediction as a majority between the three models (i.e. the label will be "correct" is at least two models predict the answer to be "correct" and vice versa).

```
Majority validation accuracy: 0.6908
Majority test accuracy: 0.6907
```

  The ensembling process gives us an accuracy that is better than both kNN and neural network, but worse than IRT. However, it is important to note that the differences in accuracy is marginal, with the biggest difference being around 1%. This is to be expected as we are essentially taking the mean of the three models, with all three models being in the 68% - 70% range.