CSC311 Final Project

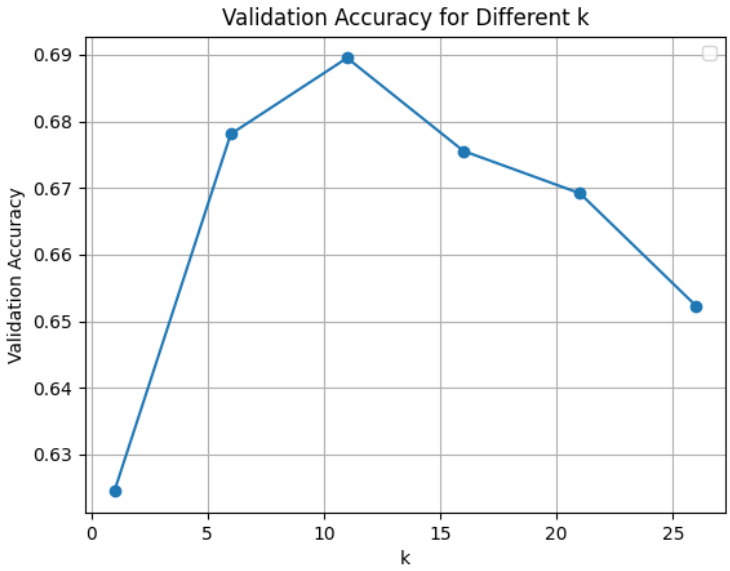
Nishi H. Vinayak M. Natasha S.

August 9th, 2024

Part A

Question 1: *K*-Nearest Neighbors

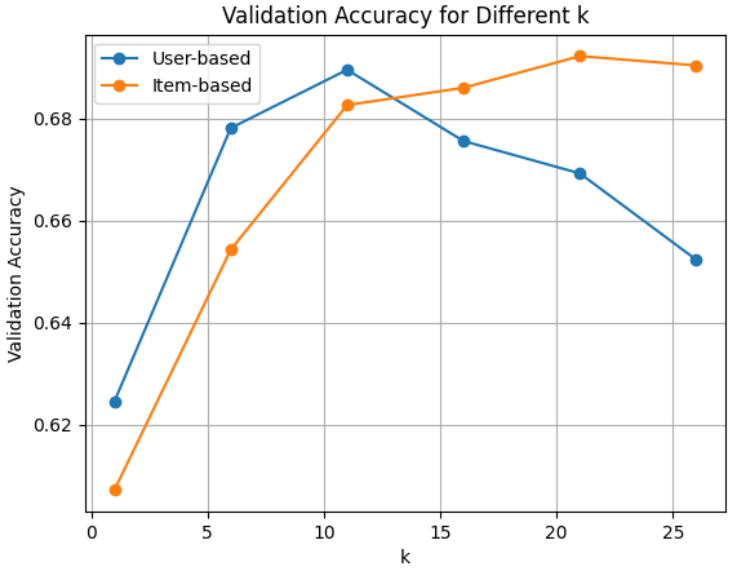
(a) Plotting and reporting the accuracy on the validation data as a function of *k*



(b) Choosing the value of k with highest performance validation data

The k value with the best performance is 11 with validation accuracy of 0.6895286480383855 and test accuracy of 0.6816257408975445.

(c) Item based collaborative filtering



The k value with the best performance for item-based collaborative filtering is 21 with validation accuracy of 0.6922099915325995 and test accuracy of 0.6816257408975445.

(d) Comparing test performance between user- and item-based collaborative filtering

Item-based collaborative filtering is slightly better as we can see from the higher validation rate that comes with it. The average validation accuracy for user-based collaborative filtering is 0.6648555837802239 while the average validation accuracy for item-based collaborative filtering is 0.6687599962367109.

(e) Two potential limitations of KNN for the given text

1. Scalability: k-NN can be computationally expensive and slow for large datasets since it requires calculating distances between all pairs of users or items.

2. Sparsity: The performance of k-NN can degrade in sparse datasets where many users have not rated many items, leading to unreliable distance calculations.

Question 2: Item Response Theory

(a) Deriving log-likelihood and the gradients for θ and β

Deriving Log-Likelihood:

- $p(c_{ij} = 1 \mid \theta_i,\ \beta_j) = \frac{\exp(\theta_i - \beta_j)}{1+\exp(\theta_i - \beta_j)} \ = \ \sigma(\theta_i - \beta_j)$
- $\log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_i \sum_j \ c_{ij} \ \log\left(p(c_{ij} \mid \theta_i, \beta_j)\right) + (1 - c_{ij}) \log\left(1 - p(c_{ij} \mid \theta_i, \beta_j)\right)$
- $\log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_i \sum_j \ c_{ij} \ \log(\sigma(\theta_i - \beta_j)) + (1 - c_{ij}) \log(1 - \sigma(\theta_i - \beta_j))$

Deriving Gradients

- $\sigma'(\theta_i - \beta_j) = \sigma(\theta_i - \beta_j) \cdot (1 - \sigma(\theta_i - \beta_j))$
  - $\sigma'(z) = \frac{\exp(-z)}{(1+\exp(-z))^2} \ = \ \sigma(z) \cdot (1 - \sigma(z))$
  - $z = \theta_i - \beta_i$
- $\frac{d}{d\theta_i} \ \log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_j \ c_{ij} \cdot \ \frac{\sigma'(\theta_i - \beta_j)}{\sigma(\theta_i - \beta_j)} \ + \ (1 - c_{ij}) \cdot \ \frac{-\sigma'(\theta_i - \beta_j)}{1 - \sigma(\theta_i - \beta_j)}$
- $\frac{d}{d\theta_i} \ \log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_j \ c_{ij} \cdot \ \frac{\sigma(\theta_i - \beta_j) \cdot (1 - \sigma(\theta_i - \beta_j))}{\sigma(\theta_i - \beta_j)} \ + \ (1 - c_{ij}) \cdot \ \frac{-\sigma(\theta_i - \beta_j) \cdot (1 - \sigma(\theta_i - \beta_j))}{1 - \sigma(\theta_i - \beta_j)}$
- $\frac{d}{d\theta_i} \ \log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_j \ c_{ij} \cdot \ (1 - \sigma(\theta_i - \beta_j)) \ - \ (1 - c_{ij}) \cdot \ (\sigma(\theta_i - \beta_j))$
- $\frac{d}{d\theta_i} \ \log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_j \ c_{ij} \ - \ \sigma(\theta_i - \beta_j)$
- $\frac{d}{d\beta_i} \ \log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_i \ c_{ij} \cdot \ \frac{-\sigma'(\theta_i - \beta_j)}{\sigma(\theta_i - \beta_j)} \ + \ (1 - c_{ij}) \cdot \ \frac{\sigma'(\theta_i - \beta_j)}{1 - \sigma(\theta_i - \beta_j)}$
- $\frac{d}{d\beta_i} \ \log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_i \ c_{ij} \cdot \ \frac{-\sigma(\theta_i - \beta_j) \cdot (1 - \sigma(\theta_i - \beta_j))}{\sigma(\theta_i - \beta_j)} \ + \ (1 - c_{ij}) \cdot \ \frac{\sigma(\theta_i - \beta_j) \cdot (1 - \sigma(\theta_i - \beta_j))}{1 - \sigma(\theta_i - \beta_j)}$
- $\frac{d}{d\beta_i} \ \log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_i \ c_{ij} \cdot \ -(1 - \sigma(\theta_i - \beta_j)) + (1 - c_{ij}) \cdot \ (\sigma(\theta_i - \beta_j))$
- $\frac{d}{d\beta_i} \ \log(\mathrm{p}(\mathbf{C} \mid \theta,\ \beta)) = \sum_i \ \sigma(\theta_i - \beta_j) \ - \ c_{ij}$

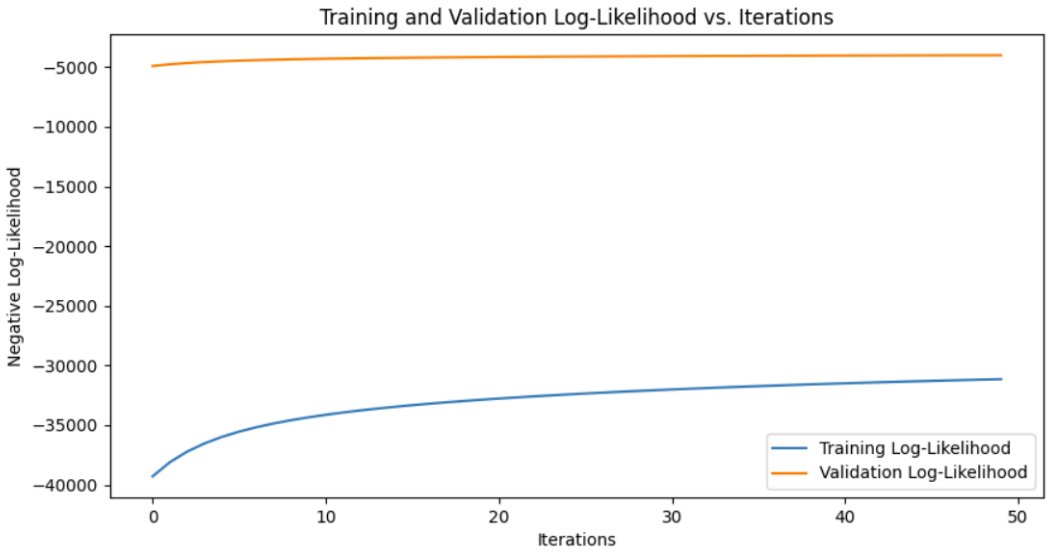
Log Likelihood: $\sum_i \sum_j \ c_{ij} \ \log(\sigma(\theta_i - \beta_j)) + (1 - c_{ij}) \log(1 - \sigma(\theta_i - \beta_j))$
Gradient with θ: $\sum_j \ c_{ij} \ - \ \sigma(\theta_i - \beta_j)$
Gradient with β: $\sum_i \ \sigma(\theta_i - \beta_j) \ - \ c_{ij}$

(b) Performing alternating gradient descent and reporting results

Functions implemented in `item_response.py`. The hyperparameters were selected as follows:

- learning rate: 0.002
- iterations: 50

Training curve:

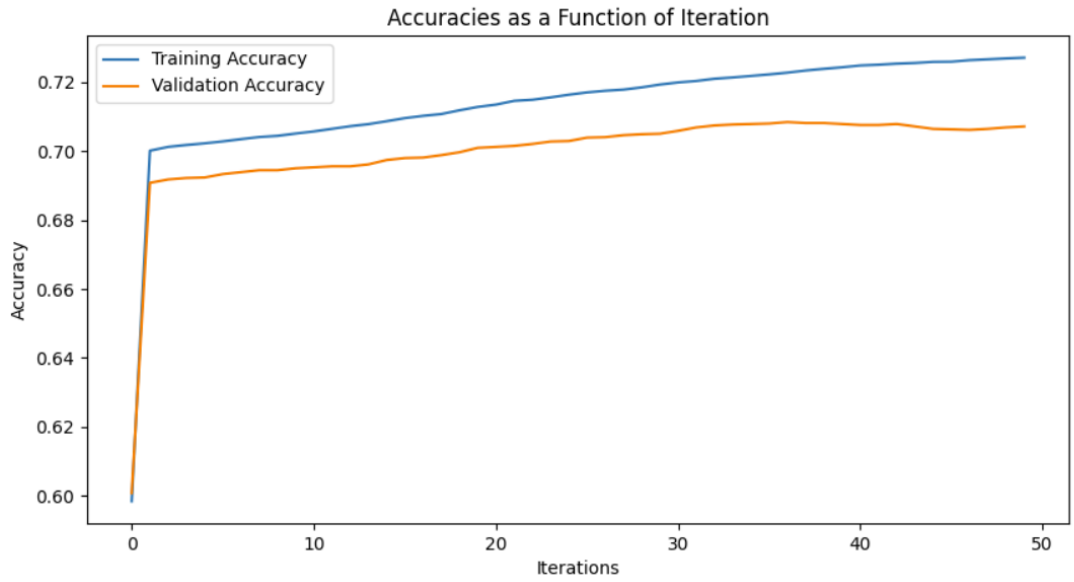


(c) Final validation and test accuracies

`Final validation accuracy:` 0.7071690657634773

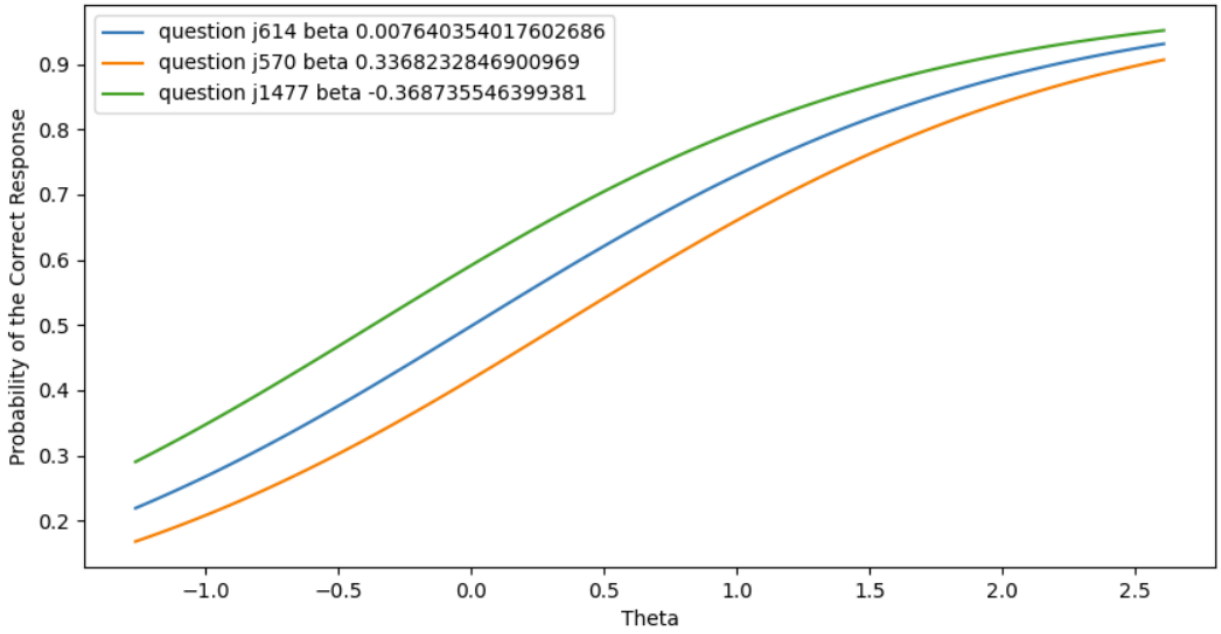
`Final test accuracy :` 0.6996895286480384

Additionally, the training and validation accuracies over the iterations can by given by the following:



(d) Probability of correct responses as a function of theta and a question j for j_1, j_2, j_3

Three questions were selected at random. Using the trained θ and β, three curves representing the probability of the correct response as a function of θ given a question j were plotted as follows:



Since the probability distribution can be likened to the logistic function such that $p(c_{ij} = 1 \mid \theta_i,\ \beta_j) = \sigma(\theta_i - \beta_j)$, the generated curves also follow the shape of the logistic function. The curves represent the probability of the selected questions (614, 570, 1477) with associated difficulties (0.0076, 0.3369, -0.3687) being answered correctly by a student i with ability θ_i.

It can be observed that as θ increases, the probability of correctness increases as well, indicating that students with higher ability levels are more likely to answer a question correctly, while students with lower ability levels are less likely to answer correctly. Additionally, it appears that the curve placed the highest on the graph (question 1477) is associated with the lowest difficulty level, while the curve placed lowest on the graph (question 570) is associated with the highest difficulty level. This indicates that the probability of answering a question correctly is higher when the difficulty is lower, and the probability is lower when the difficulty is higher.

Question 3: Neural Networks

(a) Describing three differences between ALS and neural networks

1. Algorithmic Approach

- **ALS**
  - ALS is a matrix factorization technique often used in collaborative filtering for recommendation systems.
  - It alternates between optimizing user and item matrices by solving least squares problems. This involves iteratively updating one matrix while holding the other fixed, thus minimizing the loss function.
- **Neural Networks**
  - Neural networks are computational models composed of layers of interconnected neurons that adjust their weights based on input data and a loss function.
  - The optimization in neural networks is typically done using gradient-based methods like backpropagation and stochastic gradient descent (SGD) to minimize the loss function.

2. Model Complexity

- **ALS**
  - ALS is generally simpler with fewer hyperparameters to tune, primarily focusing on the number of latent factors and regularization parameters.
  - It has a straightforward linear structure without hidden layers or complex architectures.
- **Neural Networks**
  - Neural networks can be highly complex, with multiple layers (including hidden layers), activation functions, and various hyperparameters (e.g., learning rate, batch size, number of epochs, architecture of the network).
  - The complexity can increase further with different types of neural networks such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs).

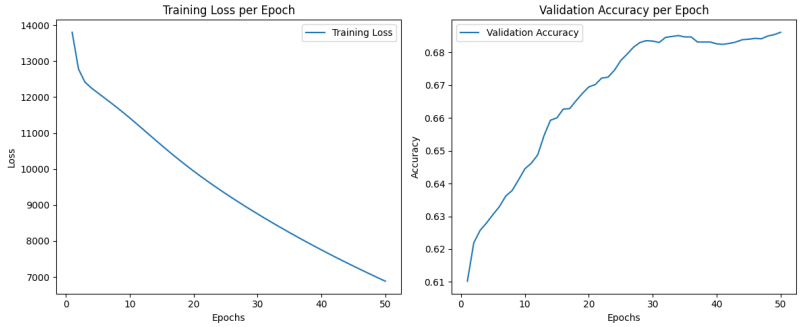
3. Applicability and Flexibility

- **ALS**
  - ALS is specifically tailored for matrix factorization problems, particularly effective in collaborative filtering and recommendation systems.
  - It might not be suitable for other types of machine learning tasks outside matrix factorization.
- **Neural Networks**
  - Neural networks are highly versatile and can be applied to a wide range of tasks, including classification, regression, image and speech recognition, natural language processing, and more.
  - They are flexible in terms of architecture and can be adapted to different types of data and tasks, providing a powerful tool for a variety of machine learning problems.

(b) and (c)

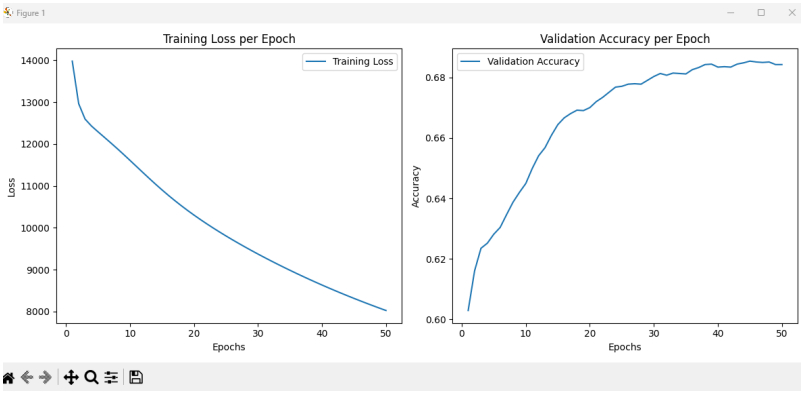
See `neural_network.py`.

(d) Report how training and validation objectives change as a function of epoch




```
Validation Accuracy for k=500: 0.6671
Best latent dimension k*: 50 with validation accuracy: 0.6863
Test Accuracy for k*=50: 0.6765
```

(e) Reporting final validation and test accuracy with chosen λ




```
Final Validation Accuracy for lambda=1: 0.6260
Best lambda: 0.001
Final Validation Accuracy for best lambda=0.001: 0.6843
Test Accuracy for best lambda=0.001: 0.6839
```

The model benefits from the regularization penalty

Question 4: Ensemble

See `ensemble.py`.

The ensemble process consists of 3 steps. The first step was creating bootstrapped samples which was done using a helper function. The helper function took in a sparse matrix and created 3 new matrices representing bagged samples of students. Students that were not chosen in the bag had a row of NaN for their question answers. The second step was training the 3 models using the bootstrapped training sets. The models were all knn models based on student similarity. Each model had it's k value hypertuned by choosing the k value that yielded the highest validation accuracy. Based on these accuracies, we were able to derive the "best" predictions per model. For the third and final step, the predictions from the 3 models were aggregated to produce "average" predictions that were used to check the final validation and test accuracy.

```
Base model validation accuracies: 0.6514253457521874, 0.650578605701383, 0.6429579452441434
Validation accuracy: 0.6593282528930285
Test accuracy: 0.6629974597798476
```

While using the ensemble produces slightly higher final validation and test accuracies in comparison to the base models' validation accuracies, it performs worse overall in comparison to doing 1 model of k-NN with the entire sparse matrix. In this implementation, all base models are k-NN models with similar hyperparameters. If the single k-NN model with the entire sparse matrix was well-optimized, the additional step of bootstrapping and averaging might not provide significant benefits. When the models are too similar, the benefit of bagging is reduced. More diverse base models might have shown greater improvement. For example using the Item Response Theory or Neural Network Models. Using smaller samples o

Part B

Question 1: Formal Description

We modified the Item Response Theory Model by adding a new parameter α_i. α_i represents the student effort parameter, which accounts for the varying levels of effort or engagement that students put into answering questions. Including this parameter helps to model the performance of students more accurately by considering both their inherent ability and their level of effort.

Our resulting model now looks like this:

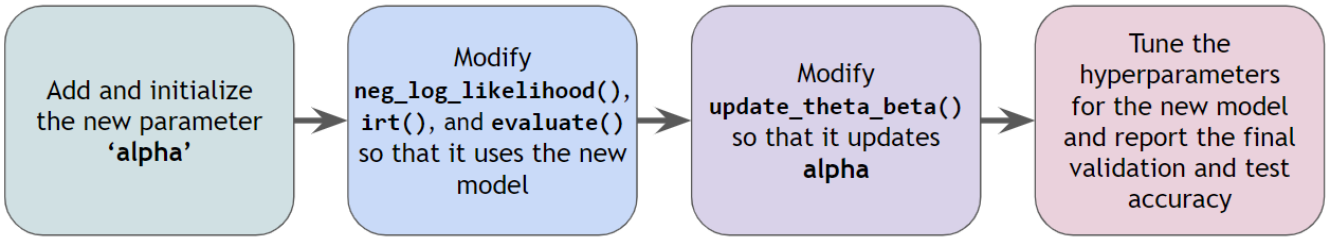
$$p(c_{ij} = 1|\theta_i, \beta_j, \alpha_i) = \frac{\exp((\theta_i + \alpha_i) - \beta_j)}{1 + \exp((\theta_i + \alpha_i) - \beta_j)}$$

In the original one-parameter IRT model, the results suggested that the model might be underfitting because both the training and validation accuracies were very low. The baseline model had a final training accuracy of 0.727120378210556, a final validation accuracy of 0.7071690657634773, and a final test accuracy of 0.6996895286480384. It appears that our model may be too simple to capture all of the nuances in the dataset, so we added a student effort parameter α_i to increase the complexity of our model, capture more details, reduce underfitting, and ultimately make better predictions.

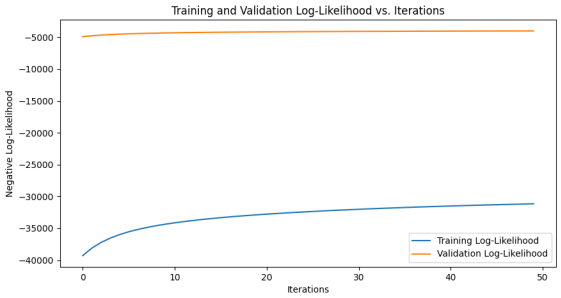
After introducing the α_i parameter, the modified model achieved a final training accuracy of 0.7330299181484617, a final validation accuracy of 0.707874682472481, and a final test accuracy of 0.7025119954840531.

We compared the log-likelihoods of the baseline model and the modified model with the new parameter α_i. The following figures show the log-likelihoods for the training and validation sets as a function of iterations.

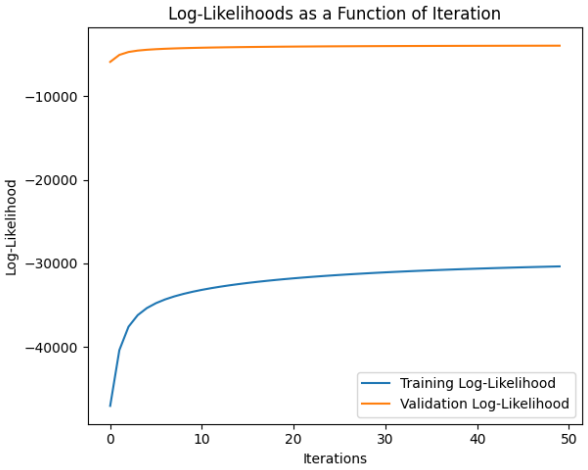
Question 2: Diagram of Model



Question 3: Comparison

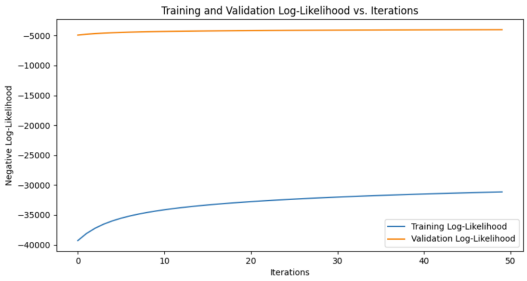


(a) Baseline Model Log-Likelihoods

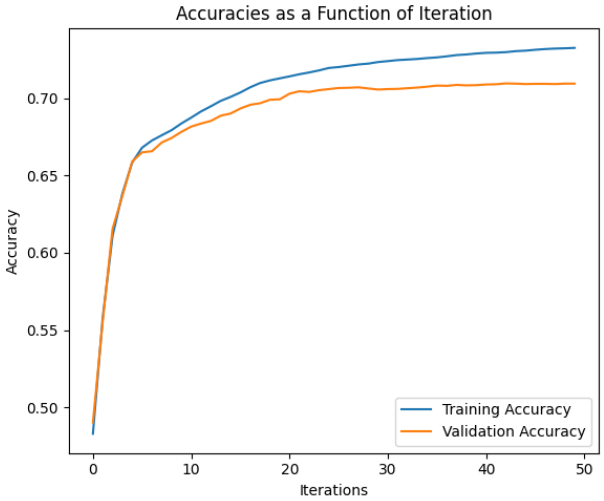


(b) Modified Model Log-Likelihoods

Figure 1: Comparison of Log-Likelihoods for Baseline and Modified Models

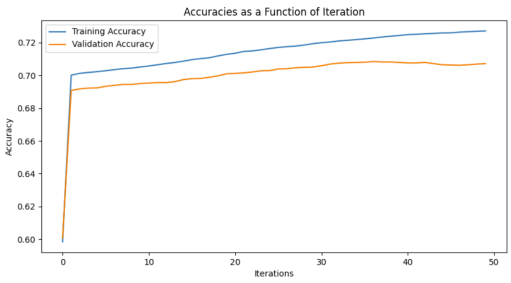


(a) Baseline Model Accuracies

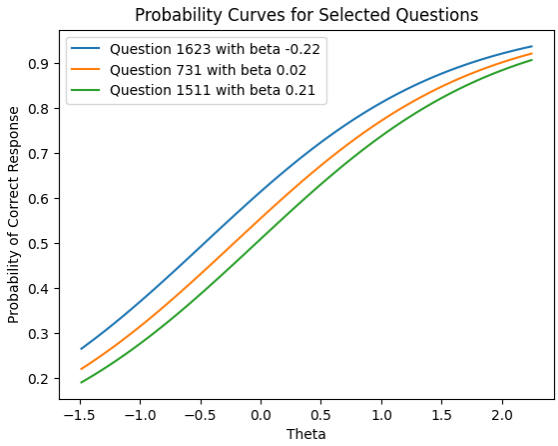


(b) Modified Model Accuracies

Figure 2: Comparison of Accuracies for Baseline and Modified Models



(a) Baseline Model Probabilities



(b) Modified Model Probabilities

Figure 3: Comparison of Probabilities for Baseline and Modified Models

Metric	**Baseline Model**	**Modified Model**
Final Validation Accuracy	0.7071690657634773	0.707874682472481
Final Test Accuracy	0.6996895286480384	0.7025119954840531
Final Training Accuracy	0.727120378210556	0.7330299181484617

Table 1: Comparison of final results for baseline and modified model

The addition of the student effort parameter α_i resulted in a noticeable improvement in our model's performance compared to the baseline. Across all key metrics, including training, validation, and test accuracies, the modified model outperformed the baseline. Furthermore, the log-likelihood values showed a consistent increase, indicating a better fit to the data. For example, after 30 iterations, the log-likelihood of the modified model exceeded 30000, whereas the baseline model struggled to reach this level.

Detailed Analysis of the Results

In the initial one-parameter IRT model, we observed signs of underfitting, which suggested that the model was too simple to capture the complexities of the dataset. To address this, we introduced the student effort parameter α_i, which enhances the model's ability to account for variability in student engagement and effort. This addition allowed the model to capture more nuanced patterns within the data that were previously overlooked by the baseline model.

By increasing the model's capacity to learn from the data, the student effort parameter contributed to more accurate predictions, particularly in scenarios where the baseline model was limited by its simpler structure. The higher log-likelihoods across iterations in the modified model signify that it was able to fit the training data more effectively while also generalizing better to unseen data, as evidenced by the improved test accuracy.

Additionally, the enhanced performance suggests that the model now has a better balance between capturing the essential features of the training data and maintaining robustness during validation and testing. This balance is critical in avoiding both underfitting and overfitting, ensuring that the model can make reliable predictions in real-world applications.

In summary, the modifications made to the model have not only addressed the underfitting issues but have also provided a more comprehensive understanding of the student response patterns. This improvement is reflected in the higher accuracies and log-likelihoods, demonstrating that the modified model is better suited for the task at hand.

To further validate the impact of the student effort parameter α_i on model performance, an experiment could be designed where we systematically vary the complexity of the model while keeping other factors constant. For example, we could compare the modified model with different levels of regularization applied to the α_i parameter. This would help us to disentangle whether the observed improvements are primarily due to better optimization—such as the model's ability to fit the training data more effectively—or due to regularization, which might be reducing overfitting and improving generalization.

By isolating the effect of α_i, we can assess its specific contribution to the model's performance. If we observe that the performance continues to improve with increasing complexity or regularization applied to α_i, it would suggest that the model benefits from enhanced capacity and better control over overfitting. Conversely, if the performance plateaus or deteriorates with further regularization, it might indicate that the primary benefit of α_i lies in its contribution to optimization rather than regularization. This experiment would provide deeper insights into the mechanisms behind the model's improvements, allowing for more informed decisions on model adjustments in future work.

Question 4: Limitations

While our modified model with the α_i parameter has demonstrated improved performance, there are several limitations that should be acknowledged.

One significant limitation is the model's reliance on a sufficiently large and diverse dataset. In scenarios where the dataset is small or the data is sparse, accurately estimating the student effort parameter α_i becomes challenging. This can result in the model overfitting to the limited data, leading to unreliable predictions and reduced generalization to new data. In such cases, a simpler model like the baseline one-parameter IRT model may provide more stable and trustworthy results.

Another concern is the potential reduction in interpretability due to the added complexity from the α_i parameter. While the parameter helps capture more nuanced aspects of student performance, it also introduces complexity that can make it difficult to distinguish between the effects of inherent ability θ_i and effort α_i. This could complicate the process of drawing clear, actionable insights, particularly in educational contexts where understanding the specific drivers of student performance is crucial.

Lastly, the generalizability of the α_i parameter across different student populations is not guaranteed. The behavior of student effort may vary significantly across different educational environments, cultures, or assessment types. If the model is applied to a substantially different population from the one on which it was trained, the parameter estimates may not be accurate, potentially leading to biased or incorrect predictions. Future work could explore adaptive mechanisms that allow the model to better generalize across diverse student groups.

Despite these limitations, the α_i parameter offers a valuable enhancement to the baseline model, providing a more detailed understanding of student behavior when sufficient and high-quality data is available.

Contributions:

Part A:

Question 1- Nishi

Question 2- Natasha

Question 3- Vinayak

Question 4- Nishi

Part B:

Vinayak

All members completed the course evaluation