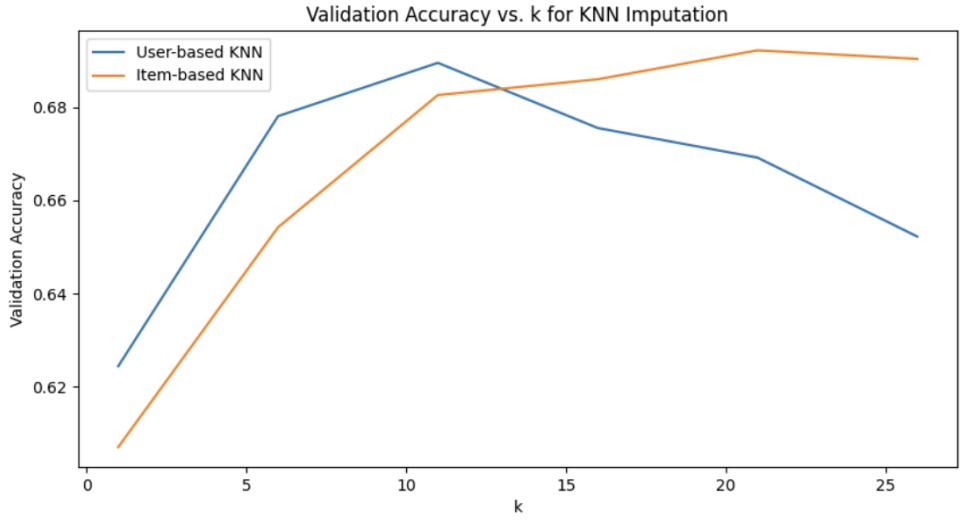
CSC311 Final Report

Part A

Q1:

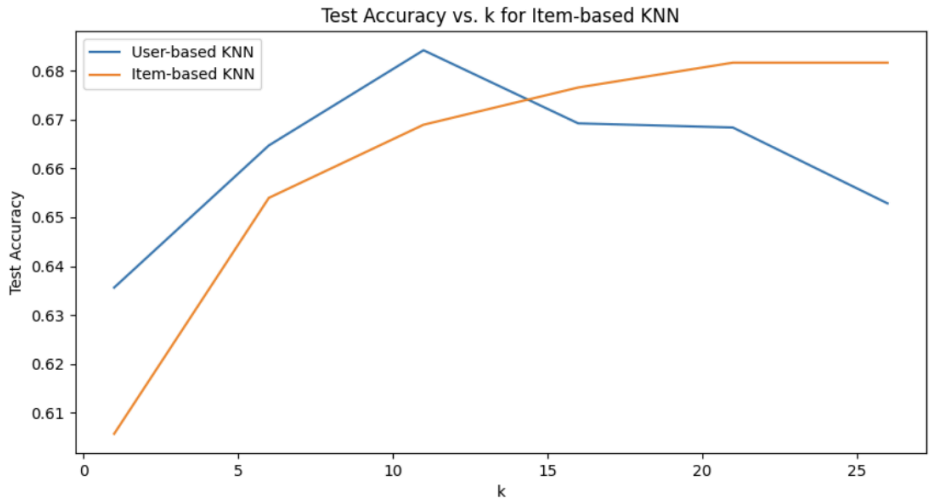


(a)

(b) Best k for user-based KNN: 11 Best k for item-based KNN: 21

User-based KNN Test Accuracy with k=11: 0.6842

Item-based KNN Test Accuracy with k=21: 0.6816



(d)

User-based KNN performs better overall with the highest test accuracy of 0.6841 at k=11.

Item-based KNN has a slightly lower peak test accuracy of 0.6816 at k=21.

User-based KNN is the preferred method for this task based on the test accuracy results.

(e)

Limitation 1: KNN gets slow with lots of data because it has to compare each data point to all others. If there are a lot of students and questions, KNN will take a long time to make predictions.

Limitation 2: If each student answers 1,000 questions, the differences between students become so large that even students who perform similarly will seem very different to KNN. This makes KNN less accurate in predicting answers.

Q2:

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

The likelihood function is: $L(\theta, \beta) = \prod_{i,j} p(c_{ij}|\theta_i, \beta_j)$

$$\Rightarrow log(L(\theta)) = \sum_{i,j} [c_{ij} \cdot log(p(c_{ij} = 1|\theta_i, \beta_j)) + (1 - c_j) \cdot log(1 - p(c_{ij} = 1|\theta_i, \beta_j))]$$

$$= \sum_{i,j} [c_{ij} \cdot log(\frac{exp(\theta_i - \beta_j)}{1+exp(\theta_i - \beta_j)}) + (1 - c_j) \cdot log(1 - \frac{exp(\theta_i - \beta_j)}{1+exp(\theta_i - \beta_j)})]$$

Where $c_{ij} \cdot log(\frac{exp(\theta_i - \beta_j)}{1+exp(\theta_i - \beta_j)}) = c_{ij} \cdot [log(exp(\theta_i - \beta_j) - (1 + exp(\theta_i - \beta_j)))]$

$$= c_{ij}(\theta_i - \beta_j) - c_{ij} \cdot (log(1 + exp(\theta_i - \beta_j)))$$

And $(1 - c_{ij}) \cdot log(1 - \frac{exp(\theta_i - \beta_j)}{1+exp(\theta_i - \beta_j)}) = (1 - c_{ij}) \cdot log(\frac{1}{1+exp(\theta_i - \beta_j)})$

$$= log(1) - log(1 + exp(\theta_i - \beta_j)) - c_{ij}log(1) + c_{ij}log(1 + exp(\theta_i - \beta_j))$$

$\Rightarrow log(L(\theta)) = \sum_{i,j} [c_{ij} \cdot (\theta_i - \beta_j) - (log(1 + exp(\theta_i - \beta_j))]$ is the log-likelihood function

Derivative with respect to θ:

$$\frac{\partial logp(C|\theta, \beta)}{\partial \theta_i} = \sum_j [c_{ij} - (\frac{exp(\theta_i - \beta_j)}{1+exp(\theta_i - \beta_j)}] = \sum_j [c_{ij} - p(c_{ij} = 1|\theta_i, \beta_j)]$$

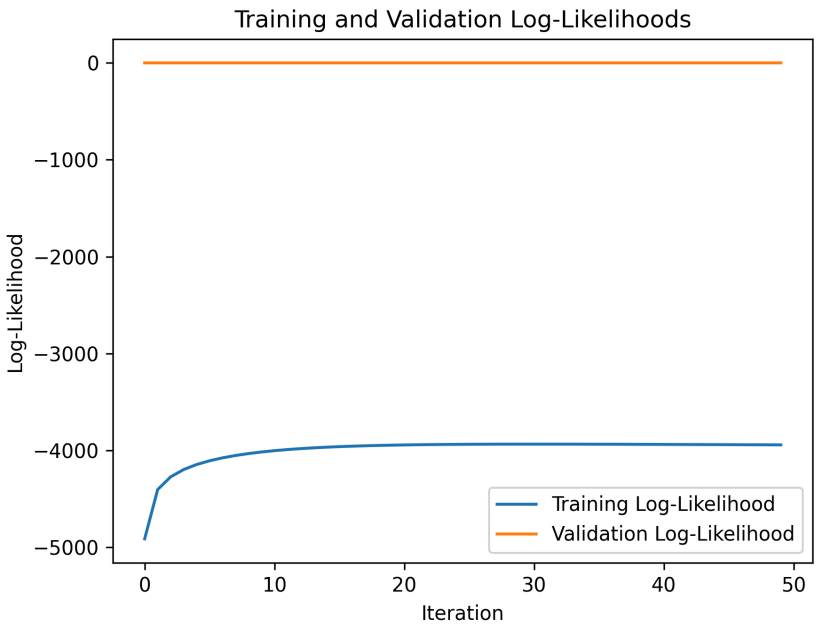
Derivative with respect to β:

$$\frac{\partial logp(C|\theta, \beta)}{\partial \beta_j} = \sum_j [- c_{ij} - (\frac{exp(\theta_i - \beta_j)}{1+exp(\theta_i - \beta_j)}] = - \sum_j [c_{ij} - p(c_{ij} = 1|\theta_i, \beta_j)]$$

(b) **Hyperparameters**: Learning Rate: 0.01, Number of Iterations: 50

Training Curve: The plot below shows the training and validation log-likelihoods over 50 iterations:

Analysis: There is a trend toward a decreasing training log-likelihood, which stabilizes at iteration 10. This suggests that the model is learning from the training set and convergent. The validation log-likelihood stays flat, indicating that additional training is not appreciably improving the model's performance on the validation set. There may be overfitting here.



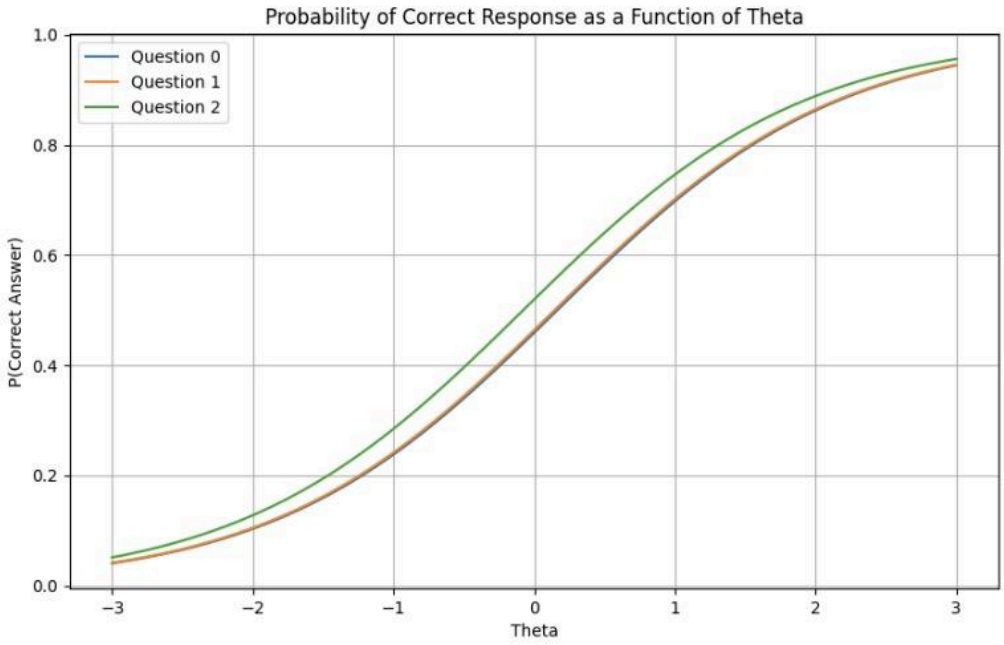
(c) The IRT model's accuracy results are as follows:

- Final Validation Accuracy: 0.7060
- Final Test Accuracy: 0.7067

The close match between Validation Accuracy and Test Accuracy suggests that the model generalizes well, with the validation set being a good predictor of test set performance.

(d) The plot shows the probability of a correct response as a function of student ability (θ) for three questions. Each curve follows an S-shaped pattern, indicating that as student ability increases, the probability of a correct response also increases. The mid-point of each curve, where the probability is 0.5, represents the difficulty of the question (βj). In this plot, the mid-points are similar, suggesting the questions have comparable difficulty levels. The steepness of the curves indicates the discrimination ability of each question; steeper curves can better differentiate between students of varying abilities. This visualization helps evaluate

how well the IRT model assesses student abilities and the characteristics of the questions.



Q3: Option Selected: **Matrix Factorization**

(a) **Hyperparameters:** Value of k tested: 1, 10, 20, 50, 100
Performance:

Value of k	Validation MSE
1	0.6428168219023427
10	0.6586226361840248
20	0.6539655659046006
50	0.648461755574372
100	0.6470505221563647

When k = 1, we have the lowest validation MSE.
Thus, the best validation MSE is 0.6428168219023427, the test MSE for the best k is 0.6477561388653683, for k =1. Higher k, which makes the model more complex, did not lead to better performance, indicating that the more basic model was more adequate.

(b) One of the limitation of SVD is the imputation of the missing entries/ values may reduce the variability in the data and introduce bias into the model. In this case, there are many options

to do fill in the missing values (e.g. fill with 0, fill with mean value). As a result, significant patterns and details in the data may be lost. The model may fail to capture the true range of student abilities and question difficulties if missing entries are filled in with the mean or some other constant.

(c)

Value of k	Validation Loss
1	2126.990991499892
10	2128.0861426200468
20	2126.180069914356
50	2126.879383394865
100	2124.008301021757

When k = 100, we have the lowest Validation Loss of 2124.008301021757
Thus, ALS Test Loss with best k=100 is 1054.2032720203401

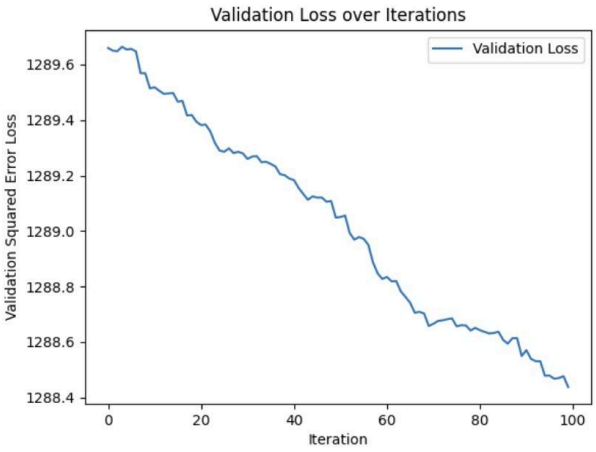
(d) **Hyperparameters:**
Learning Rates: [0.001, 0.01, 0.1]
Number of Iterations: [50, 100, 200]

Result:

k	eta	Number of iterations	Validation Loss	Test Loss
1	0.001	50	2124.949	
10	0.001	50	2125.333	
20	0.001	50	2125.762	
50	0.001	50	2126.675	
…	…	…	…	…
20	0.01	200	2122.388	1053.182

Thus, we got Best k of 20, Best Learning Rate (eta) of 0.01; Best Number of Iterations of 200; with Best Validation Loss of 2122.388 and Test Loss with Best Parameters of 1053.182.

(e) Final Validation Accuracy for ALS: 0.3992
Final Test Accuracy for ALS: 0.4042



Q4: **Performance Comparison of 3 Models Selected and the Ensemble Model**

- **User-based KNN:**
  - Validation Accuracy: 0.6842
  - Test Accuracy: 0.6842
- **IRT:**
  - Validation Accuracy: 0.7060
  - Test Accuracy: 0.7067
- **ALS:**
  - Validation Accuracy: 0.3992
  - Test Accuracy: 0.4042
- **Ensemble:**
  - Validation Accuracy: 0.6008

- Test Accuracy: 0.5958

From the results, we can see that the ensemble method did not outperform the best individual model (IRT) in terms of validation and test accuracy. The IRT model alone achieved the highest accuracy, both on the validation set (0.7060) and the test set (0.7067).

The lower performance of the ensemble can be attributed to the poor performance of the ALS model, which significantly dragged down the overall accuracy when its predictions were averaged with those of the better-performing KNN and IRT models.

In this case, the ensemble did not achieve better performance because: the individual performance of the ALS model was much lower than that of the other two models and averaging predictions with a low-performing model reduced the overall accuracy of the ensemble.

While ensemble methods are generally used to improve model performance by combining multiple base models, it is crucial that the base models themselves have reasonably good performance. In this scenario, the ensemble approach did not lead to better performance due to the significant underperformance of the ALS model. Therefore, a more selective approach in choosing and tuning base models could potentially yield better ensemble performance.

Part B - Report

1. **Formal Description**

   The algorithm we wish to modify to reach higher accuracy when predicting students' responses to diagnostic questions is the IRT Model. The original IRT model estimates student ability (θ) and question difficulty (β) by maximizing the likelihood of observed data through alternating gradient descent. However, this basic model can suffer from overfitting, optimization instability, and limited generalizability, particularly in cases with limited data. Therefore, we proposed several extensions to enhance the performance of the model, which includes L2 Regularization, Early Stopping, Grid Search, Ensemble and AUC/ROC Curve.

   The original IRT model does not penalize large parameter values, which can lead to overfitting. To address this, I plan to add L2 regularization to the negative log-likelihood function. The regularized log-likelihood function is defined as:

$$-\sum_{i,j}\left[c_{ij}(\theta_i-\beta_j)-\log(1+\exp(\theta_i-\beta_j))\right]+\frac{\lambda}{2}\left(\sum_i\theta_i^2+\sum_j\beta_j^2\right)$$

   whereas the regularization term $\frac{\lambda}{2}\left(\sum_i\theta_i^2+\sum_j\beta_j^2\right)$ penalizes large values of θ and β.

   To avoid overfitting during training, early stopping was implemented to halt training when the validation log-likelihood stops improving, preventing overfitting to the training data.

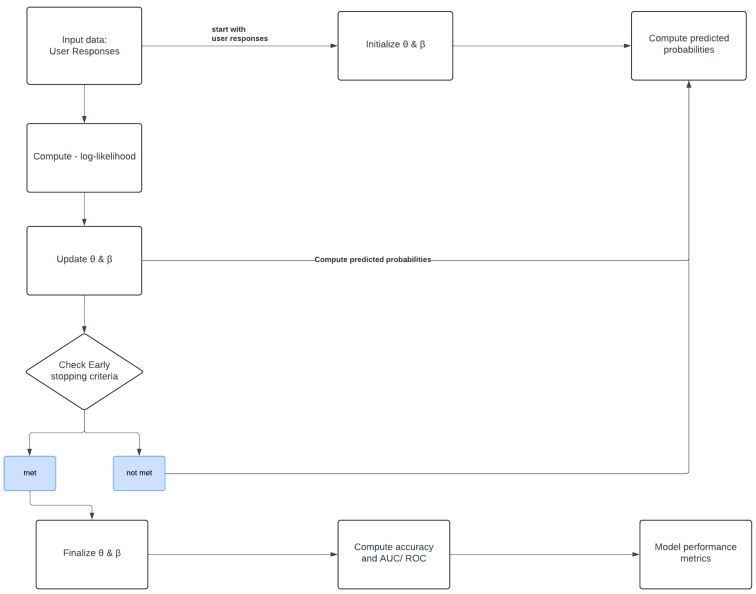
   Grid search was used to systematically explore and optimize key hyperparameters (learning rate, iterations, and regularization strength), ensuring the best possible model configuration.

   To mitigate the sensitivity of IRT models to initialization, multiple models were trained with different starting points, and their predictions were averaged, resulting in more stable and accurate outputs.

   Finally, AUC-ROC was introduced as an additional metric to evaluate the model's ability to discriminate between correct and incorrect responses, providing a more nuanced performance assessment.

   Overall, these enhancements are expected to reduce overfitting, optimize model performance, and improve robustness, resulting in better generalization and more reliable predictions.

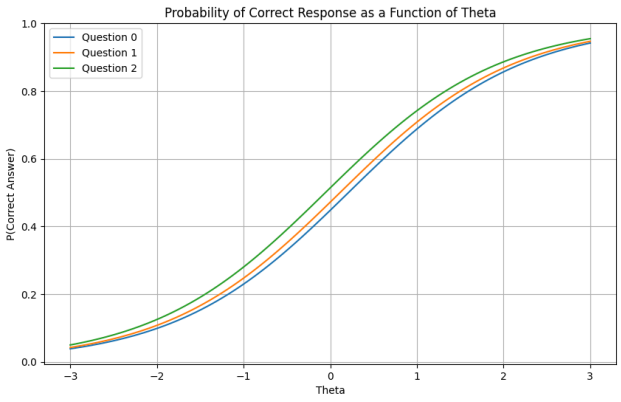
2. **Figure or Diagram**



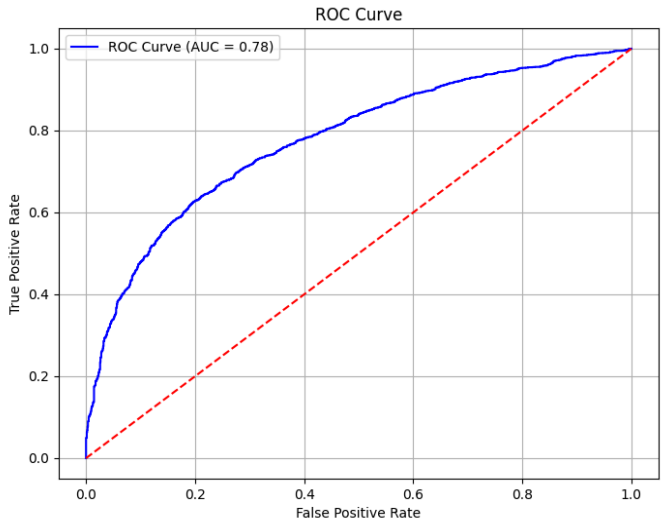
This flow chart shows the workflow for predicting user responses with an enhanced Item Response Theory (IRT) approach. The process begins with inputting user response data and setting parameters θ (user ability) and β (question difficulty). Using these parameters, the model calculates the predicted probabilities of correct responses.

The log-likelihood of predictions is computed, and parameters θ and β are adjusted. The early stopping criteria check determines whether to continue or stop training based on the validation log-likelihood. If the criteria are met, the parameters are finalized; otherwise, the process is iterative.

Once completed, the model's accuracy and AUC/ROC metrics are used to assess performance. Finally, the model's performance metrics are summarized, providing a thorough evaluation of its effectiveness.



The plot shows the probability of correct responses for three questions based on user ability (θ). Each curve represents a question and shows how the probability of a correct answer increases with user ability. The curves demonstrate the model's ability to distinguish between questions of varying difficulty, with each curve's steepness indicating sensitivity to changes in user ability. This visualization provides insights into the discriminative power of the questions within the IRT model.



The ROC curve shows the trade-off between the true and false positive rates for the IRT model at various thresholds. The blue curve shows the model's performance, with higher values indicating better discrimination between positive and negative classes.

The AUC is 0.78, indicating the model's overall ability to correctly classify responses. The red dashed line depicts a random classifier (AUC = 0.5) for comparison, demonstrating that the IRT model outperforms random guessing.

3. Comparison or Demonstration

Model	Validation Accuracy	Test Accuracy	Test AUC/ ROC
Baseline Model	0.7060	0.7067	N/A
Extended Model	0.7089	0.7053	0.7771

To test the model extensions, we designed a grid search experiment to tune hyperparameters such as learning rates, iteration counts, and regularization strength, followed by model ensemble training. Early stopping was used to prevent overfitting by monitoring the validation log-likelihood.

Expected performance enhancements:

- L2 regularization prevents overfitting by penalizing high weights.
- Early stopping ensures that the model doesn't train beyond maximum generalization.
- Grid search identifies optimal hyperparameters.
- Ensemble method averages multiple models' predictions to reduce variance and improve generalization.

Test the Hypothesis:

- Conducted experiments with various configurations.
- Tracked metrics of validation accuracy and AUC-ROC to determine the best model.

Disentangling Effects:

- To isolate the effect of L2 regularization, compare models with and without it.
- Analyze models with various early stopping criteria or hyperparameter settings to determine optimization contributions.

4. Limitations

When there is only a limited amount of training data, the model may struggle to learn accurate θ and β parameters, leading to poor generalization. A possible explanation is that ensemble methods rely on diversity among the base models, which typically requires ample training data. When data is scarce, each model in the ensemble may overfit to different subsets of the data, leading to inconsistent predictions and reduced overall performance. To address the limitations posed by limited data, data augmentation techniques could be used to generate synthetic student responses, helping to stabilize the training of ensemble models. For example, we can have a new neural network model, trained on the existing data to generate new responses that are consistent with the observed data patterns. A generative model could learn the joint distribution of student abilities, question difficulties, and responses. Once trained, this model can generate new student-question pairs with corresponding responses, providing additional data for training, which solves the limited data problem.

The IRT model assumes a simple logistic relationship between θ and β. However, this may not capture more complex interactions, such as when multiple latent traits influence a student's response. The standard IRT model is limited in its ability to model questions that require multiple skills or knowledge areas, as it assumes a single latent trait (ability). This can lead to inaccurate predictions for questions that depend on a combination of different abilities or contextual factors. To address this problem, we can use student and question metadata(we did make an attempt but failed due to IndexError) which could capture more complex relationships between abilities and question difficulties. This would allow the model to better handle questions that rely on multiple skills or knowledge areas, improving predictive accuracy in such cases.

In another setting which we have non-stationary data(i.e the students' ability changes over time and the data are updated frequently), we expect IRT model to perform poorly.The IRT model assumes that the parameters θ and β are static over time. In a learning environment, where student abilities improve or change, the model's predictions may become outdated, leading to decreased accuracy. In this case, we may add adaptive algorithms inside IRT which continuously update the θ and β parameters as new data becomes available. This approach allows the model to remain relevant and accurate even as the underlying data distribution shifts.