CSC311 Machine Learning Final Report

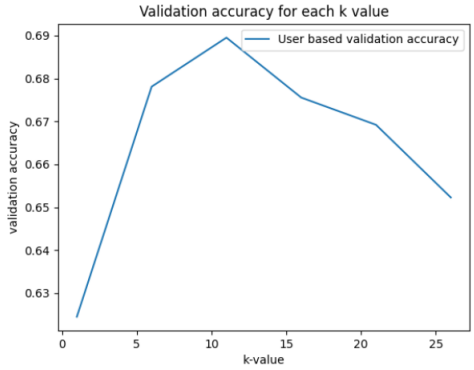
Yutong Dong, Tianhang Wang, Xiaoyuan Mao

December 15th 2020

Part A

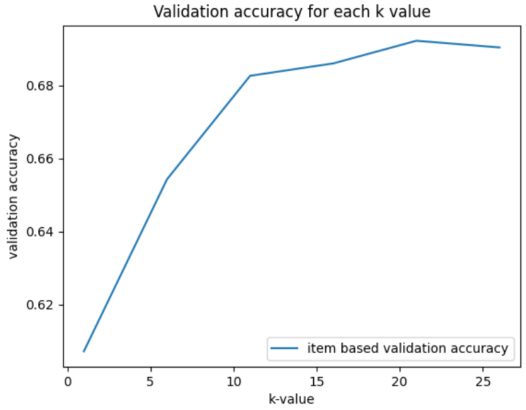
1 k-Nearest Neighbor

(a) The below graph shows the accuracy on validation data as a function of k.



(b) The chosen k* for user based collaborative filtering is 11. Its final test accuracy is 0.6841.

(c) The chosen k* for item based collaborative filtering is 21. Its final test accuracy is 0.6816.

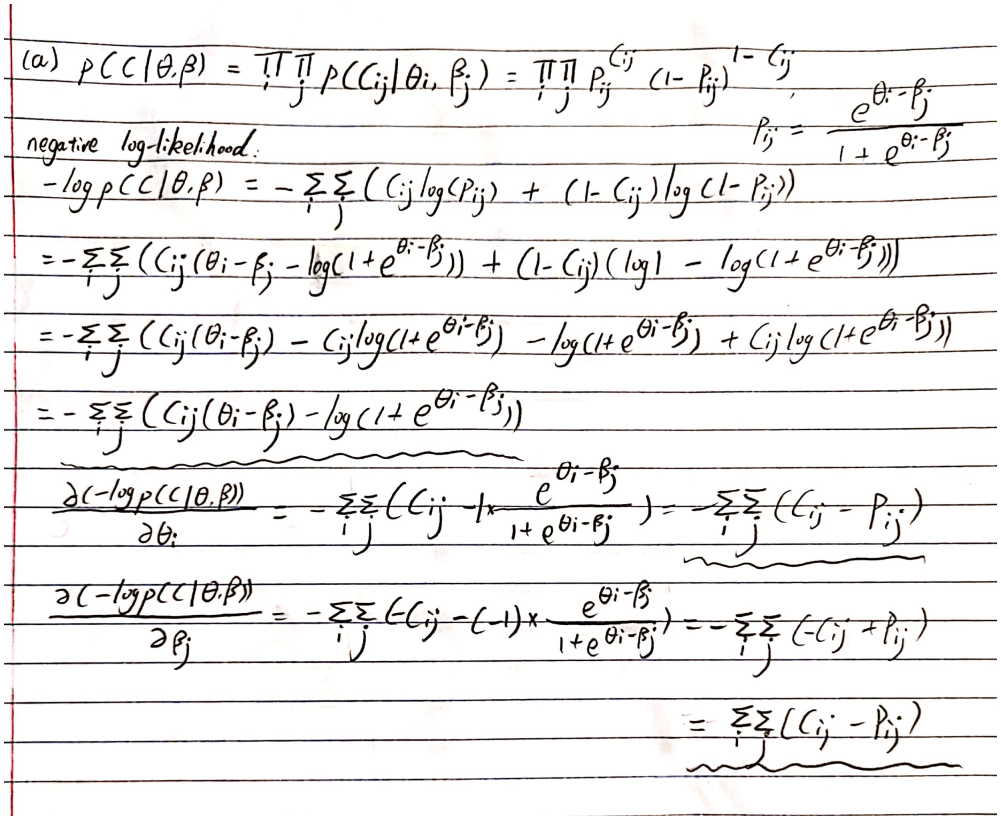


(d) Comparing the test accuracy, $0.6841 > 0.6816$. Obviously, the user based collaborative filtering performs better.

(e) 1. KNN model is sensitive to noisy and missing data. In this task, there are missing entries in the sparse matrix which may hugely affect the result. 2. KNN model does not work well for large data set. Calculating distances between each data instance would be very costly. This data is large and need relatively large memory.

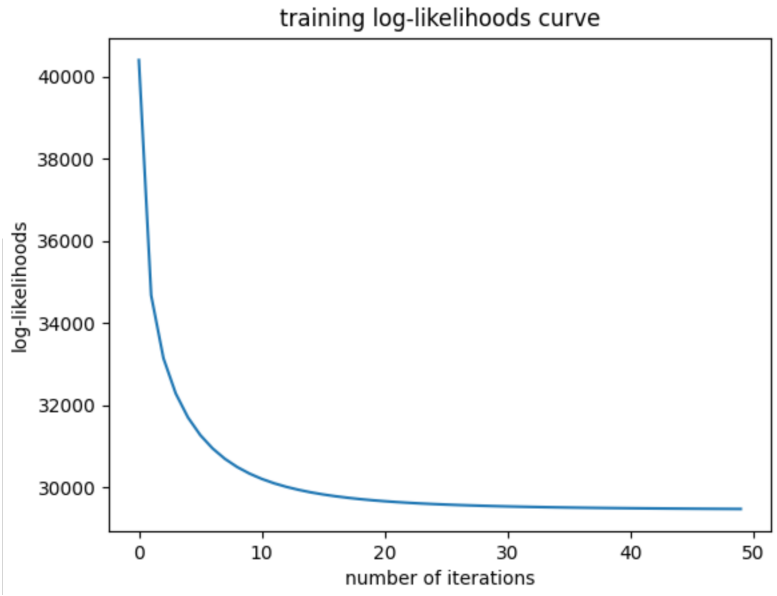
2 Item Response Theory

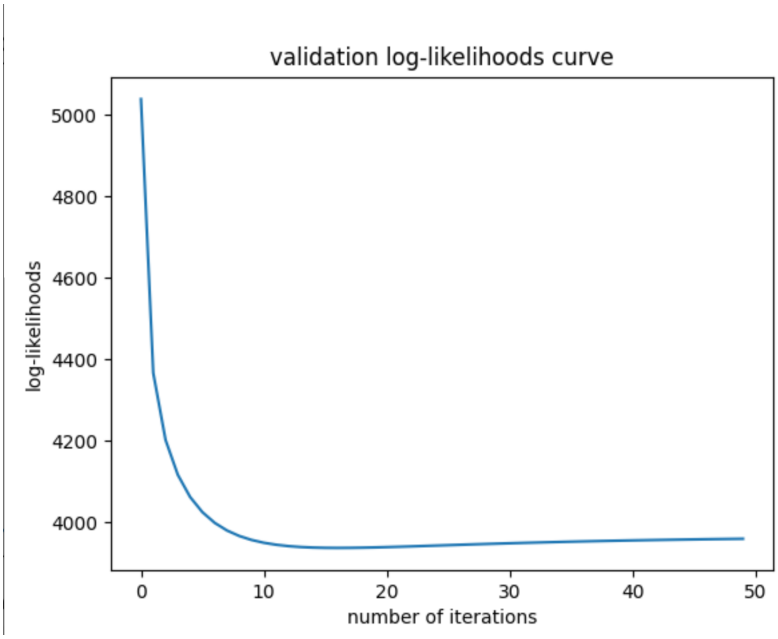
(a)



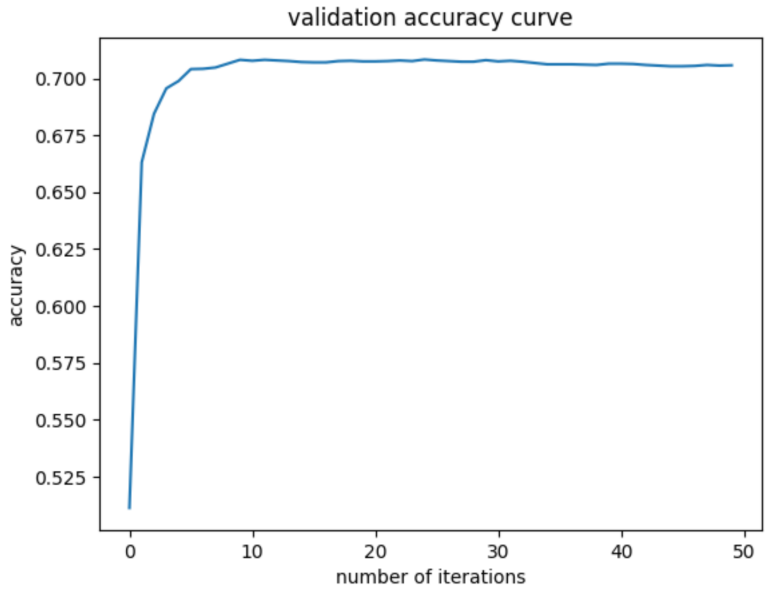
(b) Pick hyperparameters:

learning rate = 0.02 and number of iterations = 50.





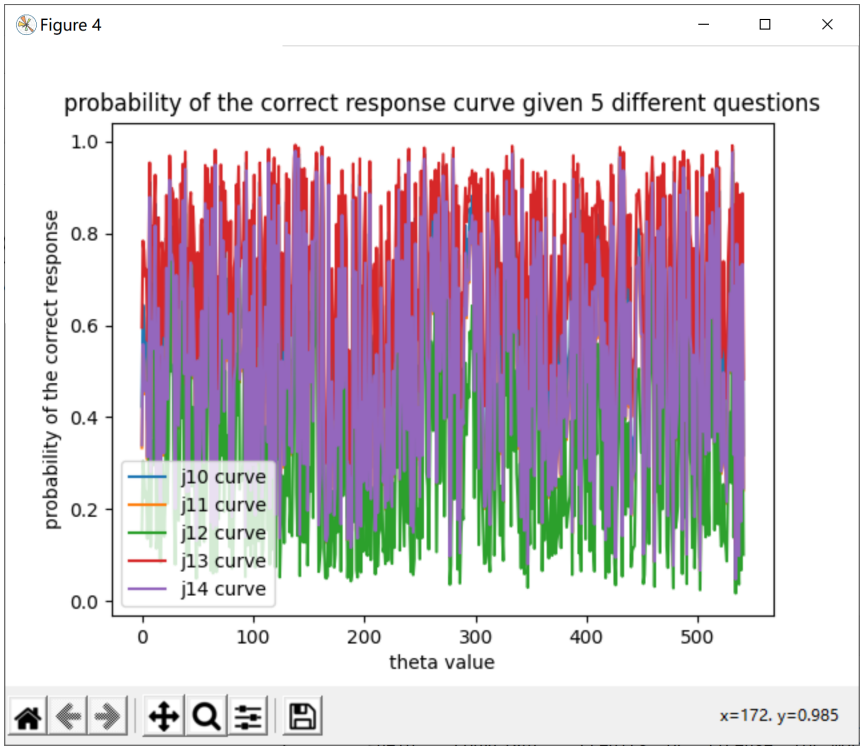
(c)



Report the final validation and test accuracy:

```
validation accuracy 0.70575783234547
test accuracy 0.7090036692068868
```

(d)



The curve keeps fluctuate between certain probabilities. The variation is pretty big. These curves represent the difficulty of these five specific questions. If the curve fluctuates between a interval with bigger probabilities(curve for question 13 mostly fluctuates between(0.5, 1) where as curve for question 12 mostly fluctuates between(0.1, 0.5)), then a student is more likely to get this question right so the question is relatively easy, vice versa. Among these 5 questions, question 13 is the easiest one and question 12 is the most difficult one.

3 Neural Network

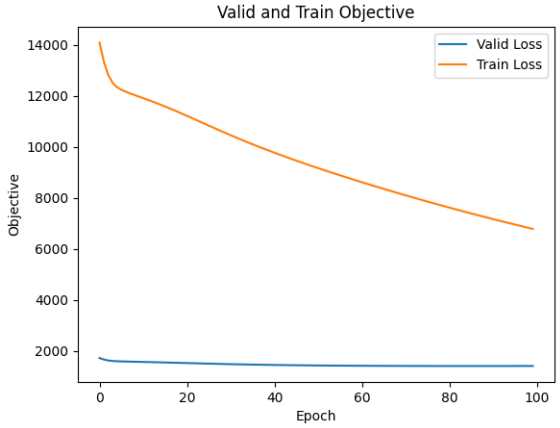
(a) 1. ALS is mainly used for sparse matrices in recommendation systems while neural network are more widely applicable, not only recommendation system.
2. ALS takes derivative of the loss function respects to two vectors alternatively until the ALS converges. The neural network uses the gradient descent method to solve the optimal solution of the network.
3. ALS calculates sparse matrices. The input for neural networks is generally not sparse matrices but images and texts.

(c) Pick hyperparameters: learning rate = 0.005 and number of iterations = 100. Train the autoencoder using latent dimensions of $k \in \{10, 50, 100, 200, 500\}$ and compare the validation accuracy.

```
D:\21' - UOFT\2020 fall\CSC311\final_project\starter_code\part_a>neural_network.py
k=10, lr=0.005, num_epoch=100, lamb=0.01
Training Cost: 9520.364537       Valid Acc: 0.6810612475303415

D:\21' - UOFT\2020 fall\CSC311\final_project\starter_code\part_a>neural_network.py
k=50, lr=0.005, num_epoch=100, lamb=0.01
Training Cost: 6802.939260       Valid Acc: 0.6857183178097658

D:\21' - UOFT\2020 fall\CSC311\final_project\starter_code\part_a>neural_network.py
k=100, lr=0.005, num_epoch=100, lamb=0.01
Training Cost: 5240.948056       Valid Acc: 0.6778154106689246

D:\21' - UOFT\2020 fall\CSC311\final_project\starter_code\part_a>neural_network.py
k=200, lr=0.005, num_epoch=100, lamb=0.01
Training Cost: 3980.359673       Valid Acc: 0.6754163138583121

D:\21' - UOFT\2020 fall\CSC311\final_project\starter_code\part_a>neural_network.py
k=500, lr=0.005, num_epoch=100, lamb=0.01
Training Cost: 3016.580798       Valid Acc: 0.6676545300592718
```

Validation accuracy is highest when k = 50 with accuracy at 0.686. Hence, we pick $k^* = 50$.

(d) Choose $k^* = 50$.



We can see clearly from the plot of training and validation objectives as a function of epoch that both objectives decrease as epoch increases. The validation data set has smaller loss because it has more data and is more complete. It's also possible for under fitting.

```
k=50, lr=0.005, num_epoch=100, lamb=0.001
Valid Acc: 0.6868473045441716
Valid Objective:  1420.8416162765498
Train Objective:  6794.154270768166
final test accuracy:  0.68077900084674
```

The final validation accuracy is 0.687 with loss 1420.84. The final test accuracy is 0.680.

(e) Choose $k^* = 50$, learning rate = 0.005 and number of iterations = 100 from previous questions. Tune the regularization penalty $\lambda \in \{0.001, 0.01, 0.1, 1\}$

```
D:\21' - UOFT\2020 fall\CSC311\final_project\starter_code\part_a>neural_network.py
k=50, lr=0.005, num_epoch=100, lamb=0.001
Valid Acc: 0.6820491109229466
Valid Objective:  1441.3996147103608
Train Objective:  6801.200692534447
final test accuracy:  0.6838837143663562

D:\21' - UOFT\2020 fall\CSC311\final_project\starter_code\part_a>neural_network.py
k=50, lr=0.005, num_epoch=100, lamb=0.01
Valid Acc: 0.6823313576065482
Valid Objective:  1437.4210389856453
Train Objective:  7118.739243507385
final test accuracy:  0.6900931414055885

D:\21' - UOFT\2020 fall\CSC311\final_project\starter_code\part_a>neural_network.py
k=50, lr=0.005, num_epoch=100, lamb=0.1
Valid Acc: 0.6907987581145921
Valid Objective:  1413.3297897534503
Train Objective:  9411.558752536774
final test accuracy:  0.6841659610499576

D:\21' - UOFT\2020 fall\CSC311\final_project\starter_code\part_a>neural_network.py
k=50, lr=0.005, num_epoch=100, lamb=1
Valid Acc: 0.6261642675698561
Valid Objective:  1619.6714406944811
Train Objective:  12625.613329172134
final test accuracy:  0.6229184307084392
```

Validation accuracy is highest when $\lambda = 0.1$ with accuracy at 0.691. The final test accuracy is 0.684.
The model accuracy increases for less than 1%. So we could conclude that irregularity has a little negative impact on the performance of this model.

4 Ensemble

```
Final_valid_acc: 0.6874588390253082
Final_test_acc: 0.682002069809013
```

We ensemble three neural network model together in this part. Firstly, we randomly sample with replacement with respect to students, in other words, we randomly select rows from the train matrix. Each sample consists 542 students, in other words consists 542 randomly selected rows. We create three such samples and and train each sample using Neural Network model. After that, we compare the predicted result to validation and test data to get validation and test accuracy for each model. Lastly, we take the average of validation accuracy and average test accuracy among the three models.
As shown above, the final validation accuracy is 0.6874, and the final test accuracy is 0.6820. Comparing the test accuracy we found in question

3, the validation accuracy decreases by 0.004 and test accuracy decreases slightly by 0.002. The performance is not better off using the ensemble but the accuracy is still very close the ones before ensemble.
Bagging or bootstrap aggregation reduces variance by taking the mean of multiple estimates. However, the bias is unchanged so the accuracy would not change much. Because the data is randomly selected from the sparse, the resulted predicting data varies. The small fluctuation results in this slightly lower validation and test accuracy.

Part B

1 Introduction

We first observe the training data. We discovered that the average correctness for females and males for each question is different. The average absolute difference for gender is 0.1545. So we could safely assume that, for each question, there are some hidden factors related to gender that has some impact on the correctness of a user. To take this hidden factor into consideration, we add an extra term to the model prediction. Since we do not know whether the factor has a negative influence or a positive influence on the correctness of each question, we use weight1 * (conditional average correctness – 0.5) to capture this influence for some weight1.

Similarly, the premium pupil also has a certain impact on the correctness. Therefore, we use a similar technique to capture this influence in the model prediction by adding weight2 * (conditional average correctness – 0.5). The following is the average absolute difference of correctness:

```
gender_diff avg:  0.15451546383121922
premium_pupil_diff avg:  0.327511069029489
```

To determine the exact value of the weight for both weight1 and weight2, we perform some experiments and select the value with the highest validation accuracy. The reason for adding weight * (conditional average correctness – 0.5) is the following: Intuitively speaking, if more than 50 percent of people with a specific gender get the question right, it is more likely for that hidden factor to have a positive impact on the result vice versa. If 99 percent of females get the question right, the student is a female and we know the hidden factor exists. It is very likely that the hidden factor has a strong positive impact on getting the question right; as a result, the female student is more likely to answer this question correctly. The bigger (conditional average correctness – 0.5) is, the stronger impact the factor tends to have. We want to find the best weight so that by adding the weight *(conditional average correctness – 0.5), it captures the influence in the best possible way.

2 Accuracy Summary

Final test accuracy:
k-Nearest Neighbor: 0.6841
Item Response Theory: 0.7010
Ensemble: 0.6820
Neural Network: 0.6833
Improved Neural Network: 0.6911

Final validation accuracy:
k-Nearest Neighbor: 0.6871
Item Response Theory: 0.7090
Ensemble: 0.6874
Neural Network: 0.6850
Improved Neural Network: 0.6937

3 Experiment

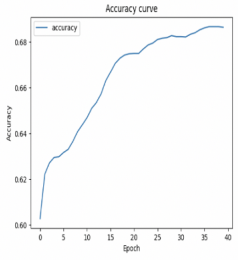
The purpose of this experiment is to find the best tuple (weight1, weight2) such that the validation accuracy is maximized. We recorded data of 13 pairs of weights we've tried in the table below. As shown in the bold number, the pair (weight1 = 0.1, weight2 = 0.2) performs the best among all.

(weight1, weight2) - validation accuracy graph

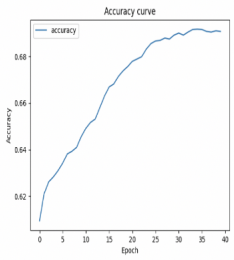
weight1\weight2	0	0.2	0.5	1.2	1.7
0	0.6850	0.6907	0.6854	0.6883	0.6851
0.1	0.6869	**0.6937**	0.6867		
0.5	0.6785	0.6809	0.6864		
1.2	0.6535				
1.7	0.6247				

Fix weight1 = 0 to find out how weight2 affects the validation accuracy.

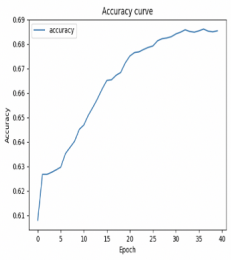
Weight2 = 0:



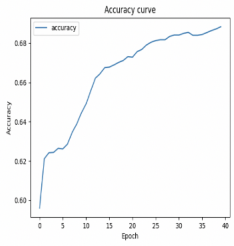
Weight2 = 0.2:



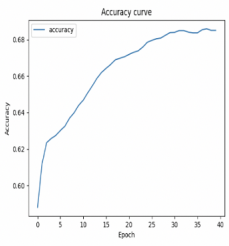
Weight2 = 0.5:



Weight2 = 1.2:



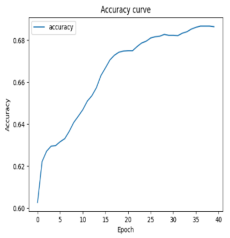
Weight2 = 1.7:



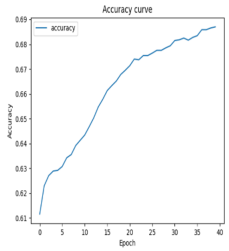
As shown from the table and graphs above, change in accuracy is not very drastic.

To get the best accuracy for weight2 fixing weight1, we would let Weight2 = 0.2. It is a good choice for weight since that it is not too big to influence other terms. Fix weight2 = 0 to find out how weight 1 affects the validation accuracy.

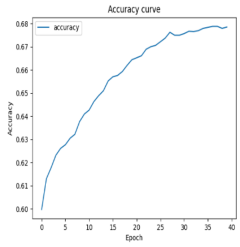
Weight1 = 0:



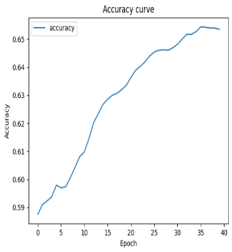
Weight1= 0.1:



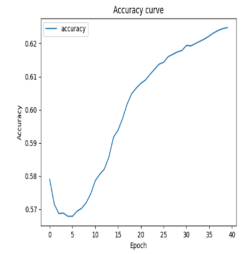
Weight1 = 0.5:



Weight1 = 1.2:



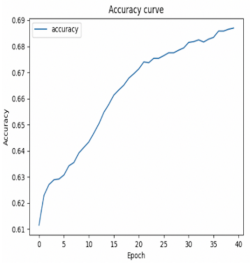
Weight1 = 1.7:



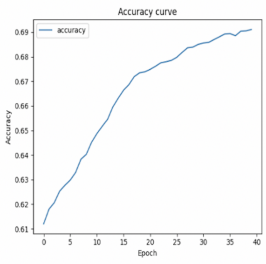
As shown in the graph, when weight1 increases, the final accuracy starts to drastically decrease. The best value for weight 1 while fixing weight2 would be 0.1.

Since the same amount of change would make more difference for weight1 than weight2, we fix weight1 = 0.1 to double check our model is correct:

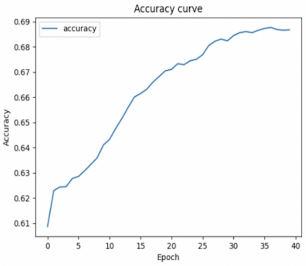
weight2 = 0



weight2 = 0.2



weight2 = 0.5



Indeed, weight1 = 0.1, weight2 = 0.2 would give the highest final validation accuracy of 0.6937.

4 Limitations

1.Hidden features

There could be other features of students that have a significant impact on the correctness. For example, the number of time students spend on the education platform or the academic experience and background of students. In this case, we fail to capture the relationship and hence could not direct the model to make accurate predictions.

To solve this issue, more meta data is required. If given the features, we could use the similar approach to capture that effect by adding weighted terms.

2.Lack of dependency

It could the case that when students are answering questions, their responses are independent of any other features. In the extreme case, for instance, the responses could be entirely random. Then, there is no information for any model to catch. Thus, any model would perform poorly.