CSC311 Final Project

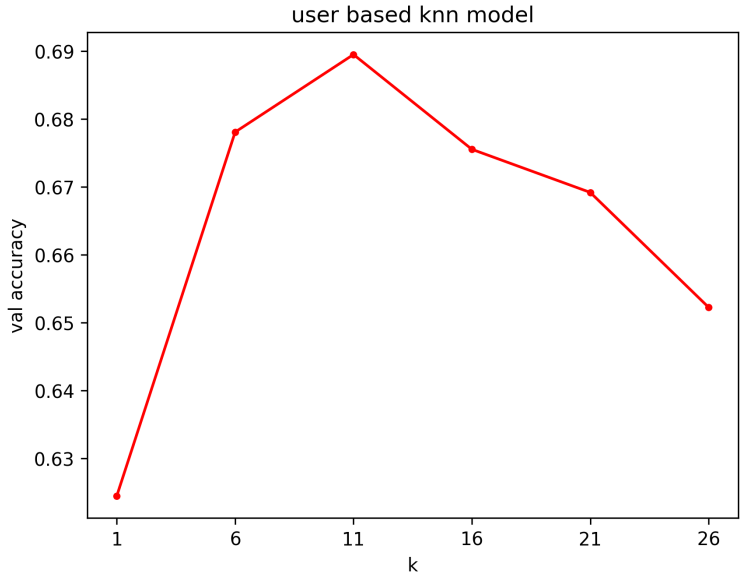
Jin Shang, Naihe Xiao, Chan Yu

December 1, 2022

Part A

Q1 KNN

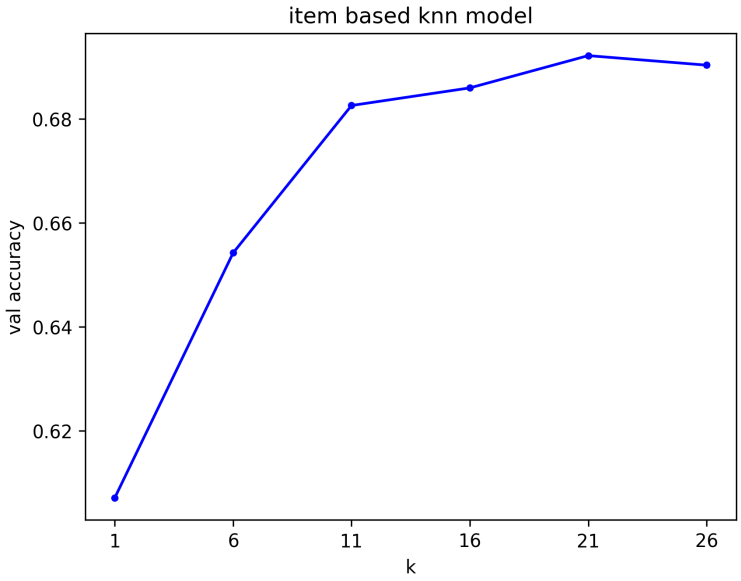
(a) Plot Accuracy of Validation Data
When $k = 1$, validation accuracy is 0.6244707874682472.



When $k = 6$, validation accuracy is 0.6780976573525261.
When $k = 11$, validation accuracy is 0.6895286480383855.
When $k = 16$, validation accuracy is 0.6755574372001129.
When $k = 21$, validation accuracy is 0.6692068868190799.
When $k = 26$, validation accuracy is 0.6522720858029918.

(b) From the plot, we can see that k^* of user-based collaborative filtering should be 11, and when taking $k^* = 11$, the final test accuracy is 0.6841659610499576.

(c) When $k = 1$, validation accuracy is 0.607112616426757.
When $k = 6$, validation accuracy is 0.6542478125882021.
When $k = 11$, validation accuracy is 0.6826136042901496.
When $k = 16$, validation accuracy is 0.6860005644933672.



When $k = 21$, validation accuracy is 0.6922099915325995.
When $k = 26$, validation accuracy is 0.69037538808919.
Underlying assumption: Student's performance on same question should be similar.
From plot we can see that k^* of item- based collaborative filtering should be 21, and when taking $k^* = 21$, the final test accuracy of item- based model is 0.6816257408975445.

(d) When taking the best hyperparameter for both model, we can see that the final test accuracy of user-based model is a little higher than item- based model. And the hyperparameter of item-based model is almost twice as large as those of user-based model. So item-based model is much more computational expensive than user-based model. Thus I prefer user-based model.

(e) Limitation of kNNs:
1. Require high memory, because it need to store all training data.
2. In this case, there is no intuitively explanation of distance between user_id or between question_id.

Q2 Item Response Theory

(a) **Derive log-likelihood** $\log p(\mathcal{C}|\theta,\beta)$

Given the probability $p(c_{ij}=1 \mid \theta_i,\beta_j)=\frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}$ for the i^{th} student answering the j^{th} question, we can get $p(c_{ij}=0 \mid \theta_i,\beta_j)=1-p(c_{ij}=1 \mid \theta_i,\beta_j)=\frac{1}{1+\exp(\theta_i-\beta_j)}$. Thus,

$$p(c_{ij} \mid \theta_i,\beta_j)=\left(\frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}\right)^{c_{ij}}\left(\frac{1}{1+\exp(\theta_i-\beta_j)}\right)^{1-c_{ij}}$$

Let N represent the number of students and D represent the number of questions. We get its likelihood function $p(\mathcal{C}|\theta,\beta)$ for all students and questions as

$$p(\mathcal{C}|\theta,\beta)=\prod_{i=1}^{N}\prod_{j=1}^{D}\left(\frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}\right)^{c_{ij}}\left(\frac{1}{1+\exp(\theta_i-\beta_j)}\right)^{1-c_{ij}}$$

Hence, the log-likelihood function $\log p(\mathcal{C}|\theta,\beta)$ for all students and questions is

$$\begin{aligned}
\ell(\theta,\beta)&=\log p(\mathcal{C}|\theta,\beta)\\
&=\sum_{i=1}^{N}\sum_{j=1}^{D}\left(c_{ij}\log\left(\frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}\right)+(1-c_{ij})\log\left(\frac{1}{1+\exp(\theta_i-\beta_j)}\right)\right)\\
&=\sum_{i=1}^{N}\sum_{j=1}^{D}\Big(c_{ij}(\theta_i-\beta_j)-c_{ij}\log(1+\exp(\theta_i-\beta_j))-(1-c_{ij})\log(1+\exp(\theta_i-\beta_j))\Big)\\
&=\sum_{i=1}^{N}\sum_{j=1}^{D}\Big(c_{ij}(\theta_i-\beta_j)-\log(1+\exp(\theta_i-\beta_j))\Big)
\end{aligned}$$

Derive log-likelihood function with respect to θ_i and β_j get

$$\begin{aligned}
\frac{\partial\ell}{\partial\theta_i}&=\sum_{j=1}^{D}\left(c_{ij}-\frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}\right)\\
\frac{\partial\ell}{\partial\beta_j}&=\sum_{i=1}^{N}\left(-c_{ij}-(-1)\frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}\right)=\sum_{i=1}^{N}\left(-c_{ij}+\frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}\right)
\end{aligned}$$

(b) **Gradient descent on** θ **&** β

Implement cross-entropy to the loss function

$$\begin{aligned}
z&=\theta_i-\beta_j\\
y&=\sigma(z)=\frac{\exp z}{1+\exp z}\\
\mathcal{L}_{CE}&=-t\log y-(1-t)\log(1-y)
\end{aligned}$$

Derive the total loss with respect to θ_i and β_j get

$$\begin{aligned}
\frac{\partial\mathcal{J}}{\partial\theta_i}&=\frac{\partial\mathcal{J}}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial\theta_i}=y-t\\
\frac{\partial\mathcal{J}}{\partial\beta_j}&=\frac{\partial\mathcal{J}}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial\beta_j}=t-y
\end{aligned}$$

Hence, we can find the optimal parameters by gradient descent with a small learning rate α

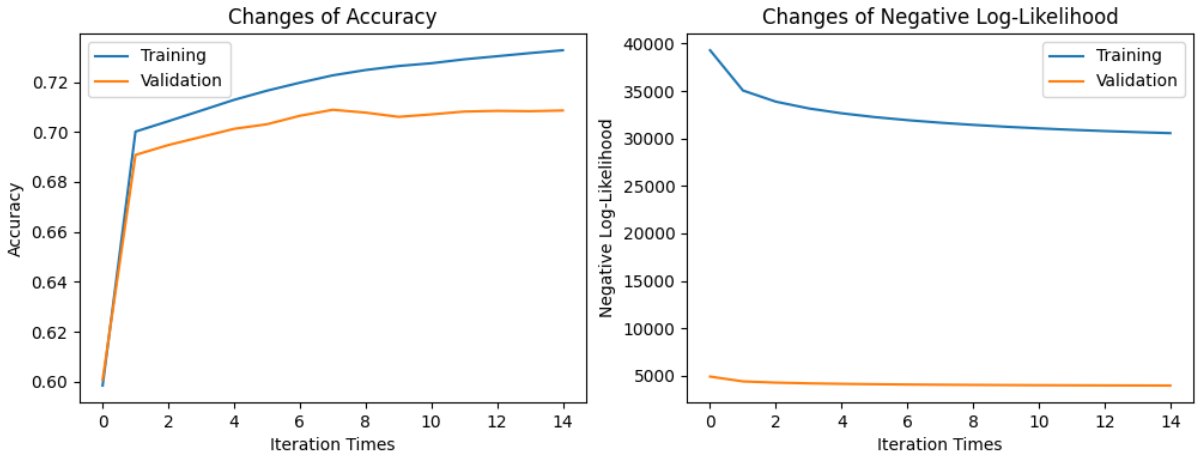
$$\theta_i \leftarrow \theta_i - \alpha \frac{\partial \mathcal{J}}{\partial \theta_i} = \theta_i - \alpha \sum_{i=1}^{N} \left(y^{(i)} - t^{(i)} \right)$$

$$\beta_j \leftarrow \beta_j - \alpha \frac{\partial \mathcal{J}}{\partial \beta_j} = \beta_j - \alpha \sum_{i=1}^{N} \left(t^{(i)} - y^{(i)} \right)$$

In this question, we set the initial vectors of θ and β to 0, and the hyperparameters that yield the highest validation accuracy is

$$\alpha = 0.01, \quad \text{number of iterations} = 15$$

The graph below shows the changes of validation accuracy and log-likelihood amount along with the increasing number of iterate.

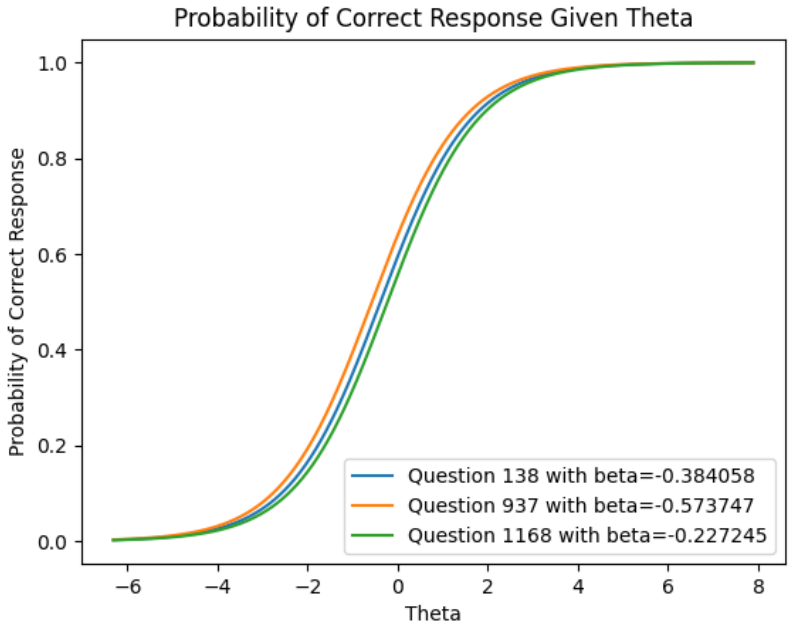


(c) **Final Validation & Test Accuracy**

Final Training Accuracy ≈ 0.7333298
Final Validation Accuracy ≈ 0.70773356
Final Test Accuracy ≈ 0.7030764889

(d) **Plot Curves to Show Probability of Correctness**

All three curves are approximately convex from $\theta = 0$ to 2, with a rapidly growing rate. Then the growing starts to slow down and the curves become flat from $\theta = 2$ to 8, converging to 1. These curves show that as a student's ability increase, the probability for them to give correct responses increases. This increase in probability is steep at first, and slows down when student's ability reaches a certain level, which is reasonable since students with high abilities do not differ in grades significantly.



Q3 Matrix Factorization

(a)
The best k is 15, with a validation accuracy of 0.6565057860570138 and a test accuracy of 0.6539655659046006
(b)
A limitation of SVD in this problem is that the hyperparameter k is chosen from a small set of options, rather than a large pool that would have ensured a better performance of the model. In addition, since SVD choose to fill missing entries of the matrix naively with 0, the performance of the model is expected to be unsatisfying.
(d)
The best learning rate and number of iterations are 0.3, 20000.
(e)
The best k is 15, with a validation accuracy of 0.6560824160316117 and a test accuracy of 0.6550945526390065.

Q4 Ensemble

We use Item response theorey model as base model. The final validation accuracy is 0.6978549252046289, and test accuracy is 0.6886819079875811. Firstly, randomly generate 3 subsample of size 20000 with replacement from train_data set. Then training three small IRT model with these subsample. Then predicting result with these 3 model respectively. Take the majority prediction as result like majority vote. Although the accuracy is not improved, in line with ensemble's properties, I believe we reduced the variance of IRT model. Because the difference between train set accuracy and test set accuracy is smaller than before, which means that ensemble model reduce the variance and prevent overfitting.

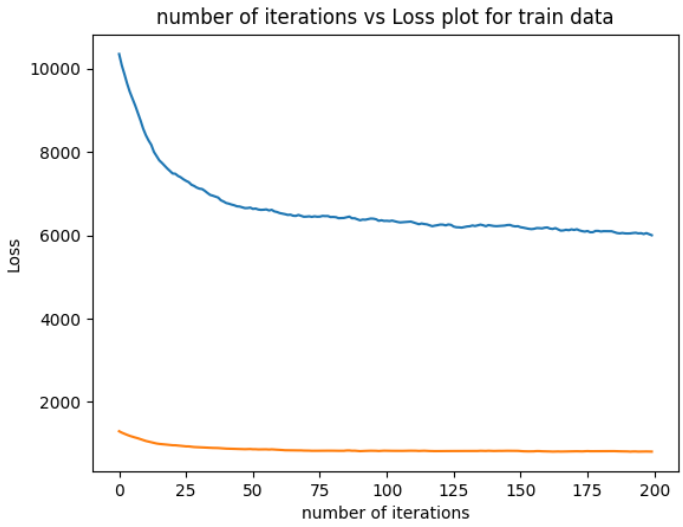


Figure 1: Q3 plot

PartB

1. Our group decide to create a better model based on the Item-Response Theory model we implemented in Part(A), due to the fact that its performance is outstanding among all models. Specifically, instead of a 1 parameter IRT model, we will implement a 2 parameter IRT model. In the 1 parameter IRT model, we only considers the question difficulty and student ability. However, we are not taking into account that sometimes a student might get the right answer by simply guessing. Thus the probability that the question j is correctly answered by student i in the 2 parameter IRT model is

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

As in the above probability, we added parameter c to the model, where
c_j is the probability that a student may get the correct answer by simply guessing question j.
The new log likelihood function is then $l(\theta, \alpha, \beta, c) =$

$$= \sum_{i=1}^{N}\sum_{j=1}^{D} c_{ij} \log\left(c_j + (1 - c_j)\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}\right) + (1 - c_{ij}) \log\left(1 - \left(c_j + (1 - c_j)\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}\right)\right)$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{D} c_{ij} \log\left(c_j + (1 - c_j)\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}\right) + (1 - c_{ij}) \log\left(1 - c_j - (1 - c_j)\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}\right)$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{D} c_{ij} \log\left(\frac{c_j + exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}\right) + (1 - c_{ij}) \log\left(\frac{1 - c_j}{1 + exp(\theta_i - \beta_j)}\right)$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{D} c_{ij} \log(c_j + exp(\theta_i - \beta_j)) - c_{ij} \log(1 + exp(\theta_i - \beta_j)) +$$

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

$$= \sum_{i=1}^{N}\sum_{j=1}^{D} c_{ij} \log(c_j + exp(\theta_i - \beta_j)) + (1 - c_{ij}) \log(1 - c_j) - \log(1 + exp(\theta_i - \beta_j))$$

Derivation with respect to θ_i, β_j, c_j

$$\frac{dl}{d\theta_i} = \sum_{j=1}^{D} \frac{c_{ij} exp(\theta_i - \beta_j)}{exp(\theta_i - \beta_j) + c_j} - \frac{exp(\theta_i - \beta_j)}{exp(\theta_i - \beta_j) + 1}$$

$$\frac{dl}{d\beta_j} = \sum_{i=1}^{N} -\frac{c_{ij} exp(\theta_i - \beta_j)}{exp(\theta_i - \beta_j) + c_j} + \frac{exp(\theta_i - \beta_j)}{exp(\theta_i - \beta_j) + 1}$$

$$\frac{dl}{dc_j} = \sum_{i=1}^{N} \frac{c_{ij}}{exp(\theta_i - \beta_j) + c_j} + \frac{c_{ij} - 1}{1 - c_j}$$

The cost function we are using is the cross entropy function. We will find the optimal parameters by gradient descent. That is, for a small learning rate α,

$$\theta_i \leftarrow \theta_i - \alpha \frac{dJ}{d\theta_i}$$

$$\beta_j \leftarrow \beta_j - \alpha \frac{dJ}{d\beta_j}$$

$$c_j \leftarrow c_j - \alpha \frac{dJ}{dc_j}$$

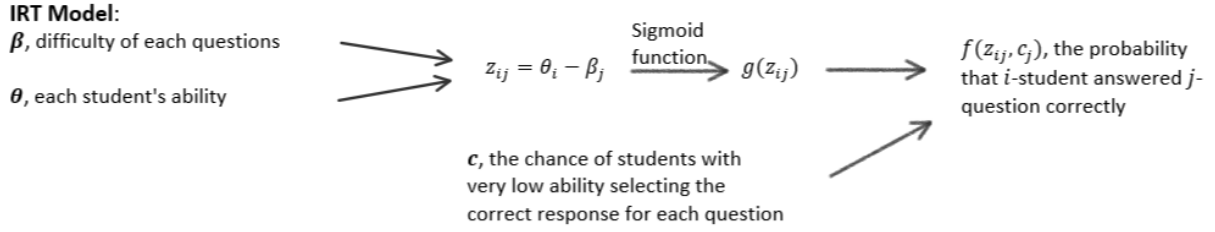
The entire algorithm can be summarized below

Algorithm 1 2 parameters IRT

Initialize β, θ, c to be 0.
for k in range iterations **do**
    for i from 1 to N, j from 1 to D **do**
        $\theta_i \leftarrow \theta_i - \alpha \frac{dJ}{d\theta_i}$
        $\beta_j \leftarrow \beta_j - \alpha \frac{dJ}{d\beta_j}$
        $c_j \leftarrow c_j - \alpha \frac{dJ}{dc_j}$
    end for
end for
Calculate the validation and test accuracy using the optimal parameters

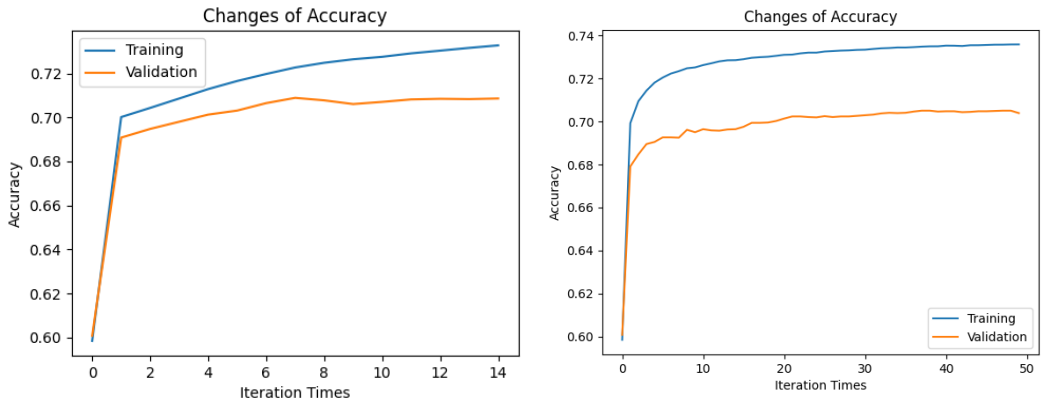
Where α, the learning rate, and the number of iterations, are hyperparameters that we will tune to optimize.

2. **Figure and diagram**



3. **Comparison**

- Comparison of Accuracy



(a) (b)

Figure 2: (a) Accuracy Change for 1PL IRT (base model) (b) Accuracy Change for 2PL IRT

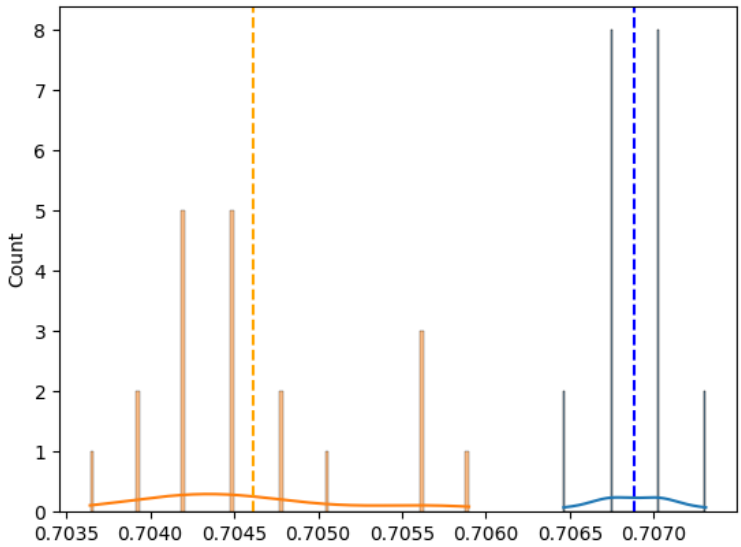
The tuned hyperparameters for the baseline model are 0.01 for the learning rate and 15 for the number of iterations; while for the 2PL IRT, the hyperparameters for the learning rate is the same as 1PL but the number of iteration is 50.
For the 2PL model, the final accuracy for validation data is about 0.704205476 and for the test data is about 0.704487722. Compared with the 1PL model, the validation accuracy decreases by 0.003528084 but test accuracy increases by 0.00141123337.
From the accuracy plots below, I can see that the training accuracy and validation accuracy of our improved model have faster growing rate at the beginning of the iterations than those of the base model. In addition, its accuracy generally higher than that of the base model, and has a smoother shape.
However, there is no noticeable change from either the maximum value of the two accuracy or the minimum value of the two accuracy, or the general pattern, so we can not conclude that we have a statistically significant improvement just from the plots.

- Test Hypothesis

Notice that we have a list of accuracy from both the base model and the improved model, let's denote them A_1, A_2. Our null hypothesis is that the true mean of the accuracy from the base model is equal to the true mean of the accuracy from the improved model, and we will conduct a two sided T test to examine this hypothesis.
The graph below illustrates the accuracy distributions, blue for the baseline model and orange for the improved model. The dashed line indicates their mean value. It shows that the mean accuracy for the improved model is lower than the baseline model.
The Two sided T-Test results shows that there is enough evidence to reject the null hypothesis.

```
Statistics=14.965, p=0.000
Different distributions (reject H_0)
```

Therefore we know that the means of the two accuracy groups differ, as expected.



4. **Limitations**

(a) The first limitation of our approach is that we are not considering the fact that some questions are able to discriminate student abilities better than other questions. That is, questions may have different discrimination abilities. For instance, a relatively easy question like a multiple choice question with one correct answer would be less discriminative than a harder multiple choice question with unspecified number of correct answers.
To overcome this restriction, we can add an additional parameter to our model, α_j, to refer to how well question j is able to discriminate between students with different ability levels. Then the log likelihood function would be

$$\sum_{i=1}^{N}\sum_{j=1}^{D} c_{ij}\log(c + exp(\alpha_j(\theta_i - \beta_j))) + (1 - c_{ij})\log(1 - c) - \log(1 + exp(\alpha_j(\theta_i - \beta_j)))$$

, and we can apply similar optimization algorithms as above.

(b) Lack of data. This is a common limitation for all recommendation systems and their approaches. No matter what optimization algorithm we are using, what probability function we are assuming, the deficiency in data would result in a ineffective model, no matter with respect to fitting or with respect to predicting.
There is nothing we can do on our side to eliminate this limitation, so the process of collecting and wrangling data should be more careful and comprehensive.

(c) No question categories. We did not take into account that the questions come from different subjects. Different students are good at different subjects, and therefore taking into account the question subject is important to having a strong performance.
We can add a predictor to our model by introducing the subject id from the question metadata. We can establish mixed effect model(random intercept and random slope models) to account for possible group effects.