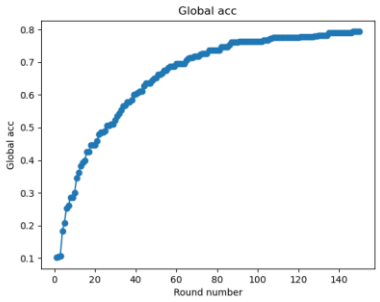
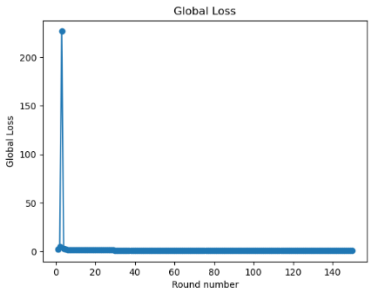
1. 比較 num_user 相同情況下，alpha 分別在 0.1 及 50.0 對訓練出來的 global model accuracy 會有甚麼影響
   a. python main.py --dataset CIFAR10-**alpha50.0**-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 150 --local_epochs 10 --num_users 10 --learning_rate 0.1 --model resnet18 --device cuda(datascience)

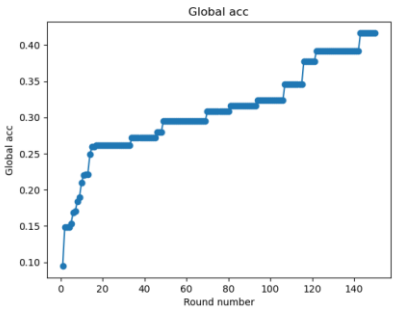
```
------------Round number: 147 ------------


All users are selected
Average Global Accurancy = 0.7870, Loss = 0.81.
Best Global Accurancy = 0.7935, Loss = 0.79, Iter = 146.


------------Round number: 148 ------------


All users are selected
Average Global Accurancy = 0.7927, Loss = 0.79.
Best Global Accurancy = 0.7935, Loss = 0.79, Iter = 146.


------------Round number: 149 ------------


All users are selected
Average Global Accurancy = 0.7847, Loss = 0.82.
Best Global Accurancy = 0.7935, Loss = 0.79, Iter = 146.
Finished training.
```

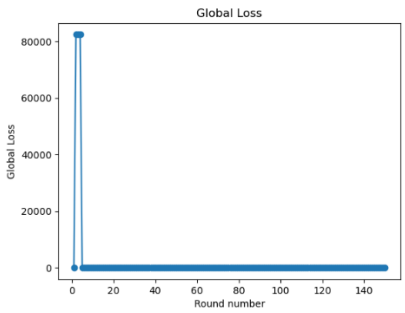




   b. python main.py --dataset CIFAR10-**alpha0.1**-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 150 --local_epochs 10 --num_users 10 --learning_rate 0.1 --model resnet18 --device cuda

```
------------Round number: 147 ------------


All users are selected
Average Global Accurancy = 0.3346, Loss = 1.73.
Best Global Accurancy = 0.3852, Loss = 1.70, Iter = 140.


------------Round number: 148 ------------


All users are selected
Average Global Accurancy = 0.3643, Loss = 1.74.
Best Global Accurancy = 0.3852, Loss = 1.70, Iter = 140.


------------Round number: 149 ------------


All users are selected
Average Global Accurancy = 0.3365, Loss = 1.82.
Best Global Accurancy = 0.3852, Loss = 1.70, Iter = 140.
Finished training.
```





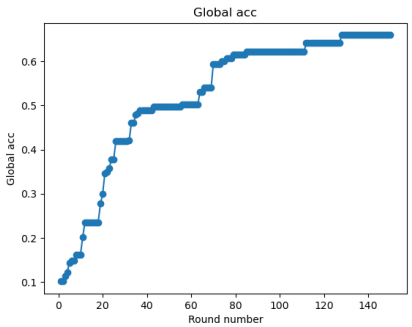
這裡的 alpha 影響抽樣結果，越小的 alpha 可能對資料切割產生以下影響

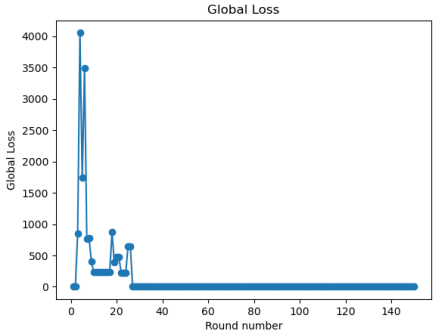
a. 增加異質性: 每個用戶所擁有的類別或樣本分布差異更大
b. 增加不均衡性: 用戶樣本數量不均衡，部分用戶掌握多樣本其餘則少
c. 減少共享資料量: 每個用戶獨特樣本增加，減少彼此間共同資料

從上面兩實驗可看出 alpha 較大者資料裁切較平均，Global acc 上升較快，最終 Global acc 接近 0.8，相較於 alpha 較小者的 Final Global acc 接近 0.4 表現要好上許多

2. 比較 alpha 相同情況下，num user 分別為 2 或 10 對 Global model accuarcy 以及收斂速度差異的影響
   a. python main.py --dataset CIFAR10-alpha50.0-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 150 --local_epochs 10 --**num_users 2** --learning_rate 0.1 --model resnet18 --device cuda

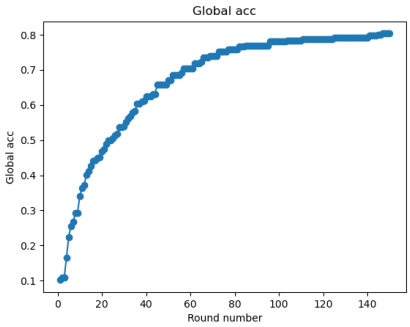
```
------------Round number: 147 -------------


Average Global Accurancy = 0.6129, Loss = 1.24.
Best Global Accurancy = 0.6594, Loss = 1.02, Iter = 127.


------------Round number: 148 -------------


Average Global Accurancy = 0.6330, Loss = 1.12.
Best Global Accurancy = 0.6594, Loss = 1.02, Iter = 127.


------------Round number: 149 -------------


Average Global Accurancy = 0.6366, Loss = 1.15.
Best Global Accurancy = 0.6594, Loss = 1.02, Iter = 127.
Finished training.
```

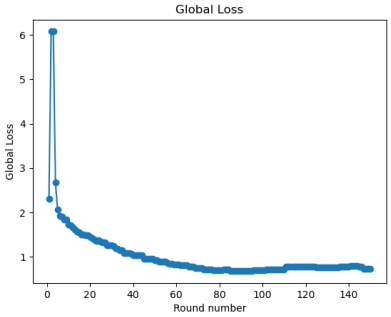




b. python main.py --dataset CIFAR10-alpha50.0-ratio1.0-users10 --algorithm FedAvg --num_glob_iters 150 --local_epochs 10 --**num_users 10** --learning_rate 0.1 --model resnet18 --device cuda

```
------------Round number: 147 ------------


All users are selected
Average Global Accurancy = 0.7982, Loss = 0.77.
Best Global Accurancy = 0.8035, Loss = 0.74, Iter = 146.


------------Round number: 148 ------------


All users are selected
Average Global Accurancy = 0.7904, Loss = 0.82.
Best Global Accurancy = 0.8035, Loss = 0.74, Iter = 146.


------------Round number: 149 ------------


All users are selected
Average Global Accurancy = 0.7948, Loss = 0.82.
Best Global Accurancy = 0.8035, Loss = 0.74, Iter = 146.
Finished training.
```



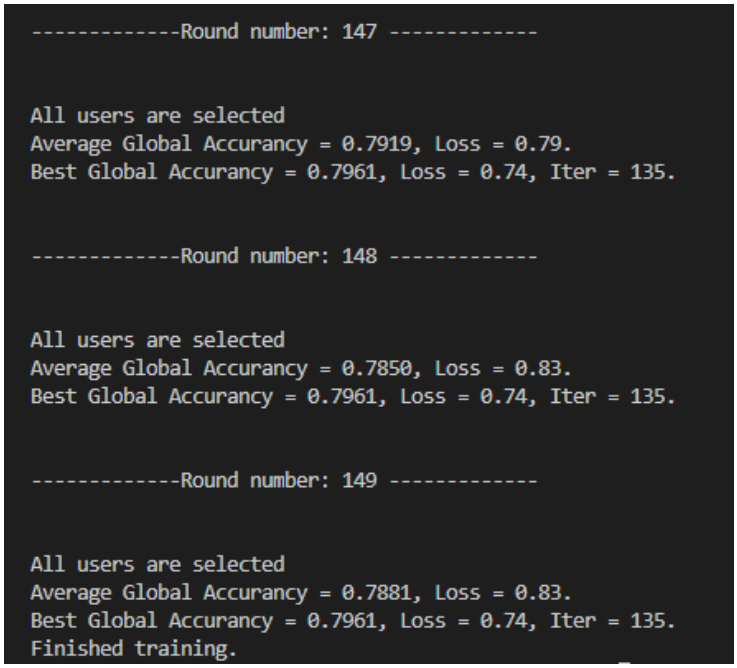


num_user 代表每輪從所有用戶中隨機選擇的用戶數量，比較此兩實驗可看出選擇用戶數量較多的實驗之 Best Global acc 較選擇用戶數量較少的 Best Global acc 高，分別為 0.8035 及 0.6594，且選擇用戶較多的實驗較快就達到收斂，選擇用戶數量增加時表現較好可能原因如下:

a. 收斂速度加快: 較多的用戶參與可以提供更多的 training data 和信息，幫助模型收斂
b. 過擬和風險降低: 模型可以從更多不同樣本和特徵中學習，降低過擬和風險
c. 全局模型準確率提升:較多用戶參與訓練代表更多樣本和信息可以用於全局模型的訓練，可能提升全局模型準確率

因為上述原因，可以解釋在此實驗中 num_user 的提升有助於 Global acc 和收斂速度的提升

3. Final acc 輸出截圖(--num user=10 –alpha=100)



4. 學到的重點

   Federating Learning 基本概念、不同超參數在其中扮演的腳色和其如何影響模型性能等