Assignment 5

1 MNIST (Neural Network)

Quite High accuracy was achieved, in just 10 epochs.

```
Epoch 1/10
300/300 [==============================] - 2s 4ms/step - loss: 0.5238 - accuracy: 0.8525 - val_loss: 0.2629 - val_accuracy: 0.9271
Epoch 2/10
300/300 [==============================] - 1s 4ms/step - loss: 0.2146 - accuracy: 0.9384 - val_loss: 0.2017 - val_accuracy: 0.9442
Epoch 3/10
300/300 [==============================] - 1s 4ms/step - loss: 0.1487 - accuracy: 0.9554 - val_loss: 0.1821 - val_accuracy: 0.9483
Epoch 4/10
300/300 [==============================] - 2s 6ms/step - loss: 0.1012 - accuracy: 0.9716 - val_loss: 0.1683 - val_accuracy: 0.9517
Epoch 5/10
300/300 [==============================] - 2s 6ms/step - loss: 0.0740 - accuracy: 0.9814 - val_loss: 0.1622 - val_accuracy: 0.9508
Epoch 6/10
300/300 [==============================] - 1s 4ms/step - loss: 0.0525 - accuracy: 0.9865 - val_loss: 0.1644 - val_accuracy: 0.9550
Epoch 7/10
300/300 [==============================] - 1s 4ms/step - loss: 0.0412 - accuracy: 0.9893 - val_loss: 0.1973 - val_accuracy: 0.9479
Epoch 8/10
300/300 [==============================] - 1s 4ms/step - loss: 0.0268 - accuracy: 0.9943 - val_loss: 0.1685 - val_accuracy: 0.9550
Epoch 9/10
300/300 [==============================] - 1s 4ms/step - loss: 0.0218 - accuracy: 0.9942 - val_loss: 0.1976 - val_accuracy: 0.9529
Epoch 10/10
300/300 [==============================] - 1s 4ms/step - loss: 0.0148 - accuracy: 0.9969 - val_loss: 0.1793 - val_accuracy: 0.9538
313/313 [==============================] - 1s 2ms/step - loss: 0.1769 - accuracy: 0.9540
Test accuracy: 0.9539999961853027
```

Collab link - https://colab.research.google.com/drive/1HuLGJA64B7IUsLbZ83-CoF0VA6Cz2Ezs?usp=sharing

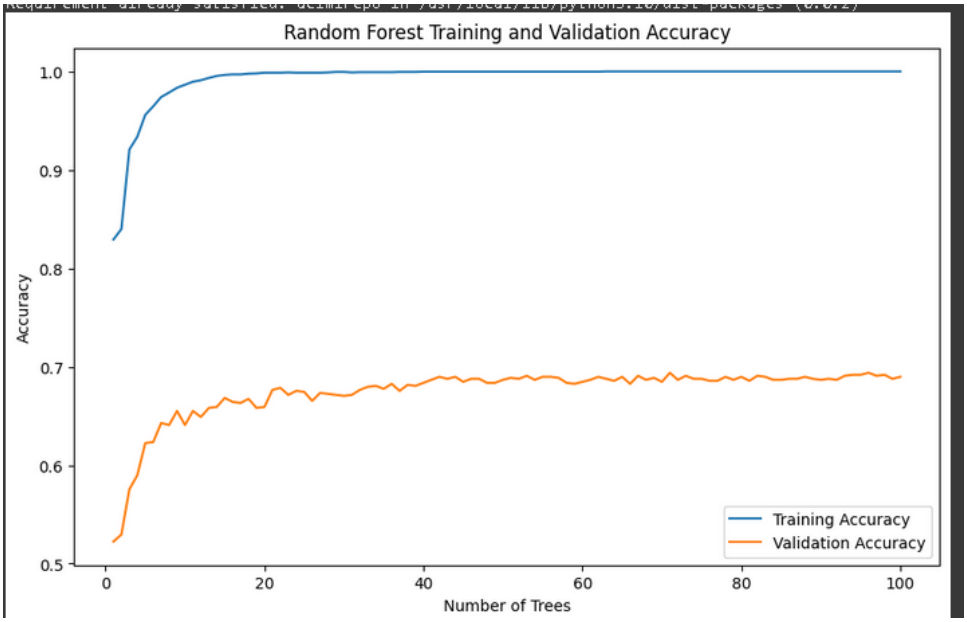
2 Wine dataset from UCI repo (XGBoost)

It is tabular data.It is compared with different gradient boosting ensample algorithms. These algorithms perform better than Neural Networks on tabular kind of data. Validation losses are high here, test accuracy is decent.

```
[16]    validation_0-mlogloss:0.56242   validation_1-mlogloss:0.93285
[17]    validation_0-mlogloss:0.54789   validation_1-mlogloss:0.92725
[18]    validation_0-mlogloss:0.53337   validation_1-mlogloss:0.92361
[19]    validation_0-mlogloss:0.51968   validation_1-mlogloss:0.92036
[20]    validation_0-mlogloss:0.50545   validation_1-mlogloss:0.91726
[21]    validation_0-mlogloss:0.49501   validation_1-mlogloss:0.91739
[22]    validation_0-mlogloss:0.48146   validation_1-mlogloss:0.91365
[23]    validation_0-mlogloss:0.47424   validation_1-mlogloss:0.91300
[24]    validation_0-mlogloss:0.46376   validation_1-mlogloss:0.91143
[25]    validation_0-mlogloss:0.45101   validation_1-mlogloss:0.91017
[26]    validation_0-mlogloss:0.44076   validation_1-mlogloss:0.90944
[27]    validation_0-mlogloss:0.43324   validation_1-mlogloss:0.90733
[28]    validation_0-mlogloss:0.42413   validation_1-mlogloss:0.90646
[29]    validation_0-mlogloss:0.41587   validation_1-mlogloss:0.90559
[30]    validation_0-mlogloss:0.40644   validation_1-mlogloss:0.90534
[31]    validation_0-mlogloss:0.39471   validation_1-mlogloss:0.90363
[32]    validation_0-mlogloss:0.38439   validation_1-mlogloss:0.90473
[33]    validation_0-mlogloss:0.37264   validation_1-mlogloss:0.90350
[34]    validation_0-mlogloss:0.36431   validation_1-mlogloss:0.90146
[35]    validation_0-mlogloss:0.35602   validation_1-mlogloss:0.90031
[36]    validation_0-mlogloss:0.35007   validation_1-mlogloss:0.90038
[37]    validation_0-mlogloss:0.34614   validation_1-mlogloss:0.90078
[38]    validation_0-mlogloss:0.33929   validation_1-mlogloss:0.90176
[39]    validation_0-mlogloss:0.33343   validation_1-mlogloss:0.90263
[40]    validation_0-mlogloss:0.32803   validation_1-mlogloss:0.90454
[41]    validation_0-mlogloss:0.32179   validation_1-mlogloss:0.90601
[42]    validation_0-mlogloss:0.31488   validation_1-mlogloss:0.90633
[43]    validation_0-mlogloss:0.31101   validation_1-mlogloss:0.90684
[44]    validation_0-mlogloss:0.30262   validation_1-mlogloss:0.90628
Test Accuracy: 0.6204081632653061
```

Collab link: https://colab.research.google.com/drive/1H8WsgdWthZgnENu9CkJCgTP6LnXZDFQ-?usp=sharing

3 Wine dataset (Random Forest)

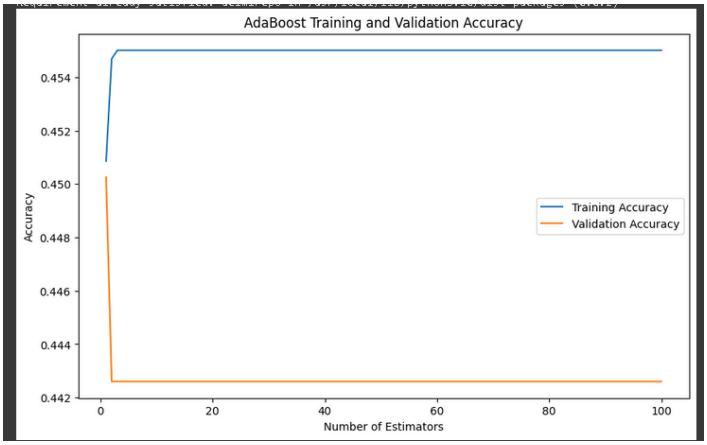



```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.2-py3-none-any.whl (7.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.2
Training Accuracy: 1.0
Validation Accuracy: 0.6581632653061225
Test Accuracy: 0.6887755102040817
```

Test accuracy was better than XGBoost.

Collab link - https://colab.research.google.com/drive/1H8WsgdWthZgnENu9CkJCgTP6LnXZDFQ-?usp=sharing

4 Wine dataset (Adaboost)



The accuracy of adaboost was lowest, and did not improve with increasing no. Of estimators.

```
print(f"Accuracy: {accuracy}")

Requirement already satisfied: ucimlrepo in /usr/local/lib/python3.10/dist-packages (0.0.2)
Accuracy: 0.4377551020408163
```

Collab link - https://colab.research.google.com/drive/1yNjVYZ3uwGBFaEN9Z-qcMWTrnP-MnDAF?usp=sharing

5 CIFAR 100 (CNN)

Accuracy was improving in CNN , with each epoch , but needed quite a bit of Data to train (50000 dataset)

```
Epoch 1/10
625/625 [==============================] - 51s 80ms/step - loss: 4.1554 - accuracy: 0.0666 - val_loss: 3.7504 - val_accuracy: 0.1233
Epoch 2/10
625/625 [==============================] - 52s 83ms/step - loss: 3.5371 - accuracy: 0.1600 - val_loss: 3.3869 - val_accuracy: 0.1824
Epoch 3/10
625/625 [==============================] - 49s 78ms/step - loss: 3.2281 - accuracy: 0.2143 - val_loss: 3.1966 - val_accuracy: 0.2262
Epoch 4/10
625/625 [==============================] - 50s 80ms/step - loss: 3.0212 - accuracy: 0.2526 - val_loss: 3.0188 - val_accuracy: 0.2501
Epoch 5/10
625/625 [==============================] - 52s 83ms/step - loss: 2.8633 - accuracy: 0.2850 - val_loss: 2.9499 - val_accuracy: 0.2662
Epoch 6/10
625/625 [==============================] - 50s 80ms/step - loss: 2.7282 - accuracy: 0.3115 - val_loss: 2.8656 - val_accuracy: 0.2904
Epoch 7/10
625/625 [==============================] - 54s 86ms/step - loss: 2.6207 - accuracy: 0.3311 - val_loss: 2.7494 - val_accuracy: 0.3152
Epoch 8/10
625/625 [==============================] - 50s 80ms/step - loss: 2.5259 - accuracy: 0.3508 - val_loss: 2.7055 - val_accuracy: 0.3151
Epoch 9/10
625/625 [==============================] - 52s 83ms/step - loss: 2.4372 - accuracy: 0.3695 - val_loss: 2.6706 - val_accuracy: 0.3313
Epoch 10/10
625/625 [==============================] - 50s 80ms/step - loss: 2.3690 - accuracy: 0.3812 - val_loss: 2.6362 - val_accuracy: 0.3334
313/313 [==============================] - 3s 11ms/step - loss: 2.6112 - accuracy: 0.3375
Test accuracy: 0.3375000059604645
```

Collab link - https://colab.research.google.com/drive/1BK6n4Hd2qDts93ZPZWxL2axLTrXemHxt?usp=sharing