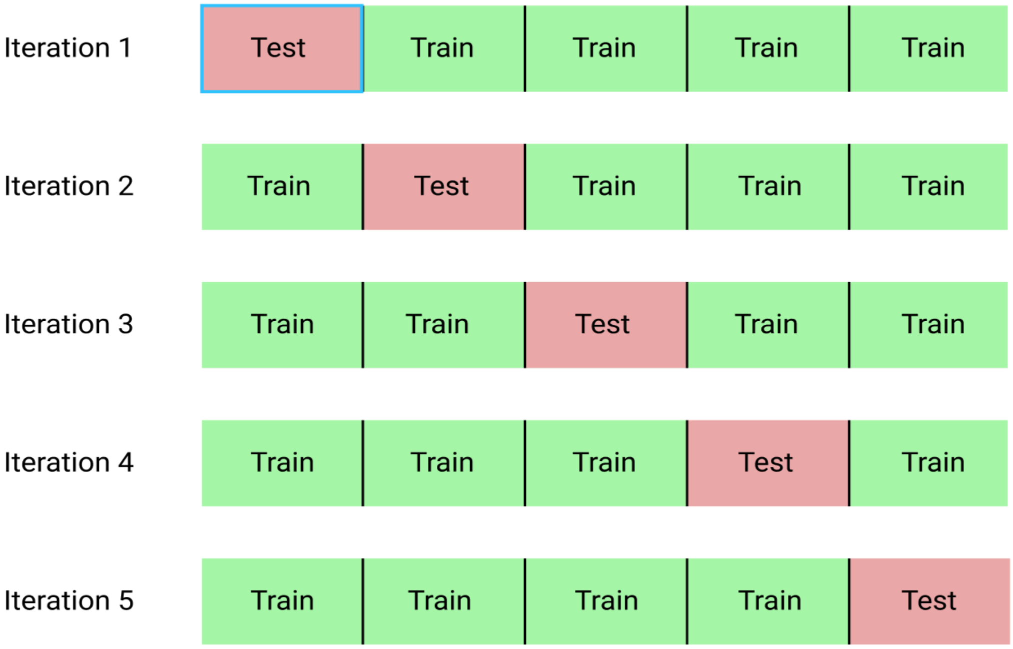
Andika William
16/398497/PA/17458

K-Fold Cross Validation

Case Explanation

The main purpose of validation is to test our current model’s performance if given a new data. There are several methods of validation. One of the simplest one is the split method validation. Basically, we split a portion of the dataset (that is evenly proportionized) that serves as a validator or a tester. For instance, in the iris data set we can divide a block of 30 data. The rest of the data of 120 data, will train the model which will then be tested into the (unused) 30 data block. However, this method does not ensure even treatment of our dataset as it does not account other blocks of data. It might be possible that our model just happen to perform well with the validating block.

In this assignment, we will be applying K-Fold Cross Validaiton unto our Iris dataset. Instead of training our model by using 120 data and validating them with the yet untrained 30 data block, we make divide the dataset into 5 blocks of 30 data. We then implement the training in 5 iteration, switching validator blocks every iteration. In other words, we will be training 5 different models. After it is finished we will calculate the average of the 5 models, and gain a better estimate of how our model will perform given a new data.



Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

Figure 1

Implementation

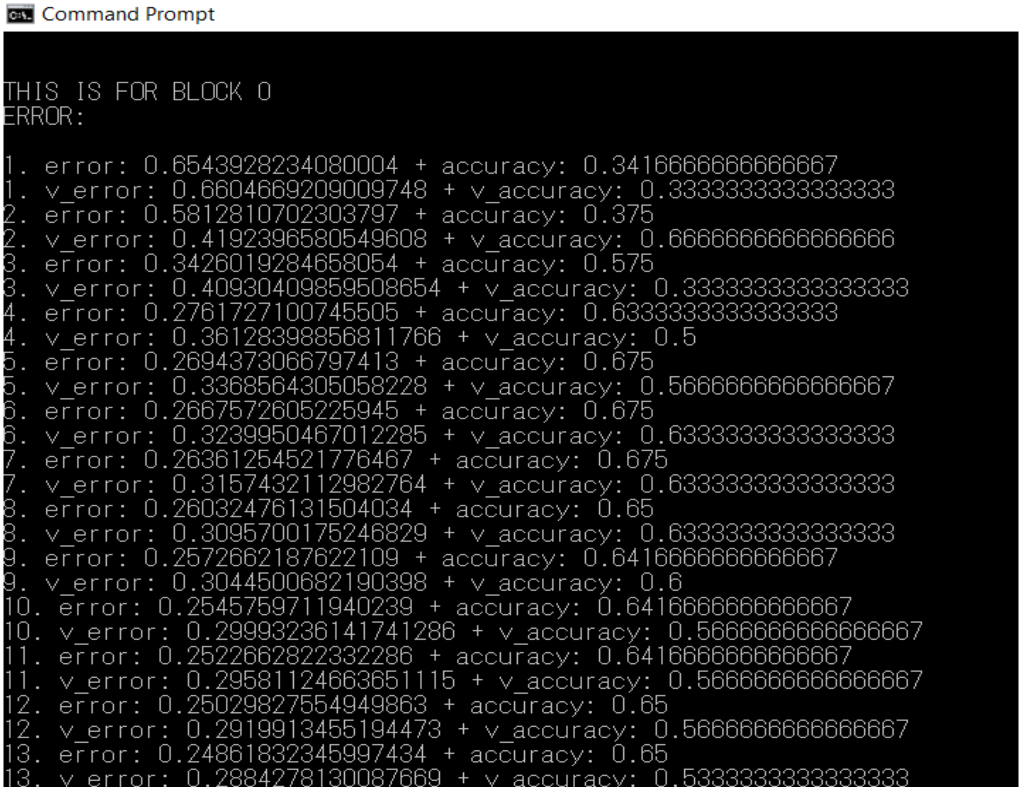
In writing the code, the language that was used in this assignment was Python. It was written in a text editor and was tested in the cmd.

https://github.com/andkwv/iris_kfold

Experiment & Result

Running the code:

Here a learning rate of 0.1 will be implemented for our data. There will be 5 iteration of 100 epochs as we have in total 5 blocks to validate.



```
THIS IS FOR BLOCK 0
ERROR:

1. error: 0.6543928234080004 + accuracy: 0.3416666666666667
1. v_error: 0.6604669209009748 + v_accuracy: 0.3333333333333333
2. error: 0.5812810702303797 + accuracy: 0.375
2. v_error: 0.4192396580549608 + v_accuracy: 0.6666666666666666
3. error: 0.3426019284658054 + accuracy: 0.575
3. v_error: 0.40930409859508654 + v_accuracy: 0.33333333333333333
4. error: 0.2761727100745505 + accuracy: 0.6333333333333333
4. v_error: 0.36128398856811766 + v_accuracy: 0.5
5. error: 0.2694373066797413 + accuracy: 0.675
5. v_error: 0.3368564305058228 + v_accuracy: 0.5666666666666667
6. error: 0.2667572605225945 + accuracy: 0.675
6. v_error: 0.3239950467012285 + v_accuracy: 0.6333333333333333
7. error: 0.26361254521776467 + accuracy: 0.675
7. v_error: 0.3157432112982764 + v_accuracy: 0.6333333333333333
8. error: 0.26032476131504034 + accuracy: 0.65
8. v_error: 0.3095700175246829 + v_accuracy: 0.6333333333333333
9. error: 0.2572662187622109 + accuracy: 0.6416666666666667
9. v_error: 0.3044500682190398 + v_accuracy: 0.6
10. error: 0.25457597119402 39 + accuracy: 0.6416666666666667
10. v_error: 0.29993236141741286 + v_accuracy: 0.5666666666666667
11. error: 0.2522662822332286 + accuracy: 0.6416666666666667
11. v_error: 0.29581124663651115 + v_accuracy: 0.5666666666666667
12. error: 0.2502982755494986 3 + accuracy: 0.65
12. v_error: 0.2919913455194473 + v_accuracy: 0.5666666666666667
13. error: 0.2486183234599743 4 + accuracy: 0.65
13. v_error: 0.2884278130087669 + v_accuracy: 0.5333333333333333
```

Figure 2.

Result :

Block 1:

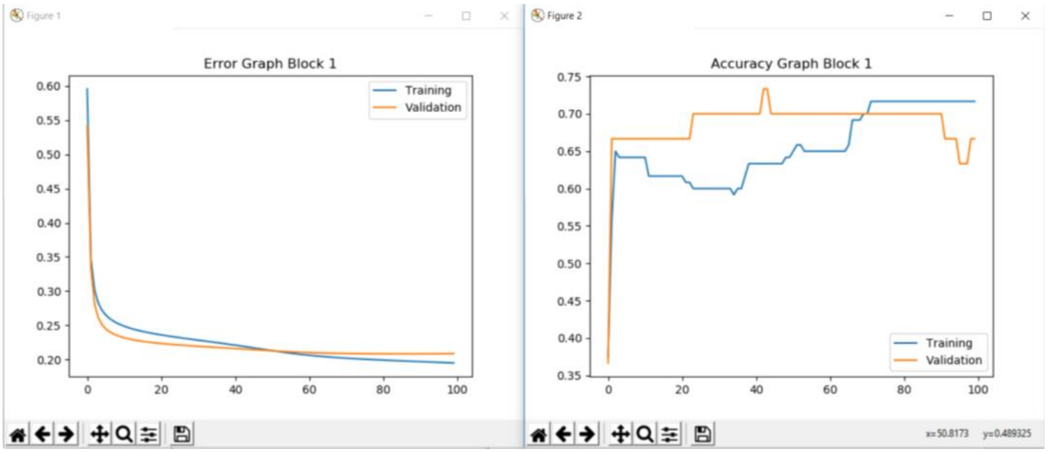


Figure 3.

Block 2:

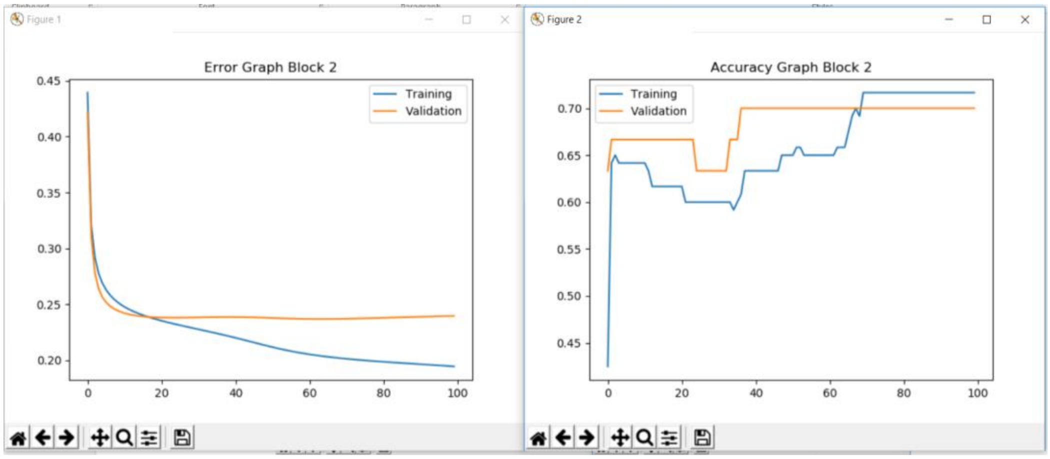


Figure 4.

Block 3:

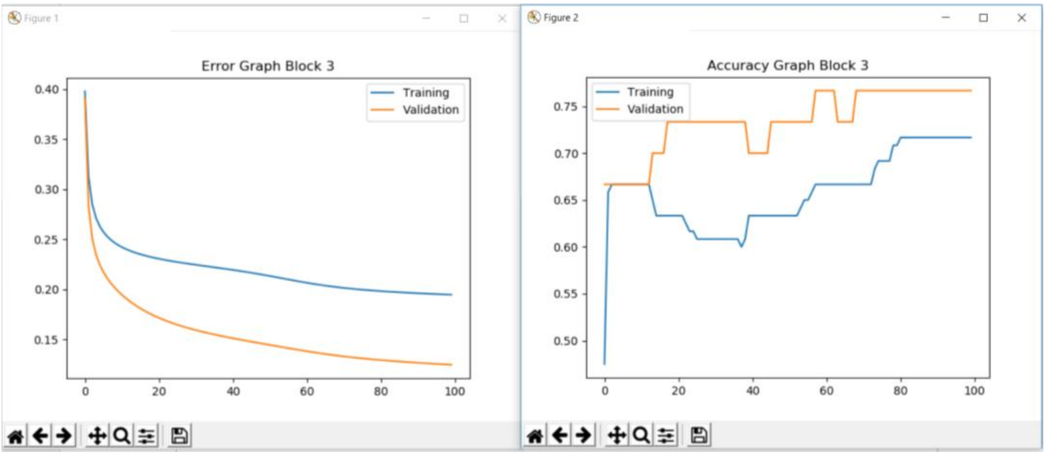


Figure 5.

Block 4:

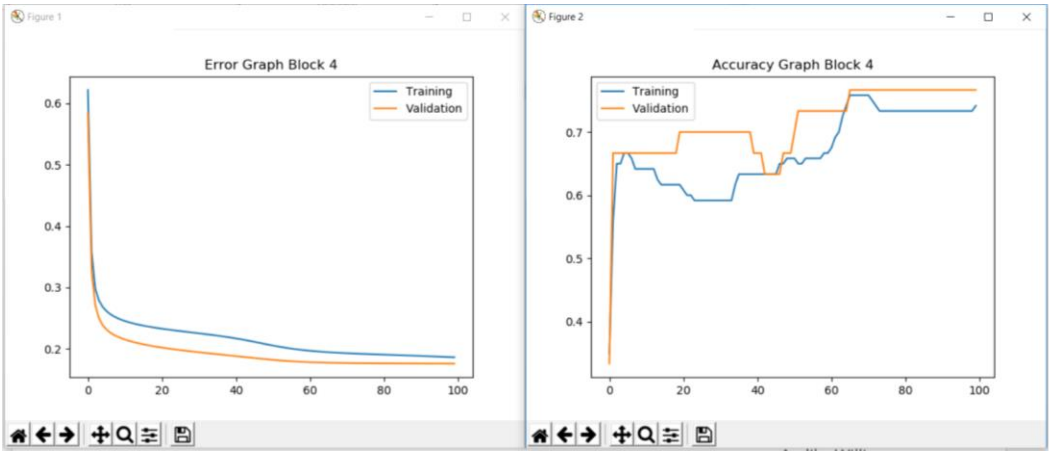


Figure 6.

Block 5:

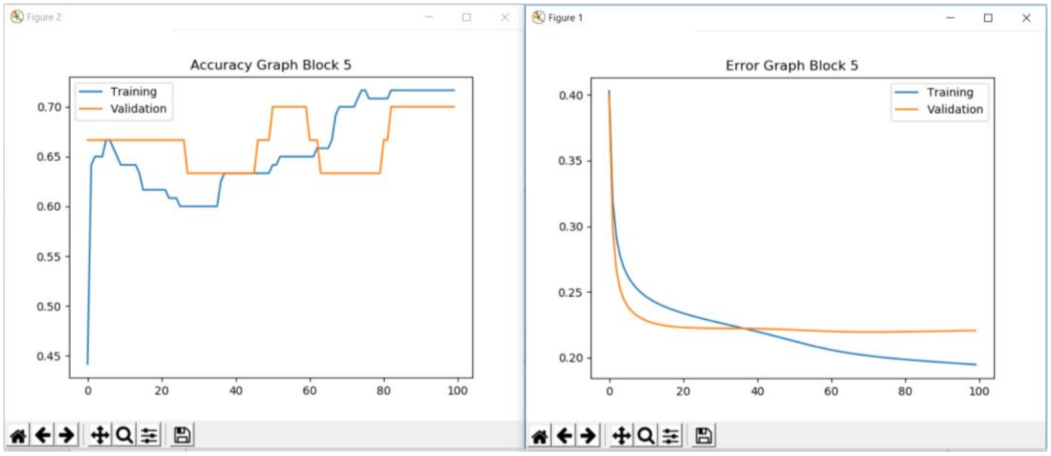


Figure 7.

AVERAGE:

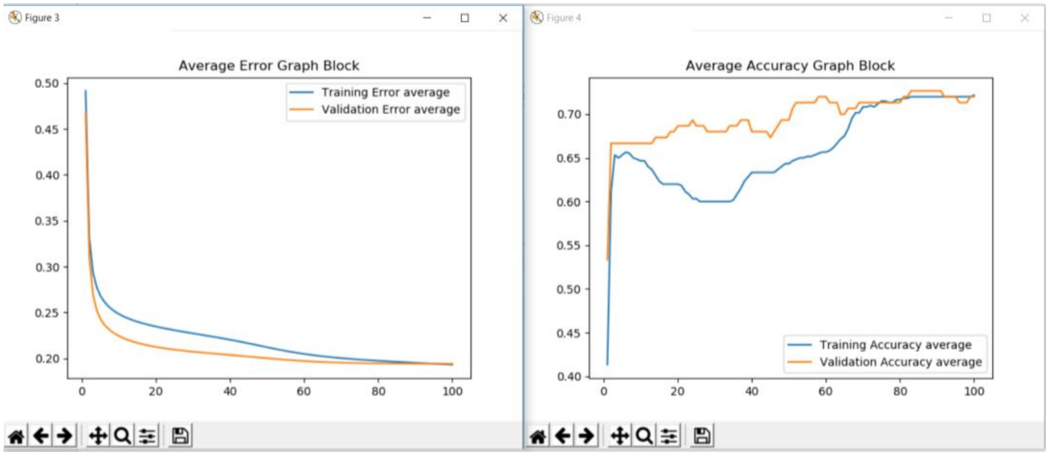


Figure 8.

Conclusion

Through our results, we can see the result of our K-Fold Cross validation. The average of all 5 graphs estimates the performance of our model when tested against a new data. As we can see in estimation the prediction accuracy converges at around 70 % of accuracy.