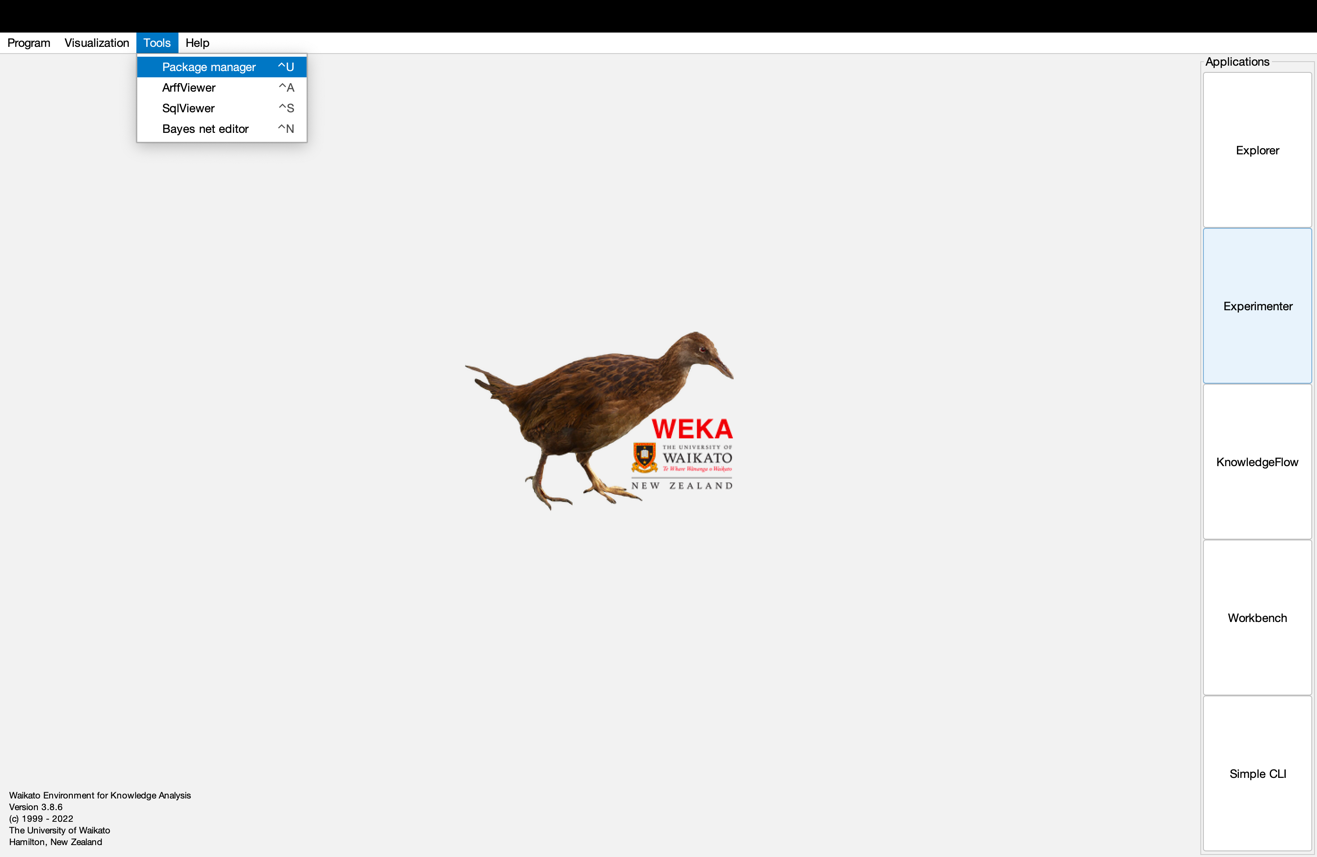
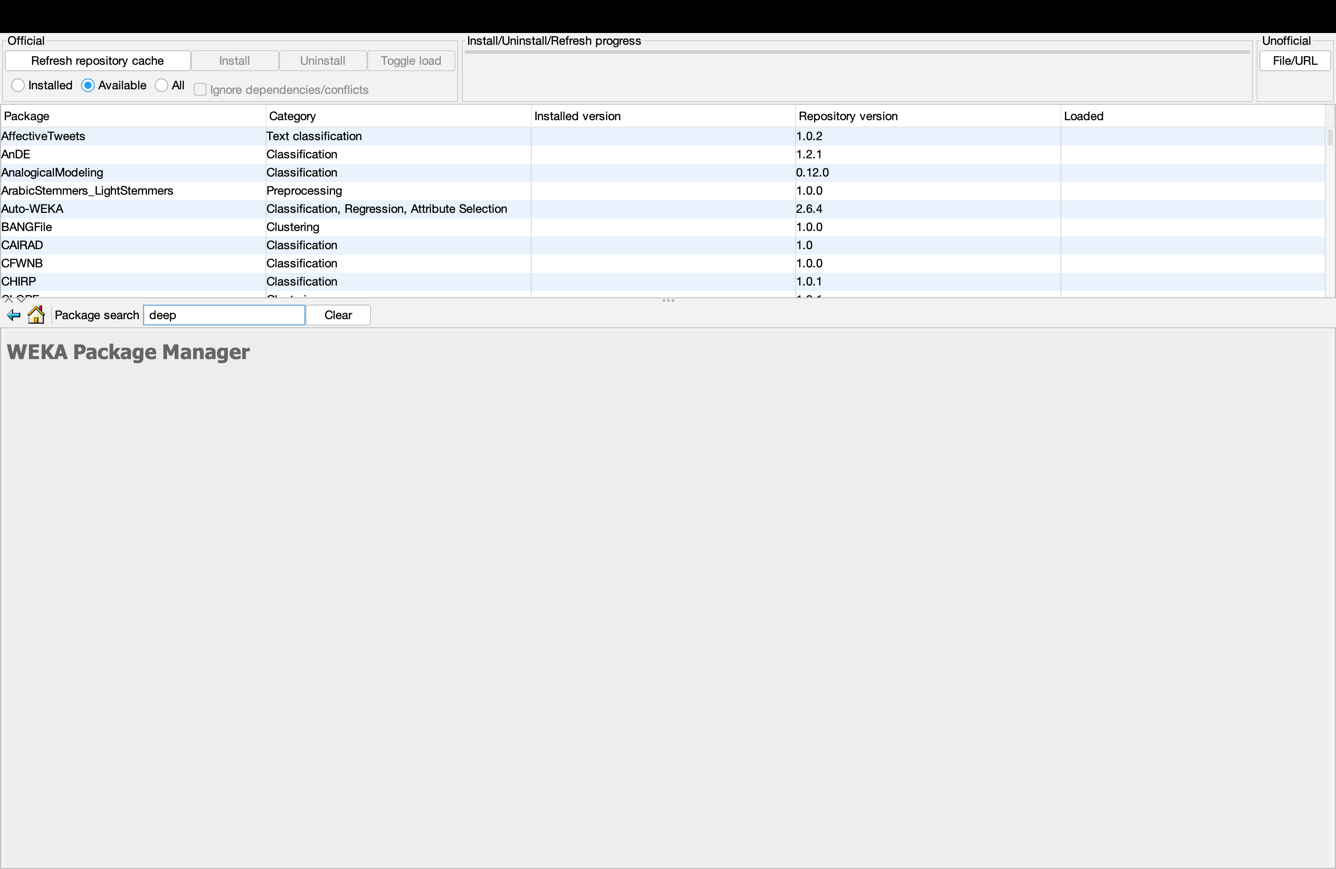
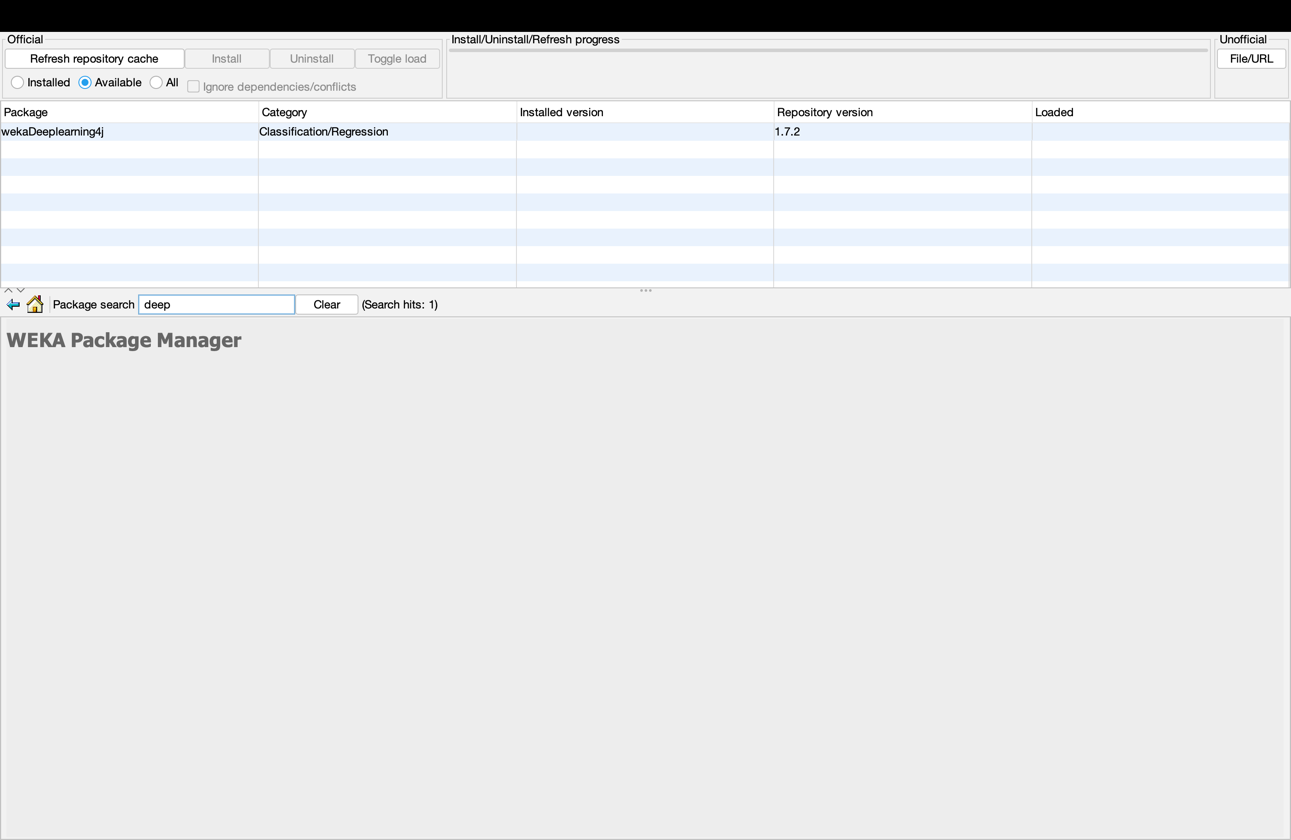
Tool -> Package manager



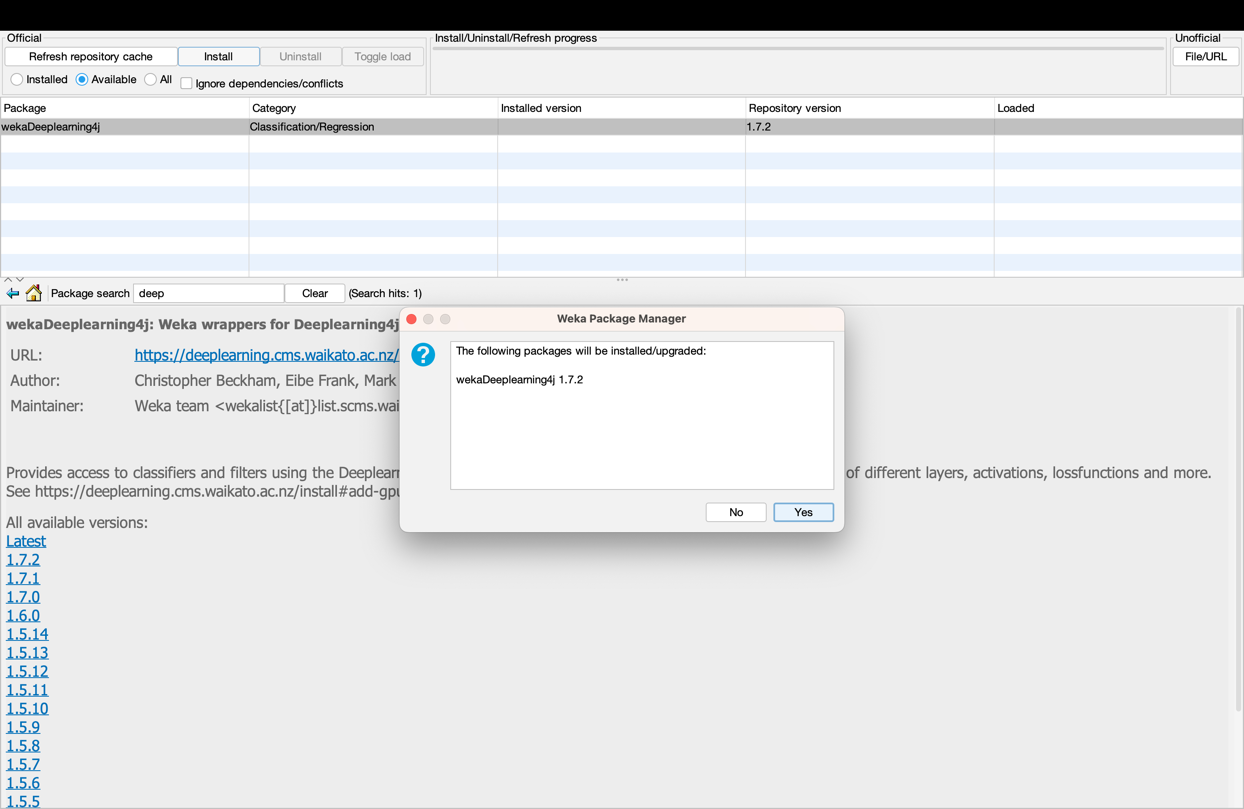
Package search box, typing “deep”

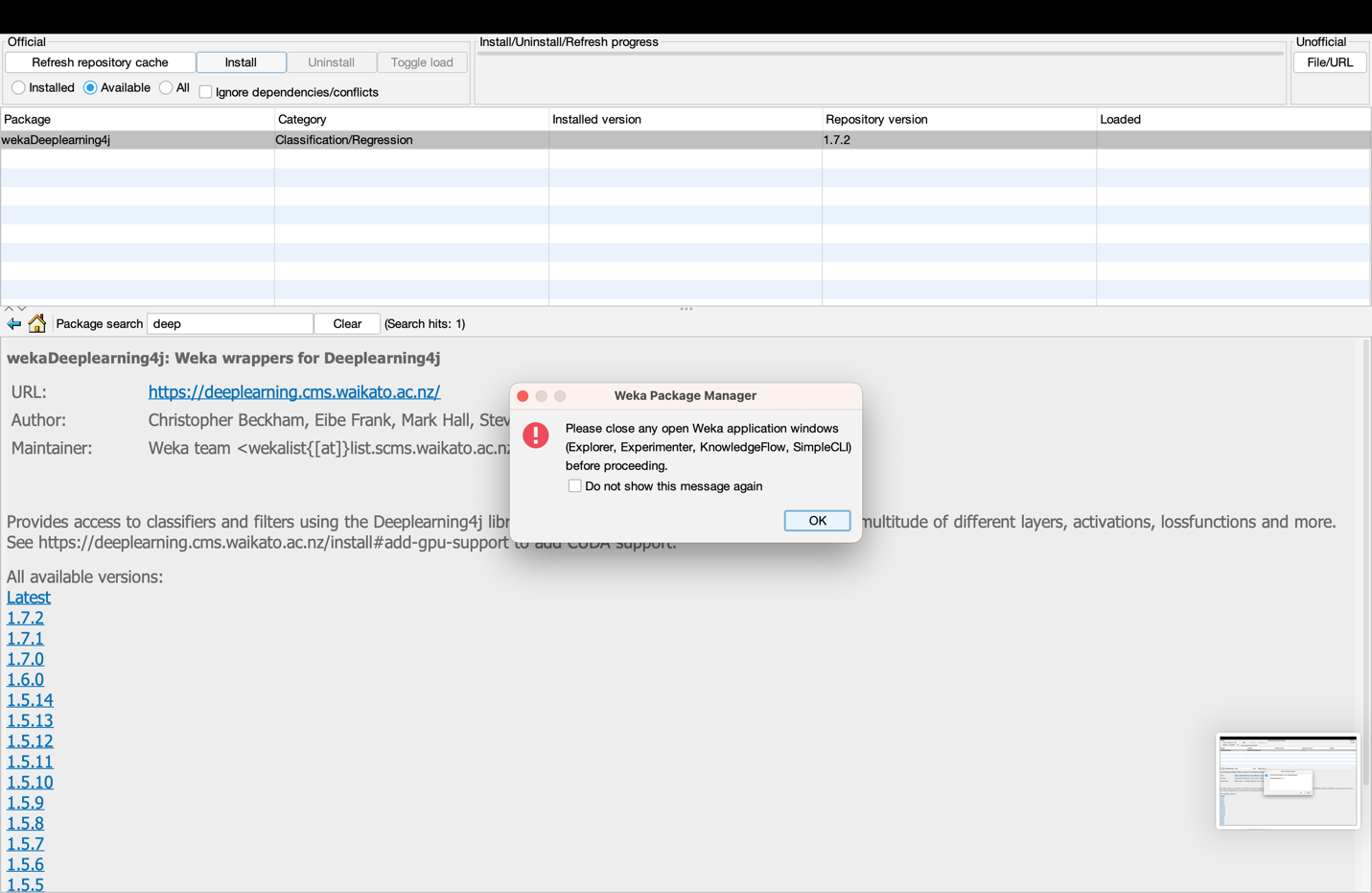


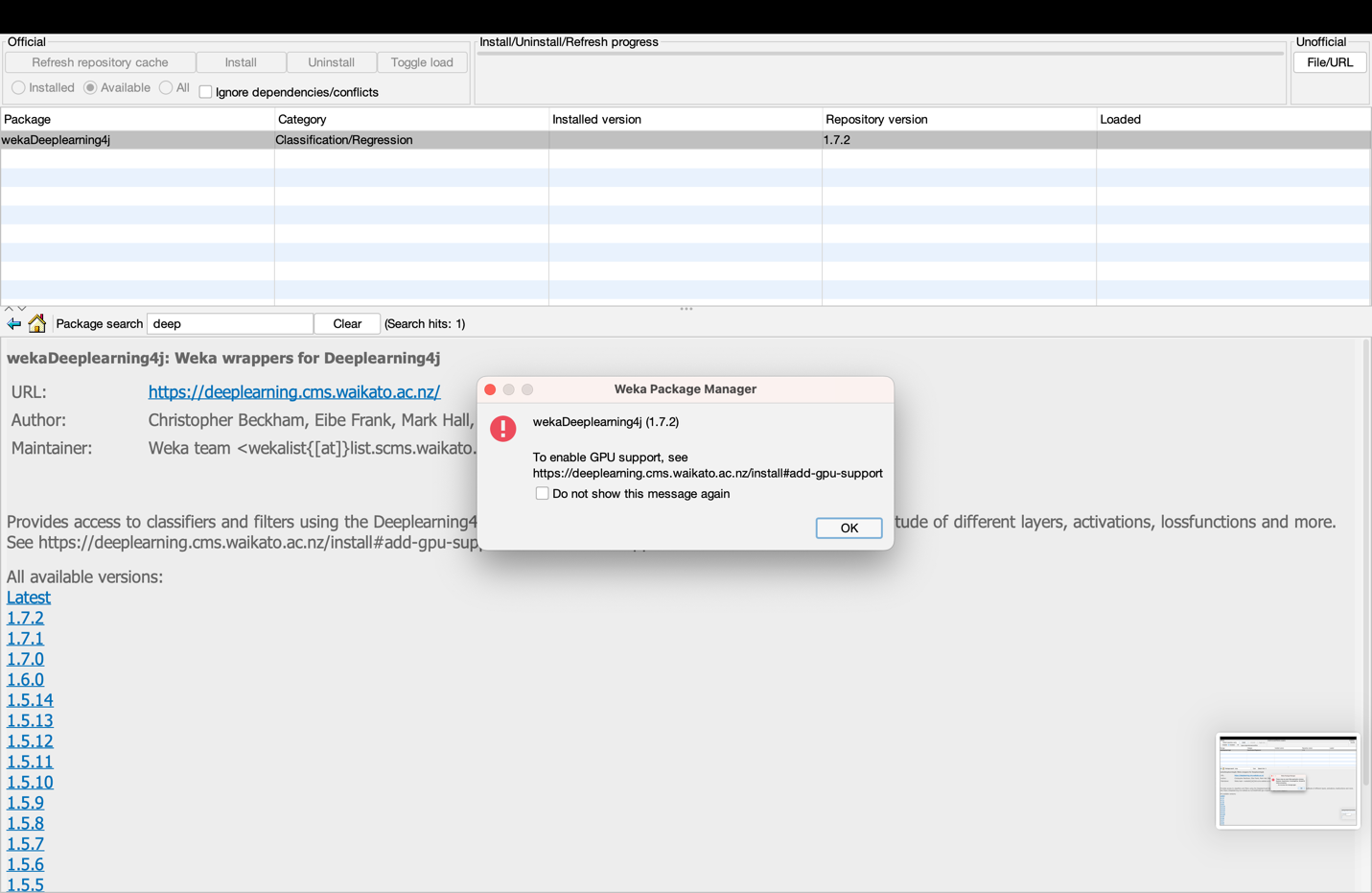
Installing “wekaDeeplearning4j”



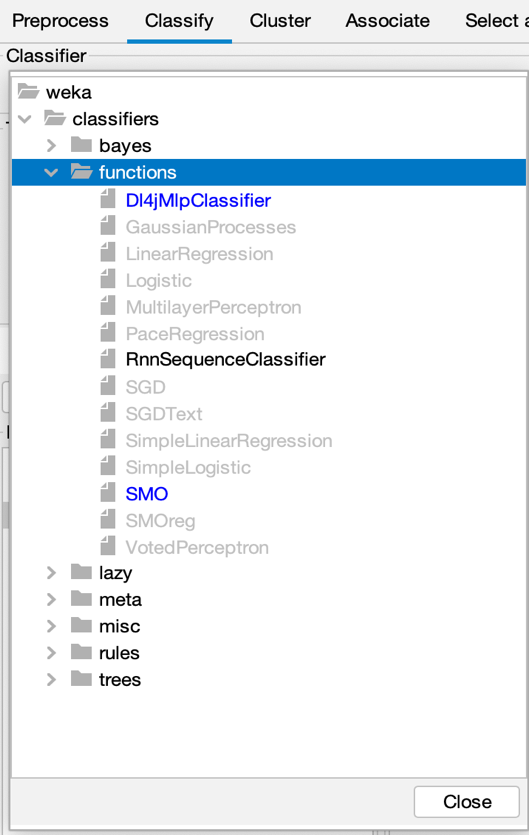
Confirmation boxes



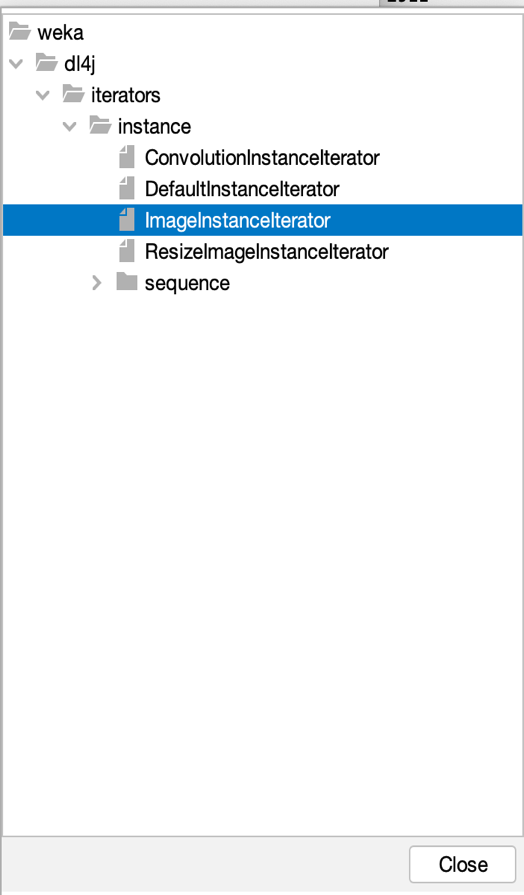




weka🡪classifiers🡪functions🡪Dl4jMlpClassifier



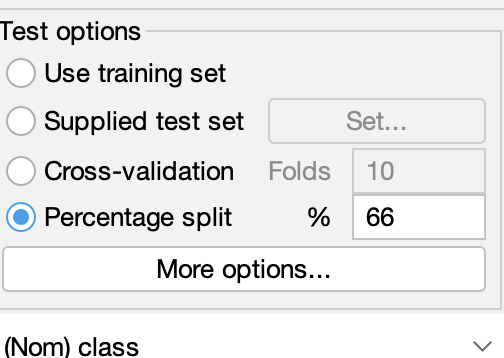
“Choose” next to instance iterator and click on ImageInstanceIterator



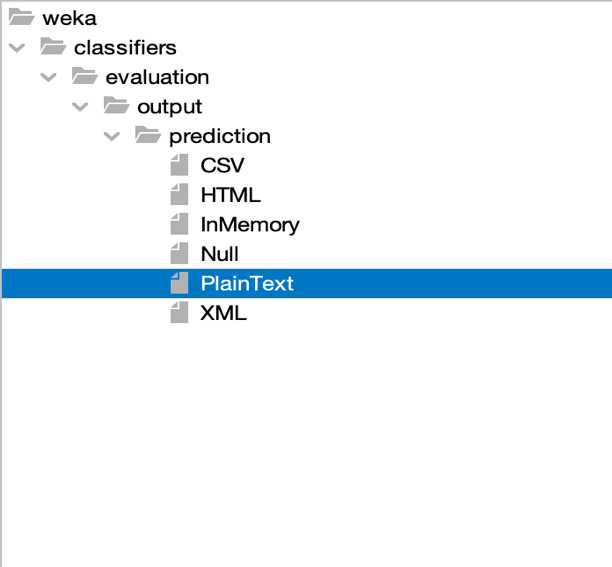
“Browse…” next to directory of images and find the folder mnist-3k with the JPG files



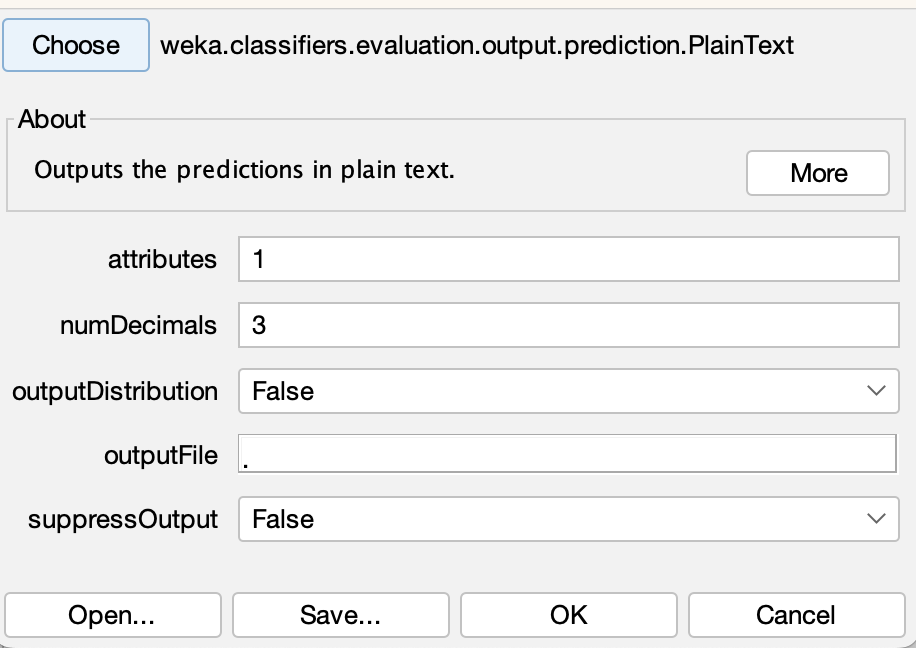
Choose “Percentage split”



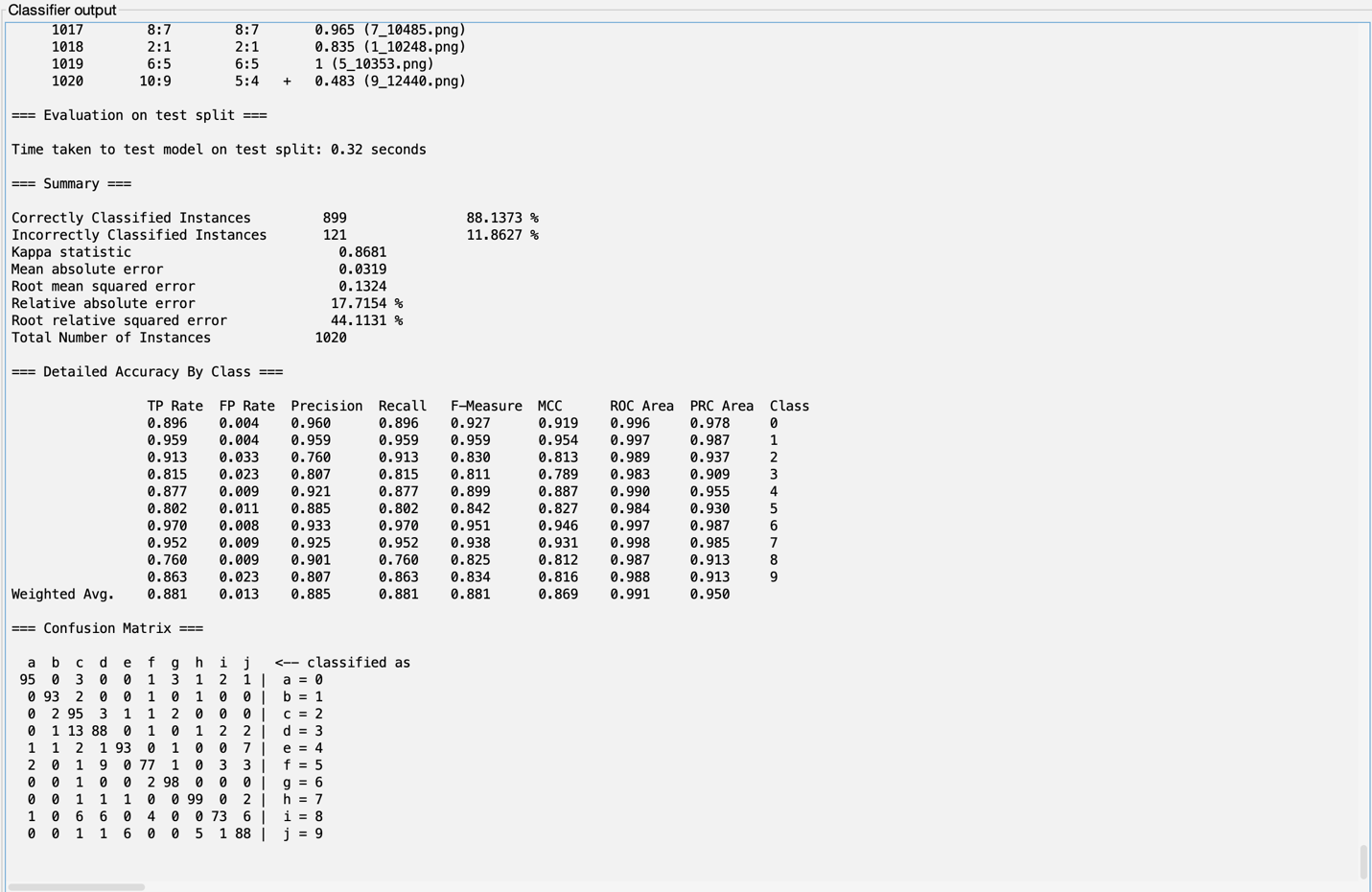
“Choose” next to Output predictions and then click PlainText



Entering 1 into the attributes of Output predictions of Plain Text



Outcomes of the model

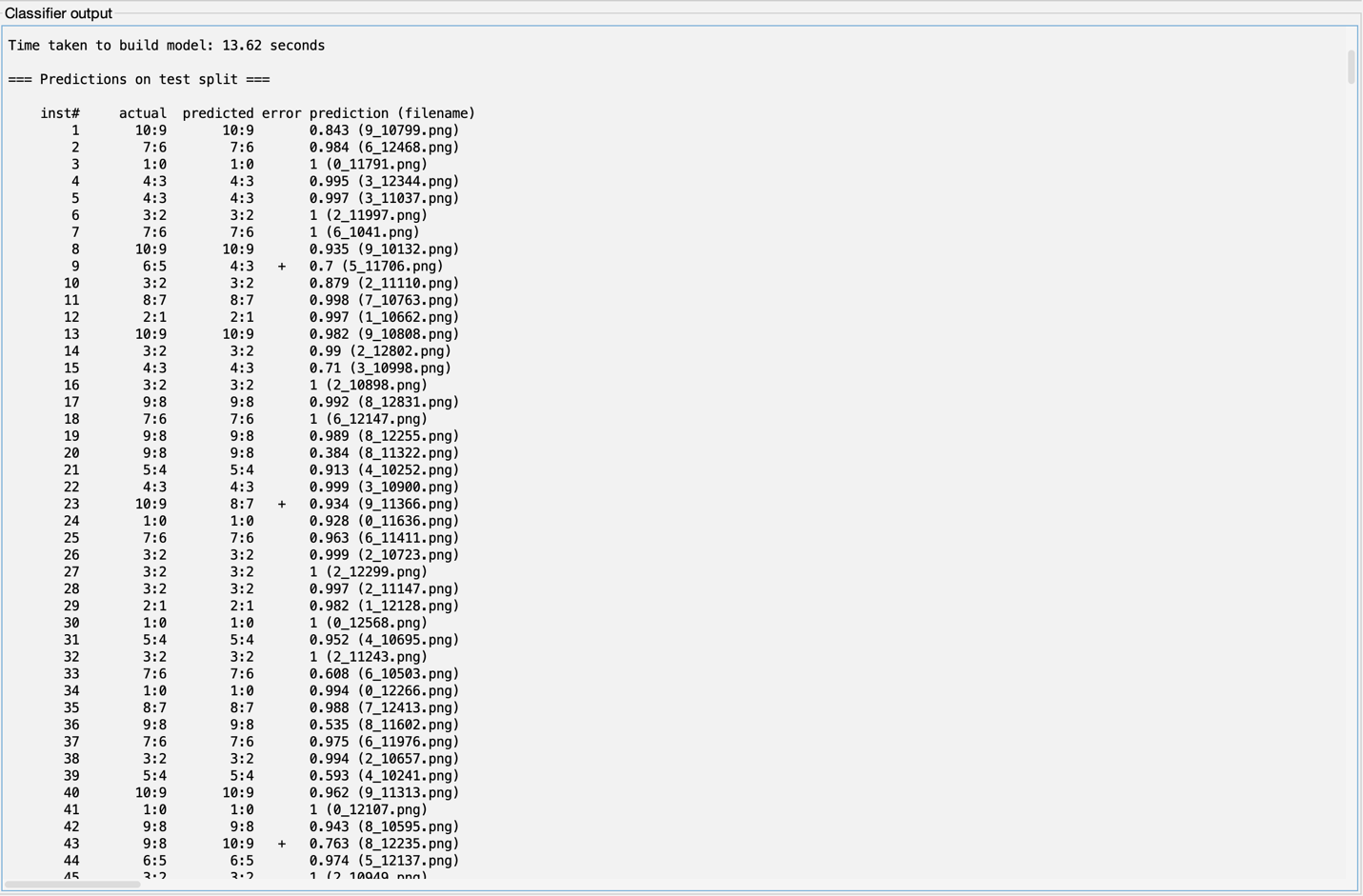


Question 1:

What is the accuracy of this initial model?

The accuracy of this initial model is in the range of 88.1373 percentage.

Time taken to build model



Question 2:

Copy three images that the classifier got wrong to your solution doc. Do these examples look like they would be difficult to classify (e.g., do they like someone might mistake them for a different number?)

Error 1

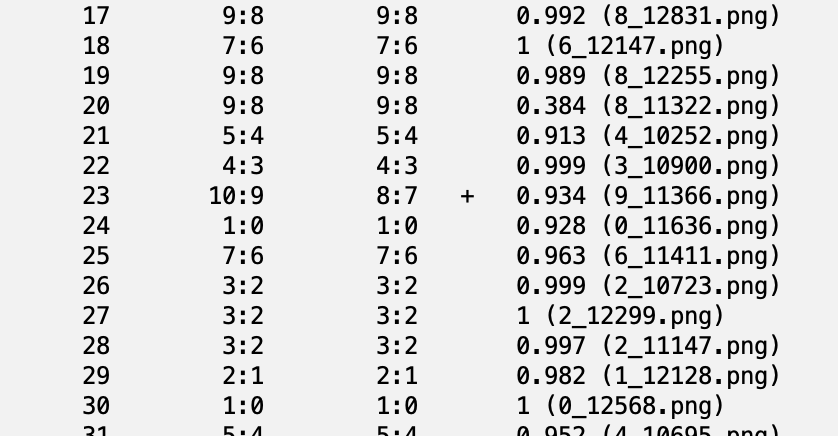
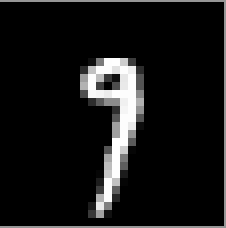


Image of the number for Error 1

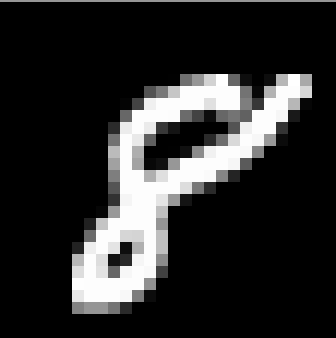


The algorithm classifies the image above as 7. But the actual number is 9. The image for number 23 can be easily classified as 9. I don’t think this example is difficult to classify. The algorithm messes up when trying to classified the image as 9 due to the size of the top of the 9 being tiny.

Error 2



Image of the number for Error 2

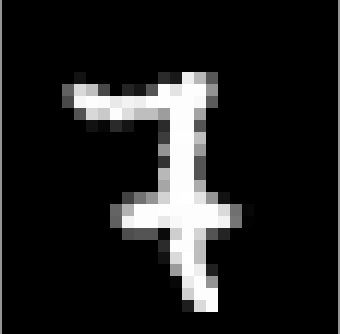


The algorithm classifies the image above as 9. But the actual number is 8. The image for number 43 can be easily classified as 8 but it is misleading due to indistinguishable gap at the top of the number. But still, it is not difficult to be classify as 8.

Error 3



Image of the number for Error 3

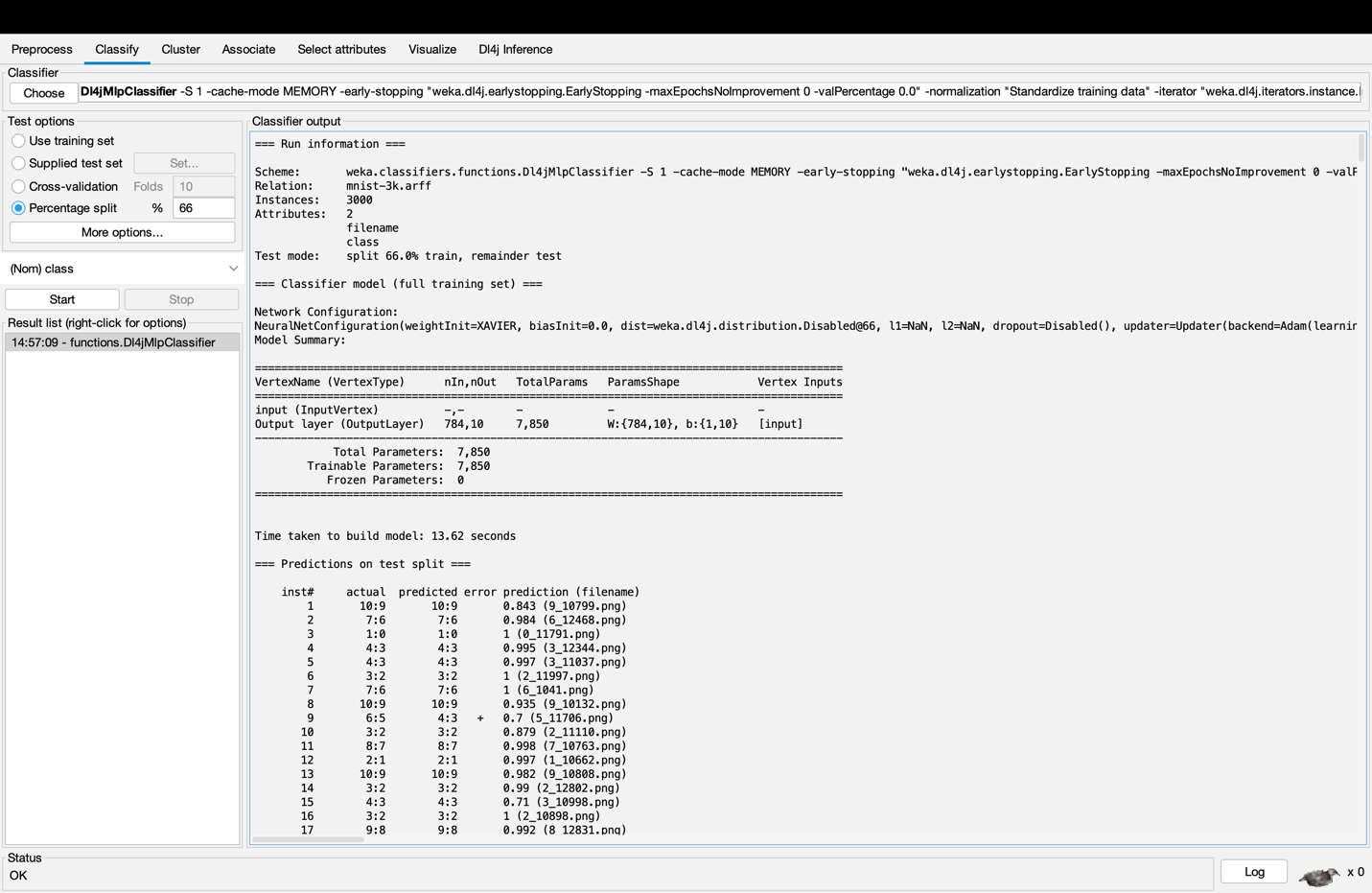


The algorithm classifies the image above as 2. But the actual number is 7. The image for number 23 can be easily classified as 7 by a human but there is a different in computer and human written way of 7. As we are human, we will understand that the image is a 7 even with a stick or line in the middle but for the algorithm to classify it, that line can be misleading.

Question 3:

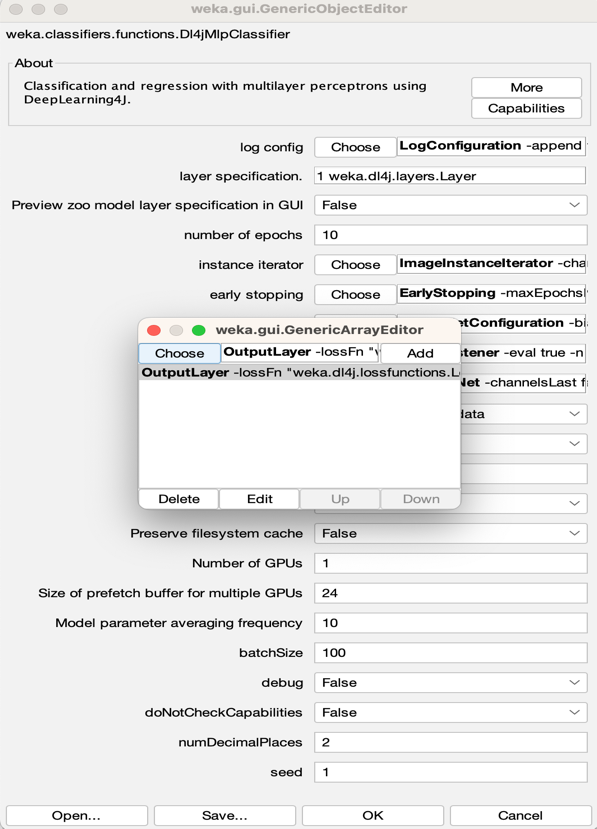
How long did it take to build the model?

Image of time taken to build the model

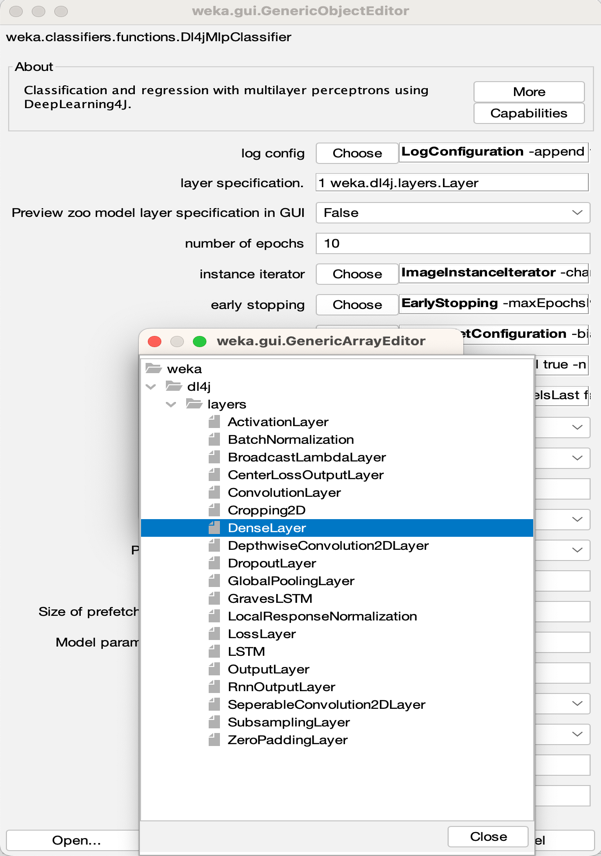


It took 13.62 seconds to build the model.

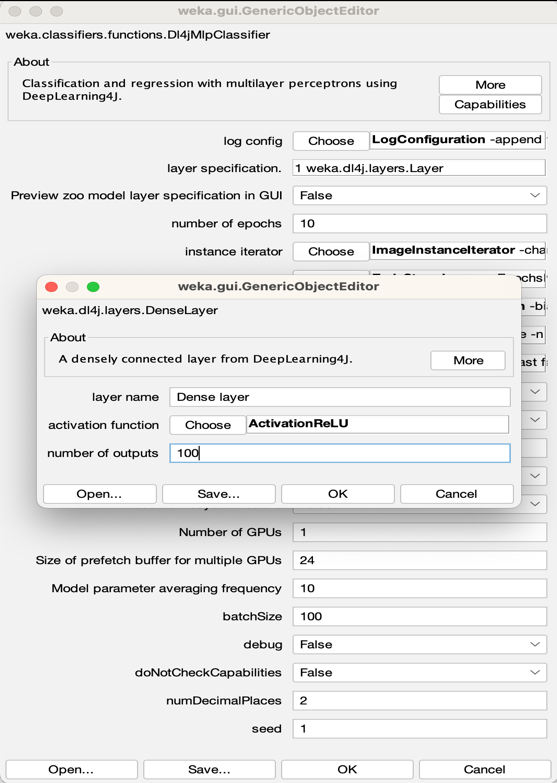
Opening the layer specification editor



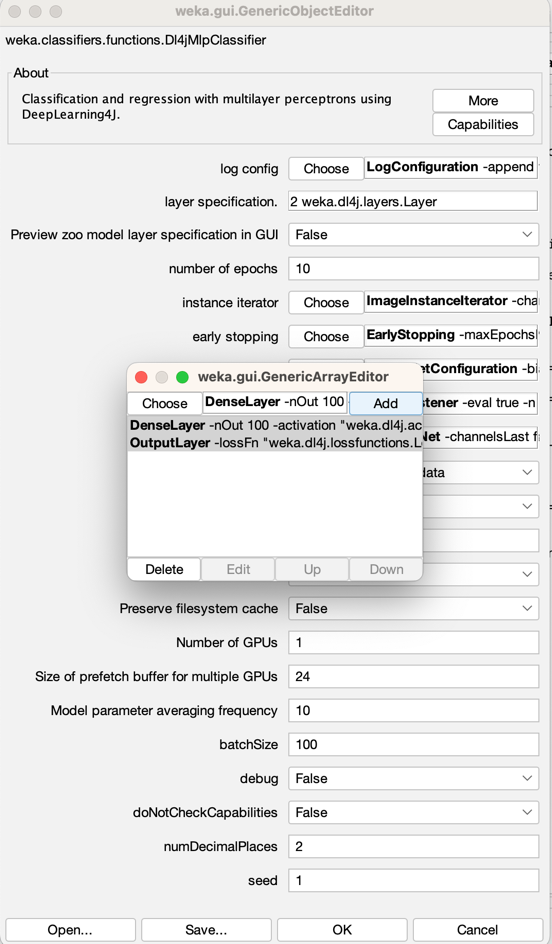
Click “Choose” and then click “DenseLayer”



Setting number of outputs to 100



Adding “DenseLayer”



Running the model and these are the result

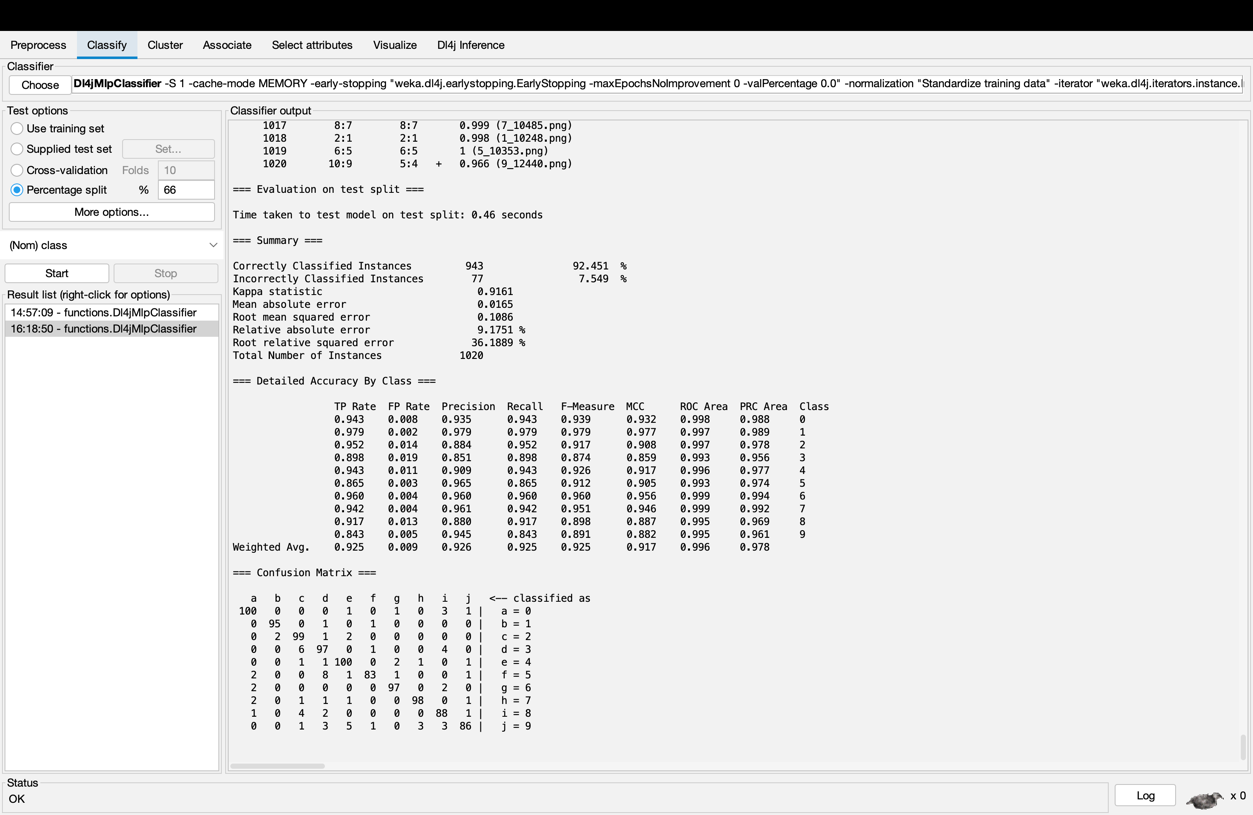
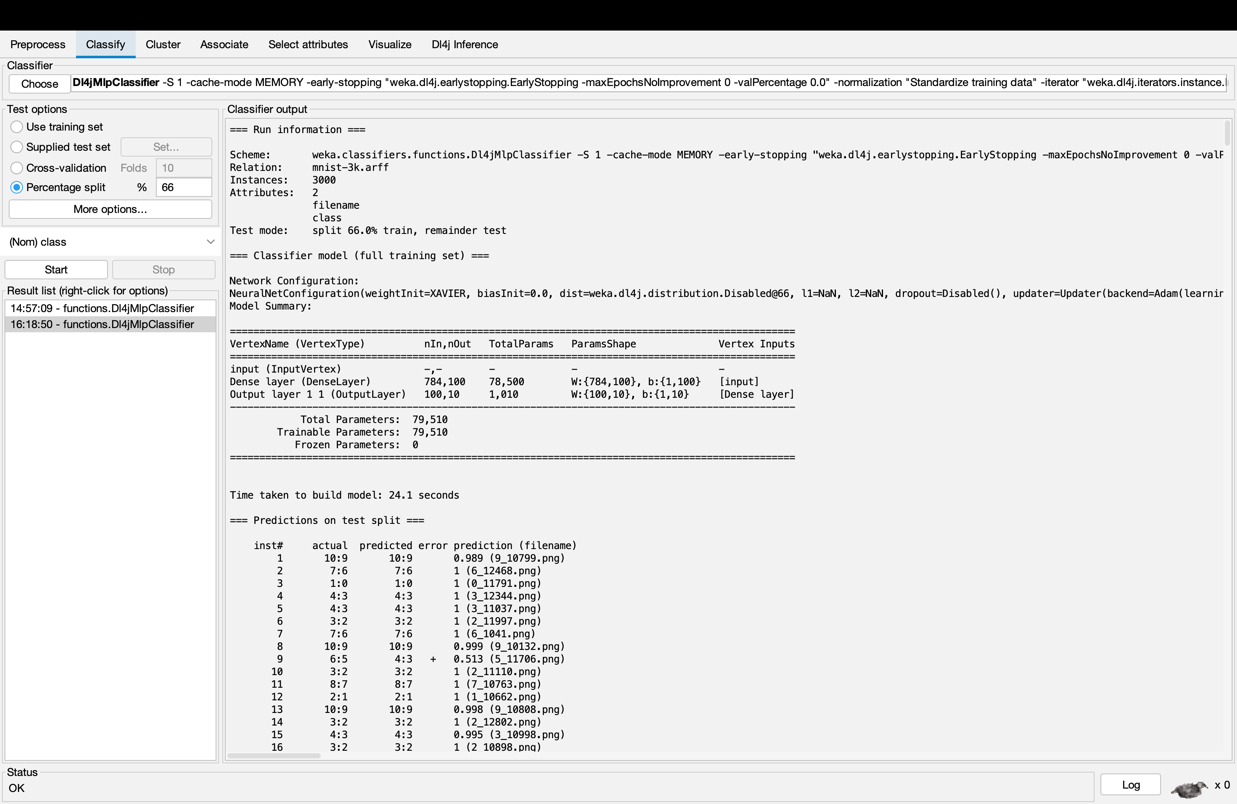


Image of time taken to build model

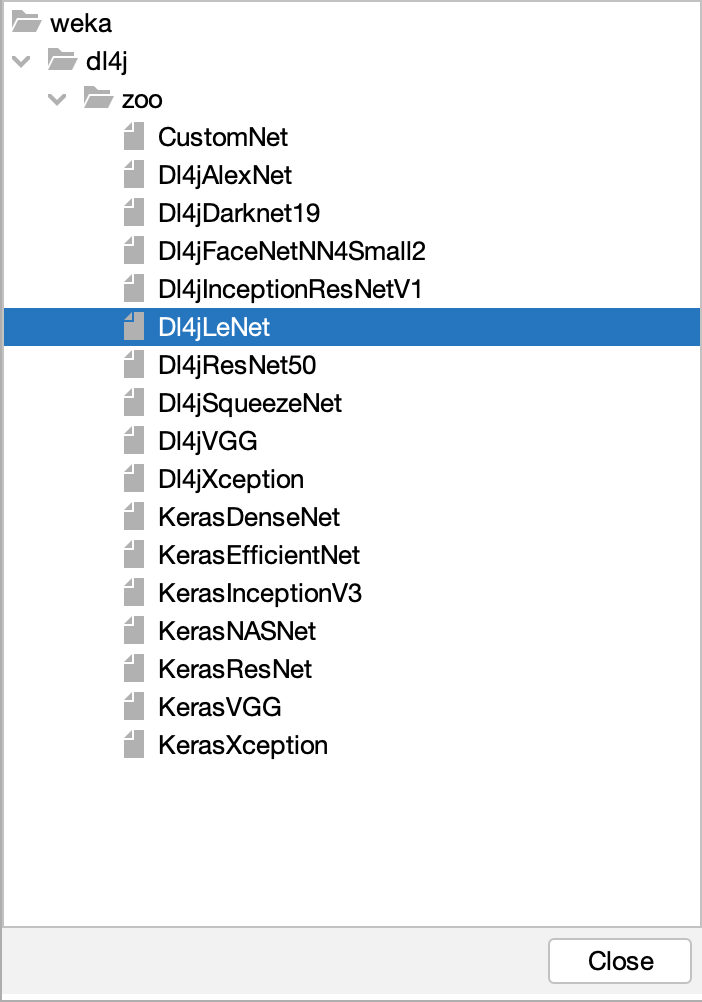


Question 4:

Compared to the first classifier, was this more accurate, and how long did it take to train?

The accuracy rate of this model is in the range of 92.451 percentage. This show that the accuracy of the model increases when comparing to the first classifier. It took 24.1 seconds to build the model and 0.46 seconds to test the model.

Choosing the LeNet from zooModel



Running the model and these are the results

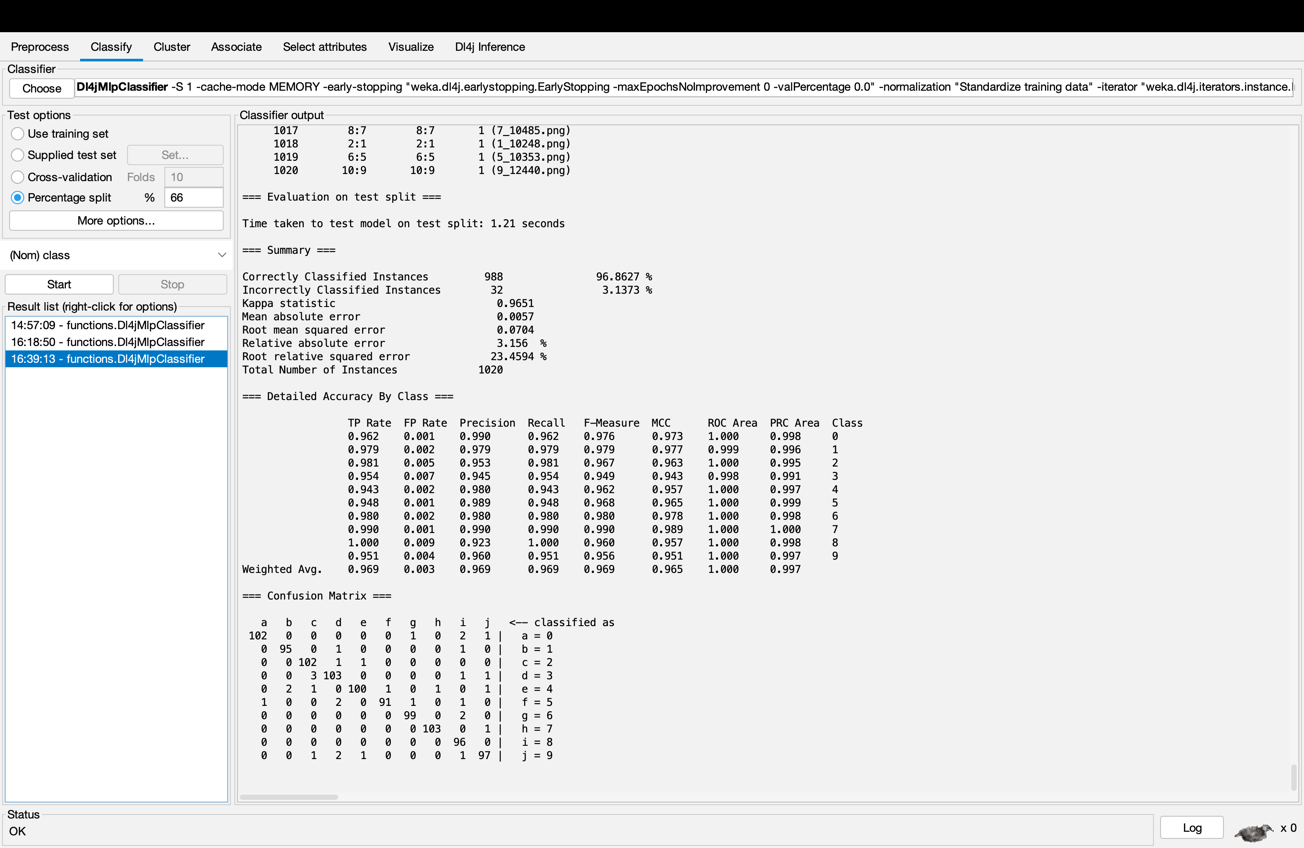
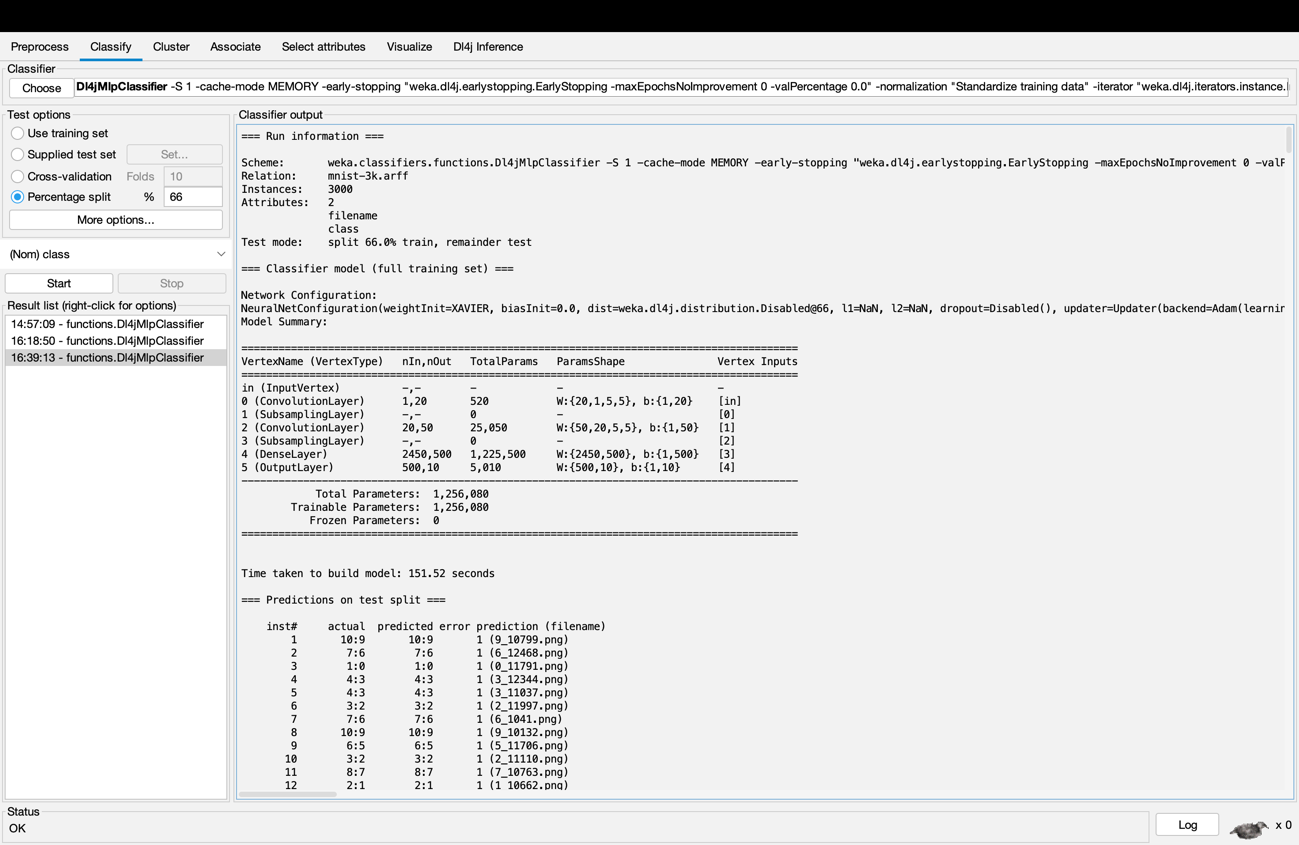


Image of time taken to build model



Question 5:

Pick some example images that the LeNet classifier got wrong. Do these look “trickier” to classify than the images that the original classifier failed on? What is the overall accuracy of this classifier?

Error 1

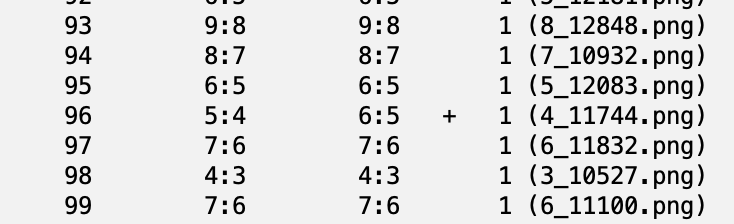
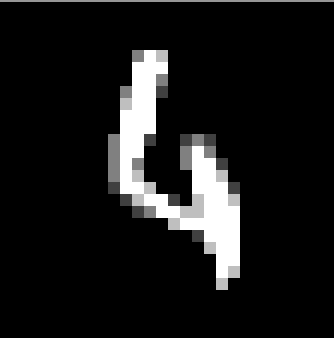


Image of the number for Error 1



The algorithm classifies the image above as 1. But the actual number is 4. The image for number 96 can be classified as 4 but unlike any other images, this image does not look a 4 but a human can classify this image as 4. For the algorithm to classify the image into the number, the difficulties increase when compare to human. Even though the model shows that it is 100 percent accurate, we can see that it is completely wrong.

Error 2

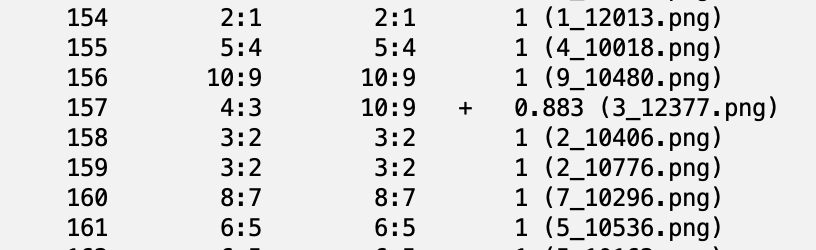


Image of the number for Error 2



The algorithm classifies the image above as 9. But the actual number is 3. The image for number 157 can be classified as 3. In the image the extra line from the top of the 3 seems to be touching the middle curve of 3. I believe that this incident causes the algorithm to classify as 9 rather than 3. Unlike other number, the number accuracy rate is not at 100.

Error 3

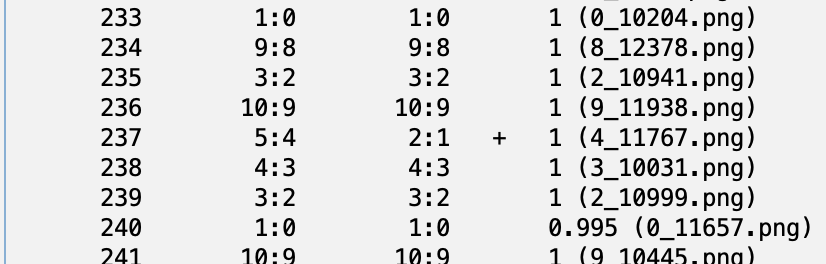
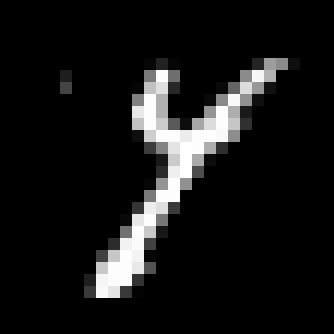


Image of the number for error 3



The algorithm classifies the image above as 1. But the actual number is 4. The image for number 237 can be classified as 4. This can be considered as a difficult image to classify due to the image looking like y. For the 4 in a computer, there is a connection at the top of the 4 but in the image, there is no connection. Therefore, the algorithm classifies as 1 and give it 100 percent accuracy although it is completely mistaken.