- **Questions**:
- Does the model perform as accurately as expected on your smartphone? List a few methods to improve the model's accuracy.

Model’s performance is overall good, but could be improved while testing using smart phone. I felt training accuracy could have improved by modifying layers in the neural network model. Adding dense layers / attention mechanism might increase the model performance.

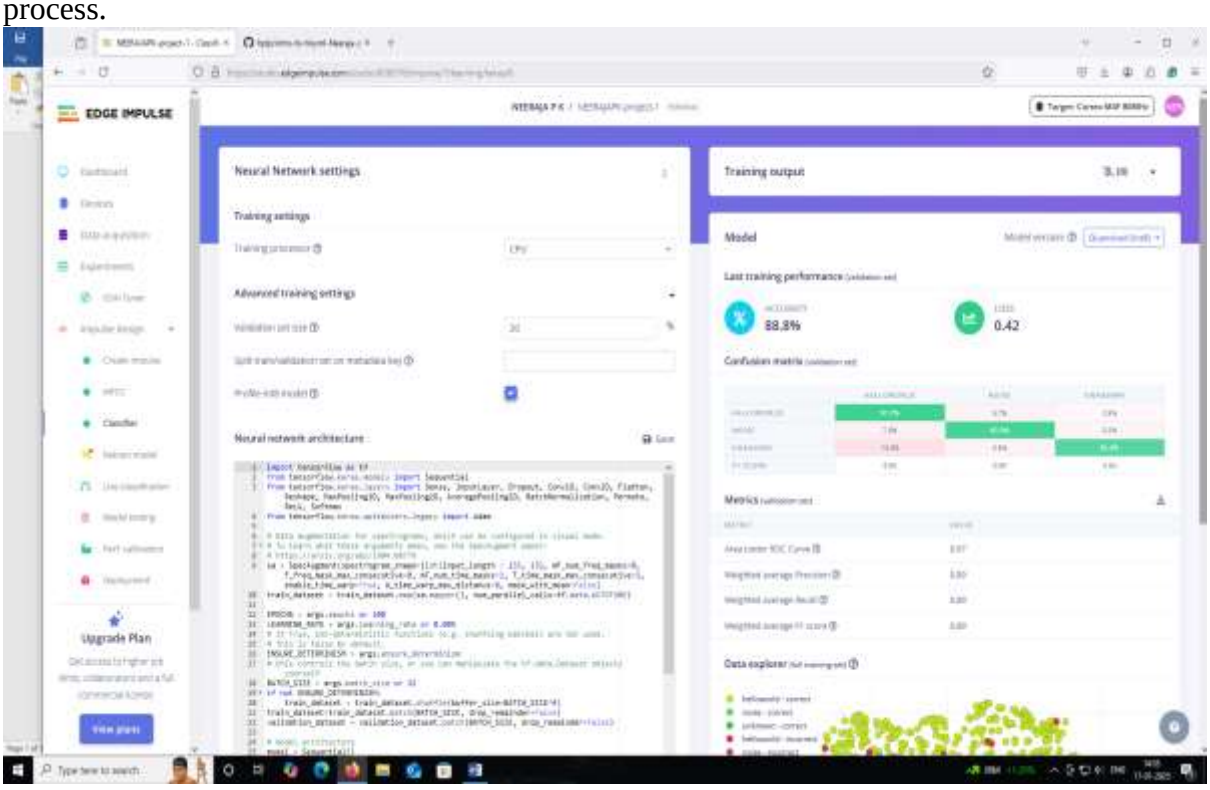
- When building a model for resource-limited hardware, how do you balance fast inference times with acceptable model accuracy? What trade-offs did you encounter?

Simplifying the model architecture (e.g., reducing the number of layers or neurons) helps achieve faster inference times but might reduce accuracy

Quantizing the model (reducing the precision of weights and activations, e.g., from 32-bit to 8-bit) can significantly reduce model size and improve inference speed, but it may result in a slight accuracy drop due to the lower precision.

To maintain model accuracy when simplifying the architecture, stronger data augmentation and regularization techniques (e.g., dropout, weight decay) can help, but they may increase training time and complexity. Using regularization techniques like dropout to prevent overfitting, while applying data augmentation to improve generalization without increasing model size

- Include screenshots of the training performance from **step 6** of the deployment process.



- **Reflections**:
- Share your experience deploying the model to your smartphone and Arduino board. Mention any technical difficulties or interesting observations.

We could only deploy the model to a smartphone and tested the recording, and the model was classified as expected. The technical difficulty arises because of the unavailability of the Arduino board (specific) as we are online participants. But overall, it was a great hands-on experiment. Really enjoyed doing this Mini-ML with the help of Edge Impulse.