**PNEUMONIA CLASSIFICATION IN CHEST X\_RAY USING FEDERATED LEARNING**

**The Challenge**

To build an algorithm that automatically identify whether a patient is suffering from pneumonia or not by looking at chest X-ray images. The algorithm had to be extremely accurate because lives of people is at stake.

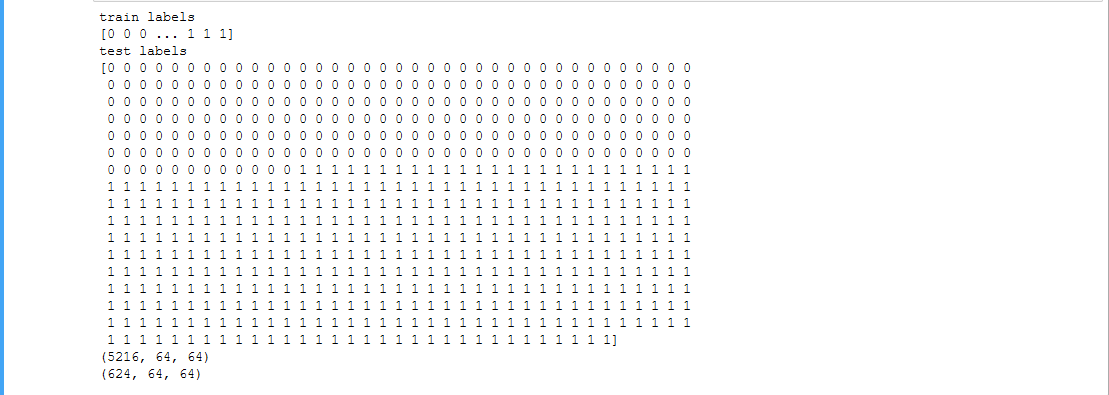
**Environment and tools**

* Anaconda
* Jupyter
* Keras
* Tensorflow backend
* Numpy

**Code**

Converter for datasets to make it compatible for IBM FL to run it

OUTPUT:

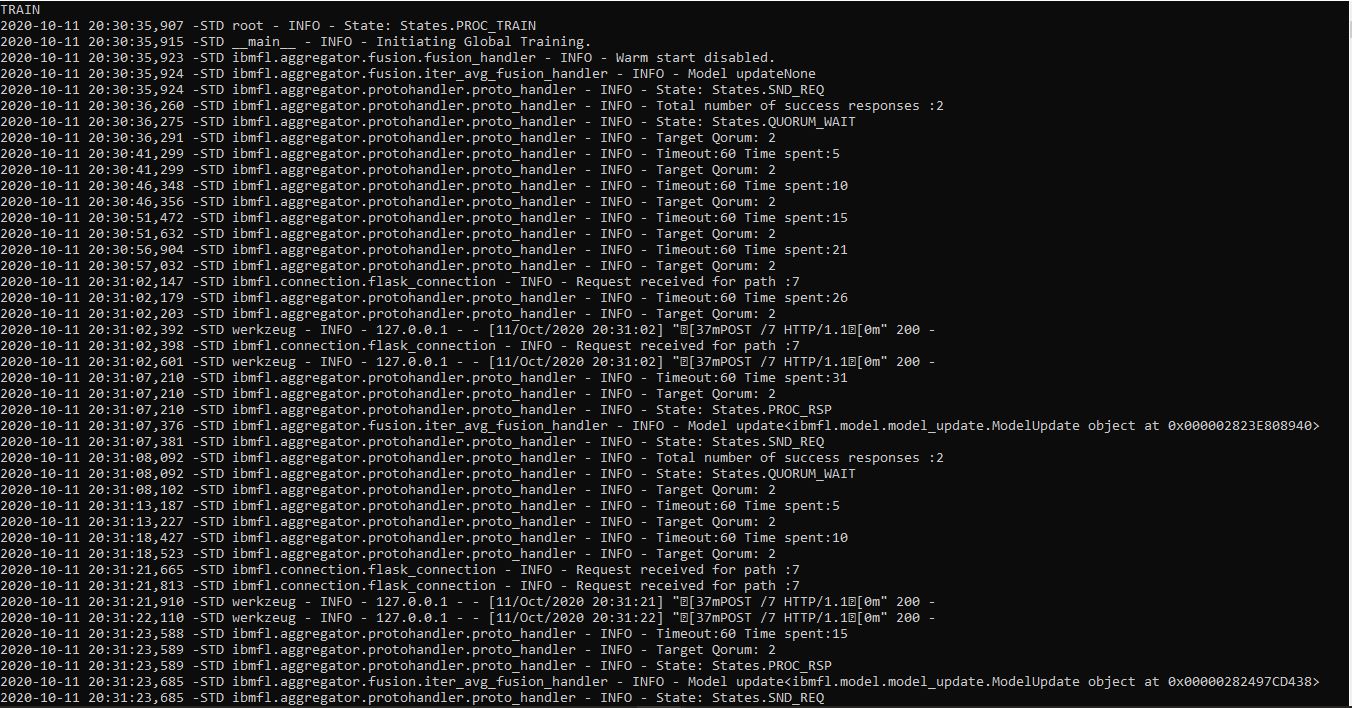
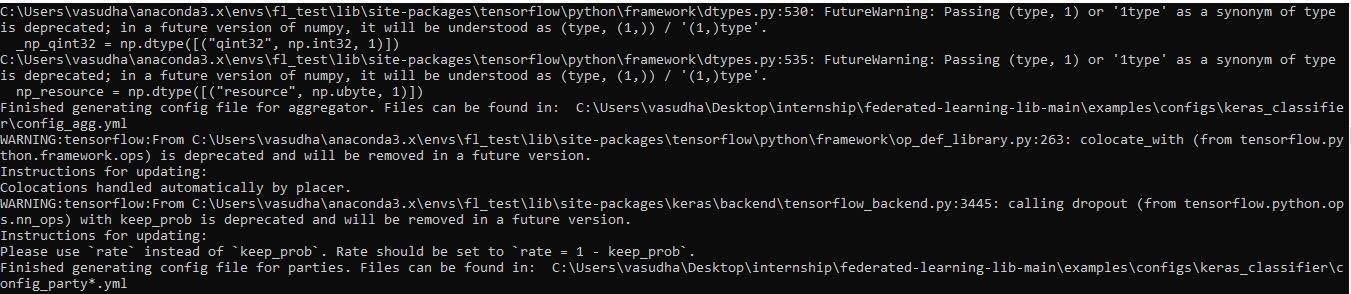
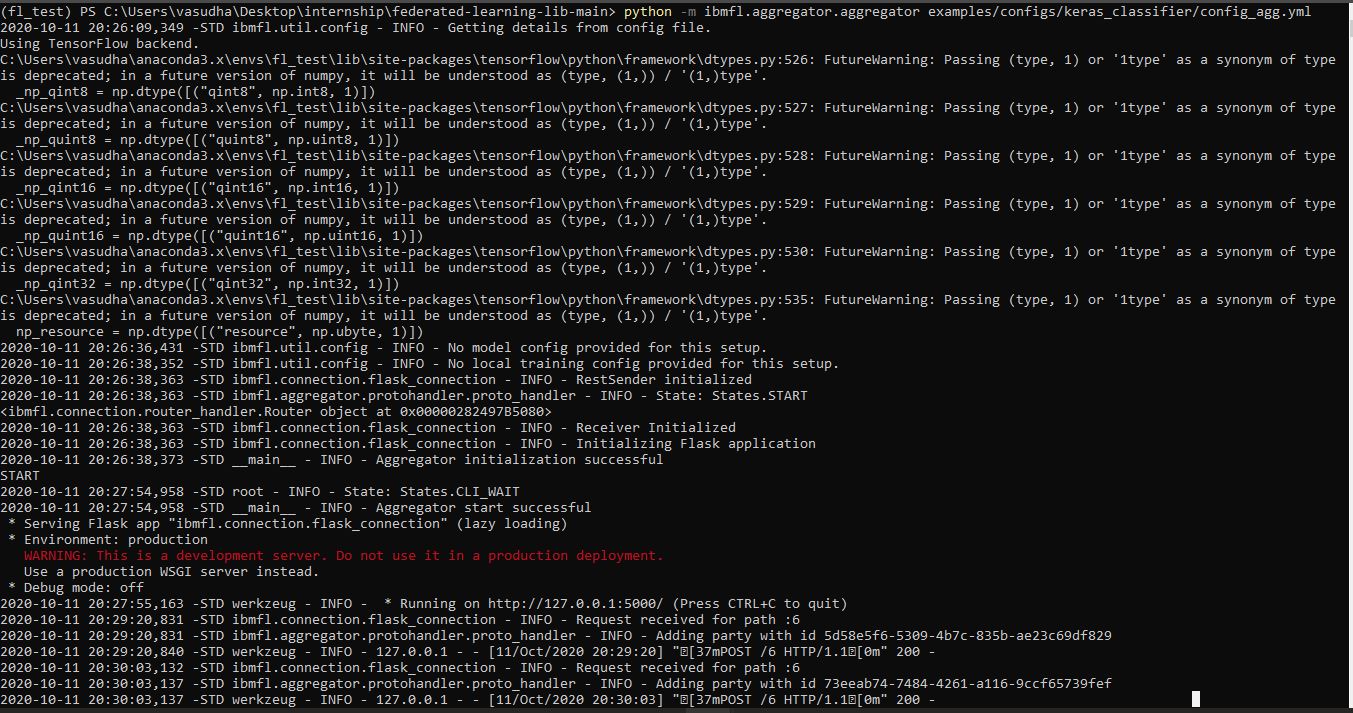
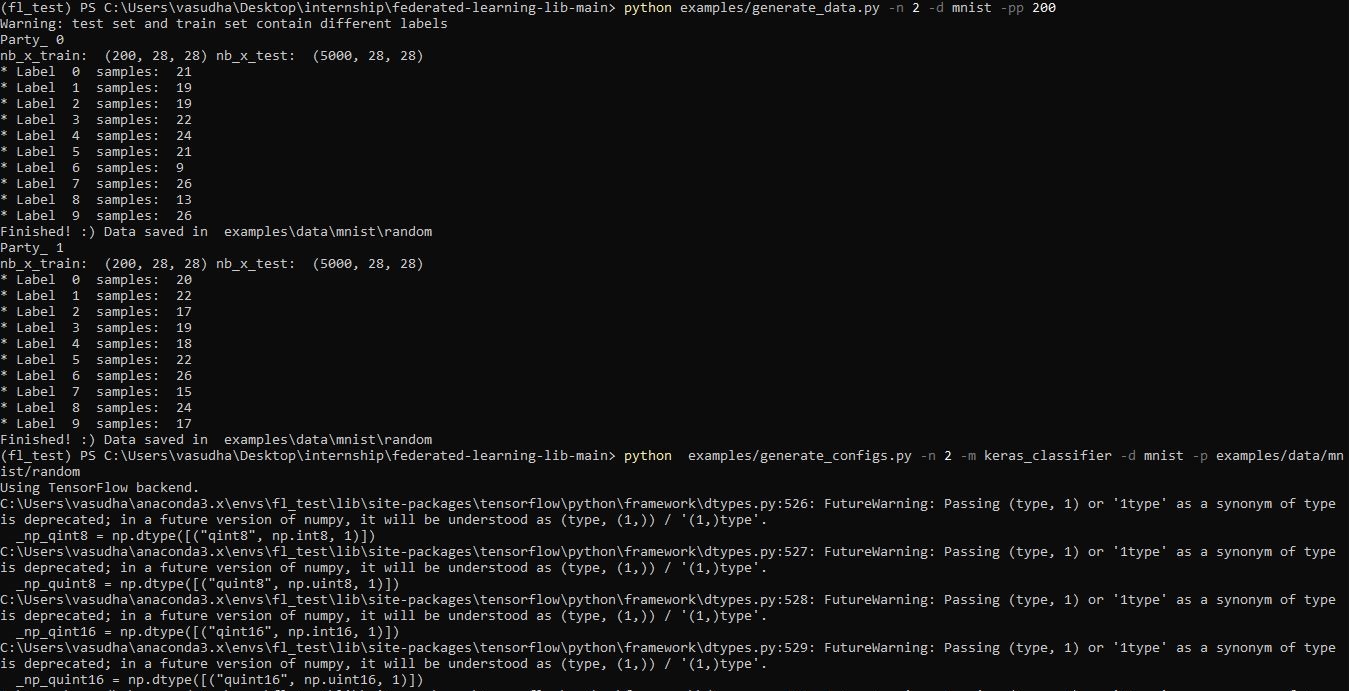


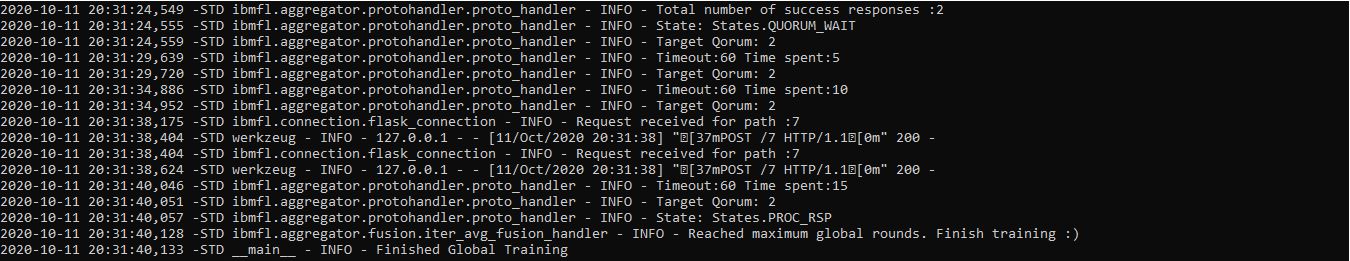
**Working with IBM FL**

**Steps to be followed:**

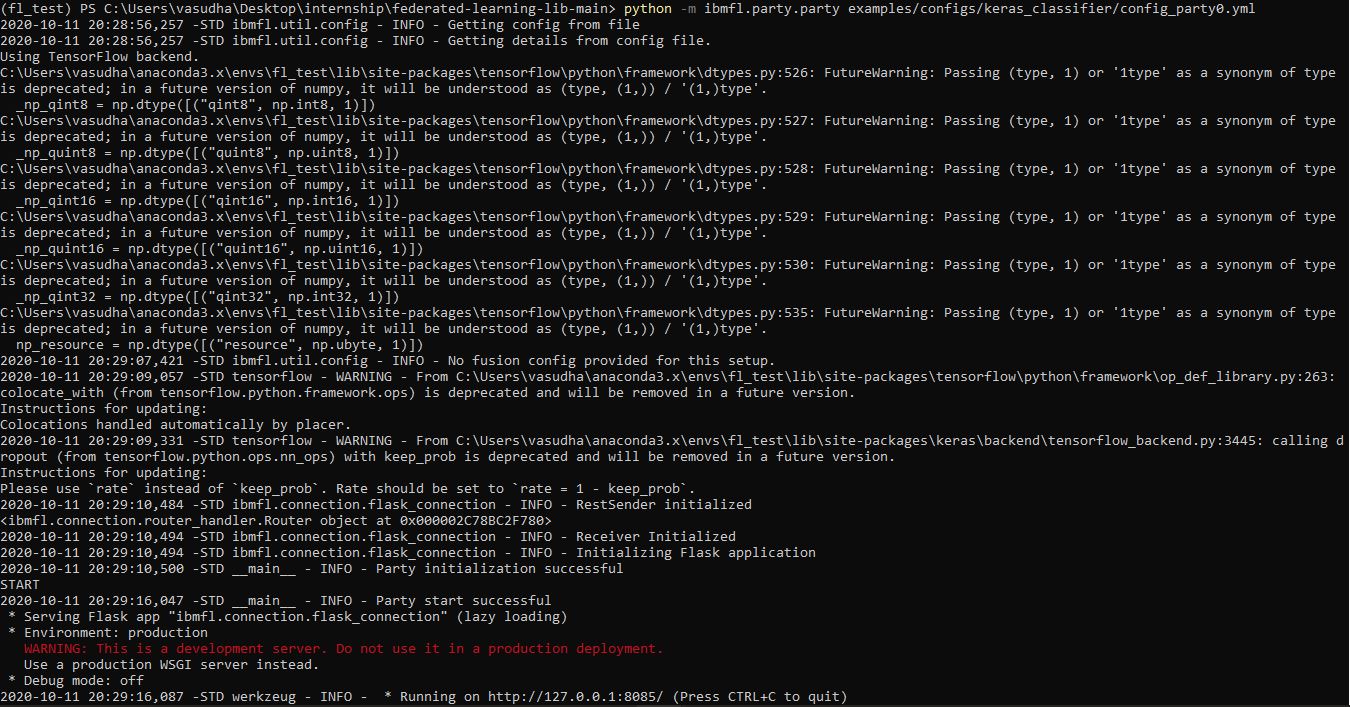
* Trained a Keras CNN model on MNIST.
* Created a new conda environment for ibm federated learning by running command lines.
* Prepared datasets for each participating parties. Generated 2 parties with 200 data points .Specified the machine learning model to be trained and generated the configuration files necessary to train a “Keras\_classifier” model assuming 2 parties join the federated learning training.
* Hence generated 2 parties in example using the ‘mnist’ dataset and ‘examples/data/mnist/random’ as our data path.
* Started aggregator by opening a terminal window running the IBM federated learning environment. Started and registered a new party , running the IBM federated learning environment set up beforehand in the same directory. Then the aggregator terminal will prompt out INFO that it received the party’s registration message.
* Then initiated training from the aggregator, had 2 parties joining the training
* Modified generate\_data and generate\_configs scripts from the example directory to use our dataset, data handler and fl\_object.

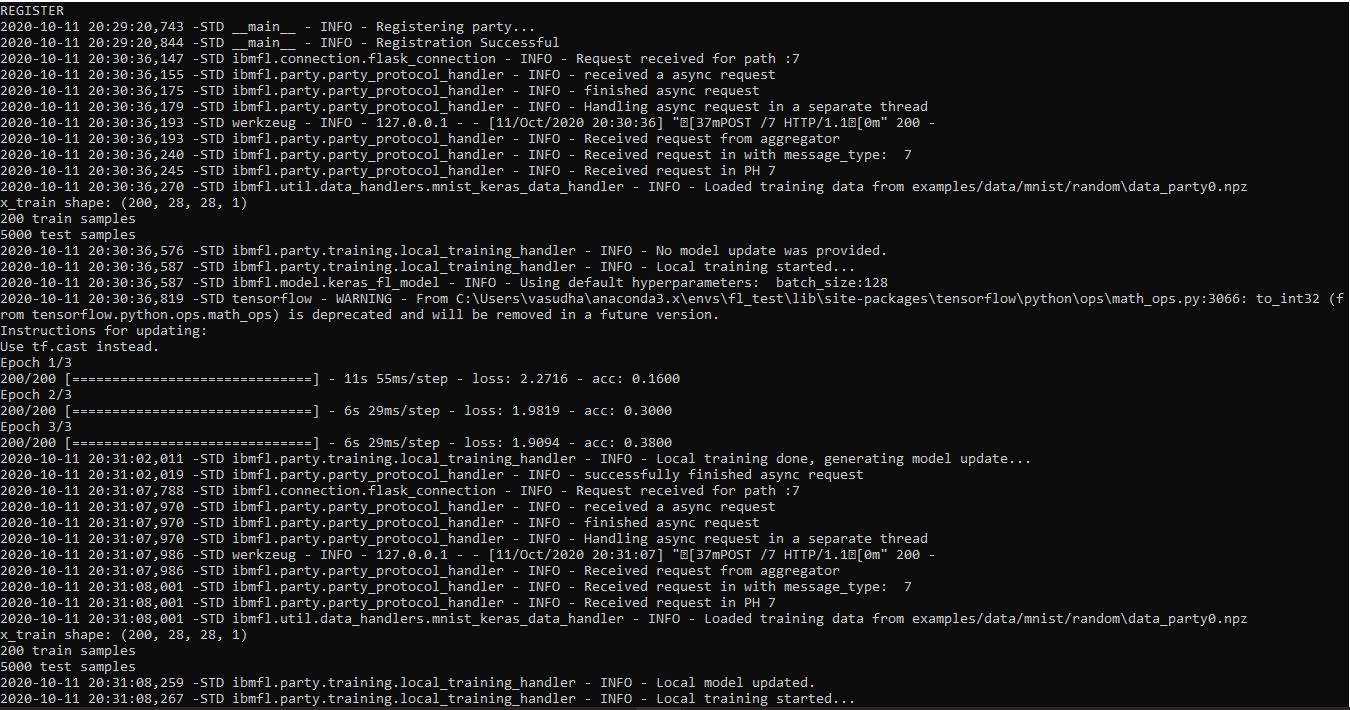
**Aggregator And Parties**

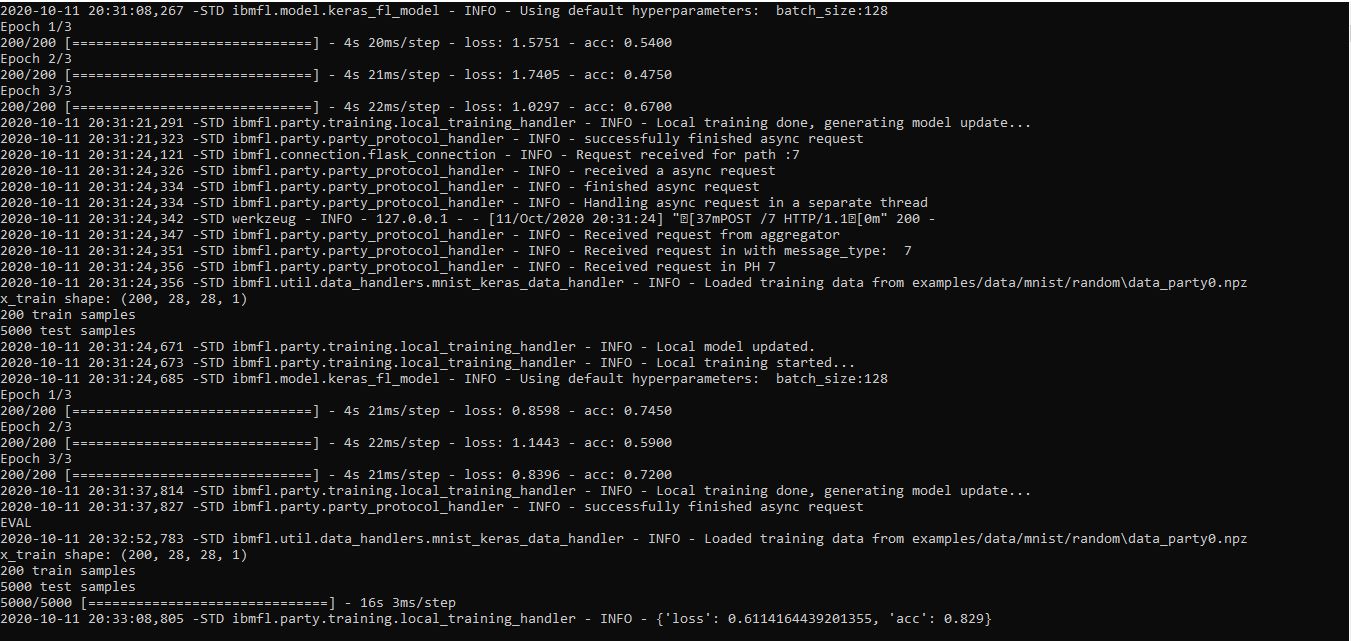
Aggregator

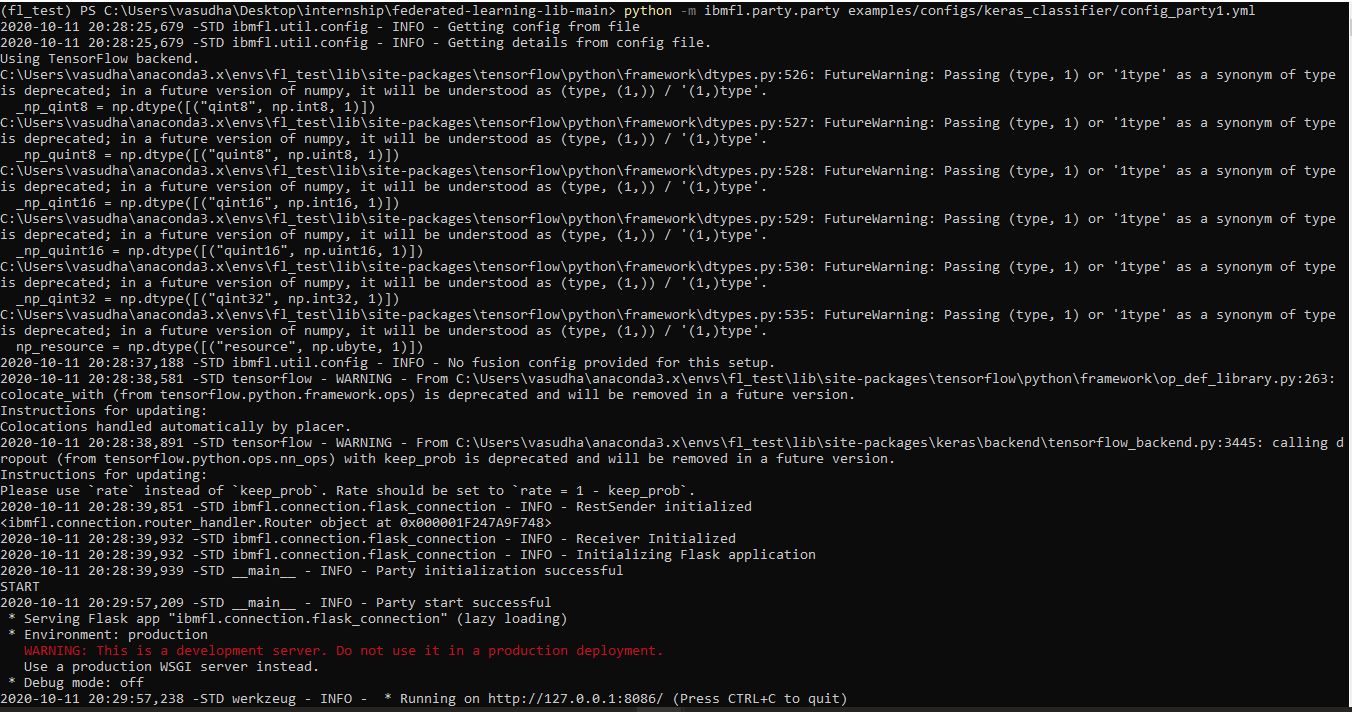


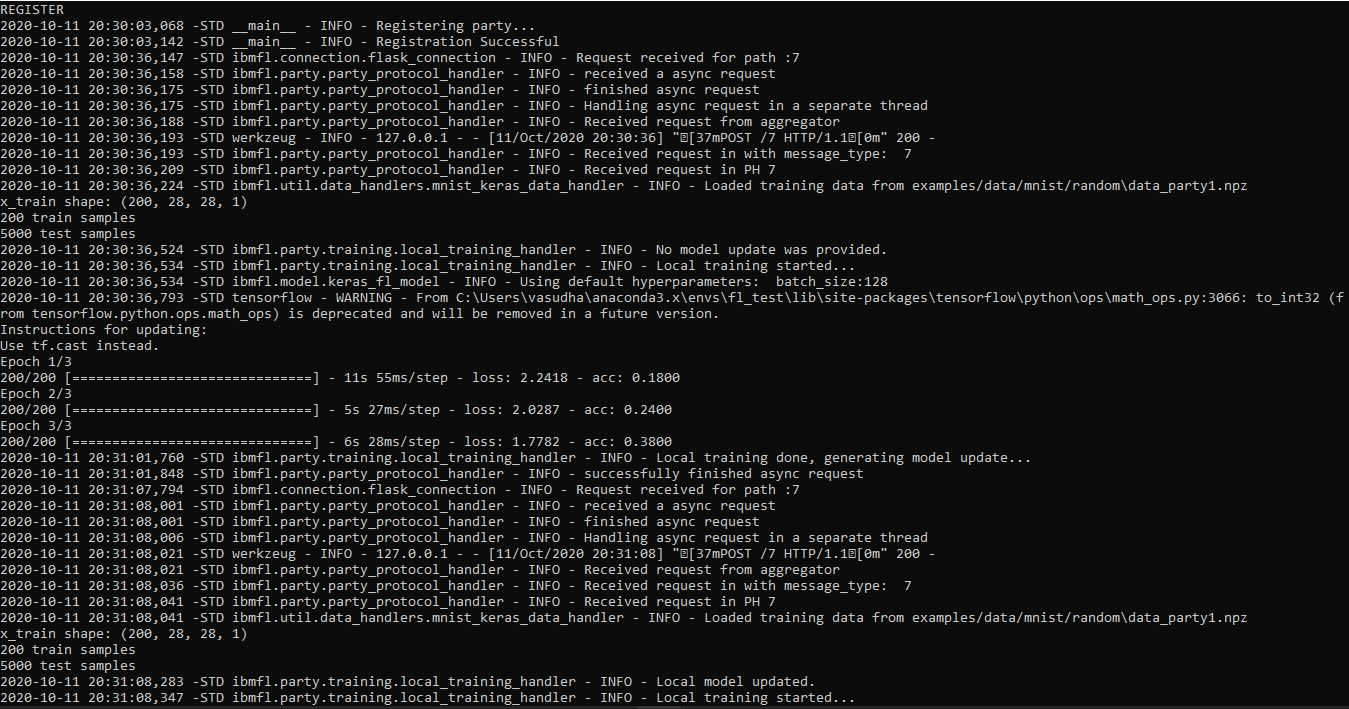
Party0

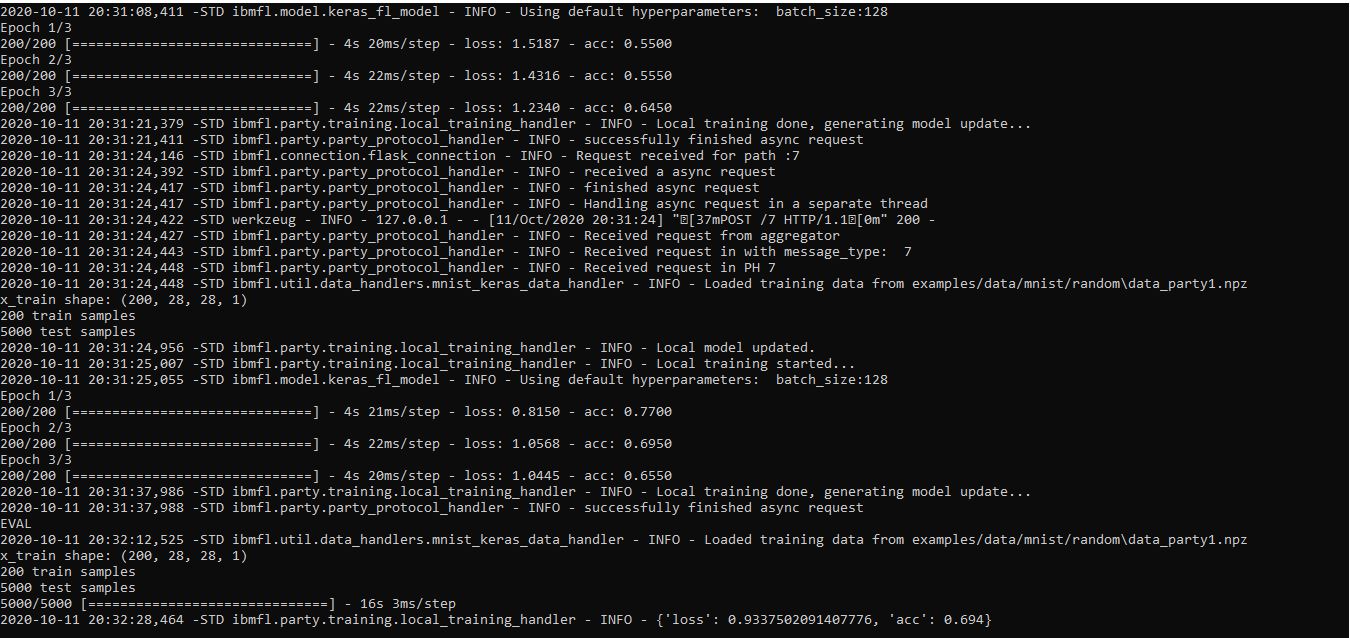






Party1





**Conclusions**

Applying Federated Learning requires machine learning practitioners to adopt new tools and a new way of thinking: model development, training, and evaluation . It enables many participating clients to train shared ML models, while keeping their data locally.It offers some distinct benefits over distributed machine learning. User privacy is protected by not having to upload massive amounts of personal data to a central server, and cost is brought down because devices do not have to be in a central data center location

Federated Learning allows for smarter models, lower latency, and less power consumption, all while ensuring privacy. It was interesting to see how federated learning evolved over time to account for information security attack.