Image Classification Model

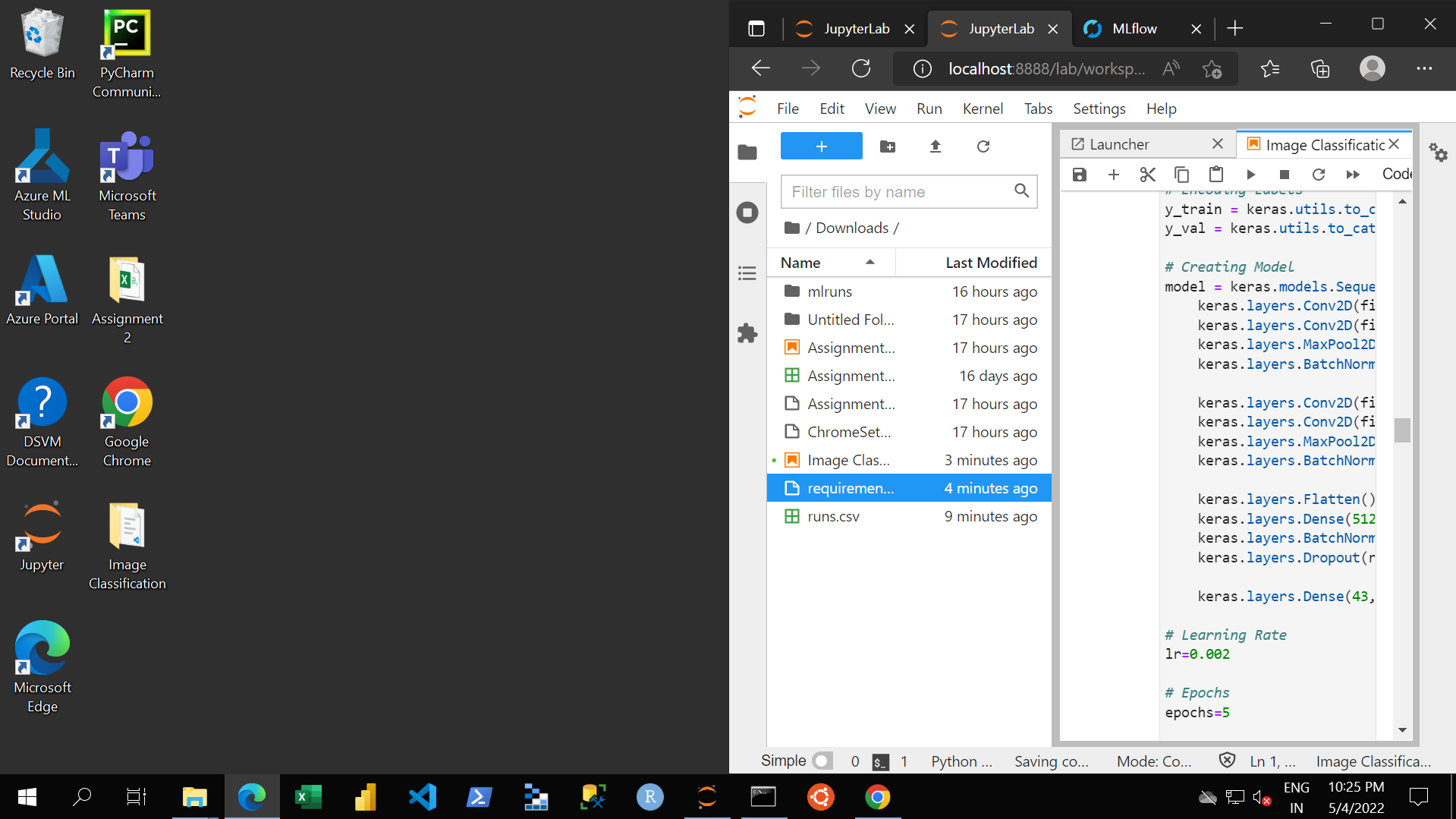
**Description**

Image classification is a very common problem today especially in the evolving field of self-driving cars. A Deep Learning ML model is used to classify the traffic sign images.

**Getting Started**

**Dependencies**

* Python 3
* Scikit-Learn
* Pandas
* Numpy
* TensorFlow
* OpenCV
* Current Web Browser
* Jupyter Notebook
* MLflow
* Azure/ AWS Cloud
* Virtual Machine Azure



**Installing**

* [www.anaconda.com/products/individual](http://www.anaconda.com/products/individual) (Python 3, Scikit-Learn & Jupyter Notebook)
* pandas.pydata.org/pandas-docs/stable/getting\_started/install.html
* [www.mlflow.org/docs/latest/quickstart.html#installing-mlflow](http://www.mlflow.org/docs/latest/quickstart.html#installing-mlflow)

**Executing program**

To Start MLflow, go to anaconda prompt and execute:

1. Start MLflow server:

mlflow server --backend-store-uri 'C:/temp/mlflow/localserver'

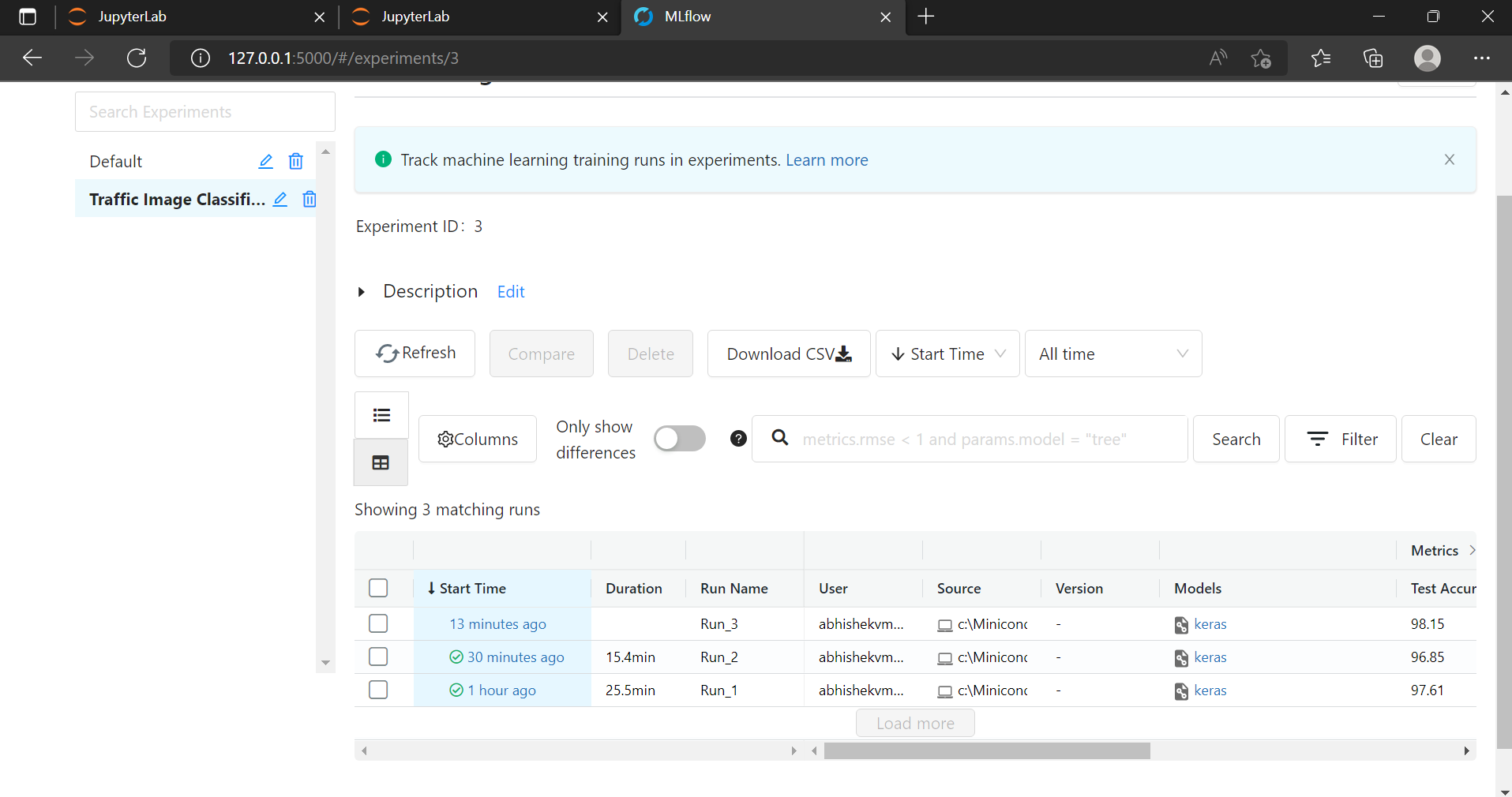
1. Open browser and go to http://localhost:5000 for MLflow UI

Once MLflow is running, open a new anaconda prompt and execute:

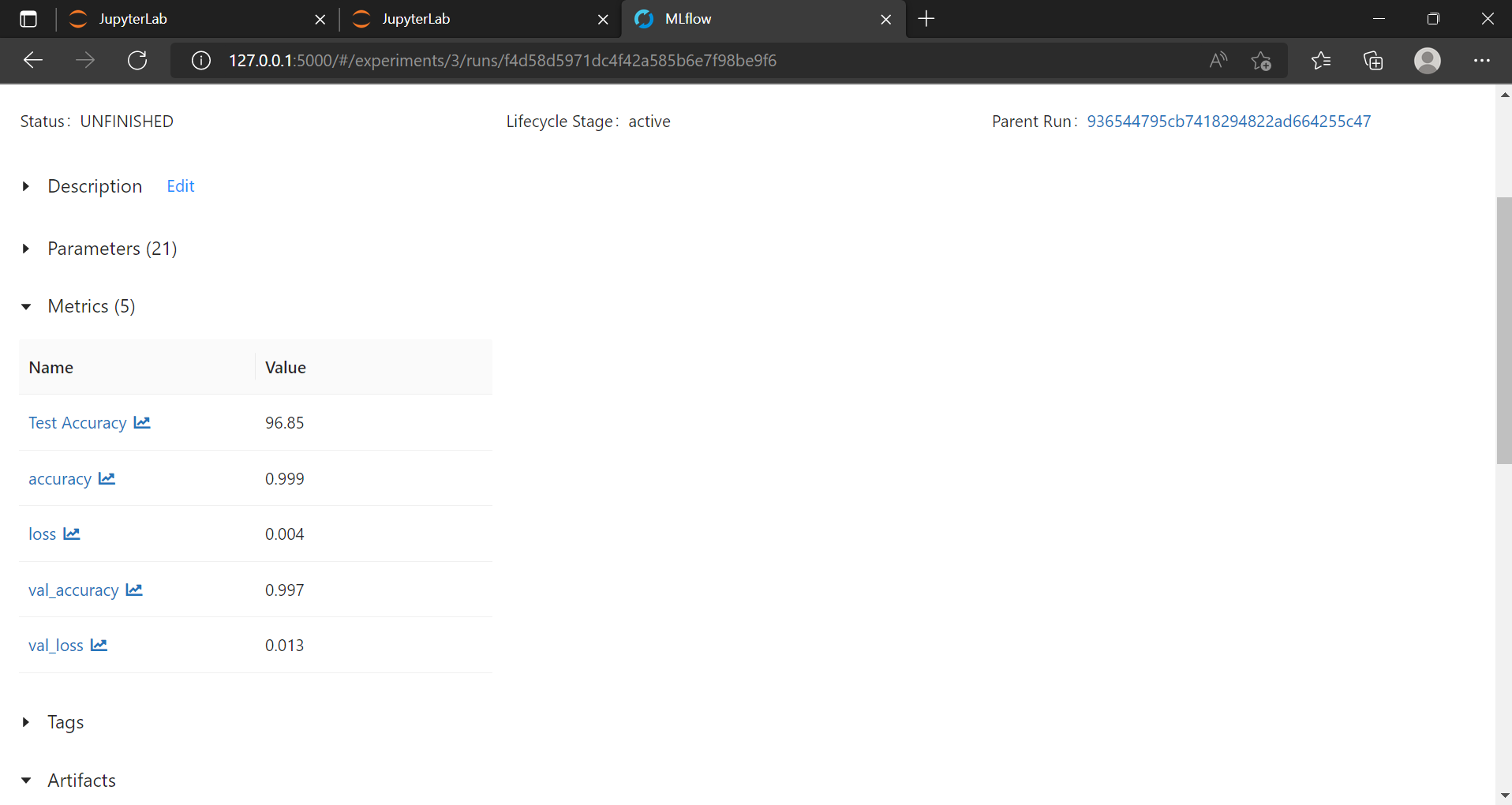
Jupyter Notebook

The above url will change as every virtual machine will take you to a different address. In our case, it was <http://127.0.0.1:5000>.

**Expected Results**

****

Run the first experiment, and go to <http://localhost:5000> and view the results in MLflow. After running all code blocks, you should see three runs with green checkmarks.

****

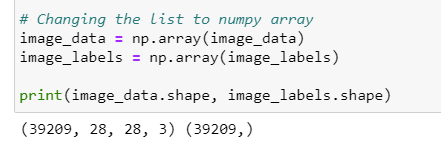
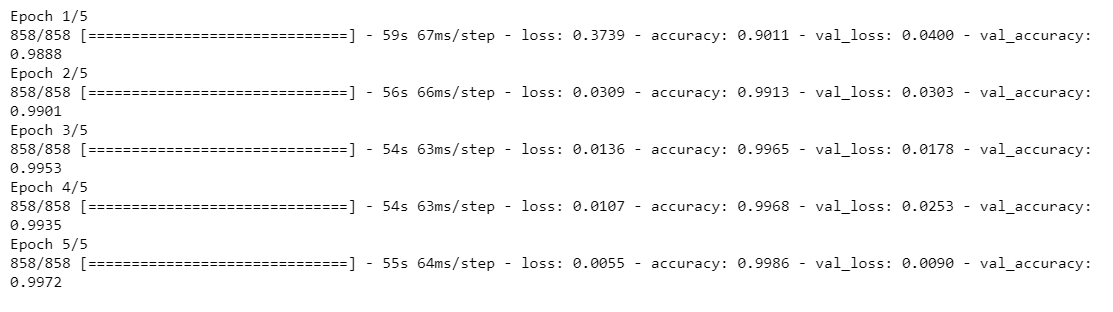
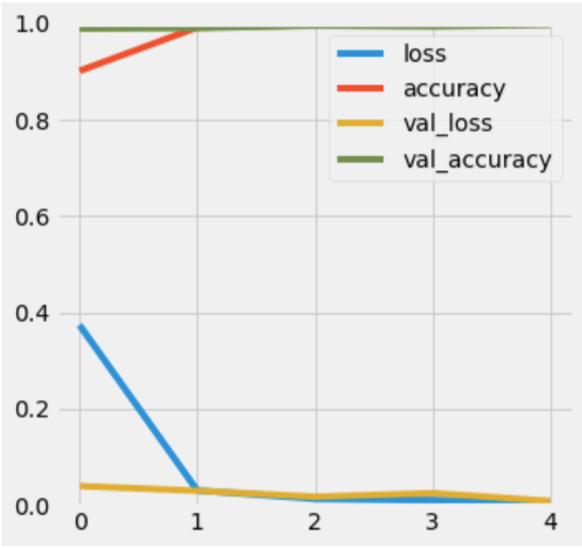
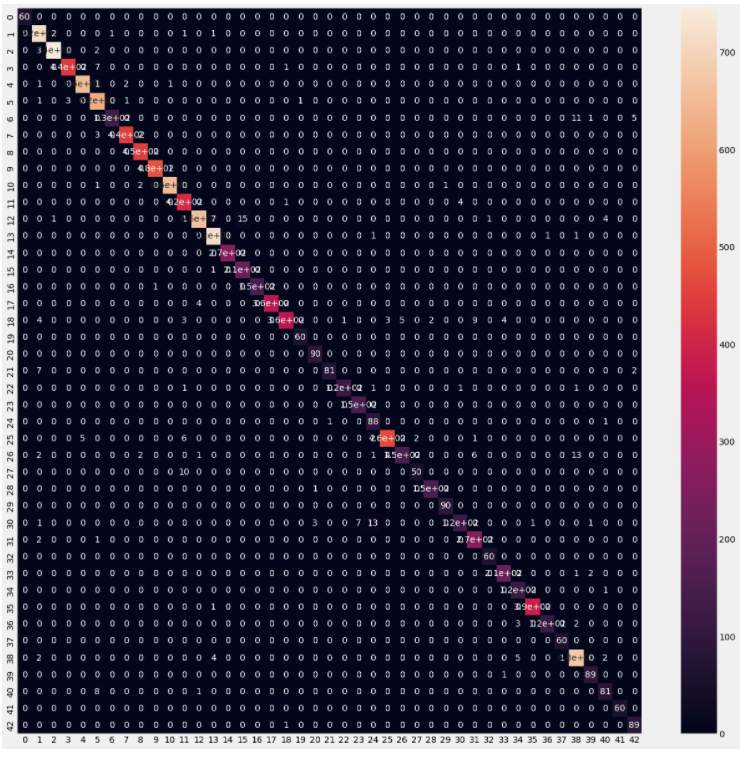
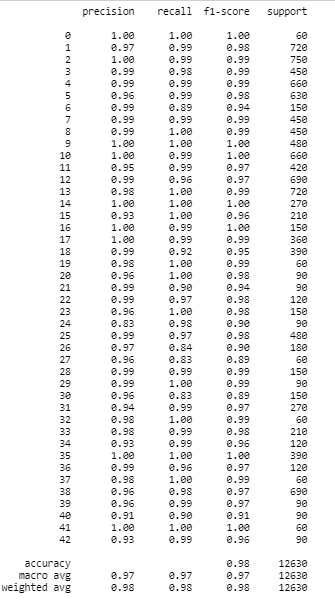
**TODO**

* + Set up cloud environment to run TensorFlow on Azure/ AWS
  + Upload your dataset to your cloud-based ML environment (Azure Blob or S3)
  + Perform exploratory data analysis (EDA)
  + Create a deep learning model to predict image class using TensorFlow
  + Use MLFlow to track the experiment, log artifacts and model

**Authors**

Abhishek Mote, Anurag Yadav

**Approach:**

* After the MLFlow was set, we started our experiment with assigning the path as we had csv as well as the image file as well.
* We provided the path directly to the notebook to access the entire dataset.
* Provided with labels already, so no need to generate separately, we jumped to data visualization to check how the data looks like,
* Our next step was to check how the testing data looks like, so we checked random data of 25 images.
* 
* The approach should be to provide the path of directly, the traditional method won’t work here.
* In addition, we didn’t store the data into S3 or blob, instead, we created the virtual machine, downloaded the data into the VM, provided the path to the notebook.
* Next step is to convert the image data into array, in order to run the algorithm.
* The data will look like this  
  
* Then comes the modeling part, splitting the data into training and testing data set.
* We used keras.model.sequential to initiate the model, used ‘relu’ and ‘softmax’ as the activation functions, adam as optimizer and loss function as categorical cross entropy.
* We ran the experiment with 5 epochs.   
  
* Our model ran successfully maintaining the accuracy of 99% throughout the epochs.
* We kept 5 epochs because of the performance limitations of the VM. The plot captured the details of the experiment  
    
  
* To check the model performance, we calculated the confusion matrix.   
  
* And the classification report is as follows:  
  
* And to conclude, how our experiment worked, we checked the predicted images vs the actual images.
* 
* The model predicted the images accurately, as it can be seen that actual = = predicted can be seen under each image**.**

**Explain the performance differences and advantages of running TensorFlow on cloud platform vs running it on a local machine**

The performance differences

* A virtual machine needs to be setup initially in order to run the experiment
* Virtual space like Blob or S3 needs to be purchased to store the data
* Both of these involves cost.
* VM comes with a limitation that it misses out on some of the libraries that needs to be pip installed before running.
* Azure or AWS, has the older version of python, that is 3.6 however the latest version is 3.10, so that needs to be updated in order to run few libraries.
* These problems are not associated while working in local machine as any update can be done through conda cmd and pip install.
* Also, can’t keep VM machine running for a longer time as it may cost you a lot based on the usage whereas the local machine is free to use as long as we want.

The Advantages

* We can use virtual machine practically on any laptop or desktop, as it’s a cloud service, we just need to connect it with Azure or AWS.
* Easy to install and create a VM with documentation
* Very fast when we provide multi-CPU’s
* The model runs efficiently without affecting the local machine.
* Cost is as per the use, and not on the subscription model. So you need to pay for the time you used the machine, which is mostly hourly based and can be as low as $0.20 per hour.