



Science & Technology Facilities Council
Rutherford Appleton Laboratory

Machine Learning Tutorial Overview

Emmanuel Olaiya

Rutherford Appleton Laboratory

15th June 2026

Agenda

Today, Monday
15th June, this
afternoon

- Tutorial walkthrough
- 1400 - 1540
- We will have breaks
- 1600 - 1730

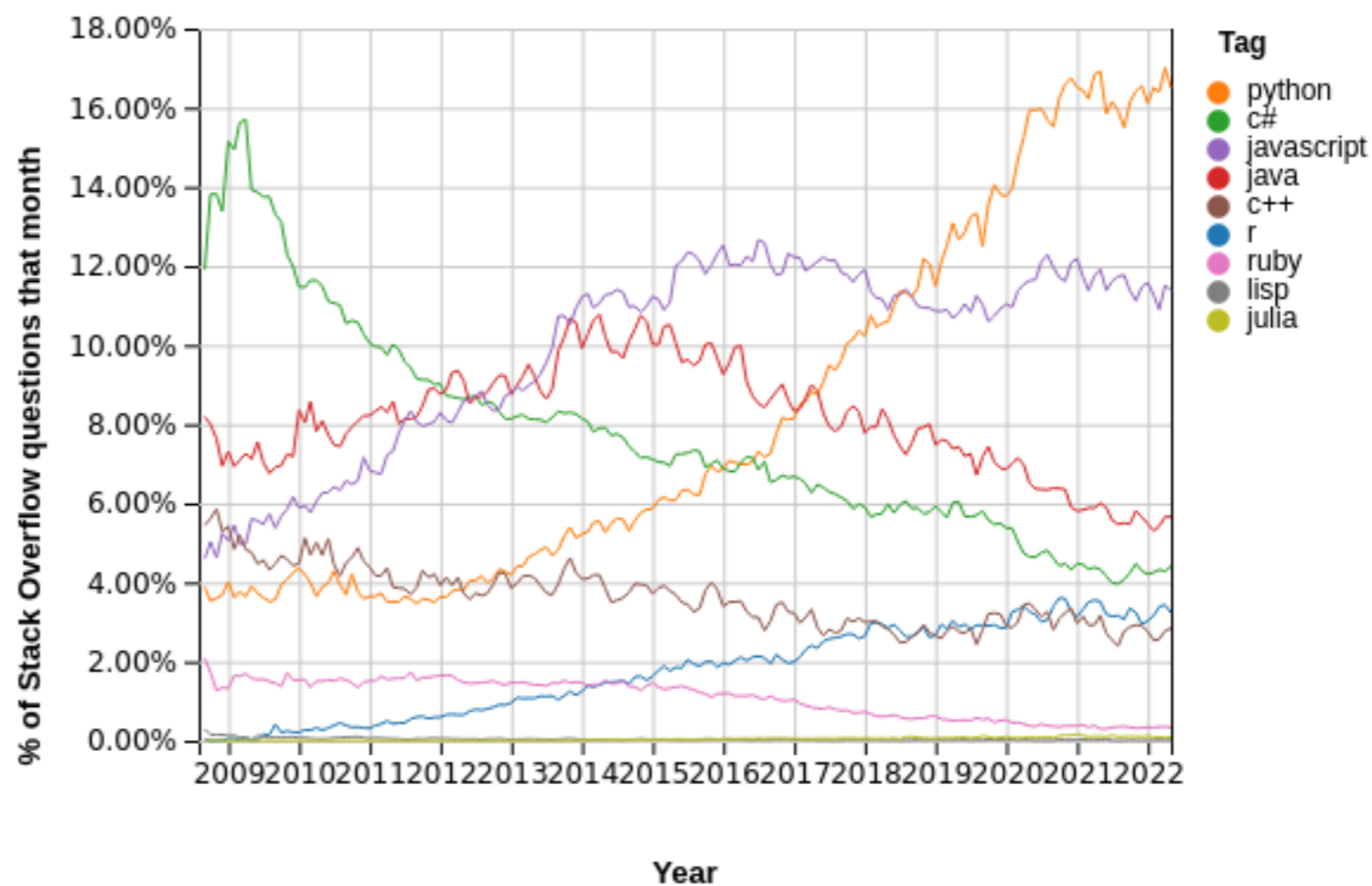
12:00	Machine Learning lectures and tutorial (emmanuel.olaiya@stfc.ac.uk) <i>Hamilton Room, RAL, Visitors Centre</i>	<i>Emmanuel Olaiya</i> 12:00 - 12:55
13:00	Lunch <i>Hamilton Room, RAL, Visitors Centre</i>	13:00 - 14:00
14:00	Machine Learning lectures and tutorial (emmanuel.olaiya@stfc.ac.uk) <i>Hamilton Room, RAL, Visitors Centre</i>	<i>Emmanuel Olaiya</i> 14:00 - 15:40
15:00	coffee/tea <i>Hamilton Room, RAL, Visitors Centre</i>	15:40 - 16:00
16:00	Machine Learning lectures and tutorial (emmanuel.olaiya@stfc.ac.uk) <i>Hamilton Room, RAL, Visitors Centre</i>	<i>Emmanuel Olaiya</i> 16:00 - 17:30
17:00		

Content

- What is the goal of the tutorial
 - Continue the learning. Learn by doing!
 - Give you some hands on experience
 - Introduce you to some simple tools that have a wide application
 - Again this is an introduction. So the examples will be straightforward. Anyone who has done any machine learning before may find it easy. This workbook is for beginners!
 - The tutorial will be a walkthrough
 - It will be interactive. You will be able to execute code. Modify it if you want
 - There will be some exercises for you to practice implementing machine learning code

Contents

- What programming languages are used for machine learning
 - Python, R, Java, Julia, LISP, C++, , lots!
- Python is a language that continues to grow in popularity. Python is the language we will use



Contents

- So what are the machine learning toolkits available
 - Again lots: Tensorflow, Pytorch, Sklearn, Amazon Machine Learning (AML), Shogun,
 - We will be using Tensorflow, a platform owned by Google, it is well supported, easy to use and has a wide range of features
 - GPU support is seamless (for Nvidia cards!)
 - Good at detecting if you have GPUs and then just using them without any need for configuring

Jupyter Notebooks

- We will use Jupyter Notebooks for this tutorial
- What are they?
- They are a web-based Interactive Development Environment (IDE), from which you can program and run code

How do we run the notebooks

- There are many ways to run the notebooks. I am going to suggest 4

If you have a preferable method then that is fine

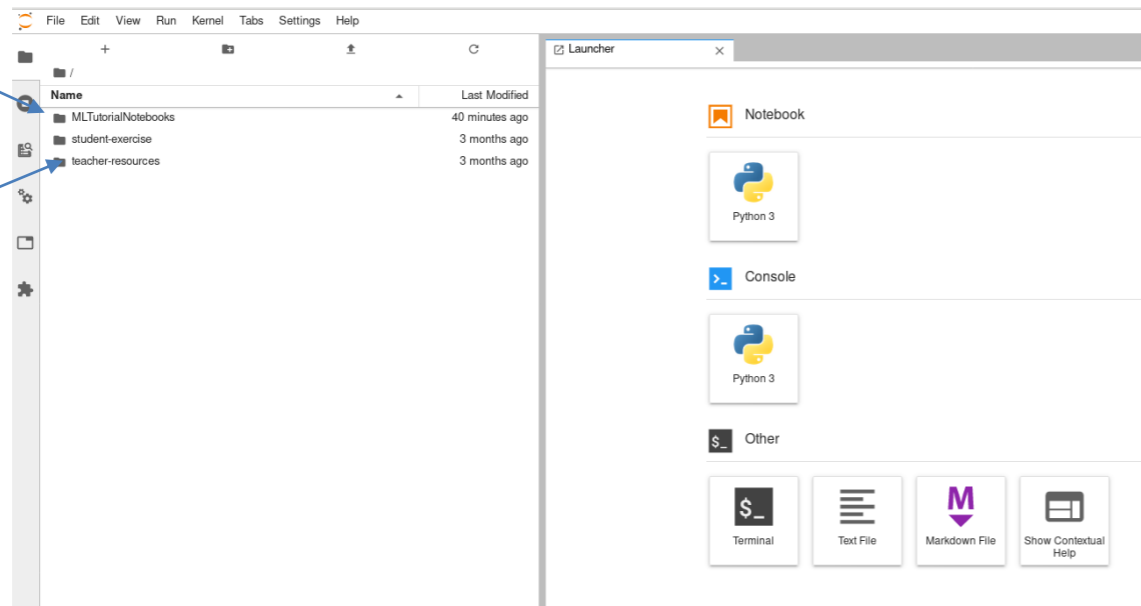
My suggestions are:

1. Using the RAL computing cluster
2. Using mybinder.org
3. Using Google Colab
4. Running on your own computer

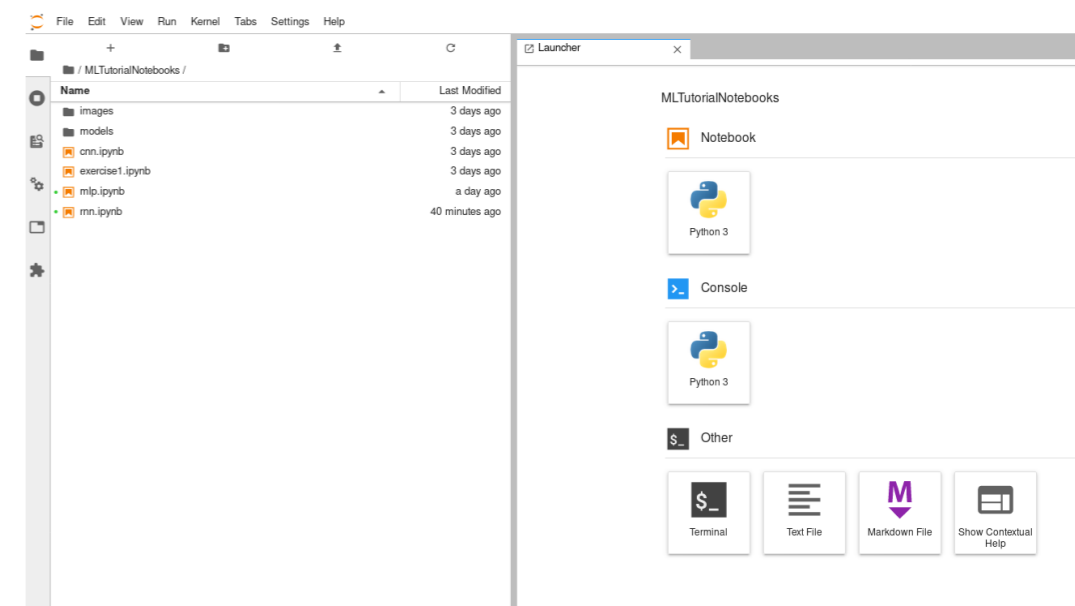
Using the RAL Cluster

- If you want to use the RAL cluster I will provide you with a username and a password
- Go to the page <https://monty-backup.stfc.ac.uk> Log in with your username and password. You should see the following:

Click on MLTutorialNotebooks



The student-exercise and teacher-exercise tabs won't be there



Click on one of the .ipynb files to run the notebook

Using mybinder.org

- Everyone can use this option
- Just click on this link:
 - <https://mybinder.org/v2/gh/olaiya/MLTutorialNotebooks.git/HEAD>
- After a while you should be able to see the notebook



The screenshot shows the JupyterLab interface. At the top left is the "jupyter" logo. On the top right, there are three buttons: "Visit repo", "Copy Binder link", and "Quit". Below the logo, there are three tabs: "Files", "Running", and "Clusters". The "Files" tab is active, showing a file browser. The browser displays a list of files and folders with columns for "Name", "Last Modified", and "File size". The files listed are:

Name	Last Modified	File size
0		
/		
images	12 minutes ago	
models	12 minutes ago	
cnn.ipynb	12 minutes ago	12.7 kB
exercise1.ipynb	12 minutes ago	8.57 kB
mlp.ipynb	12 minutes ago	29.3 kB
mn.ipynb	12 minutes ago	85.7 kB
requirements.txt	12 minutes ago	49 B

If you are using mybinder, connect to the above link 5 mins before the start of the tutorial. Sometimes it can take a few minutes to fire up the container that hosts the notebook

Using Google Colab

- People with a Google account can use this option
- Make sure you are logged into your Google account
- Go to the workbook files on github:
 - <https://github.com/olaiya/MLTutorialNotebooks>

The screenshot displays the GitHub repository page for `olaiya/MLTutorialNotebooks`. The repository is currently on the `master` branch, which has 1 branch and 0 tags. The commit history shows a recent commit by Emmanuel Olaiya and Emmanuel Olaiya, adding a link to Google Colab. The repository contains several files and directories, including `images`, `models`, `cnn.ipynb`, `exercise1.ipynb`, `mlp.ipynb`, `requirements.txt`, and `rnn.ipynb`. The sidebar on the right provides information about the repository, including the absence of a description, releases, or packages, and a 100.0% Jupyter Notebook language usage.

File	Commit Message	Time Ago
images	Added exercise1.ipynb and rnn.ipynb	3 days ago
models	Added models directory	3 days ago
cnn.ipynb	Added link to google colab	13 hours ago
exercise1.ipynb	Added exercise1.ipynb and rnn.ipynb	3 days ago
mlp.ipynb	Added link to google colab	13 hours ago
requirements.txt	Added requirements file	3 days ago
rnn.ipynb	Added link to google colab	13 hours ago

Using Google Colab

- When we work on a specific workbook, say mlp.ipynb, click on it

The screenshot shows a GitHub repository page for 'olaiya / MLTutorialNotebooks'. The repository has 1 issue, 0 pull requests, 0 actions, 0 projects, 0 wiki pages, 0 security issues, 0 insights, and 0 settings. The current branch is 'master'. The file 'mlp.ipynb' is selected, and the commit history shows a commit by Emmanuel Olaiya titled 'Added link to google colab' from 13 hours ago. The file content is displayed, showing the title 'Building and training a Multi Layered Perceptron (MLP) using Tensorflow'. Below the title is an 'Open in Colab' button, which is highlighted by a yellow callout box with the text 'Click on open in Colab'. The callout box has a blue arrow pointing to the button. The text below the title reads: 'In this workbook we will use the python library Tensorflow to implement an MLP. We will implement MLPs for classification as a way of dipping into Tensorflow. We will also cover considerations for training such as batch sizes and learning rates as well as ways to avoid overfitting. We will also looking at the training loss output as well as saving and loading models. To run a code cell, click on the cell the press "Shift + Enter"'. Below this is a code cell with the following code:

```
In [ ]: import numpy as np
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
from matplotlib import cm

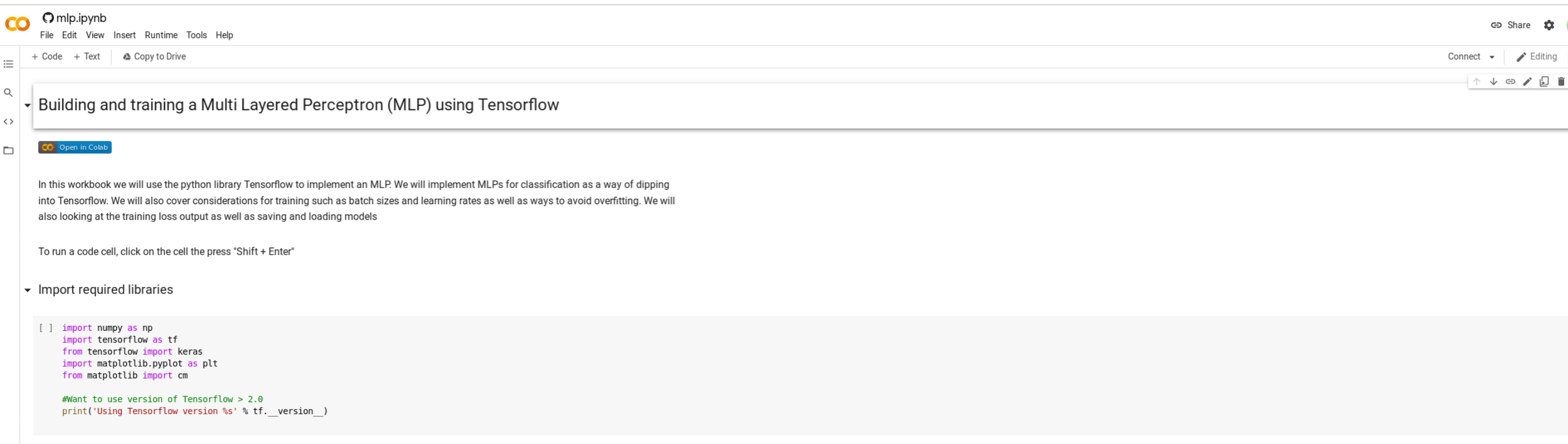
#Want to use version of Tensorflow > 2.0
print('Using Tensorflow version %s' % tf.__version__)
```

 Below the code cell is the section 'Create the data' with the text: 'Let's generate a dataset consisting of two data types which we call signal and background. Each data type is normally generated around a point in the x-y plane'.

- If clicking on icon doesn't work, copy and paste link in your browser

Using Google Colab

- The workbook should load



The screenshot shows a Google Colab notebook interface. At the top, the notebook title is "Building and training a Multi Layered Perceptron (MLP) using Tensorflow". Below the title, there is a blue button labeled "Open in Colab". The main content of the notebook consists of two text cells. The first cell contains an introductory paragraph: "In this workbook we will use the python library Tensorflow to implement an MLP. We will implement MLPs for classification as a way of dipping into Tensorflow. We will also cover considerations for training such as batch sizes and learning rates as well as ways to avoid overfitting. We will also looking at the training loss output as well as saving and loading models". The second cell contains a short instruction: "To run a code cell, click on the cell the press 'Shift + Enter'". Below this, there is a code cell titled "Import required libraries" containing the following Python code:

```
[ ] import numpy as np
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
from matplotlib import cm

#Want to use version of Tensorflow > 2.0
print('Using Tensorflow version %s' % tf.__version__)
```

Using Your Own Computer

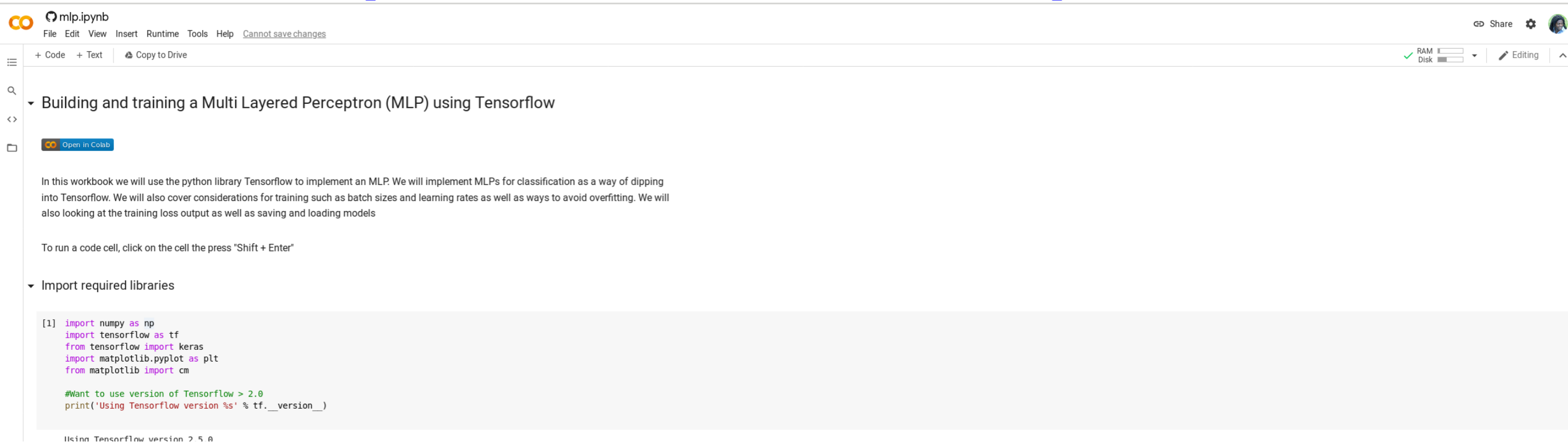
- If you want to run the workbook on your own computer that is fine
- Workbook is intentionally lightweight for this purpose
- Many setups you can use, install all the software, use a container such as Docker, use Conda
- What you need is git, python3 and the following libraries, tensorflow (>2.0.0), numpy, pandas, sklearn, matplotlib, jupyter
 - You don't need a GPU
- To pull the workbook onto you computer run:
 - `git clone https://github.com/olaiya/MLTutorialNotebooks.git`
 - See the requirements.txt file for the python modules required
 - `pip install -r requirements.txt`
 - `conda install -file requirements.txt`

Or download the workbook directly from the webpage: <https://github.com/olaiya/MLTutorialNotebooks>

- Run jupyter notebook

Running A Workbook

- Workbooks are a collection of cells. The cells are either code cells or markdown cells (adding text or images)
- To run a piece of code in a cell, select the cell and press Shift+Enter



The screenshot shows a Jupyter Notebook interface. The top bar includes the Jupyter logo, the URL 'mlp.ipynb', and a menu with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. A notification 'Cannot save changes' is visible. Below the menu, there are buttons for '+ Code', '+ Text', and 'Copy to Drive'. On the right, there are indicators for 'RAM' and 'Disk' usage, and a status 'Editing'. The main content area is titled 'Building and training a Multi Layered Perceptron (MLP) using Tensorflow'. It contains a paragraph of introductory text, a code cell with the following Python code, and a status message 'Using Tensorflow version 2.5.0'.

```
[1] import numpy as np
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
from matplotlib import cm

#Want to use version of Tensorflow > 2.0
print('Using Tensorflow version %s' % tf.__version__)
```

- To add a cell click Insert->Code/Text cell
 - Some instances Insert->Cell Above/Below
 - Then select Cell->Cell type (to change between code and markdown)

Today's Workbook Tutorial

- Today we will walk through workbooks on:
 - MLPs
 - Maybe a little on autoencoders
 - CNNs
 - RNNs
- Hopefully we will have a bit of time for you modify the code and even create and run some code yourself
- Decide how you want to access the workbook. These slides will be available on the Indico agenda if you want to review the options again at a later date. Access to monty.stfc.ac.uk will be available all week