National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Venkata Sai Deepika Devisetty
Student ID: X20251483

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Venkata Sai Deepika Deviestty
Student ID:	X20251483
Programme:	Data Analytics **Year:** 2023
Module:	Research Project
Lecturer:	Dr. Catherine Mulwa
Submission Due Date:	14/08/2023
Project Title:	Identification and Detection of Brain Tumors using Machine learning and Deep Learning methods
Word Count:	648
Page Count:	6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Venkata Sai Deepika Devisetty

Date: 14/08/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Venkata Sai Deepika Devisetty
Student ID: x20251483

1 Introduction

This document provides a concise explanation of the manner in which the project to identify and detect brain tumours was implemented. This document contains every technology and technique required for implementation. We learn how to set up an environment for programming in part 2. Section 3 provides illustrations of various tools and software. Section 4 details how the project was executed out.

2 Environment set up

The configuration that was set up to implement the project are stated below:

1. Processor: Intel i7
2. Memory: 16GB RAM
3. Programming language: Python3
4. Python Environment: Google Colab, Jupyter Notebook

3 Softwares utilised

We have used Google Colab for training model and obtained accuracies for the models since there was limited availability of computer resources. Google Colab supports many libraries like PyTorch, TensorFlow, Keras, OpenCV which are necessary for the execution.

4 Implementation

4.1 Step 1: We have taken google colab into consideration for python code. Go to https://colab.research.google.com/ url and sign in with your account.

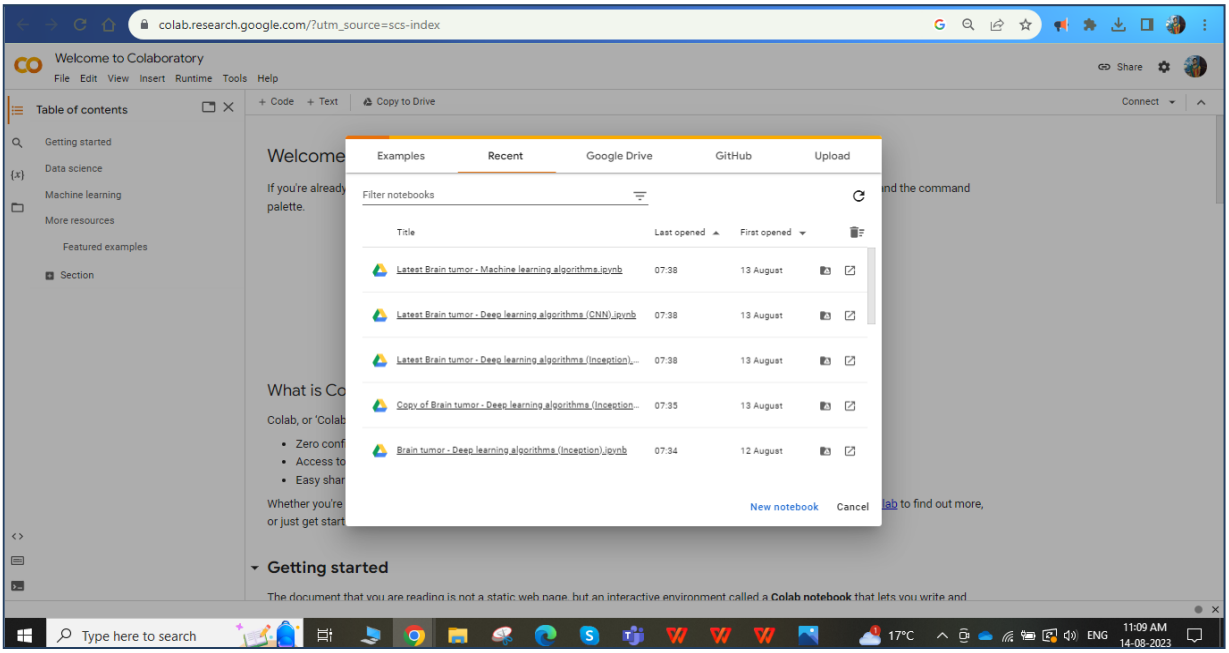

Fig 1 States the opening of Google Colab

4.2 step 2: Open the required programme for implementation - Machine learning - 'Latest Brain tumor - Machine learning algorithms.ipynb'

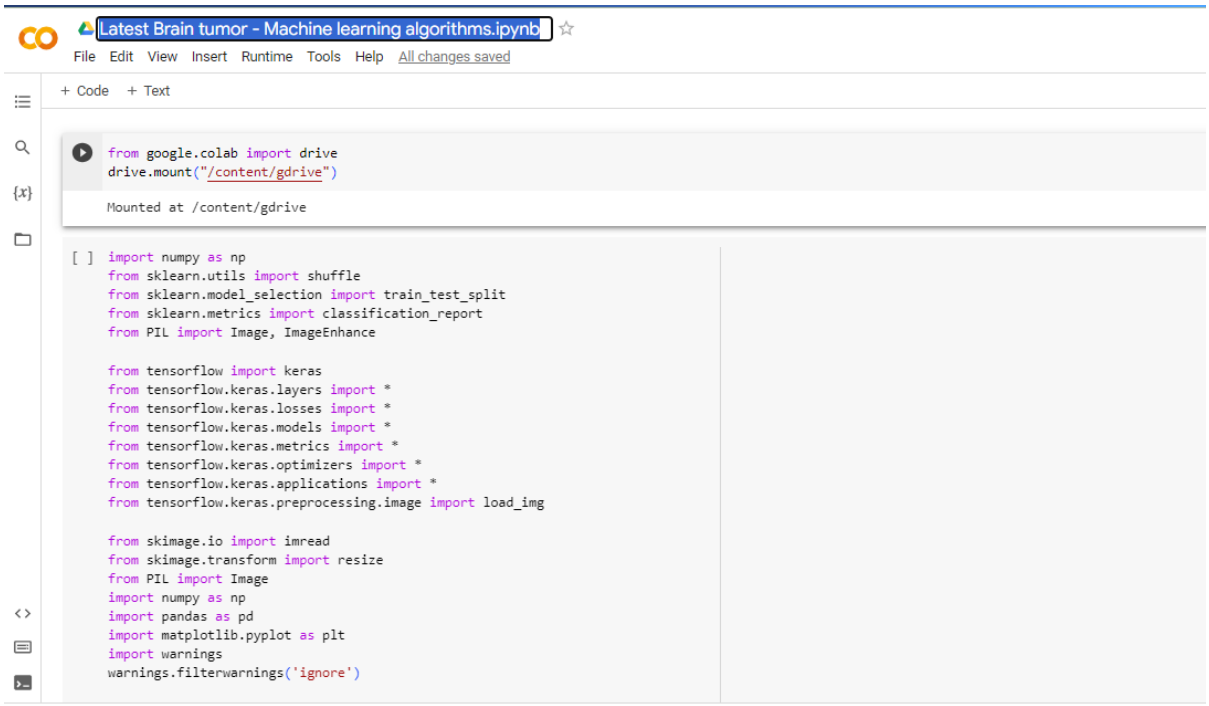

Fig 2 Shows the dataset been loaded from drive and libraries are being imported

Parallely, open the program for CNN implemtation by opening 'Latest Brain tumor - Deep learning algorithms (CNN).ipynb'

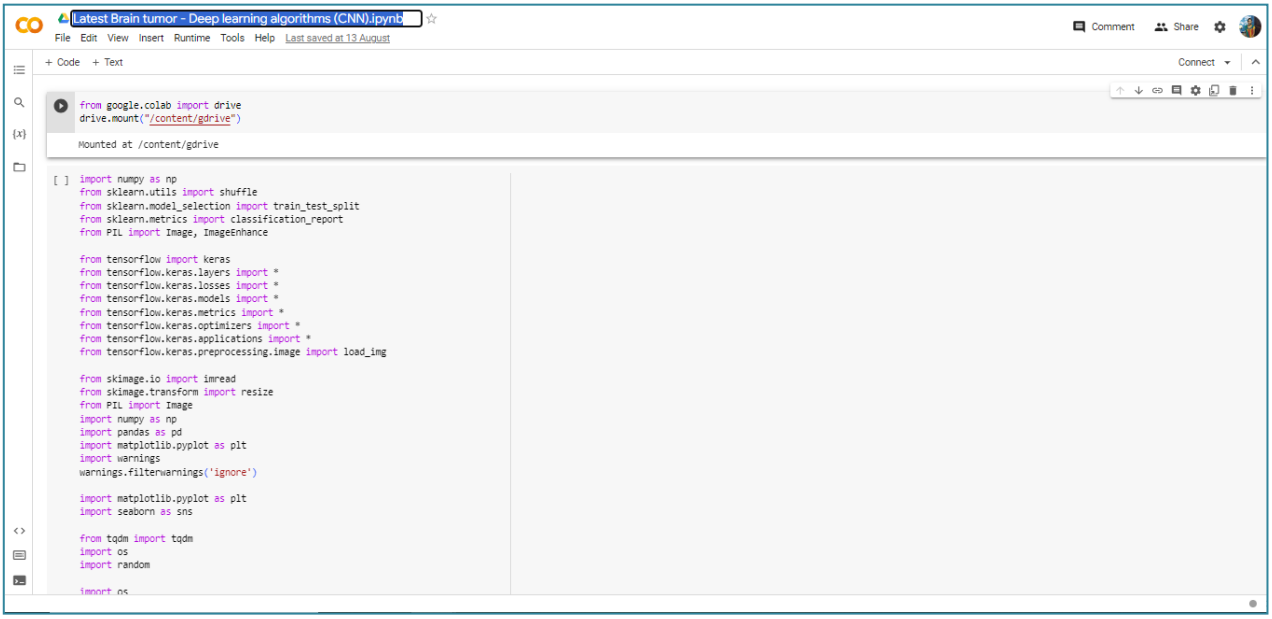

Fig 3 Shows the dataset been loaded from drive and libraries are being imported

On the other side open the programme for Inception method with 'Latest Brain tumor - Deep learning algorithms (Inception).ipynb'

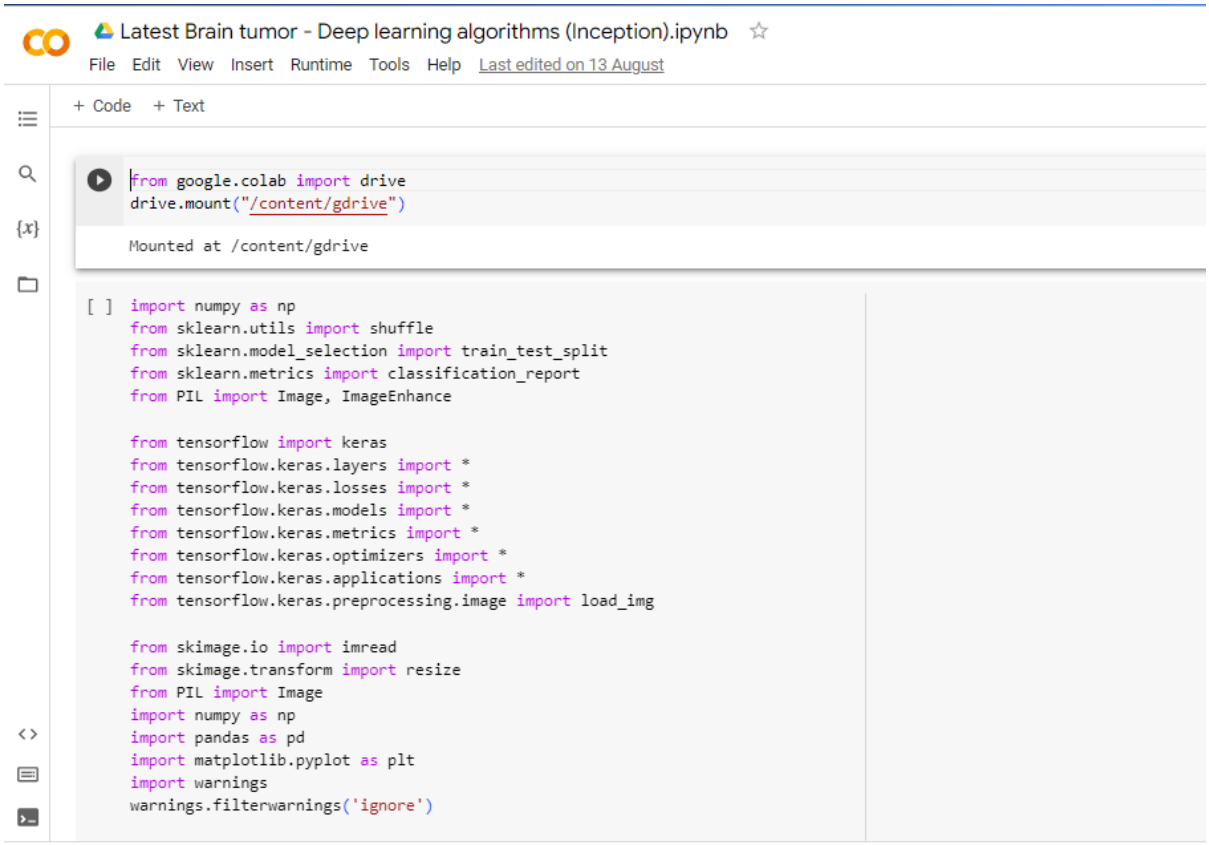

Fig 4 Shows the dataset been loaded from drive and libraries are being imported

Similarly open the programme for implementation‘Latest Brain tumor - Deep learning algorithms (VGG).ipynb and loading of the dataset using pandas libraries which is stored in google drive to access in google collab has been done for all the models.

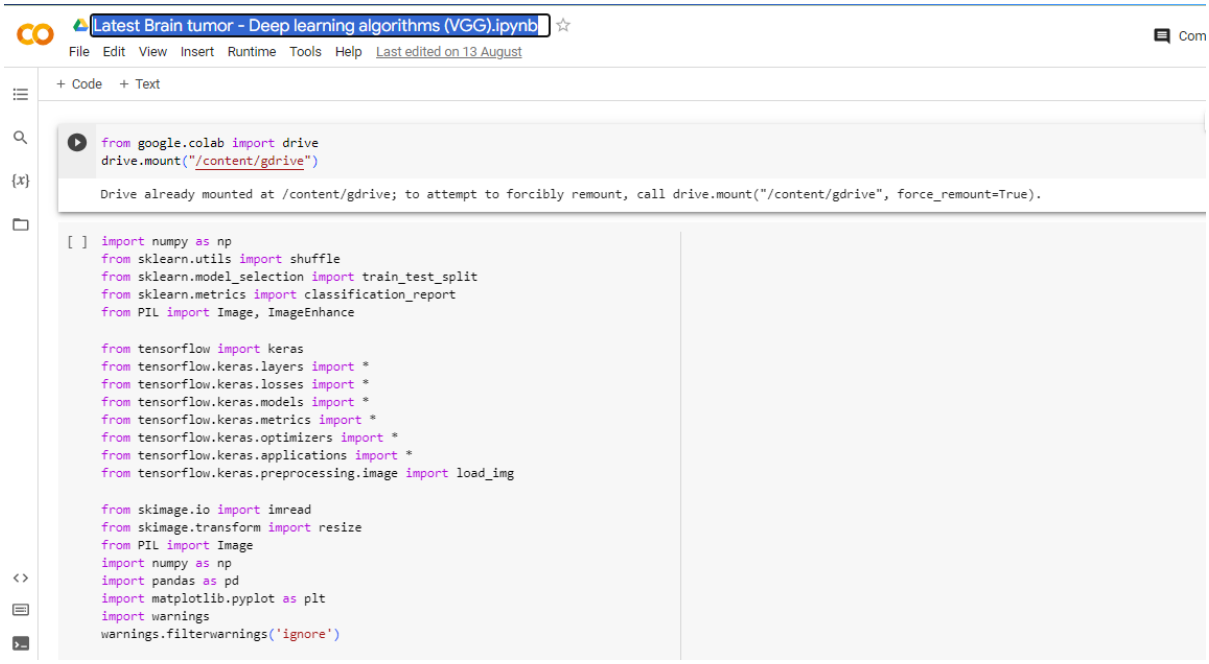

Fig 5: Shows the dataset been loaded from drive and libraries are being imported

4.3 Step 4 Code in Google collab can be executed by using Run button

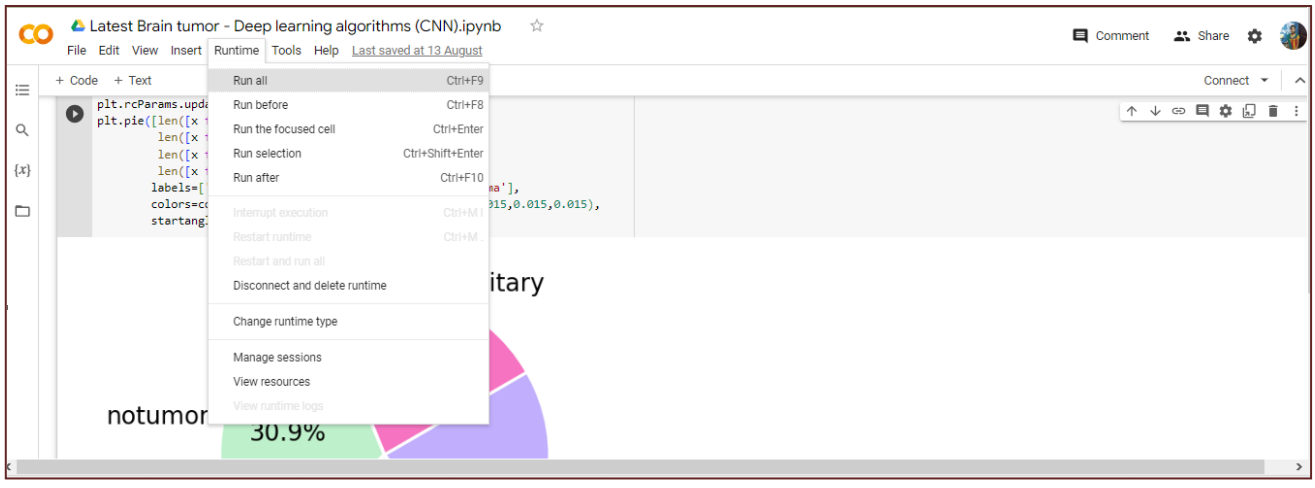

Fig 5: Shows to run the code

4.4 Step 4:Training the pretrained model with the test data and getting accuracies for CNN Method

```
[ ]  print(classification_report(y_true, y_pred))
```

	precision	recall	f1-score	support
glioma	0.84	0.83	0.84	300
meningioma	0.75	0.74	0.75	306
notumor	0.94	0.92	0.93	405
pituitary	0.90	0.95	0.93	300
accuracy			0.87	1311
macro avg	0.86	0.86	0.86	1311
weighted avg	0.87	0.87	0.87	1311

```
[ ]  accuracy = accuracy_score(y_true, y_pred)
```

```
[ ]  accuracy
```

```
0.8657513348588863
```

Fig 6: Accuracy of the CNN models

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

cm = confusion_matrix(y_true, y_pred)

# Define the font size
font_size = 20

# Plot the confusion matrix
plt.figure(figsize=(8,8))
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d", xticklabels=unique_labels, yticklabels=unique_labels, annot_kws={"fontsize": font_size}, cbar=False)
plt.xlabel("Predicted Label", fontsize=font_size)
plt.ylabel("True Label", fontsize=font_size)
plt.xticks(fontsize=font_size)
plt.yticks(fontsize=font_size, rotation=0)
plt.show()
```

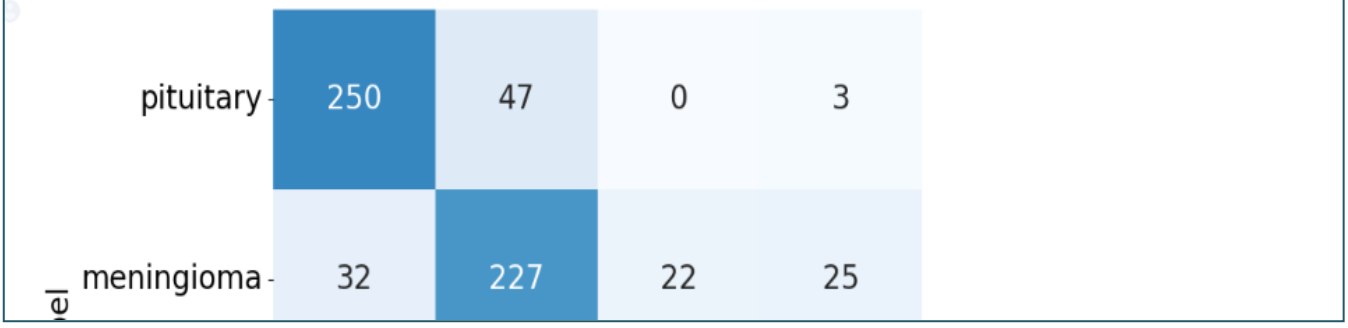

Fig 7: Confusion matrix for CNN

Training the pretrained model with the test data and getting accuracies for Machine learning Methods

```
Training NaiveBayes...
NaiveBayes trained in 6.90 seconds
Accuracy for NaiveBayes: 0.60
Classification Report for NaiveBayes:
              precision    recall  f1-score   support

      glioma       0.53      0.84      0.65       300
  meningioma       0.34      0.23      0.28       306
     notumor       0.73      0.58      0.64       405
   pituitary       0.76      0.77      0.77       300

    accuracy                           0.60      1311
   macro avg       0.59      0.61      0.58      1311
weighted avg       0.60      0.60      0.59      1311

Confusion Matrix for NaiveBayes:
[[252  22   1  25]
 [127  71  87  21]
 [ 45 100 235  25]
 [ 55  13   1 231]]
```

Fig 8: Results for NaiveBayes

```
Training SVC...
SVC trained in 711.72 seconds
Accuracy for SVC: 0.82
Classification Report for SVC:
              precision    recall  f1-score   support

      glioma       0.76      0.79      0.77       300
  meningioma       0.73      0.60      0.66       306
     notumor       0.88      0.91      0.90       405
   pituitary       0.87      0.95      0.91       300

    accuracy                           0.82      1311
   macro avg       0.81      0.81      0.81      1311
weighted avg       0.82      0.82      0.82      1311

Confusion Matrix for SVC:
[[236  50   0  14]
 [ 49 185  50  22]
 [ 20  10 370   5]
 [  5   9   0 286]]
```

Fig 9: Results for SVC

Training the pretrained model with the test data and getting accuracies for Inception Method

	precision	recall	f1-score	support
glioma	0.85	0.79	0.82	300
meningioma	0.76	0.74	0.75	306
notumor	0.95	0.94	0.94	405
pituitary	0.85	0.95	0.90	300
accuracy			0.86	1311
macro avg	0.85	0.86	0.85	1311
weighted avg	0.86	0.86	0.86	1311

Fig 10: Results for Inception

Training the pretrained model with the test data and getting accuracies for VGG Method

	precision	recall	f1-score	support
glioma	0.91	0.89	0.90	293
meningioma	0.86	0.85	0.86	297
notumor	0.97	0.97	0.97	396
pituitary	0.94	0.98	0.96	294
accuracy			0.93	1280
macro avg	0.92	0.92	0.92	1280
weighted avg	0.93	0.93	0.93	1280

Fig 11: VGG results

As seen by exection we have concluded that VGG has higher results with accuracy of 0.93.

References

www.tutorialspoint.com. (n.d.). Google Colab - What is Google Colab? - Tutorialspoint. [online] Available at: https://www.tutorialspoint.com/google_colab/what_is_google_colab.htm.