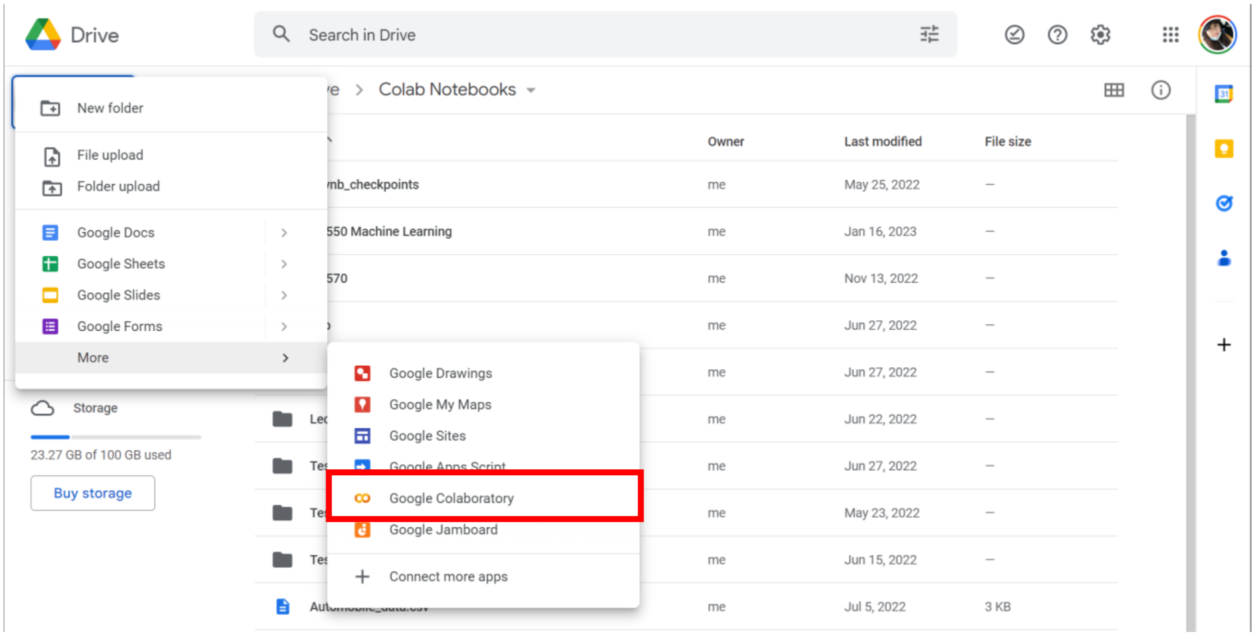
Getting Started with Colab

Step 1: Create a Colab project from Scratch

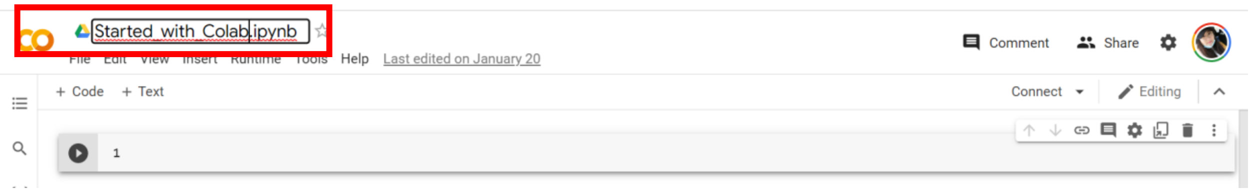
1) Create a New Notebook

➔ If you never used Colab before, go to the website to created new one
https://colab.research.google.com/notebooks/intro.ipynb

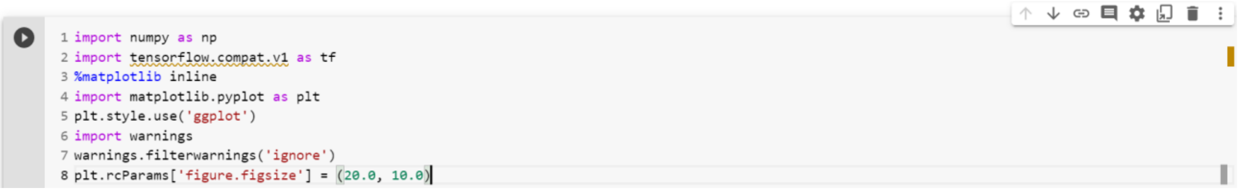
➔ I have used Colab before, go to
Google Drive -> Colab Notebooks -> Add a new Folder -> Create New Laboratory



2) Change File Name

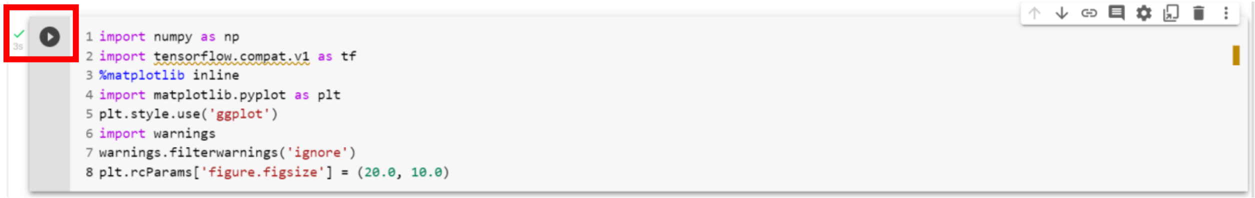

Started_with_Colab.ipynb
File Edit View Insert Runtime Tools Help
Last edited on January 20
Comment
Share
+ Code
+ Text
Connect
Editing
1

3) Copy Paste Code to Notebook



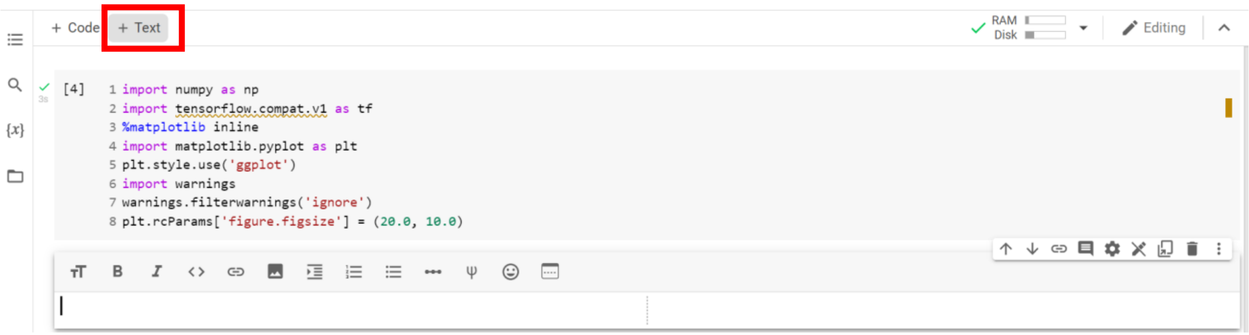
```python
import numpy as np
import tensorflow.compat.v1 as tf
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (20.0, 10.0)
```

4) Run the Code



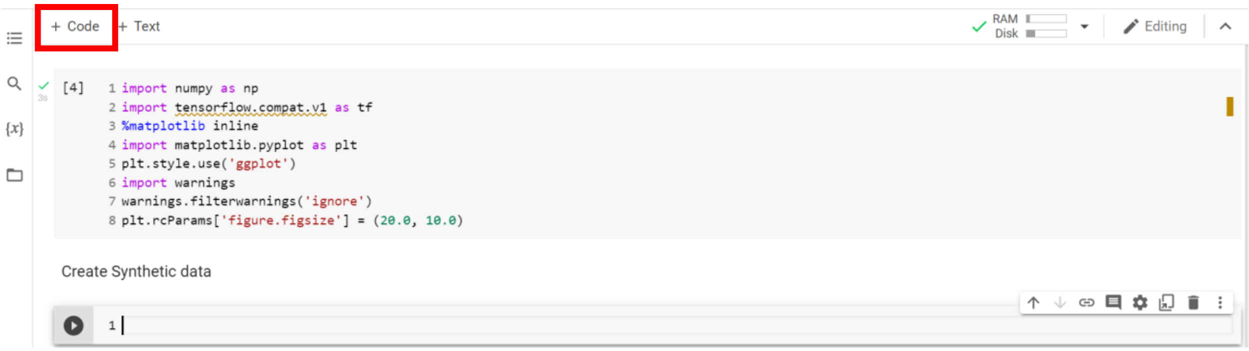
```
import numpy as np
import tensorflow.compat.v1 as tf
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (20.0, 10.0)
```

5) Add Text Block



```
import numpy as np
import tensorflow.compat.v1 as tf
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (20.0, 10.0)
```

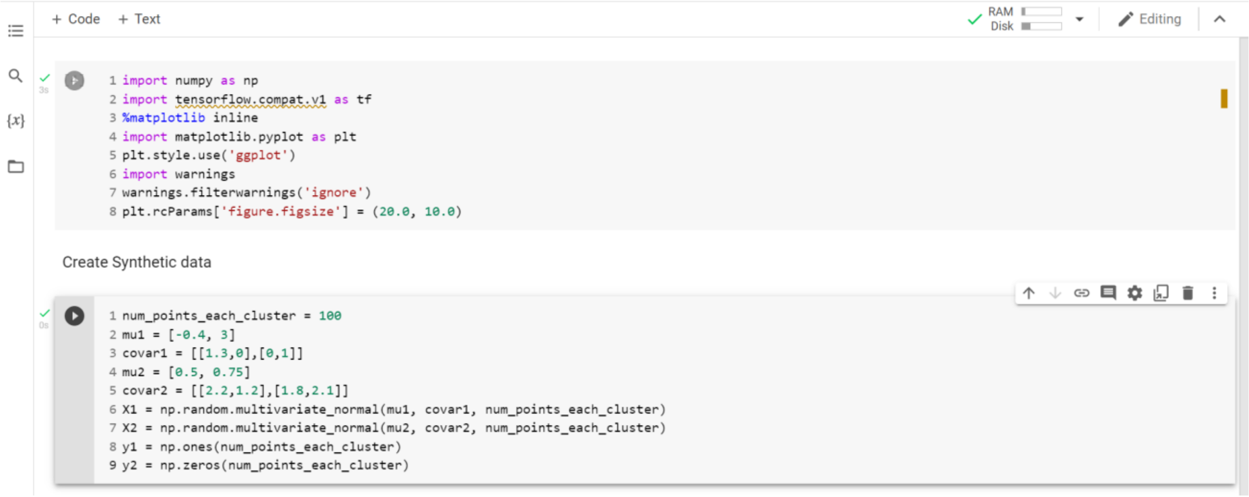
6) Add New Code Block



```
import numpy as np
import tensorflow.compat.v1 as tf
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (20.0, 10.0)
```

Create Synthetic data

7) Add New Code and Run



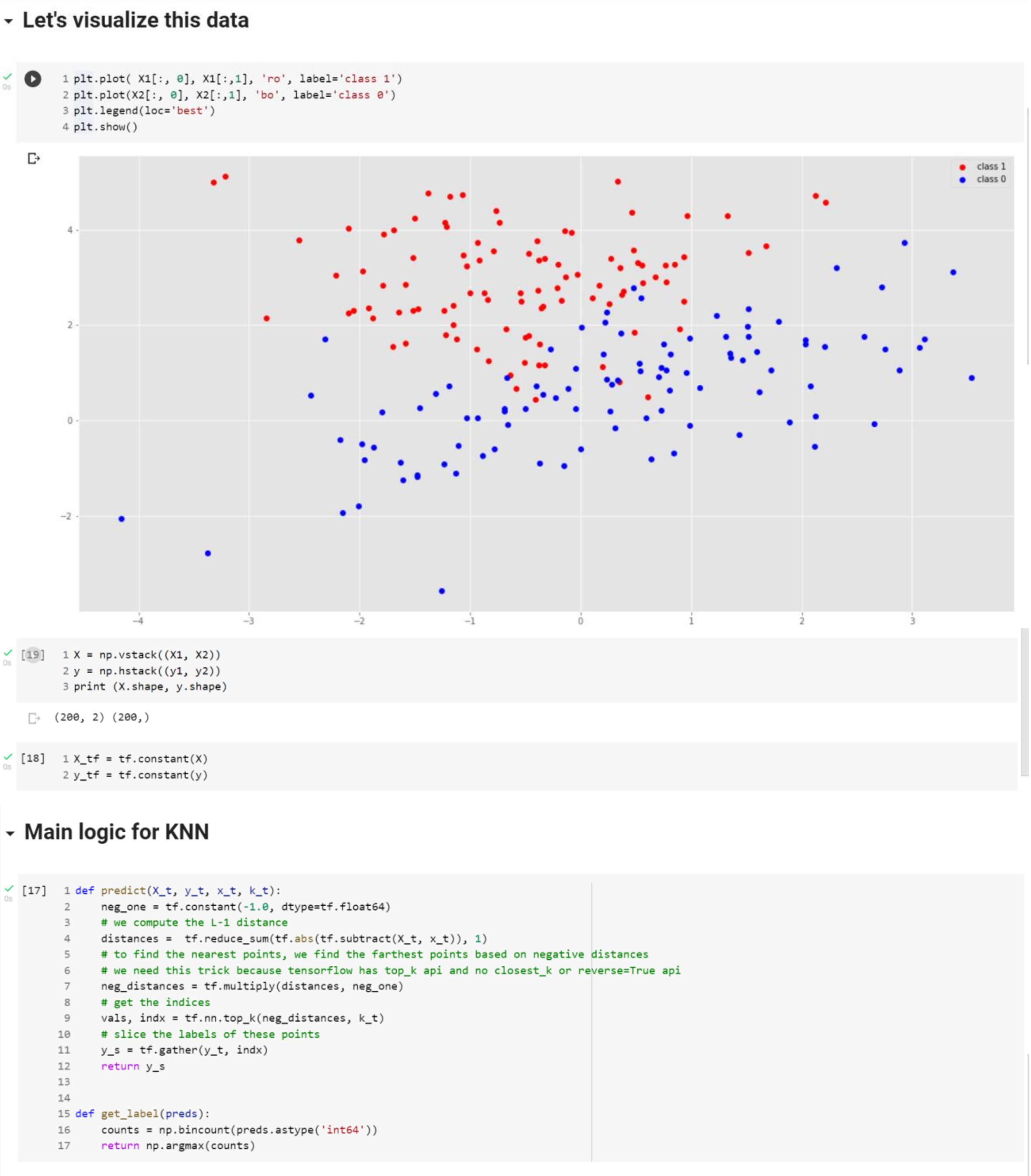
```
import numpy as np
import tensorflow.compat.v1 as tf
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (20.0, 10.0)
```

Create Synthetic data

```
num_points_each_cluster = 100
mu1 = [-0.4, 3]
covar1 = [[1.3,0],[0,1]]
mu2 = [0.5, 0.75]
covar2 = [[2.2,1.2],[1.8,2.1]]
X1 = np.random.multivariate_normal(mu1, covar1, num_points_each_cluster)
X2 = np.random.multivariate_normal(mu2, covar2, num_points_each_cluster)
y1 = np.ones(num_points_each_cluster)
y2 = np.zeros(num_points_each_cluster)
```

8) Follow the Procedure to Finish the Lab

https://hc.labnet.sfbu.edu/~henry/sfbu/course/data_science/algorithm/slide/knn.html



Let's visualize this data

```
plt.plot( X1[:, 0], X1[:,1], 'ro', label='class 1')
plt.plot(X2[:, 0], X2[:,1], 'bo', label='class 0')
plt.legend(loc='best')
plt.show()
```

[19]
```
X = np.vstack((X1, X2))
y = np.hstack((y1, y2))
print (X.shape, y.shape)
```

```
(200, 2) (200,)
```

[18]
```
X_tf = tf.constant(X)
y_tf = tf.constant(y)
```

Main logic for KNN

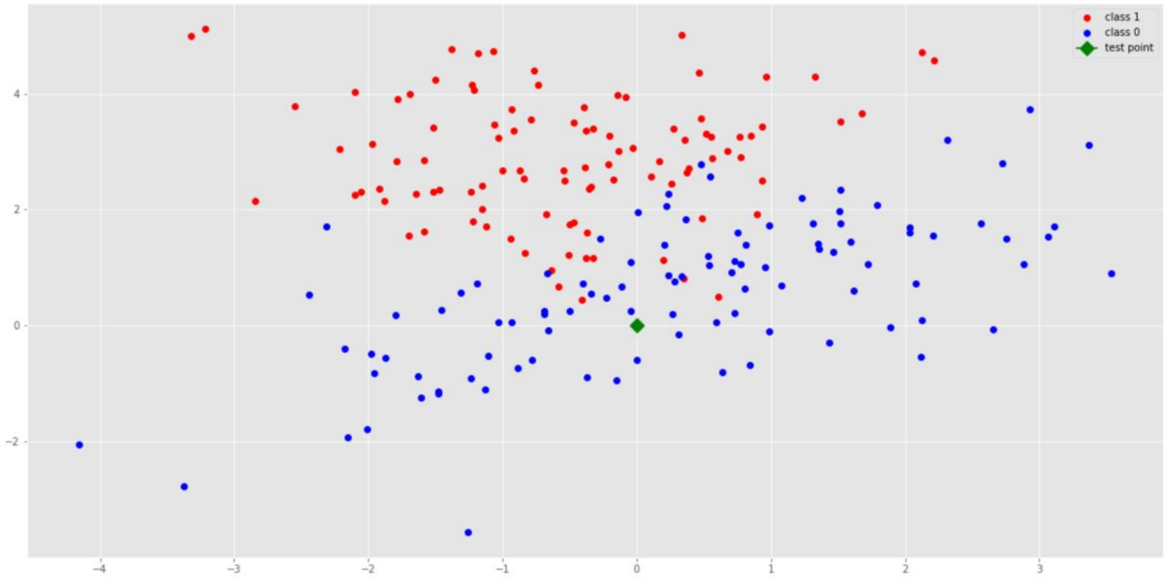
[17]
```
def predict(X_t, y_t, x_t, k_t):
    neg_one = tf.constant(-1.0, dtype=tf.float64)
    # we compute the L-1 distance
    distances =  tf.reduce_sum(tf.abs(tf.subtract(X_t, x_t)), 1)
    # to find the nearest points, we find the farthest points based on negative distances
    # we need this trick because tensorflow has top_k api and no closest_k or reverse=True api
    neg_distances = tf.multiply(distances, neg_one)
    # get the indices
    vals, indx = tf.nn.top_k(neg_distances, k_t)
    # slice the labels of these points
    y_s = tf.gather(y_t, indx)
    return y_s


def get_label(preds):
    counts = np.bincount(preds.astype('int64'))
    return np.argmax(counts)
```

Generate a test point

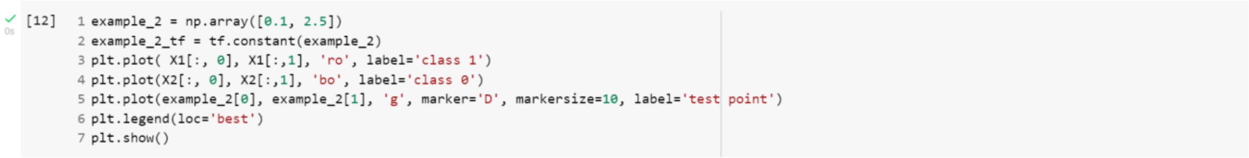
```
example = np.array([0, 0])
example_tf = tf.constant(example,dtype=tf.float64)

plt.plot( X1[:, 0], X1[:,1], 'ro', label='class 1')
plt.plot(X2[:, 0], X2[:,1], 'bo', label='class 0')
plt.plot(example[0], example[1], 'g', marker='D', markersize=10, label='test point')
plt.legend(loc='best')
plt.show()
```

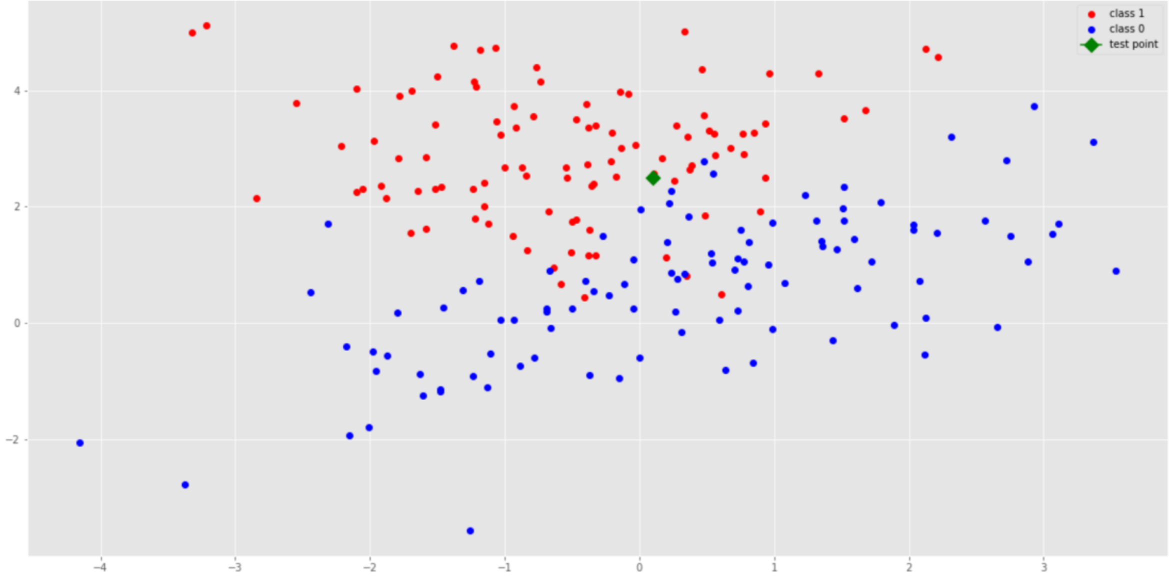


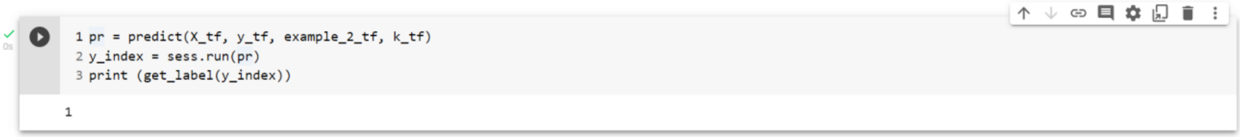
```
k_tf = tf.constant(3)
tf.disable_v2_behavior()
with tf.compat.v1.Session() as sesss:
  pr = predict(X_tf, y_tf, example_tf, k_tf)
  sess = tf.compat.v1.Session()
  y_index = sess.run(pr)
  print (get_label(y_index))
  # print(sess.run(pr))
```

```
0
```



```
example_2 = np.array([0.1, 2.5])
example_2_tf = tf.constant(example_2)
plt.plot( X1[:, 0], X1[:,1], 'ro', label='class 1')
plt.plot(X2[:, 0], X2[:,1], 'bo', label='class 0')
plt.plot(example_2[0], example_2[1], 'g', marker='D', markersize=10, label='test point')
plt.legend(loc='best')
plt.show()
```

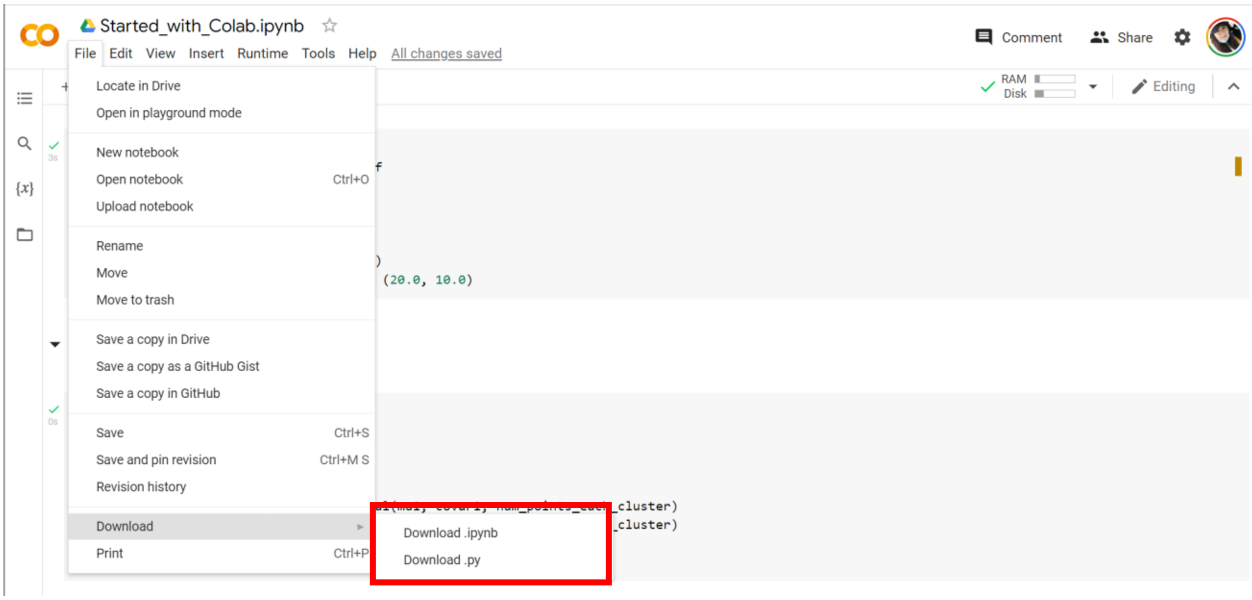




```
pr = predict(X_tf, y_tf, example_2_tf, k_tf)
y_index = sess.run(pr)
print (get_label(y_index))
```

```
1
```

9) Download File

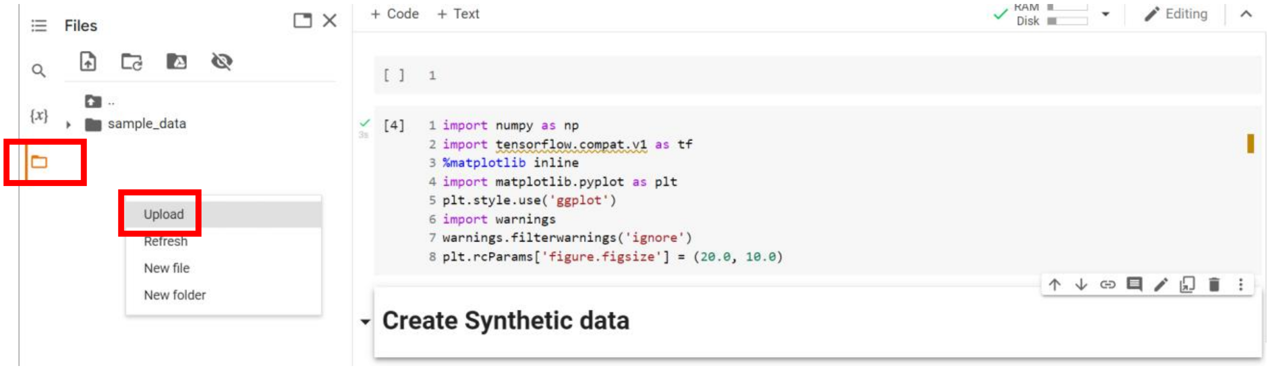


10) Download as PDF

➔ *Method 1:*

.ipynb -> html -> PDF

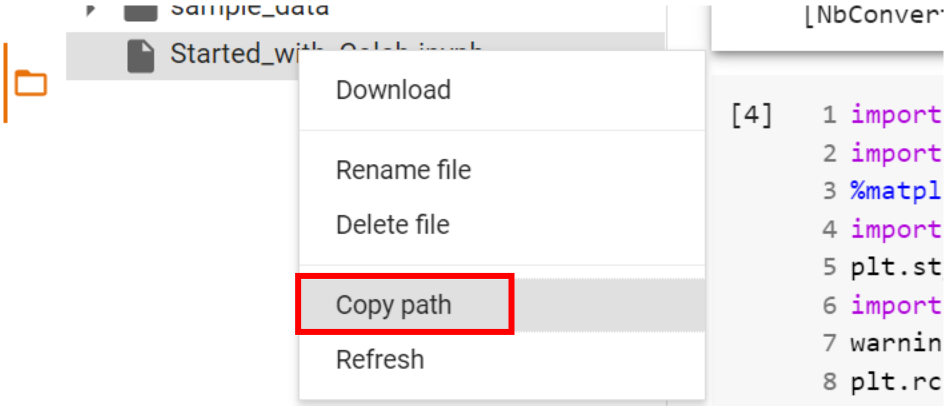
Open File Folder and Upload the .ipynb file



```
import numpy as np
import tensorflow.compat.v1 as tf
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (20.0, 10.0)
```

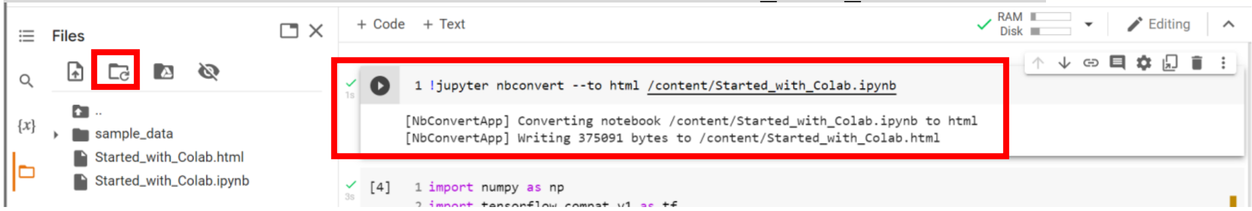
Create Synthetic data

Copy the path of the file



Run Command (!jupyter nbconvert --to html <path>)

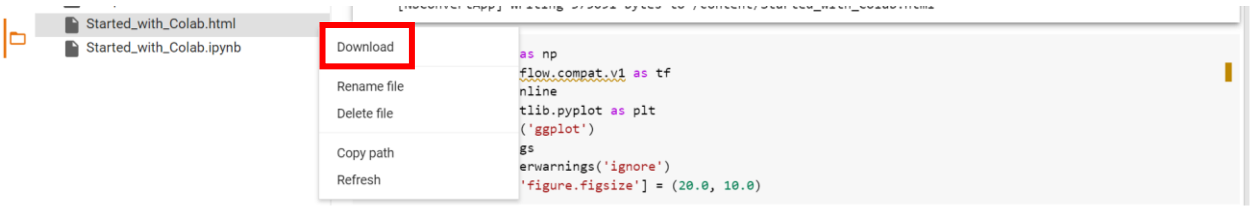
```
!jupyter nbconvert --to html /content/Started_with_Colab.ipynb
```



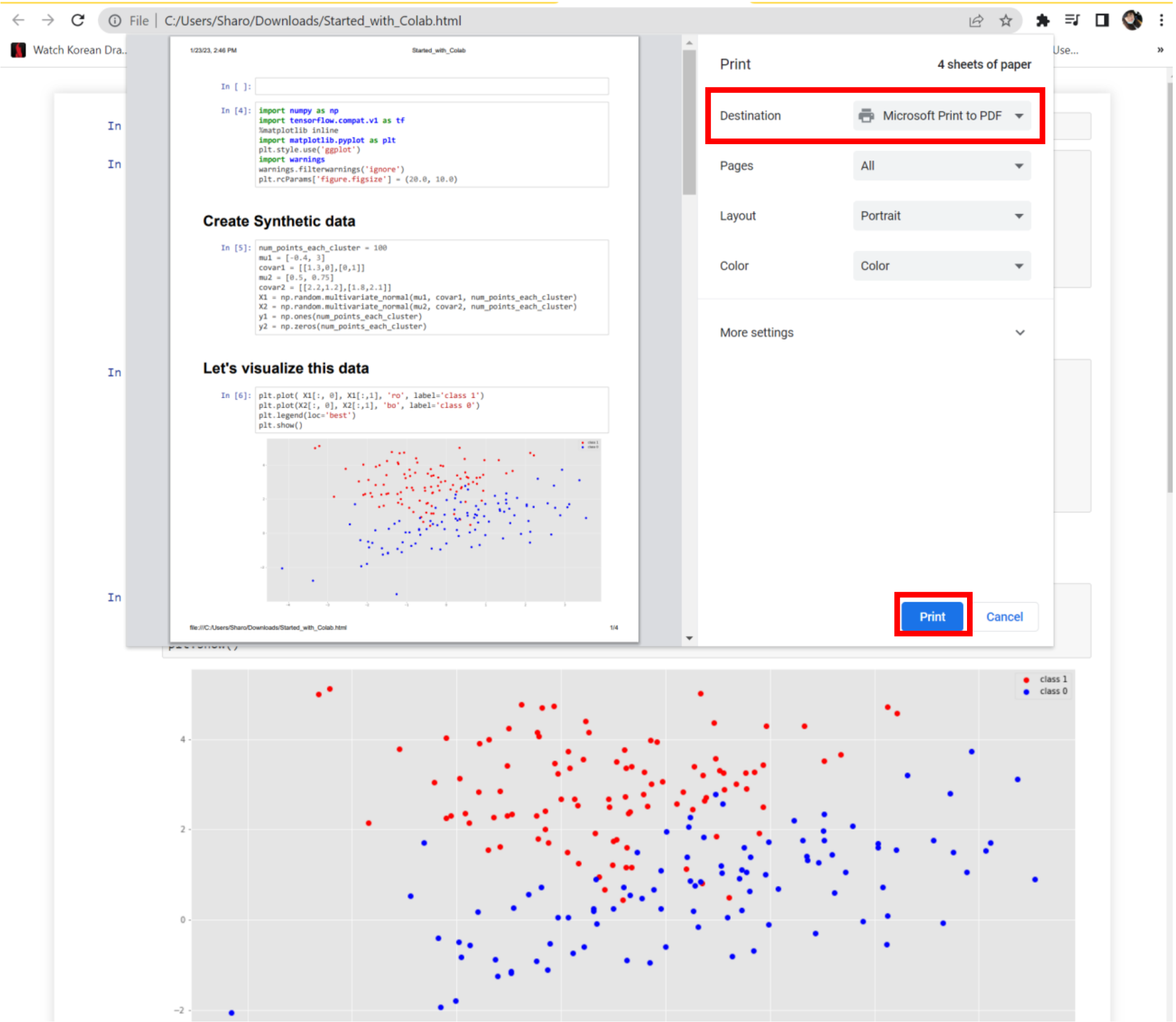
```
!jupyter nbconvert --to html /content/Started_with_Colab.ipynb
[NbConvertApp] Converting notebook /content/Started_with_Colab.ipynb to html
[NbConvertApp] Writing 375091 bytes to /content/Started_with_Colab.html
```

```
import numpy as np
```

Refresh the Folder and Download the html File



Open the html then press Ctrl+P to save as PDF



File | C:/Users/Sharo/Downloads/Started_with_Colab.html
Watch Korean Dra...
1/23/23, 2:46 PM
Started_with_Colab
file:///C:/Users/Sharo/Downloads/Started_with_Colab.html
1/4
In
class 1
class 0
4
2
0
−2

```
In [ ]:
```

```
In [4]: import numpy as np
        import tensorflow.compat.v1 as tf
        %matplotlib inline
        import matplotlib.pyplot as plt
        plt.style.use('ggplot')
        import warnings
        warnings.filterwarnings('ignore')
        plt.rcParams['figure.figsize'] = (20.0, 10.0)
```

Create Synthetic data

```
In [5]: num_points_each_cluster = 100
        mu1 = [-0.4, 3]
        covar1 = [[1.3,0],[0,1]]
        mu2 = [0.5, 0.75]
        covar2 = [[2.2,1.2],[1.8,2.1]]
        X1 = np.random.multivariate_normal(mu1, covar1, num_points_each_cluster)
        X2 = np.random.multivariate_normal(mu2, covar2, num_points_each_cluster)
        y1 = np.ones(num_points_each_cluster)
        y2 = np.zeros(num_points_each_cluster)
```

Let's visualize this data

```
In [6]: plt.plot( X1[:, 0], X1[:,1], 'ro', label='class 1')
        plt.plot(X2[:, 0], X2[:,1], 'bo', label='class 0')
        plt.legend(loc='best')
        plt.show()
```

Print — 4 sheets of paper

Destination: Microsoft Print to PDF

Pages: All

Layout: Portrait

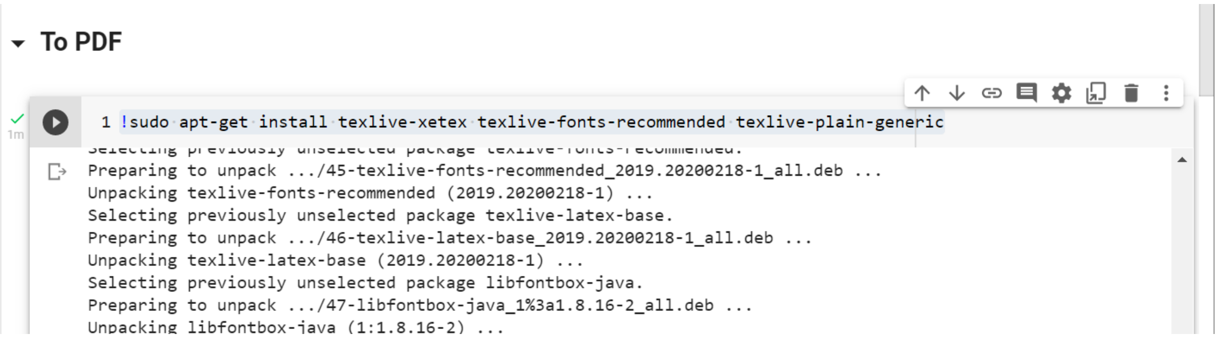
Color: Color

More settings

➔ ***Method 2:***

Install Package

```
!sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-
plain-generic
```

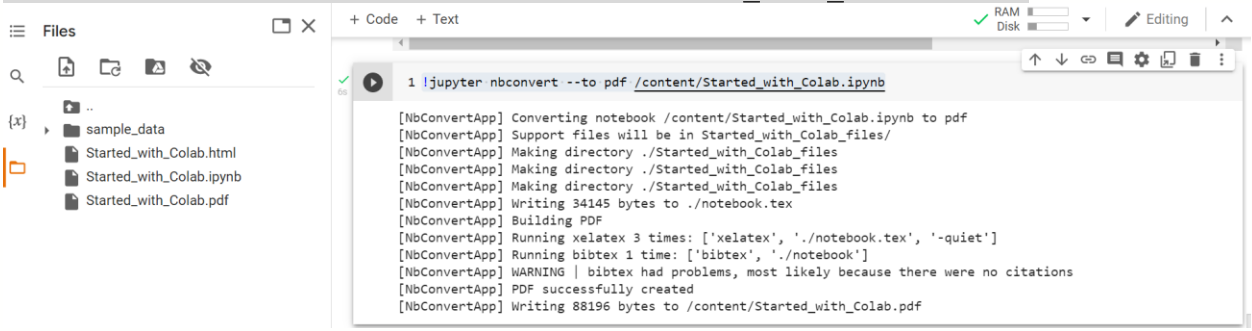


```
!sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-plain-generic
```

```
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack .../45-texlive-fonts-recommended_2019.20200218-1_all.deb ...
Unpacking texlive-fonts-recommended (2019.20200218-1) ...
Selecting previously unselected package texlive-latex-base.
Preparing to unpack .../46-texlive-latex-base_2019.20200218-1_all.deb ...
Unpacking texlive-latex-base (2019.20200218-1) ...
Selecting previously unselected package libfontbox-java.
Preparing to unpack .../47-libfontbox-java_1%3a1.8.16-2_all.deb ...
Unpacking libfontbox-java (1:1.8.16-2) ...
```

Convert

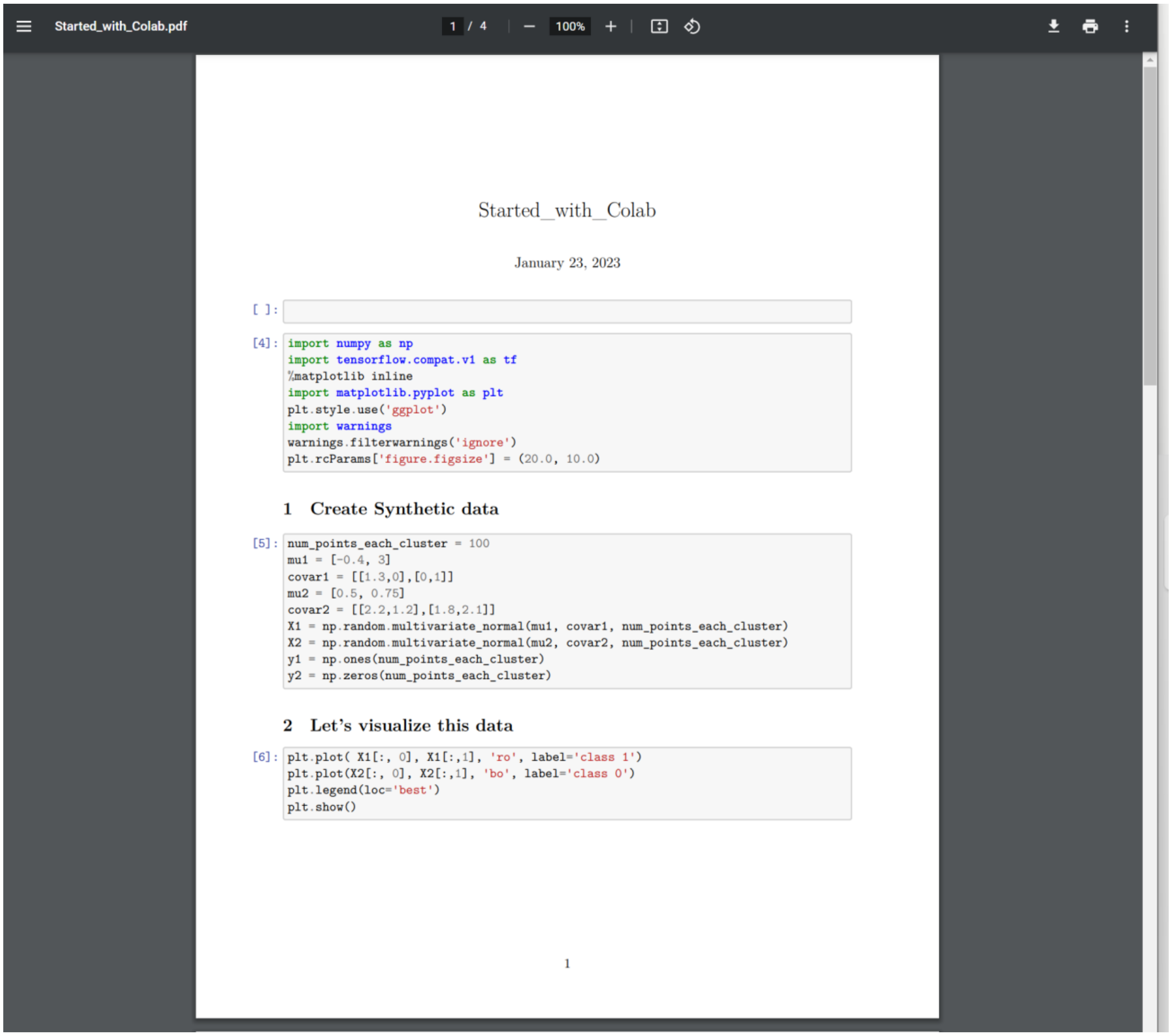
```
!jupyter nbconvert --to pdf /content/Started_with_Colab.ipynb
```



```
!jupyter nbconvert --to pdf /content/Started_with_Colab.ipynb
```

```
[NbConvertApp] Converting notebook /content/Started_with_Colab.ipynb to pdf
[NbConvertApp] Support files will be in Started_with_Colab_files/
[NbConvertApp] Making directory ./Started_with_Colab_files
[NbConvertApp] Making directory ./Started_with_Colab_files
[NbConvertApp] Making directory ./Started_with_Colab_files
[NbConvertApp] Writing 34145 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', './notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', './notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 88196 bytes to /content/Started_with_Colab.pdf
```

Refresh and Download



Started_with_Colab

January 23, 2023

[ ]:

[4]:

```
import numpy as np
import tensorflow.compat.v1 as tf
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (20.0, 10.0)
```

1 Create Synthetic data

[5]:

```
num_points_each_cluster = 100
mu1 = [-0.4, 3]
covar1 = [[1.3,0],[0,1]]
mu2 = [0.5, 0.75]
covar2 = [[2.2,1.2],[1.8,2.1]]
X1 = np.random.multivariate_normal(mu1, covar1, num_points_each_cluster)
X2 = np.random.multivariate_normal(mu2, covar2, num_points_each_cluster)
y1 = np.ones(num_points_each_cluster)
y2 = np.zeros(num_points_each_cluster)
```

2 Let's visualize this data

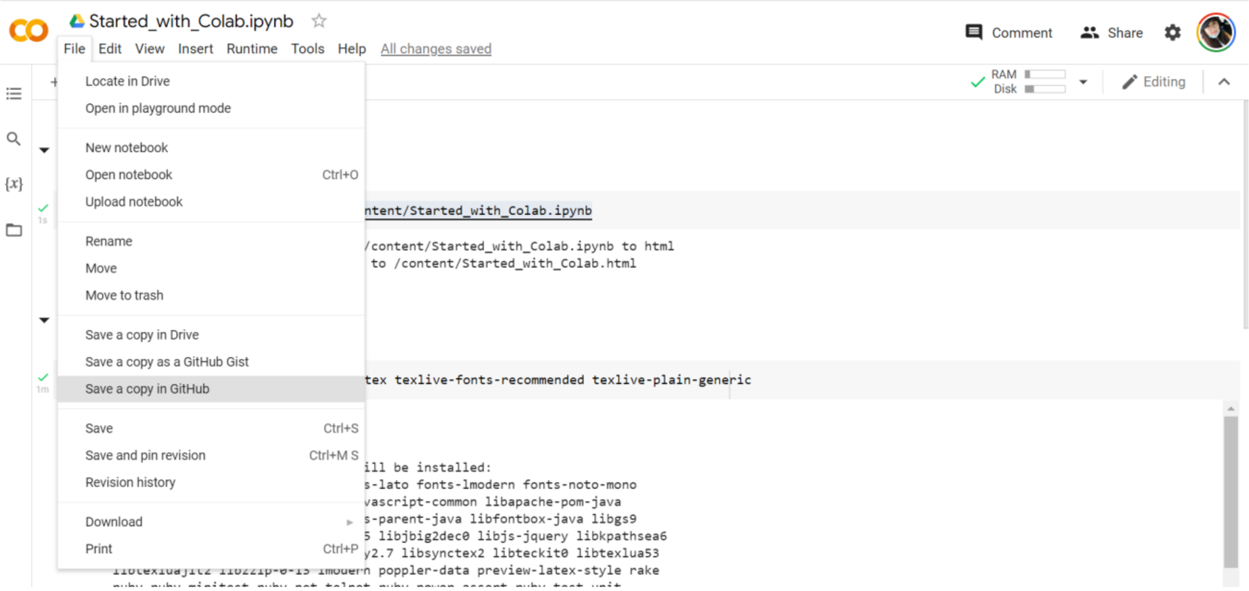
[6]:

```
plt.plot( X1[:, 0], X1[:,1], 'ro', label='class 1')
plt.plot(X2[:, 0], X2[:,1], 'bo', label='class 0')
plt.legend(loc='best')
plt.show()
```

1

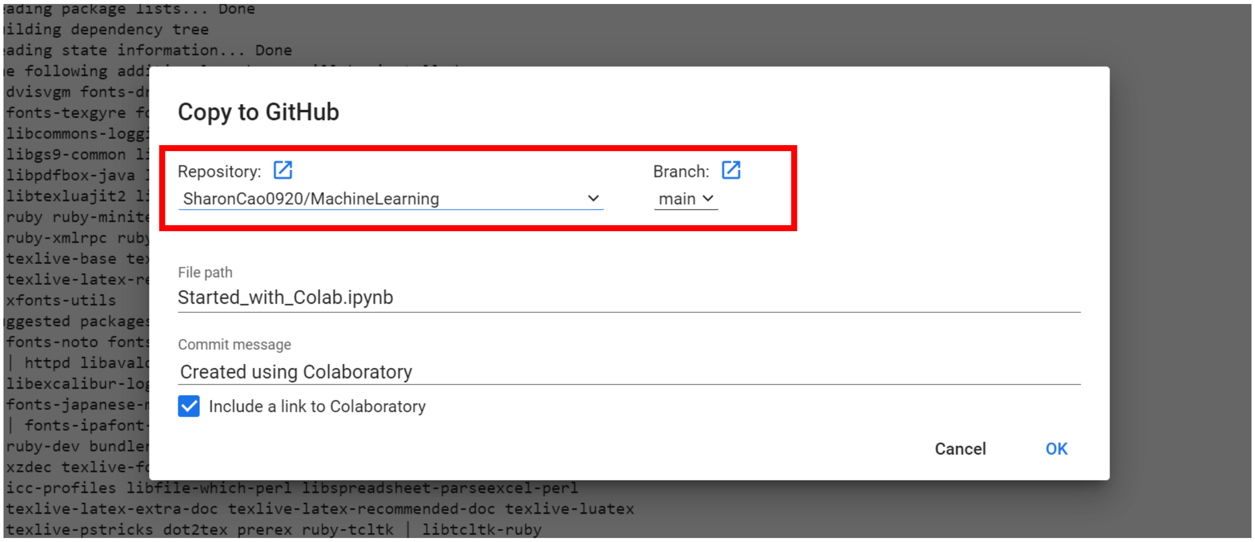
11) Save on GitHub

https://github.com/SharonCao0920/MachineLearning/blob/main/Started_with_Colab.ipynb



```
ntent/Started_with_Colab.ipynb
/content/Started_with_Colab.ipynb to html
 to /content/Started_with_Colab.html
tex texlive-fonts-recommended texlive-plain-generic
ill be installed:
s-lato fonts-lmodern fonts-noto-mono
vascript-common libapache-pom-java
s-parent-java libfontbox-java libgs9
5 libjbig2dec0 libjs-jquery libkpathsea6
y2.7 libsynctex2 libteckit0 libtexlua53
libtexluajit2 libzzip-0-13 lmodern poppler-data preview-latex-style rake
```

Select Repository



```
ading package lists... Done
ilding dependency tree
ading state information... Done
e following addi
dvisvgm fonts-dr
fonts-texgyre fo
libcommons-logg
libgs9-common li
libpdfbox-java l
libtexluajit2 li
ruby ruby-minite
ruby-xmlrpc ruby
texlive-base tex
texlive-latex-re
xfonts-utils
ggested package
fonts-noto font
| httpd libavalc
libexcalibur-log
fonts-japanese-m
| fonts-ipafont-
ruby-dev bundler
xzdec texlive-fo
icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl
texlive-latex-extra-doc texlive-latex-recommended-doc texlive-luatex
texlive-pstricks dot2tex prerex ruby-tcltk | libtcltk-ruby
```

Copy to GitHub

Repository: SharonCao0920/MachineLearning

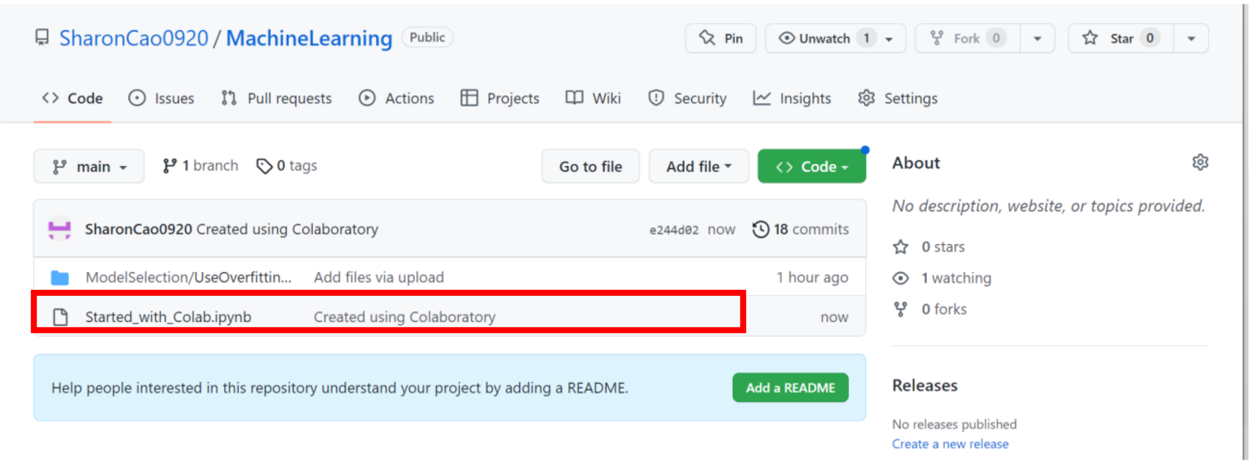
Branch: main

File path: Started_with_Colab.ipynb

Commit message: Created using Colaboratory

☑ Include a link to Colaboratory

Verify



SharonCao0920 Created using Colaboratory e244d02 now 18 commits

ModelSelection/UseOverfittin... Add files via upload 1 hour ago

Started_with_Colab.ipynb Created using Colaboratory now

Help people interested in this repository understand your project by adding a README.

No description, website, or topics provided.

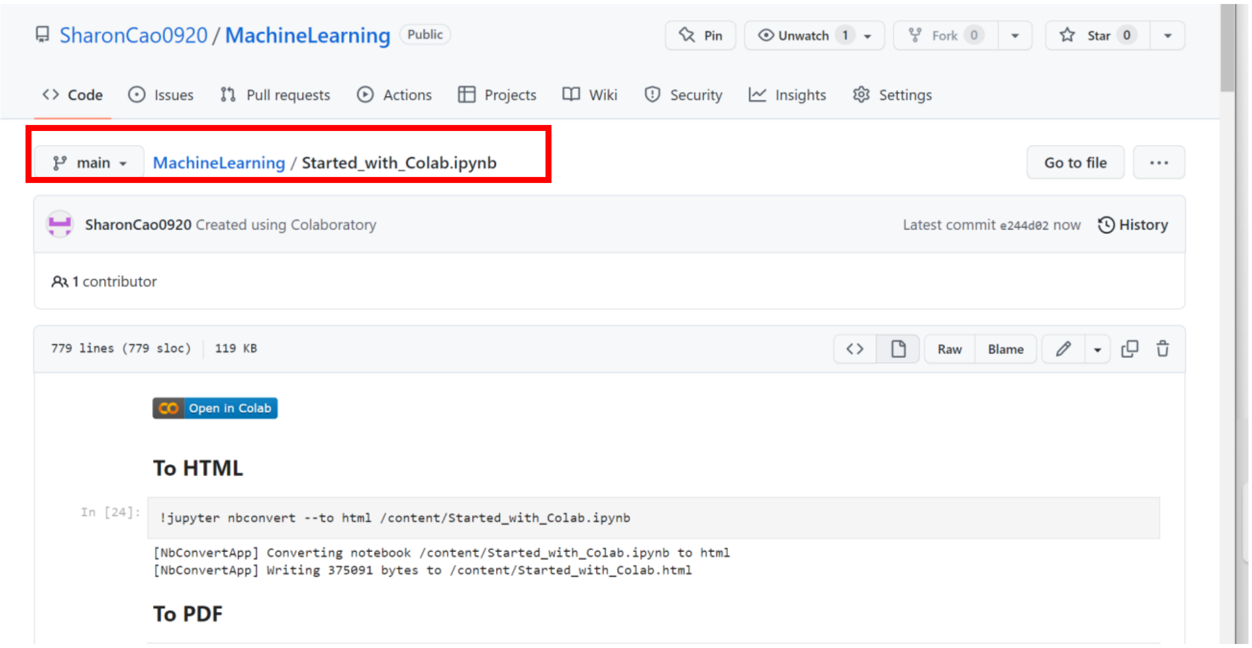
0 stars

1 watching

0 forks

No releases published

Create a new release



MachineLearning / Started_with_Colab.ipynb

SharonCao0920 Created using Colaboratory Latest commit e244d02 now

1 contributor

779 lines (779 sloc) 119 KB

To HTML

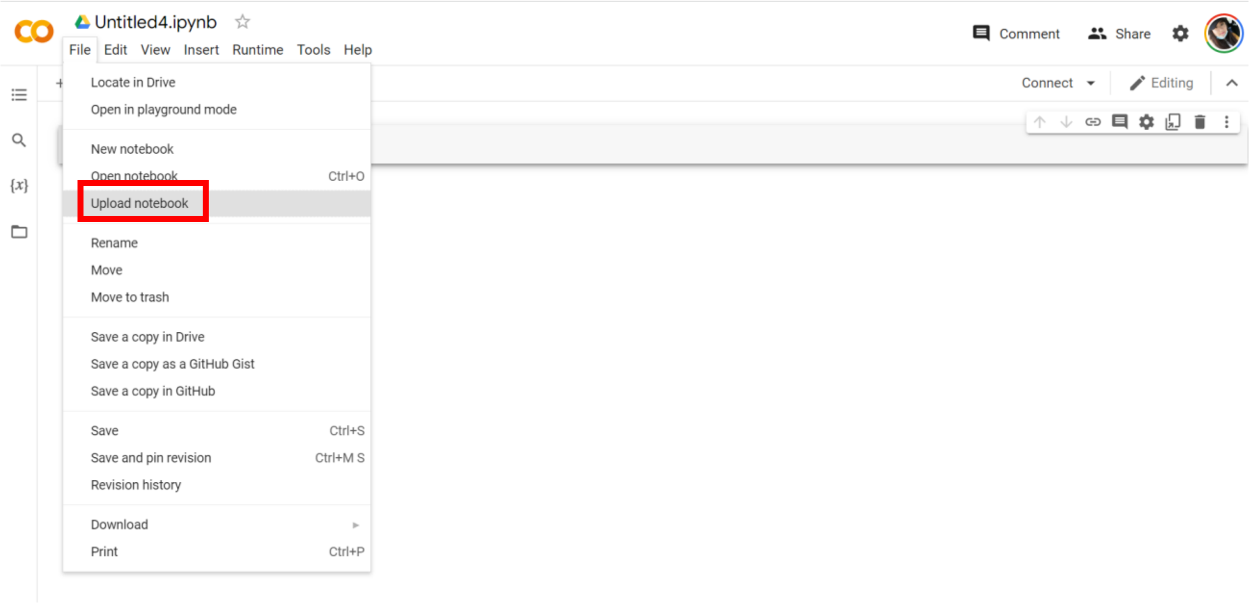
```
In [24]: !jupyter nbconvert --to html /content/Started_with_Colab.ipynb
```

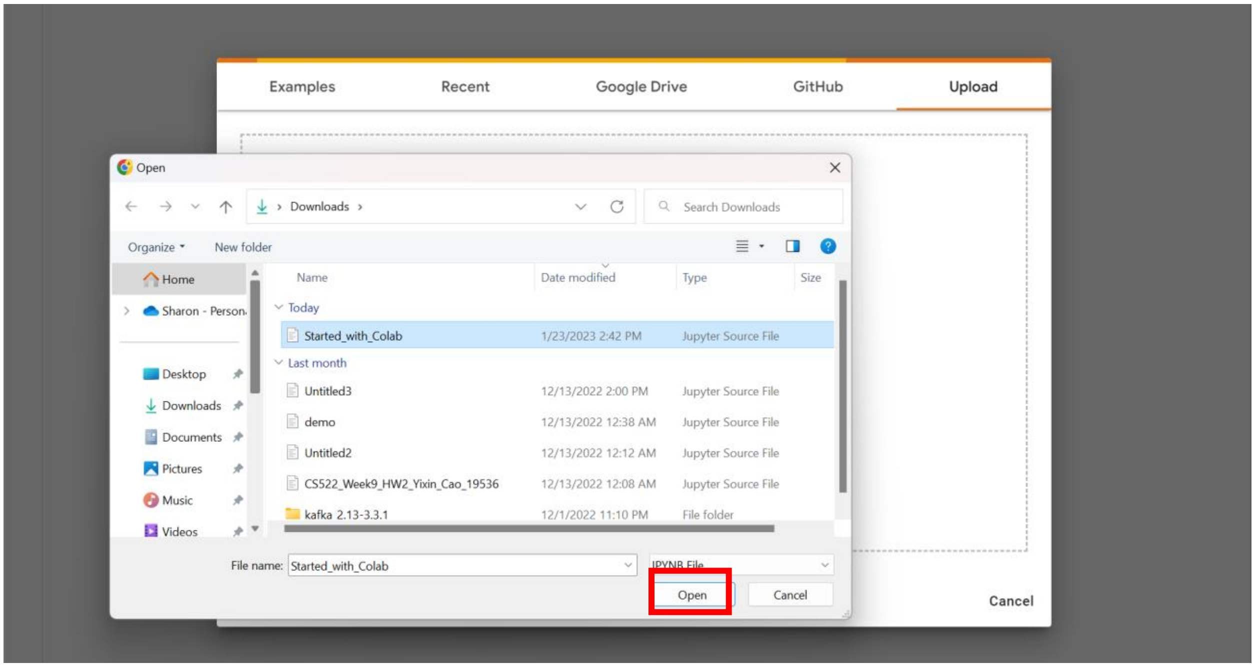
```
[NbConvertApp] Converting notebook /content/Started_with_Colab.ipynb to html
[NbConvertApp] Writing 375091 bytes to /content/Started_with_Colab.html
```

To PDF

Step 2: Modify an existing Colab Project

➔ File-> Upload Notebook

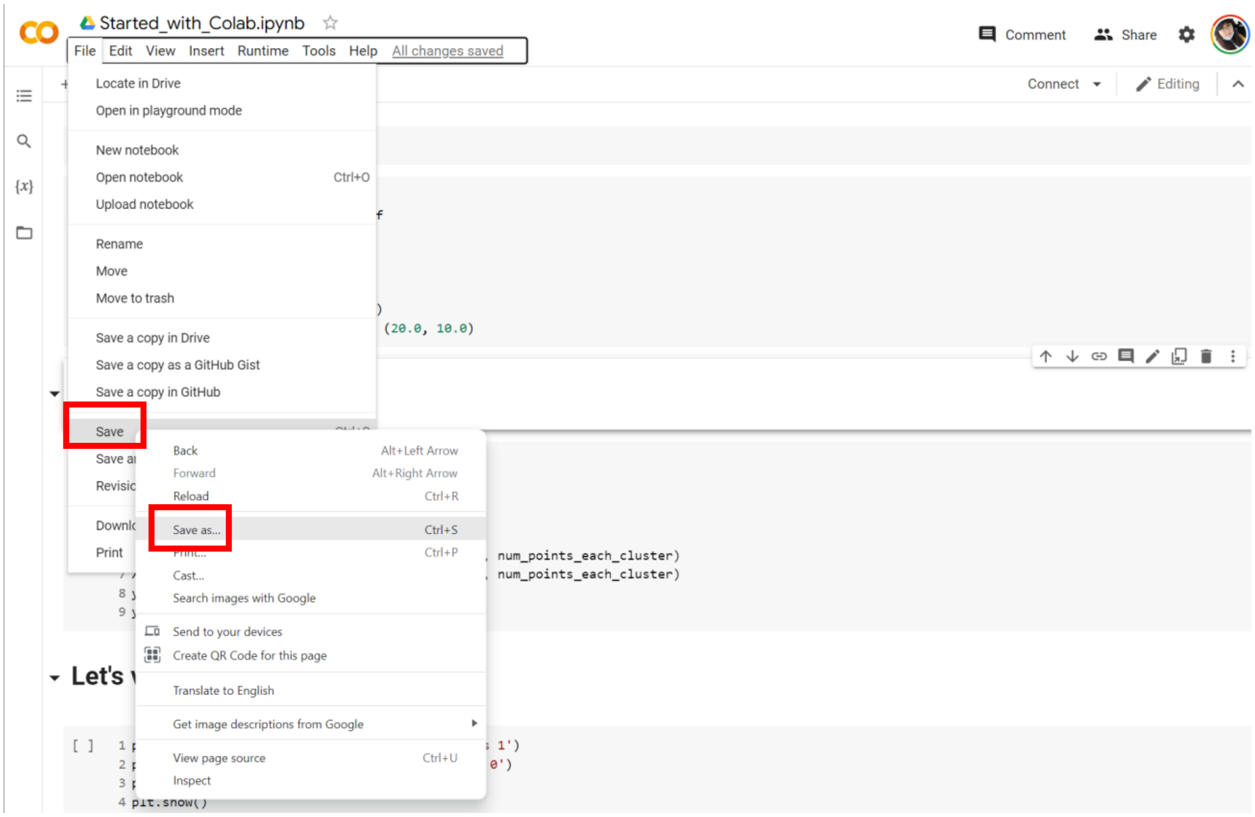


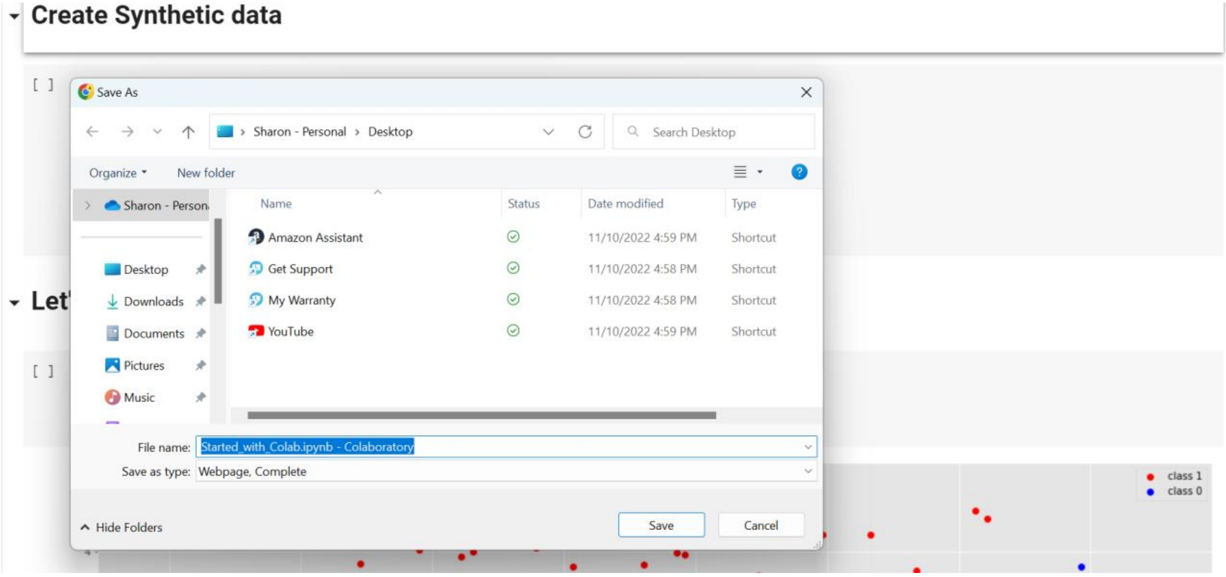


Name	Date modified	Type	Size
Today			
Started_with_Colab	1/23/2023 2:42 PM	Jupyter Source File	
Last month			
Untitled3	12/13/2022 2:00 PM	Jupyter Source File	
demo	12/13/2022 12:38 AM	Jupyter Source File	
Untitled2	12/13/2022 12:12 AM	Jupyter Source File	
CS522_Week9_HW2_Yixin_Cao_19536	12/13/2022 12:08 AM	Jupyter Source File	
kafka_2.13-3.3.1	12/1/2022 11:10 PM	File folder	

File name: Started_with_Colab

- ➔ Save and download modified file as in Step 1
- ➔ Save as html (right click)

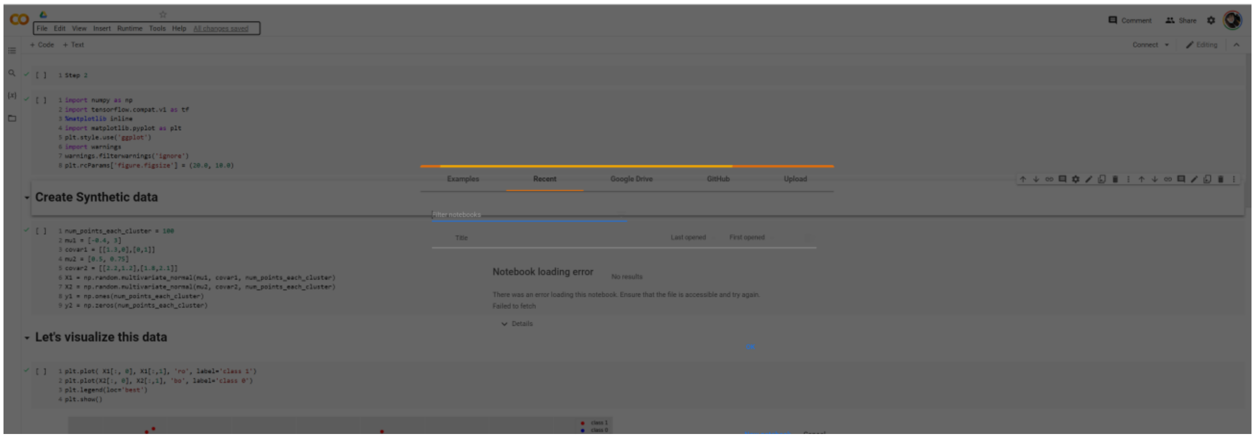



Create Synthetic data
Save As
Sharon - Personal › Desktop
Search Desktop
Organize
New folder
Sharon - Personal
Desktop
Downloads
Documents
Pictures
Music
Name
Status
Date modified
Type
Amazon Assistant 11/10/2022 4:59 PM Shortcut
Get Support 11/10/2022 4:58 PM Shortcut
My Warranty 11/10/2022 4:58 PM Shortcut
YouTube 11/10/2022 4:59 PM Shortcut
Hide Folders
Save
Cancel
class 1
class 0

File name: Started_with_Colab.ipynb - Colaboratory

Save as type: Webpage, Complete

(This method the html file is not open correctly)



```
Step 2
```

```
import numpy as np
import tensorflow.compat.v1 as tf
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (20.0, 10.0)
```

Create Synthetic data

```
num_points_each_cluster = 100
mu1 = [-0.4, 3]
covar1 = [[1.3,0],[0,1]]
mu2 = [0.5, 0.75]
covar2 = [[2.2,1.2],[1.8,2.1]]
X1 = np.random.multivariate_normal(mu1, covar1, num_points_each_cluster)
X2 = np.random.multivariate_normal(mu2, covar2, num_points_each_cluster)
y1 = np.ones(num_points_each_cluster)
y2 = np.zeros(num_points_each_cluster)
```

Let's visualize this data

```
plt.plot( X1[:, 0], X1[:,1], 'ro', label='class 1')
plt.plot(X2[:, 0], X2[:,1], 'bo', label='class 0')
plt.legend(loc='best')
plt.show()
```

Notebook loading error

There was an error loading this notebook. Ensure that the file is accessible and try again.
Failed to fetch