

# Introduction to Google Colab & PyTorch

Ju Sun

Computer Science and Engineering

Mar 2025



UNIVERSITY OF MINNESOTA

Driven to Discover<sup>SM</sup>

# Outline

- **Google Colab**
- Basics of PyTorch

# Google Colab

<https://colab.research.google.com/>



Pay As You Go	Recommended Colab Pro	Colab Pro+	Colab Enterprise
<p><a href="#">\$9.99 for 100 Compute Units</a></p> <p><a href="#">\$49.99 for 500 Compute Units</a></p> <p>You currently have 0 compute units. Compute units expire after 90 days. Purchase more as you need them.</p> <ul style="list-style-type: none"><li>✓ No subscription required. Only pay for what you use.</li><li>✓ Faster GPUs Upgrade to more powerful GPUs.</li></ul>	<p><a href="#">\$9.99 per month</a></p> <ul style="list-style-type: none"><li>✓ 100 compute units per month Compute units expire after 90 days. Purchase more as you need them.</li><li>✓ Faster GPUs Upgrade to more powerful GPUs.</li><li>✓ More memory Access our highest memory machines.</li><li>✓ Terminal Ability to use a terminal with the connected VM.</li></ul> <p>Select countries and 18+ only:</p> <ul style="list-style-type: none"><li>✓ AI-enabled autocompletions Intelligent multi-line suggestions automatically rendered while you type.</li><li>✓ Code generation Generate code with natural language, including an integrated chatbot.</li></ul>	<p><a href="#">\$49.99 per month</a></p> <p>All of the benefits of Pro, plus:</p> <ul style="list-style-type: none"><li>✓ An additional 400 compute units for a total of 500 per month. Compute units expire after 90 days. Purchase more as you need them.</li><li>✓ Faster GPUs Priority access to upgrade to more powerful premium GPUs.</li><li>✓ Background execution With compute units, your actively running notebook will continue running for up to 24hrs, even if you close your browser.</li></ul>	<p><a href="#">Pay for what you use</a></p> <ul style="list-style-type: none"><li>✓ Integrated Tightly integrated with Google Cloud services like BigQuery and Vertex AI.</li><li>✓ Enterprise notebook storage Replace your usage of Google Drive notebooks with GCP notebooks, stored and shared within your cloud console.</li><li>✓ Productive Generative AI powered code completion and generation.</li></ul>

<https://colab.research.google.com/signup>

Everyone registered to the class receives **3-month subscription to Colab Pro**,

# Create a new notebook

<https://colab.research.google.com/>

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share Settings Ju

RAM Disk

ed code history view,

Table of contents

Getting started

Data science

Machine learning

More Resources

Featured exam

Section

Open notebook

Examples >

Recent >

Google Drive >

GitHub >

Upload >

Search notebooks

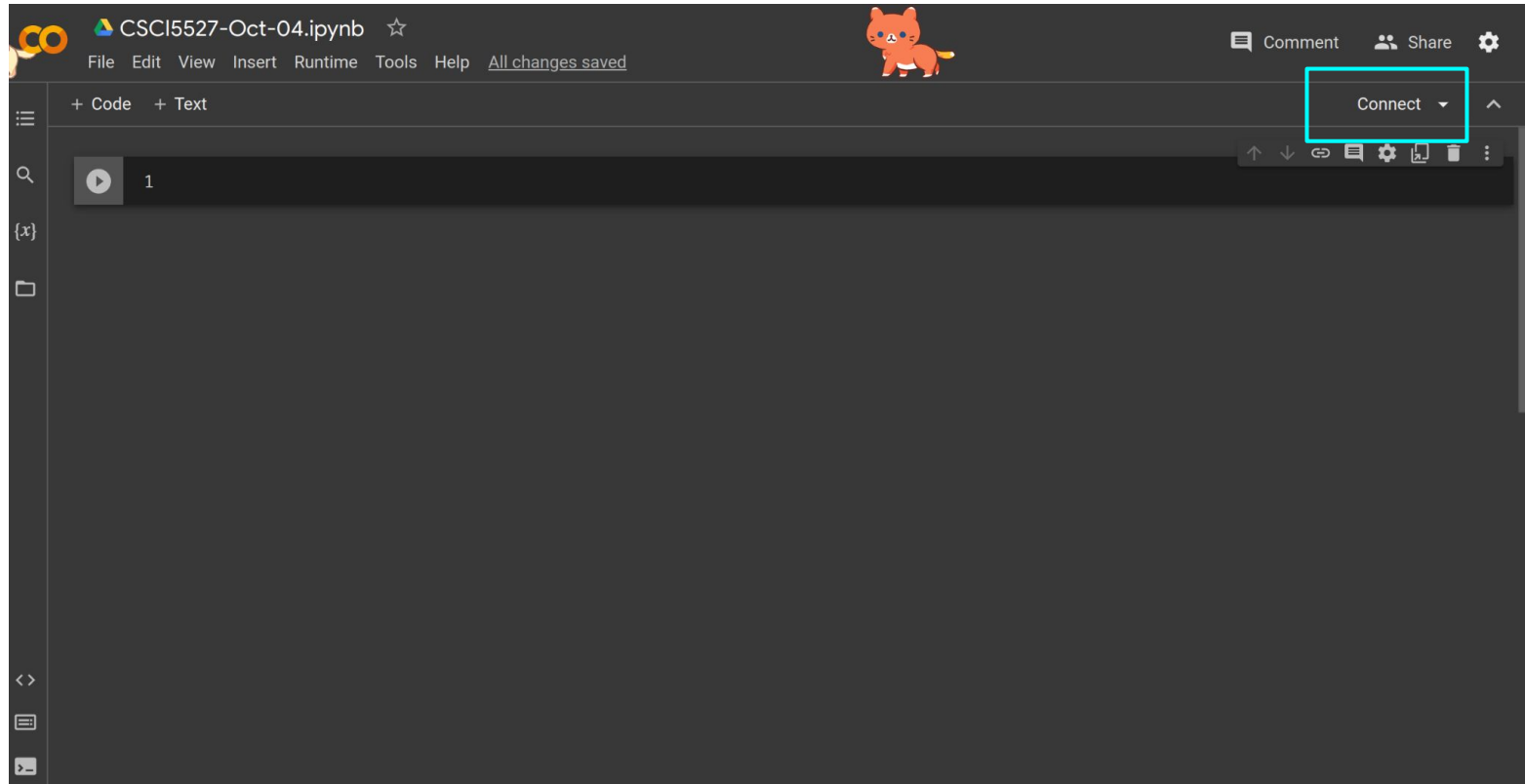
Title	Last opened	First opened	
CSCI5527-Oct-04.ipynb	3:25 PM	3:25 PM	
Welcome To Colaboratory	3:24 PM	Feb 27, 2020	
Copy of Problem1_A3.ipynb	September 7	September 7	
Problem1_A2.ipynb	September 7	March 15	
FFT_Flip.ipynb	June 30	Apr 3, 2022	

+ New notebook

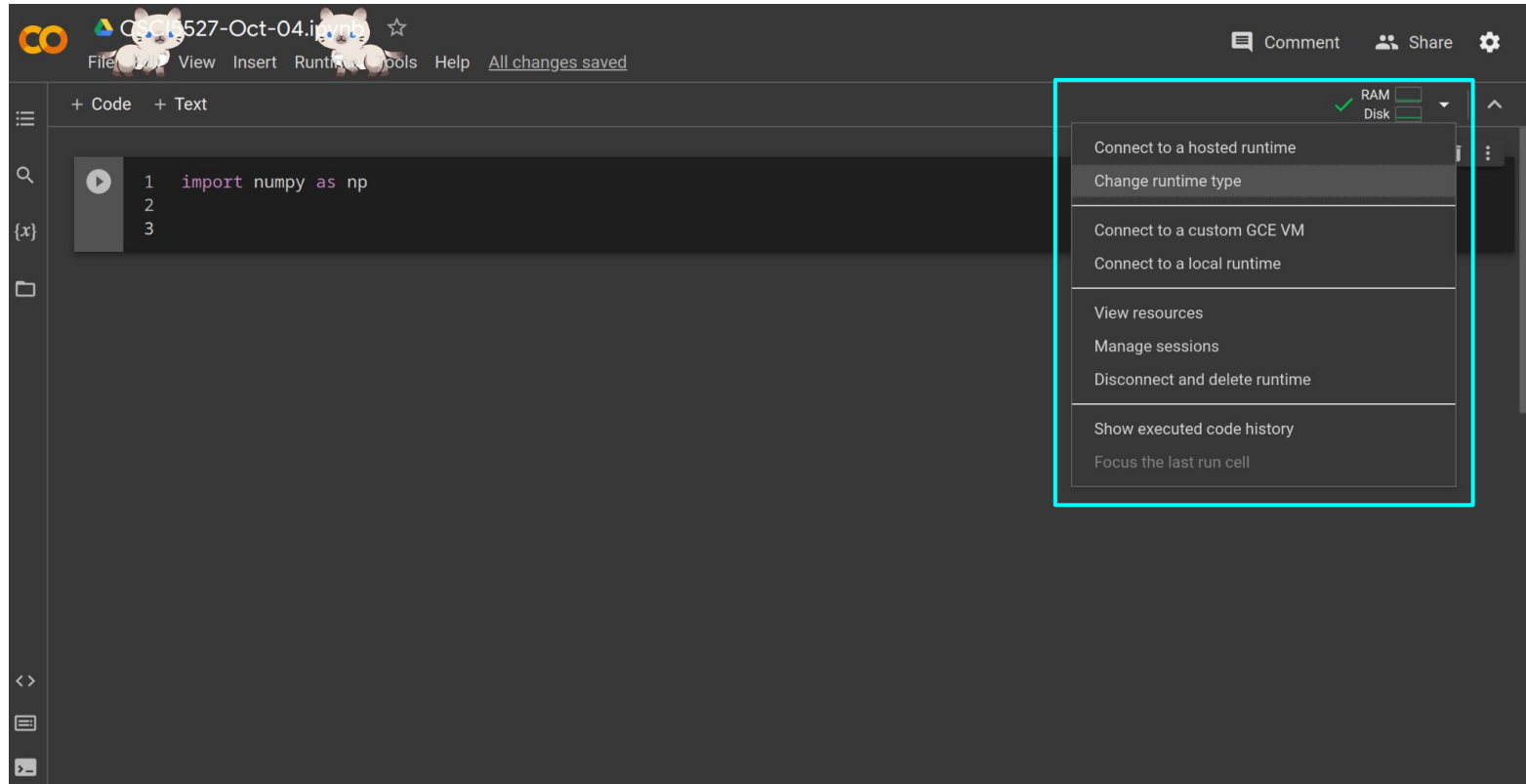
Cancel

• Easy sharing

# Connect to a runtime session



# Change runtime type



The screenshot shows the JupyterLab interface. At the top, there's a header with the Google Colab logo, a file name '527-Oct-04.ipynb', and a star icon. Below the header is a menu bar with 'File', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. The 'Runtime' menu is open, showing a dropdown menu with the following options:

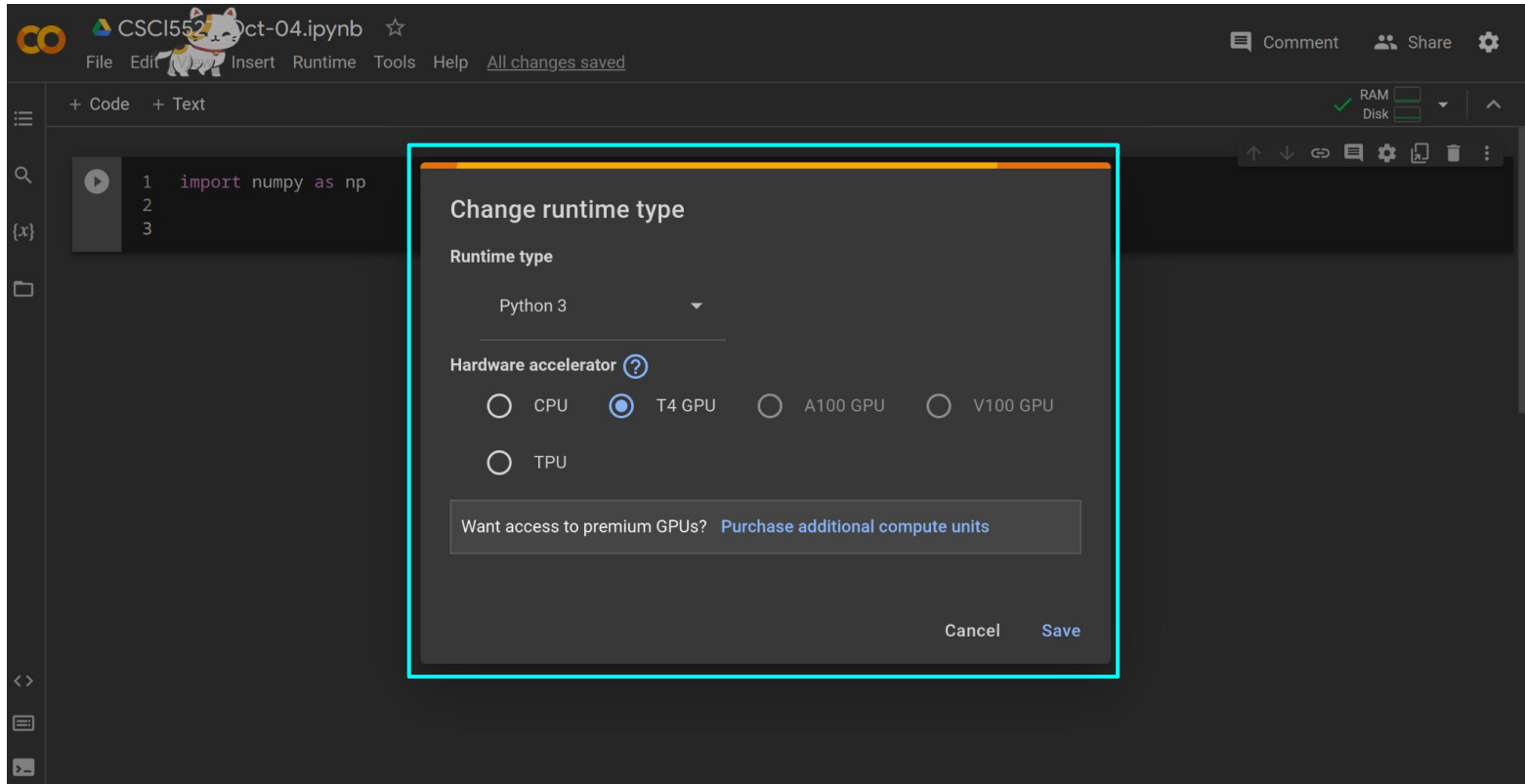
- Connect to a hosted runtime
- Change runtime type
- Connect to a custom GCE VM
- Connect to a local runtime
- View resources
- Manage sessions
- Disconnect and delete runtime
- Show executed code history
- Focus the last run cell

The code cell contains the following Python code:

```
1 import numpy as np
2
3
```

The 'Change runtime type' option is highlighted in the dropdown menu. The interface also shows a sidebar on the left with icons for home, search, and file management, and a bottom toolbar with navigation and execution icons.

# Choose GPU's to speed up



The screenshot displays the JupyterLab interface for a notebook named 'CSCI552-act-04.ipynb'. The main area shows a code cell with the following content:

```
1 import numpy as np
2
3
```

A modal dialog box titled 'Change runtime type' is open in the center. It features a dropdown menu for 'Runtime type' currently set to 'Python 3'. Below this, the 'Hardware accelerator' section is visible, with a help icon (?). The available options are:

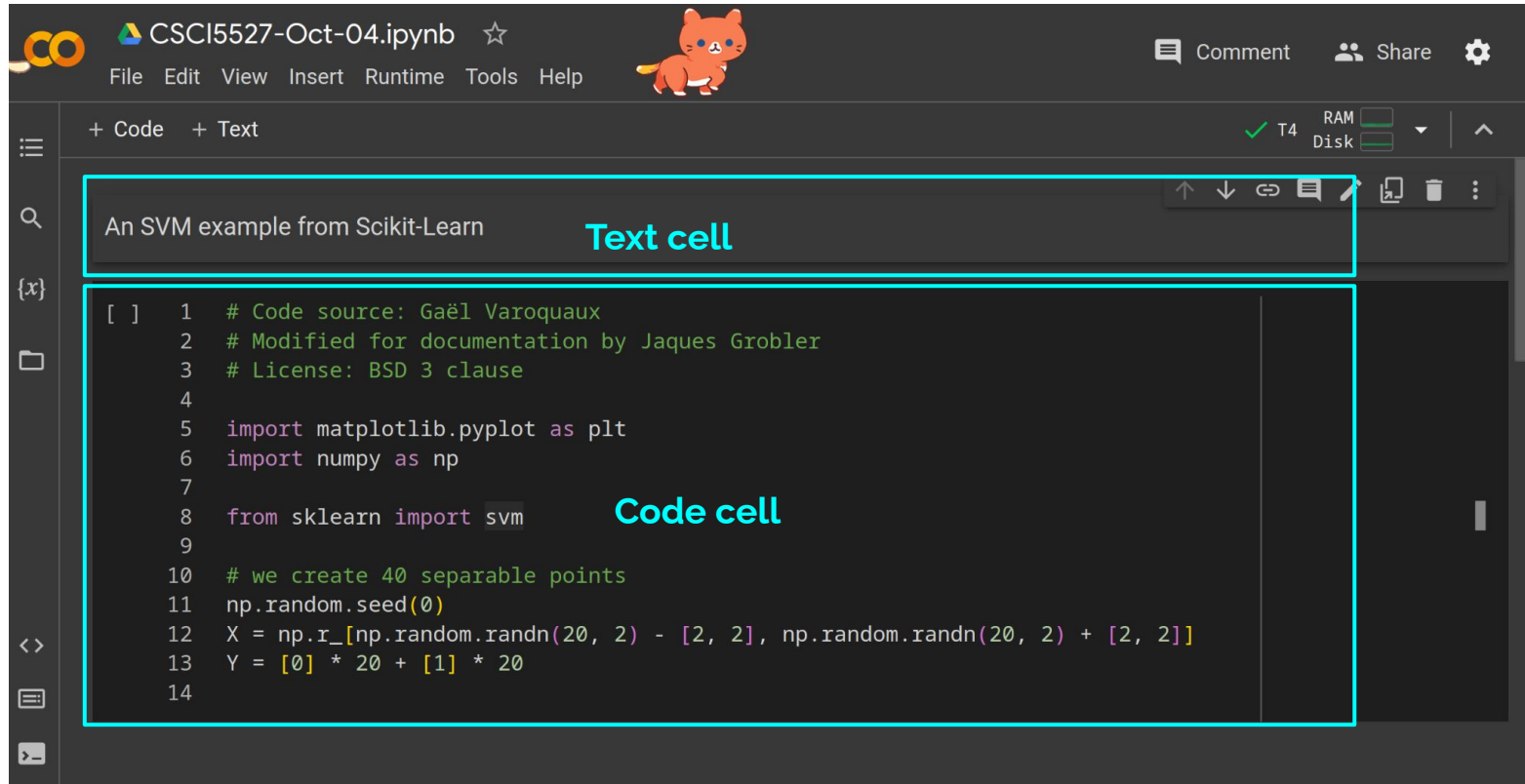
- CPU
- T4 GPU
- A100 GPU
- V100 GPU
- TPU

At the bottom of the dialog, there is a link: 'Want access to premium GPUs? [Purchase additional compute units](#)'. The dialog has 'Cancel' and 'Save' buttons at the bottom right.

# Text cells and code cells

Text cells use Markdown syntax

[https://colab.research.google.com/notebooks/markdown\\_guide.ipynb](https://colab.research.google.com/notebooks/markdown_guide.ipynb)



The screenshot displays a Google Colab notebook interface. At the top, the notebook title is "CSCI5527-Oct-04.ipynb". The menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A small orange cat icon is visible in the top right corner. The interface shows two cells:

- Text cell:** The first cell is a text cell containing the text "An SVM example from Scikit-Learn". It is labeled "Text cell" in cyan text. The cell has a toolbar with icons for up, down, refresh, comment, and edit.
- Code cell:** The second cell is a code cell containing Python code for generating separable points. It is labeled "Code cell" in cyan text. The code is as follows:

```
[ ] 1 # Code source: Gaël Varoquaux
2 # Modified for documentation by Jaques Grobler
3 # License: BSD 3 clause
4
5 import matplotlib.pyplot as plt
6 import numpy as np
7
8 from sklearn import svm
9
10 # we create 40 separable points
11 np.random.seed(0)
12 X = np.r_[np.random.randn(20, 2) - [2, 2], np.random.randn(20, 2) + [2, 2]]
13 Y = [0] * 20 + [1] * 20
14
```

The interface also shows a sidebar on the left with icons for file explorer, search, and other notebook functions. The top right corner displays system information: "T4", "RAM", and "Disk" usage.

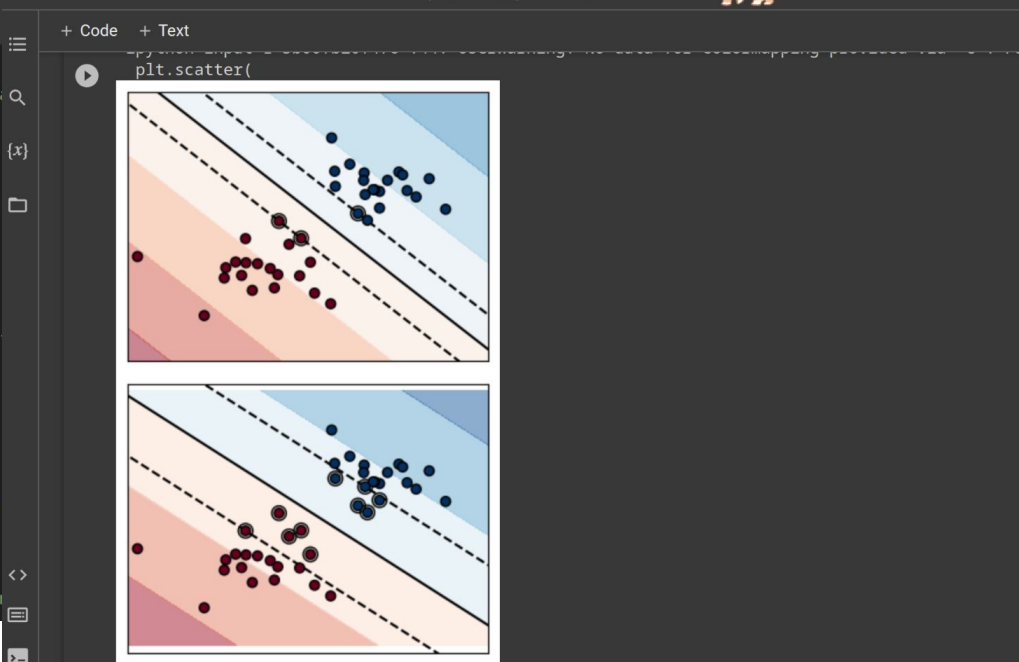
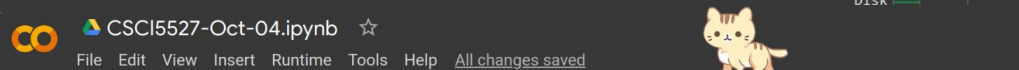


# Click and run



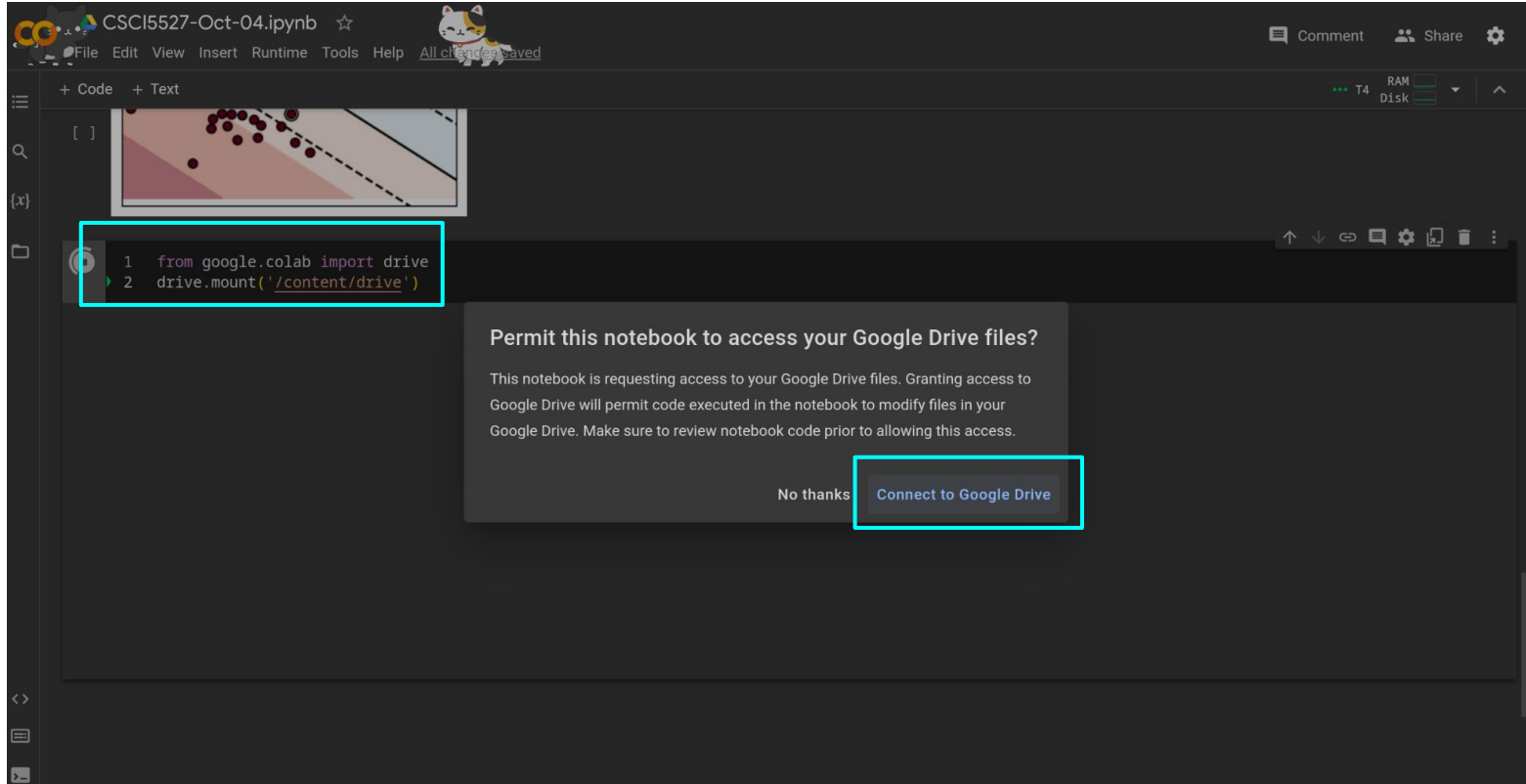
```
An SVM example from Scikit-Learn

1 # Code source: Gaël Varoquaux
2 # Modified for documentation by J.
3 # License: BSD 3 clause
4
5 import matplotlib.pyplot as plt
6 import numpy as np
7
8 from sklearn import svm
9
10 # we create 40 separable points
11 np.random.seed(0)
12 X = np.r_[np.random.randn(20, 2)
13          Y = [0] * 20 + [1] * 20
14
15 # figure number
16 fignum = 1
17
18 # fit the model
19 for name, penalty in (("unreg", 1
20                      clf = svm.SVC(kernel="linear"
21                      clf.fit(X, Y)
22
23 # get the separating hyperplane
```



# External data

<https://colab.research.google.com/notebooks/io.ipynb>



The screenshot shows a Google Colab notebook titled "CSCI5527-Oct-04.ipynb". The interface includes a top menu bar with options like File, Edit, View, Insert, Runtime, Tools, and Help. A code cell is active, containing the following Python code:

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

Below the code cell, a modal dialog box is displayed with the title "Permit this notebook to access your Google Drive files?". The dialog contains the following text:

This notebook is requesting access to your Google Drive files. Granting access to Google Drive will permit code executed in the notebook to modify files in your Google Drive. Make sure to review notebook code prior to allowing this access.

At the bottom of the dialog, there are two buttons: "No thanks" and "Connect to Google Drive".

# Google drive accessible from the session

The screenshot displays a Google Colab notebook titled "CSCI5527-Oct-04.ipynb". The interface includes a top navigation bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help" menus. A file explorer on the left shows a "drive" folder containing a "MyDrive" subfolder, which is highlighted with a red box. Below the file explorer, a terminal window is open, showing the following code and output:

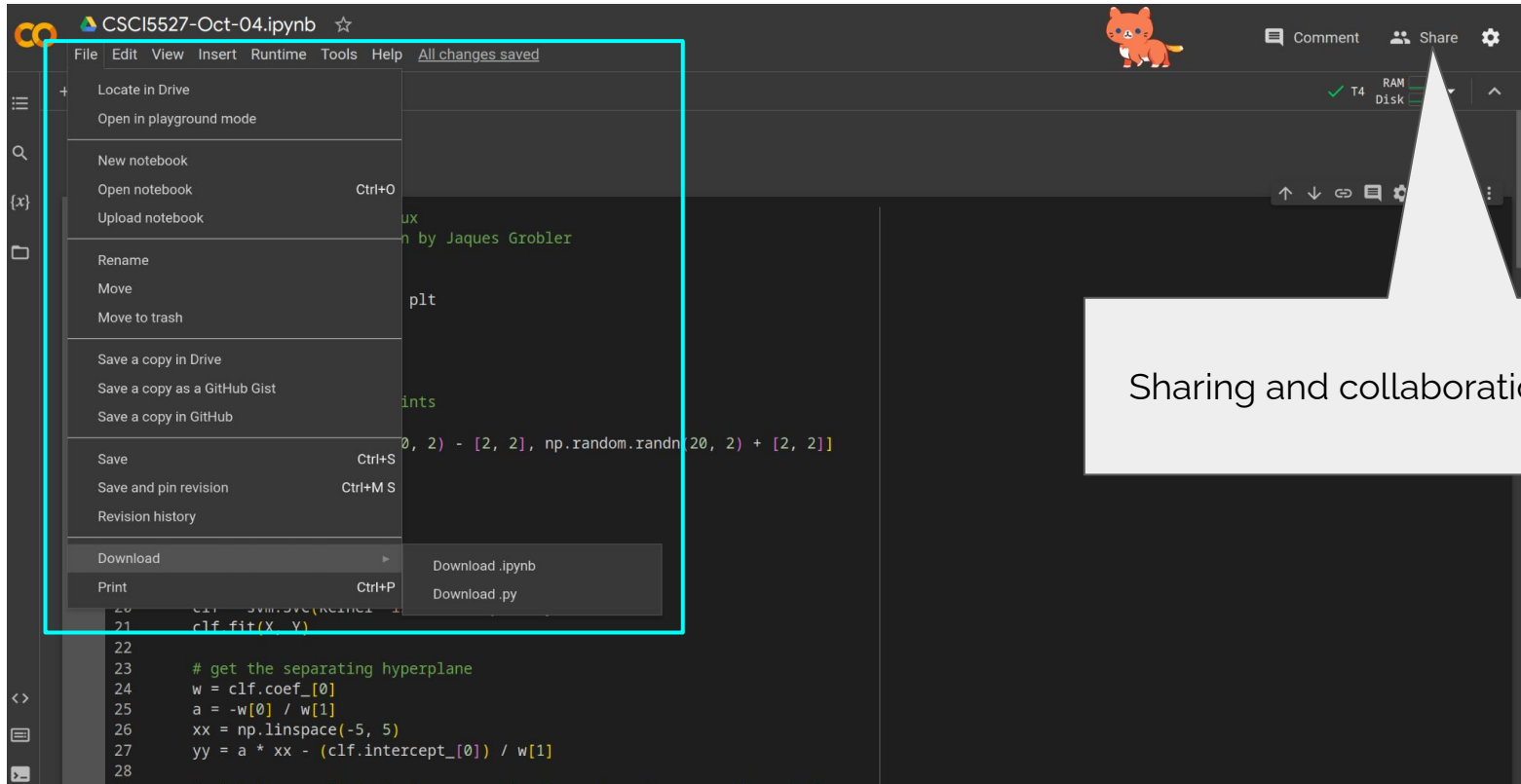
```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

Mounted at /content/drive

At the bottom of the terminal, a file explorer shows a "Review" folder and a "TDL2020Spring-Project" folder, both highlighted with red boxes. A callout box on the left contains the text: "Built in terminal (available to Pro version and above)".

Built in terminal (available to Pro version and above)

# Download the notebook



The image shows a Jupyter Notebook interface for a file named "CSCI5527-Oct-04.ipynb". The "File" menu is open, and the "Download" option is highlighted. The "Download" submenu is also visible, showing "Download .ipynb" and "Download .py". The notebook content includes Python code for a linear classifier and plotting a separating hyperplane.

```
File Edit View Insert Runtime Tools Help All changes saved
+ Locate in Drive
Open in playground mode
New notebook
Open notebook Ctrl+O
Upload notebook
Rename
Move
Move to trash
Save a copy in Drive
Save a copy as a GitHub Gist
Save a copy in GitHub
Save Ctrl+S
Save and pin revision Ctrl+M S
Revision history
Download
Print Ctrl+P
Download .ipynb
Download .py
```

```
20 clf = svm.SVC(kernel='linear')
21 clf.fit(X, Y)
22
23 # get the separating hyperplane
24 w = clf.coef_[0]
25 a = -w[0] / w[1]
26 xx = np.linspace(-5, 5)
27 yy = a * xx - (clf.intercept_[0]) / w[1]
28
```

Sharing and collaboration

# Codey — AI-powered code generation



# More resources about Colab

<https://colab.research.google.com/notebooks/intro.ipynb>

# Outline

- Google Colab
- **Basics of PyTorch**

# PyTorch A deep learning software framework

## Basic components


- Tensors — basic data objects
- Autograd — auto-differentiation
- Optimizer — optimization algorithms
- Both CPU & GPU support

## Specialized components

- Neural network modules (torch.nn)
- Domain-specific modules (torchvision, torchtext, torchaudio, etc)



# Cross-platform, cross-language, cross hardware

 [Get Started](#) [Ecosystem](#) [Mobile](#) [Blog](#) [Tutorials](#) [Docs](#) [Resources](#) [GitHub](#)

## INSTALL PYTORCH

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also [install previous versions of PyTorch](#). Note that LibTorch is only available for C++.




PyTorch Build	<input checked="" type="radio"/> Stable (2.1.0)	<input type="radio"/> Preview (Nightly)		
Your OS	<input checked="" type="radio"/> Linux	<input type="radio"/> Mac	<input type="radio"/> Windows	
Package	<input type="radio"/> Conda	<input checked="" type="radio"/> Pip	<input type="radio"/> LibTorch	<input type="radio"/> Source
Language	<input checked="" type="radio"/> Python		<input type="radio"/> C++ / Java	
Compute Platform	<input checked="" type="radio"/> CUDA 11.8	<input type="radio"/> CUDA 12.1	<input type="radio"/> ROCm 5.6	<input type="radio"/> CPU

Run this Command:

```
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
```

## QUICK START WITH CLOUD PARTNERS

Get up and running with PyTorch quickly through popular cloud platforms and machine learning services.

-  [Amazon Web Services](#)
-  [Google Cloud Platform](#)
-  [Microsoft Azure](#)

<https://pytorch.org/>

**NOTE:** PyTorch LTS has been deprecated. For more information, see [this blog](#).

# Learning PyTorch with examples

[https://pytorch.org/tutorials/beginner/pytorch\\_with\\_examples.html](https://pytorch.org/tutorials/beginner/pytorch_with_examples.html)

A graph is created on the fly

```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))
```



<https://pytorch.org/tutorials/beginner/basics/intro.html>