ECE 469/ECE 568 Machine Learning

Textbook:
Machine Learning: a Probabilistic Perspective by Kevin Patrick Murphy

Southern Illinois University

August 23, 2024

Introduction to Software Tools for Machine Learning

This lecture will cover the following topics.

- Software platforms for machine learning
- Anaconda
- Installing Anaconda
- Anaconda Navigator
- Creating a new environment
- Installing a software platform
- Installing an IDE
- Coding
- Using Google Colab for machine learning

Machine learning frameworks

- Software platforms/frameworks are the backbone for developing and deploying intelligent machine learning (ML) systems. They provide the libraries and tools needed to build, train, validate/test machine learning models.
- There are a variety of ML frameworks, geared at different purposes. Nearly all ML the frameworks are written in Python.
- Scikit-learn, TensorFlow, and PyTorch are the most popular ML frameworks.

Figure: ML frameworks

Anaconda

- Anaconda is a popular open-source Python distribution platform.
- It is a Python/R distribution that contains conda, which is a package and environment manager. This helps user to manage a collection of open-source data science and machine learning packages.
- It also comes with Anaconda Navigator, a GUI desktop application built on conda, with options to create environments based on different frameworks and launch other applications from your managed environment.

Figure: Anaconda distribution platform

Installing Anaconda

You can follow the below steps to install Anaconda in Windows PCs.

1. Use the following link to go the Anaconda web page. https://docs.anaconda.com/free/anaconda/install/windows/
2. Download the Anaconda installer.
3. Go to your Downloads folder and launch the installer. To prevent permission errors, do not launch the installer from the Favorites folder.
4. Follow the steps in the website to complete the installation.
5. After a successful installation you will see the “Thanks for installing Anaconda" dialog box.
6. Open Anaconda Navigator from the Windows Start Menu.

Anaconda Navigator

The basic parts of the Anaconda Navigator is explained below.

1. Navigator pages: Access the main pages of the Navigator application. The Home page is open by default.
    - Home page - Displays all of the available applications that you can manage with Navigator.
    - Environments page - Allows you to create and manage installed environments, packages, and channels.
    - Learning page - Learn more about Navigator, the Anaconda platform, and open data science.
    - Community page - Learn more about support forums and social networking relating to Navigator.
2. Application/package tile: Install or launch Python applications that works well with Navigator.
3. Show all the applications installed in the base (root) folder. Root directory is the place where Anaconda is installed.

Anaconda Navigator

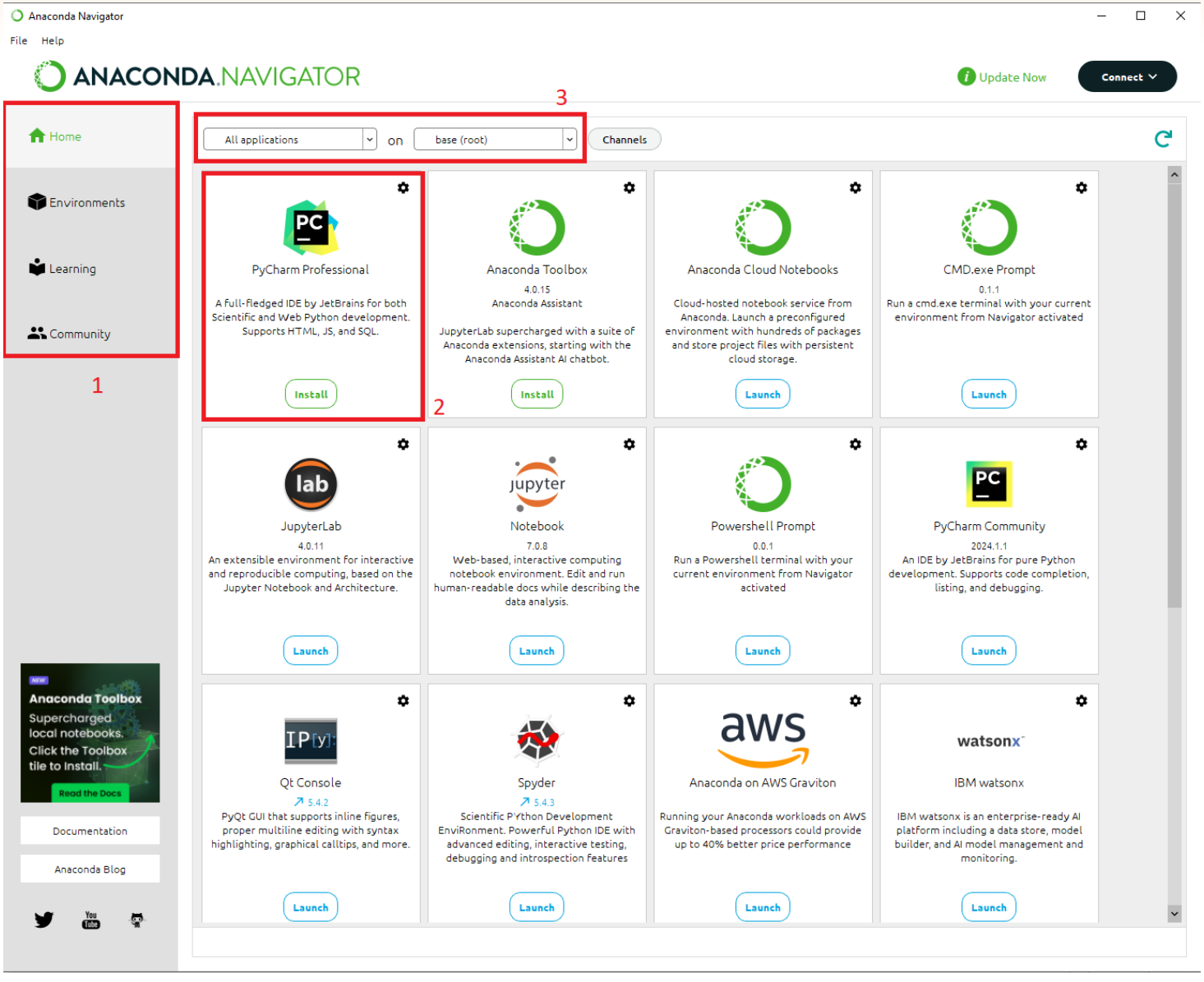

Figure: Basic parts of Anaconda Navigator

Creating an Environment

You can use Anaconda Navigator to create a new conda envrionment. Conda environments are directories within your file system that contain a specific collection of packages and package dependencies. Environments are isolated from one another. Different environments can be created according to your project needs with different collection of packages.

1. Go to Environment page.
2. On the Environment page, click Create.
3. Type a descriptive name for your environment.
4. Choose the Python version you want to install.
5. Click Create. Navigator creates the new environment and activates it.

Creating an Environment

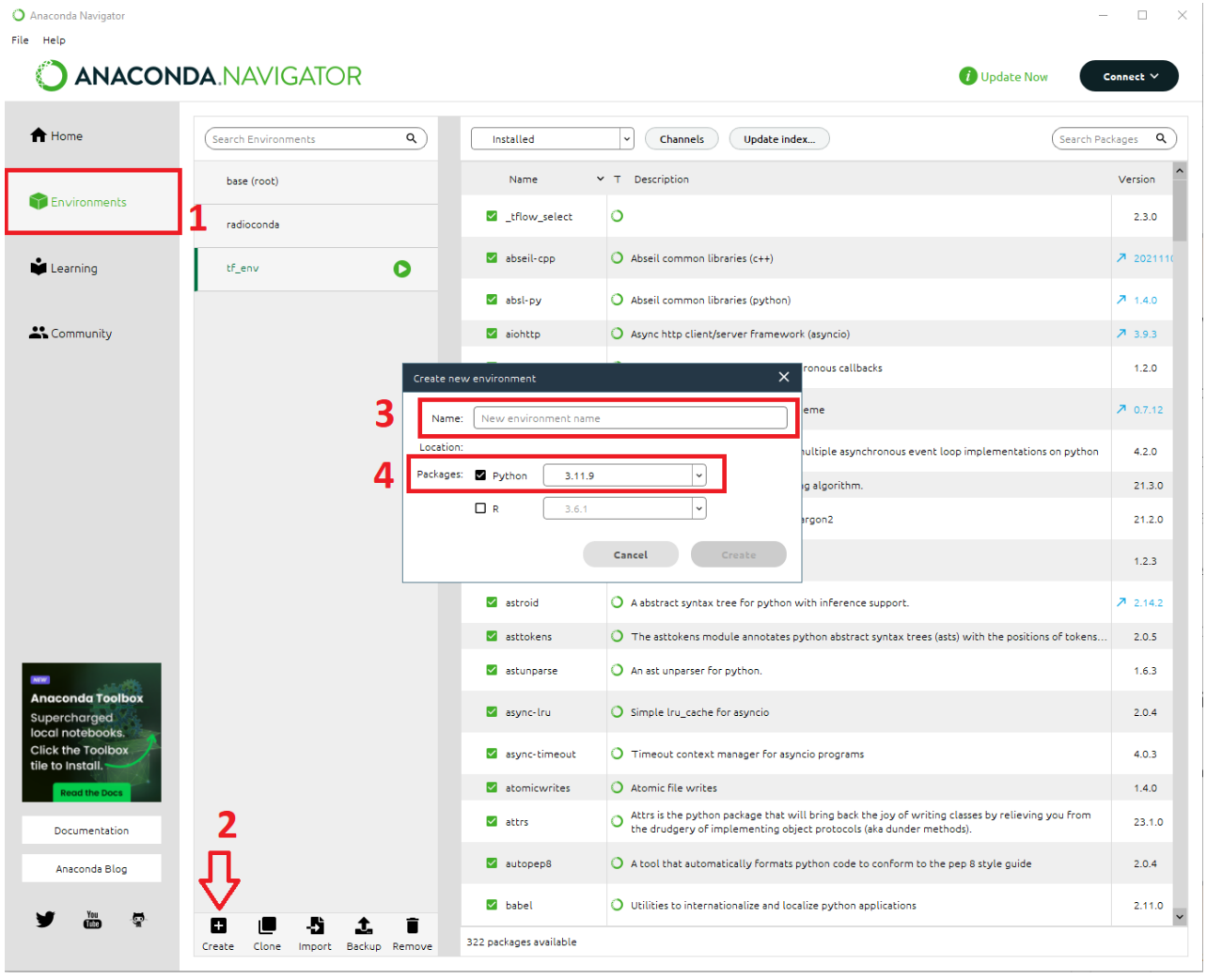

Figure: Creating a new environment

Deep learning frameworks

- In general, deep learning (DL) frameworks conveniently offer building blocks for designing, training, and validating deep neural networks through a high-level programming interface.
- Some of the commonly used DL frameworks are as follows:
  - PyTorch
  - TensorFlow
  - Keras
  - Mxnet
  - PaddlePaddle

Deep learning frameworks

- PyTorch: This is an open source machine learning (ML) framework that supports Python programming language and the Torch library.
  - PyTorch package supports Python and provides two high-level features:
    - Tensor computation (like numpy) with strong GPU acceleration
    - Deep Neural Networks (DNNs) built on a tape-based autograd system
  - We can reuse Python packages, such as numpy, scipy and Cython, to extend PyTorch when needed.
  - PyTorch is originally developed by Meta AI and now part of the Linux Foundation umbrella.
  - You can get started with PyTorch here: `https://pytorch.org/tutorials/`

Deep learning frameworks

- TensorFlow: This is an end-to-end open-source deep learning framework to offer better visualization, which allows developers to debug better and track the training process.
  - TensorFlow is also a Python-friendly open source library for developing neural networks faster and easier via data flow graphs.
  - Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays that flow between them.
  - It has a flexible architecture that allows you to deploy computation to one or more CPUs or GPUs.
  - TensorFlow offers TensorBoard, a suite of visualization tools, for visualizing its results.
  - TensorFlow was developed by the Google Brain team for internal Google use in research and production.
  - TensorFlow package can be used in many programming languages, including Python, JavaScript, C++, and Java.
  - This feature flexibility lends itself to a range of applications in many different sectors.
  - get started here: `https://www.tensorflow.org/tutorials`

Deep learning frameworks

- Keras: This is a high-level, deep learning Application Programming Interface (API) developed by Google for implementing neural networks.
  - Keras is written in Python.
  - This is also open-source neural network library, which is designed to provide fast experimentation with deep neural networks.
  - Keras was adopted and integrated into TensorFlow in mid-2017. Users can access it via the tf.keras module.
  - Keras focuses on being modular, user-friendly, and extensible.
  - However, the Keras library can still operate separately and independently.
  - You can get started with Keras here: `https://keras.io/getting_started/`

Deep learning frameworks

- How to chose your deep learning package: Keras vs PyTorch vs TensorFlow:

	Keras	PyTorch	TensorFlow
API Level	High	Low	High and Low
Architecture	Simple, concise, readable	Complex, less readable	Not easy to use
Datasets	Smaller datasets	Large datasets, high performance	Large datasets, high performance
Debugging	Simple network, so debugging is not often needed	Good debugging capabilities	Difficult to conduct debugging
Does It Have Trained Models?	Yes	Yes	Yes

Deep learning frameworks

- How to chose your deep learning package: Keras vs PyTorch vs TensorFlow:

	Keras	PyTorch	TensorFlow
Popularity	Most popular	Third most popular	Second most popular
Speed	Slow, low performance	Fast, high-performance	Fast, high-performance
Written In	Python	Lua	C++, CUDA, Python

Installing a Platform/Framework (Tensorflow and Keras)

Since you have successfully created a new conda environment, you can proceed to install the machine learning platform/framework that you would like to work with. The following steps will guide you to install Tensorflow and Keras frameworks.

1. Create new envrionment. You can name it as 'tensorflow_env'. Make sure that you install Python 3.6.
2. At your new 'tensorflow_env' environment, select 'Not installed' category. Then, type in 'tensorflow' in the search packages bar. Then, tick 'tensorflow' and 'Apply'. The pop-up window will appear, go ahead and apply. The installation may take several minutes.
3. Follow the same approach for installing Keras.

Note that you can follow this same approach to install any python package you want for your projects.

Installing a Framework

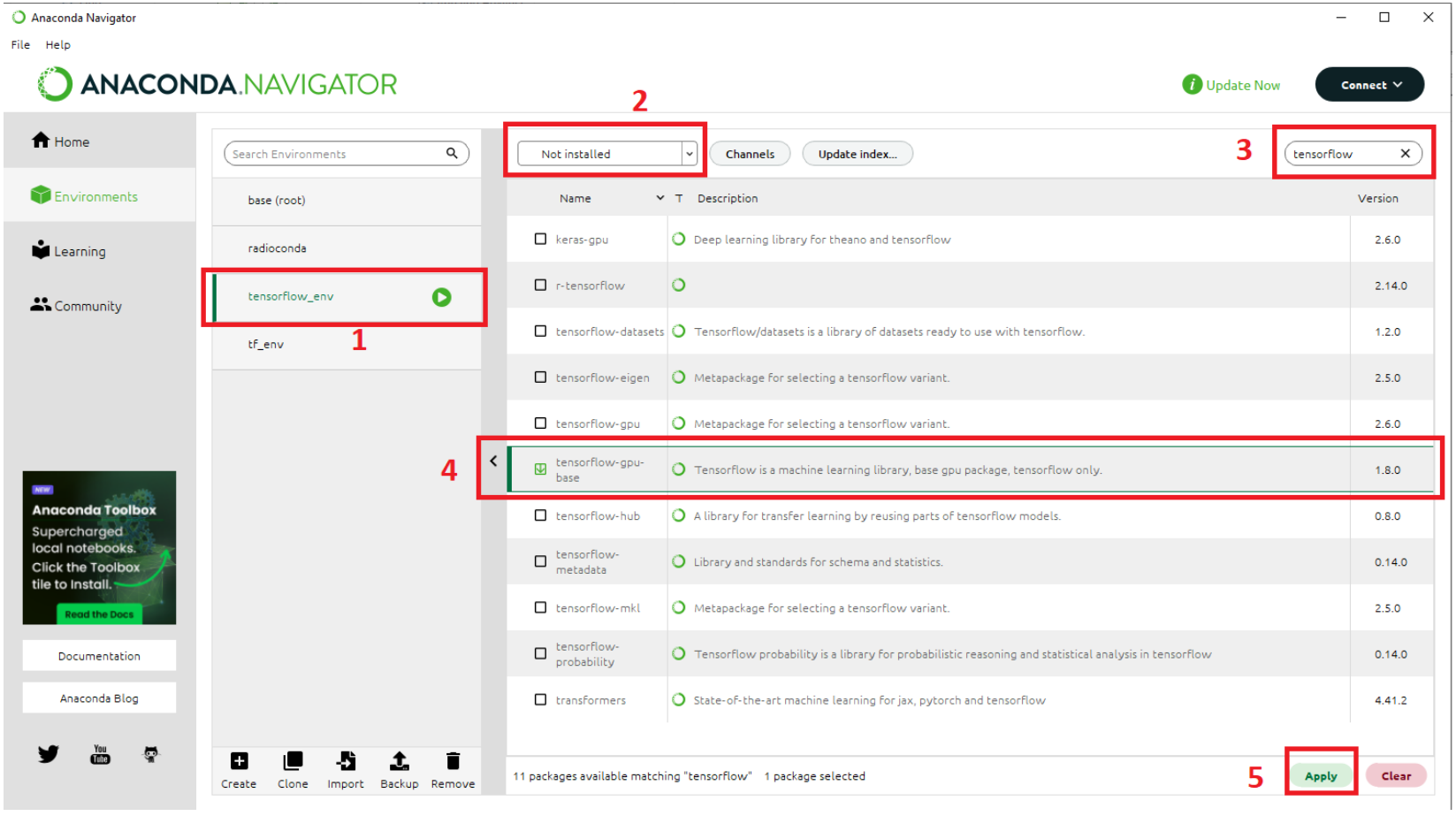

Figure: Installing a framework

Installing an IDE

- A Python Integrated Development Environment (IDE) is a software application that helps programmers write, debug, and test code by combining various aspects of writing a computer program.
- Several options are available in Anaconda Navigator such as PyCharm, Spyder, and Jupyter Notebook.
- Follow the following steps to install an IDE.
  1. Go to Home page in the Navigator. Make sure that you have chosen the correct environment.
  2. You can see the IDEs available for installation.
  3. Select the IDE you like to work with an click 'Install'.
  4. After installation, you will be able to launch the application.
- You can create a new project and start working on your project. Now that you have installed Tensorflow and Keras, you can import the libraries into your code.

Installing an IDE

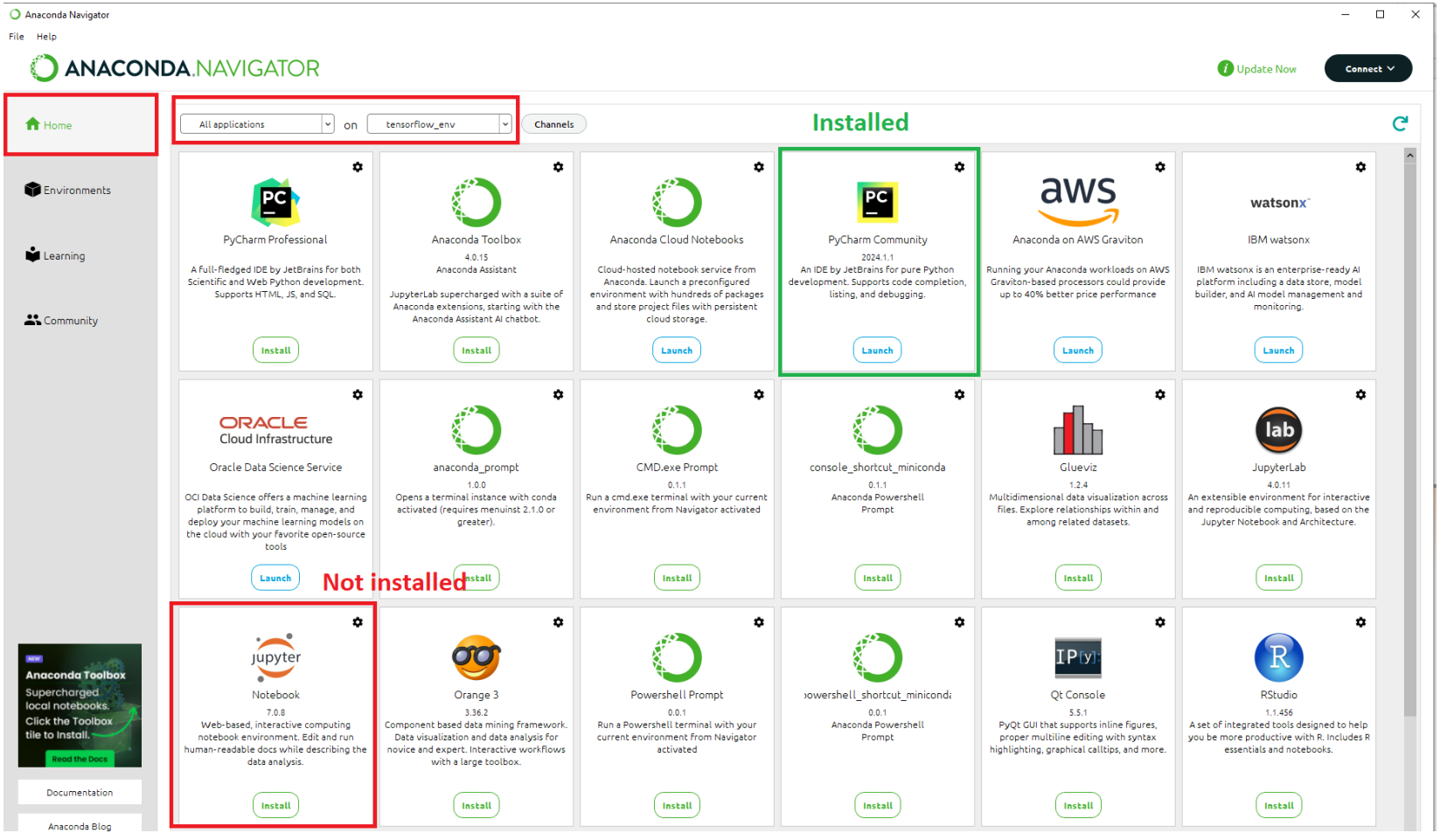

Figure: Installing an IDE

Coding

- Now everything is setup. You can start coding.
- You can create a new project and start working on your project.
- Remember that you can install any python library that is required for your project in a similar approach using Anaconda Navigator.

Using Google Colab

- Google Colab is a free, cloud-based platform that allows you to write and Python codes in a Jupyter notebook environment.
- Using Google Colab
  1. Open your web browser and go to Google Colab. (https://colab.research.google.com/)
  2. Sign in with your google account.
  3. Once you are in the homepage, click on File > New notebook.
  4. This will open a new tab with a fresh notebook where you can start writing and executing Python code.
- Colab comes with preinstalled libraries such as Tensorflow, Keras, and PyTorch.
- You can find wide range of tutorials on topics like machine learning and data visualization. To access them, go to File > Open Notebook and select the Examples tab.
- You can save your work on Google Drive or download it to your computer as a .ipynb or .py file.