

MACHINE LEARNING

Instructor: Ishaya, Jeremiah Ayock

Lecture 02: Introduction to Machine Learning

January 20, 2025

Academic City University College, Agbogba Haatso, Ghana.

Course Outline

Introduction to Machine Learning.

- Introduction to Course, Requirements, Tools, and Resources
- Students Setup Python IDE (Anaconda)
- Installation of Numpy, Pandas, Seabon, Matplotlib, Scikit-Learn, and TensorFlow

Notations and Definitions

- Python
- Numpy
- Pandas
- Scipy
- Matplotlib/Seaborn
- Data Exploration

Supervised Learning 1

- Regression problems
 - Simple Linear Regression
 - Multiple Regression
 - Ridge Regression **
 - Lasso Regression **
- Evaluating Regression Algorithms

Supervised Learning II

- Classification Problems
 - Logistic regression
 - Decision Tree
- Evaluating Classification Algorithms

Unsupervised Learning I

- Clustering
 - KMeans
 - Hierarchical
 - DBSCAN

Unsupervised Learning 2

- Dimensionality Reduction
- Gaussian Mixtures

Ensemble Learning

- Voting Classifiers
- Bagging and Pasting
- Random Forests
- Boosting

Neural Networks I

- Artificial Neural Networks
- Multilayer Perceptrons
- Feedforward Neural Networks
- Gradient Descent
- Back Propagation

Neural Networks II

- Convolutional Neural Networks
- Recurrent Neural Networks

- Recommendation System

- Introduction to Large-Language Models

- Seminar Presentation and Workshop

Week 1

**Introduction to Course/Machine
Learning**

LECTURE OUTLINE

- Introduction to Course
- Introduction to Course, Requirements, Tools, and Resources
- Students setup a Python IDE (anaconda), create a virtual Environment
- Installation of TensorFlow, Keras
- Github Repository

Main Focus

- Supervised Learning
- Unsupervised Learning
- Neural Networks
- Recommender Systems
- Large Language Models

RECOMMENDED BOOKS/RESOURCES

1. Dangeti, P. (2017). Statistics for machine learning. Livery Place, Birmingham: Packt Publishing;
2. Karim, M.R. and Kaysar, M. (2016). Large-scale machine learning with spark. Livery Place, Birmingham: Packt Publishing;
3. Shai.S. S amp; Shai B.D (2014). , Understanding machine learning from theory to algorithms, Cambridge University Press
4. Christopher M. Bishop(2006). Pattern Recognition and Machine Learning, Springer
5. Mitchell Tom. Machine Learning. McGraw Hill, 1997.
6. Richard O. Duda, Peter E. Hart, David G. Stork. Pattern classification (2nd edition). Wiley,

Getting Started with Anaconda

INSTALLING ANACONDA

Anaconda is a distribution of packages built for data science. It comes with *conda*, a package and environment manager. You'll be using *conda* to create environments for isolating your projects that use different versions of Python and/or different packages. You'll also use it to install, uninstall, and update packages in your environments. Using Anaconda has made my life working with data much more pleasant.

```
conda create -n tea_facts python=3
```

```
source activate tea_facts
```

```
conda list
```

```
conda install numpy pandas matplotlib
```

INSTALLING ANACONDA CON'T

Anaconda comes with the Jupyter notebook package. If you are using Miniconda, you can install Jupyter notebook as:

conda install jupyter notebook

WHAT IS ANACONDA DISTRIBUTION?

Anaconda is a program to manage (*ie. install, upgrade, or uninstall*) Python packages and environments. It is simple to install packages with Anaconda and create virtual environments to work on multiple projects conveniently.

Even if you already have Python installed, it will be beneficial to use *Anaconda/Miniconda* because:

- Anaconda comes with a bunch of data science packages; you'll be all set to start working with data.
- Using **conda** to manage your packages and environments will reduce future issues dealing with the various libraries you'll be using.

A **package** is a bunch of modules, where each module consists of a set of classes and function definitions. After installing a particular package, you can **import** and use the functions defined in that package.

If we install Anaconda, then a basic few packages are installed by default. However, you can install any more packages, if needed.

ANACONDA DISTRIBUTION

Anaconda is a fairly large download (500 MB) because it comes with Python's most common data science packages. Anaconda is a software distribution that includes the following:

- **Anaconda Navigator**- It is a graphical user interface that helps open up any installed applications, such as **Jupyter notebook or VS code editor**. We will learn more about the notebook in the next lesson. See a snapshot of Anaconda Navigator below:

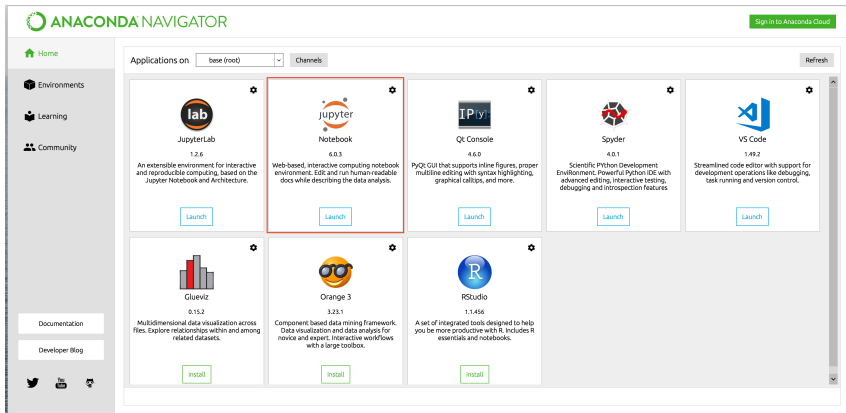


Figure 1: Anaconda Navigator GUI, same in both macOS or Linux and Windows.

ANACONDA DISTRIBUTION CON'T

- **conda**- A command-line utility for package and environment management. Mac/Linux users can use the Terminal, and Windows users can use the "Anaconda Prompt" to execute conda commands. Windows users must run the Anaconda Prompt as an Administrator. Your first command should be **conda --version**
- Python - The latest version of Python gets installed as an individual package.
- Over 160 scientific packages and their dependencies are also installed.

If you don't need all the packages or need to conserve bandwidth or storage space, there is an option for you -**Miniconda**.

Either *Anaconda* or *Miniconda* is adequate for this course. Miniconda is a smaller distribution as compared to Anaconda, which includes only conda and Python. Miniconda can do everything Anaconda is capable of, but **doesn't have the preinstalled packages.**

OVERVIEW - MANAGING PACKAGES USING EITHER PIP OR CONDA

The **conda** and **pip** both are the Python package managers. Package managers are used to installing libraries and other software on your computer. **pip** is the default package manager for Python libraries, whereas conda focuses only on the packages that are available from the Anaconda distribution.

Both **pip** and **conda** gets installed when you install either Anaconda or Miniconda. On the next page, we will see details to install Anaconda/Miniconda. However, pip also comes preinstalled with the current Python 3 version.

WHICH ONE SHOULD I PREFER -PIP OR CONDA?

There are two points you can consider before making a choice:

- The available packages available from the Anaconda distribution in **conda** focus on data science, whereas **pip** is for general use. Conda installs precompiled packages. For example, the Anaconda distribution comes with Numpy, Scipy, and Scikit-learn compiled with the [MKL](#) library, speeding up various math operations. **But, sometimes, you may need packages other than the ones listed on the Anaconda distribution.**
- Pip can install both Python and non-Python packages. Pip can install any package listed on the Python Package Index [PyPI](#).

You can (and will) still use pip alongside conda to

ENVIRONMENTS


A Python environment comprises a particular version of each of the following:


- Python interpreter,
- Python-packages, and
- The utility scripts, such as pip. It is possible to have two or more environments residing on the same computer virtually. If you are using Anaconda, you are in the base(root) environment by-default.

ANACONDA NAVIGATOR

 Home

 Environments

 Learning

 Community

Documentation

Developer Blog

Search Environments



base (root)



deep-learning

Installed

Name

☒ _ipyw_jlab_nb_ex...

☒ absl-py

☒ alabaster

☒ alembic

☒ anaconda

☒ anaconda-client

☒ anaconda-project

27

WHY DO YOU NEED A VIRTUAL ENVIRONMENT?

Each virtual environment remains isolated from other virtual environments and the default “system” environment.

Environments allow you to separate and isolate the packages you are using for different projects. Often you'll be working with code that depends on different versions of some library.

You can also export the list of packages in an environment to a file, and then include that file with your code. This allows other people to easily load all the dependencies for your code. Pip has similar functionality with **pip freeze > requirements.txt**.

CONTINUOUS ASSESSMENTS

- Attendance, 50%
- Participation 50%

Total = 5%

CONTINUOUS ASSESSMENTS CON'T

- Homework / Assignments 15%
- Quizzes 20%
- Research Project – Team 45%
- Mid-Semester Evaluations 20%

Total = 35%

CONTINUOUS ASSESSMENTS CON'T

- Exam 60%

Total = 60%

Final Evaluations/Exam = 100%

END OF PRESENTATION

THANK YOU