Lab 1: Environment Setup & Data Handling

Part 2 of this assignment is due to Gradescope by the start of lab next week. You may discuss concepts and ideas with others in the class, but you must submit your own code.

Introduction

The purpose of this lab is to get you started running Jupyter notebooks with the Python libraries we will use throughout the semester. You will also get some practice loading and analyzing data – a prerequisite for training machine learning models.

Provided Files

- `Lab1.pdf`: This file
- `Lab1.ipynb`: Jupyter notebook with Part 2 instructions
- `cta-ridership.csv`: Data for exploration

1 Environment Setup

The first part of this lab involves preparing your programming environment. Follow the instructions in each section below. All commands should be run in Command Prompt (Windows) or Terminal (Mac).

1.1 Python 3

We will be using Python 3 for all class and lab programming. You should first check to see whether you already have Python 3 installed by running the following command: `python3 --version`.

If you do not have Python 3 installed, you should download and install Python **3.8** here: https://www.python.org/downloads/.

1.2 Python Virtual Environments

A **virtual environment** is an isolated instance of Python that will allow you to install packages without affecting (or being affected by) other packages on your computer (e.g. from other courses). Using virtual environments is a best practice for Python development and will significantly reduce the chance of encountering environment-related bugs during the semester.

1.2.1 Virtual Environment Creation

First, install the `virtualenv` tool with the following command: `pip3 install virtualenv`. Once this is complete, you should be able to run `pip3 list` and see `virtualenv` listed.

Next, `cd` into the directory where you would like to keep the virtual environment for this class. This is up to you, but your Desktop, Documents, or a folder specific to this class are all good choices.

Run `python3 -m virtualenv venv-COSC410` to create a new virtual environment named "venv-COSC410". If you run `ls` (Mac) or `dir` (Windows), you should see the newly created `venv-COSC410` directory containing the environment.

1.2.2 Virtual Environment Use

In order to use the virtual environment, you need to tell your Terminal or Command Prompt to use the environment's Python instance. Do this on Mac with `source venv-COSC410/bin/activate` or on Windows with `.\venv-COSC410\Scripts\activate`. If this works, you will notice that the prompt in your shell is now appended with `(venv-COSC410)` to indicate that you are "inside" of the virtual environment.

While inside the environment, all Python commands and scripts will use the environment's Python instance and have access to the Python packages you have installed in the environment. They will NOT have access to any Python packages you installed in your computer's default Python instance (this is a good thing...it enforces environment modularity!).

If you want to exit the virtual environment, you can either 1) run the command `deactivate` or 2) close Terminal or Command Prompt (when you re-open it, you will no longer be in the environment).

IMPORTANT: Remember to enter the virtual environment for all programming you do for this course.

1.3 JupyterLab

JupyterLab is a browser-based interface for interactive Python notebooks. Jupyter notebooks are cell-based, meaning that your code (or text markup!) are divided into cells that can be run in any order. The notebooks are backed by a Python kernel that maintains state between cell executions. Jupyter notebooks are a very common programming environment for machine learning and data science. The cell-based execution and in-place plotting makes data exploration particularly convenient. We will be using Jupyter notebooks for some labs and most in-class exercises.

You can download and install JupyterLab with the command `pip install jupyterlab` from **inside** the virtual environment (additional instructions here if necessary: https://jupyter.org/install).

Once you have JupyterLab installed, try opening it with the `jupyter-lab` command and playing around with the interface.

1.4 Additional Python Libraries

Install the following Python libraries (from **inside** the virtual environment):

- **NumPy**. `pip install numpy`. Documentation: https://numpy.org/
- **SciPy**. `pip install scipy`. Documentation: https://docs.scipy.org/doc/scipy-1.6.0/reference/
- **Pandas**. `pip install pandas`. Documentation: https://pandas.pydata.org/
- **Excel Read**. `pip install xlrd`. Documentation: https://pypi.org/project/xlrd/
- **Matplotlib**. `pip install matplotlib`. Documentation: https://matplotlib.org/
- **Seaborn**. `pip install seaborn`. Documentation: https://seaborn.pydata.org/
- **Scikit-Learn**. `pip install scikit-learn`. Documentation: https://scikit-learn.org/stable/index.html
- **TensorFlow**. `pip install tensorflow`. Documentation: https://www.tensorflow.org/
- **Keras**. `pip install keras`. Documentation: https://keras.io/

2 Data Handling

The second part of this lab involves practice importing, manipulating, and plotting data using the provided `Lab1.ipynb` file. You should open and work through this file in JupyterLab.

3 Deliverables

Upload the following files to Gradescope:

- Completed `Lab1.ipynb`

4 Extra Credit Opportunity

If you find a bug anywhere in this lab, please inform Prof. Apthorpe. The first student(s) to find any particular bug will be given a small amount of extra credit. This will help make the course better for students in future years.