

# Instructions for Lab 1 (Using Python to Analyze Decision Trees)

## Learning Objectives:

- Execute Python code using Spyder and Jupyter Notebook. (Code)
- Describe the syntax and function of a command by reading Python help. (Code)
- Modify a given piece of Python code. (Code)
- Write Python code to analyze a decision tree. (Code/Analyze)

## Preliminaries

If you have not yet done, so follow [these instructions](#) to install Miniconda, Jupyter notebook, Spyder, and Graphviz. Make sure you follow the instruction I provided to install Graphviz (above) rather than using pip to install (the solution commonly found on the Internet). This is because conda is incompatible with pip.

After installation, create a new folder on your Desktop for this class, and give it a name of your choice. Then navigate to the Folder for this lab. For the two .py files, left click on each of the file, then click "Raw," then save it into the folder you just created (Ctrl-S on Windows and Apple-S on Mac). Make sure the extension is still .py. For the Jupyter notebook file with extension .ipynb, click into the file and click download notebook on the top right. Eventually you should have three files in the same folder: decisiontrees.py, graphic\_corporation.py and worksheet.ipynb.

## Running Python Code

### Opening Jupyter Notebook

Open a Anaconda prompt (in Windows) or a terminal (on Mac or Linux). Navigate to your home directory. To do this on windows, type

```
cd %userprofile%
```

On a Mac or Linux, type

```
cd ~
```

Then open Jupyter notebook by typing.

```
jupyter notebook
```

This opens a web browser with a File explorer. Go to your Desktop and enter the folder you created above, and double click the worksheet.ipynb file.

## Running Cells

A Jupyter notebook is composed of cells. Each particular cell has several types, which you can edit by clicking the cell and choosing the type in the Dropdown menu at the top. The relevant cell types we will use are "Markdown" and "Code." Markdown cells are for creating easily formatted text. Code cells are for executing Python code. To execute a cell, click on the cell and use the menu at the top: Cell -> Run Cells. You can also use the shortcut Ctrl-Enter to run a cell. The shortcut Alt-Enter runs the current cell and creates a new one below it.

## Running Code in Spyder

Alternatively, you can run a Python script by opening it in Spyder and type F5, or click on the run button at the top.

Spyder is good for larger projects involving editing several scripts. Jupyter notebook is good for sharing your results with others. In this lab, we will focus on using Jupyter notebook instead of Spyder.

## Editing a Jupyter Notebook

### Command and Edit Modes

There are two modes in Jupyter notebook:

- Command mode: Enter this mode by typing ESC. This is for adding cells.
- Edit mode: Enter this mode by clicking a cell. This is for changing a cell.

You can create a new cell also by using the top menu: Insert -> ... Alternatively, you can create a new cell below a particular cell, by clicking on the cell you want, and type "ESC" to enter command mode, then "b". All of the keyboard shortcuts can be found by typing "ESC" to enter command mode, then "h".

### Editing Markdown Cells

Markdown cells are used to easily create formatted text. To edit a Markdown cell, double click it. To execute it (render it nicely), run it using instructions above. This webpage gives you examples for using Markdown: <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

You can learn more about the notebook tool here: <http://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Notebook%20Basics.html>

## Using Python Help

In Spyder, you can load the help file for a certain function by highlighting it and typing **Ctrl-I**. This displays the help on the top right screen.

In Jupyter notebook, you can access the help for a command by starting a code cell and typing the command followed by a question mark, then executing.

For help on Jupyter notebook or writing Markdown, you can click on the Help menu in Jupyter notebook. Or use one of the above links in section 1.2.

To look for more documentation and examples using a command, I would recommend using a Google search. Python has a large community so it's easy to find support for what you need.

## Decision Tree Analysis

In this lab, you will use Python to analyze decision trees. An explanation of the files you downloaded:

- `decisiontrees.py`: This contains a simple module I wrote to implement decision trees in Python. It is the same as that in the notes to Session 3 (decision trees), except that I added comments (which are indicated by triple single quotes for multi-line comment and hash signs for single line comments).

- `graphic_corporation.py`: This encodes the decision tree from Session 3 on the Graphic Corporation case using the above module. Note that we build the tree from the bottom nodes up: the list of descendants is needed to create a node, which means that the descendants must have been created first.
- `worksheet.ipynb`: This is the Jupyter note book that **you will be completing today and submitting on Blackboard to receive the credit for this lab**. (If you didn't make it to class, you can submit it within 24 hours of class.)

Use Spyder to open the .py scripts and apply what you learned from last class (Reading Python Code), as well as Python help, to understand it. Work together with your neighbor and ask your table leader for help if needed.

After you familiarized yourself with the files, complete the `worksheet.ipynb` following the instructions therein. **Take your time and play around with the code and the interface. Focus on learning rather than finishing the worksheet as you will get full credit even with partial completion.**

## FAQ

**Question:** What to do if I'm using Mac and I'm getting an error message when installing matplotlib about an incompatible package called libpng?

**Answer:** Open a new terminal and type the following command:

```
conda install -c conda-forge libpng
```

**Question:** I have a space in my user name and I'm unable to run the `graphic_corporation.py` script.

**Answer:** Unfortunately, there is a bug with Graphviz. In order to use the package, you must reinstall Miniconda in a directory without any spaces in the names, rather than in your home directory.

**Question:** How come I cannot open the Jupyter notebook `worksheet.ipynb` file?

**Answer:** It's likely that you did not download it correctly. You must click download notebook in nbviewer, or in Github click "Raw" then save. Make sure you save it in the correct extension of .ipynb.