Lesson Title	**Intro to MNIST**
Sequence	**4 of 6**
Duration	● **45 minute class to set up environment, run code, and reflect**
Materials	● **Python IDE**
Objectives	**Students will examine the existing MNIST script and make connections between the output probabilities and the applications of handwriting recognition they encounter in other arenas of life.**
Standards	**Indiana** ● CSII-6.1 Describe the function of a computing artifact ● CSII-6.2 Identify the purposes of a computing artifact **ITEEA** ● 3 Students will develop an understanding of the relationship among technologies and the connections between technology and other fields of study.

Lesson Notes:

This is designed to introduce students to the MNIST digit classification network. For students to run the MNIST python code, they will first need to establish a TensorFlow environment. After they have downloaded the TensorFlow package, they will run the provided MNIST script. Students will be asked to examine the output data and make observations regarding the resulting probabilities. They will also be asked to reflect on how this type of digit classification maps to the technology they have explored with apps like Photomath.

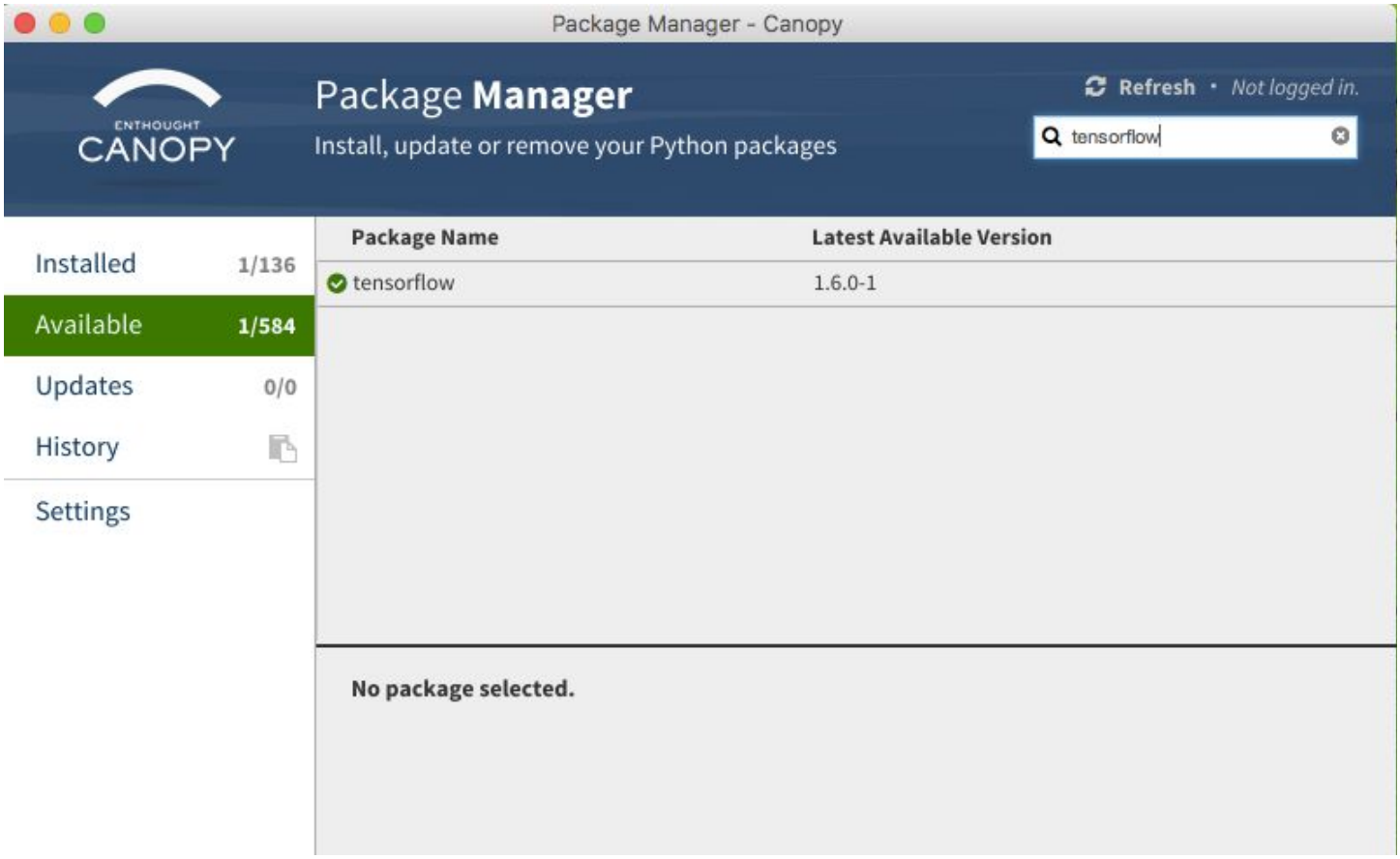
Assessment: This task will be assessed for thorough responses, clear communication, and timeliness.

Lesson 4 of 6: Introduction to MNIST

Part I: Establish the Environment

Before you can run any of the MNIST script, you need to establish a TensorFlow environment.

1. Open Canopy.
2. Navigate to the Tools Menu and select Package Manager.
3. In the Package Manager window, choose the available tab on the left sidebar. Use the search tool to locate the TensorFlow package. Choose install.

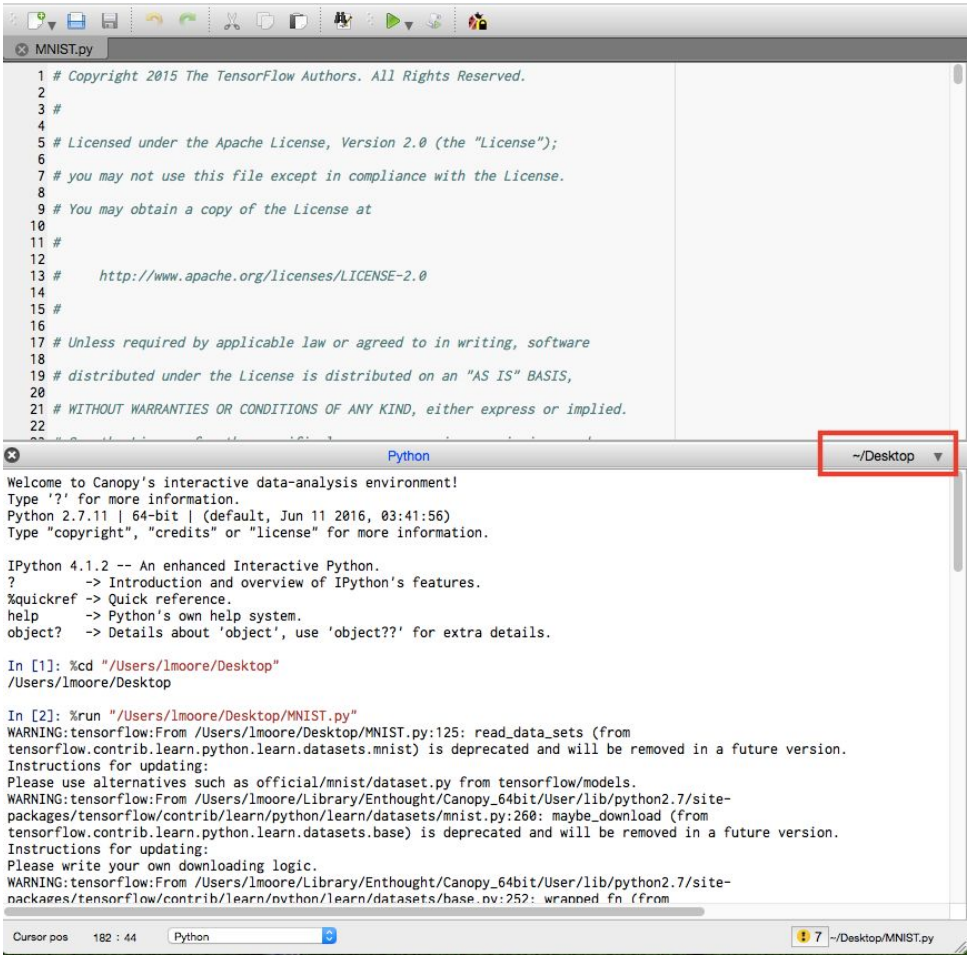


4. When you open a new editor window in canopy, you will be able to import the TensorFlow module like you would with random, matplotlib, or others you have encountered in this course.

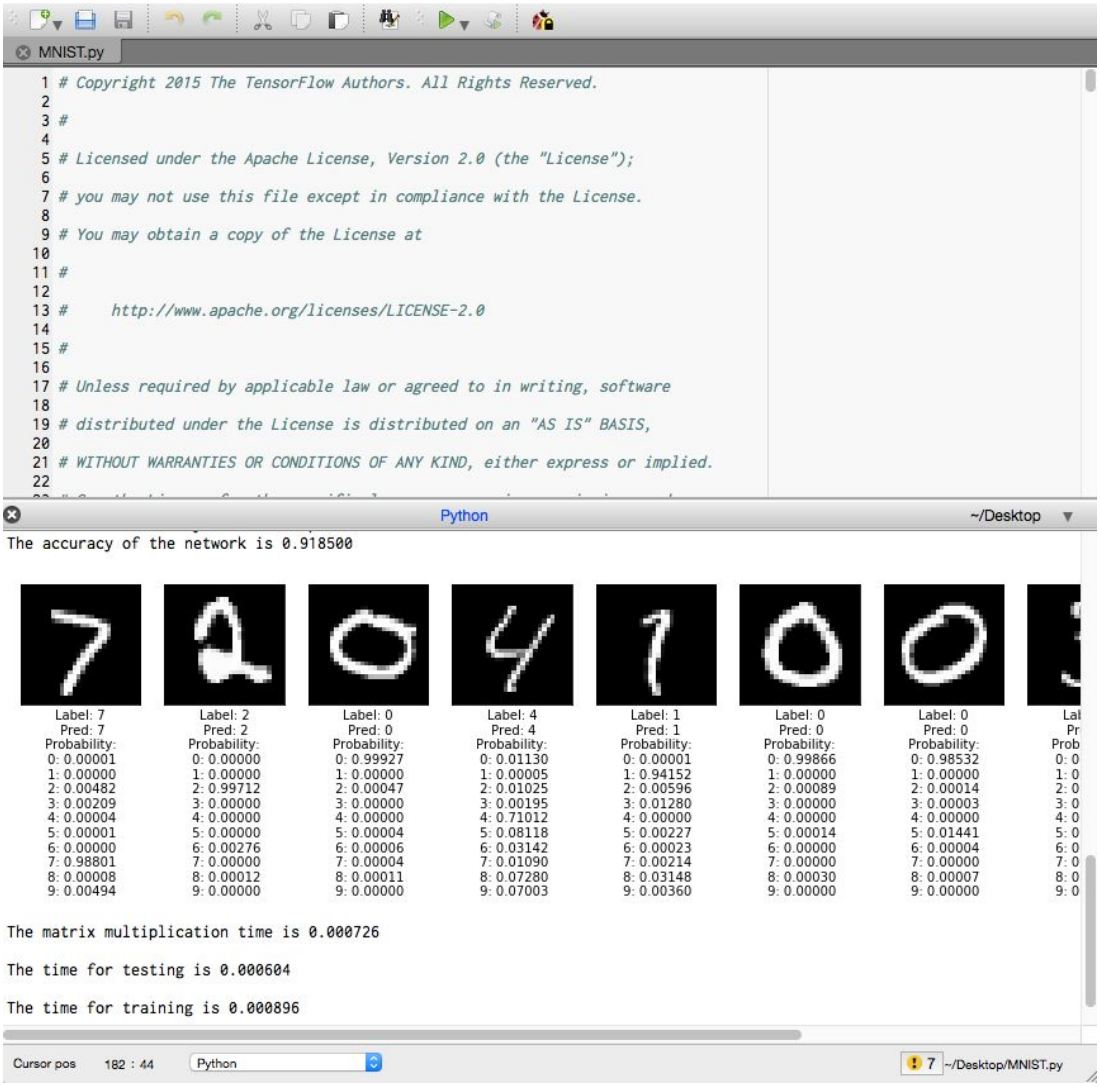
Part II: Download the MNIST package

---

1. Download this zip file and unzip the contents. I would recommend moving the file to your desktop so it will be easy to identify errors when you establish your working directory.
2. Change your working directory to the location where you placed the python file.



3. Try to run the python script. If it runs successfully, you should receive an output similar to the one below.



Part III: Reflection

Looking at your results from the MNIST script, consider the following questions.

1. What information is included in the output? What does this communicate?
2. Were all of the digits correctly classified? If not, which ones were incorrectly classified?
3. Look more closely at the probabilities that are displayed. Pick one of the digits that was correctly classified, but had a lower probability. Do you see how the computer could misinterpret that image?
4. Look at your own handwriting. How does it map to the digits displayed in the output?
5. How might this technology of image classification connect to applications like Photomath?