*These two guides, Installation and User, are relevant to anyone looking to run the Digit Recognition Dashboard for Genetique. We assume that you have no experience with Anaconda or Jupyter.*

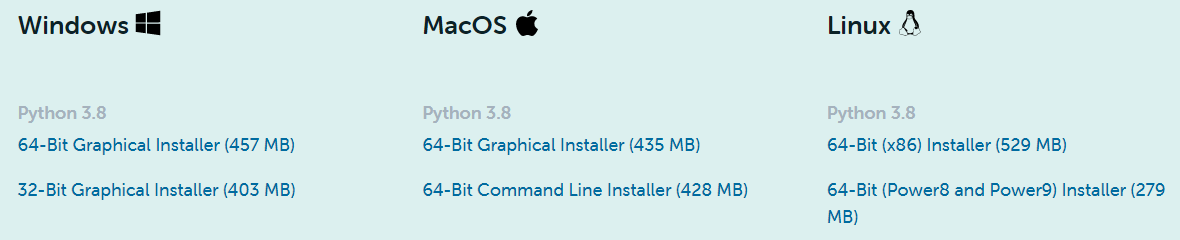
*We also assume that you are on a desktop or laptop running Windows 10 64-bit, although this document should still work as a general guide for other Windows versions, as well as OS X and Linux. Several pictures are included to make the process easier.*

**Installation Guide**

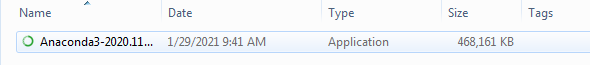
To run this application, you must install Anaconda and some libraries, through which you will run Jupyter Notebook.

Go to <https://www.anaconda.com/products/individual> and click the Download button to install the Individual version of Anaconda.

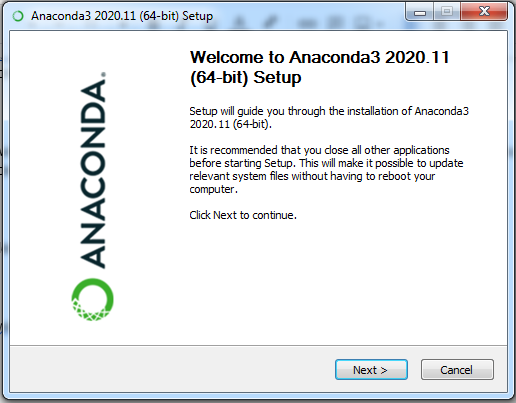
Clicking the Download button will take you to a list of operating systems to choose from. For our example, we’ll be using the Windows 64-Bit Graphical Installer.



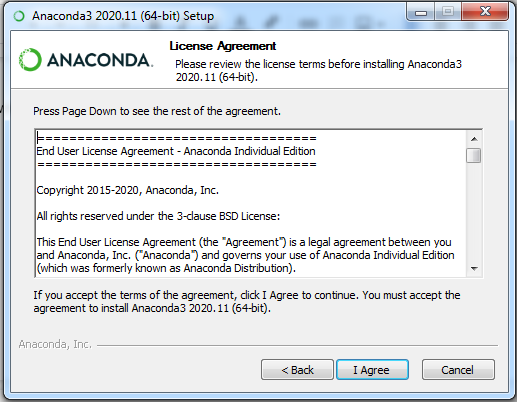
Once the installer is downloaded, run it.



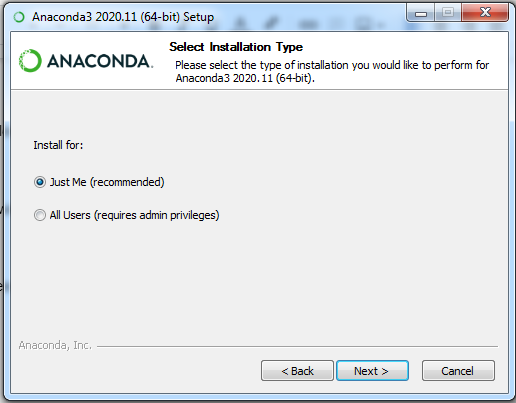
An Installation window will pop up. Click “Next”.



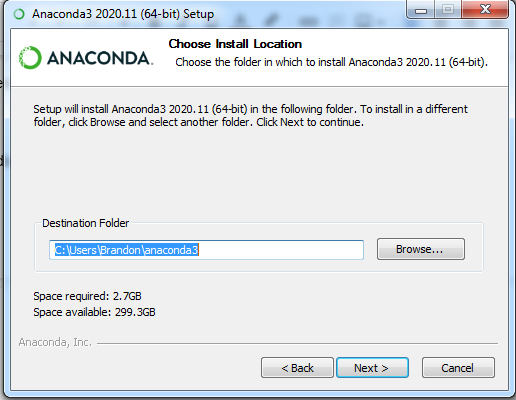
Next you will see a License Agreement. Click “I Agree”.



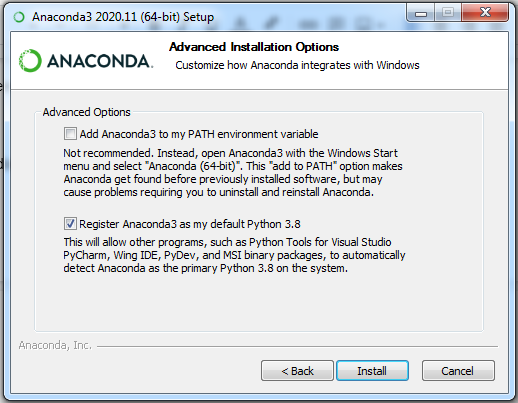
In the next window, select “Just Me” and then click “Next”



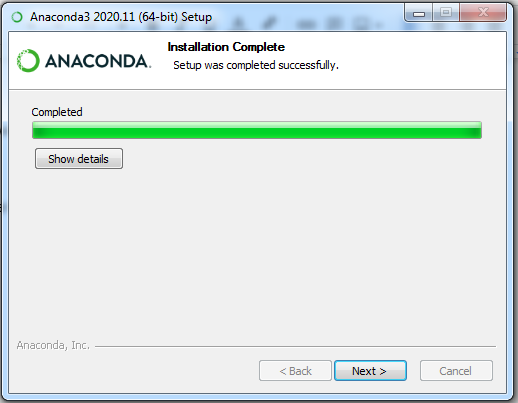
Choose an installation path in the next window. The default is usually fine. Click “Next”.



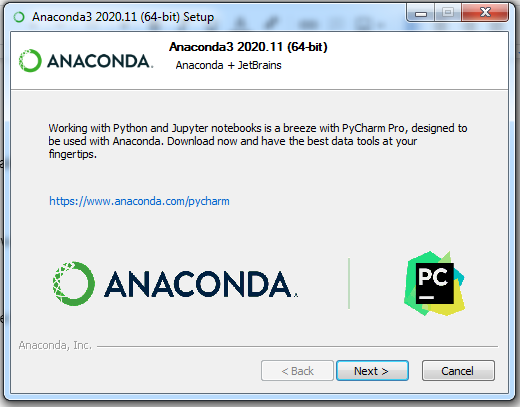
The next window has advanced options, and the options we chose are below. Click “Install”.



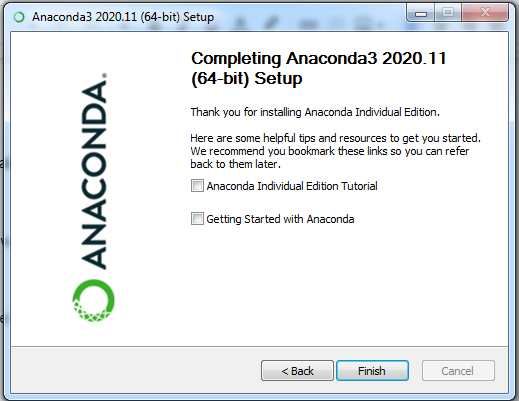
Allow some time for Anaconda to install, then click “Next” once it is done.



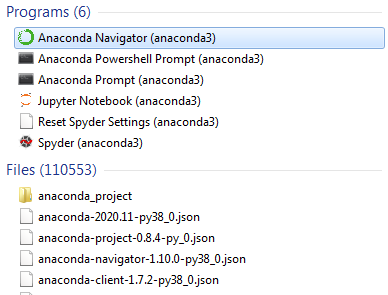
Click “Next” unless you would like to install PyCharm for Anaconda first.



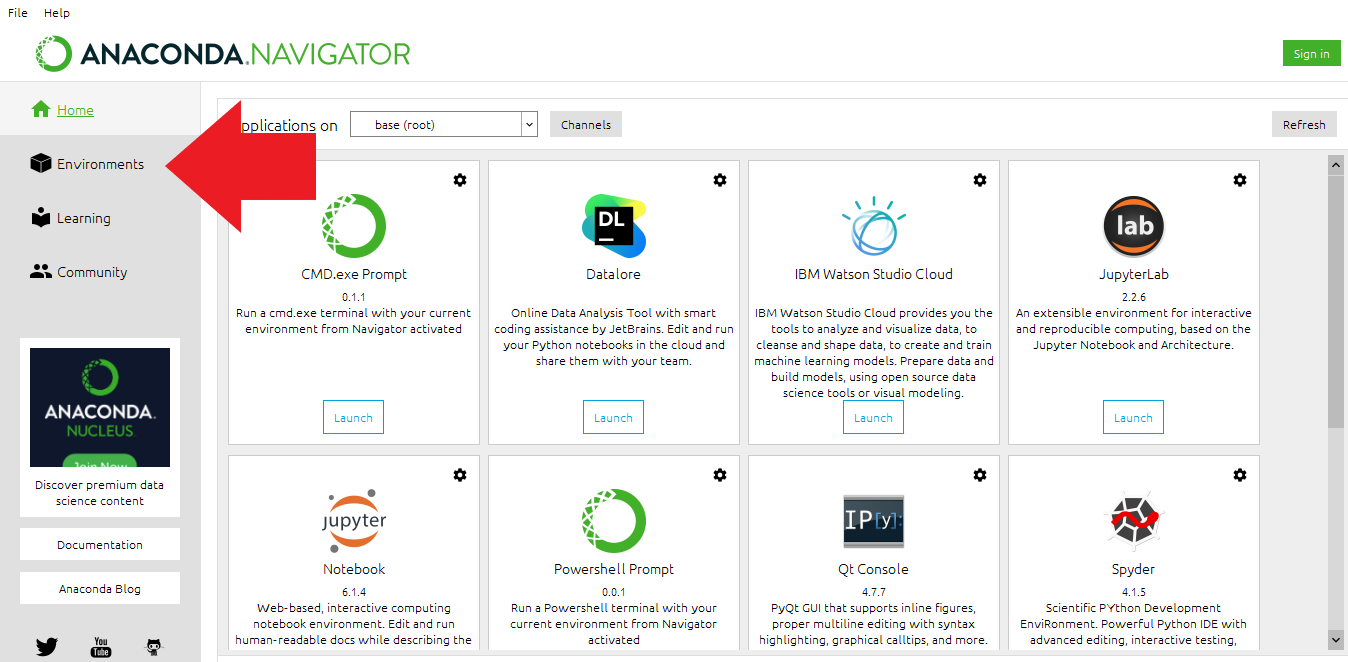
Optionally check the two boxes on the next screen, then click “Finish” to complete the installation.



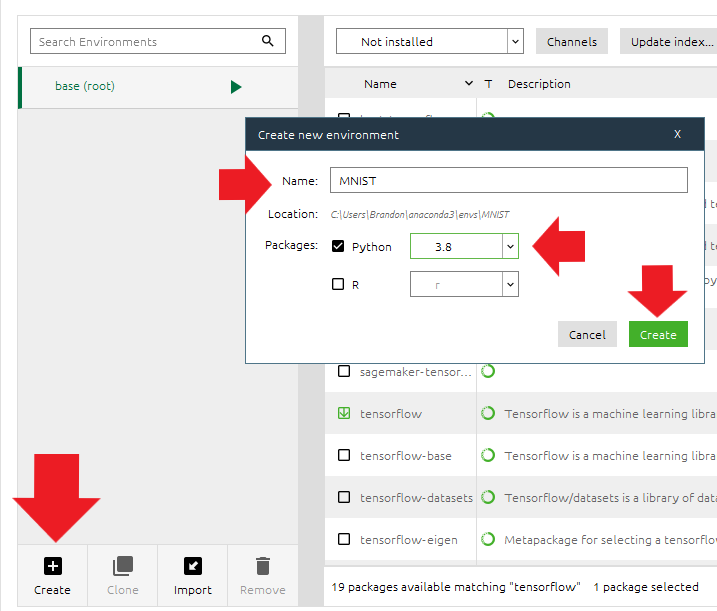
Next, we need to search for and run Anaconda Navigator. It may take up to a few minutes to initialize for the first time.



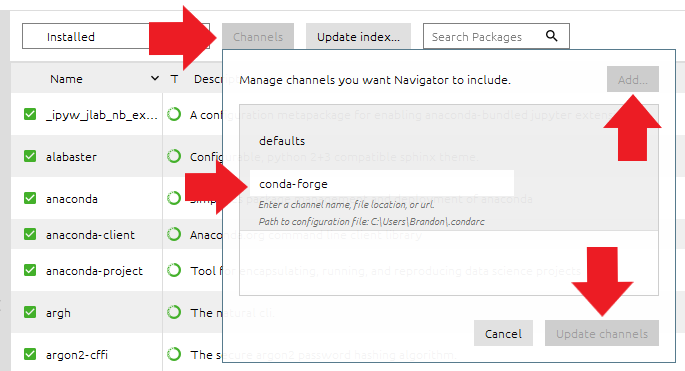
Once you have it open, click on “Environments”. We’re going to need to create a new environment, then install some libraries for this application to run properly.



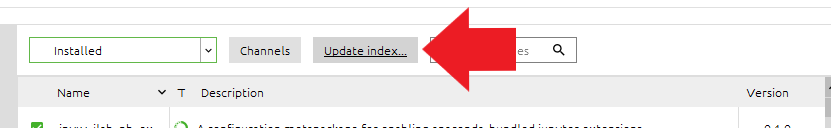
To create a new environment, first we will click the “Create” button. This will bring up a window as shown below. You can name it anything you want, but we are going to name it MNIST for this example. It is going to use Python 3.8. Click “Create”. This will create a new environment where we can install packages.



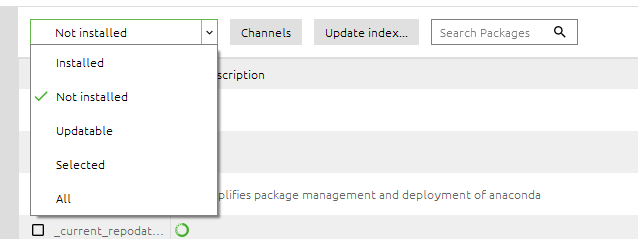
With the Environments tab open *and MNIST selected*, click on “Channels”, then click “Add...” and type in “conda-forge” (without the quotes) and hit enter. Click Update Channels and wait for Anaconda to finish loading the new channel.



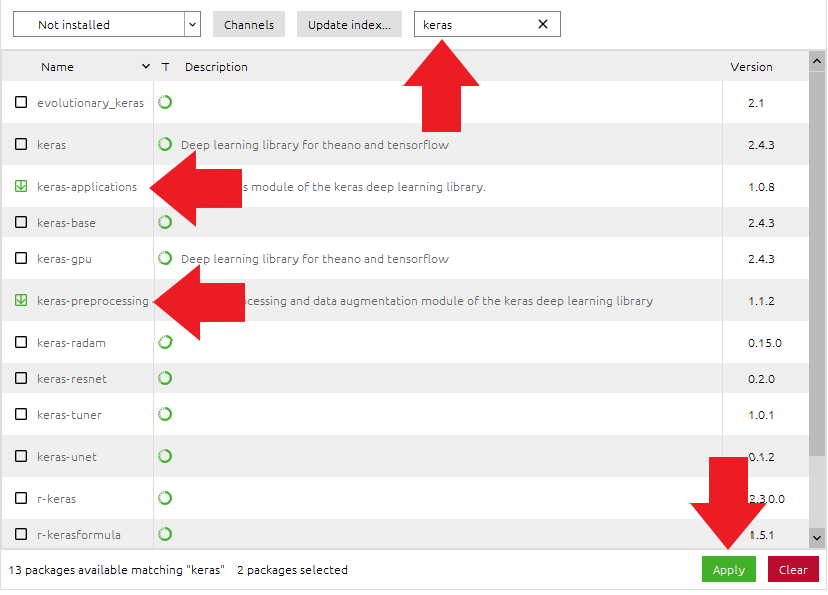
Click “Update index…” to refresh the available libraries.



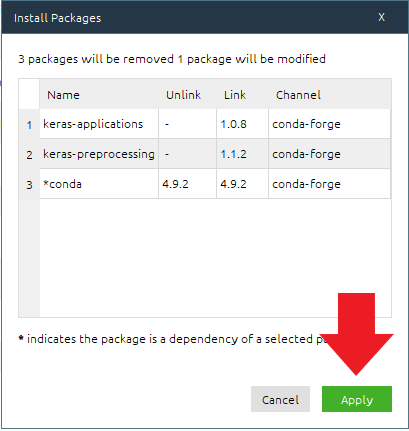
Now, click the drop-down where it says “Installed”. We’re going to change this to “Not Installed”.



Now that the search is showing uninstalled libraries, we can search for the libraries that we need. First, let’s do a search for “Keras”. Once you see Keras-applications and Keras-preprocessing, you can check the boxes next to them and then hit “Apply”. It can sometimes take a while to load, depending on the library.



Once it finishes loading, you will have a pop-up telling you what packages are going to be modified. Click “Apply” to accept the modifications. This installs the library so that you may use it with any of your applications in Anaconda, assuming you are in the MNIST environment.



Now that you know how to install a library, install the rest of the needed libraries below:

Ipympl



Tensorflow



Voila



Voila-material



Pandas



Scikit-learn



Seaborn

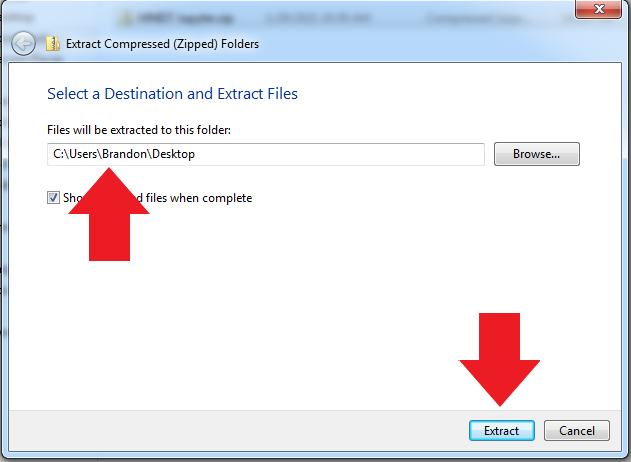


After you have done that, now you will need the application’s code and data to use with Jupyter Notebook.

Download the source code for the model and dashboard, as well as the appropriate data, using this link:

<https://drive.google.com/file/d/1QpnXANgahu0f8D-MXS48fa85Sof19QWr/view?usp=sharing>

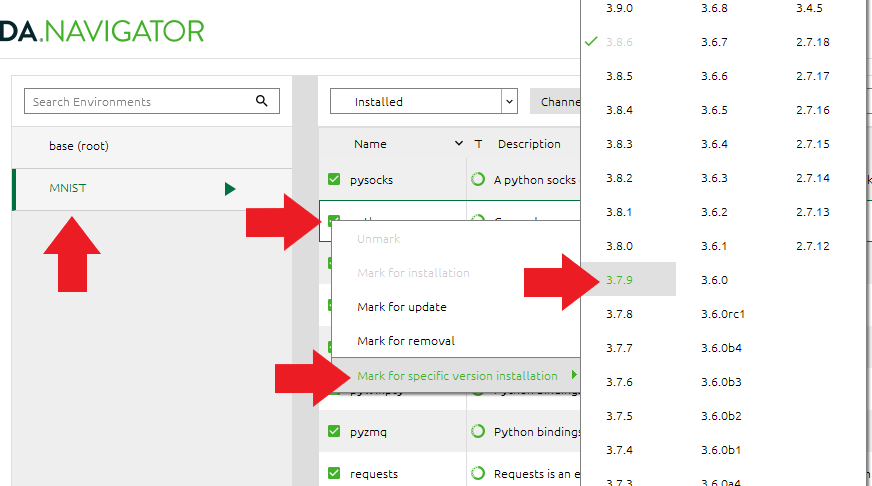
Once it is downloaded, right-click the zip file called “MNIST Jupyter.zip” and click “Extract All”. Extract it to a convenient place. For our example, we simply put it on the Desktop. Once you have chosen a location to extract, click “Extract”.



**You have now finished the installation! You will see a guide that explains how to use the application below.**

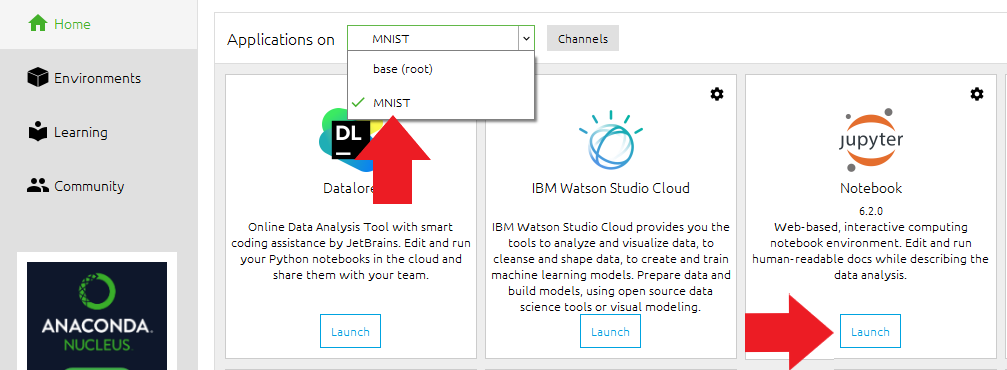
*Note: Occasionally there will be an issue with the currently installed version of Python, causing you to get a dead kernel when you run an application. If this happens, do the following:*

*Go back to the Environments tab in Anaconda Navigator. Select the MNIST environment. Find “python” in the package list and click the check mark next to it. In the window that pops up, go to “Mark for specific version installation” and click on version 3.7.9. After that, hit “Apply” and install it like you would a library. This process can take a while to complete.*

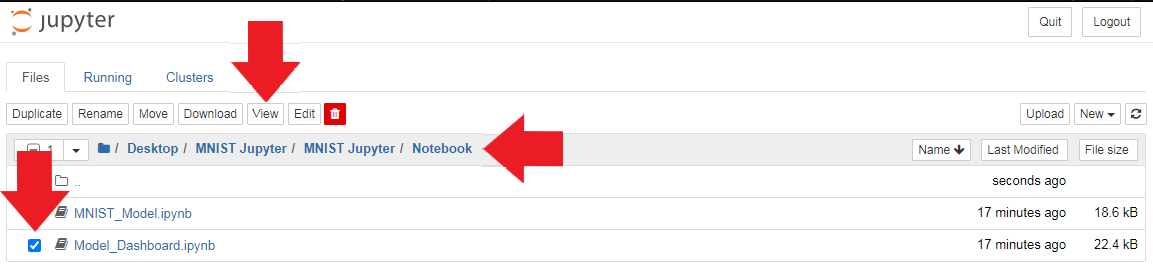
**

**User Guide**

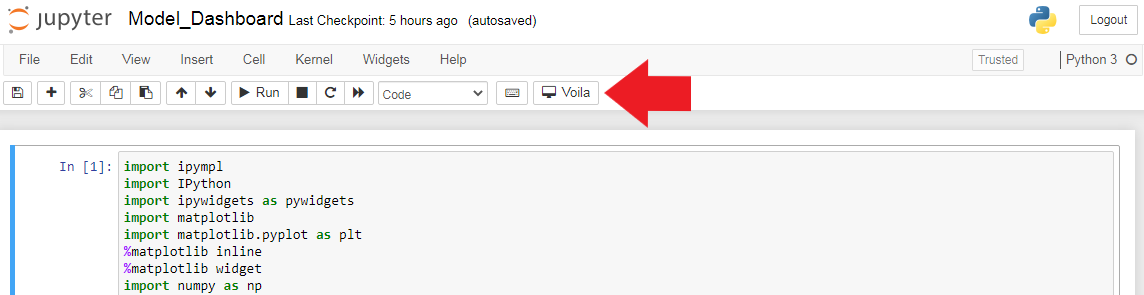
Now that the code files and data are ready to be used, go to the main menu in Anaconda Navigator, switch to the MNIST environment, and open Jupyter Notebook by clicking “Launch”.



Once you launch it, you should have a browser open with a list of files and folders. Within this list, find the folder that you just extracted called “MNIST Jupyter” and open it. Then open the Notebook folder. Next to the file “Model\_Dashboard.ipynb” there is a checkbox. Click the checkbox, then click the “View” button to open it.

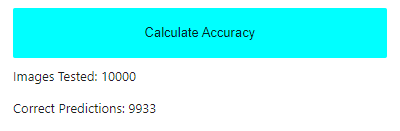


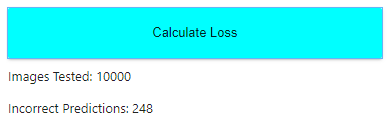
Once it is open, you should see a button named “Voila”. This button will launch the dashboarding software so that you can use the dashboard without executing code in the notebook.



As the model (MNIST\_Model.ipynb) is changed or refined, all of the different parts of the dashboard will change to represent the new model.

At the top of the dashboard, you will see two cyan buttons. As their labels suggest, these will calculate the accuracy and loss results of the model. These can take a minute to load, so it is normal if nothing immediately pops up after clicking them.

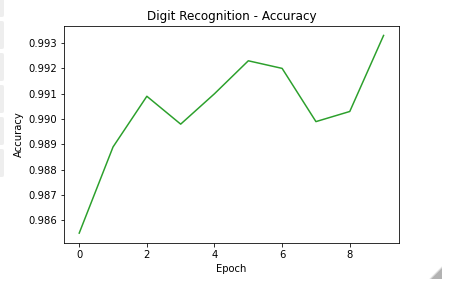




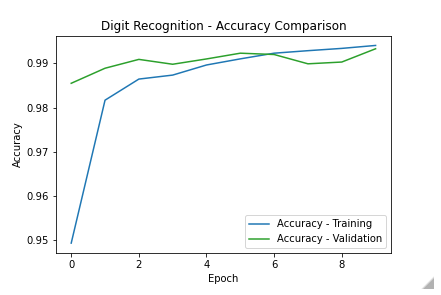
Next, we have six different graphs.

Each of these graphs are an analytical visualization of the following:

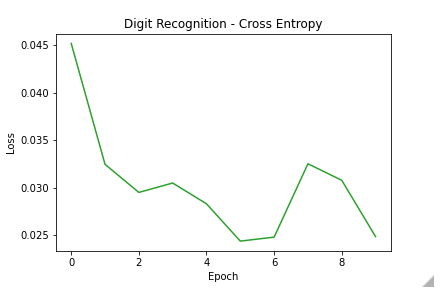
The accuracy of the model when predicting images in training over ten epochs.



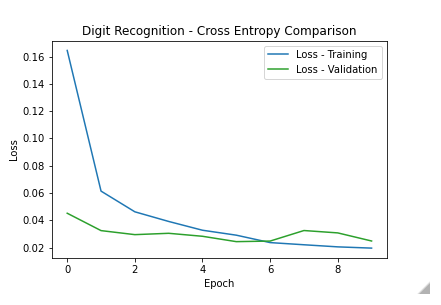
The accuracy of the model when predicting images in training versus the accuracy of the model when predicting images in validation (testing) over ten epochs.



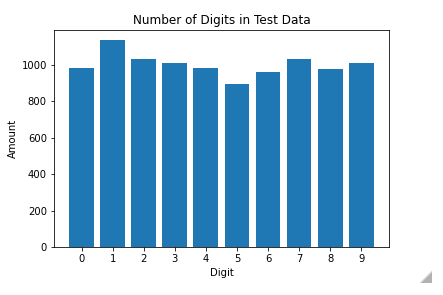
The inaccuracy (loss) of the model when predicting images in training over ten epochs.



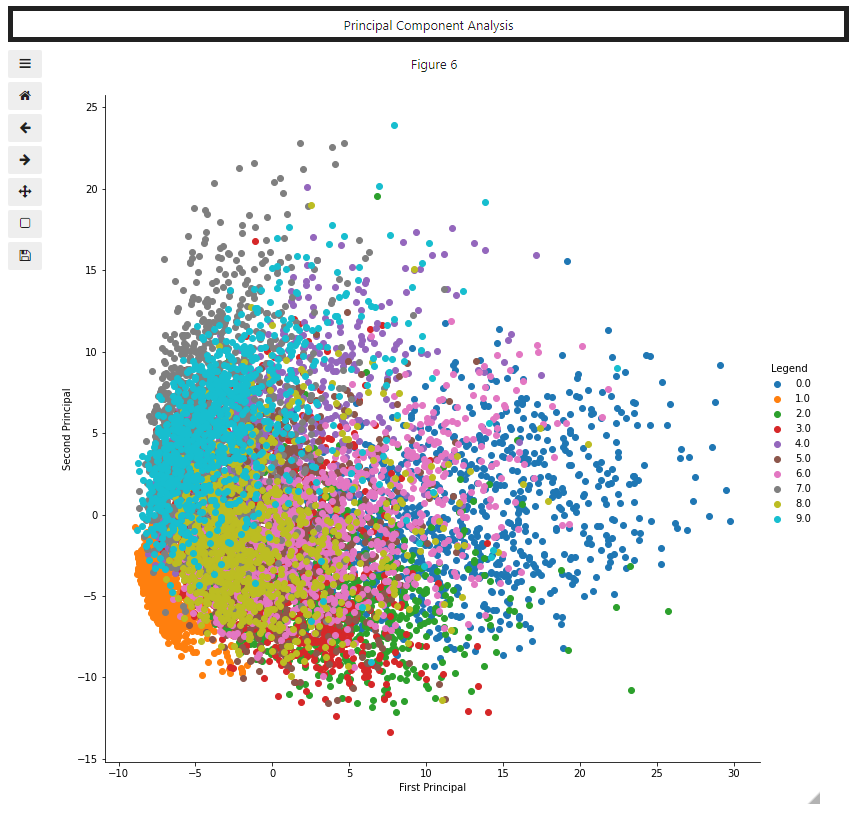
The inaccuracy of the model when predicting images in training versus the inaccuracy of the model when predicting images in validation over ten epochs.



The amount of each digit in the validation data set.



The comparison among principal components. The first principal component deals with the differences among images of each of the different numbers in the dataset, 0-9. The second principal component deals with the difference between the images of the same number, due to all of the different ways a number can be written. The closer together the points are, the more similar the images are.



Lastly, there is a widget that allows you to grab an image from the MNIST dataset and predict what number is in the image. Clicking on the image will give you a prediction, shown directly above the image. Using the “Clear” button will clear the prediction. Using the “Get New Image” button will grab a new image to be predicted.

