**Files requirement:**

**1. Required submission files:**

Summit at least two files to the team directory on the server. One file is able to print out your prediction number. The program can be written in any programming language.

e.g. prog.py

The other file is **runner.sh**, a script that we use to run your file. Do not change the filename.

In this case, the content of runner.sh to run prog.py is

# runner.sh

python3 prog.py

Remember that we only run the runner.sh file using following command line on the server. The output should be your team’s prediction number.

$ sh runner.sh

2873.33

There’s no limitation on the numbers of submission. Please make sure runner.sh runs the file you want.

**2. Input data file:**

The data file that would be used as the inputs is updated daily. It’s provided in the home directory of the server. You can access the file after log in server via ssh.

The directory and name of the datafile is ”~/data.txt”. Make sure to write code that can read the data file. Connect to the server to check if your program can read the data file and work.

**Tutorial for files submission**

**Step 1: Upload your files to the server**

Open the "Terminal" application. Use prog.py and runner.sh file for example. Type the following command in your terminal, replacing **USERNAME** with **your username**:

$ scp prog.py runner.sh USERNAME@neural.pstat.ucsb.edu:~/

When you type it in your password, nothing will show on the screen. However what you type is still being sent. Press ENTER to login once you are finished with your password. When you see the following, submissions are complete.

prog.py 100% 74 3.3KB/s 00:00

runner.sh 100% 27 1.0KB/s 00:00

**Step 2: Connecting to server via SSH to check if the files can run properly on the server.**

***On Mac OS X or Linux:***

Open the "Terminal" application. Type the following command in your terminal, replacing **USERNAME** with **your username**:

$ ssh USERNAME@neural.pstat.ucsb.edu

SSH will first ask you a question which looks like this:

The authenticity of host ‘neural.pstat.ucsb.edu (128.111.43.14)' can't be established.

ECDSA key fingerprint is 90:ab:6a:31:0b:81:62:25:9b:11:50:05:18:d3:1a:b5.

Are you sure you want to continue connecting (yes/no)?

Type **yes** and then ENTER to continue. When you type it in your password, nothing will show on the screen. However what you type is still being sent. Press ENTER to login once you are finished with your password,

**You should now be remotely connected to the server!** You can now do anything you could normally do in a terminal window.

Type “exit” to disconnect the server.

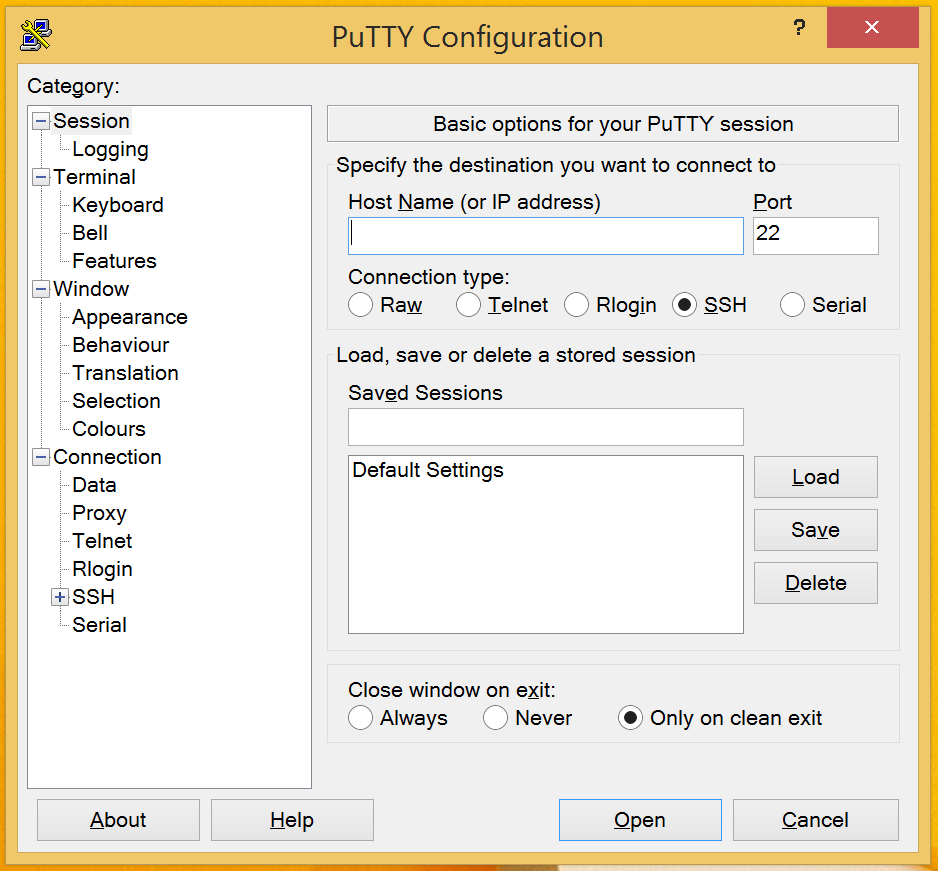
***On Windows:***

To connect remotely on Windows machines, we recommend using a program called PuTTY. This program is a well-known and widely-used SSH client for the Windows OS.

First, download the program from [**http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html**](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html). You only need the executable file **putty.exe**, but feel free to download any other programs that you want. The page includes portable versions and a version with an installer. *Always make sure to download PuTTY from this site*, so that you can make sure it is the correct program.

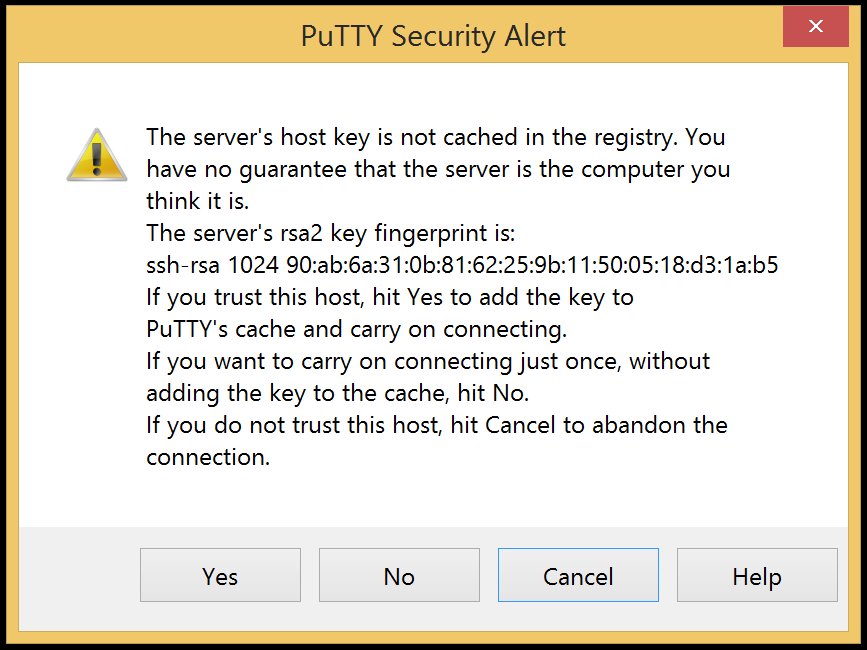
Once downloaded, run PuTTY like you would open other programs. If you just download the **putty.exe** file, you can open it from your downloads folder directly. You can also move it to any other location on your machine and open it from there. If you used the installer, open PuTTY from the Start Menu.

When PuTTY opens, you should see a window that looks like this:



Type **neural.pstat.ucsb.edu** into the box labeled “Host Name (or IP address)”. Leave the “Port” setting at 22 and leave the “SSH” button checked.

Then click on the “Open” button to connect. PuTTY will then show a prompt which looks like this:



Click “Yes” to accept and have PuTTY remember the key.

Once a connection is made, server will ask for both your username and then your password. Type in your server username and password. The password will not be shown on the screen, but the characters you type are being used.

Once you have logged in successfully, you should be connected remotely to the server. You can now do anything you could normally do in a terminal window.

Type “exit” to disconnect the server.

**Step 3: Check if your files can run properly on the server**

After entering remotely into the server, you can now use command lines to check if the current directory contains at least two files.

Type the following commands to make sure your runner.sh file is executable.

$ chmod 775 runner.sh

Now you can play around and test to see if your runner.sh gives the result.

$ sh runner.sh