How to make your life easier with CIS519

Let's face it: Our crappy laptops probably won't be powerful enough to train decent estimators.

Go to a Linux machine in Moore 100A. Dozens of Core i7, 16GB RAM workstations are lying around, wasting electricit$. Make use of them.

Use Spyder

First of all, no matter whether you are going to use your own laptop or a computer in CETS, let me talk you into using the IDE called Spyder.

1. The kernel-client design of IPython allows one to work on intermediate results instruction-by-instruction.

   In English, this means, if you spent 10 minutes parsing training data from JSON to a numpy array and then started to start developing your classifier, you don't have to reloa$ your training data every time you want to test out your classifier. You can even directly inspect shapes and values of variables.

2. Jupyter Notebook uses IPython kernels, too. Why not use that instead?

   In two words: file format. You can create beautiful reports out-of-the-box with Jupyter notebooks, but in this course we are required to submit pure Python scripts instead. W$ th Spyder, you don't have to convert your code from a `.ipynb` file to a `.py`.

Install Anaconda

Install Anaconda with Python 2.7

Considering that, throughout this course, we will be programming with Python 2, let's install a separate environment for that.

1. Download Anaconda2 here.

```
bash Anaconda2-5.0.1-Linux-x86_64.sh
```

3. Follow the instructions on terminal.
4. Finally, since Anaconda installer didn't know I was using zsh, I had to append this line manually to my `~/.zshrc`:

   ```
   export PATH="/home1/m/myli/anaconda2/bin:$PATH"
   ```

5. In a new terminal, you should be now able to use `spyder`.

At present, the icons on the menu bar is going to be distorted (replaced by some random Korean symbols). This is visually annoying, but not unbearable.

Install Anaconda with Python 3.6

As a no-brainer for automatic machine learning, I choose to use [Auto-sklearn](https://$ utoml.github.io/auto-sklearn/stable/installation.html). However, it only runs on Python 3, so I have to install a separate environment.

1. Create an environment:

   ```
   conda create -n python3 python=3.6 anaconda
   ```

2. To activate this environment, use: `source activate python3`.

   To deactivate an active environment, use: `source deactivate`.

Install Auto-sklearn

1. First, activate the `python3` environment.
2. Install dependencies by

   ```
   curl https://raw.githubusercontent.com/automl/auto-
   sklearn/master/requirements.txt |
   xargs -n 1 -L 1 pip install --user
   ```

   Please note that I have added the option `--user` to bypass the root privilege requi$ ement for `pip`.

3. Install `auto-sklearn` via `pip install auto-sklearn`.
4. Start `spyder`.

Bypass auto-logout

Now the evil part. Workstations in MOORE 100 are logged out every 30 minutes of idleness . Of course you want your estimator to be trained for as many hours as possible. You nee d a little script that automatically jiggles the mouse cursor once in a while.

Save this to `~/jiggle.sh`:

```
#!/bin/bash
while :
do
        xdotool mousemove_relative 1 1
        xdotool mousemove_relative -- -1 -1
        sleep 1500
done
```

Give it executable permission by calling `chmod +x ~/jiggle.sh`.

Now, whenever you feel like walking away for a couple of hours while leaving your estima tor learning, simply run `~/jiggle.sh` in a separate terminal.

Please only use this script when you are absolutely sure no one is going to use this mac hine. Please be considerate of your peers. I drafted this instruction to help people mak e use of **unused** computational resources in MOORE 100, not **robbing** them from othe r potential users.

Side Note: How To Install Common Linux Software Without Root Access

Let me use Typora, a great GUI Markdown edi tor, as example.

1. Download Typora.
2. Unzip the downloaded archive using `tar -zxvf Typora-linux-x64.tar.gz`.
3. Additionally, I moved the folder `Typora-linux-x64` to `~/Software`, and created a sy mbolic link `~/bin/typora` pointing to `~/Software/Typora-linux-x64/Typora`. Of course, I also added this line to my `.zshrc` file:

   ```
   export PATH="/home1/m/myli/bin:$PATH"
   ```