University of Notre Dame, Computer Science and Engineering

CSE 40647/60647 Data Mining — Assignment 0
Due Date: January 22nd, 2014 at 11:59pm ET

Setting-up Your Software Environment

January 17, 2014

This assignment simply requires you to setup the programming environment that we will be using for the remainder of the course. As mentioned in the syllabus, we are providing a virtual machine that comes pre-loaded with all of the software and Python modules you will need. If you choose to use the provided virtual machine, you will need to download VirtualBox, which is available for Linux, Mac, and Windows operating systems (follow section 1 below). If you prefer, you may also install the required software in your personal computer, in which case we suggest you carefully ensure that everything you install is up-to-date, which will help avoid compatibility issues when your assignments are graded (follow section 2 below).

1 Using the course virtual machine

Step 1 — Downloading the required files

First, you will need to download and install VirtualBox, as well as our class VM available here. These links can also be found on the course website under *Assignments*.

Step 2 — Loading and booting the virtual machine

Open up VirtualBox and select *New*. Fill in the required information as:

- **Name:** DataMiningVM
- **Type:** Linux
- **Version:**Ubuntu

Next, select an amount of memory that you would like to provide your virtual machine. It should be safe to provide close to half of your physical memory. So, if your PC has 4GB of RAM, select 2GB.

On the next window, select *Use an existing virtual hard drive file* and locate the course VM file extracted from the ZIP file you downloaded in step 1. The file should be called *cse40647_vm.vdi*.

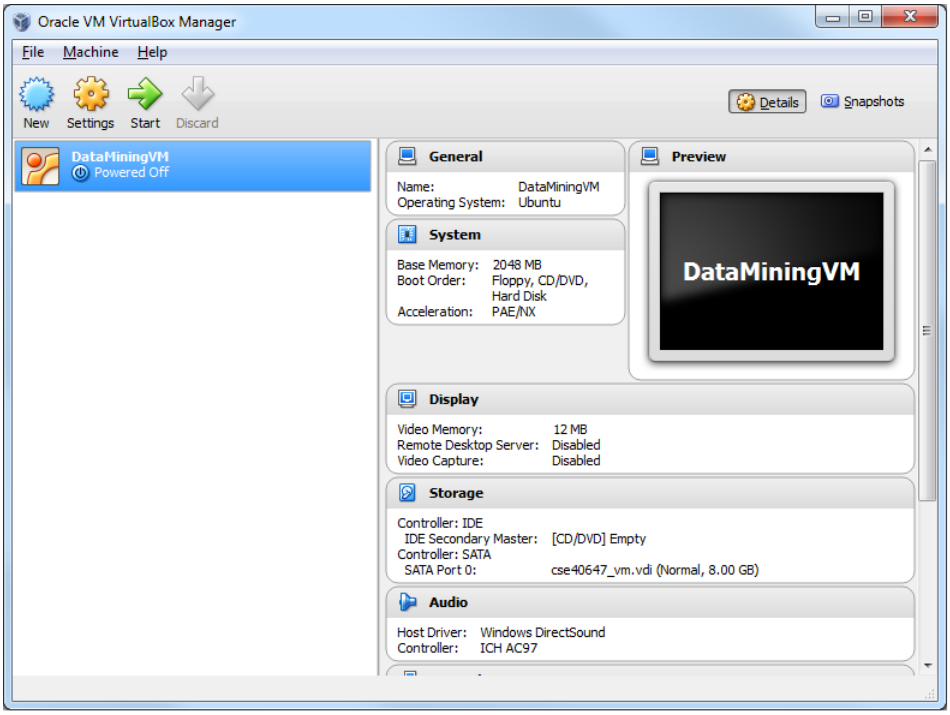

Figure 1.1: What you should see after successfully creating your VM from the provided VDI file.

When the process is complete, what you see should be roughly identical to the above window. Now select the DataMiningVM and click *Start*. When the machine boots, you will log in using the **student** account. The password for that account is **student**.

Done! That's it. You have everything you need.

Step 3 — Submission

Create an IPython Notebook and run the code in Listing 1 with any modifications you desire (e.g., print your name somewhere). Be sure to modify/add at least one line.

To create a new IPython Notebook, you simply need to open the terminal and run the following command:

```
ipython notebook --pylab inline
```

This will bring up the IPython web interface from where you may select *New Notebook*. Once you are finished, rename your Notebook **yourNetID_assignment0** and save it by going to File → Download as → IPython. This will generate an .ipynb file. Place this file in your dropbox at:

```
/afs/nd.edu/courses/cse/cse40647.01/dropbox/yourNetID
```

The final notebook should be similar to this one here.

2 Using your personal machine

Ensure that your machine has the following software installed:

- Python 2.7.3
- NumPy 1.8.0
- SciPy (library) 0.9.0
- Matplotlib 1.3.1
- pandas 0.12.0
- IPython 1.1.0
- scikit-learn 0.14.1

Complete this assignment by following *Step 3* above.

Listing 1: Sample Python code for testing required modules

```
# Testing pandas
import pandas as pd
ts = pd.Series(np.random.randn(1000),
               index=pd.date_range('1/1/2000', periods=1000))
ts = ts.cumsum()
ts.plot()

# Testing NumPy
import numpy as np
np.arange(15).reshape(3, 5)

# Testing SciPy
import scipy as sp
sp.linspace(0, 10, 5000)

#Testing matplotlib
import matplotlib.pyplot as plt
x = np.linspace(0, 1)
y = np.sin(4 * np.pi * x) * np.exp(-5 * x)
plt.fill(x, y, 'r')
plt.grid(True)
plt.show()

# Testing Scikit Learn
from sklearn.svm import SVC
from sklearn.datasets import load_digits
from sklearn.feature_selection import RFE

# Load the digits dataset
digits = load_digits()
X = digits.images.reshape((len(digits.images), -1))
y = digits.target

# Create the RFE object and rank each pixel
svc = SVC(kernel="linear", C=1)
rfe = RFE(estimator=svc, n_features_to_select=1, step=1)
rfe.fit(X, y)
ranking = rfe.ranking_.reshape(digits.images[0].shape)

# Plot pixel ranking
matshow(ranking)
colorbar()
title("Ranking of pixels with RFE")
show()
```