DS 6600: Data Engineering 1

Lab assignment 1

Instructions: use a Jupyter notebook to complete the following questions. Use a markdown cell for text and a code cell for Python code. Once your notebook is complete, follow these instructions to save the notebook as a PDF file and submit your assignment to Gradescope on Canvas.

1. We are going to use Git and GitHub for this lab assignment. Please complete all of the following steps prior to starting a Jupyter Notebook file for your work on this lab.

   a. Create a new GitHub repository for this lab assignment called "DS6600_lab1". Make it public, add a README.md file, choose a .gitignore file that is specific to Python-based projects. Also, choose a licence (using choosealicense.com) that allows other people to use the code in your repository for both commercial and non-commercial use, including full rights to distribute and modify the code, but does not allow anyone to modify and distribute your code with a more restrictive license than the one included in your repository. [1 point]

   b. In your terminal, choose a location on your computer for the local copy of this repository and use `git clone` to download it and connect this local directory to the GitHub repository. [1 point]

   c. In your README.md file, explain (in your own words! Copy-and-pasted answers will receive no credit) why you chose the license you did and what would have happened had you not supplied a license. In addition, examine your .gitignore file to confirm that a file named ".env" will not be pushed to GitHub. Copy and paste the relevant section of the .gitignore file in your README.md file. [1 point]

   d. As you work on this assignment, add, commit, and push your changes to GitHub. We will look at your repository's commit history to make sure there are at least 6 commits (one for each of the questions on this lab). [1 point]

For your assignment, all you need to supply in this notebook is a URL for your GitHub repository. Please type it here.

https://github.com/JackBeerman/DS6600_lab1

2. Use the `pyenv` package to install Python 3.11.4, if you don't currently have this version of Python installed. Then use the `pipenv` package to deploy a virtual environment that runs Python 3.11.4, `numpy` 1.25.2, `pandas` 2.0.3, `matplotlib` 3.7.2, `requests`

2.31.0, `jupyterlab` 4.0.5, and `ipykernel` 6.25.1. Finally, launch the virtual environment and use Jupyter Lab to launch the notebook you will use to write this lab assignment. Once you've got the virtual environment running, perform the following tasks:

a. In a raw cell (find the box in JupyterLab that allows you to select Code, Markdown, or Raw), copy and paste everything that appears in the Pipfile that was automatically generated by `pipenv`. [1 point]

b. In a code cell, type `!python --version` to display the version number of Python running in this virtual environment. [1 point]

c. In a code cell, import `numpy`, `pandas`, `matplotlib`, and `requests` to demonstrate that they can all be loaded without error. [1 point]

d. In a code cell, import `sklearn` and display the error to show that it has not been installed. [1 point]

After creating the pipfile and pipfile.lock files, commit these changes and push them to GitHub.

[[source]] url = "https://pypi.org/simple" verify_ssl = true name = "pypi" [packages] numpy = "==1.25.2" pandas = "==2.0.3" ipykernel = "==6.25.1" matplotlib = "==3.7.2" requests = "==2.31.0" jupyterlab = "==4.0.5" jupyter = "*" lab = "*" [dev-packages] [requires] python_version = "3.11" python_full_version = "3.11.4"

In [1]:
```
!python --version
```

```
Python 3.11.4
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib
import requests
```

In [3]:
```
import sklearn
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[3], line 1
----> 1 import sklearn

ModuleNotFoundError: No module named 'sklearn'
```

3. Describe, in words, whether a virtual machine, a container, a virtual environment, or the global environment of a single computer is best suited for each of the following situations. Be clear about why the option you choose works best, and also about why the other options are insufficient or are overkill.

   a. The local chapter of Meals on Wheels worked with some UVA computer science students and volunteers from WillowTree Apps, who built them a web portal for volunteers to sign up for shifts and a mobile app that shows volunteers their driving

routes with GPS mapping. The portal and app have been extremely useful for the local Meals on Wheels chapter, but unfortunately, the portal and app were built only for the Charlottesville chapter and is not usable by any other Meals on Wheels chapter at present. You've been asked to help every other Meals on Wheels chapter to benefit from these products by generalizing the code, and then making it available for free to any chapter that wants to use the portal and app. Every chapter has their own database of volunteers and clients, necessitating a large amount of storage that must also be accessible through an internet connection. [1 point]

b. The Legal Aid Justice Center, a Charlottesville-based legal aid nonprofit that advocates for the rights of immigrants and the working poor, has used a Freedom of Information Act request to get Virginia's Department of Labor and Industry (DOLI) to share their data on all official complaints of wage theft in Virginia over the last five years. The DOLI stored this data in an Excel file with horrible formatting: the column headers are given vague names, text has misspelling throughout the file (you see locations such as "Rochmond" and "Norflok"), data types are inconsistent within columns (sometimes a `$` is included with the wage amounts, sometimes `$` is excluded), and so on. They've asked you to clean the data and generate a report that lists the average claim of wage theft and a frequency table of the number of claims by locality. They don't need your code or data, just the report. [1 point]

c. An international NGO called Save the Children provides housing, food, medical care, and other assistance to individuals who have been displaced from their homes by conflict or natural disasters. But a major problem is that these displacement events can happen suddenly, without giving Save the Children any time to mobilize. Major migration events are more likely when the infrastructure and buildings in a city or town have been destroyed. Save the Children have asked you to build a predictive model that can detect from land-satellite images the extent of infrastructure damage in an area of interest and that can be deployed by several different individuals working for Save the Children. You build this model, but in addition to Python you require several open-source image processing software packages that only work on Ubuntu Linux. [1 point]

d. Sometime in the near future, Python announces the release of Python version 4. In this major update, all commands are now in Swedish, and any command written with English words such as "read" will now fail to run. You are willing to relearn Python in Swedish to use this much needed and long overdue upgrade, after all Python är ett programmeringsspråk som låter dig arbeta snabbt och integrera system mer effektivt. But you don't want the code you are writing for your current project in Python 3 to stop working once Python 4 is released. [1 point]

A

In this scenario, it would be best to have each chapter on a virtual machine. The main reason behind this selection is the need for a large amount of storage for indiviudal databases of each chapter. If we used a container, we would be using the processing power of individual devices. The VM allows users to connect over the internet with the OS set so there are no issues running the application. A virtual environment and global environment would have issues with the OS and the storage for databases.

B

I would use my global environment for this scenario because the organization does not need to run my code and just wants the final report. My global environment has the capaibility to complete these tasks and probably has all the nessecary packages in Python installed for this task. Any other option would be an overkill.

C

In this scenario, I would utilize a VM because of the Python packages that only work on Ubuntu Linux. The VM would have the OS as Ubuntu Linux. In addition, the land-satellite images are going to require a large amount of processing power that my own device could probably not manage. Other people may not be able to manage the amount of data if they were using a container so it would be best to have them connect to a VM.

D

For this last scenario, I would use a virtual environment to learn Python version 4. Using a virtual environment will ensure that I do not create issues on my global environment. A VM or Container could create the speration from my global environment but would just be an overkill.

4. The official Python images on Docker Hub use a version of Linux called Debian. However, sometimes you might need to install additional software in a container other than Python and Python packages, and a lot of open source software only works on another version of Linux called Ubuntu.

   a. Write a Dockerfile that builds an image from the `ubuntu:latest` image on Docker Hub. Run the following Linux command to install Python 3 onto Ubuntu: `apt-get install -y python3`. Once Python is installed, run the following command to launch Python from the command line: `python3`. Copy and paste the Dockerfile here. [2 points]

   b. Prove that the Dockerfile is written correctly by building the image associated with this Dockerfile. Once you are able to build the image successfully, copy and paste the output of the build in your terminal here. [2 points]

c. We've mostly run Docker containers by using the `-p` tag to map the operations of a container to a port on our local machine. But another way to run a container is in "interactive mode" using the `-it` tag, which immediately replaces your command line with the command line that exists inside the container. If you've correctly specified your Dockerfile to both install and launch Python inside the container, running this container in interactive mode should result in showing you a Python command line. Type `docker run -it` followed by the name of the image you created in parts (a) and (b). Confirm that the Python prompt appears. Then copy-and-paste the output from your terminal from the `docker run` command here. [1 points]

syntax=docker/dockerfile:1 FROM ubuntu:latest RUN apt-get update && apt-get install -y python3 WORKDIR /DS6600_lab1 EXPOSE 8888 CMD ["python3"][+] Building 1.8s (11/11) FINISHED docker:default => [internal] load build definition from Dockerfile 0.1s => => transferring dockerfile: 203B 0.0s => [internal] load .dockerignore 0.0s => => transferring context: 2B 0.0s => resolve image config for docker.io/docker/dockerfile:1 0.8s => [auth] docker/dockerfile:pull token for registry-1.docker.io 0.0s => CACHED docker-image://docker.io/docker/dockerfile:1@sha256:ac85f3 0.0s => [internal] load metadata for docker.io/library/ubuntu:latest 0.3s => [auth] library/ubuntu:pull token for registry-1.docker.io 0.0s => [1/3] FROM docker.io/library/ubuntu:latest@sha256:aabed3296a3d45c 0.0s => CACHED [2/3] RUN apt-get update && apt-get install -y python3 0.0s => CACHED [3/3] WORKDIR /DS6600_lab1 0.0s => exporting to image 0.0s => => exporting layers 0.0s => => writing image sha256:5834e3773faf1d498b263314becec199ca999001e 0.0s => => naming to docker.io/library/ds6600lab1PS C:\Users\jackt\OneDrive\Desktop\UVA\DataEngineering\DS6600_lab1> docker run -it ds6600lab1 Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux Type "help", "copyright", "credits" or "license" for more information. >>>

5. For this problem, you will create everything you need for a data engineering project that uses Python and postgreSQL. There's nothing you need to write on this document. We will look on Github to find all the files for this problem. [7 points]

    - Create a subdirectory of your project folder called "problem5". (You can create this folder on your computer -- it will be pushed to GitHub when you run the `git add`, `git commit`, and `git push` commands). Inside this folder, create four new files: Dockerfile, requirements.txt, .env, and compose.yaml.
    - In your requirements.txt file, list numpy 1.25.2, pandas 2.0.3, sqlalchemy 2.0.20, psycopg 3.1.10, and jupyterlab 4.0.5.
    - In your .env file, create an environmental variable called POSTGRES_PASSWORD, and choose a password for this variable.
    - In your Dockerfile, build from the `python:3.11.4-bookworm` image. Update pip and install the packages from the requirements.txt file. Expose port 8888 and call the root working directory "problem5". Then run Jupyter Lab.
    - Build an image from your Dockerfile, and post it to your public Docker hub account.
    - In your compose.yaml file, create two services: one for jupyterlab and one for postgres. Also create a volume called "postgresdata" and a network called "dbnetwork".
    - For the jupyterlab service: load the image you just posted to Docker hub, load the environmental variables file, map the container's "problem5" directory to your local project directory, and map the container's port 8888 to a port on your own computer. Finally, attach this service to the "dbnetwork" network.

- For the postgres service: load the `postgres:latest` image from Docker hub, load the (same) environmental variables file, attach the "postgresdata" volume to the container's /var/lib/postgresql/data folder, and map the containers port 5432 to a port on your own computer. Finally, attach this service to the "dbnetwork" network.
- Confirm that your code is correct by running `docker compose up` in your terminal.
- Make sure all of these files are contained within the "problem5" folder and pushed to GitHub.

6. In the terminal, run the following command: `docker run -it matsuu/nethack` After answering the first few questions, you will receive a passage from a sacred book, scroll, or text. Copy the passage and paste it here. [1 point]

It is written in the Book of Lugh: ------ After the Creation, the cruel god Moloch rebelled .u.... against the authority of Marduk the Creator. |.@..| Moloch stole from Marduk the most powerful of all |....| the artifacts of the gods, the Amulet of Yendor, |..... and he hid it in the dark cavities of Gehennom, the ------ Under World, where he now lurks, and bides his time. Your god Lugh seeks to possess the Amulet, and with it to gain deserved ascendance over the other gods. You, a newly trained Gallant, have been heralded from birth as the instrument of Lugh. You are destined to recover the Amulet for your deity, or die in the attempt. Your hour of destiny has come. For the sake of us all: Go bravely with Lugh! --More-- Jack_Beerman the Gallant St:14 Dx:9 Co:12 In:7 Wi:15 Ch:18 Lawful Dlvl:1 $:0 HP:16(16) Pw:3(3) AC:3 Xp:1 Weapons a - a blessed +1 long sword (weapon in hand) ------ b - a +1 lance (alternate weapon; not wielded) .u.... Armor |.@..| c - an uncursed +1 ring mail (being worn) |....| d - an uncursed +0 helmet (being worn) |..... e - an uncursed +0 small shield (being worn) ------ f - an uncursed +0 pair of leather gloves (being worn) Comestibles g - 12 uncursed apples h - 13 uncursed carrots (end) No Points Name Hp [max] 0 Jack_Beerm-Kni-Hum-Mal-Law quit in The Dungeons of Doom on level 1. 16 [16]