MCIS6273 Data Mining (Prof. Maull) / Spring 2024 / HW0

Points Possible	Due Date	Time Commitment (estimated)
20	Wednesday February 7 @ Midnight	*up to* 12 hours

- **GRADING:** Grading will be aligned with the completeness of the objectives.
- **INDEPENDENT WORK:** Copying, cheating, plagiarism and academic dishonesty *are not tolerated* by University or course policy. Please see the syllabus for the full departmental and University statement on the academic code of honor.

OBJECTIVES

- Familiarize yourself with Github and basic git
- Familiarize yourself with the JupyterLab environment, Markdown and Python
- Explore JupyterHub Linux console integrating what you learned in the prior parts of this homework
- Perfom basic data engineering in Python using NOAA weather data

WHAT TO TURN IN

You are being encouraged to turn the assignment in using the provided Jupyter Notebook. To do so, make a directory in your Lab environment called `homework/hw0`. Put all of your files in that directory. Then zip or tar that directory, rename it with your name as the first part of the filename (e.g. `maull_hw0_files.zip`, `maull_hw0_files.tar.gz`), then download it to your local machine, then upload the `.zip` to Blackboard.

If you do not know how to do this, please ask, or visit one of the many tutorials out there on the basics of using zip in Linux.

If you choose not to use the provided notebook, you will still need to turn in a `.ipynb` Jupyter Notebook and corresponding files according to the instructions in this homework.

ASSIGNMENT TASKS

(0%) Familiarize yourself with Github and basic git

Github (https://github.com) is the *de facto* platform for open source software in the world based on the very popular git (https://git-scm.org) version control system. Git has a sophisticated set of tools for version control based on the concept of local repositories for fast commits and remote repositories only when collaboration and remote synchronization is necessary. Github enhances git by providing tools and online hosting of public and private repositories to encourage and promote sharing and collaboration. Github hosts some of the world's most widely used open source software.

If you are already familiar with git and Github, then this part will be very easy!

§ Task: Create a Zotero account.

Learn about Zotero and if you haven't already, create a free account:

- https://zotero.org

§ Task: Create a public Github repo named `"mcis6273-f24-datamining"` and place a `README.md` file in it.

Create your first file called `README.md` at the top level of the repository.

Please put your Zotero username in the file. Aside from that you can put whatever text you like in the file (If you like, use something like lorem ipsum to generate random sentences to place in the file.). Please include the link to **your** Github repository that now includes the minimal `README.md`. You don't have to have anything elaborate in that file or the repo.

§ Task: Fork the course repository.

Learn to use Github workflows and fork the class repo:

- https://github.com/kmsaumcis/mcis6273_f24_datamining/

(0%) Familiarize yourself with the JupyterLab environment, Markdown and Python

As stated in the course announcement Jupyter (https://jupyter.org) is the core platform we will be using in this course and is a popular platform for data scientists around the world. We have a JupyterLab setup for this course so that we can operate in a cloud-hosted environment, free from some of the resource constraints of running Jupyter on your local machine (though you are free to set it up on your own and seek my advice if you desire).

You have been given the information about the Jupyter environment we have setup for our course, and the underlying Python environment will be using is the Anaconda (https://anaconda.com) distribution. It is not necessary for this assignment, but you are free to look at the multitude of packages installed with Anaconda, though we will not use the majority of them explicitly.

As you will soon find out, Notebooks are an incredibly effective way to mix code with narrative and you can create cells that are entirely code or entirely Markdown. Markdown (MD or `md`) is a highly readable text format that allows for easy documentation of text files, while allowing for HTML-based rendering of the text in a way that is style-independent.

We will be using Markdown frequently in this course, and you will learn that there are many different "flavors" or Markdown. We will only be using the basic flavor, but you will benefit from exploring the "Github flavored" Markdown, though you will not be responsible for using it in this course – only the "basic" flavor. Please refer to the original course announcement about Markdown.

§ Task: THERE IS NOTHING TO TURN IN FOR THIS PART.

Play with and become familiar with the basic functions of the Lab environment given to you online in the course Blackboard.

§ Task: THERE IS NOTHING TO TURN IN FOR THIS PART.

Please *create a markdown document* and read the documentation for basic Markdown here. Learn to use all of the following:

- headings (one level is fine),
- bullets,
- bold and italics

Again, the content of your documentcan be whatever you like, just learn some of the basic functionality of Markdown.

(0%) Explore JupyterHub Linux console integrating what you learned in the prior parts of this homework

The Linux console in JupyterLab is a great way to perform command-line tasks and is an essential tool for basic scripting that is part of a data scientist's toolkit. Open a console in the lab environment and familiarize yourself with your files and basic commands using git as indicated below.

1. In a new JupyterLab command line console, run the `git clone` command to clone the new repository you created in the prior part. You will want to read the documentation on this command (try here https://www.git-scm.com/docs/git-clone to get a good start).
2. Within the same console, modify your `README.md` file, check it in and push it back to your repository, using `git push`. Read the documentation about `git push`.
3. The commands `wget` and `curl` are useful for grabbing data and files from remote resources off the web. Read the documentation on each of these commands by typing `man wget` or `man curl` in the terminal. Make sure you pipe the output to a file or use the proper flags to do so.

§ Task: THERE IS NOTHING TO TURN IN FOR THIS PART.

(70%) Perfom basic data engineering in Python using NOAA weather data

You should already know by now that data science is one of Python's strengths.

In this part, you will interact directly with those strengths, but in a way that will allow you to see the challenges that you will face and confront as a real-world data scientist.

Data engineering as you have learned from the readings is about transforming data from one form to another so that it can be used in the appropriate analysis contexts.

One area of intense work is in transforming data. More importantly, producing statistical analyses of these data is often difficult.

In this part of the homework you will prepare data for analysis. We will make it simple this time so that it won't take too much time to extract.

The last week or so, as you know, has been cold all over the US. One source for weather data that is free comes from NOAA (National Oceanic and Atmospheric Administration).

For this first part, you will need to get an API key from NOAA to obain their data.

The API key is free – just go to:

https://www.ncdc.noaa.gov/cdo-web/token

put in your email address and a few seconds you will get the key at that address. This will be used for this part of the assignment.

In this part, we will grab data from NOAA, store it on disk and then use it for the second part.

Your code must be implemented in Jupyter as a notebook – you will be required to turn in a `.ipynb` file.

§ Task: Use Python to make HTTP/API calls to NOAA services and obtain data.

You will produce a series of `.json` files which will have all the data we are looking for.

You will need to write the code to do the following:

- obtain data for the Denver International Airport (zipcode 80249) for the date ranges listed below and store them in JSON files on the file system in a folder called `'data/'` relative to your notebook **you will lose points if you do not make the folder and all files are in that folder**
- use the `'/data'` API endpoint at `https://www.ncei.noaa.gov`
- you can learn how to use it here: https://www.ncdc.noaa.gov/cdo-web/webservices/v2#data
- you will need to make certain to restrict the data to the following query parameters:
  - `'datasetid=GHCND'`
  - `'locationid=ZIP:80249'`
  - `'units=standard'`
- you will need to make repeated calls to obtain data for the following years:
  - 2008-2022
- you will need to obtain data ONLY for the following ranges:
  - 12/15 to 1/21 (the following year)
  - the `'startdate'` and `'enddate'` query paramaters will be useful
- to avoid making multiple calls to the endpoint, set the query paramter `'&limit=1000'`

We are basically going to look at data just over the 2 weeks before the end of the year (12/15-12/31) through the first 3 weeks of the beginning of the year (1/1-1/21) for all years of interest (2008-2022).

You will end up with 15 JSON files in the `'data\'` folder:

- they will be named `'winter_2008-2009.json'`, `'winter_2009-2010.json'`, and so on until the final final `'winter_2021-2022.json'`
- you will store the data from the calls to the API in each of the files.

MINIMUM EXPECTED ANSWER FOR THIS TASK PART

Your answer must include:

- the `'data/'` folder
- the 15 JSON files corresponding to the 15 calls to the API that retrieve the data being requested above *in* the `'data/'` folder
- code that performs the API retrieval and data storage

§ Task: Extract, transform and export JSON data.

You will combine all the JSON files into a single CSV file with the following characteristics:

- the row index will be the date in the form 20nn-mm-d, (e.g. 2008-12-31)
- the columns will be restricted to 3 columns, `TMAX`, `TMIN` and `TAVG`
  - `TAVG` is derived and not in the data, thus you will need to calculate it with $t_{avg} = \frac{t_{max}+t_{min}}{2}$.
- the file must be CSV and be named `'all_data_max_min_avg.csv'` and be in the `'data/'` folder

§ Task: Filter, transform and export CSV data.

You will now loop over all the data files to produce a single file that compiles the data such that

- each column in the file contain the year (e.g. '2008-2009') and
- the rows contain the dates (`'12-15'`, `'12-16'`, ..., `'1-21'`).
- and the values in each row-column are the `'TAVG'`

Your data will look like this:

	2008-2009	**2009-2010**	⋯	**2020-2021**	**2021-2022**
12-15	17.8	36.9	41.4	10.9	40
12-16	36.6	7.5	0.4	12.2	3.4
⋮			⋱		⋮
1-20	8.2	31.4	34.5	44.7	2
1-21	31.6	24.7	38.5	35.7	-9.9

- you must store the data table in a CSV file called `'all_data_min.csv'` in the `'data/'` folder

§ Task: Write functions to compute temperature averages.

You will need to write three functions:

- one called `average()` which takes a DataFrame as a parameter and returns the average the DataFrame
- one called `average_high()` which takes a DataFrame as a parameter and returns the average of the maximum `'TMAX'` of the DataFrame
- one called `average_low()` which takes a DataFrame as a parameter and returns the average of the minimum `'TMIN'` of the DataFrame

You will need these for the online assessment! Do your best.

§ Task: Complete the online HW0 assessment.

Once you are done with the coding part of the assignment, you will need to complete the online assessment for the final 6 points of your grade.

§ Task: BONUS

You can earn up to 3 points extra for this part of the assignment.

Use the Seaborn library to create a heatmap of the TAVG for 2008-2022.

- the y-axis should be the year (e.g. `2008-2009`, `2010-2011`, etc) and x-axis the day (e.g. `12-15`, `12-16`, etc)