MCIS6273 Data Mining (Prof. Maull) / Fall 2020 / HW0

This assignment is worth up to 20 POINTS to your grade total if you complete it on time.

Points Possible	Due Date	Time Commitment (estimated)
20	Wednesday, Sep 2 @ Midnight	*up to* 20 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 the JupyterLab environment, Markdown and Python
- Familiarize yourself with Github and basic git
- Explore JupyterHub Linux console integrating what you learned in the prior parts of this homework
- Listen to the O'Reilly Data Show Podcast from July 18, 2019: Acquiring and shairing high-quality data with Rogen Chen
- Explore Python for data munging and analysis, with an introduction to CSV and Pandas

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 that directory, rename it with your name as the first part of the filename (e.g. `maull_hw0_files.zip`), 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 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 contraints 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 envitonment 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.

§ **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.

§ **PLEASE *CREATE A MARKDOWN DOCUMENT* CALLED `semester_goals.md` WITH 3 SENTENCES/FRAGMENTS THAT ANSWER THE FOLLOWING QUESTION:**

- **What do you wish to accomplish this semester in Data Mining?**

Read the documentation for basic Markdown here. Turn in the text `.md` file *not* the processed `.html`. In whatever you turn in, you must show the use of *ALL* the following:

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

Again, the content of your document needs to address the question above and it should live in the top level directory of your assignment submission. This part will be graded but no points are awarded for your answer.

(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!

§ **CREATE A PUBLIC GITHUB REPO NAMED `"mcis6273-F20-datamining"` AND PLACE A README.MD FILE IN IT.** Create your first file called `README.md` at the top level of the repository. 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.

(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.

§ **THERE IS NOTHING TO TURN IN FOR THIS PART.**

(40%) Listen to the O'Reilly Data Show Podcast from July 18, 2019: Acquiring and shairing high-quality data with Rogen Chen

Data is one of the most important components of modern business and, of course, data manifests in many forms, and involves producers, consumers and value-added uses of data. Recently, data has become the source of contention, especially as large companies continue to take control of individual's data and use it in ways that do no benefit, and in some cases are used in ways that were neither approved or known by, the the individual directly.

There are a number of interesting movements in data protection, such as the European Union's GDPR (General Data Protection Regulation - https://eugdpr.org) that are trying to hold companies account for how individual's data are collected, accessed, shared, distrubuted and protected. Some argue these efforts don't go far enough to acknowledge the true value of data and don't provide access to controlling data in ways that would benefit the individual directly, should they want to have it knowingly used and offered as a product to companies. In scenarios like these, the individual is then brought into the data transaction as a producer, and companies act as consumers that add value to other consumers, etc.

Blockchain has been proposed as a technology that might provide a way to coordinate the development of decentralized and distrubuted data "ledgers" that may provide a more complete platform to share data, trace it's lineage, but also validate and facilitate data transactions between consumers and producers.

In this podcast, Ben Lorica interviews Roger Chen, CEO of Computable Labs, about his recent paper *"Fair value and decentralized governance of data"*. Though it is not necessary, you are encouraged to read the very accessible white paper Chen, et al published in June which forms the basis of parts of the conversation in the podcast (that paper is on Github here: https://git.io/fjpTv).

Please listen to the Podcast released July 18, 2019 *"Acquiring and Sharing High-Quality Data"* on the O'Reilly Data Show. You can listen to it from one of the two links below:

- https://soundcloud.com/oreilly-radar/acquiring-and-sharing-high-quality-data
- https://www.oreilly.com/ideas/acquiring-and-sharing-high-quality-data

§ PLEASE ANSWER THE FOLLOWING 8 QUESTIONS AFTER LISTENING TO THE PODCAST:

1. Early in the interview Roger was asked what problems were being approached (and attempted to being solved) through the white paper "Fair value and decentralized governance of data". What two core areas of data did Roger say he and his Computable Labs company trying to solve?
2. In the Enterprise use case, what did Roger say about how to determine who owns data?
3. What did Roger say are some of the problems with company's centralizing data as it stands right now?
4. Roger describes a simple example of the basic workflow of how data is served by his technology through the use of decentralized "protocol contracts" and what he calls the "Datatrust" storage. In your own words, describe how this process works.
5. Roger suggests a data marketplace model of service providers so that there is decentralized universal access to data, and mentions how "next generation crowd-source models" will propel the fair market value of data. What property did he say this fair market value would be based on to derive this value?
6. Roger is bullish on Data privacy and encryption technologies and how we might have a future where data on the Internet is open, fully encrypted and secure "in the open". What two companies did he think would be one's to watch in this space?
7. What is a "synthetic" dataset (as described in the interview), and what problems did Ben challenge Roger on with a market where these datasets might exist?
8. Which Blockchain network will the project be available for testing?

(60%) Explore Python for data munging and analysis, with an introduction to CSV and Pandas

Python's strengths shine when tasked with data munging and analysis. As we will learn throughout the course, there are a number of excellent data sources for open data of all kinds now available for the public. These open data sources are heralding the new era of transparency from all levels from small municipal data to big government data, from transportation, to science, to education.

To warm up to such datasets, we will be working with an interesting dataset of enrollment data from the University of Hawaii system. This data is in a CSV (comman-separated-values) format and like many datasets out there, requires a bit of restructuring to derive value from it.

CSV is a format that you will often see as a *de facto* format for data that is relatively flat or well adapted to tabular analyses. You will also see JSON as the standard format for developing structured data, but luckily we will be able to use Python tools (specifically Pandas) to work with these types of data easily.

The data we are going to use can be accessed via APIs but we will not need to access it that way for this part of the exercise. Instead, we will be able to load it directly.

In this part of the assignment, we will make use of Python libraries to pull the data from the endpoint and use Pandas to plot the data. The raw CSV data is readily imported into Pandas from the following URL:

- Hawaii.gov Open Data for University of Hawaii enrollment

Please take a look at the page, on it you will notice a link to the raw CSV file:

- https://opendata.hawaii.gov/dataset/01dc1eab-6627-4843-87c3-f78b922fca4e/resource/18857159-3524-4abc-8cb1-107b4207e734/download/enrollment-by-zipcode.csv

We are going to explore this dataset to learn a bit more about the enrollment characteristics of the University system over the past half decade or so.

§ WRITE THE CODE IN YOUR NOTEBOOK TO LOAD AND RESHAPE THE COMPLETE CSV ENROLLMENT DATASET:

You will need to perform the following steps:

1. **use `pandas.read_csv()` method to load the dataset** into a Pandas DataFrame;
2. **clean the data so that all zip codes are normalized to just the ZIP5 format (e.g. 80246) not the ZIP5 + 4 (e.g. 80248-1234) format**; some of the data do not adhere the ZIP5 at all, so just remove them completely;
3. **create a column `YEAR` in your dataset for the year** you will notice that the original column `SEMESTER` is the form `Fall 20nn`. You will need to split this and just keep the year;
4. **drop the `HAWAIIAN_LEGACY` column** – we will not need it;
5. **eliminate any rows of data with `NaN` data in it**;
6. **store the entire dataset back into a new CSV** file called `hawaii_enrollments.csv`.

HINTS: *Here are some a code hints you might like to study and use to craft a solution:*

- study the `pandas.Series.str` functionality to understand how you might turn the zip codes into ZIP5 format.
- study the `pandas.DataFrame.dropna()` method to drop rows with `NaN` values.
- study the `pandas.DataFrame.drop(columns='xxx')` to understand how to remove columns from your DataFrame.

§ USE PANDAS TO LOAD THE CSV DATA TO A DATAFRAME AND ANSWER THE FOLLOWING QUESTIONS:

1. How many total students were cummulatively enrolled in total from 2014 to 2019?
2. Which campus had the most cummulatively enrolled students from 2014-2019 (and what is that value)?
3. Which campus has the least?
4. Which Hawaii zip code does the largest number of students come from (across all campuses)?
5. Which Hawaii zip code does the largest number of students come from for the Manoa flagship campus?

To answer these questions, you'll need to dive further into Pandas, which is the standard tool in the Python data science stack for loading, manipulating, transforming, analyzing and preparing data as input to other tools such as Numpy (http://www.numpy.org/), SciKitLearn (http://scikit-learn.org/stable/index.html), NLTK (http://www.nltk.org/) and others.

For this assignment, you will only need to learn how to load and select data using Pandas.

- **LOADING DATA** The core data structure in Pandas is the `DataFrame`. You will need to visit the Pandas documentation (https://pandas.pydata.org/pandas-docs/stable/reference/) to learn more about the library, but to help you along with a hint, read the documentation on the `pandas.read_csv()` method.
- **SELECTING DATA** The tutorial here on indexing and selecting should be of great use in understanding how to index and select subsets of the data to answer the questions.
- **GROUPING DATA** You may use `DataFrame.value_counts()` or `DataFrame.groupby()` to group the data you need for these questons.

CODE HINTS

Here is example code that should give you clues about the structure of your code for this part.

```
import pandas as pd

df = pd.read_csv('your_json_file.json')

# code for question 1 ... and so on
```

§ GATHER NON-HAWAII STUDENT INFORMATION

Now that we have a decent idea of the in-state profile of students going to all campuses in the University of Hawaii system, let's look at the out-of-state profile. As you may know, many universities have large out-of-state student populations and some universities rely on out-of-state tuition as an important source of operating revenue for the university. Let's explore what the out of state profile looks like at the flagship Manoa campus.

For this part you will do well to use an API to assign a state code to each zip code. What you will be expected to do is the following steps:

1. aggregate all zip codes from 2014-2019 for the Manoa campus,
2. assign a state code to the zip codes,
3. get an aggregate count of all enrollment numbers by state from 2014-2019.

In order to assign state codes, you may use a number of techniques.

OPTION ONE (SLOWEST): USE AN API

One option is to use external APIs, such the one free API zipcodeapi.com which allows you to query by zipcode and return a JSON payload which contains information such as city, state, etc. To help you along, you will can just adapt the code below:

```
import requests
import time

# 967xx-969xx are Hawaii zip codes
if len(z) == 5 and z[:3] not in ['969', '968', '967']:
    url = f"https://www.zipcodeapi.com/rest/
            gUA29rOpQB4A2Nt6ZxL69KDokVOKy2xfoR9KZR5iIOTi2rj6GoCaaJtQWxNkcW2q/info.json/{z}/degrees"
    r = requests.get(url)

    if r.status_code == 200:
        payload = r.json()
        # do work  with  your JSON payload to obtain the state code
    else:
        pass
        # print(r.status_code)
    time.sleep(2)  # NOTE: the delay is to avoid rate limiting!
```

Just remember the JSON payload looks like this (verify for yourself):

```
{
"zip_code":"40504",
"lat":38.042391,
"lng":-84.543931,
"city":"Lexington",
"state":"KY",
"timezone":{
    "timezone_identifier":"America\/New_York",
    "timezone_abbr":"EDT",
    "utc_offset_sec":-14400,
    "is_dst":"T"},
"acceptable_city_names":[],
"area_codes":[859]
}
```

OPTION TWO (EASIEST): USE A GEONAMES.ORG PYTHON LIBRARY

Lucky for us, there is yet another service called Geonames.org which not only has an API, but others have created Python libraries on top of that API. Geonames, much like zipcodeapi.com, provides mappings from zip to city, state and other information. It also provides much, much more if you ever need to obtain latitude or longitude data for your programming and data needs. One such library written for Python is the open source `pgeocode`, and you may use this library.

In order to install it, open a terminal in your JupyterLab session on the Hub and type:

```
pip install pgeocode
```

Full documentation for the library is here https://pgeocode.readthedocs.io/en/latest/overview.html, but your solution might look something like this:

```
import pgeocode

nomi = pgeocode.Nominatim('us')
nomi.query_postal_code("13804")['state_code']
```

Which would return:

```
'ny'
```

With this you could craft a solution quite easily to map the zipcode to the state and then build up what is being asked.

OPTION THREE (EASIER): USE A STATIC FILE FROM GEONAMES.ORG

Another last option would be to load the full Geonames.org zipcode data file into a DataFrame and query it to obtain the data you're needing.

- the Geonames file is here: https://download.geonames.org/export/zip/US.zip

You can download it from the terminal with (the capitalization matters):

```
wget https://download.geonames.org/export/zip/US.zip
unzip US.zip
```

Or you can create a Jupyter cell with the magic command:

```
!wget https://download.geonames.org/export/zip/US.zip
!unzip US.zip
```

Whichever way you choose, you will then have a file `US.txt`. Load it into a DataFrame with:

```
import pandas as pd
df_zip = pandas.read_csv("US.txt", header=None, sep='\t')
```

Then you will be able to get the state with something like this:

```
df_zip[ df_zip[1]==57641 ][4].item()  # the zero-based index 1st item is zip, 4th the state code
```

Which will return:

```
'SD'
```

No matter the method you choose, your solution will include a DataFrame that will simply be the aggregation of the sum of zip codes or something like this (HINT: `value_counts()` will be useful again):

	ENROLLMENT
…	…
AR	911
ND	110
WV	610
DC	86
DE	26
…	…

§ Export that table to a CSV using `DataFrame.to_csv('zip_code_summary.csv')`.

Make sure the file name is as specified, so I can verify the contents.

§ Plot the top 15 (non-Hawaii) states by enrollment:

Using the `Series.plot(kind='bar')`, plot a simple bar plot of the top 15 states using the solution from the previous part. This is relatively easy and is one line of code.