MCIS6273 Data Mining (Prof. Maull) / Spring 2024 / HW2

Points Possible	Due Date	Time Commitment (estimated)
30	Wednesday April 24 @ Midnight	*up to* 24 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

- Perfom basic data engineering, visualization and data analysis in Python using an external set data.
- Use GeoPandas analyze and map datasets
- Perform clustering using Gaussian Mixture Model (EM) in SciKit-Learn
- Complete the online HW assessment.

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/hwX`. 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_HWX_files.zip`, `maull_HWX_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

(20%) Perfom basic data engineering, visualization and data analysis in Python using an external set data.

As usual, we will continue the practice of data engineering to prepare data for analysis.

This time, we will use two different open datasets and along the way learn some new tools that will be valuable in our analysis and data mining toolkit.

For this particular assigment, we will *pretend that we are data scientists who have been hired to understand the electronic vehicle (EV) market* in the state of Minnesota.

We have been given a dataset of all the registrations of EVs in the state since 2010. You will need to obtain this dataset from here:

- State EV Registration Download (Atlas EVHUB)
- Direct link to CSV file: MN_EV_Registrations.csv

And you will be asked to first run some basic statistics over it, then merge it with other data for further analysis. You should explore the website a bit and understand the data columns (there aren't many).

As before, all of your code must be implemented in Jupyter as a notebook – you will be required to turn in a **working** `.ipynb` file.

§ Task: Use Python to obtain and prepare data.

- Load the *original* file and save the CSV to a folder called `"data/"`.

§ Task: Transform, filter and store the data as a new CSV

- Create a new column called "`registration_year`", to represent the registration year.
  You will find `pandas.to_datetime()` on the `Registration Date` column to be useful. You will need to strip the date for just the year (see `.dt`).
- Drop any "`Vehicle Make`" which did not sell more than 100 cars over the entire time frame of the date. **HINT**: There are 22 makes and a total of 455 rows of data which will be removed. You will end up with about 367.6k rows of final data.
- Filter the columns to just the set: `'ZIP Code', 'Vehicle Make', 'Vehicle Model', 'registration_year', 'Vehicle Model Year'`
- Store the final file in the `"data/"` folder and name it "`FILTERED_MN_EV_Registrations.csv`".

§ Task: Plot the data using a Bar graph

- Using a bar graph, plot the frequency of registrations by year. That is the x-axis will contain the year, and y-axis the frequency.

§ Task: React to the following statements:

- The largest number of *new* registrations was in 2023.
- The number of new registrations slowed in 2019.

Use evidence to support your reactions!

NOTE: Assume a car needs to be registered each year, so the data is cummulative.

(30%) Use GeoPandas analyze and map datasets

We have been using Pandas for most of our work, and in general it is the data manipuation workhorse of our data science workflows. Interestingly, there are other libraries which you will run across that can do things Pandas cannot do, and while we're not going to get into large enough datasets in this assigment, you will well to learn about Dask, Datatable and Vaex.

We will turn our attention, however, to an area of great interest in data science and analytics: geospatial data. We are all familiar with the basic visualizations of US maps showing the incidence of poverty, crime, income or any host of demographic information. Nearly all of these types of maps use specialized data formats which encode the geospatial details of the underlying data so that you don't have to.

One especially significant format in this GIS or Geographic Information Systems space is called "Shapefiles" or ".shp" files. Without going into too much detail, these files can encode not only the geospatial information (coordinates, polygons representing areas, etc), they can store underlying data such as demographic information or nearly anything else.

Learn about GIS and SHP files here:

- ESRI.com | "What is GIS?"
- gisresources.com | "Understanding Shapefile File Format"
- wiki.gis.com | Shapefile

In Python, working with Shapefiles occurs through a number of libraries, but the one important to this assignment is GeoPandas. Learn about it here:

- GeoPandas

This particular library is beyond cool and can do amazing things with very little code, and we are going to do just that.

In this next part of the assignment, we are going to use GeoPandas to show information about the demographics of Minnesota ZIP Codes.

Before we begin, we might be asking ourselves, why? Weren't we talking about EVs before and what does this have to do with working as pretend data scientists to understand the EV market?

Often, to understand the dynamics of new technology, and especially technology which requires significant financial investment, it is important to understand the details of who might be buying these items, and often it is not as easy as looking at a single dataset containing all required data. We will usually need to combine data from multiple sources to get at some of the questions we have.

Some of the questions being asked of the pretend data science team are the following:

- What is the frequency of and association within the ZIP codes of EV registration in Minnesota?
- Are there demographic patterns in these ZIP codes?
- Can we visualize these inquiries?

Before we continue, let's first ask, what do ZIP codes have to do with anything? There are many aspects of US ZIP codes which might surprise you. Social, economic and health scientists have uncovered staggering relationships between ZIP codes amd income, education, health outcomes, crime, etc., for example see the sample of these papers:

- Gobaud, Ariana N., et al. "Absolute versus Relative Socioeconomic Disadvantage and Homicide: A Spatial Ecological Case–Control Study of US Zip Codes." Injury Epidemiology, vol. 9, no. 1, Feb. 2022, p. 7. DOI.org (Crossref), https://doi.org/10.1186/s40621-022-00371-z.
- Nguyen, Quynh C., et al. "Twitter-Derived Neighborhood Characteristics Associated with Obesity and Diabetes." Scientific Reports, vol. 7, no. 1, Nov. 2017, p. 16425. DOI.org (Crossref), https://doi.org/10.1038/s41598-017-16573-1.
- Thomas, Avis J., et al. "ZIP-Code-Based versus Tract-Based Income Measures as Long-Term Risk-Adjusted Mortality Predictors." American Journal of Epidemiology, vol. 164, no. 6, Sept. 2006, pp. 586–90. DOI.org (Crossref), https://doi.org/10.1093/aje/kwj234.

These are but a short list of what you will find in the research. On to the task at hand …

We will be gathering demographic data from the *American Community Survey (ACS)* which gathers data about US demographics. We will use the latest survey data tethered to the 2020 Census, which covers 2016-2020.

NOTE: You will need to install `GeoPandas`, `folium` and `mapclassify` at the top of your notebook – they are **not** installed in the default environment on our Hub. Use :

```
!pip install geopandas mapclassify folium
```

to install the required libraries.

§ Task: Load the Shapefile from the Minnesota ACS

You will find the main dataset here:

- https://gisdata.mn.gov/dataset/us-mn-state-metc-society-census-acs

The ZIP file can be downloaded directly from here:

- https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_metc/society_census_acs/shp_society_census_acs.zip

NOTE: You will unzip the file into a folder called "`acs/`" and load the file containing ZIP Code demographic summaries: `CensusACSZipCode.dbf`.

§ Task: Build a map of the ZIP Census Tract Areas (ZCTA)

This can be accomplished with `.plot()`

§ Task: Explore variables in the data

You will need to study the data fields that you have now loaded.

See this for more information on what each column name means:

- https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_metc/society_census_acs/metadata/metadata.html

Answer the following questions:

1. What is the mean Houshold Income (`MEDIANHHI`) in the dataset?
2. How does this compare with the median HHI for the entire US in 2020? (You will need to find that yourself.)
3. Which ZIP code has the highest HHI?
4. What are the top 5 ZIP codes with the largest percent population under 18 years of age? (You will need to remember to use the total population of the ZIP as the denomenator for each ZIP.)
5. Which 5 ZIP codes have the highest percent of professional / graduate degrees?

§ Task: Plot an interactive demographics map of ZIP codes with high value homes

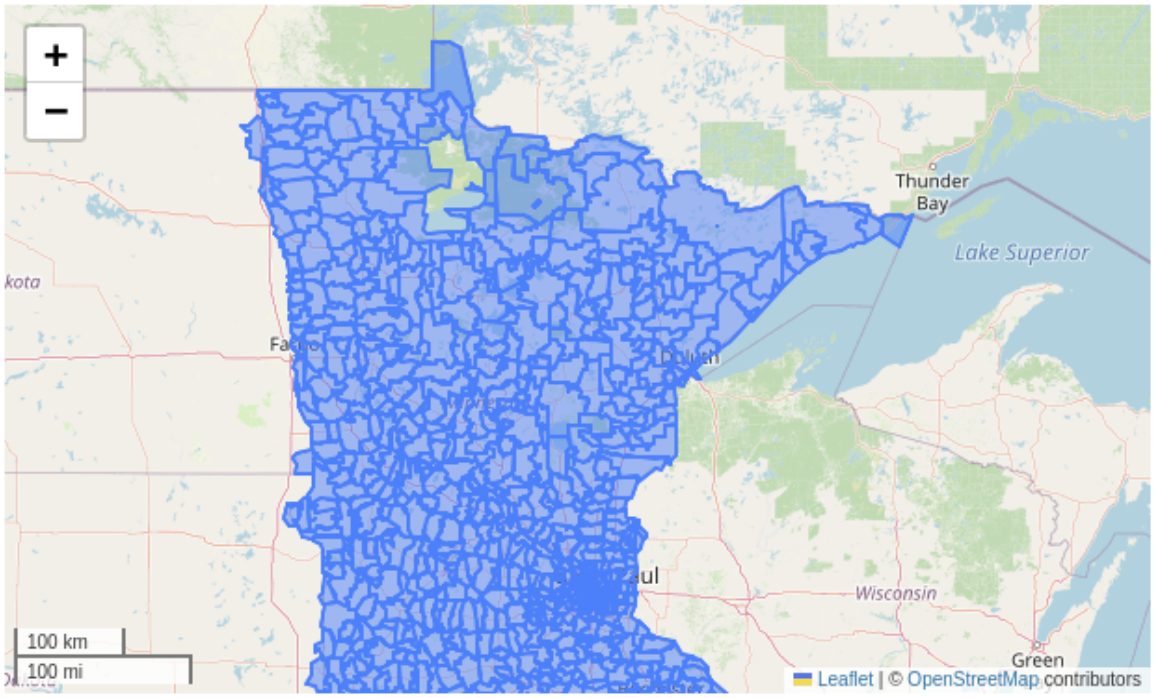
Use the ACS data to show the ZIP codes with homes greater than 0.5M (500K). You will need to do the following:

1. Filter the data to the homes with value > 500 (in the ACS data)
2. Aggregate those and use `VAL_DENOM` as the denomenator to get a percentage
3. Add the calculated field to the filtered GeoDataFrame
4. Run the `GeoDataFrame.explore()` function on the dataset

Your output will look something like this (but with colored ZCTAs indicating the high value homes ZCTA):

```
[33]: df_shp_zip.explore()
```

[33]:



§ Task: Plot demographics and EV

You will now take your EV dataset from the first part and analyze it with the ACS data.

You will find that GeoPandas works just like Pandas in allowing for operations on DataFrames.

You will take the EV data and merge it with eht ACS data, and make the following plots:

1. plot (using GeoPandas `plot()`) `MEDIANHHI` using the ZCTA
2. plot (using GeoPandas `plot()`) `HOMEOWNPCT` using the ZCTA
3. create a correlation matrix of `MEDIANHHI` and `ev_count` for all ZIP codes
4. plot an interactive plot (using `explore()`) of the correlation; to do this you will need to find the correlation for all ZIP codes then merge these back into the GeoDataFrame.

(25%) Perform clustering using Gaussian Mixture Model (EM) in SciKit-Learn

In lecture, we learned about GMMs or the Expectation Maximization algorithm.

Again, you should become familiar with the algorithm and implementation in ScikitLearn:

- sklearn.mixture.GaussianMixture

In this part we will explore three variations to clustering:

- **variation 1:** *cluster based on a pre-defined set of features*
- **variation 2:** *cluster based on a high variance features*

- **variation 3:** *cluster based on all features*

Usually in clustering, we will need to set the number of clusters ahead of time. To simplify things, we will only seek three cluster sizes, 5, 9 and 12. These are arbitrary for now and in a bonus opportunity, you will use other methods to find optimal cluster sizes.

§ Task: Use the GMM algorithm to cluster the data with pre-defined features

You will use GMM to cluster, but you will only use the following features:

- `HIGHSCHOOL, SOMECOLLEG, ASSOCIATE, BACHELORS, GRADPROF, R300_399, R400_499, R500_599, R600_699, R700_799, R800_899, R900_999, R1000_1249, R1250_1499, R1500_1999, R2000up, VAL40_69, VAL70_99, VAL100_124, VAL125_149, VAL150_174, VAL175_199, VAL200_249, VAL250_299, VAL300_399, VAL400_499, VAL500_749, VAL750_999, VAL1MIL, MEDIANHHI, AGEUNDER18, AGE18_39, AGE40_64, AGE65UP, LIVEDALONE, MARRKIDS, UNMARRKIDS, FAMNOKIDS, NONFAMILY, POPTOTAL`

You might notice these features seem to be around demographic features: income, homeownership, etc.

Complete the following:

1. Perform the clustering with `n_components=` 5, 9 and 12 (each will be a separate run).
2. **Only for the `n_components=5` cluster**, describe each cluster in real words. Bring attention to which features seem to dominate the cluster.
3. Make an interactive plot using the ZIP codes in each cluster for the `n_components=5` and `n_components=12` clusters. This will require you to get ZIP codes of each cluster, assign them a label, add that label to each of the ZIP codes in the original GeoPandas DataFrame, then execute `plot()`. Amazingly, the function will do all the hard work of coloring those clusters for you as long as the labels are distinct. **NOTE:** the ZIP Code is hiding out in the `GEOG_UNIT` feature, so you will need to match each data point with the cluster it belongs to, then get the `GEOG_UNIT`, assign a label, then plot. There is a little work involved in this.

§ Task: Use the GMM algorithm to cluster the high variance features

Often, we have too many features to manually select as we did in the prior task. Instead, we can perform a simple removal of features with low variance. The idea behind this is that low variance features do not contribute much to the overall shape of the data, so when determining which features would truly make a difference in clustering, they can often be removed with much effect.

- study the `sklearn.feature_selection.VarianceThreshold`

1. Use `VarianceThreshold` on the entire dataset to eleminate features. set `threshold=0.4`
2. Perform GMM as before, this time with just `n_components=5`
3. Make an interactive plot as before, compare this plot with the previous with a 2-3 sentence summary of the differences.

§ Task: Use the GMM algorithm to cluster all features

Usually, the full feature set will not be easily accomplished since the amount of memory resources is rather high.

1. Perform clustering with `n_components=5`.
2. It is possible that the system will crash. If it does, please report this and leave the notebook in the crashed state for this cell.

(20%) Complete the online HW assessment.

Once you are done with the coding part of the assignment, you will need to complete the online assessment for the final **6 (20%) points of your grade** for this assignment.

§ Task: Go to the course Blackboard and complete the assessment.