HW #1 [Data Engineering, Distance Metrics and PCA]

HW #1 [Data Engineering, Distance Metrics and PCA]

HW #1 (Text version)

Points Possible	Due Date	Time Commitment (estimated)
20	Monday, Sep 19 @ 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

- Learn more about data science tools in the wild for practitioners
- Perform basic data engineering
- Manipulate and analyze the data from NHTSA

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

(30%) Learn more about data science tools in the wild for practitioners

Lot's of interesting things are going on in the data science landscape. While we won't be talking about visualization in detail until later in the semester this podcast hits a number of issues on the head, not only with visualization, but with data science writ large and with the ability for large numbers of "data geeks" from all ranges of technical skill and ability can come together and work out data explorations together.

You will listen to the podcast FLOSS Weekly **#677 April 20, 2022**: *Open Source and Data Visualization* featuring an interview with Melody Meckfessel from Observable (https://observablehq.com) who goes into detail about the platform, the purpose and the community that has emerged there.

You can listen to / watch the show from one of the links below (there are others (i.e. Apple Podcasts) if you search):

- (main page) TWiT: FLOSS WEEKLY 677 OPEN SOURCE AND DATA VISUALIZATION
- (mp3 direct) TWiT: MP3 file direct download
- (youtube) YouTube: FLOSS Weekly #677

§ Task: Listen to the podcast / watch the video and write a 3-5 sentence *reaction* to the podcast. State in your own words what you learned, what expanding your knowledge of the topic and what you found *interesting* about the information you received.

§ Task: Go to observablehq.com and find ONE project. With the one project please enumerate the following:

1. please give the title, URL and description of the project on Observable,
2. describe the datasets used in the project (you can just provide 1 sentence summary of the dataset),
3. provide a brief description of the visualizations used (1 sentence),
4. describe why you found this dataset / project interesting (no more than 2 sentences).

(40%) Perform basic data engineering

As data scientists, especially in a small organization, you will be tasked with getting data prepared for analysis. The scope of this task may be broad – you might have to wait for the data, you may need to clean the data or build subsets of it for consumption by yourself (or others).

It will, nonetheless, be a part of your life – and it will consume time and energy.

In this part of the assignment we will be working with a dataset from the US National Highway Transportation Safety Administration (NHTSA) which provides a number of safety and administrative functions to the US national roadways. One of the important functions of this organizations is to collect data about a wide array of safety issues, one of which is data collection of accidents in all US states and territories.

The data, at least in electronic form, goes back to the mid-1970s and as we will see, gets more detailed over the years.

Our focus for this assignment will be on the FARS (Fatal Accident Reporting System) because we are keenly interested in some very specific trends in fatal accidents, which we will learn about shortly.

All of the data in FARS system can be downloaded. For example, you can go here and access any number of years of data, which is *exactly* what we will be doing in an automated way.

IMPORTANT RESOURCES

- you will most certainly need to refer to the FARS Analytical User's Manual to understand some of the codes being used within the data files.

§ Task: (*Data Extraction, Selection and Transformation*)

You will be downloading a large amount of data from FARS in preparation for analysis. For this part, you will create a Jupyter notebook, which will go to the FTP files of NHTSA.

Your Python program / notebook must do the following:

1. fetch to your local file system every 5th year of data starting with 1975 to 2020. That is you will **automate** downloading the .zip files at the static FTP site and store it locally for 1975, 1980, 1985, … 2015 and 2020.

2. Once you have fetched the .zip files locally, you will then unzip them automatically to a folder corresponding to their year. Thus you will have 10 folders "./1975","./1980", … which will contain the contents of their correspoding .zip file from the NHTSA.

3. You will notice each of these folders contain many .csv files. You will create a folder at the same level as your notebook called accident_all_years and you will copy (not move), each of the yearly files (i.e. "1975/accident.csv") to the folder "accident_all_years/1975_accident.csv". The new folder will contain just the accident.csv files for all years downloaded.

You will (minimally) need to study the following Python libraries to complete this task:

- requests
- zipfile
- os

You do not want to overthink this **and use functions** to perform the mundane automation of this assignment.

(30%) Manipulate and analyze the data from NHTSA

Now that we have data, let's analyze it.

There are so many questions that can be asked of this data but something of extreme interest to a lot of people is the impact driving under the influence of alcohol has on road safety. Driving under the influence has been illegal for many decades (100 years ago cars existed, but were not yet as common as today so the rules of drinking and driving had to be made up as the number of drivers increased), but the limits defining "legally" under the influence have changed dramatically since the 1970s. One might consider one simple fact: more people owned and drove cars post WWII and by the 1970s many US households began owning *two* cars, and as suburban sprawl emerged, the distances people were driving also increased. Also realize the legal drinking age was 18 in many US states, until it was federally changed to 21 in the 1970s.

Blood Alcohol Concentration (BAC) has usually been the standard to measure the level of alcohol in the blood if one is stopped and asked to perform an alcohol test (implied consent is nearly universal in the US). The *federal* BAC limit has *dropped 50%* since the 70s from .15 to .08 today, but one state (Utah) has a limit of .05!

We're going to build an analysis to ascertain fatality relationships with driving under the influence and really explore the historical trends since 1975.

You will need to study the data a bit and understand what you're looking at, as the first part of this assignment will have you explore it in depth.

§ Task: (*Descriptive Statistics / Exploratory Data Analysis*)

One of the first questions we might like to ask of the data, is about the number of fatalities, and then understand how many of those fatalities involved a driver who was over the legal limit for alcohol consumption (keeping in mind that limit changes over time).

You will use your data from the first part to answer the following:

1. What is the overall number fatal accidents for the entire dataset period from 1975-2020? (**Note:** the denomenator is the total number of accidents)

2. How many people died over that period? How many *total* people were involved (fatal and non-fatal)?

3. What proportion of accidents occurred between 9pm and 4am (overnight)?

4. What proportion of accidents occurred when the weather was snowy?

5. Build a line graph that shows the total fatalities by year (using the dataset with just every 5 years of data).

6. Make a general statement about what you observe in the line graph.

To be successful you will need to study the following:

- Pandas Dataframe.groupby(), include sum() and count()
- Pandas selection (include Dataframe.loc[])
- Pandas Dataframe.plot()

§ Task: (*Descriptive Statistics / Exploratory Data Analysis*)

Now that we have a feel for the data, let's go ahead and dig a little deeper into analysis at the state level.

On page 45 of the FARS manual you will see state codes. These will be useful soon.

Of course, we would hope that over the years states recognized that there were higher accidents and deaths due to drivers under the influence of alcohol, and thus tried to *do* something about it. Much of what has been done over the past 45 years has been through awareness, police force training, public messaging and stricter laws.

You will need to cozy up with the data once more and answer the following questions. All of your answers will need to be coded in the Jupyter notebook using Python and Pandas:

1. From 1975-2020, what was the *average* (mean) rate of fatal accidents which involved an intoxicated driver? This would be over *all* states.

2. In 1975 which 5 states had the highest rate of fatal accidents involving an intoxicated driver? Which 5 had the least? Please list the states and the rates in a table in the notebook.

3. By 1990, how much had the top and bottom 5 changed (if at all)?

4. What was the average (mean) rate in 2020?

5. Plot a graph with the top and bottom five states, showing just the rate over time (from 1975-2020). You can plot these in two graphs (top 5 and bottom 5 do not have to be in the same graph).

6. What is your interpretation of the trend – pretend you have no knowledge about the changes in law, changes in BAC thresholds or changes in public messaging about DUIs.

To be successful you will need to study the following:

- Pandas Dataframe.groupby().agg()
- Pandas Dataframe.head() and Dataframe.tail()

§ Task: (*Distance Metrics / Exploratory Data Analysis*)

In lecture, we were introduced to a number of distance metrics, and also to normalization and data scaling.

In this last and final part of the assignment, we want to begin to understand how these metrics form the basis of algorithms such as K-Means and others – but we will do it from the ground up and get a feel for the intuition behind clustering, which will be in the next assignment.

What we are interested in doing is creating a similarity matrix so that we might visually see which data are most similar to one another. This makes it much easier to then perform grouping of data that are "near" one another, then by studying the characteristics that make up these groups, we can understand the underlying composition of the emergent clusters.

In this part we will perform an ad hoc similarity analysis, then we will use PCA to determine which features might be the most useful and re-run the similarity analysis with those features.

1. *You will create a subset of the data over all years.*

- include 5000 random rows of data
- restrict data only to the following columns:

'STATE',
'MONTH',
'DAY',
'YEAR',
'HOUR',
'PERSONS',
'MAN_COLL',
'LGT_COND',
'WEATHER',
'SCH_BUS',
'FATALS',
'DAY_WEEK',
'DRUNK_DR',

- you may need to eliminate rows with NaN data to simplify the analysis

2. Scale the data from part (a) such that all values are between 0 and 1.
3. Compute the distance metric of all 5000 values using *Euclidean* distance and build a distance table.
4. Pick 2 random rows from your original 5000 and find the 20 nearest neighbors of each . Use a single sentence to describe the two collections of 20. You can pretend these each represent a cluster. You must show full work in your notebook. You may also want to optionally bring all the original columns back (i.e. they contain years, which might be helpful)

To be successful you may need to study the following:

- SciKitLearn preprocessing.MinMaxScaler
- SciKitLearn DistanceMetric.get_metric('euclidean')
- SciKitLearn DistanceMetric.pairwise()
- *or* SciPy scipy.spatial.distance.cdist
- Pandas Dataframe.apply()

§ Task: (*PCA / Exploratory Data Analysis*)

The final part of this assignment is to explore dimensionality reduction a bit. We talked about principal components analysis (PCA) in the readings and in lectures, but now we're going to apply it to the dataset.

Recall, that what we're after is a reduction in the feature space such that the relevant components are preserved and are capable of representing the majority of the data. Doing so greatly reduces a number of computation concerns, especially when dimensions are large and in reality there may only be a few dominant features.

We will only go into PCA a little so you can get a flavor for the technique, but the curious would be advised to dig deeper, as there are a number of interesting variations that may be useful to you in the future.

In this part, simply use PCA to discover *how many* components are necessary to represent the data.

1. How many components are necessary to capture 90% of the data in the datast from the prior section. Show the plot of the cummulative variance using PCA.explained_variance_ratio_

2. Please list which feature dominates the first component. To do this, you will need to get the feature with the largest value, which corresponds to the column of the

feature you are looking for, requiring a little backtracking.

You may study the following resources (and more):

- Scikit-Learn PCA documentation
- Medium articles by Ruksan Pramoditha
  - Statistical and Mathematical Concepts behind PCA
  - Principal Components Analysis (PCA) with Scikit-Learn