MCIS6273 Data Mining (Prof. Maull) / Fall 2024 / HW1

Points Possible	Due Date	Time Commitment (estimated)
40	Monday November 25 @ 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

- Perfom basic data engineering and visualization in Python using an external set data.
- Perfom basic data analysis in Python your data file.
- Perform basic K-Means clustering of cookie 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

(30%) Perfom basic data engineering and visualization in Python using an external set data.

Like last homework, you will continue your practice of data engineering to prepare data for analysis.

This time, we will get a little more exposure using a real dataset.

For this part, we will be using data from Open Food Facts (OFF), which is a non-profit organization and online platform that provides a comprehensive database of food products from around the world. The database includes information on over 1 million food products from more than 200 countries.

While I would love to analyze the entire dataset, you will know that the full CSV dump of the data is 9GB and over 3.5 million rows of data! Far too much data for us to go through in a short assignent.

In this first part, we will perform some data engineering as per usual, but to spoil the suprise, we will be analyzing the *cookies* in the dataset so this first part is designed to actually work with a random subset that I have provided and made available on Github.

Remember too, that all of your code must be implemented in Jupyter as a notebook – you will be required to turn in a `.ipynb` file.

§ Task: Filter data to a subset for further use. Normally we would like to get the data and work with it directly, but I have had to reduce the size and scope for you. But, if you are at all curious, you will visit https://world.openfoodfacts.org/data and download the 9GB file and play with it. DO NOT do this on the HUB – your account does not have enough disk space and will cause serious problems for you to be able to run other notebooks. Do this on your own computers and know you will need considerable amounts of free RAM (>32GB minimum).

I have created a file `en.openfoodfacts.products.cookies.csv.tar.gz` and in it are ~37k lines of data which are just cookies. This file is on the Github for `hw1`: https://github.com/kmsaumcis/mcis6273_f24_datamining/tree/main/hw0. It is compressed so you will need to grab it an decompress it with `!tar xvzf en.openfoodfacts.products.cookies.csv.tar.gz` in a notebook cellpw. As a side note, this is only about 1% of the data, and you can think about how many cookies are in the world to consider the scope of that number!

Now that you have a useful file, we will want to filter it further so we can restrict it to just the data we are interested in.

Specifically, we want only a subset the data so that we can ignore data that we do not need.

Load your `en.openfoodfacts.products.cookies.csv` and do some filtering as such:

- filter from the following countries only:
  - `'United States', 'France', 'Spain', 'United Kingdom', 'Canada', 'Italy', 'Australia', 'Switzerland', 'Brazil', 'India'`
- filter the data to just the following columns:
  - `'countries_en', 'completeness', 'serving_size', 'energy-kcal_100g', 'fat_100g', 'saturated-fat_100g', 'trans-fat_100g', 'cholesterol_100g', 'carbohydrates_100g', 'sugars_100g', 'fiber_100g', 'proteins_100g', 'sodium_100g', 'vitamin-a_100g', 'vitamin-d_100g', 'vitamin-e_100g', 'vitamin-k_100g', 'vitamin-c_100g', 'vitamin-b1_100g', 'vitamin-b2_100g', 'vitamin-pp_100g', 'vitamin-b6_100g', 'vitamin-b9_100g', 'folates_100g', 'vitamin-b12_100g', 'potassium_100g', 'calcium_100g', 'iron_100g', 'magnesium_100g', 'zinc_100g', 'copper_100g', 'manganese_100g', 'iodine_100g', 'caffeine_100g'`
- filter the data further so that you only include data with `completeness` > 0.60
- finally filter the data so that `energy-kcal_100` > 0.0
- name the final file `cookies.data-filtered.csv`

§ Task: Plot the data

Produce the following 2 plots:

- plot a bar plot of the frequency counts of cookies by country (use the `countries_en` column); study `DataFrame.value_counts()` to understand how to do this
- bin the `completeness` column into 5 bins using `pandas.cut`, plot the frequency of the completeness bins using a bar plot

To do these plots please study `DataFrame.plot()`. This will provide all you will need to do what is expected. You will need to only use the data in the `countries_en` column.

(30%) Perfom basic data analysis in Python your data file.

Now that we have some sample data (cookies from the countries of interest) we are going to do a little more data normalization to do analysis.

If you look at the data, you might notice that most columns have numeric data, but one of the most useful columns `serving_size` has some very poor quality.

We will produced one more file which will be the **final** file and then we will do some analysis thereafter.

§ Task: Clean up the `serving_size` column and normalize it to numeric data only

If you notice there is some variation in the data entry for `serving_size` … sometimes it is `'10g'`, others `'10 g'` and others are EU decimals like `'2,5g'` using commas and others using periods like `'2.5g'`. There are some serious issues that need to be addressed since the value should just be a number. We won't be able to fix all variations, but many can be addressed with a bit of normalization.

You will need to normalize this so that it is just a floating point number.

To do this you will need to study `Series.str.match` and `Series.str.replace`. To help you, the regular expression `r"\d+[\W,.]?\d*?g$"` is sufficient for this part. If this regular expression fails, you can ignore the data in that cell, and you may need to do a find `dropna()` based on that column to obtain your final results. **NOTE:** your data will be reduced to a much smaller number of data points after `dropna()` on the column, but you should have between 1000-2000 data points left, so if you are in this ballpark, you are in good shape!

Don't forget once done, you will need to use `DataFrame.astype(float)` to coerce the data to a number.

- save your final DataFrame into the final file called `cookies.data-filtered-final.csv`

§ Task: Perform the descriptive statistics on the data.

You will need to study `pandas.DataFrame.describe`

1. What is the mean and median `serving_size` over all cookies?
2. What is the standard deviation of `cholesterol_100g`?
3. Which are the top sweetest cookies normalized by `serving_size`. **HINT:** take the `sugars_100g` divide by 100 and multiple by `serving_size` and sort accordingly. Make sure to show the DataFrame for this.
4. For all French cookies in your final data, what is the average `sugars_100g`? **HINT:** use the `countries_en` column.
5. How does this compare to the United States?

(40%) Perform basic K-Means clustering of cookie data.

One of the most robust clustering methods we can use in unsupervised learning is K-Means.

The K-Means algorithm is a partitioning algorithm which clusters data into k groups, and like the unsupervised algorithms in our toolkit, does not require any labeled data. We can often use K-Means as a starting point for uncovering such labels (with some care).

K-Means is summarized below. The algorithm:

- clusters data into k groups with equal variance
- minimizes "inertia" (within-cluster sum-of-squares)
- requires number of clusters to be specified
- scales well for large datasets, used in various fields
- divides samples into disjoint clusters, each described by a centroid (mean)

You will need to remember that:

- by minimizing inertia, we end up with a measure for the internal coherence of clusters
- centroids are not necessarily data points themselves
- one of the drawbacks, is that it may have trouble measuring internal cluster consistency

Study the ScikitLearn documentation *carefully*:

- K-Means Clustering Algorithm

§ Task: Using a cluster size $k = 5$, perform K-Means clustering on the subset of columns provided.

We now have final dataset what we can work with and to do this last part, you will need to study the ScikitLearn K-Means clustering algorithm KMeans Clustering.

Simply provide the code that produces the clusters.

You will need to reduce your date to just the following column attributes **before** you do the clustering:

- `'serving_size', 'energy-kcal_100g', 'fat_100g', 'cholesterol_100g', 'carbohydrates_100g', 'sugars_100g', 'fiber_100g', 'proteins_100g', 'sodium_100g',  'vitamin-c_100g', 'vitamin-b12_100g', 'potassium_100g', 'calcium_100g', 'iron_100g'`

§ Task: Provide a description of the centroids of each of the 5 clusters.

Your answer must include the description of each of the clusters by accessing the `cluster_centers_`, which are the the *centroids* of the cluster. See the documentation in ScikitLearn mentioned above.

The centroids are the *representatives* of each cluster, so we will (for now) let that stand as what type of cookie is in the cluster.

Your answers will be similar to the example given below, but will include mention of the relevant attribu:tes above

> *Cluster #1 has a serving size of 2g, carbohydrate content of 20g and sugar content of 25g*