CS/INFO 5304 Assignment 2: Data Preparation

Credit: 95 points
Grade: 20% of final grade
Submission: Files that need to be submitted for are listed at the end of each question.
Please submit the assignment on Gradescope
Tip: Read the document fully before you begin working on the assignment
Due date: March 9th, 11:59PM

Question 1: Extract, Transform Load (ETL) (35 points)

Often, we are given different datasets that represent the same population, but each dataset contains different information about this population. ETL pipelines can be used to extract each data source, transform the data sources so we can map one dataset onto the other, and then load them into a single data source that we can use for analytics.

In this question, we will create an ETL pipeline to combine two raw datasets from a sampled population, and estimate the value of a parameter that exists in one dataset, but not the other. The datasets we are looking at come from the U.S. Center for Disease Control (CDC). The first dataset is called the Behavioral Risk Factor Surveillance Survey, or **BRFSS**, and the second dataset is called the National Health Interview Survey, or **NHIS**. Within the question you will be asked to perform the following data analysis:

- Extract the raw datasets into a spark SQL **DataFrame**
- Read the guidebooks for each of these datasets to understand how to **map** similar features to each other
- Perform an exact **join** on these features, and approximate *a disease prevalence* statistic within the US population

For this question, you are **required to use Spark** and will heavily use the **Spark SQL** API. You may use any other Python functions/objects when performing map/reduce on sections of the DataFrame. We recommend **reading this entire section** before beginning coding, so you can get an idea about the direction of the problem.

Step 1: Load data (4 points)

Download data from Google Drive(a smaller dataset than the real one). There should be two data files, called **brfss_input.json** and **nhis_input.csv**. We have also **provided starter code**

for you, in **p1.py**. Complete the function called `create_dataframe` that takes in a path to a file, the type of file to upload (e.g. “csv”), a spark session, and returns the spark DataFrame for that file.

Step 2: Make test example (6 points)

When working with large datasets, it is often helpful to create a small subset as test example that can be used to validate whether our code works, within a short runtime.

Analyze the columns of the BRFSS and NHIS data you downloaded, and identify what columns represent similar information (for example, is there a column in BRFSS, and a respective column in NHIS, that represent a person’s age?). To do this, you will need the codebooks for each dataset, which can be found here:

- 2017 BRFSS codebook
- 2017 Sample Adult File NHIS codebook (Here is a copy on Drive in case you have any issues.)

After analyzing the columns, prepare **three files** that can be used as a test case:

- Download the 5-row BRFSS “dummy” dataset and the 5-row NHIS “dummy” dataset from Google Drive. They are named test_xxx.
- Manually create a *joined_test.csv* file that represents the expected output of your program with the above dummy input. Only exact matches should be kept, and all **null’s** should be dropped. This file should essentially have:
    a. All the BRFSS columns
    b. The column(s) of the NHIS dataset that are not initially within the BRFSS dataset

  You could use multiple columns to join. Do read the codebook to know what does each value represent.

To map the values of NHIS data onto the BRFSS data, for example, **IMPRACE** = 4 = American Indian/Alaskan Native, non-Hispanic; **MRACBPI2** = 3 = Indian (American) (includes Eskimo, Aleut); Map them to same value before joining.
Another example, they both have a column for age. If I had a three row BRFSS dataset with one column representing age, and a two row NHIS dataset with two columns representing age and gender, the expected joined DataFrame would look like the following:

BRFSS Age Column		NHIS Age Column	NHIS Gender		Joined BRFSS Age	Created BRFSS Gender
18-24	+	63	1	=	18-24	2
85+		20	2		60-64	1
60-64		-	-		-	-

Step 3: Map data (12 points)

Next, based on the practice in step 2, you will create a function that maps the NHIS data onto the BRFSS data. In other words, if you have columns in each dataset that represent similar information, but with potentially different column names and values, we want you to write a function that transforms the matching columns within the NHIS DataFrame to resemble the BRFSS DataFrame.

Please complete the function `transform_nhis_data(nhis_df)` that inputs the NHIS DataFrame, and returns a DataFrame where matching NHIS/BRFSS columns have been mapped to BRFSS formatting.

Step 4: Join data (3 points)

Now that you have transformed columns in NHIS, we can actually join the two dataframes together into one entire dataset. Please join together your two Spark DataFrames in function , such that only **exact matches** remain. (Hint: The end number of columns should be the |# of BRFSS columns| **+1**). Please also **drop any rows containing null values** (this is not necessarily correct to do in practice, but since we're focused on ETL, we'll save missing data issues to the next problem).
To clarify, you would have mapped a new column onto your BRFSS dataset. This column represents the prevalence of **a specific disease** throughout the U.S. (look at the codebook to find out what disease!).

You can test and compare with your small data created in Step 2.

(Hint: there are 3 columns in both NHIS and BRFSS representing the same demographic features. Join using all of them and assume each combination is unique while joining, i.e. there is **no duplication**. Is the no duplication assumption correct? Do not need to write answer for this, but **think about this may be helpful for answering part 5.**)

Step 5: Report prevalence (10 points)

Based on the joined data, report the prevalence of this disease by:

- Race and ethnic background

- Gender
- BRFSS categorical age

Complete function `report_summary_stats(joined_df)` that finds these summary statistics. The function should input your joined DataFrame, and does not need to have a return value (you could just print the stats and record them):

In your write-up:

- record the found prevalence you calculated for each category
- research what the **actual prevalence** is
- write a short paragraph **comparing** your found prevalence within the joined dataset to the actual prevalence, by gender, race/ethnic background, and age
- assess how you might **improve** the prevalence you calculated.
    - **Note:** Does each row in the BRFSS dataset correspond to a single person in the U.S.? Check-out the BRFSS documentation.

Turning in:

For this question, please turn in the following:

- Your code, in a file called **p1.py**
- The ***joined_test.csv*** of your created test case in "csv" format
- Your **p1_write_up.pdf** including the answer to step 5 as well as any documentation to run your code.
- Please only submit the above files and do not zip your files

Note: You are ***not*** submitting the joined DataFrame for the actual BRFSS/NHIS data, just reporting the prevalence statistics.

We should be able to run your p1.py file, **from the command line** using the command:

```
spark-submit p1.py /path/to/brfss.json /path/to/nhis.csv -o path_to_output
```

The arguments are as follows:

- `/path/to/brfss.json`: The path to an inputted BRFSS file
- `/path/to/nhis.csv`: The path to an inputted NHIS file
- `path_to_output`: Save the resulting joined DataFrame.

Please document any specific information we need to run your code/understand your outputs in your written report. If you use a different coding language, document how to run your code.

Question 2: Dealing with messy and missing data (40 points)

In this question, we will perform data cleaning and data filling with human-generated sensor data. The intention is to let you practice exploratory data analysis using real-world data, making sense of the data while cleaning it. The form of the submission is a report containing your results and your reasoning.

We will be using a modified version of the Extrasensory dataset from UCSD. Here is the link to it. In short, this dataset is from a study to perform context recognition with smartphone and smartwatch sensors. Filling in data often requires domain expertise, and we choose this dataset because we are certain all of you are domain experts at using a smart phone. :)

In this Question, you can use Pandas or Pandas API on Spark to load the data to Pandas Dataframe or Spark dataframe respectively. All the functions in the pandas library are allowed. Numpy, Scipy.stats, .etc are also allowed.

A JupyterNotebook template to answer this question is also in the data folder.

Understanding data structure

You are provided a `dict` of the modified dataset. There are 60 participants in this dataset. We give you two pickle files: `Extrasensory_individual_data.p and Extrasensory_sensor_data.p`. Please refer here on how to open pickle files.

`Extrasensory_individual_data.p` contains a DataFrame with the participant ID and the following data:

Sensor	**Feature**

demographic	`age`
	`gender`
	`phone_system`

Phone usage	`self_reported_average_screen_time`
	`actual_average_screen_time`
	`hours_of_study`

`Extrasensory_sensor_data` contains a dict with participant ID as the `key` of the provided `dict`. The `value` contains a matrix of ~ 2000 - 6000 recorded samples of 10 features. The number of records corresponds to the number of minutes that the participant underwent the study. It contains the data from the following sensors:

Sensor	Feature

accelerometer	`raw_acc:3d:mean_x`
	`raw_acc:3d:mean_y`
	`raw_acc:3d:mean_z`

location	`location:raw_latitude` `location:raw_longitude`

phone state	`discrete:app_state:is_active`
	`discrete:app_state:is_inactive`
	`discrete:app_state:is_background`
	`discrete:app_state:missing`
	`lf_measurements:battery_level`

As with all studies that are deployed to humans in the wild, there's a lot of missing data. In this dataset, **some of the missing values are recorded as NaN or are left blank. Some other missing values are recorded as -1 for non-negative values.**

Please use one jupyterNotebook(template giving) to write your answers to all the questions of Q2. Please include all the figures and code in the notebook.

Case 1: Actual screen time (15 points)

This study recorded the average screen time every 24 hours for each participant. Some data are missing due to app permissions. Because this is an important reference for analysis later, we need to fill in those missing values.

A) Observe this data or plot a simple plot. How are missing values represented for this feature?

B) Ignore the missing values for now (i.e. mute them or temporarily remove them, so that they don't affect this analysis). **Provide a histogram to illustrate this distribution of** `actual_average_screen_time`**.**
   a) Does it have outliers?If so, how many?
      An outlier is defined as being any point of data that lies over 1.5 IQRs below the first quartile (Q_1) or above the third quartile (Q_3)in a data set.
      High = (Q_3) + 1.5 IQR
      Low = (Q_1) – 1.5 IQR
   b) Is it skewed? If so, is it left skewed or right skewed? What's the skewness?

C) Fix the missing data problem using these methods: 1) filling with median, 2) filling with mean, 3) filling with a random value within range of your choice. Reproduce the distribution figure for this feature filled by all three methods. Overlay a figure with the original distribution and the 3 new distributions. How did you choose the random value from method 3)? What do the distributions look like after you implement the three filling methods? (Compare them)

D) We could compare your filled distribution with the real distribution. Some **Research** shows that American adults have an average screen time of 3.85 hours per day, with a standard deviation of 1.25 hours. You could generate this distribution by:

```
np.random.normal(mean, std, number of samples)
```

Perform a t-test between the three distributions you filled in part C and this population distribution. Report the three p-values. Which one of the filling methods reconstruct this feature to be closest to the **research** distribution? Why do you think this is the case?

Case 2: Perceived average screen time (10 points)

In this study, the participants were asked to self-report how many hours they spend on their phone. Participants could choose not to report if they do not know the number of hours, or if they did not want to.

Let's find out whether these missing values are MAR(missing at random) or MNAR(missing not at random). Said another way, let's see if participants did not report these values because they do not know their screen time, or they did not report because they are self-conscious that they might have spent too much time.

Let's see if there's a correlation between the `actual_average_screen_time` and `perceived_average_screen_time`.

recap:
`actual_average_screen_time` is the screen time recorded by phone and can be missing due to permission or technical reason
`perceived_average_screen_time` is the screen time recorded by the user itself and can be missing or wrongly recorded for various reasons.b

A) Temporarily ignore the missing values. Plot a histogram of the perceived screen time.
   a) Does it have outliers?If so, how many?
      An outlier is defined as being any point of data that lies over 1.5 IQRs below the first quartile (Q_1) or above the third quartile (Q_3)in a data set.
      High = (Q_3) + 1.5 IQR
      Low = (Q_1) – 1.5 IQR
   b) Is it skewed? If so, is it left skewed or right skewed? What's the skewness?

B) Let's define an *intense phone user* as someone whose average screen time is at least one standard deviation larger than the mean screen usage time. How many of them are intense phone users? (Note: Do not remove outliers) (Note: think of which data should be using `actual_average_screen_time` or `perceived_average_screen_time`)

C) Create two binomial distributions of A) missing `perceived_average_screen_time` and B) intense phone users. In another word, generate a boolean array for A) and B). Perform a Chi-square test on these two distributions. What is the p-value? Do you think they are correlated? What does this mean? Do you think this feature is MAR or MNAR? (Note and

hint: If the user's actual screen time is missing, you should not count that user as either intensive or non-intensive, you should filter out those users)

Case 3: Location (15 points)

Now let's look at the stream for location data. Each participant has two sets of location data, `location:raw_latitude` and `location:raw_longitude`. They are very similar, so for ease sake we will only look at `location:raw_latitude`.

There are two thoughts on why location data might be missing: 1) app recording errors/technical errors, which are random or 2) some people turn off location services to save battery when the battery level is low. We assume 2) is a consistent behavior. Let's say people configure their phone to turn off location service if the battery is below 20%.

A) (8 points) Identify these people who behave as 2) and report their uuid. Their characteristics should be that their location data are consistently lost when their battery level is below 20%. We need to find these people because if this was a real-world analysis, their location traces give noise to downstream classification.

   Explain how you find these people in the Notebook. For each of these individuals, how many minutes of location data have we lost due to turning-off of location service?

   Note: This is a half open ended question. There's no standard solution for this, your solution can be loose but you need to explain your solution.

Now, we will only proceed with people with uninterrupted location traces. Find subject `F50235E0-DD67-4F2A-B00B-1F31ADA998B9`. We'll start to do some filling for time series here.

We introduce the following naive methods:

- *Forward filling:* Assume the missing value occurs at `tn`, fill in the value using `tn-1`.
- *Backward filling*: Assume the missing value occurs at `tn`, fill in the value using `tn+1`.
- *Linear interpolation:* For each chunk of missing values, perform linear interpolation with the value of one sample before the missing chunk, and one value after the missing chunk. (If you want to implement more sophisticated linear interpolation, note it in your write-up).

B) (7 points) Perform a forward filling, backward filling and linear interpolation on the `location:raw_latitude` trace. Produce a figure with 4 traces overlay on each other (4 traces including the original, unfilled trace and trace produced by your three filling methods). Compare the 4 traces. What do you see? If you were to use this dataset for further analysis, which filling method will you choose?

Turning it in:

- Your write-up, including answers to the above questions and figures to help support your answers. **One Jupyter Notebook file(q2.ipynb) that includes write-up and code and figures is required.**

Question 3: Outlier Detection (10 points)

Your task is to analyze the data and detect the outliers in the dataset. For this activity, let us use the following dataset given below. **You do not need to use Spark for this question.** You can use Numpy, Pandas, etc. Please wrap up your code and output graphs in one **Jupyter Notebook**.

Input Dataset:

- Top 270 Computer Science / Programming Books. High rated book information in the field of computer science and programing
- Download link

Task 1: Univariate Outlier detection (4 points)

Box plots are efficient in identifying the outliers in numerical values. For this dataset, use the IQR method to identify any univariate outliers in this dataset. Plot the outliers in the context of the feature's distribution for all the four **numerical** features (Rating, reviews, number of pages, price). Your output should be the box plots of each feature.

Task 2: Multivariate Outlier detection (6 points)

Lets use the DBSCAN Method for multivariate outlier detection. For this exercise, use the columns [“Price”, “Number_Of_Pages”,“Rating”, “Reviews”, “Type”] to fit DBSCAN.

Your first task is to perform a bivariate analysis on all possible pairs of the above features and identify any outliers. The output should be all the DBSCAN plots and the outlier data row(index&value) for each combination.

Your second task is to look for all combinations of three variables in the above dataset to identify multivariate outliers. The output should again be all the DBSCAN plots(3D) and the outlier data row for all combinations.

Use the following hints for answering this question:

- For categorical variables, you can convert them into numerical features for the analysis.
- Try scaling the data first.
- Tune the eps and min_samples.

Turning in

1) A Jupyter Notebook(q3.ipynb) with all code, output and graphs.
2) Please export your q3.ipynb to a q3.py and upload this q3.py as well. We will need to run a plagiarism check on the code. (do not zip these two files)

Question 4: Data Visualization (10 points)

Your task is to design a visualization that you believe effectively communicates the data and provide a short write-up describing your design. **You can use any data visualization tool you want.**

Start with the dataset weather.csv which contains weather measurements nearest to Cornell Tech every day from 1950 to the present.

Notice that the temperature (Ktemp) is in Kelvin rather than in Fahrenheit. Define a variable that expresses the temperature in Fahrenheit, using the following formula:

Ftemp = (Ktemp − 273.15) * (9/5) + 32

Part A) For every month of the year, plot the average temperature (in Fahrenheit) using a scatter plot or line plot. Your visual should be configurable so that a user can easily look at a specific year's weather curve (use a sliding scale filter). (6 points)

Part B) Based on all of the data, when is the first year where the year's average temperature passes 55 degrees (when will Cornell Tech finally be warm?) (2 points)

Part C) Create a new sheet where you do something creative through data visualization. Express something about the temperature over time(e.g. Look in the cycle of temperature over seasons, etc) using this dataset, or add a new dataset(any available dataset online is fine) and find some correlation with the temperature(e.g. Number of some kind of fish in the ocean, etc, does the number of it go up and down following the temperature trend? Etc.). (2 points)

You will be graded based on the quality of completion. A simple plot of temperature in Celsius will not get you full points.

Note: Your visualization should be interpretable and accompanied by a short write-up (you do not need to write more than a sentence or two). Do not forget to include title, axis labels, or legends as needed!

Turning it in:

1. Export everything to one pdf file q4.pdf and submit it in gradescope. If there's any dynamic visualization, please include the URL to the visualization in the pdf. (consider graders as newspaper readers, do not expect graders to have all the tools installed. For example, if using tableau, do not submit the .twb file, but push the visualization to tableau public and include an url)