HOSPITAL MORTALITY DATA: EDA

INTRODUCTION

This set of exercises will demonstrate how to import, clean, merge, and reshape a dataset.

Our goals for this lab are:

- Import data
- Merge data
- Reshape data

DATA USED IN THIS EXERCISE:

We will use the following datasets in this exercise:

- data/healthcare/ Clinic.csv
- data/healthcare/OutpatientVisit.csv
- data/healthcare/ DiseaseMap.csv

Concept of the data:

- Each time a patient visits a clinic, there is a record in the 'OutpatientVisit' table which has details about the visit, including the patientID, date of the visit, the clinicID, and up to three ICD Codes (Diagnosis codes)
- The Clinic table contains the mapping from ClinicID to Clinic Type
- The DiseaseMap table contains a record for every possible ICD Code, as well as the disease that code corresponds to. For example, there are many different types of cancer. The following ICD Codes all indicate some form of cancer: C9590, C9591, C966, etc…the DiseaseMap table contains all of these disease to code mappings. The OutpatientVisit table includes only the diagnosis code (ICD Code). You will need to merge these tables.

QUESTIONS

1) Start by writing the import statements for pandas and numpy. Also, set up the working directory to point at the folder you have established for this class. Confirm the working directory is what you expect.
2) Import the OutpatientVisit data and perform the standard checks:
    a. First inspect the text file with a text editor
    b. Import the dataset
    c. Check the shape, dtypes, summarize each column, check for missing data
    d. Change visitdate to an actual date variable and check your result
    e. Create a histogram of the visit dates. Interpret the graph, what does that tell you about your data?
    f. Examine this file. Is it 'long', or 'wide' (or both maybe)?
3) We now want to explore the number of visits each patient had. For example, what is the minimum number of visits per patient? What about the max, or the mean?
    a. This is a small amount of code. You can use a .groupby() then a .count() to get a dataframe of the number of visits by patient. After that, you can simply .describe() the dataframe.
    b. How many patients have only one visit? What percent of the total is that?
    c. How many patients have more than 30 visits? What percent of the total is that?
4) Import the clinic codes dataframe. Perform the standard checks, then merge it to the OutpatientVisit data so we can identify the type of clinic each patient visit occurred in.
    a. First inspect the text file with a text editor
    b. Import the dataset
    c. Check the shape, dtypes, summarize each column, check for missing data
    d. Merge the clinic and OutpatientVisit data. Check the shape of the data pre and post merge!
    e. Now create a dataframe with type of clinic as columns, patientID as rows, and number of visits as the values. You can use groupby, count, and then unstack the result. After unstacking, examine your result…it has missing values. These should be replaced with zeros (use fillna), think about why…
    f. What is the mean number of primary care visits per patient?
5) Import the disease codes and perform standard checks, then merge to outpatient visit data.
    a. First inspect the text file with a text editor
    b. Import the dataset
    c. Check the shape, dtypes, summarize each column, check for missing data
    d. Now we want to merge the disease map and OutpatientVisit data. The columns in OutpatientVisit containing ICDCodes are ICD1, ICD2, and ICD3. There is only one column in the disease map table with the ICDCodes…what should you do?

The end goal is to identify which patients have a diagnosis for a given condition. So we will need to somehow merge to all three columns…

e. You have two options. You could perform three merges, but that is not the best way. Instead, 'melt' the outpatientVisit data into a long form, then do a single merge between the outpatientVisit table and the disease map table.

f. After the merge, we want to pivot this 'long' data into a 'wide' dataframe with PatientID as the rows, and each unique condition as the Columns. You can use the pivot_table function for this task.

g. Finally, we want to calculate the proportion of Patients with each disease. Depending on how you did the step above, you may need to convert the dataframe to zeros and ones, (zero means no diagnosis, and one means one or more diagnosis). You can then simply take the mean of each column in the dataframe. Finally, sort the result and tell me what is the most common and least common disease?