D206 Performance Assessment

August 12, 2021

**Part 1: Research Question**

1. Question:

Do patients who get readmitted into the hospital have more pre-existing conditions than patients admitted for a first time?

1. Required Variables:

The data set “medical\_raw\_data.csv” contains 10000 records of patients. The factors that are important for this analysis are the “ReAdmis” column for the readmission data as well as the columns that contained data for pre-existing conditions. All the data that will be measured in this analysis will be using Boolean values to check if a condition is present or not and what effect that has on the patient. All data was changed to be either 1 for yes, or 0 for no and compared to if the patient was readmitted or not (ex. “HighBlood” = yes -> “HighBlood” = 1). Using some simple calculations, you can start seeing how many preexisting conditions increase your chances of getting readmitted to the hospital.

**Part 2: Data-Cleaning Plan**

1. 1. Plan to find anomaly:

First, I plan to save the medical data into a datafile using pandas in python. Next, I was going to isolate the readmission column along with all the pre-existing condition columns. I then need to find the null values and delete them to not skew the data. After clearing out the null values, I’ll then replace all the Yes/No values into 1/0 to make calculations easier and stand out more. The data sets will then be split into 2 tables where one has people who are first entered into the hospital, and the other having the patients who are being readmitted. Principle component analysis will be done on both data tables and will also be charted to see the relations between the principal components and explained variance.

2. Approach for assessing quality of data:

The data I will be assessing will pretty much be entirely in Yes/No format which doesn’t work very well with python, so I switched the data from Yes/No into 1/0 format to do easier addition and also allows for easier graphing of the data for better visual representation.

3. Programming language and libraries used:

I used python since I’m more comfortable using that language and have more experience in that than R. The libraries I needed to use were pandas for database operations, numpy for cumsum operations, and seaborn for graphing. I also used matplotlib.pyplot in order to display and make graphs as well as sklearn.decomposition in order to use principle component analysis.

4. Code:

See attached PDF of final code print out. GitHub link as well.

<https://github.com/helchild/d206/blob/main/D206.ipynb>

**Part 3: Data Cleaning**

1. 1. During my data cleaning phase, I found that every column denoting pre-existing conditions was full of values containing either Yes/No or 1/0. When checking for null values, there was 982 null values in “Overweight” and 984 null values in “Anxiety”. After dropping all null values, there were still 8126 patients within the system.

2. The justification for deleting the null rows was because the other options would be to either get rid of the entire columns or fill the data in. Filling the data in would be a complete guessing game, but if randomly allocated, it could be roughly 50% accurate because there’s only 2 options. I personally don’t like the guessing game since it would add a huge amount of uncertainty in the dataset, so I didn’t do this option. Another option would be to get rid of the entire columns, but I don’t think those would be a good idea since they are important factors in health and health outcomes, so I figured it’d be better to proceed with deleted rows, rather than ditch 2 whole health factors. I also decided to convert all data into 1/0 simply because it will make charting and analysis easier.

3. First step in the data cleaning process, was to check for duplicates in the dataset and see if any duplicates need to be removed. Next, we isolate the columns necessary for analysis into its own datafile consisting of readmission status along with pre-existing conditions. After completing this step, we check for null values and delete those null values. After that, we have all the data we need, now we transcribe the data into 1/0 by swapping all Yes/No for 1/0. The next step is to break the data table into 2, a table that is readmission, and a table that is first time admission. Then for each table, you add up the columns, get a sum, and chart them to get a bell curve along with median, mean, and mode to figure out the patterns between readmission and regular admission.

4. Code: See PDF and GitHub link mentioned above.

5. Copy of cleaned dataset: See attached PDF

6. The limitations that this dataset offers primarily resides on the missing values. Since there are 3 options, delete rows, columns, or fill-in. If I delete the rows, I’m now missing almost 2 thousand patients, which is roughly 20% of the data which could cause a few issues, but still a very good amount for an analysis. If I delete the columns, then the analysis would be missing 2 essential factors in readmission, which is the primary focus of this analysis, so deleting those dependent factors would start altering the data. The final limit of filling in the null values would be that there is no way to accurately figure out what those values would be. You could randomly associate data with it, but then you’re basically just randomly generating data which isn’t typically reliable and adds a lot of uncertainty into the data.

1. 1. List of principal components in data set would be the columns of pre-existing conditions:
   * HighBlood – PC1
   * Stroke – PC2
   * Overweight – PC3
   * Arthritis – PC4
   * Diabetes – PC5
   * Hyperlipidemia – PC6
   * BackPain – PC7
   * Anxiety – PC8
   * Allergic\_rhinitis – PC9
   * Reflux\_esophagitis – PC10
   * Asthma – PC11

2. I used intuition of the data in order to choose my components. All the factors listed are important in discussing a patient’s health and chance of readmission. The only factor that wasn’t used in the principal component analysis was the readmission column since that was used simply to identify the 2 groups (readmitted and first time) to perform the analysis on.

3. An organization could use this to see which pre-existing conditions tend to lead to higher chances of readmittance into the hospital. They can assess the values of each component to focus on what to target with their patients and either prescribe medicines, perform procedures, or do some planning with the patient to help avoid them being readmitted into the hospital. The analysis of pre-existing conditions did show in the end that there seemed to be no overall effect on how many pre-existing conditions a patient being readmitted would have. The PCA analysis did show that certain pre-existing conditions had more of an effect on the outcome when in tandem of other pre-existing conditions, but the overall number of pre-existing conditions was 4 for both camps.

**Part 4: Supporting Documents**

1. Panopto recording: <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=8f0d3d93-cf00-4591-bc29-ad8c015552c4>
2. References for code:

Pandas documentation: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>

Lab notes and code:

Larose, C. D., & Larose, D. T. (2019). *Data science using Python and R.* John Wiley & Sons. ISBN: 978-1-119-52684-1

Other code references from stack overflow:

<https://stackoverflow.com/questions/40901770/is-there-a-simple-way-to-change-a-column-of-yes-no-to-1-0-in-a-pandas-dataframe> , 3novak

<https://stackoverflow.com/questions/47182183/pandas-chained-assignment-warning-exception-handling> , HeXor

1. Other sources:

<https://www.youtube.com/watch?v=FgakZw6K1QQ> , Principal Component Analysis, StatQuest with Josh Starmer

<https://www.youtube.com/watch?v=TJdH6rPA-TI> , Principal Component Analysis, Computerphile