**Medical Dataset: Cleaning, Wrangling and Imputation Techniques**

**for Primary Component Analysis**

Bratislav Petkovic

College of Information Technology, WGU

Dr. Keiona Middleton

November 17th, 2022

Upon careful examination of the datasets provided, the assessment requirements, and data science applications in medical and biological fields, the medical dataset was chosen for the project. While getting to know the different features of the data provided, there was particular intrigue around patient responses to Items1-9 and if any existing patterns would give the hospital more insight into what is most important and to which patients is it important. Thus the research question was constructed as follows:

How can driving indicators of the patients’

responses to Items 1-9 best be identified using clustering techniques?

Because the project required Primary Component Analysis, an unsupervised learning technique, clustering was found to be applicable. Before proceeding into data cleaning it is crucial to introduce the data, which is best grouped by the below categories

**Table 1.**

*Breakdown of data by category*

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Category | Data Type(s) | Examples |
| CaseOrder | Unique case ID, 1 – 10,000 | Integer, Primary Key | 1 |
| Customer\_id | Unique Patient ID, 1 – 10,000 | String | C412403 |
| Interaction, UID | Unique Patient transaction ID | String | 8cd49b13-f45a-4b47-a2bd-173ffa932c2f |
| City, State, County, Zip, Lat, Lng, Area, Population, TimeZone | Patient Location Metadata | String, String, String, Integer, Float, Float, Category, Integer, Category | West Point, VA |
| Job, Employment, Income | Patient Professional Metadata | String, Category, Float | Chief Executive Officer |
| Children, Age, Education, Marital, Gender | Patient Personal Information | Int, Int, Category ,Category, Category | ‘Some College, 1 or More Years, No Degree’ |
| ReAdmis, Doc\_visits, Full\_meals\_eaten, VitD\_supp, Initial\_admin, Complication\_risk, Services, Initial\_days, TotalCharge, Additional\_charges | Patient Hospital Stay Metadata | Category, Int, Int, Int, Category, Category, Category, Category, Float, Float, Float | 17.80233049 ng/mL |
| Soft\_drink, HighBlood, Stroke, Overweight, Arthritis, Diabetes, Hyperlipidemia, Backpain, Anxiety, Allertgic\_rhinitis, Reflux\_esophagitis, Asthma | Patient Lifestyle and Conditions | Category (all) | Yes |
| Items 1 through 9 | Survey Responses ( research question target variable) | Int ( 1-8) | 4 |

Python was the programming language of choice because of its proven reputation in the data science community and the vast supporting packages that it provides for all steps of the data science life cycle. Jupyter made a great story telling tool due to its ease of use and accessibility.

Evidently, many of the columns were strings and categories which are incompatible with most Machine Learning Algorithms. Therefore, to prepare the data for modeling, all non-numeric features needed to be converted into numeric ones. To achieve this, 3 different encodings were used, each from Python’s sklearn preprocessing library (Scikit Learning Developers, n.d.) . One hot encoding was applied to non-ordinal categorical columns which had a handful of variables such as Gender, Marital, Area, Employment, Services and Initial\_admin. Label Encoding was utilized for non-ordinal categorical data which had too many columns for one hot encoding : Job, State, Country, Timezone, Customer\_id, Interaction, UID and City. Lastly, Ordinal encoding was leveraged for data where a particular order could be established : all patient lifestyle and conditions, Education, ReAdmis and Complication Risk. Additionally, I decided to round Initial\_days and Income to integers and TotalCharge and Additional\_charges to cents to increase readability of data. It is important to note that the newly wrangled data had 11 more columns than the original dataset.

While the data did not exhibit any duplicates, it did unfortunately exhibit missing values in Children (.25 missing) , Age (.24) , Income ( .25), Soft\_drink (.25), Overweight (.1), Anxiety (.1) and Initial\_days (.1). Using the missingno package (ResidentMario, n.d. ), missingness of the data was analyzed. A correlation matrix was computed between the missingness of every variable and the findings stated that the data was Missing Completely at Random (MCAR) which means that there is no pattern to the missing data. Coupled with the quantity of data missing, it was decided to impute the missing records instead of removing the rows or columns. Because of its accuracy, the K-nearest Neighbors Imputation technique was used from fancyimpute (Iskandr, n.d.). KNN Imputation technique usually proves more accurate according to (Bard, 2019) than imputations via mean, median and mode as it relies on more sophisticated techniques such as feature similarity and weighted averages. It is important to note that KNN imputed data needs to be rounded, in the case of discrete and/or categorical data such as with Soft\_drink and Children. KNN proved to be a good technique as it did not drastically change the mean of columns with previously missing data. Missing categorical data did not have multiple categories which is a KNN limitation. However, despite the small amount of missing data, imputation did take a significant amount of time.

**Figure 1.**

*Correlation Matrix. Darker blue signifies more correlation*

Chart, scatter chart

Description automatically generated

Before embarking on the more complex dimensionality reduction technique of PCA, a correlation matrix was constructed with the wrangled, imputed dataset (Figure 1) and it exposed that Zip and Lng were 90% correlated. Therefore, I decided to remove zip from the dataset because it was not adding any additional valuable information (variance) . Additionally, as outliers gravely impact PCA, outlier detection was performed using box plot distribution graphs and minimum and maximum value examinations of all variables. No outliers were present in the data – all values were found to be feasible and valid.

PCA can severely underperform if it contains categorical data and if the data is not normalized. All previously encoded columns, despite now being in numeric form, were forked into another data frame along with Items 1-9. The remaining columns which all contained numeric continuous variables were set aside and normalized. During the creation of the model it was found that sklearn’s StandardScaler was not any more accurate than applying the normalization formula to the dataset. The PCA model, constructed using the PCA object from sklearn’s decomposition package, was given 13 – dataset column count – for n\_components – the number of primary components parameter. PCA works by projecting records in a vector space containing n\_components vectors, and each record has a particular magnitude and direction associated with each vector. Then, using Lagrange multipliers and optimization techniques, a primary component is constructed in such a way as to preserve as much information in the dataset. A primary component can be described by its loadings, which is the set of all of the dataset’s columns each defined by a particular magnitude and direction (Figure 2).

By examining the loadings, it is evident that PC1’s highest contributor was TotalCharge (0.69) and VitD\_levels (0.52) while PC3’s highest contributors were location attributes (Lat, Lng, Population). To further

**Figure 2.**

*PCA Model Loadings*

Table

Description automatically generated

expose patterns in the data Pipeline was used and a scatterplot was created, matching PC1 vs PC2 and grouping by categories previously left out. Figures 3 and 4 show the valuable findings. It would be possible to construct a Naïve Bayes, KNN or SVM clustering algorithm to group whether or not a patient is more likely to be readmitted to the hospital or if they suffer from high blood pressure. Important to note that HighBlood has a weaker pattern compared to ReAdmis. Coupled with loadings, it can be said that Vitamin D levels and total charge, as well as age and addional\_charges were driving indicators for patient readmission. The hospital greatly benefits from PCA findings because they gain patient insight – the patterns associated with a patient being readmitted are better understood which in turn can drive administration decision making.

**Figure 3.**

*PC1 vs PC2 by patient readmission*

Chart, scatter chart

Description automatically generated

**Figure 4.**

*PC1 vs PC2 by high blood pressure*

Chart, scatter chart

Description automatically generated

**Figure 5.**

*PCs by Eigenvalue*

Chart, line chart

Description automatically generated

**Figure 6.**

*PCs by Cumulative Variance Explained*

Chart, line chart

Description automatically generated

In selecting the extracted features, Figures 5 and 6 demonstrate the two techniques used to reduce dimensionality – Kaiser Rule and variance contribution, respectively. The Kaiser Rule states that all primary components having an eigenvalue greater than or equal to 1 should be preserved which would be the first 6 PCs in the case above (Figure 5). In the project, the Kaiser Rule was preferred but it is important to recognize the variance contribution technique which selects the first n features that cumulatively make up 90% of the variance in this study, the technique would choose the first 9 PCs (Figure 6).

**References**

Badr, W. (2019, Janaury 5). *6 different ways to compensate for missing data (data imputation with examples).*

Towards Data Science. [https://towardsdatascience.com/6-different-ways-to-compensate-for-missing- values-data-imputation-with-examples-6022d9ca0779](https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-   values-data-imputation-with-examples-6022d9ca0779)

Iskandr. (n.d.). *ISKANDR/fancyimpute* [Multivariate imputation and matrix completion

algorithms implemented in Python]. GitHub.

<https://github.com/iskandr/fancyimpute>

Jaadi, Z., & Powers, J. (2022, September 26). *A step-by-step explanation of principal component*

*analysis (PCA).* Built In.

<https://builtin.com/data-science/step-step-explanation-principal-component-analysis>

ResidentMario. (n.d.). *Residentmario/Missingno* [Missing data visualization module for python].

GitHub <https://github.com/ResidentMario/missingno>

Scikit Learn Developers. (n.d.). *6.3. Preprocessing Data.*

<https://scikit-learn.org/stable/modules/preprocessing.html>