Cleaning Medical Data

Curtis Hammons

This paper details the process of cleaning a medical dataset for use by a data science team. We will start with a potential question to be answered by the data science team and going over the present data. Then we will clean the data by finding suitable replacement for missing values and removing outliers. Finally, we will perform Principle Component Analysis (PCA) on the cleaned data set.

The full code used in this project can be found in the included notebook data-cleaning.ipynb

Question: What are the primary indicators of patient readmission?

Hospital readmission has become such an issue in recent years that Centers for Medicare and Medicaid Services (CMS) has begun to impose fines on hospitals excessive readmissions. Despite this, the percentage of hospitals fined for readmissions has increased every year. Our goal is to determine if there is a pattern among patients who are readmitted so that we can detect early on if a patient has a high readmission risk and take steps to combat it.

The Dataset

Our data consist of 10,000 patient records with 50 features. The features can be sub-divided into the following categories:

1. Hospital system variables
2. Patient Demographics
3. Patient Health
4. Visit details
5. Patient Survey results

Hostpital system variables

- **CaseOrder**: Variable to preserve original order of the data file
- **Customer_id**: ID unique to each patient
- **Interaction**: ID of each patient interaction
- **UID**: transaction ID

Patient Demographics

These features give us basic demographical imformation about the patient. They are as follows:

- **City**: City of residence as listed on the billing statement
- **State**: Patient state of residence as listed on the billing statement
- **County**: Patient county of residence as listed on the billing statement
- **Zip**: Patient zip code of residence as listed on the billing statement
- **Lat, Lng**: GPS coordinates of patient residence as listed on the billing statement
- **Population**: Population within a mile radius of patient, based on census data
- **Area**: Area type (rural, urban, suburban), based on unofficial census data
- **TimeZone**: Time zone of patient residence based on patient's sign-up information
- **Job**: Job of the patient (or primary insurance holder) as reported in the admissions information
- **Children**: Number of children in the patient's household as reported in the admissions information
- **Age**: Age of the patient as reported in admissions information
- **Education**: Highest earned degree of patient as reported in admissions information
- **Employment**: Employment status of patient as reported in admissions information
- **Income**: Annual income of the patient (or primary insurance holder) as reported at time of admission
- **Marital**: Marital status of the patient (or primary insurance holder) as reported on admission information
- **Gender**: Customer self-identification as male, female, or nonbinary

Patient Health

- **VitD_levels**: The patient's vitamin D levels as measured in ng/mL
- **Full_meals_eaten**: Number of full meals the patient ate while hospitalized (partial meals count as 0, and some patients had more than three meals in a day if requested)
- **VitD_supp**: The number of times that vitamin D supplements were administered to the patient
- **Soft_drink**: Whether the patient habitually drinks three or more sodas in a day (yes, no)
- **Complication_risk**: Level of complication risk for the patient as assessed by a primary patient assessment (high, medium, low)

- **HighBlood**: Whether the patient has high blood pressure (yes, no)
- **Stroke**: Whether the patient has had a stroke (yes, no)
- **Overweight**: Whether the patient is considered overweight based on age, gender, and height (yes, no)
- **Arthritis**: Whether the patient has arthritis (yes, no)
- **Diabetes**: Whether the patient has diabetes (yes, no)
- **Hyperlipidemia**z: Whether the patient has hyperlipidemia (yes, no)
- **BackPain**: Whether the patient has chronic back pain (yes, no) Anxiety: Whether the patient has an anxiety disorder (yes, no)
- **Allergic_rhinitis**: Whether the patient has allergic rhinitis (yes, no)
- **Reflux_esophagitis**: Whether the patient has reflux esophagitis (yes, no)
- **Asthma**: Whether the patient has asthma (yes, no)

Visit Details

- **Doc_visits**: Number of times the primary physician visited the patient during the initial hospitalization
- **ReAdmis**: Whether the patient was readmitted within a month of release or not (yes, no).
- **Initial_admin**: The means by which the patient was admitted into the hospital initially (emergency admission, elective admission, observation)

Patient Survey Results

Patients are given an eight queston survey in which they are asked to rate the importance of several factors/surfaces on a scale of 1 to 8 (1 = most important, 8 = least important)

- **Item1**: Timely Admission
- **Item2**: Timely treatment
- **Item3**: Timely visits
- **Item4**: Reliability
- **Item5**: Options
- **Item6**: Hours of treatment
- **Item7**: Courteous staff
- **Item8**: Evidence of active listening from doctor

Cleaning The Data

We'll be using python in JupyterLab for this project. Libraries included: pandas and numpy for basic data handling and math operations, scipy for calculating zscores, sklearn for Principle Component Analysis (PCA), and matplotlib for generating plots.

First we import the raw csv file into a pandas DataFrame called `medical_raw` and run `medical_raw.info()` to see what we're dealing with.

The clean data can be found in `medical_clean.csv`.

Inconsistent column names

The first thing we notice is the columns of the dataset follow the naming convention of using underscores between words except for three outliers: TotalCharge, BackPain, HighBlood, and ReAdmis, which simply capitalizes the second word we'll rename the columns as such:

- CaseOrder: Case_order
- TotalCharge: Total_charge
- BackPain: Back_pain
- HighBlood: High_blood
- ReAdmis: Re_admis

Missing Values

Upon initial inspection of the dataset we find that the following columns contain null values:

- Children
- Age
- Income
- soft_drink
- Overweight
- Anxiety
- Initial_days

Before we continue we shall find suitable replacements for these values using the the `fillna()` function of DataFrames.

Methodology

Children, Age, Income, and Initial_days are numeric features, so we will fill the null values with the mean of the non-null values. The line of code that does this looks like:

```
medical_raw['Children'].fillna(round(medical_raw['Children'].mean()), inplace=True)
```

Soft_drink, Overweight, and Anxiety are boolean features. They are either "Yes" and "No" or "1" and "0" (we'll fix this inconsistency as well, replacing "0" and "1" with "Yes" and "No"). These will be replaced with the mode of the non-null values. Like this:

```
medical_raw['Overweight'].fillna(medical_raw['Overweight'].mode()[0], inplace=True)
```

We now have a fully non-null dataset!

Limitations

While replacing the missing values with the mean does not disrupt the consistency of the column as a whole it can provide misleading data when a specific observation is looked at. This is still preferable to a null value.

Outliers

Methodology

In order to find outliers we will be calculating *zscores*, or *zvalues* for the numeric fields. Once we have standardized z-scores we'll use the standardized values to easily remove the outliers.

The numeric fields we'll be getting zscores for are:

- Population
- Children
- Age
- Income
- Doc_visits
- Full_meals_eaten
- Initial_days
- Total_charge
- Additional Charges
- Item1
- Item2
- Item3
- Item4
- Item5
- Item6
- Item7
- Item8

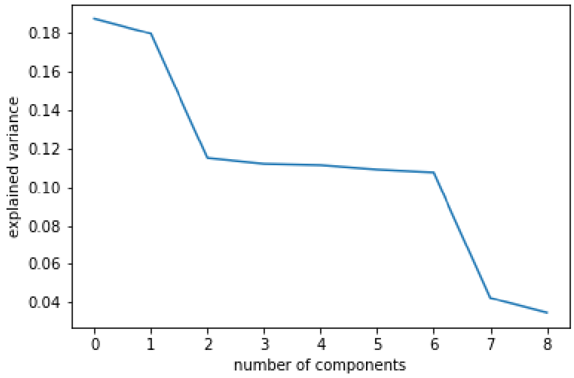
We drop all entries that have outliers (below -3 and above 3 zscore). This brings our total number of entries down to 8,960 from 10,000. Just over 10% data loss.

Limitations

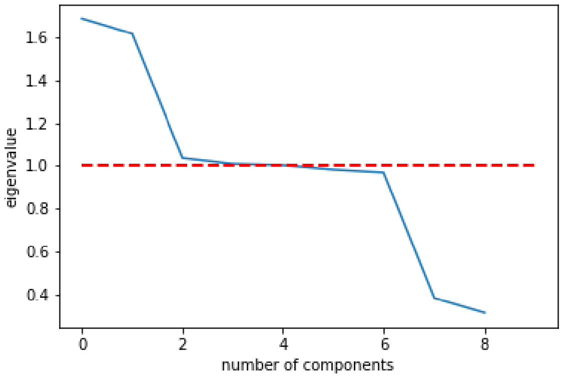
Losing over 10% of the dataset is not ideal. We may decide that a certain amount of missing values is workable or that we only remove an observation if it is missing multiple values.

Principle Component Analysis

We perform PCA on the cleaned dataset and graph the explained variance.



We'll also look at the eigenvalues.



We're interested in components 1-5, since those are the ones with an eigenvalue greater than 1. Now let's look at the loadings for each of the components.

-	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Population	0.012890	-0.029496	0.566470	-0.202993	-0.464120	-0.009102	0.649096	-0.004586	0.005747
Children	0.006824	-0.009582	0.314918	0.678089	0.221842	0.616316	0.103571	-0.010360	0.031454
Age	0.171370	0.685131	0.015768	0.009590	-0.024775	-0.018818	-0.007192	-0.691725	0.146230
Income	-0.009770	0.005838	0.342582	-0.012432	0.770602	-0.479587	0.241783	-0.004712	0.006415
Doc_visits	-0.007036	0.010192	0.303928	-0.681197	0.275902	0.540884	-0.272890	-0.016777	-0.004034
Full_meals_eaten	-0.031205	0.039583	-0.606501	-0.183863	0.252871	0.311409	0.659363	-0.010330	-0.004949
Initial_days	0.682545	-0.183392	-0.011978	0.006127	0.023240	0.011010	0.012222	-0.159547	-0.688515
Total_charge	0.687647	-0.160156	-0.032895	-0.028656	0.016700	-0.001301	0.005732	0.155665	0.689234
Additional_charges	0.174856	0.684577	0.029752	0.006963	-0.007265	0.007744	0.005106	0.686506	-0.168571

Let's break down the weight for each of the PCs.

- **PC1**: Significant weight is given to Initial_days and Total_charge, almost .7 for each. The rest of the values are hardly worth mentioning.
- **PC2**: Age and Additional_charges are given similar weight that Initial_days and Total_charge were in PC1.
- **PC3**: PC3 is a bit more balanced than the previous 2, giving similar weight to Income, Doc_visits, and Children at around 0.3. Population is given a larger weight at 0.5. Full_meals_eaten sits at -0.68, the heaviest.
- **PC4**: This one has Children and Doc_visits at opposite extremes, 0.67 and -0.68 respectively. The rest of the variables are pretty insignificance, with only Population being the only one weighted more than 0.2
- **PC5**: Income is the heaviest at 0.77, but the rest of the values are fairly balanced.

Using PCA

We can use these components to replace the original data if we wish. A csv file called `medical_reduced.csv` can be found alongside the `medical_cleaned.csv` file.

Conclusion

We have cleaned the dataset of any disruptive anomalies and prepared it for use in a data analysis project. We have also performed Principle Component Analysis and provided a copy of the reduced data for the data analytics team to use if they wish.