Data Cleaning – D206

Western Governor's University

Oviya Selvaraj

Part I: Research Question

A1. My research question is ‘What are the key demographic, health-related, lifestyle and service-related factors influencing patient readmission? ’

A2. The dataset contains 52 columns with information related to patients and their readmission.

- **CaseOrder (quantitative, datatype - integer):** This column functions as an index. Example, 1.
- **Customer_id (qualitative, datatype- string):** This column contains unique value to identify each patient. Example, C412403.
- **Interaction (qualitative, datatype-string):** Unique ID related to patient transactions, procedures and interactions. Example, 8cd49b13-f45a-4b47-a2bd-173ffa932c2f
- **UID (qualitative, datatype-string):** Unique ID related to patient transactions, procedures and interactions. Example, 3a83ddb66e2ae73798bdf1d705dc0932
- **City (qualitative, datatype-string):** Example, Eva
- **State (qualitative, datatype-string):** Example, AL
- **County (qualitative, datatype-string):** Example, Morgan
- **Zip (qualitative, datatype-integer):** Example, 35621
- **Lat (quantitative, datatype-float):** Example, 34.3496
- **Lng (quantitative, datatype-float):** Example, -86.7251

- **Population (quantitative, datatype-integer):** Example, 2951
- **Area (qualitative, datatype-string):** Example, Suburban
- **Timezone (qualitative, datatype-string):** Example, America/Chicago
- **Job (qualitative, datatype-string):** Example, Psychologist, sport and exercise
- **Children (quantitative, datatype-float): Example**, 1
- **Age (quantitative, datatype-float):** Example, 53
- **Education (qualitative, datatype-string):** Example, Some College, Less than 1 Year
- **Employment (qualitative, datatype-string):** Example, Full Time
- **Income (qualitative, datatype-float):** Example, 86575.93
- **Marital (qualitative, datatype-string):** Example, Divorced
- **Gender (qualitative, datatype-string):** Example, Male
- **ReAdmins (qualitative, datatype-string):** Contains 'Yes/No' values on whether patients were readmitted within a month of release. Examaple, Yes
- **VitD_levels(quantitative, datatype-float):** Patients vitamin D levels measured in ng/mL. Example, 17.80233049
- **Doc_visits (quantitative, datatype-integer):** Number of times the Primary doctor visited the patient on the initial hospitalization. Example, 6
- **Full_meals_eaten (quantitative, datatype-integer):** Number of full meals eaten during hospitalization. Example, 0

- **VitD_supp (quantitative, datatype-integer):** Number of vitamin D supplement tablets administered to patient. Example, 0
- **Soft_drink(qualitative, datatype-string):** Whether the patient consumes three or more sodas a day. Example, (Yes/No).
- **Initial_admin(qualitative, datatype-string):** Type of initial admission patient was admitted as. Example, Emergency Admission.
- **HighBlood(qualitative, datatype-string):** Whether patient has high blood pressure or not. Example, Yes/No.
- **Stroke (qualitative, datatype-string):** Whether patient had a Stroke or not. Example, Yes/No.
- **Complication_risk(qualitative, datatype-string):** Level of risk of patient assessed by Primary Doctor. Example, High.
- **Overweight (qualitative, datatype-float):** Whether patient is considered overweight based on heaight, age and gender.
- **Arthritis (qualitative, datatype- string):** Whether the patient has arthritis or not. Example, Yes/No.
- **Diabetes (qualitative, datatype-string):** Whether patient has diabetes or not. Example, Yes/No.
- **Hyperlipidemia (qualitative, datatype-string):** Whether patients have hyperlipidemia. Example, Yes/No.
- **BackPain (qualitative, datatype-string):** Whether the patient suffers from backpain or not. Example, Yes/No.
- **Anxiety (qualitative, datatype-float):** A binary variable, this column indicates if the patient has an anxiety disorder. Example: 1

- **Allergic_rhinitis (qualitative, datatype-string)**: A binary variable, this column indicates whether or not the patient has allergic rhinitis.Example, Yes/No
- **Reflux_esophagitis (qualitative, datatype-string):** Whether patient has reflux esophagitis or not. Example, Yes/no.
- **Asthma (qualitative, datatype-string):** This column indicates if the patient has asthma. Example, Yes/No
- **Services (qualitative, datatype-string):** This column notes the service provided to the patient during their hospitalization. Example, Blood Work.
- **Initial_days (quantitative, datatype-float):** The number of days the patient was admitted to the hospital for during their initial visit. Example, 10.585770
- **TotalCharge (quantitative, datatype-float):** This column provides the cost in USD charged to the patient for their care. Reflects average per patient based on total charge divided by number of days hospitalized. Does not include specialized treatments. Example, 3191.048774
- **Additional_charges (quantitative,datatype-float):** This column provides the cost in USD charged to the patient for miscellaneous procedures, treatments, medicines, anesthesiology, etc. Example:17939.403420
- **Item1 (qualitative, datatype-integer)**: This column documents the patient's response to a survey question regarding the importance of timely admission. The survey uses 1 to indicate that this is "most important" and an 8 to indicate that this is "least important". Example, 3

- **Item2 (qualitative, datatype-integer):** This column documents the patient's response to a survey question regarding timely treatment. The survey uses 1 to indicate that this is "most important" and an 8 to indicate that this is "least important". Example, 1
- **Item3 (qualitative, datatype-integer):** This column documents the patient's response to a survey question regarding timely visits. The survey uses 1 to indicate that this is "most important" and an 8 to indicate that this is "least important". Example,
- **Item4 (qualitative, datatype-integer):** This column documents the patient's response to a survey question regarding reliability. The survey uses 1 to indicate that this is "most important" and an 8 to indicate that this is "least important". Example, 2
- **Item5 (qualitative, datatype-integer):** This column documents the patient's response to a survey question regarding the importance of options. The survey uses 1 to indicate that this is "most important" and an 8 to indicate that this is "least important". Example, 2
- **Item6 (qualitative, datatype-integer):** This column documents the patient's response to a survey question regarding hours of treatment. The survey uses 1 to indicate that this is "most important" and an 8 to indicate that this is "least important". Example, 3
- **Item7 (qualitative, datatype-integer):** This column documents the patient's response to a survey question regarding courteous staff. The survey uses 1 to indicate that this

is "most important" and an 8 to indicate that this is "least important". Example, 3

- **Item8 (qualitative, datatype-integer):** This column documents the patient's response to a survey question regarding the evidence of active listening from doctor. The survey uses 1 to indicate that this is "most important" and an 8 to indicate that this is "least important". Example, 3

Part II: Data cleaning plan:

B1. Plan to identify data anomaly

To assess the quality of the provided dataset:

- Import the csv file and use the provided data dictionary to understand each variable.
- Using the info() function, examine the data provided to find any irrelevant or redundant columns and assess the datatypes of each column to determine if they are appropriate to the given data.
- Using the 'duplicated()' function, identify any duplicate records.
- Examine each column to look for data that is incorrect or irrelevant using function 'unique()'.
- Columns that must contain unique values data such as Customer_id can be verified using 'value_counts().count()'
- Using the 'isnull().sum()' function to the dataframe, examine the columns with missing values.

- Find outliers that show statistical significance using histograms.

B2. Justification of Plan

The proposed approach for assessing the quality of data is comprehensive and systematic. Understanding the context, purpose, and structure using the data dictionary helps defy expectations for each variable. Ensuring the data falls between the expected range helps determine its correctness and reliability. Identifying and quantifying missing and duplicate data is essential for understanding the impact of the analysis and making decisions to handle it. Identifying outliers avoids misleading conclusions. Correct data types are necessary for accurate analysis to prevent misinterpretations.

B3. Programming language

The programming language used to analyze, clean, and store the dataset is Python. It is a simple yet powerful tool. The packages used for my analysis are:

- **Pandas**: used to analyze, clean, explore and manipulate data.
- **NumPy**: provides mathematical function used to transform or standardize the data.
- **Sklearn**: used to perform machine learning and statistical modeling. For this assignment, it was used to perform Principal Component Analysis.

- **Seaborn**: used to make statistical graphs in Python. In this assignment is was used to graph screeplots as a part of PCA.
- **Matplotlib**: graph plotting library that serves visual utility.
- **Scipy**: used to solve scientific and math problems

Part III: Data cleaning

C1. Findings of data quality issues

- Zipcode is stored as integer and thus, lose their leading 0's.
- Area is stored as a string but would be easier to interpret and analyze if categorical.
- Timezone values must be mapped to the standardized time zones across the US. Instead, they appear to be broken down by the city.
- Job is stored as a string but would be easier to interpret and analyze if categorical.
- Children is stored as float datatype, should be an integer.
- Age is stored as float datatype, should be an integer.
- Education is stored as a string but would be easier to interpret and analyze if categorical.
- Employment is stored as a string but would be easier to interpret and analyze if categorical.
- Marital is stored as a string but would be easier to interpret and analyze if categorical.

- Gender is stored as a string but would be easier to interpret and analyze if categorical. According to the data dictionary self-identification must say ‘Male’, ‘Female’, and ‘nonbinary’
- Initial_admin is stored as a string but would be easier to interpret and analyze if categorical.
- Complication_risk is stored as a string but would be easier to interpret and analyze if categorical.
- Services is stored as a string but would be easier to interpret and analyze if categorical.
- Item1 is stored as a string but would be easier to interpret and analyze if categorical. Item2 column name needs to be renamed to ‘Timely Admission’.
- Item2 is stored as a string but would be easier to interpret and analyze if categorical. Item2 column name needs to be renamed to ‘Timely treatment’.
- Item3 is stored as a string but would be easier to interpret and analyze if categorical. Item3 column name needs to be renamed to ‘Timely visits’.
- Item4 is stored as a string but would be easier to interpret and analyze if categorical. Item4 column name needs to be renamed to ‘Reliability’.
- Item5 is stored as a string but would be easier to interpret and analyze if categorical. Item5 column name needs to be renamed to ‘Options’.
- Item6 is stored as a string but would be easier to interpret and analyze if categorical. Item6 column name needs to be renamed to ‘Hours of treatment’

- Item7 is stored as a string but would be easier to interpret and analyze if categorical. Item7 column name needs to be renamed to ‘Courteous staff’.
- Item8 is stored as a string but would be easier to interpret and analyze if categorical. Item8 column name needs to be renamed to ‘Evidence of active listening from doctor’.
- Initial_days is float, should be an integer.
- Total_charge is float and can be rounded up to two decimal places for better interpretation.
- Additional_charged is float and can be rounded up to decimal values for better interpretation.
- The columns Children, Age, Income, Soft_drink, Overweight, Anxiety and Initial_days do not contain 10,000 entries suggesting they have NULL values.

C2. Mitigating data quality issues

The Zipcode was converted to string and front filled with 0s to ensure each column is five digits long. The columns Area, Job, Education, Employment, Marital, Complication_risk, and Services need to be stored as categorical and were recast using the ‘astype()’ function. Similarly, the Age and Initial_days can be converted to integers. Using the function ‘round()’, the values for Total_charges and Additional_charges were rounded to two decimal places.

The Timezone values have 26 time zones broken down by the city. This data must be replaced with the nine standardized time zones in the US. The Gender column needs to replace 'Prefer not to answer' values with 'Non Binary' according to the data dictionary. With the help of the data dictionary provided, the survey_responses were created to store ordered categories representing a range of survey scores, with "1" ("most important") being greater or higher than "8" ("least important") and the columns Item1 to Item8 were renamed. Each column was then cast as a category datatype.

The missing values for the columns Age, Children, Income, and Initial_days were imputed using the median of the values. The median is the most frequent variable and maintains the overall distribution of the data to ensure the general pattern and characteristics of the dataset remain intact. The mode value imputes the columns Soft_drink, Overweight, and Anxiety. The mode value is best for values that are categorical or discrete.

C3. Outcome of data cleaning

The data cleaning process provides high–quality data for accurate analysis. For example, it is renaming the values of the Timezone column to nine standardized values rather than 26 time zones for more straightforward interpretation. Another example is the conversion of specific columns to categorical datatypes, which restricts the potential values that can be input into the column, standardizing the entry. Renaming of column names helps to make the data more readable.

C4. Limitations of the data cleaning process:

Finding errors in the datatypes like typos, incorrect values were time consuming. Imputing the missing value with mode or median can distort the distribution of data. It can also cause loss of information when imputation is done replacing a missing value with a single value which does not account for uncertainty of what the missing value should be.

C5. Discussion of limitations

The median of the values was used to impute the missing values of the columns Age, Children, Income, and Initial_days. The median can distort the data distribution, mainly because many missing values exist. The columns Soft_drink, Overweight, and Anxiety were inputted using mode that can lead to loss of information and impact the data analysis down the line. Both mode and median imputation assume that the data is missing completely at random, and if the missing data depends on other variables, this could introduce bias.

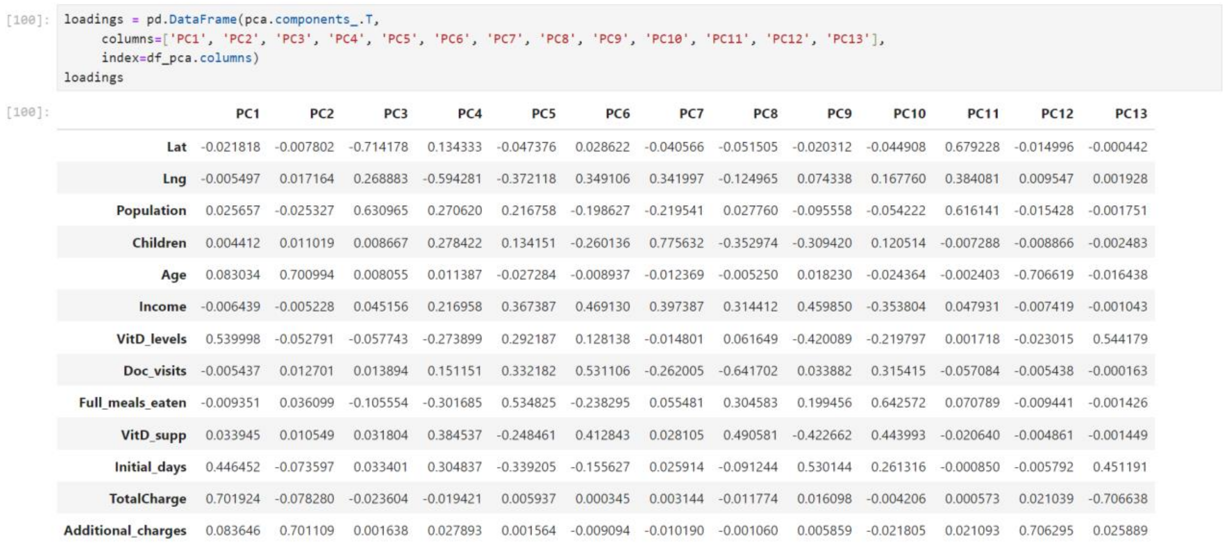
D1. See attached file for code titled ‘D206PA_Oviya’.

D2. See attached CSV file for cleaned dataset titled ‘output’.

E1. Principal Component Analysis and Loading Matrix

The variables used in the Principal Component Analysis are Latitude, Longitude, Population, Children, Age, Income, Vitamin D levels, Doctor visits, Full meals eaten, Vitamin D supplements, Initial days, Total Charge and additional charges.

Below is the screenshot of the output of the principal components loading matrix:



```
[100]: loadings = pd.DataFrame(pca.components_.T,
            columns=['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'PC7', 'PC8', 'PC9', 'PC10', 'PC11', 'PC12', 'PC13'],
            index=df_pca.columns)
       loadings
```

[100]:

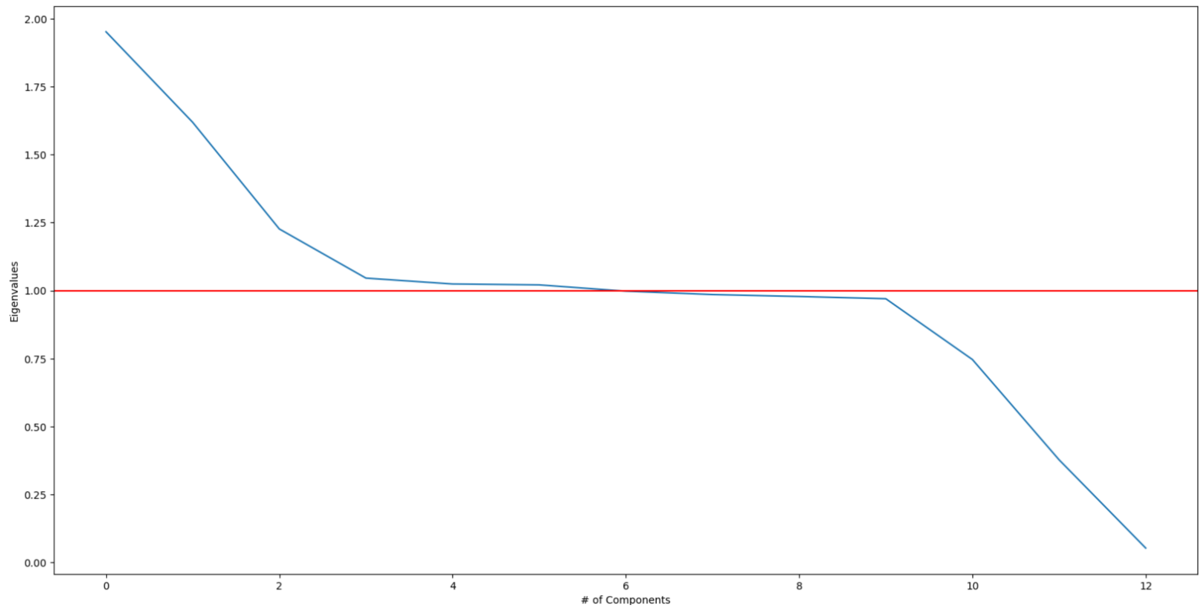
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
Lat	-0.021818	-0.007802	-0.714178	0.134333	-0.047376	0.028622	-0.040566	-0.051505	-0.020312	-0.044908	0.679228	-0.014996	-0.000442
Lng	-0.005497	0.017164	0.268883	-0.594281	-0.372118	0.349106	0.341997	-0.124965	0.074338	0.167760	0.384081	0.009547	0.001928
Population	0.025657	-0.025327	0.630965	0.270620	0.216758	-0.198627	-0.219541	0.027760	-0.095558	-0.054222	0.616141	-0.015428	-0.001751
Children	0.004412	0.011019	0.008667	0.278422	0.134151	-0.260136	0.775632	-0.352974	-0.309420	0.120514	-0.007288	-0.008866	-0.002483
Age	0.083034	0.700994	0.008055	0.011387	-0.027284	-0.008937	-0.012369	-0.005250	0.018230	-0.024364	-0.002403	-0.706619	-0.016438
Income	-0.006439	-0.005228	0.045156	0.216958	0.367387	0.469130	0.397387	0.314412	0.459850	-0.353804	0.047931	-0.007419	-0.001043
VitD_levels	0.539998	-0.052791	-0.057743	-0.273899	0.292187	0.128138	-0.014801	0.061649	-0.420089	-0.219797	0.001718	-0.023015	0.544179
Doc_visits	-0.005437	0.012701	0.013894	0.151151	0.332182	0.531106	-0.262005	-0.641702	0.033882	0.315415	-0.057084	-0.005438	-0.000163
Full_meals_eaten	-0.009351	0.036099	-0.105554	-0.301685	0.534825	-0.238295	0.055481	0.304583	0.199456	0.642572	0.070789	-0.009441	-0.001426
VitD_supp	0.033945	0.010549	0.031804	0.384537	-0.248461	0.412843	0.028105	0.490581	-0.422662	0.443993	-0.020640	-0.004861	-0.001449
Initial_days	0.446452	-0.073597	0.033401	0.304837	-0.339205	-0.155627	0.025914	-0.091244	0.530144	0.261316	-0.000850	-0.005792	0.451191
TotalCharge	0.701924	-0.078280	-0.023604	-0.019421	0.005937	0.000345	0.003144	-0.011774	0.016098	-0.004206	0.000573	0.021039	-0.706638
Additional_charges	0.083646	0.701109	0.001638	0.027893	0.001564	-0.009094	-0.010190	-0.001060	0.005859	-0.021805	0.021093	0.706295	0.025889

The code for the Principal Component Analysis and loading matrix can be found in the attached 'D206PA_Oviya.ipynb' file.

E2. Identification of Principal Components

The Kaiser criterion suggests retaining the principal components with eigenvalues greater than 1. This is because an eigenvalue greater than 1 indicates that the component explains more variance than an original value. The Eigen values found are: 1.951558666398377, 1.6188577832320286, 1.226714356402982, 1.0459139185057014, 1.024377766460927, 1.0210626201621402, 0.9979285924453598, 0.9854967960265328, 0.978282713729198, 0.9701602928158023, 0.7466662433116886, 0.37810167558295293, 0.05357857492629257. According to Kaiser criterion the first six components can be retained as they all have values greater than 1.

Screenshot of the screeplot based on my Eigen values.



E3. Benefits of PCA

PCA helps us reduce the dimensionality of data to generalize machine learning models better. Models that run on high-dimensional data face challenges like overfitting, distance between data points become negligible and computing high dimensional data can be expensive and complex. PCA helps include reduction of noise in the data, feature selection and the ability to produce uncorrelated features of data. From the PCA for the medical dataset it can be observed that Total_charge has the highest possible positive loading suggesting that total charge variations are a primary factor in this component. Vitamin_D levels have the second highest positive loading and are closely related to Total_charge. Initial_days has a substantial positive loading on PC1 suggesting a relationship between initial_days, total_charges and Vitamin_D levels. Additional_charges and age also play a small role in the variance captured in PC1.

Part IV: Supporting Documents

F. Panapto recording:

G. Web sources:

WGU courseware : https://my.wgu.edu/courses/course/34480019

https://www.datacamp.com/blog/curse-of-dimensionality-machine-learning

https://www.geeksforgeeks.org/introduction-to-pandas-in-python/

H. References:

WGU courseware : https://my.wgu.edu/courses/course/34480019 helped me understand PCA.

The Benefits of PCA (Principal Component Analysis)

Retrived from: https://www.bigabid.com/what-is-pca-and-how-can-i-use-it/#:~:text=PCA%20can%20help%20us%20improve,uncorrelated%20features%20of%20the%20data.

Avi Chawla (2023, May). The Advantages and Disadvantages of PCA to Consider before using it.

Retrieved: https://blog.dailydoseofds.com/p/the-advantages-and-disadvantages