**D206 Essay**

Darien Nguyen

Western Governors University

D206: Data Cleaning

November 11, 2022

**Part I: Research Question**

**A.  Describe one question or decision that you will address using the data set you chose. The summarized question or decision must be relevant to a realistic organizational need or situation.**

I will be addressing the health score of patients in relation to doctor visits found within the medical records in order to analyze the correlation between the two.

**B.  Describe the variables in the data set and indicate the specific type of data being described. Use examples from the data set that support your claims.**

The variables are the patients test reports and overall health. Test results being diabetes, rhinitis, esophagitis, asthma, etc. and health being anxiety, backpain, hyperlipidemia and others. The other set of data being used is each patients’ number of doctor visits.

**Part II: Data-Cleaning Plan**

**C.  Explain the plan for cleaning the data by doing the following:**

**1.  Propose a plan that includes the relevant techniques and specific steps needed to identify anomalies in the data set.**

The first step used to clean up the data set is through Python panadas fillna function. Fillna allows the user to fill any void or na variables with any character chosen by the user. In this case the integer 0 was chosen to fill in any needed void values.

**2.  Justify your approach for assessing the quality of the data, include:**

**•  characteristics of the data being assessed,**

**•  the approach used to assess the quality.**

The qualitative data was assessed through the replacement of “Yes” and “No” values to 1 and 0. By doing so, it allowed the data to be analyzed quantitatively. By converting our data to a numerical state, we can perform a wide range of statistical analyses that cannot be observed otherwise.

**3.  Justify your selected programming language and any libraries and packages that will support the data-cleaning process.**

I chose Python as my means of approach as I know Python allows for a wider range of tools through packages that provide many useful functions that would be otherwise tiresome to write from scratch. The packages used were pandas, NumPy, SciPy, matplotlib, sklearn, and seaborn. Pandas provides a greater range of manipulation for our data structures. Numpy helps to mathematically tackle those arrays. SciPy, Matplotlib, and seaborn aids in visualization for our data. Sklearn would assist in creating linear regressions and statistical analysis for our data.

**4.  Provide the code you will use to identify the anomalies in the data.**

hs = df.loc[ : , 'health\_score']

print (hs)

df['zscore\_hs'] = stats.zscore(hs)

df['zscore\_hs'].head()

dv = df.loc[ : , 'Doc\_visits']

print (dv)

df['zscore\_dv'] = stats.zscore(dv)

df['zscore\_dv'].head()

df.info()

df.hist(['health\_score'])

df.boxplot('health\_score')

df.boxplot('Doc\_visits')

df.hist('Doc\_visits')

df['health\_score'].notnull().any()

df['Doc\_visits'].notnull().any()

df.info()

df.hist(['zscore\_hs'])

df.hist(['zscore\_dv'])

**Part III: Data Cleaning**

**D.  Summarize the data-cleaning process by doing the following:**

**1.  Describe the findings, including all anomalies, from the implementation of the data-cleaning plan from part C.**

Within the medical records, we experience many null values within our desired data set that could simply not have been tested for or asked from the medical providers. The function “fillna” fills in all void components within the medical record with any desired value from the user. In this case, 0 was used as a place holder to quantify the finalized health score. Since we are assuming that the patients are negative for those void values, that patient could be presumed as a false positive and would need additional testing or medical counsel for confirmation. The reason we opted for this route is due to a lack of information so as not to conduct any unnecessary testing or visits from a false positive.

**2.  Justify your methods for mitigating each type of discovered anomaly in the data set.**

A simple print function was used to visually see any void values within are needed variables. The options to void the unknown values as “0” was to assume that the medical patient does not have that condition. If we were to presume otherwise, a patients’ health score would be presented as a false positive and give more warranty than what is entailed. Thus, resulting in more medical visits and testing that was not necessary.

**3.  Summarize the outcome from the implementation of *each* data-cleaning step.**

Fillna will fill all void values with 0.

Replace was used to convert qualitative “Yes” and “No” into quantitative “1” and “0”.

**4.  Provide the code used to mitigate anomalies.**

df = df.fillna((0))

df = df.replace(to\_replace = {'HighBlood': {'Yes' : 1 , 'No' : 0}})

print (df['HighBlood'])

df = df.replace(to\_replace = {'Stroke': {'Yes' : 1 , 'No' : 0}})

print (df['Stroke'])

df = df.replace(to\_replace = {'Overweight': {'Yes' : (1) , 'No' : (0)}})

df['Overweight'] = df['Overweight'].astype(int)

print (df['Overweight'])

df = df.replace(to\_replace = {'Arthritis': {'Yes' : 1 , 'No' : 0, 'NA' : 0}})

print (df['Arthritis'])

df = df.replace(to\_replace = {'Diabetes': {'Yes' : 1 , 'No' : 0}})

print (df['Diabetes'])

df = df.replace(to\_replace = {'Hyperlipidemia': {'Yes' : 1 , 'No' : 0}})

print (df['Hyperlipidemia'])

df = df.replace(to\_replace = {'BackPain': {'Yes' : 1 , 'No' : 0}})

print (df['BackPain'])

df = df.replace(to\_replace = {'Anxiety': {'Yes' : 1 , 'No' : 0}})

df['Anxiety'] = df['Anxiety'].astype(int)

print (df['Anxiety'])

df = df.replace(to\_replace = {'Allergic\_rhinitis': {'Yes' : 1 , 'No' : 0}})

print (df['Allergic\_rhinitis'])

df = df.replace(to\_replace = {'Reflux\_esophagitis': {'Yes' : 1 , 'No' : 0}})

print (df['Reflux\_esophagitis'])

df = df.replace(to\_replace = {'Asthma': {'Yes' : 1 , 'No' : 0}})

print (df['Asthma'])

**5.  Provide a copy of the cleaned data set.**

df.to\_csv(r'D:\Personal\cleaned\_data.csv', index = False)

Copy of cleaned data found in file submission

**6.  Summarize the limitations of the data-cleaning process.**

All null values were converted with a placeholder of “0”. This results in the loss of the null value itself and cannot be converted back to the original void value. Trying to revert the value back to 0 would also capture the values that held their original character as “0”.

**7.  Discuss how the limitations in part D6 affect the analysis of the question or decision from part A.**We are assuming that any patient with a filled in null value has been tested negative regarding that health test. This assumption might skew the relationship between that individual’s health score and their amount of doctor visits.

**E.  Apply principal component analysis (PCA) to identify the significant features of the data set by doing the following:**

**1.  List the principal components in the data set.**

Graphical user interface, text

Description automatically generated

**2.  Describe how you identified the principal components of the data set.**

To identify the components in a principal component analysis (PCA), I needed to calculate the eigenvectors and eigenvalues of the covariance matrix of the data. The eigenvectors are the principal components, and the eigenvalues indicate the amount of variance in the data that is explained by each component. Standardizing the data and taking the dot product of the standardized data matrix with its transpose gave me the covariance matrix, then a mathematical algorithm can then be used to calculate the eigenvectors and eigenvalues. Once I had identified the principal components, I used them to transform the data into a lower-dimensional space for easier visualization and analysis. The code below and visualizations showcases the processes and results. Based on the elbow rule of the scree plot, the number of clusters needed that capture the most variation is only one—anything more is considered irrelevant.

newdf\_normalized = (newdf-newdf.mean())/newdf.std()

pca = PCA(n\_components = newdf.shape[1])

pca.fit(newdf\_normalized)

newdf\_pca = pd.DataFrame(pca.transform(newdf\_normalized), columns = ['p1', 'p2', 'p3'])

cov\_matrix = np.dot(newdf\_normalized.T, newdf\_normalized) / newdf.shape[0]

eigenvalues = [np.dot(eigenvector.T, np.dot(cov\_matrix, eigenvector)) for eigenvector in pca.components\_]

loading = pd.DataFrame(pca.components\_.T,

columns = ['p1', 'p2', 'p3'],

index = newdf.columns)

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

plt.plot(pca.explained\_variance\_ratio\_)

plt.xlabel('# of components')

plt.ylabel('explained variance')

plt.show()

plt.plot(eigenvalues)

plt.xlabel('# of components')

plt.ylabel('eigenvalues')

plt.show()

**3.  Describe how the organization can benefit from the results of the PCA**

If a hospital used principal component analysis (PCA) to identify the most important features of their patients' health scores and hospital readmission rates, they could use this information to improve their care and reduce readmission rates. By identifying the principal component and clusters in the data, the hospital could gain a better understanding of the factors that are most important for predicting readmission rates (Jolliffe & Cadima, 2016). This could help them to identify the patients who are most at risk for readmission, and to develop targeted interventions to prevent readmissions. For example, if the principal component was related to patients' age and comorbidities, the hospital could focus their interventions on older patients with multiple chronic conditions.

**Part IV. Supporting Documents**

**F. Video**

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=f1624c10-c57d-49f5-a76f-af5e014db865>

**G.  Code Sources**

*Pandas.dataframe.fillna#*. pandas.DataFrame.fillna - pandas 1.5.2 documentation. (n.d.). Retrieved November 30, 2022, from <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html>

*Pandas.dataframe.replace#*. pandas.DataFrame.replace - pandas 1.5.2 documentation. (n.d.). Retrieved November 30, 2022, from <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.replace.html>

**H.  Intext Sources**

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202.