**HOW DOES INCOME AFFECTS MONTHLY CHARGES ON INTERNET SERVICES?**

NUM2 – NUM2 TASK1: DATA CLEANING

DATA CLEANING – D206

PRFA – NUM2

STUDENT NAME: IBRAHIM SULEIMAN

DATA ANALYTICS

STUDENT NUMBER: 001429984

COLLEGE OF INFORMATION TECHNOLOGY, WESTERN GOVERNORS UNIVERSITY

December 5th, 2021

**HOW DOES INCOME AFFECTS MONTHLY CHARGES ON INTERNET SERVICES?**

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.**

**The description states a question or decision that can be addressed through analysis of the chosen data set.**

How does income affects monthly charges on internet services?

The question is relevant to a realistic organizational need or situation in creating innovative solution for the need of having internet security based on the customer’s income and subscriptions with regards to the rate at which internet service is used.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Names** | **Data Types** | **Descriptions of the Variables** |  |
| Age | float64 | 7525 non-null |  |
| Children | float64 | 7505 non-null |  |
| Income | float64 | 7510 non-null |  |
| Tenure | float64 | 9069 non-null |  |
| Monthly charges | float64 | 10000 non-null |  |
| Bandwidth\_GB\_Year' | float64 | 8979 non-null |  |

**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 description includes the variables in the data set and indicates the specific type of data being described, and includes examples from the data set to support claims.***

The data set is Churn CSV dataset which will be imported in order to describe the datasets to support the claims (Deepanshu Bhalla,2020.).

**Data Table**

|  |
| --- |
| import pandas as pd  # Info of column names along with the number of non –null values in each column, Categorical  df = pd.read\_csv('churn\_data')  df.info()  <class 'pandas.core.frame.DataFrame'>  RangeIndex: 10000 entries, 0 to 9999  Data columns (total 52 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 Unnamed: 0 10000 non-null int64  1 CaseOrder 10000 non-null int64  2 Customer\_id 10000 non-null object  3 Interaction 10000 non-null object  4 City 10000 non-null object  5 State 10000 non-null object  6 County 10000 non-null object  7 Zip 10000 non-null int64  8 Lat 10000 non-null float64  9 Lng 10000 non-null float64  10 Population 10000 non-null int64  11 Area 10000 non-null object  12 Timezone 10000 non-null object  13 Job 10000 non-null object  14 Children 7505 non-null float64  15 Age 7525 non-null float64  16 Education 10000 non-null object  17 Employment 10000 non-null object  18 Income 7510 non-null float64  19 Marital 10000 non-null object  20 Gender 10000 non-null object  21 Churn 10000 non-null object  22 Outage\_sec\_perweek 10000 non-null float64  23 Email 10000 non-null int64  24 Contacts 10000 non-null int64  25 Yearly\_equip\_failure 10000 non-null int64  26 Techie 7523 non-null object  27 Contract 10000 non-null object  28 Port\_modem 10000 non-null object  29 Tablet 10000 non-null object  30 InternetService 10000 non-null object  31 Phone 8974 non-null object  32 Multiple 10000 non-null object  33 OnlineSecurity 10000 non-null object  34 OnlineBackup 10000 non-null object  35 DeviceProtection 10000 non-null object  36 TechSupport 9009 non-null object  37 StreamingTV 10000 non-null object  38 StreamingMovies 10000 non-null object  39 PaperlessBilling 10000 non-null object  40 PaymentMethod 10000 non-null object  41 Tenure 9069 non-null float64  42 MonthlyCharge 10000 non-null float64  43 Bandwidth\_GB\_Year 8979 non-null float64  44 item1 10000 non-null int64  45 item2 10000 non-null int64  46 item3 10000 non-null int64  47 item4 10000 non-null int64  48 item5 10000 non-null int64  49 item6 10000 non-null int64  50 item7 10000 non-null int64  51 item8 10000 non-null int64  dtypes: float64(9), int64(15), object(28)  memory usage: 4.0+ MB |

|  |
| --- |
| df.columns  Index(['Unnamed: 0', 'CaseOrder', 'Customer\_id', 'Interaction', 'City',  'State', 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area',  'Timezone', 'Job', 'Children', 'Age', 'Education', 'Employment',  'Income', 'Marital', 'Gender', 'Churn', 'Outage\_sec\_perweek', 'Email',  'Contacts', 'Yearly\_equip\_failure', 'Techie', 'Contract', 'Port\_modem',  'Tablet', 'InternetService', 'Phone', 'Multiple', 'OnlineSecurity',  'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',  'StreamingMovies', 'PaperlessBilling', 'PaymentMethod', 'Tenure',  'MonthlyCharge', 'Bandwidth\_GB\_Year', 'item1', 'item2', 'item3',  'item4', 'item5', 'item6', 'item7', 'item8', 'Z\_Score\_Age',  'Z\_Score\_Children', 'Z\_Score\_Income', 'Z\_Score\_Tenure',  'Z\_Score\_Bandwidth\_GB\_Year'],  dtype='object') |

Part II: **Data-Cleaning Plan**

Python programming language was used for implementing coding solutions, manipulating the data, and creating visual representations.

**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 proposal includes a detailed description of the techniques and steps needed for identifying anomalies in the selected data set.***

|  |  |
| --- | --- |
| **Method** | **Description of the technique** |
| Deletion/ Dropping | This will remove the rows or columns unwanted |
| Univariate Imputation | This will replace missing values of specific variables by applying mean, median and mode respectively |

**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 justification includes the characteristics of the data being assessed and references the approach used to assess the quality of the data. The justified approach aligns with the selected data set.***

[1.] The data being assessed has several inconsistency and empty rows with columns of ‘phone’, ‘Tenure’, and ‘Techie’.

[2.] The approach used to assess the quality of the data was to find missing values within the data set using python which were inform of null. Therefore, the function used to assess the null values in the data was isnull()sum function respectively.

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

***The justification describes the benefits of using the programming language, including any libraries and packages used to clean the data, and includes specific examples of how these tools are ideal in this scenario as opposed to other available tools.***

[1]. The selected programming language is Python executed within Jupyter notebook environment.

[2]. Microsoft Excel package was utilized in the data-cleaning process and saved as csv files.

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

***The submission provides the complete and executable code, which could be used to identify anomalies in the data set.***

|  |  |
| --- | --- |
| Code |  |
| df.isnull().sum() | This is a relevant technique that is needed to identify anomalies in the data set |

**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.**

***The description accurately includes all of the anomalies found by running the code from part C4.***

|  |
| --- |
| # Finding whether we have null values in the data  isnull().sum()  Unnamed: 0 0  CaseOrder 0  Customer\_id 0  Interaction 0  City 0  State 0  County 0  Zip 0  Lat 0  Lng 0  Population 0  Area 0  Timezone 0  Job 0  Children 2495  Age 2475  Education 0  Employment 0  Income 2490  Marital 0  Gender 0  Churn 0  Outage\_sec\_perweek 0  Email 0  Contacts 0  Yearly\_equip\_failure 0  Techie 2477  Contract 0  Port\_modem 0  Tablet 0  InternetService 0  Phone 1026  Multiple 0  OnlineSecurity 0  OnlineBackup 0  DeviceProtection 0  TechSupport 991  StreamingTV 0  StreamingMovies 0  PaperlessBilling 0  PaymentMethod 0  Tenure 931  MonthlyCharge 0  Bandwidth\_GB\_Year 1021  item1 0  item2 0  item3 0  item4 0  item5 0  item6 0  item7 0  item8 0  dtype: int64 |

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

***The justification includes the specific mitigation methods for each type of anomaly listed in part D1.***

|  |
| --- |
| # TREATING MISSING VALUES  # EXAMINE THE DISTRIBUTION  df.hist()  array([[<AxesSubplot:title={'center':'Unnamed: 0'}>,  <AxesSubplot:title={'center':'CaseOrder'}>,  <AxesSubplot:title={'center':'Zip'}>,  <AxesSubplot:title={'center':'Lat'}>,  <AxesSubplot:title={'center':'Lng'}>],  [<AxesSubplot:title={'center':'Population'}>,  <AxesSubplot:title={'center':'Children'}>,  <AxesSubplot:title={'center':'Age'}>,  <AxesSubplot:title={'center':'Income'}>,  <AxesSubplot:title={'center':'Outage\_sec\_perweek'}>],  [<AxesSubplot:title={'center':'Email'}>,  <AxesSubplot:title={'center':'Contacts'}>,  <AxesSubplot:title={'center':'Yearly\_equip\_failure'}>,  <AxesSubplot:title={'center':'Tenure'}>,  <AxesSubplot:title={'center':'MonthlyCharge'}>],  [<AxesSubplot:title={'center':'Bandwidth\_GB\_Year'}>,  <AxesSubplot:title={'center':'item1'}>,  <AxesSubplot:title={'center':'item2'}>,  <AxesSubplot:title={'center':'item3'}>,  <AxesSubplot:title={'center':'item4'}>],  [<AxesSubplot:title={'center':'item5'}>,  <AxesSubplot:title={'center':'item6'}>,  <AxesSubplot:title={'center':'item7'}>,  <AxesSubplot:title={'center':'item8'}>, <AxesSubplot:>]],  dtype=object) |
| # Skewed distribution positively skewed right  df.hist(column='Children')  array([[<AxesSubplot:title={'center':'Children'}>]], dtype=object) |
| # Spread narrow range  df.hist(column='Age')  array([[<AxesSubplot:title={'center':'Age'}>]], dtype=object) |
| # Categorical variables  # Skewed distribution positively skewed right  df.hist(column='Income')  array([[<AxesSubplot:title={'center':'Income'}>]], dtype=object) |
| # Bi modal distribution with two modes non-symmetric  df.hist(column='Tenure')  array([[<AxesSubplot:title={'center':'Tenure'}>]], dtype=object) |
| # Bi modal distribution non-symmetric  df.hist(column='Bandwidth\_GB\_Year')  array([[<AxesSubplot:title={'center':'Bandwidth\_GB\_Year'}>]], dtype=object) |

**# Performing Treatment of Outliers using Imputation**

|  |
| --- |
| df['Children'].fillna(df['Children'].median(), inplace=True) |
| df['Age'].fillna(df['Age'].mean(), inplace=True) |
| df['Income'].fillna(df['Income'].mean(), inplace=True) |
| df['Tenure'].fillna(df['Tenure'].mean(), inplace=True) |
| df['Bandwidth\_GB\_Year'].fillna(df['Bandwidth\_GB\_Year'].mean(), inplace=True) |
| df['TechSupport'].fillna(df['TechSupport'].mode()[0])  0 No  1 No  2 No  3 No  4 Yes  ...  9995 No  9996 No  9997 No  9998 Yes  9999 No  Name: TechSupport, Length: 10000, dtype: object  df.dropna(subset=['TechSupport'])  df.dropna()  df.dropna(subset = ["TechSupport"], inplace=True)  df.replace('TechSupport', " ", inplace=True)  nan\_value = float("NAN")  df.replace("", nan\_value, inplace=True)  df.dropna(subset = ["TechSupport"], inplace=True) |
| df['Techie'].fillna(df['Techie'].mode()[0])  0 No  1 Yes  2 Yes  3 Yes  4 No  ...  9995 No  9996 No  9997 No  9998 No  9999 No  Name: Techie, Length: 10000, dtype: object  df.dropna(subset=['Techie'])  df.dropna()  df.dropna(subset = ["Techie"], inplace=True)  df.replace('Techie', " ", inplace=True)  nan\_value = float("NAN")  df.replace("", nan\_value, inplace=True)  df.dropna(subset = ["Techie"], inplace=True) |
| df['Phone'].fillna(df['Phone'].mode()[0])  0 Yes  1 Yes  2 Yes  3 Yes  4 No  ...  9995 Yes  9996 Yes  9997 Yes  9998 No  9999 Yes  Name: Phone, Length: 10000, dtype: object  df.dropna(subset=['Phone'])  df.dropna()  df.dropna(subset = ["Phone"], inplace=True)  df.replace('Phone', " ", inplace=True)  nan\_value = float("NAN")  df.replace("", nan\_value, inplace=True)  df.dropna(subset = ["Phone"], inplace=True) |

|  |
| --- |
| # Lets Verify if NAs (NANs) were imputed using the isnull().sum function  df.isnull().sum()  Unnamed: 0 0  CaseOrder 0  Customer\_id 0  Interaction 0  City 0  State 0  County 0  Zip 0  Lat 0  Lng 0  Population 0  Area 0  Timezone 0  Job 0  Children 0  Age 0  Education 0  Employment 0  Income 0  Marital 0  Gender 0  Churn 0  Outage\_sec\_perweek 0  Email 0  Contacts 0  Yearly\_equip\_failure 0  Techie 0  Contract 0  Port\_modem 0  Tablet 0  InternetService 0  Phone 0  Multiple 0  OnlineSecurity 0  OnlineBackup 0  DeviceProtection 0  TechSupport 0  StreamingTV 0  StreamingMovies 0  PaperlessBilling 0  PaymentMethod 0  Tenure 0  MonthlyCharge 0  Bandwidth\_GB\_Year 0  item1 0  item2 0  item3 0  item4 0  item5 0  item6 0  item7 0  item8 0  dtype: int64 |

**# Detecting Outliers**

|  |
| --- |
| import numpy as np  import pandas as pd  from pandas import DataFrame  import scipy.stats as stats  # The code for calculating the z-score and a new column/variable for the z-score calculation.  df['Z\_Score\_Age']=stats.zscore(df['Age'])  # The code to display the calculated z-score and the values  df[['Age','Z\_Score\_Age']].head  <bound method NDFrame.head of Age Z\_Score\_Age  0 68.000000 0.817916  1 27.000000 -1.459588  2 50.000000 -0.181964  3 48.000000 -0.293062  4 83.000000 1.651149  ... ... ...  9995 53.275748 0.000000  9996 48.000000 -0.293062  9997 53.275748 0.000000  9998 39.000000 -0.793002  9999 28.000000 -1.404039  [10000 rows x 2 columns]>  # Matplotlibis deployed to plot and Visualize bars  %matplotlib inline  import matplotlib.pyplot as plt  plt.hist(df['Z\_Score\_Age'])  plt.show()    # Using Seaborn to provides visualization  import seaborn  # Outliers are not dictated  boxplot=seaborn.boxplot(x='Age',data=df)    df['Z\_Score\_Children']=stats.zscore(df['Children'])  df[['Children','Z\_Score\_Children']].head  <bound method NDFrame.head of Children Z\_Score\_Children  0 1.0 -0.427079  1 1.0 -0.427079  2 4.0 1.130655  3 1.0 -0.427079  4 0.0 -0.946323  ... ... ...  9995 3.0 0.611410  9996 4.0 1.130655  9997 1.0 -0.427079  9998 1.0 -0.427079  9999 1.0 -0.427079  [10000 rows x 2 columns]>  plt.hist(df['Z\_Score\_Children'])  plt.show()    # Outliers are dictated  boxplot=seaborn.boxplot(x='Children',data=df)    df['Z\_Score\_Income']=stats.zscore(df['Income'])  df[['Income','Z\_Score\_Income']].head  <bound method NDFrame.head of Income Z\_Score\_Income  0 28561.990000 -4.628805e-01  1 21704.770000 -7.419255e-01  2 39936.762226 1.184340e-15  3 18925.230000 -8.550350e-01  4 40074.190000 5.592431e-03  ... ... ...  9995 55723.740000 6.424290e-01  9996 39936.762226 1.184340e-15  9997 39936.762226 1.184340e-15  9998 16667.580000 -9.469069e-01  9999 39936.762226 1.184340e-15  [10000 rows x 2 columns]>  plt.hist(df['Z\_Score\_Income'])  plt.show()    # Outliers are dictated  boxplot=seaborn.boxplot(x='Income',data=df)    df['Z\_Score\_Tenure']=stats.zscore(df['Tenure'])  df[['Tenure','Z\_Score\_Tenure']].head  <bound method NDFrame.head of Tenure Z\_Score\_Tenure  0 6.795513 -1.100355e+00  1 1.156681 -1.324325e+00  2 15.754144 -7.445255e-01  3 17.087227 -6.915765e-01  4 1.670972 -1.303898e+00  ... ... ...  9995 68.197130 1.338469e+00  9996 61.040370 1.054208e+00  9997 34.498858 -2.822221e-15  9998 71.095600 1.453594e+00  9999 63.350860 1.145979e+00  [10000 rows x 2 columns]>  plt.hist(df['Tenure'])  plt.show()    # Outliers are not dictated  boxplot=seaborn.boxplot(x='Tenure',data=df)    df['Z\_Score\_Bandwidth\_GB\_Year']=stats.zscore(df['Bandwidth\_GB\_Year'])  df[['Bandwidth\_GB\_Year','Z\_Score\_Bandwidth\_GB\_Year']].head  <bound method NDFrame.head of Bandwidth\_GB\_Year Z\_Score\_Bandwidth\_GB\_Year  0 904.536110 -1.203462  1 800.982766 -1.253425  2 2054.706961 -0.648524  3 2164.579412 -0.595512  4 271.493436 -1.508895  ... ... ...  9995 6511.253000 1.501687  9996 5695.952000 1.108318  9997 4159.306000 0.366911  9998 6468.457000 1.481039  9999 5857.586000 1.186303  [10000 rows x 2 columns]>  In [ ]:  plt.hist(df['Bandwidth\_GB\_Year'])  plt.show()    # Outliers are not dictated  boxplot=seaborn.boxplot(x='Bandwidth\_GB\_Year',data=df)    df['Income'].quantile  <bound method Series.quantile of 0 28561.990000  1 21704.770000  2 39936.762226  3 18925.230000  4 40074.190000  ...  9995 55723.740000  9996 39936.762226  9997 39936.762226  9998 16667.580000  9999 39936.762226  Name: Income, Length: 10000, dtype: float64>  df['Age'].quantile  <bound method Series.quantile of 0 68.000000  1 27.000000  2 50.000000  3 48.000000  4 83.000000  ...  9995 53.275748  9996 48.000000  9997 53.275748  9998 39.000000  9999 28.000000  Name: Age, Length: 10000, dtype: float64>  df['Children'].quantile  <bound method Series.quantile of 0 1.0  1 1.0  2 4.0  3 1.0  4 0.0  ...  9995 3.0  9996 4.0  9997 1.0  9998 1.0  9999 1.0  Name: Children, Length: 10000, dtype: float64> |

**# Dropping outliers systematically**

|  |
| --- |
| Cleaned\_df = df[df['Children'] < 100]  Cleaned\_df.head()  df.duplicated()  0 False  1 False  2 False  3 False  4 False  ...  9995 False  9996 False  9997 False  9998 False  9999 False  Length: 10000, dtype: bool |

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

***The summary details the outcome from the implementation of each data-cleaning step. The summarized expected outcomes are plausible given the interventions.***

The outcome from implementing each data cleaning step is that the Null and Na and Nan values were cleaned and treated using pandas, and empty rows were dropped, however, duplicates were being dropped where outliers were detected.

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

***The submission provides complete and executable code that could be used to mitigate the anomalies.***

# Performing Treatment of Outliers by Dropping Rows containing outliers

|  |
| --- |
| df.drop\_duplicates(inplace = True)  df.head() |

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

***The submission includes a clean data set created from the raw data.***

The provided data set includes the complete list of variables from the chosen data set in part A.

icu\_csv\_file.csv

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

***The submission accurately summarizes the limitations of the implemented data-cleaning process.***

The main limitation of the data cleaning is matplotlib because it is not interactive and therefore there is lack of uniformity the environment for diverse functions.

**7.  Discuss how the limitations in part D6 affect the analysis of the question or decision from part A.**

***The submission includes a discussion of the impact of the limitations from part D6. The discussion logically aligns with the question or decision from part A.***

The deletion technic comes a long way in helping treat missing values however it has certain limitations such as:

[1.] There is quantum of data decrease and disappearance in the process.

[2.] The essential amount of the sample data is diminished.

The Univariate Imputation Technique also has certain limitations when it comes to the analysis:

[1.] It could possibly alter data dissemination

[2.] It could possibly account for ambiguity and confusion as a result of lost data

**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.**

[1.] Principal component 1 : This appears to be linear putting together the main predictable variables that entails the enormous variance within the data set.(it narrows the total of squared )

[2.]Principal component 2 : This is a straight mixture of the main predictors that contain the rest of the variance within the data set which appears to be uncorrelated with the variables in principal component 1 respectively.

***The submission lists all principal components of the data set.***

According to Carolina Bento (2020), Principle component analysis is a mechanism of acquiring extensive variables (in form of components) from a gigantic set of variables accessible in a data set.

['Zip', 'Lat', 'Outage\_sec\_perweek', 'MonthlyCharge', 'Z\_Score\_Children', 'Z\_Score\_Age', 'Z\_Score\_Income', 'Z\_Score\_Tenure', 'Z\_Score\_Bandwidth\_GB\_Year']

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

***The description of how the principle components of the data set were identified is accurate and complete.***

Standardization: It is important to standardized the data in order to predict accuracy in machine learning and algorithm .

PCA Projection to 2D: During the process of PCA, the original data has 9 columns ['Zip', 'Lat', 'Outage\_sec\_perweek', 'MonthlyCharge', 'Z\_Score\_Children', 'Z\_Score\_Age', 'Z\_Score\_Income', 'Z\_Score\_Tenure', 'Z\_Score\_Bandwidth\_GB\_Year']. However, the code executed upon the original data which is 9 dimensional into 2 dimentionals(2 dimensions of variations).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # pca analysis  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from sklearn.decomposition import PCA  from sklearn.preprocessing import StandardScaler  %matplotlib inline  df = pd.read\_csv('icu\_csv\_file.csv')  df.head(5)  features = ['Zip', 'Lat', 'Outage\_sec\_perweek', 'MonthlyCharge', 'Z\_Score\_Children', 'Z\_Score\_Age', 'Z\_Score\_Income', 'Z\_Score\_Tenure', 'Z\_Score\_Bandwidth\_GB\_Year']  x = df.loc[:, features].values  y = df.loc[:,['Z\_Score\_Income']]  x = StandardScaler().fit\_transform(x)  pd.DataFrame(data = x, columns = features).head()   |  | **Zip** | **Lat** | **Outage\_sec\_perweek** | **MonthlyCharge** | **Z\_Score\_Children** | **Z\_Score\_Age** | **Z\_Score\_Income** | **Z\_Score\_Tenure** | **Z\_Score\_Bandwidth\_GB\_Year** | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **0** | 1.837971 | 3.201134 | -0.636575 | -0.060044 | -0.434856 | 0.816535 | -0.470630 | -1.090283 | -1.193002 | | **1** | -0.014872 | 1.019010 | 0.082112 | 1.588539 | -0.434856 | -1.458704 | -0.754117 | -1.314279 | -1.242887 | | **2** | 1.737533 | 1.206977 | -0.170032 | -0.336952 | 1.124163 | -0.182350 | -0.000382 | -0.734411 | -0.638921 | | **3** | 1.551982 | -1.060615 | 0.537052 | -1.240604 | -0.434856 | -0.293337 | -0.869027 | -0.681455 | -0.585991 | | **4** | 1.026011 | -1.717106 | -0.353240 | -0.537074 | -0.954530 | 1.648940 | 0.005299 | -1.293849 | -1.497962 |   pca = PCA(n\_components=2)  principalComponents = pca.fit\_transform(x)  principalDf = pd.DataFrame(data = principalComponents  , columns = ['principal components 1', 'principal components 2'])  principalDf.head(5)   |  | **principal components 1** | **principal components 2** | | --- | --- | --- | | **0** | -1.588990 | -0.232701 | | **1** | -1.711159 | 1.398628 | | **2** | -0.963462 | -0.360283 | | **3** | -0.927081 | -0.640579 | | **4** | -2.002967 | -0.755230 |   df[['Z\_Score\_Income']].head(5)   |  | **Z\_Score\_Income** | | --- | --- | | **0** | -0.470630 | | **1** | -0.754117 | | **2** | -0.000382 | | **3** | -0.869027 | | **4** | 0.005299 |   finalDf = pd.concat([principalDf, df[['Z\_Score\_Income']]], axis = 1)  finalDf.head(5)   |  | **principal components 1** | **principal components 2** | **Z\_Score\_Income** | | --- | --- | --- | --- | | **0** | -1.588990 | -0.232701 | -0.470630 | | **1** | -1.711159 | 1.398628 | -0.754117 | | **2** | -0.963462 | -0.360283 | -0.000382 | | **3** | -0.927081 | -0.640579 | -0.869027 | | **4** | -2.002967 | -0.755230 | 0.005299 |   fig = plt.figure(figsize = (8, 8))  ax = fig.add\_subplot(1,1,1)  ax.set\_xlabel('Principal Component 1', fontsize = 15)  ax.set\_ylabel('Principal Component 2', fontsize = 15)  ax.set\_title('2 Component PCA', fontsize = 15)  Z\_Score\_Income = ['-0.470630', '-0.754117', '0.005299']  colors = ['b', 'g', 'r']  for Z\_Score\_Income, color in zip(Z\_Score\_Income,colors):  indicesToKeep = finalDf['Z\_Score\_Income'] == Z\_Score\_Income  ax.scatter(finalDf.loc[indicesToKeep, 'principal components 1']  , finalDf.loc[indicesToKeep, 'principal components 2']  , c = color  , s = 40)  ax.legend(Z\_Score\_Income)  ax.grid()    pca.explained\_variance\_ratio\_  array([0.21122418, 0.12641358]) |

Conclusively, the variance inform us on how the data was assigned separately among the two principal components. Therefore the first two components have 33.76% of data. The first principal component contains 21.12% variance and the second principal component 12.64% variance. Therefore, the third, fourth, fifth, sixth, seventh, eighth and nineth principal component entails the remaining variance of the processed data set respectively.

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

***The description of how the organization can benefit from the results of the PCA is logical and accurate.***

The fundamental recognition of PCA is dimensionality (appearance) contraction. It advance algorithm execution by eliminating related appearance nevertheless there are some data shrinkage. However, Principal Component Analysis(PCA) is extensively well-used in minimization of great proportion. Therefore, Evidently, We invariably extract tremendous dimensional statistics due to numeric as well as categorical data, and PCA is a helping hand in this scenario. That is the reason why it is vehement to take systematic consideration when cleaning data. Most importantly, these also concentrate principally according to how PCA is practically, comparatively than mathematical expression and some relevant points position focal point that requires maximum consideration when performing PCA . diversely, the organization could ascertain erroneous data.

Part IV. Supporting Documents

**F.  Provide a Panopto recording that demonstrates the warning- and error-free functionality of the code used to support the discovery of anomalies and the data cleaning process and summarizes the programming environment.**

***The Panopto video recording demonstrates the warning-and error-free functionality of the code used to support the discovery of anomalies and the data cleaning process. An accurate summary of the programming environment is provided in the video.***

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=5d73349d-585c-4244-92be-adf6002520ac>

**G.  Reference the web sources used to acquire segments of third-party code to support the application. Be sure the web sources are reliable.**

***The submission records all web sources used to acquire data or third-party code and all of the web sources are reliable.***

Deepanshu, B. (2020). *How to Import Data in Python.*

RSGB Business Consultant Pvt. Ltd. <https://www.listendata.com/2017/02/import-data-in-python.html#Import-CSV-files>

Carolina B. (2020). *Principal Component Analysis algorithm in Real-Life: Discovering patterns in a real-estate dataset.*

Medium. <https://towardsdatascience.com/principal-component-analysis-algorithm-in-real-life-discovering-patterns-in-a-real-estate-dataset-18134c57ffe7>

Pierre-Louis B. (2020). *Principle Components Analysis(PCA), Fundamentals, Benefits & Insights for Industry.*

Medium. <https://towardsdatascience.com/principal-components-analysis-pca-fundamentals-benefits-insights-for-industry-2f03ad18c4d7>

**H.** **The submission includes in-text citations for sources that are properly quoted, paraphrased, or summarized and a reference list that accurately identifies the author, date, title, and source location as available.**

No source was used.