Jason Smith  
Student ID: 293963  
jsmi268@wgu.edu  
WGU - MSDA  
D209- Data Mining I  
Data - Medical  
Program Mentor: Christiana Okhipo

In [1]:

*# Import libraries, set display parameters, import data set*

In [2]:

*# B3. List the packages or libraries you have chosen for Python and justify how each item on the list supports the analysis.*

*# Pandas - supports ability to create dataframes and multi-dimensional arrays*

**import** pandas **as** pd

*# Numpy - Used to create error curves which helped to determine an ideal K-value. Used in section D2*

**import** numpy **as** np

*# Seaborn - data visualization*

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**from** sklearn.preprocessing **import** LabelEncoder

*# Mute copy warnings*

**import** warnings

**from** pandas.core.common **import** SettingWithCopyWarning

warnings**.**simplefilter(action**=**"ignore", category**=**SettingWithCopyWarning)

*# Standard scaler - creates a common scale for data variables and reducing biased outcomes*

**from** sklearn.preprocessing **import** StandardScaler

*# KNN algorithm - classification algorithm*

**from** sklearn.neighbors **import** KNeighborsClassifier

*# Split data into train and test sets*

**from** sklearn.model\_selection **import** train\_test\_split

*# Determine accuracy\_score of KNN classifier*

**from** sklearn.metrics **import** accuracy\_score

**from** sklearn.metrics **import** classification\_report

*# Use conMaking the Confusion Matrix*

**from** sklearn.metrics **import** confusion\_matrix

*# Change display max rows and columns*

pd**.**options**.**display**.**max\_rows **=** **None**

pd**.**set\_option("display.max\_columns", 1000)

In [3]:

*# Import Clean Data Set*

df **=** pd**.**read\_csv('med\_data\_clean.csv', index\_col**=False**)

Part III: Data Preparation

In [4]:

*# C3. Explain each of the steps used to prepare the data for the analysis. Identify the code segment for each step.*

*# Identify number of rows and columns in the data set*

print("# of rows: ", df**.**shape[0])

print("# of columns: ", df**.**shape[1])

# of rows: 9999

# of columns: 52

In [5]:

*# Describe data set*

df**.**describe()

Out[5]:

|  | **Unnamed: 0** | **Zip** | **Latitude** | **Longitude** | **Population** | **VitD\_levels** | **Doc\_visits** | **Full\_meals\_eaten** | **VitD\_supp** | **TotalCharge** | **Additional\_charges** | **Admissions** | **Treatment** | **Visits** | **Reliability** | **Options** | **Hours** | **Curteous** | **Dr\_comm** | **Children** | **Age** | **Income** | **Overweight** | **Anxiety** | **Initial\_days** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.00000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 | 9999.000000 |
| **mean** | 5000.000000 | 50162.829383 | 38.750924 | -91.244184 | 9962.09761 | 19.412574 | 5.012201 | 1.001500 | 0.398840 | 5891.257302 | 12934.657732 | 3.518752 | 3.506751 | 3.511151 | 3.515252 | 3.496950 | 3.522252 | 3.493949 | 3.509751 | 2.098660 | 53.296795 | 40479.829026 | 0.709214 | 0.322255 | 33.938594 |
| **std** | 2886.607005 | 27468.724995 | 5.403327 | 15.206357 | 14822.13967 | 6.723606 | 1.045787 | 1.008118 | 0.628507 | 3377.610179 | 6542.915985 | 1.032006 | 1.034864 | 1.032794 | 1.036223 | 1.030232 | 1.032130 | 1.021444 | 1.042352 | 1.855754 | 17.994487 | 24880.572722 | 0.431281 | 0.443767 | 24.852238 |
| **min** | 1.000000 | 610.000000 | 17.967190 | -174.209690 | 0.00000 | 9.519012 | 1.000000 | 0.000000 | 0.000000 | 1256.751699 | 3125.702716 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.000000 | 18.000000 | 154.080000 | 0.000000 | 0.000000 | 1.000000 |
| **25%** | 2500.500000 | 27600.500000 | 35.255120 | -97.353525 | 694.50000 | 16.513078 | 4.000000 | 0.000000 | 0.000000 | 3252.917043 | 7986.270943 | 3.000000 | 3.000000 | 3.000000 | 3.000000 | 3.000000 | 3.000000 | 3.000000 | 3.000000 | 1.000000 | 41.000000 | 23948.445000 | 0.000000 | 0.000000 | 8.000000 |
| **50%** | 5000.000000 | 50208.000000 | 39.419060 | -88.399240 | 2765.00000 | 18.080368 | 5.000000 | 1.000000 | 0.000000 | 5851.488972 | 11573.903260 | 4.000000 | 3.000000 | 4.000000 | 4.000000 | 3.000000 | 4.000000 | 3.000000 | 3.000000 | 2.058606 | 53.300937 | 40484.397511 | 1.000000 | 0.000000 | 34.000000 |
| **75%** | 7499.500000 | 72412.500000 | 42.044220 | -80.439205 | 13938.00000 | 19.789360 | 6.000000 | 2.000000 | 1.000000 | 7614.327715 | 15626.601495 | 4.000000 | 4.000000 | 4.000000 | 4.000000 | 4.000000 | 4.000000 | 4.000000 | 4.000000 | 3.000000 | 65.000000 | 46463.540000 | 1.000000 | 1.000000 | 59.000000 |
| **max** | 9999.000000 | 99929.000000 | 70.560990 | -65.290170 | 122814.00000 | 53.019124 | 9.000000 | 7.000000 | 5.000000 | 21524.224210 | 30566.073130 | 8.000000 | 7.000000 | 8.000000 | 7.000000 | 7.000000 | 7.000000 | 7.000000 | 7.000000 | 10.000000 | 89.000000 | 207249.130000 | 1.000000 | 1.000000 | 71.000000 |

In [6]:

*# Rename mis-spelled column*

df**.**rename(columns**=**{'Curteous':'Courteous'}, inplace**=True**, errors**=**'raise')

In [7]:

*# Reduce size of data set to include most relavent columns*

df2 **=**df[['ReAdmis', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'VitD\_supp',

'Soft\_drink', 'Initial\_admin', 'HighBlood', 'Stroke',

'Complication\_risk', 'Arthritis', 'Diabetes', 'Hyperlipidemia',

'BackPain', 'Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma',

'Services', 'TotalCharge', 'Additional\_charges', 'Admissions',

'Treatment', 'Visits', 'Reliability', 'Options', 'Hours', 'Courteous',

'Dr\_comm', 'Children', 'Age', 'Income', 'Overweight', 'Anxiety',

'Initial\_days', 'Population']]

df2**.**head()

Out[7]:

|  | **ReAdmis** | **VitD\_levels** | **Doc\_visits** | **Full\_meals\_eaten** | **VitD\_supp** | **Soft\_drink** | **Initial\_admin** | **HighBlood** | **Stroke** | **Complication\_risk** | **Arthritis** | **Diabetes** | **Hyperlipidemia** | **BackPain** | **Allergic\_rhinitis** | **Reflux\_esophagitis** | **Asthma** | **Services** | **TotalCharge** | **Additional\_charges** | **Admissions** | **Treatment** | **Visits** | **Reliability** | **Options** | **Hours** | **Courteous** | **Dr\_comm** | **Children** | **Age** | **Income** | **Overweight** | **Anxiety** | **Initial\_days** | **Population** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | No | 17.802330 | 6 | 0 | 0 | No | Emergency Admission | Yes | No | Medium | Yes | Yes | No | Yes | Yes | No | Yes | Blood Work | 3191.048774 | 17939.403420 | 3 | 3 | 2 | 2 | 4 | 3 | 3 | 4 | 3.000000 | 51.0 | 46805.99000 | 1.0 | 0.320908 | 15 | 2951 |
| **1** | No | 18.994640 | 4 | 2 | 1 | No | Emergency Admission | Yes | No | High | No | No | No | No | No | Yes | No | Intravenous | 4214.905346 | 17612.998120 | 3 | 4 | 3 | 4 | 4 | 4 | 3 | 3 | 3.000000 | 53.0 | 14370.14000 | 1.0 | 0.321309 | 4 | 11303 |
| **2** | No | 17.415889 | 4 | 1 | 0 | No | Elective Admission | Yes | No | Medium | No | Yes | No | No | No | No | No | Blood Work | 2177.586768 | 17505.192460 | 2 | 4 | 4 | 4 | 3 | 4 | 3 | 3 | 0.000000 | 78.0 | 39741.49000 | 0.0 | 0.322654 | 1 | 17125 |
| **3** | No | 17.420079 | 4 | 1 | 0 | No | Elective Admission | No | Yes | Medium | Yes | No | No | No | No | Yes | Yes | Blood Work | 2465.118965 | 12993.437350 | 3 | 5 | 5 | 3 | 4 | 5 | 5 | 5 | 2.035524 | 22.0 | 1209.56000 | 0.0 | 0.000000 | 1 | 2162 |
| **4** | No | 16.870524 | 5 | 0 | 2 | Yes | Elective Admission | No | No | Low | No | No | Yes | No | Yes | No | No | CT Scan | 1885.655137 | 3716.525786 | 2 | 1 | 3 | 3 | 5 | 3 | 4 | 3 | 2.079285 | 76.0 | 40484.44717 | 1.0 | 0.000000 | 5 | 5287 |

In [8]:

*# Create subset of the data*

df3 **=** df2[['Soft\_drink', 'HighBlood', 'Stroke', 'Arthritis', 'Diabetes',

'Hyperlipidemia', 'BackPain', 'Allergic\_rhinitis', 'Reflux\_esophagitis',

'Asthma', 'Initial\_admin', 'Complication\_risk', 'Services', 'Courteous'

, 'Initial\_days', 'Doc\_visits', 'Population', 'Treatment'

, 'Hours', 'Admissions', 'VitD\_levels','ReAdmis']]

df3**.**head()

Out[8]:

|  | **Soft\_drink** | **HighBlood** | **Stroke** | **Arthritis** | **Diabetes** | **Hyperlipidemia** | **BackPain** | **Allergic\_rhinitis** | **Reflux\_esophagitis** | **Asthma** | **Initial\_admin** | **Complication\_risk** | **Services** | **Courteous** | **Initial\_days** | **Doc\_visits** | **Population** | **Treatment** | **Hours** | **Admissions** | **VitD\_levels** | **ReAdmis** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | No | Yes | No | Yes | Yes | No | Yes | Yes | No | Yes | Emergency Admission | Medium | Blood Work | 3 | 15 | 6 | 2951 | 3 | 3 | 3 | 17.802330 | No |
| **1** | No | Yes | No | No | No | No | No | No | Yes | No | Emergency Admission | High | Intravenous | 3 | 4 | 4 | 11303 | 4 | 4 | 3 | 18.994640 | No |
| **2** | No | Yes | No | No | Yes | No | No | No | No | No | Elective Admission | Medium | Blood Work | 3 | 1 | 4 | 17125 | 4 | 4 | 2 | 17.415889 | No |
| **3** | No | No | Yes | Yes | No | No | No | No | Yes | Yes | Elective Admission | Medium | Blood Work | 5 | 1 | 4 | 2162 | 5 | 5 | 3 | 17.420079 | No |
| **4** | Yes | No | No | No | No | Yes | No | Yes | No | No | Elective Admission | Low | CT Scan | 4 | 5 | 5 | 5287 | 1 | 3 | 2 | 16.870524 | No |

In [9]:

*# Identify data types and whether Null values exist*

df3**.**info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 9999 entries, 0 to 9998

Data columns (total 22 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Soft\_drink 9999 non-null object

1 HighBlood 9999 non-null object

2 Stroke 9999 non-null object

3 Arthritis 9999 non-null object

4 Diabetes 9999 non-null object

5 Hyperlipidemia 9999 non-null object

6 BackPain 9999 non-null object

7 Allergic\_rhinitis 9999 non-null object

8 Reflux\_esophagitis 9999 non-null object

9 Asthma 9999 non-null object

10 Initial\_admin 9999 non-null object

11 Complication\_risk 9999 non-null object

12 Services 9999 non-null object

13 Courteous 9999 non-null int64

14 Initial\_days 9999 non-null int64

15 Doc\_visits 9999 non-null int64

16 Population 9999 non-null int64

17 Treatment 9999 non-null int64

18 Hours 9999 non-null int64

19 Admissions 9999 non-null int64

20 VitD\_levels 9999 non-null float64

21 ReAdmis 9999 non-null object

dtypes: float64(1), int64(7), object(14)

memory usage: 1.7+ MB

In [10]:

*# Encode binary data values*

df3['ReAdmis'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['ReAdmis']]

df3['Soft\_drink'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['Soft\_drink']]

df3['HighBlood'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['HighBlood']]

df3['Stroke'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['Stroke']]

df3['Arthritis'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['Arthritis']]

df3['Diabetes'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['Diabetes']]

df3['Hyperlipidemia'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['Hyperlipidemia']]

df3['BackPain'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['BackPain']]

df3['Allergic\_rhinitis'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['Allergic\_rhinitis']]

df3['Reflux\_esophagitis'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['Reflux\_esophagitis']]

df3['Asthma'] **=** [1 **if** value**==**'Yes' **else** 0 **for** value **in** df2['Asthma']]

*# Encode Categorical Data*

*# Instantiate LabelEncoder*

le **=** LabelEncoder()

df3**.**Initial\_admin **=** le**.**fit\_transform(df2**.**Initial\_admin)

*# 'Emergency Admission' = 1, 'Elective Admission' = 0, 'Observation Admission' = 3*

df3**.**Complication\_risk **=** le**.**fit\_transform(df2**.**Complication\_risk)

*# 'Medium' = 2, 'High'= 0, "low" = 1*

df3**.**Services **=** le**.**fit\_transform(df2**.**Services)

*# 'Blood Work' = 0, 'Intravenous'= 2, 'CT Scan' = 1, 'MRI' = 3*

df3**.**head()

Out[10]:

|  | **Soft\_drink** | **HighBlood** | **Stroke** | **Arthritis** | **Diabetes** | **Hyperlipidemia** | **BackPain** | **Allergic\_rhinitis** | **Reflux\_esophagitis** | **Asthma** | **Initial\_admin** | **Complication\_risk** | **Services** | **Courteous** | **Initial\_days** | **Doc\_visits** | **Population** | **Treatment** | **Hours** | **Admissions** | **VitD\_levels** | **ReAdmis** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 2 | 0 | 3 | 15 | 6 | 2951 | 3 | 3 | 3 | 17.802330 | 0 |
| **1** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 3 | 4 | 4 | 11303 | 4 | 4 | 3 | 18.994640 | 0 |
| **2** | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 4 | 17125 | 4 | 4 | 2 | 17.415889 | 0 |
| **3** | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 0 | 5 | 1 | 4 | 2162 | 5 | 5 | 3 | 17.420079 | 0 |
| **4** | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 4 | 5 | 5 | 5287 | 1 | 3 | 2 | 16.870524 | 0 |

In [11]:

*# Cross tabulation of discrete target and features*

contvars **=** ['VitD\_levels', 'Initial\_days']

**for** a **in** df3**.**columns:

**if** a **not** **in** contvars:

print(pd**.**crosstab(df**.**ReAdmis, df[a]), '\n', '--------------------------------')

*#table = pd.crosstab(df.ReAdmis, df.Services)*

*#table*

Soft\_drink No Yes

ReAdmis

No 5109 1222

Yes 2946 722

--------------------------------

HighBlood No Yes

ReAdmis

No 3747 2584

Yes 2162 1506

--------------------------------

Stroke No Yes

ReAdmis

No 5071 1260

Yes 2935 733

--------------------------------

Arthritis No Yes

ReAdmis

No 4086 2245

Yes 2340 1328

--------------------------------

Diabetes No Yes

ReAdmis

No 4591 1740

Yes 2670 998

--------------------------------

Hyperlipidemia No Yes

ReAdmis

No 4206 2125

Yes 2422 1246

--------------------------------

BackPain No Yes

ReAdmis

No 3758 2573

Yes 2127 1541

--------------------------------

Allergic\_rhinitis No Yes

ReAdmis

No 3825 2506

Yes 2234 1434

--------------------------------

Reflux\_esophagitis No Yes

ReAdmis

No 3726 2605

Yes 2138 1530

--------------------------------

Asthma No Yes

ReAdmis

No 4462 1869

Yes 2644 1024

--------------------------------

Initial\_admin Elective Admission Emergency Admission Observation Admission

ReAdmis

No 1608 3156 1567

Yes 896 1904 868

--------------------------------

Complication\_risk High Low Medium

ReAdmis

No 2135 1343 2853

Yes 1223 781 1664

--------------------------------

Services Blood Work CT Scan Intravenous MRI

ReAdmis

No 3335 737 2027 232

Yes 1929 488 1103 148

--------------------------------

Courteous 1 2 3 4 5 6 7

ReAdmis

No 136 830 2184 2243 808 125 5

Yes 79 515 1272 1243 466 87 6

--------------------------------

Doc\_visits 1 2 3 4 5 6 7 8 9

ReAdmis

No 4 36 375 1511 2416 1558 392 37 2

Yes 2 22 220 874 1406 878 242 24 0

--------------------------------

Population 0 1 4 5 6 7 8 9 \

ReAdmis

No 69 2 1 2 1 2 1 2

Yes 40 0 0 3 4 0 0 1

Population 10 11 12 13 14 15 16 17 \

ReAdmis

No 2 3 5 0 5 2 1 1

Yes 0 1 0 2 3 0 1 1

Population 18 19 20 21 22 24 25 26 \

ReAdmis

No 2 6 1 3 0 2 4 2

Yes 1 1 3 3 3 2 2 2

Population 27 28 29 30 31 32 33 34 \

ReAdmis

No 1 1 1 0 1 2 2 8

Yes 0 1 2 1 1 2 3 1

Population 35 36 37 38 39 40 41 42 \

ReAdmis

No 2 3 2 6 2 3 3 2

Yes 1 2 0 1 4 4 1 2

Population 43 44 45 46 47 48 49 50 \

ReAdmis

No 3 1 3 6 3 4 2 1

Yes 2 2 2 1 0 0 3 0

Population 51 52 53 54 55 56 57 58 \

ReAdmis

No 3 3 5 2 1 0 1 2

Yes 2 1 3 0 1 2 1 1

Population 59 60 61 63 64 65 66 67 \

ReAdmis

No 2 0 0 5 4 3 3 4

Yes 0 1 4 1 2 0 1 0

Population 68 69 70 71 72 73 74 75 \

ReAdmis

No 2 8 1 1 1 1 3 1

Yes 1 2 0 2 0 1 2 2

Population 76 77 78 79 80 81 82 83 \

ReAdmis

No 1 3 4 2 4 5 2 2

Yes 1 3 1 4 6 1 1 0

Population 84 85 86 87 88 89 90 91 \

ReAdmis

No 2 1 4 2 3 4 1 2

Yes 2 2 3 0 3 0 1 3

Population 92 93 94 95 96 97 98 99 \

ReAdmis

No 5 3 3 4 3 2 3 2

Yes 5 4 1 2 1 3 0 1

Population 100 101 102 104 105 106 107 108 \

ReAdmis

No 1 1 6 1 5 0 4 2

Yes 1 5 0 2 3 2 2 1

Population 109 110 111 112 113 114 115 116 \

ReAdmis

No 2 3 3 1 3 2 4 1

Yes 2 3 2 1 1 3 7 0

Population 117 118 119 120 121 122 123 124 \

ReAdmis

No 1 2 2 2 4 3 2 1

Yes 2 1 3 1 1 3 2 2

Population 125 126 127 128 129 130 131 133 \

ReAdmis

No 0 3 2 2 1 2 2 1

Yes 1 1 0 2 1 1 2 2

Population 134 135 136 137 138 139 140 141 \

ReAdmis

No 3 1 2 5 4 3 7 3

Yes 0 1 0 2 2 1 1 1

Population 142 143 144 145 146 147 148 149 \

ReAdmis

No 4 1 0 6 5 3 8 0

Yes 1 4 2 1 2 0 1 1

Population 150 151 152 153 154 155 156 157 \

ReAdmis

No 6 5 4 1 4 6 2 1

Yes 1 1 2 0 4 0 0 0

Population 158 159 160 161 162 163 164 165 \

ReAdmis

No 1 3 4 4 4 2 5 4

Yes 0 3 1 2 1 2 1 3

Population 166 167 168 169 170 171 172 173 \

ReAdmis

No 4 3 4 2 2 2 3 4

Yes 4 0 0 0 0 4 0 3

Population 174 175 176 177 178 179 180 181 \

ReAdmis

No 3 3 2 2 5 0 1 5

Yes 1 3 4 2 6 1 1 1

Population 182 184 185 186 187 188 189 190 \

ReAdmis

No 3 1 3 0 3 4 5 4

Yes 3 1 2 3 0 0 2 4

Population 191 192 193 194 195 196 197 198 \

ReAdmis

No 2 2 5 2 10 2 0 2

Yes 1 0 2 4 4 1 2 0

Population 199 200 201 202 203 204 205 206 \

ReAdmis

No 4 3 3 4 2 3 6 2

Yes 1 1 1 3 2 3 3 3

Population 207 208 209 210 211 212 213 214 \

ReAdmis

No 2 2 3 1 3 2 2 4

Yes 0 1 3 1 2 3 1 2

Population 215 217 218 219 220 221 222 223 \

ReAdmis

No 5 4 2 3 2 4 2 1

Yes 2 1 1 3 0 2 4 3

Population 224 225 226 227 228 229 230 231 \

ReAdmis

No 0 2 4 4 1 2 4 5

Yes 3 1 1 3 2 1 1 0

Population 232 233 234 235 236 237 238 239 \

ReAdmis

No 1 2 4 4 3 4 2 1

Yes 2 1 0 1 2 3 1 1

Population 240 241 242 243 245 246 247 248 \

ReAdmis

No 1 2 1 0 5 7 3 1

Yes 0 2 2 3 1 3 2 1

Population 249 250 251 252 253 254 255 256 \

ReAdmis

No 4 5 4 4 0 1 6 1

Yes 0 3 0 1 2 0 2 3

Population 257 258 259 260 261 262 263 264 \

ReAdmis

No 4 6 1 2 2 2 2 5

Yes 1 1 2 2 1 0 4 2

Population 265 266 267 268 269 270 271 272 \

ReAdmis

No 1 1 5 1 3 1 5 2

Yes 0 1 0 0 2 0 2 1

Population 273 274 275 276 277 278 279 280 \

ReAdmis

No 4 7 4 2 4 1 4 4

Yes 2 1 3 0 1 0 0 1

Population 281 282 283 284 285 286 287 288 \

ReAdmis

No 4 3 2 2 9 2 2 1

Yes 1 1 3 0 2 2 1 1

Population 289 290 291 292 293 294 295 296 \

ReAdmis

No 4 6 2 2 5 6 2 1

Yes 0 0 3 3 1 4 5 0

Population 297 298 299 300 301 302 303 304 \

ReAdmis

No 4 2 3 4 5 2 4 5

Yes 1 3 1 0 3 1 2 2

Population 305 306 307 308 309 310 311 312 \

ReAdmis

No 1 2 3 7 5 1 0 6

Yes 1 2 0 2 0 0 1 1

Population 313 314 315 316 318 319 320 321 \

ReAdmis

No 2 1 2 5 1 2 3 2

Yes 1 1 2 1 1 1 0 2

Population 322 323 324 325 327 328 329 330 \

ReAdmis

No 1 3 3 1 1 2 1 4

Yes 2 1 0 3 0 2 0 3

Population 331 332 333 334 335 336 337 338 \

ReAdmis

No 1 3 5 2 2 3 3 3

Yes 0 1 1 1 3 3 1 1

Population 339 340 341 342 343 344 345 346 \

ReAdmis

No 1 3 5 3 4 3 3 6

Yes 0 1 1 4 1 1 2 4

Population 347 348 349 350 351 352 353 354 \

ReAdmis

No 0 2 1 0 1 0 4 2

Yes 2 2 1 1 0 1 2 1

Population 355 356 357 358 359 360 361 362 \

ReAdmis

No 4 2 1 3 5 2 2 0

Yes 0 1 3 0 4 2 4 1

Population 363 364 365 366 367 368 369 370 \

ReAdmis

No 3 4 0 4 1 2 3 7

Yes 1 2 1 1 1 2 1 0

Population 371 372 373 375 376 377 378 379 \

ReAdmis

No 1 2 3 3 2 1 1 1

Yes 0 1 0 2 3 0 0 1

Population 380 381 382 383 384 385 386 387 \

ReAdmis

No 1 2 2 2 1 3 4 2

Yes 0 1 0 3 2 3 1 0

Population 388 389 390 391 392 393 394 395 \

ReAdmis

No 1 3 1 2 0 4 1 2

Yes 1 0 0 2 1 0 4 5

Population 396 397 398 399 401 402 403 404 \

ReAdmis

No 1 4 3 3 2 4 5 2

Yes 0 0 2 1 0 2 1 1

Population 405 406 407 408 409 410 411 412 \

ReAdmis

No 2 1 1 2 2 1 1 1

Yes 0 1 2 2 1 1 1 0

Population 413 414 415 416 417 420 421 422 \

ReAdmis

No 1 1 2 4 2 5 3 3

Yes 1 1 1 0 0 0 1 3

Population 423 424 425 426 427 429 430 431 \

ReAdmis

No 7 2 3 5 1 1 3 1

Yes 2 2 0 0 2 4 0 0

Population 432 433 434 436 437 438 439 440 \

ReAdmis

No 3 0 1 1 1 1 2 3

Yes 1 1 2 0 0 1 1 2

Population 441 442 443 444 445 446 447 448 \

ReAdmis

No 6 2 1 1 3 1 1 1

Yes 1 0 0 3 3 1 5 0

Population 450 451 452 453 454 455 456 459 \

ReAdmis

No 1 3 2 2 3 3 1 2

Yes 1 2 1 1 1 0 0 1

Population 460 461 462 463 464 465 466 467 \

ReAdmis

No 1 2 2 3 1 3 3 5

Yes 1 1 0 1 1 1 1 0

Population 468 469 470 471 473 474 475 476 \

ReAdmis

No 0 4 1 7 3 0 3 1

Yes 1 1 2 2 2 1 2 0

Population 478 479 480 481 482 483 484 486 \

ReAdmis

No 1 2 1 3 2 3 3 1

Yes 1 0 2 1 1 1 1 1

Population 487 488 489 490 491 492 493 494 \

ReAdmis

No 1 1 1 2 1 4 1 2

Yes 0 1 0 2 0 1 0 1

Population 495 496 497 498 499 501 502 503 \

ReAdmis

No 4 5 2 2 2 1 2 1

Yes 2 3 1 1 0 1 0 2

Population 504 505 506 507 508 509 510 511 \

ReAdmis

No 1 0 1 3 2 2 1 4

Yes 0 1 0 1 2 0 1 1

Population 512 513 514 515 516 517 518 519 \

ReAdmis

No 2 2 0 2 3 6 0 5

Yes 0 1 2 0 0 1 1 0

Population 520 522 523 524 ... 39803 39828 39841 \

ReAdmis ...

No 2 1 1 1 ... 1 0 1

Yes 2 0 1 1 ... 0 1 0

Population 39861 39902 39946 39949 39985 39994 40003 40014 \

ReAdmis

No 1 0 2 1 1 1 1 0

Yes 0 2 0 0 0 0 1 1

Population 40019 40029 40051 40056 40063 40071 40173 40186 \

ReAdmis

No 0 1 1 0 1 0 1 1

Yes 1 0 0 1 0 1 0 0

Population 40209 40266 40272 40305 40340 40376 40402 40434 \

ReAdmis

No 1 1 1 1 1 1 0 1

Yes 0 0 0 0 0 1 1 0

Population 40437 40441 40447 40490 40511 40532 40603 40615 \

ReAdmis

No 1 1 1 1 1 1 0 0

Yes 0 0 0 0 0 0 1 1

Population 40724 40733 40737 40766 40773 40778 40824 40830 \

ReAdmis

No 1 1 2 1 0 1 1 2

Yes 0 0 0 0 1 0 0 0

Population 40837 40905 40926 41000 41002 41023 41059 41107 \

ReAdmis

No 1 1 0 1 0 0 0 0

Yes 0 0 1 0 1 1 1 1

Population 41113 41173 41205 41243 41277 41297 41346 41440 \

ReAdmis

No 1 1 1 1 1 1 1 1

Yes 0 2 0 0 0 0 0 0

Population 41449 41451 41478 41479 41494 41503 41510 41539 \

ReAdmis

No 0 1 1 0 1 2 0 1

Yes 1 0 0 1 0 0 1 0

Population 41550 41551 41556 41669 41671 41698 41716 41839 \

ReAdmis

No 2 0 0 0 0 1 0 1

Yes 0 1 1 1 1 0 1 1

Population 41877 41938 41942 41973 42028 42045 42063 42107 \

ReAdmis

No 1 1 1 2 1 1 1 1

Yes 0 0 1 0 0 0 1 0

Population 42171 42197 42231 42234 42259 42283 42397 42442 \

ReAdmis

No 1 0 1 1 0 1 1 1

Yes 0 1 0 0 1 0 0 0

Population 42471 42514 42553 42583 42589 42602 42614 42623 \

ReAdmis

No 1 1 1 0 1 1 0 1

Yes 0 0 0 1 0 0 1 0

Population 42653 42654 42672 42737 42776 42780 42813 42817 \

ReAdmis

No 0 1 1 0 1 1 0 1

Yes 1 0 1 1 0 0 1 1

Population 42821 42882 42916 42930 42933 42960 43019 43055 \

ReAdmis

No 1 1 1 1 2 1 0 0

Yes 0 0 0 0 0 0 1 1

Population 43103 43110 43119 43132 43155 43208 43232 43374 \

ReAdmis

No 2 1 0 0 1 1 0 1

Yes 0 0 1 1 0 0 1 0

Population 43408 43458 43498 43607 43634 43641 43652 43687 \

ReAdmis

No 1 0 0 1 1 1 0 1

Yes 0 1 1 0 0 0 1 0

Population 43703 43706 43736 43801 43960 44036 44103 44205 \

ReAdmis

No 0 1 0 1 1 1 0 1

Yes 1 0 2 0 0 0 1 0

Population 44233 44239 44272 44294 44328 44335 44352 44380 \

ReAdmis

No 1 1 1 0 1 1 0 2

Yes 0 0 0 1 0 0 1 0

Population 44406 44571 44680 44740 44750 44759 44824 44828 \

ReAdmis

No 1 1 1 1 0 1 1 0

Yes 0 0 0 0 1 0 0 1

Population 44870 44908 44994 45202 45282 45411 45514 45535 \

ReAdmis

No 1 1 1 1 1 1 2 1

Yes 0 0 0 0 1 0 0 0

Population 45596 45694 45697 45700 45736 45827 45851 45881 \

ReAdmis

No 0 2 0 0 1 1 1 1

Yes 1 0 1 1 0 1 0 0

Population 45910 45952 45958 45963 45995 46008 46058 46064 \

ReAdmis

No 2 0 0 1 1 1 1 0

Yes 0 1 1 2 0 0 0 2

Population 46165 46174 46215 46261 46276 46299 46369 46418 \

ReAdmis

No 1 1 1 1 2 1 0 1

Yes 0 0 0 0 0 0 1 0

Population 46442 46470 46545 46604 46647 46675 46775 46841 \

ReAdmis

No 1 0 0 0 1 0 1 0

Yes 0 1 1 1 0 1 0 1

Population 46875 46888 47021 47084 47227 47231 47247 47261 \

ReAdmis

No 1 1 1 1 0 2 1 1

Yes 1 0 0 0 1 0 0 0

Population 47419 47442 47576 47612 47628 47692 47834 47887 \

ReAdmis

No 0 1 1 0 2 1 1 1

Yes 1 1 0 1 0 0 0 0

Population 48036 48110 48211 48351 48362 48436 48438 48443 \

ReAdmis

No 2 0 1 1 1 1 1 2

Yes 0 1 0 0 0 0 1 0

Population 48483 48509 48560 48810 48827 48833 48853 48854 \

ReAdmis

No 1 1 0 0 1 0 1 1

Yes 0 0 1 1 0 1 0 0

Population 48861 49034 49158 49172 49194 49232 49312 49492 \

ReAdmis

No 1 1 0 0 0 0 1 1

Yes 0 1 1 1 1 1 0 0

Population 49508 49536 49556 49583 49657 49664 49699 49860 \

ReAdmis

No 1 0 1 0 0 2 0 1

Yes 0 1 0 1 1 0 1 0

Population 49907 49934 50019 50127 50348 50436 50488 50541 \

ReAdmis

No 1 0 0 0 1 1 1 1

Yes 0 1 1 1 0 1 0 0

Population 50675 50782 50804 50895 50927 50953 50972 50992 \

ReAdmis

No 0 1 0 1 1 1 1 0

Yes 2 0 1 0 0 0 0 1

Population 50997 51066 51082 51130 51230 51250 51298 51333 \

ReAdmis

No 1 0 1 1 1 0 0 0

Yes 0 1 0 0 0 1 1 1

Population 51378 51419 51447 51459 51496 51568 51652 51667 \

ReAdmis

No 0 1 1 1 0 1 0 1

Yes 1 0 0 1 1 0 1 0

Population 51984 52088 52117 52246 52272 52454 52484 52564 \

ReAdmis

No 1 0 1 1 1 0 1 1

Yes 0 1 0 0 0 1 0 0

Population 52587 52688 52785 52833 53142 53162 53210 53211 \

ReAdmis

No 1 0 0 0 0 1 1 2

Yes 0 1 1 1 1 0 0 0

Population 53239 53256 53277 53328 53357 53461 53520 53547 \

ReAdmis

No 1 0 0 2 0 1 1 0

Yes 0 1 1 0 1 0 1 1

Population 53552 53565 53582 53688 53772 53845 53866 53886 \

ReAdmis

No 0 0 0 1 0 1 1 1

Yes 1 1 1 0 1 0 0 0

Population 53910 53933 53976 54035 54216 54311 54413 54453 \

ReAdmis

No 1 1 1 0 1 0 1 1

Yes 0 1 0 1 0 1 0 0

Population 54460 54507 54647 54701 54776 54823 54862 54882 \

ReAdmis

No 1 0 2 1 1 1 0 2

Yes 0 1 0 0 0 0 1 0

Population 54975 55061 55107 55130 55200 55222 55269 55316 \

ReAdmis

No 1 1 0 1 1 0 1 1

Yes 0 0 1 0 0 1 0 0

Population 55361 55519 55548 55663 55857 55937 56000 56155 \

ReAdmis

No 1 1 1 0 1 1 1 0

Yes 0 0 0 2 0 1 0 1

Population 56173 56420 56446 56458 56496 56511 56614 56645 \

ReAdmis

No 1 1 1 1 1 0 0 1

Yes 1 0 0 0 0 1 1 0

Population 56651 56716 56765 56790 56811 56838 56870 56931 \

ReAdmis

No 0 1 0 2 0 0 1 0

Yes 1 0 1 0 1 1 0 1

Population 56964 57000 57035 57181 57226 57344 57369 57642 \

ReAdmis

No 1 0 1 1 0 1 0 0

Yes 0 1 0 0 1 1 1 1

Population 57659 57775 57925 57955 58160 58167 58178 58252 \

ReAdmis

No 0 1 1 1 0 0 0 1

Yes 1 2 0 0 1 1 1 0

Population 58304 58327 58431 58541 58882 58893 59001 59129 \

ReAdmis

No 1 1 0 2 1 1 0 2

Yes 0 0 2 0 0 0 1 0

Population 59191 59256 59502 59633 59637 59699 60033 60081 \

ReAdmis

No 1 1 0 1 2 1 0 0

Yes 0 0 1 0 0 0 1 1

Population 60107 60138 60191 60238 60254 61301 61324 61550 \

ReAdmis

No 1 1 1 1 1 0 1 0

Yes 0 0 0 0 0 1 0 2

Population 61660 61803 61904 62226 62348 62392 62430 62718 \

ReAdmis

No 1 0 1 1 1 0 1 2

Yes 1 1 1 0 0 2 0 0

Population 63078 63153 63177 63380 63425 63852 64034 64126 \

ReAdmis

No 1 1 1 1 0 0 1 1

Yes 0 0 0 0 1 1 0 0

Population 64197 64264 64417 64548 64723 65123 65151 65247 \

ReAdmis

No 1 0 0 1 0 0 1 1

Yes 0 1 1 0 1 1 1 0

Population 65256 65300 65335 65347 65603 65742 65791 65885 \

ReAdmis

No 0 0 0 0 0 1 0 1

Yes 1 1 1 1 1 0 1 0

Population 66134 66245 66295 66590 67040 67053 67094 67339 \

ReAdmis

No 1 0 1 0 1 1 0 0

Yes 0 1 1 1 0 0 1 1

Population 67597 67730 67959 68102 68201 68263 68646 68773 \

ReAdmis

No 2 1 0 0 1 1 1 1

Yes 1 0 1 1 0 0 0 1

Population 68796 68803 68807 69413 69546 69685 69802 70052 \

ReAdmis

No 1 0 1 1 0 1 2 1

Yes 0 1 0 0 1 0 0 0

Population 70199 71023 71308 71581 71587 72332 72799 75666 \

ReAdmis

No 1 0 1 1 1 0 1 1

Yes 0 1 0 0 0 1 0 0

Population 75899 76177 76495 76882 77334 78475 78767 78900 \

ReAdmis

No 0 0 1 1 1 0 0 0

Yes 1 1 0 0 0 1 1 1

Population 79010 79298 79699 80143 80213 80264 80524 82534 \

ReAdmis

No 1 1 0 1 0 0 0 0

Yes 0 0 1 0 1 1 1 1

Population 82572 82920 83179 83960 84418 85166 87027 87509 \

ReAdmis

No 1 1 1 2 2 1 0 0

Yes 0 0 0 1 0 0 1 1

Population 87986 88349 90036 90340 90761 92108 92146 94512 \

ReAdmis

No 1 1 0 0 1 1 0 1

Yes 0 1 1 1 1 0 1 0

Population 95666 96081 96487 96530 98219 102624 105799 107700 \

ReAdmis

No 0 1 0 0 0 1 1 0

Yes 1 0 1 2 1 0 1 1

Population 122814

ReAdmis

No 1

Yes 1

[2 rows x 5950 columns]

--------------------------------

Treatment 1 2 3 4 5 6 7

ReAdmis

No 150 848 2163 2115 917 133 5

Yes 63 512 1275 1236 504 71 7

--------------------------------

Hours 1 2 3 4 5 6 7

ReAdmis

No 130 829 2120 2192 899 153 8

Yes 83 490 1251 1272 504 66 2

--------------------------------

Admissions 1 2 3 4 5 6 7 8

ReAdmis

No 134 823 2137 2184 891 155 7 0

Yes 79 492 1267 1270 486 70 3 1

--------------------------------

ReAdmis No Yes

ReAdmis

No 6331 0

Yes 0 3668

--------------------------------

In [12]:

*# Create data frame containing discrete variables*

disc\_df **=** df2[['ReAdmis', 'Doc\_visits', 'Full\_meals\_eaten', 'VitD\_supp',

'Soft\_drink', 'Initial\_admin', 'HighBlood', 'Stroke',

'Complication\_risk', 'Arthritis', 'Diabetes', 'Hyperlipidemia',

'BackPain', 'Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma',

'Services','Admissions',

'Treatment', 'Visits', 'Reliability', 'Options', 'Hours', 'Courteous',

'Dr\_comm','Overweight', 'Initial\_days']]

disc\_df**.**head()

Out[12]:

|  | **ReAdmis** | **Doc\_visits** | **Full\_meals\_eaten** | **VitD\_supp** | **Soft\_drink** | **Initial\_admin** | **HighBlood** | **Stroke** | **Complication\_risk** | **Arthritis** | **Diabetes** | **Hyperlipidemia** | **BackPain** | **Allergic\_rhinitis** | **Reflux\_esophagitis** | **Asthma** | **Services** | **Admissions** | **Treatment** | **Visits** | **Reliability** | **Options** | **Hours** | **Courteous** | **Dr\_comm** | **Overweight** | **Initial\_days** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | No | 6 | 0 | 0 | No | Emergency Admission | Yes | No | Medium | Yes | Yes | No | Yes | Yes | No | Yes | Blood Work | 3 | 3 | 2 | 2 | 4 | 3 | 3 | 4 | 1.0 | 15 |
| **1** | No | 4 | 2 | 1 | No | Emergency Admission | Yes | No | High | No | No | No | No | No | Yes | No | Intravenous | 3 | 4 | 3 | 4 | 4 | 4 | 3 | 3 | 1.0 | 4 |
| **2** | No | 4 | 1 | 0 | No | Elective Admission | Yes | No | Medium | No | Yes | No | No | No | No | No | Blood Work | 2 | 4 | 4 | 4 | 3 | 4 | 3 | 3 | 0.0 | 1 |
| **3** | No | 4 | 1 | 0 | No | Elective Admission | No | Yes | Medium | Yes | No | No | No | No | Yes | Yes | Blood Work | 3 | 5 | 5 | 3 | 4 | 5 | 5 | 5 | 0.0 | 1 |
| **4** | No | 5 | 0 | 2 | Yes | Elective Admission | No | No | Low | No | No | Yes | No | Yes | No | No | CT Scan | 2 | 1 | 3 | 3 | 5 | 3 | 4 | 3 | 1.0 | 5 |

In [13]:

*# Countplot of discrete target and features*

**for** a **in** disc\_df**.**columns:

print(sns**.**countplot(data **=** df2

,x **=** df2[a]

),

plt**.**show(),'\n', '--------------------------------')

A picture containing bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

A picture containing logo

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

A picture containing logo

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

A picture containing chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

A picture containing bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

A picture containing logo

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

A picture containing logo

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

A picture containing logo

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart

Description automatically generated with medium confidence

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, bar chart

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Shape, square

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

Chart, histogram

Description automatically generated

AxesSubplot(0.125,0.125;0.775x0.755) None

--------------------------------

In [ ]:

In [14]:

*#Boxplot of discrete target and features*

*# Box plot*

ax **=** sns**.**boxplot(data**=**disc\_df,orient**=**"h", palette**=**"cool\_r")

*#\*\*\* convert to numbers*

Text

Description automatically generated with medium confidence

In [15]:

*# Correlation table of target and features*

corrmat **=** df3**.**corr()

corrmat

Graphical user interface, application, table, Excel

Description automatically generated

In [16]:

*# Heatmap of discrete target and features*

heatmap **=** sns**.**heatmap(data**=**corrmat, cmap**=**"plasma", center **=** 0 ,

linewidths **=** 3, linecolor **=** "Black",

xticklabels **=** **True**, yticklabels **=** **True**)

plt**.**xticks(rotation**=**90)

plt**.**title('Heatmap of discrete target and features')

Out[16]:

Text(0.5, 1.0, 'Heatmap of discrete target and features')

A picture containing nature, light, bright

Description automatically generated

In [17]:

*#C2 - Variables to be used with Datatypes*

dataset **=**df3[['Initial\_days', 'Population','Treatment','Doc\_visits', 'VitD\_levels', 'ReAdmis']]

dataset**.**head()

Out[17]:

|  | **Initial\_days** | **Population** | **Treatment** | **Doc\_visits** | **VitD\_levels** | **ReAdmis** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 15 | 2951 | 3 | 6 | 17.802330 | 0 |
| **1** | 4 | 11303 | 4 | 4 | 18.994640 | 0 |
| **2** | 1 | 17125 | 4 | 4 | 17.415889 | 0 |
| **3** | 1 | 2162 | 5 | 4 | 17.420079 | 0 |
| **4** | 5 | 5287 | 1 | 5 | 16.870524 | 0 |

In [18]:

*# Check for null values*

df3**.**isna()**.**sum()

Out[18]:

Soft\_drink 0

HighBlood 0

Stroke 0

Arthritis 0

Diabetes 0

Hyperlipidemia 0

BackPain 0

Allergic\_rhinitis 0

Reflux\_esophagitis 0

Asthma 0

Initial\_admin 0

Complication\_risk 0

Services 0

Courteous 0

Initial\_days 0

Doc\_visits 0

Population 0

Treatment 0

Hours 0

Admissions 0

VitD\_levels 0

ReAdmis 0

dtype: int64

In [19]:

*# C4. Export clean data set*

df3**.**to\_csv("C:\\Users\\smith\\OneDrive\\Jason\\wgu\\Assignments\\D209 Data Mining 1\\Code\\nvm2.csv")

In [20]:

*# Part IV: Analysis*

In [21]:

*# ID data frame to be used in analysis*

print("Dataset DF:")

print(dataset**.**head())

print("\n # of Rows, columns:", dataset**.**shape)

Dataset DF:

Initial\_days Population Treatment Doc\_visits VitD\_levels ReAdmis

0 15 2951 3 6 17.802330 0

1 4 11303 4 4 18.994640 0

2 1 17125 4 4 17.415889 0

3 1 2162 5 4 17.420079 0

4 5 5287 1 5 16.870524 0

# of Rows, columns: (9999, 6)

In [22]:

*# split the data into inputs and outputs*

X **=** dataset**.**iloc[:, [0,4]]**.**values

y **=** dataset**.**iloc[:,5]**.**values

In [23]:

*# D1. Split the data into training and test data sets and provide the file(s).*

*# training and testing data*

**from** sklearn.model\_selection **import** train\_test\_split

*# assign train 75% and test data size 25%*

X\_train, X\_test, y\_train, y\_test **=**train\_test\_split(X,y,test\_size**=** 0.25, random\_state**=**0)

*# Create data frames*

xtrain\_df **=** pd**.**DataFrame(X\_train)

xtest\_df **=** pd**.**DataFrame(X\_test)

ytrain\_df **=** pd**.**DataFrame(y\_train)

ytest\_df **=** pd**.**DataFrame(y\_test)

*# # Save to CSV*

xtrain\_df**.**to\_csv("C:\\Users\\smith\\OneDrive\\Jason\\wgu\\Assignments\\D209 Data Mining 1\\Code\\xtrain.csv")

xtest\_df**.**to\_csv("C:\\Users\\smith\\OneDrive\\Jason\\wgu\\Assignments\\D209 Data Mining 1\\Code\\xtest.csv")

ytrain\_df**.**to\_csv("C:\\Users\\smith\\OneDrive\\Jason\\wgu\\Assignments\\D209 Data Mining 1\\Code\\ytrain.csv")

ytest\_df**.**to\_csv("C:\\Users\\smith\\OneDrive\\Jason\\wgu\\Assignments\\D209 Data Mining 1\\Code\\ytest.csv")

In [24]:

*# D2. Describe the analysis technique you used to appropriately analyze the data. Include screenshots of the intermediate calculations you performed.*

In [25]:

*# D3. Classification Analysis using KNN*

*# Scaling the input data*

sc **=** StandardScaler()

*# Scaling training and testing data set*

X\_train **=** sc**.**fit\_transform(X\_train)

X\_test **=** sc**.**transform(X\_test)

In [26]:

*# applying standard scale method*

**from** sklearn.preprocessing **import** StandardScaler

sc **=** StandardScaler()

*# Scaling training and testing data set*

X\_train **=** sc**.**fit\_transform(X\_train)

X\_test **=** sc**.**transform(X\_test)

In [27]:

*# Run KNN with an inital guess at the K value and an ideal K-value*

*# Initial K-value set to be 3*

*# classifer = KNeighborsClassifier(n\_neighbors=5 ) # 0.81*

classifer **=** KNeighborsClassifier(n\_neighbors**=**15 ) *# 0.0.84*

*# model training*

classifer**.**fit(X\_train,y\_train)

*# testing the model*

y\_pred**=** classifer**.**predict(X\_test)

In [28]:

*# Determing KNN accuracy based on the current K-value*

print(accuracy\_score(y\_test,y\_pred))

0.8364

In [29]:

*# Determing KNN accuracy based on the current K-value*

*#from sklearn.metrics import classification\_report*

target\_names **=** ['Readmitted', 'Not Readmitted']

print(classification\_report(y\_test, y\_pred, target\_names**=**target\_names))

precision recall f1-score support

Readmitted 0.93 0.81 0.86 1595

Not Readmitted 0.72 0.89 0.80 905

accuracy 0.84 2500

macro avg 0.83 0.85 0.83 2500

weighted avg 0.85 0.84 0.84 2500

In [30]:

*# Use error curves to choose an ideal K-value (Tarek Atwan, P.1)*

*# creating sets for errors*

error1**=** []

error2**=** []

*# for loop*

**for** k **in** range(1,50):

*# using KNN algorithm*

knn **=** KNeighborsClassifier(n\_neighbors**=**k)

knn**.**fit(X\_train,y\_train)

y\_pred1 **=** knn**.**predict(X\_train)

*# stroring the errors*

error1**.**append(np**.**mean(y\_train**!=** y\_pred1))

y\_pred2 **=** knn**.**predict(X\_test)

error2**.**append(np**.**mean(y\_test **!=** y\_pred2))

*# ploting the graphs for testing and training*

plt**.**plot(range(1,50), error1, label**=**"train")

plt**.**plot(range(1,50), error2, label**=**"test")

plt**.**title('Error Rate vs. K-Values')

plt**.**xlabel('k Value')

plt**.**ylabel('Error')

plt**.**legend()

Out[30]:

<matplotlib.legend.Legend at 0x199ebb65460>

A picture containing chart

Description automatically generated

In [31]:

*#Plotting error rate*

*# import numpy*

**import** numpy **as** np

error\_rate **=** []

*# searching k value upto 40*

**for** i **in** range(1,50):

*# knn algorithm*

knn **=** KNeighborsClassifier(n\_neighbors**=**i)

knn**.**fit(X\_train, y\_train)

*# testing the model*

pred\_i **=** knn**.**predict(X\_test)

error\_rate**.**append(np**.**mean(pred\_i **!=** y\_test))

*# Configure and plot error rate over k values*

plt**.**figure(figsize**=**(10,4))

plt**.**plot(range(1,50), error\_rate, color**=**'blue', linestyle**=**'dashed', marker**=**'o', markerfacecolor**=**'red', markersize**=**10)

plt**.**title('Error Rate vs. K-Values')

plt**.**xlabel('K-Values')

plt**.**ylabel('Error Rate')

Out[31]:

Text(0, 0.5, 'Error Rate')

Shape

Description automatically generated

In [32]:

*# Evaluating KNN algorithm performance*

*# Confusion Matrix*

*# # importing seaborn*

*# import seaborn as sns*

*# # Making the Confusion Matrix*

*# from sklearn.metrics import confusion\_matrix*

*# providing actual and predicted values*

cm **=** confusion\_matrix(y\_test, y\_pred)

*# If True, write the data value in each cell*

sns**.**heatmap(cm,annot**=True**)

*# formatting*

plt**.**clf()

plt**.**imshow(cm,interpolation**=**'nearest', cmap**=**plt**.**cm**.**Wistia)

plt**.**ylabel('Predicted Values')

plt**.**xlabel('Actual Values')

classNames **=** ['Positive', 'Negative']

tick\_marks **=** np**.**arange(len(classNames))

plt**.**xticks(tick\_marks, classNames, rotation**=**45)

plt**.**yticks(tick\_marks, classNames)

plt**.**title('Confusion Matrix')

s **=** [['TP','FP' ], ['FN','TN']]

**for** i **in** range(2):

**for** j **in** range(2):

plt**.**text(j,i, str(s[i][j])**+**" = "**+**str(cm[i][j]))

*# saving confusion matrix in png form*

plt**.**savefig('confusion\_Matrix.png')

*#print(cm)*

plt**.**show()

Chart, treemap chart

Description automatically generated