Performance Assessment 2 – Revision 1

D208 – Predictive Modeling

Jessica Hosey

MSDA, College of Information Technology

Western Governors University

June 24th, 2024

Part I: Research Question

A.1. Summarize one research question.

What factors are connected to a patient's choice of drinking soft drinks? Essentially, does drinking soft drinks like Dr. Pepper or Diet Coke cause you to have other symptoms/diseases (have a stroke, high blood pressure, overweight, diabetes, Hyperlipidemia, etc.) that might be life-threatening?

A.2. Define the goals of the data analysis.

Most people in America drink soda soft drinks (I love Diet Coke for me). The goal of this analysis is to see if the choice of drinking a soda is directly related to having health issues that lead to hospital visits. Therefore, if you drink soft drinks, you are or are not more likely to have a stroke, high blood pressure, be overweight, have diabetes, have Hyperlipidemia, etc. This information may be available to the public if there are direct connections to drinking soft drinks. Without direct connections, I can continue enjoying my Diet Coke with more peace of mind.

Part II: Method Justification

B.1. Summarize four assumptions of a logistic regression model.

1. The dependent variable (y) is binary (Yes/No, M/F, etc.) in nature.

2. None of the independent variables (x) are correlated to one another. This means multicollinearity does not exist between any two variables. If multicollinearity exists, it skews the mean and several different metrics. It will ruin the model's ability to predict accurately. Observations (single data points in the dataset) need to be independent of each other.

3. There is a large sample size to analyze insights from. As someone from a biological science background, we were taught large sample sizes are more accurate and reliable than small sample sizes that do not show much of the information you are researching.

4. No extreme outliers would skew the mean and affect the model's ability to predict correctly.

B.2. Describe two benefits of using Python or R in support of various phases of the analysis.

Python was the language of choice, and it was easy to make that decision as it is relatively easy to learn and use. In addition to ease of use, python is excellent for data science processes and contains a variety of packages that aid in analyzing this research question. The packages that will be utilized in this project are as follows:

- Pandas – load and handle the dataset within the jupyter notebook.
- NumPy is used to complete any calculations or change the data types of some columns.
- Seaborn and Matplotlib – for creating vibrant and easy-to-read visualizations of the research results.
- SciPy's statmodels – allow the computation of residuals and variance inflation factor for the regression model.

- Sklearn's preprocessing – will enable us to transform the data as we see fit.

B.3. Explain why logistic regression is an appropriate technique to analyze the research question.

Both linear and logistic regression are great statistical tools to help a company predict an outcome. Logistic regression allows us to predict a binary outcome (IBM). My research question was, "What factors are connected to a patient's choice of soft drinks?" This question was a curiosity of my own as I am a fan of Diet Coke (and possibly addicted to it), so I need to understand what factors (independent variables) that patients come into the hospital with might indicate whether they drink soft drinks or not. Logistic regression provides insights into whether those independent variables are tied to the patient's choice of drink. I assume that if you have many of those factors, like back pain, high blood pressure, a stroke, etc., you also drink soft drinks. My results below will shock you, but this is from data from a single day of hospital operation.

Part III: Data Preparation

C.1. Describe your data cleaning goals and the steps used to clean the data.

As stated in my previous task submission (Hosey 2024), the dataset is clean, with no repeats, null values, etc. Area, Timezone, Martial, Initial_admin, Complication_risk, and Services were changed to a string using `.astype("category")`.

The main cleaning task is the remapping and changing of the types of variables to allow statistical analysis. Our variables need to be numeric, and there are several columns with Yes/No responses. Boolean remapping, `.map(bool_mapping)`, was used to convert those Yes/No responses to a 1 or a 0. The columns that required this conversion were as follows: ReAdmis, Soft_drink, HighBlood, Stroke, Overweight, Arthritis, Diabetes, Hyperlipidemia, BackPain, Anxiety, Allergic_rhinitis, Reflux_esophagitis, and Asthma.

In addition to the above changes, several code statements were used to verify the dataset's information. `.isnull().sum()` was used to ensure that the data columns contained no null values. The `.head()` and `.dtypes` were used to view the data and types.

C.2. Describe the dependent variable and all independent variables using summary statistics.

Dependent variable (y – Response): Soft Drink – Whether a patient drinks carbonated soft drinks.

```
#Dependent Variable
df.Soft_drink.value_counts()
```

```
Soft_drink
No     7425
Yes    2575
Name: count, dtype: int64
```

Summary: Approximately 75% of patients said they do not drink soft drinks.

Independent Variables (x – Explanatory):

```
#Independent Variables to follow:
df.Age.describe()

count    10000.000000
mean        53.511700
std         20.638538
min         18.000000
25%         36.000000
50%         53.000000
75%         71.000000
max         89.000000
Name: Age, dtype: float64
```

Summary: Most patients were 53 years old. The oldest patient was 89, and the youngest was 18 (I assume due to HIPAA regulations and policies).

```
df.Doc_visits.describe()

count    10000.000000
mean         5.012200
std          1.045734
min          1.000000
```

Summary: This was almost 50/50 between women and men. More women visited the hospital.

Summary: Most patients had a vitamin D level of 17.9.

Summary: Most patients have visited the doctor approximately 5 times.

```
rt_index()
```

```
df.Complication_risk.value_counts().sort_index()

Complication_risk
High      3358
Low       2125
Medium    4517
Name: count, dtype: int64
```

Summary: Half (50%) of hospital admissions were emergency-related, 25% were elective, and 25% were for observation.

Summary: About 45% were admitted with medium complication risk, 33% were at high risk, and 21% were low-risk patients. patients.

```
Arthritis
No      6426
Yes     3574
Name: count, dtype: int64
```

Summary: 64% of patients said they did not have or are not affected by arthritis. 35% of patients said they did.

Summary: 72% of patients did not have diabetes, and 27% of patients did.

```
df.BackPain.value_counts()

BackPain
No      5886
Yes     4114
Name: count, dtype: int64
```

Summary: 58% responded to having back pain, and 41% answered that they did not.

```
count    10000.000000
mean      5312.172769
```

Summary: Most patients were charged $5,312.17 for their visit to the hospital.

```
df.Initial_days.describe()

count    10000.000000
mean        34.455299
```

Summary: Most patients were at the hospital for 34 days after admission.

```
df.Hyperlipidemia.value_counts()

Hyperlipidemia
No      6628
Yes     3372
Name: count, dtype: int64
```

Summary: 66% of patients did not have Hyperlipidemia, and 33% did.

Summary: 67% of patients do not deal with anxiety, and 32% do.

Summary: 70% of patients were overweight, and 29% were not.

```
Yes     3215
Name: count, dtype: int64
```

```
Stroke
No      8007
Yes     1993
Name: count, dtype: int64
```

Summary: 80% of patients have not had a stroke, while 19% of patients have had one.

```
HighBlood
No      5910
```

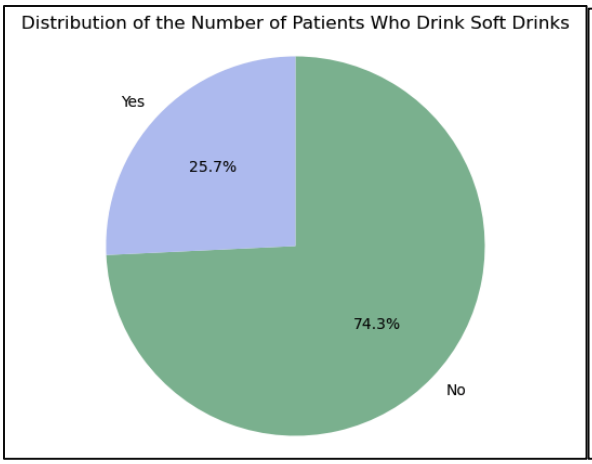
Summary: 59% of patients did not have high blood pressure, while 40% did.

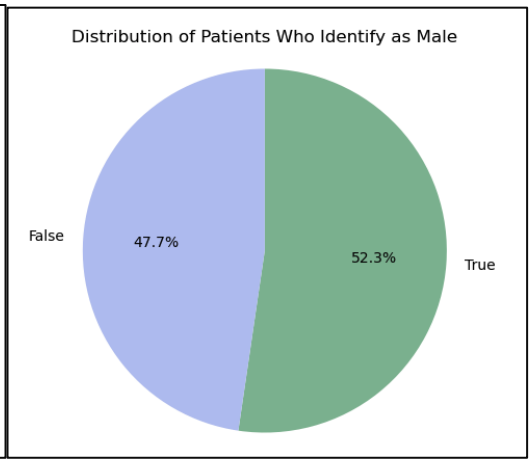
```
df.Asthma.value_counts()

Asthma
No      7107
Yes     2893
Name: count, dtype: int64
```

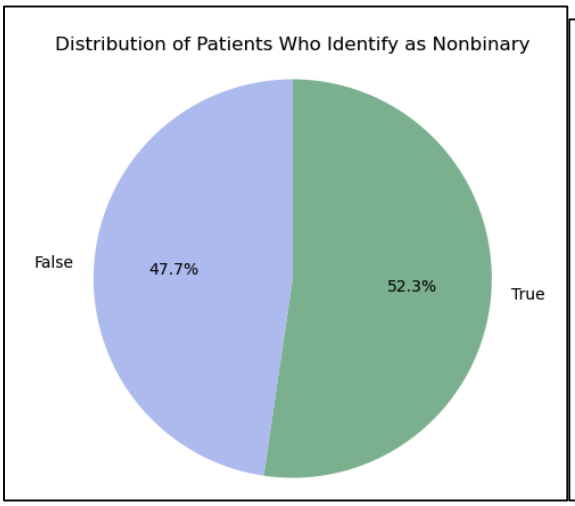
Summary: 71% of patients did not have Asthma, while 28% did.

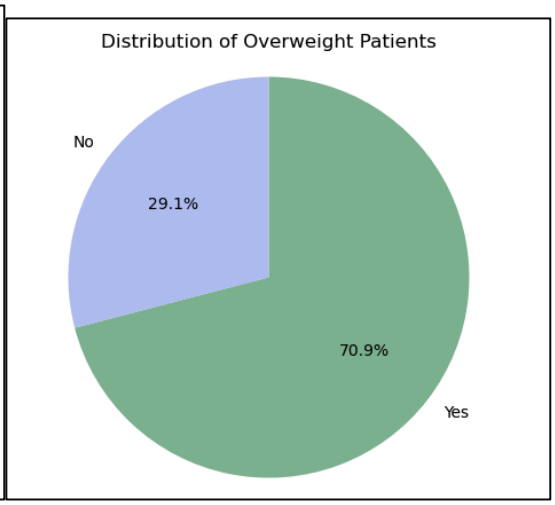
C.3. Generate univariate and bivariate visualizations of the distributions of the dependent and independent variables, including the dependent variable in your bivariate visualizations.

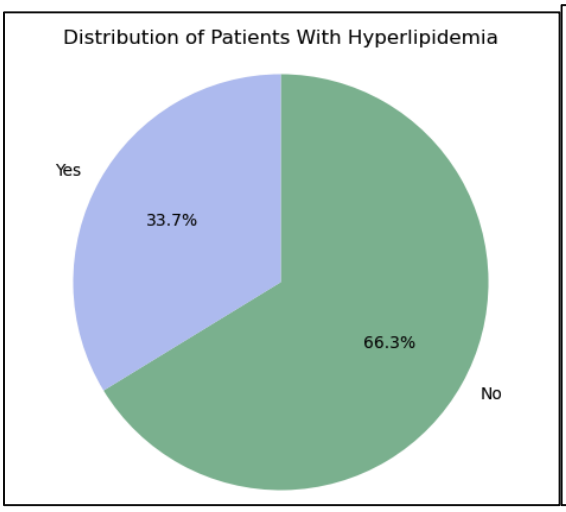
<u>Univariate Visualizations – Categorical Variables</u>

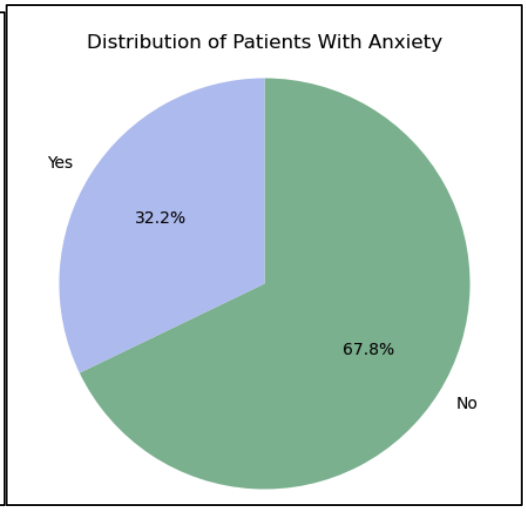

Distribution of the Number of Patients Who Drink Soft Drinks
Yes
25.7%
74.3%
No

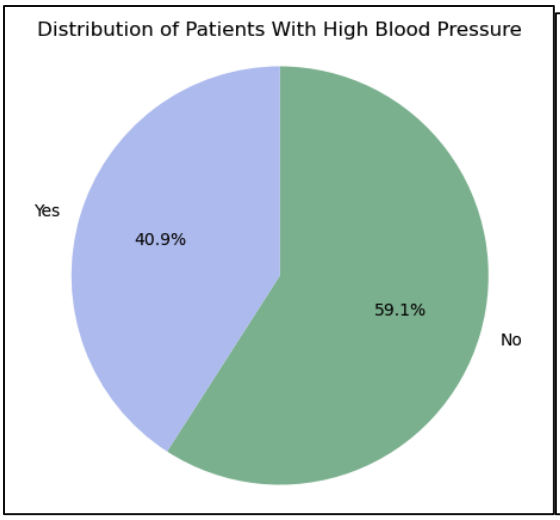

Distribution of Patients Who Identify as Male
False
47.7%
52.3%
True

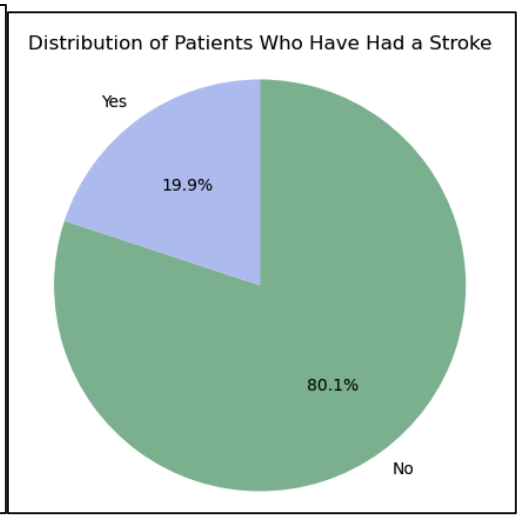

Distribution of Patients Who Identify as Nonbinary
False
47.7%
52.3%
True

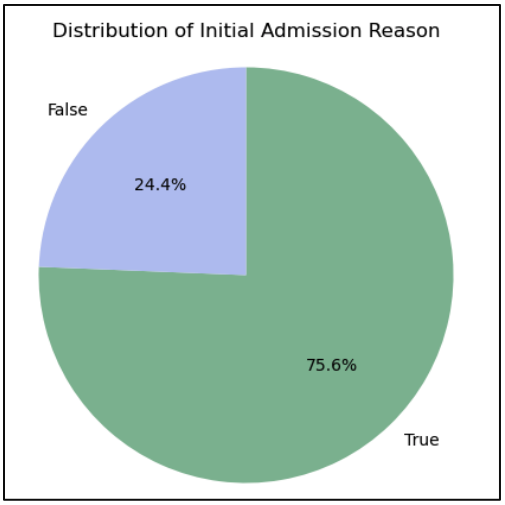

Distribution of Overweight Patients
No
29.1%
70.9%
Yes


Distribution of Patients With Hyperlipidemia
Yes
33.7%
66.3%
No

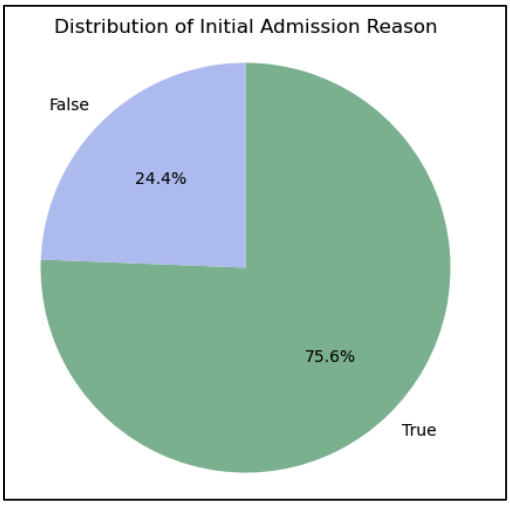

Distribution of Patients With Anxiety
Yes
32.2%
67.8%
No



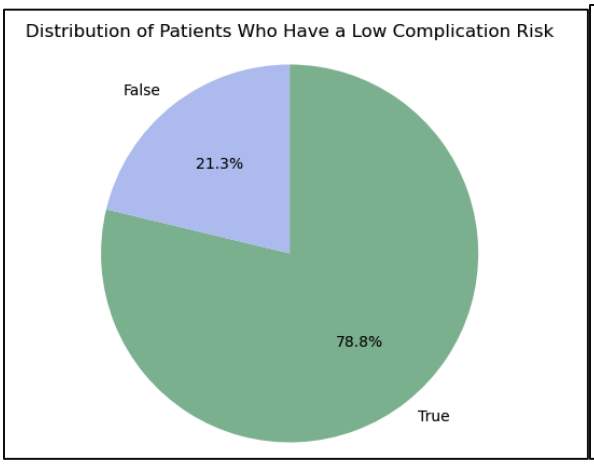


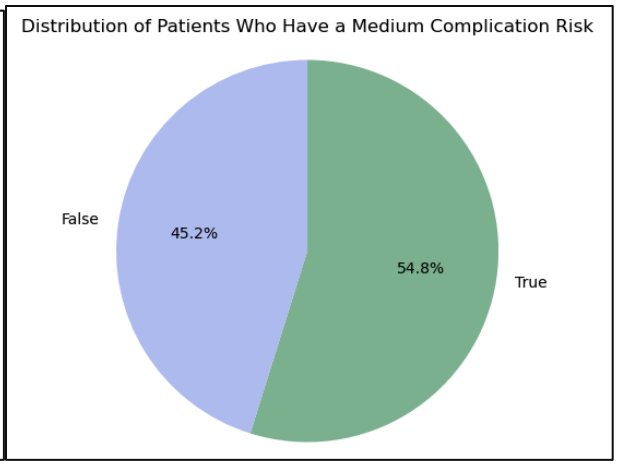


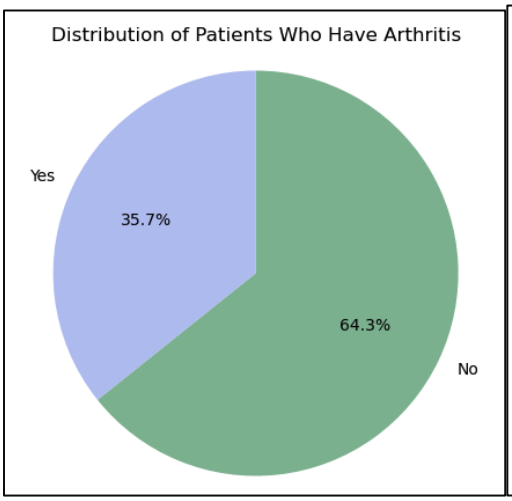
Reason - Observation

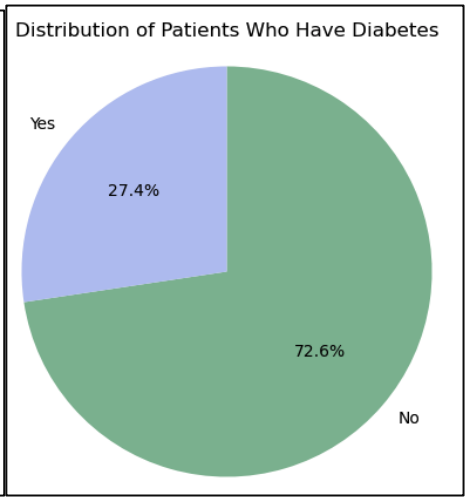


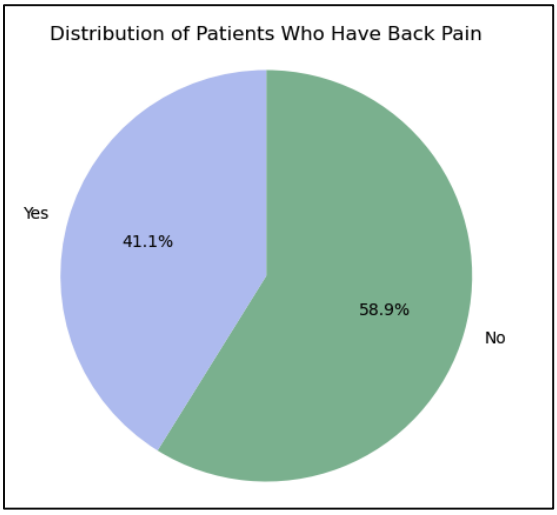
Reason – Emergency


Distribution of Patients Who Have a Low Complication Risk
False
21.3%
78.8%
True

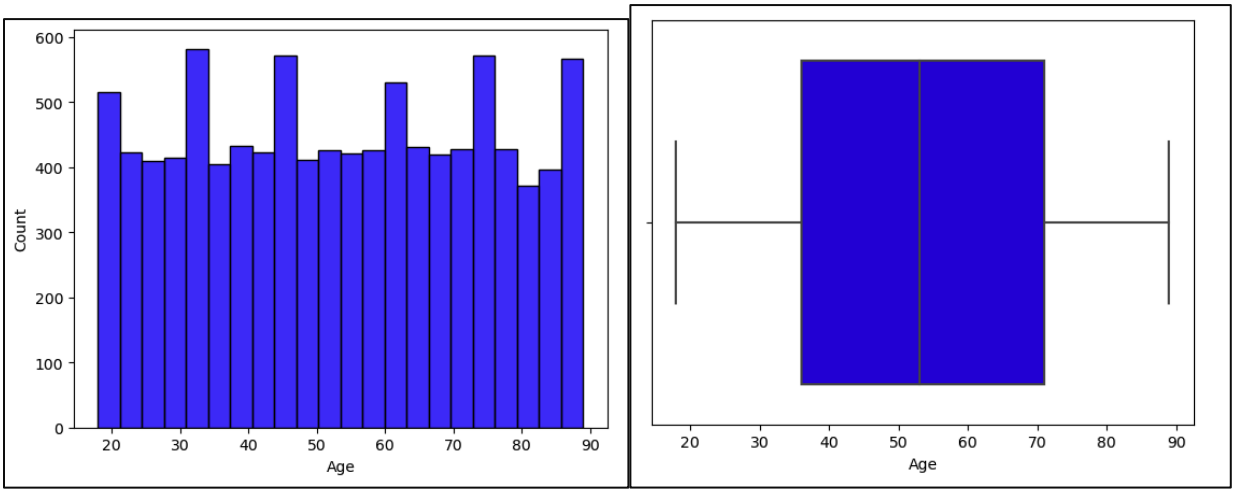

Distribution of Patients Who Have a Medium Complication Risk
False
45.2%
54.8%
True

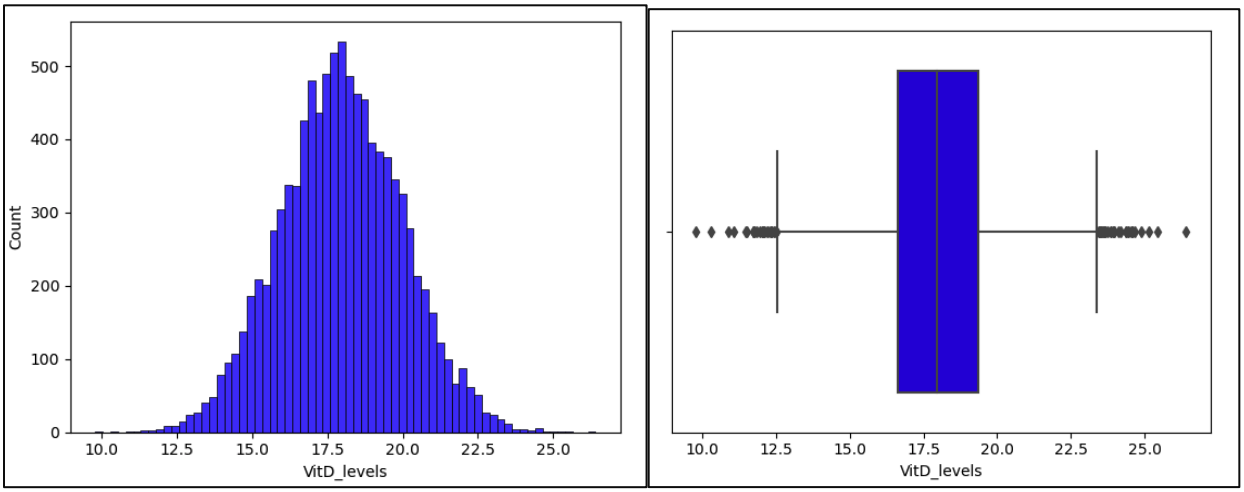

Distribution of Patients Who Have Arthritis
Yes
35.7%
64.3%
No

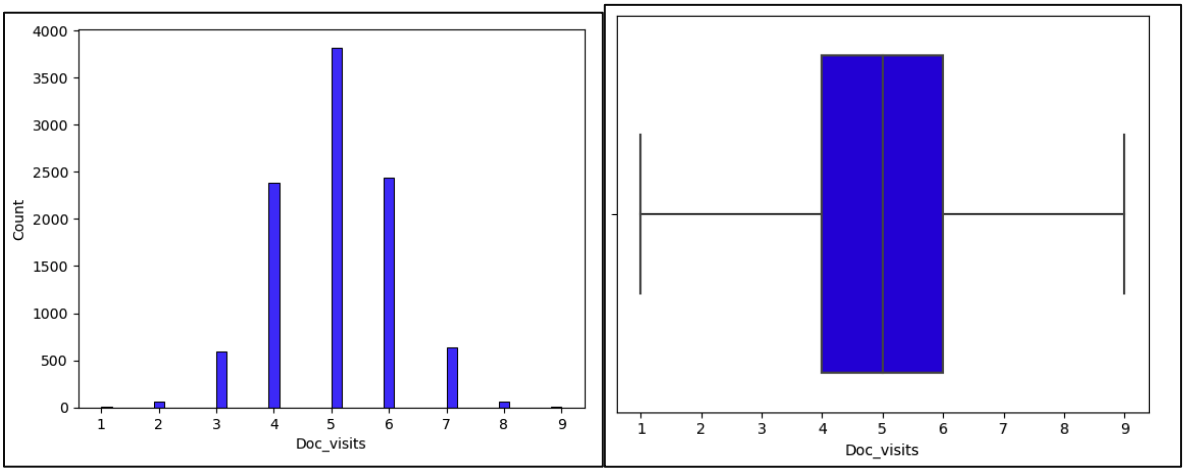

Distribution of Patients Who Have Diabetes
Yes
27.4%
72.6%
No

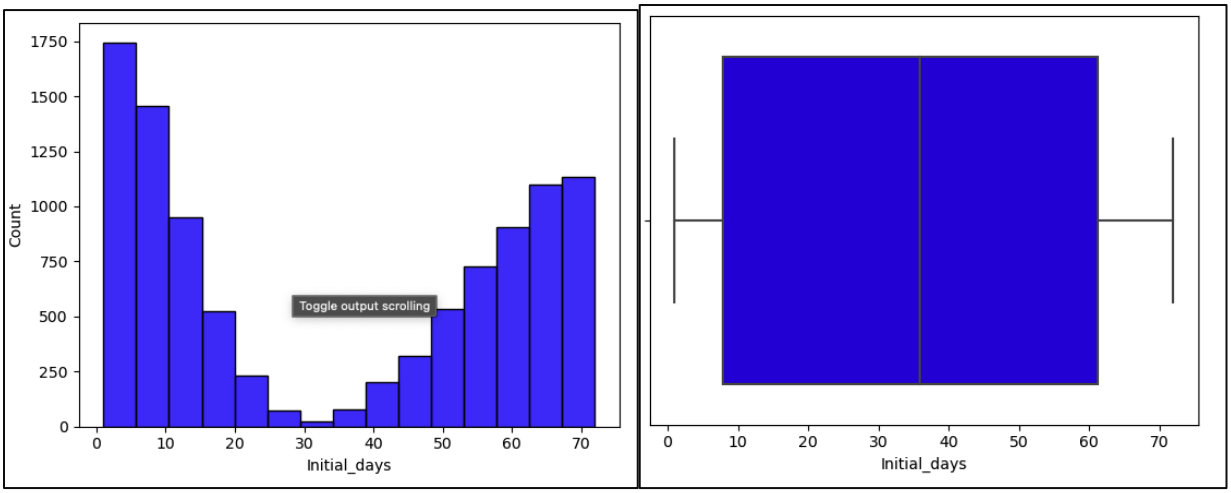

Distribution of Patients Who Have Back Pain
Yes
41.1%
58.9%
No

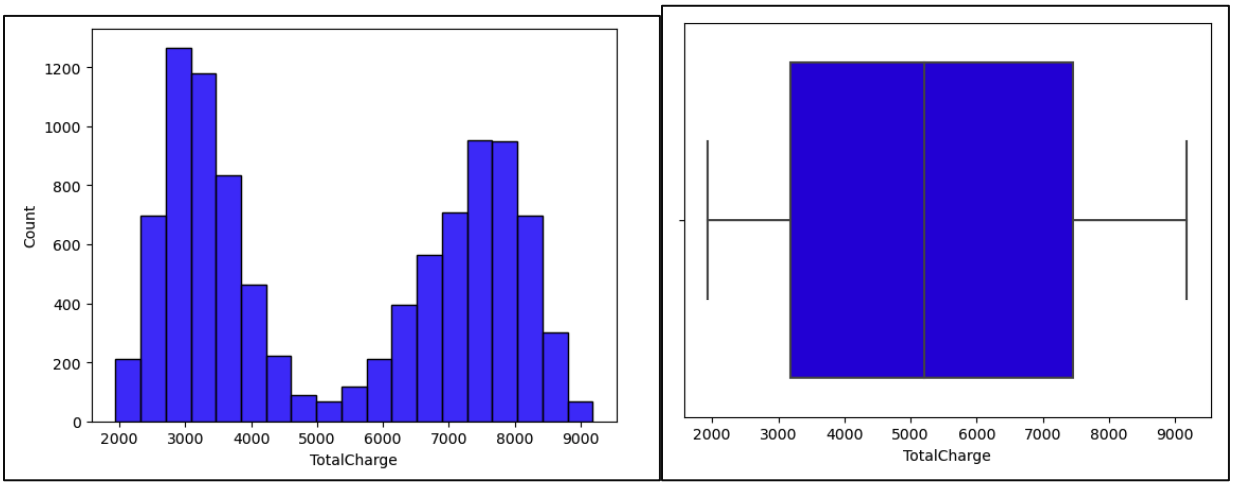
Univariate Visualizations – Continuous Variables



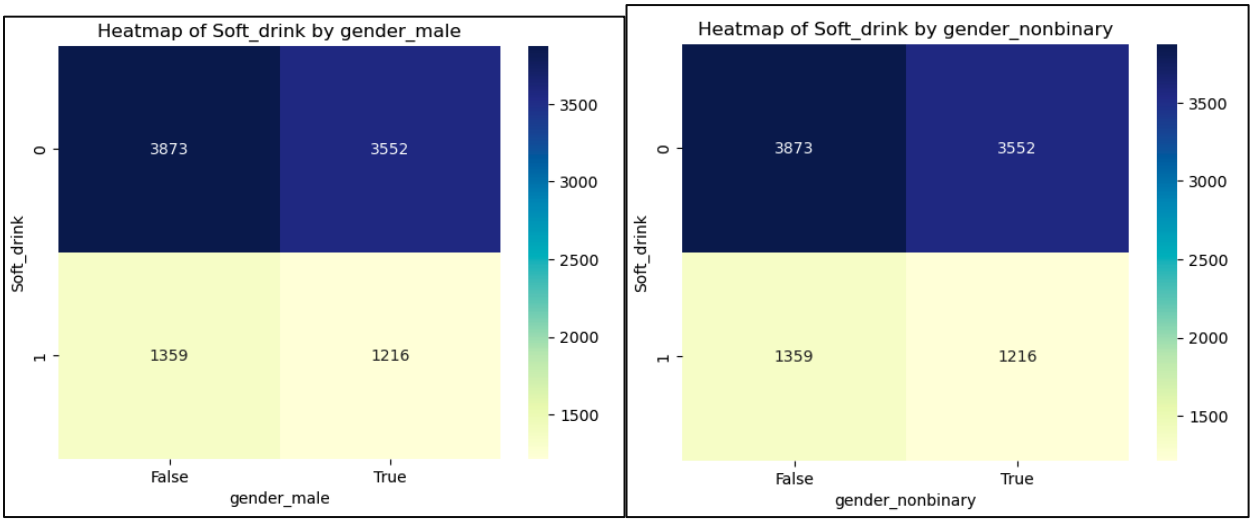


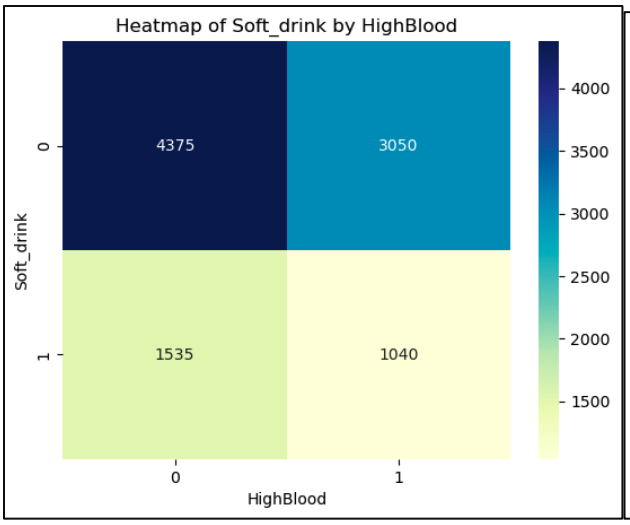


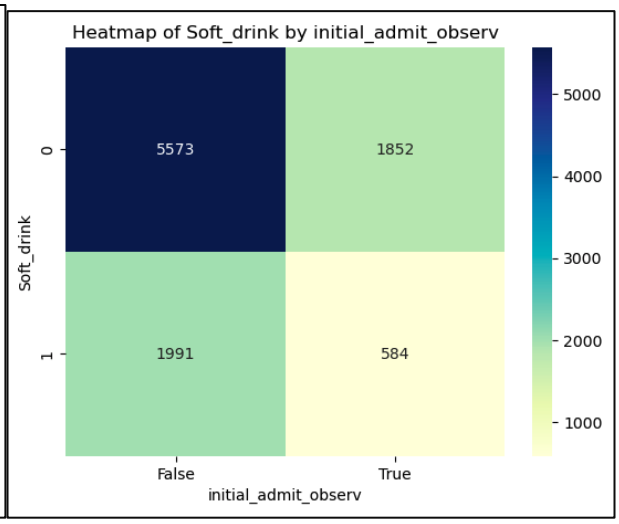


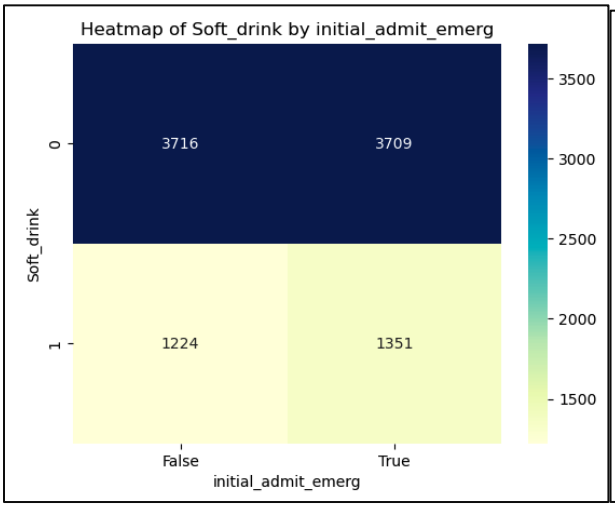


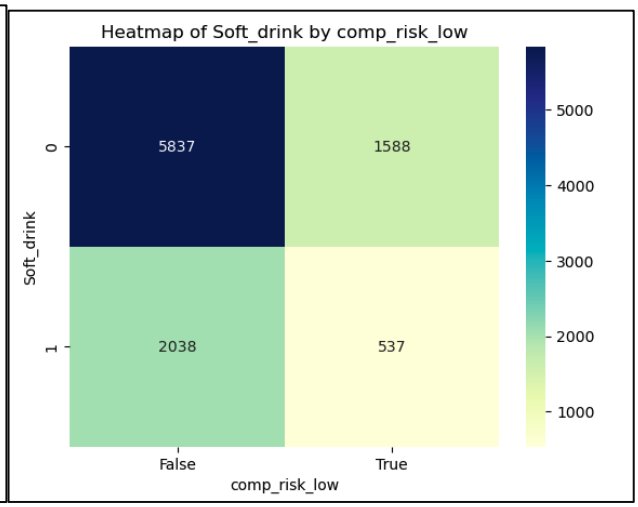
Bivariate Visualizations – Categorical Variables

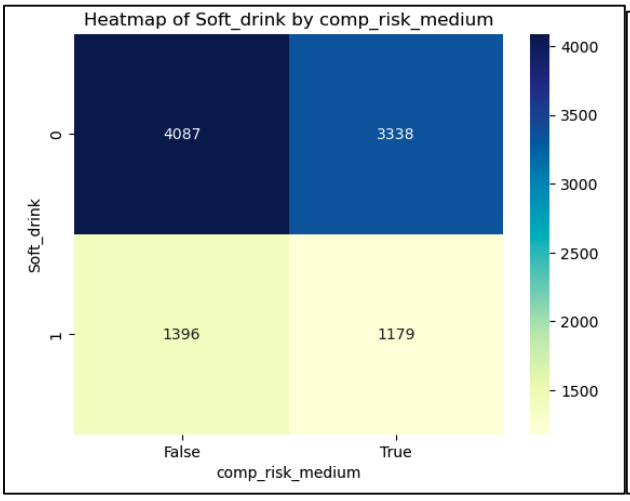


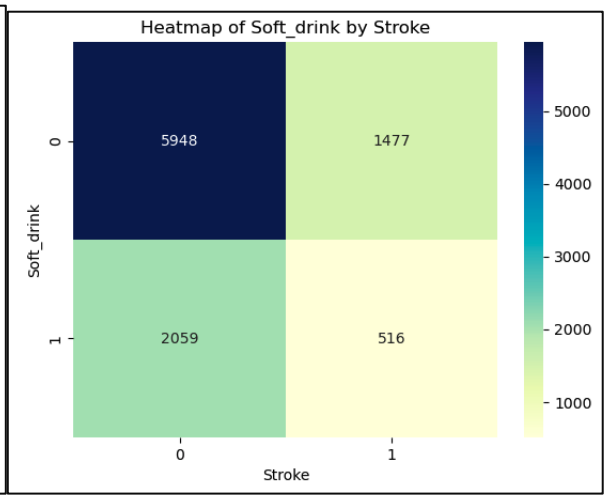

Heatmap of Soft_drink by HighBlood
Soft_drink
HighBlood
0
1
4375
3050
1535
1040
4000
3500
3000
2500
2000
1500

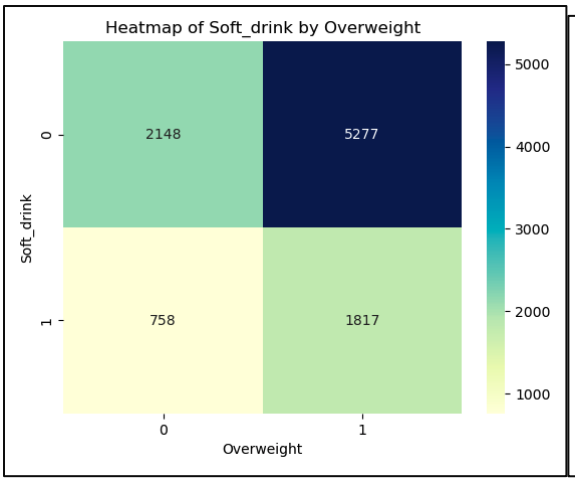

Heatmap of Soft_drink by initial_admit_observ
Soft_drink
initial_admit_observ
False
True
5573
1852
1991
584
5000
4000
3000
2000
1000

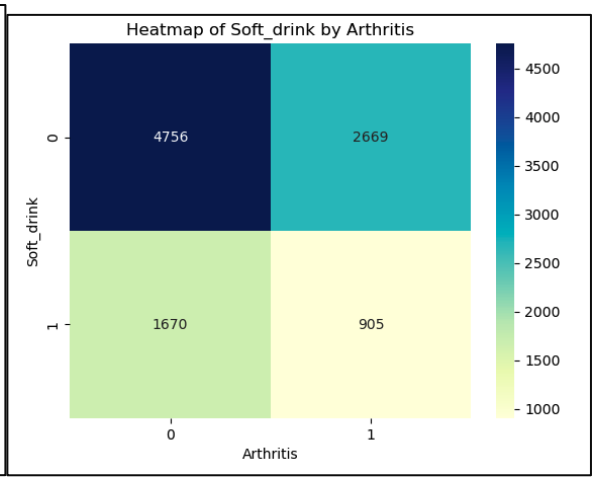

Heatmap of Soft_drink by initial_admit_emerg
Soft_drink
initial_admit_emerg
False
True
3716
3709
1224
1351
3500
3000
2500
2000
1500

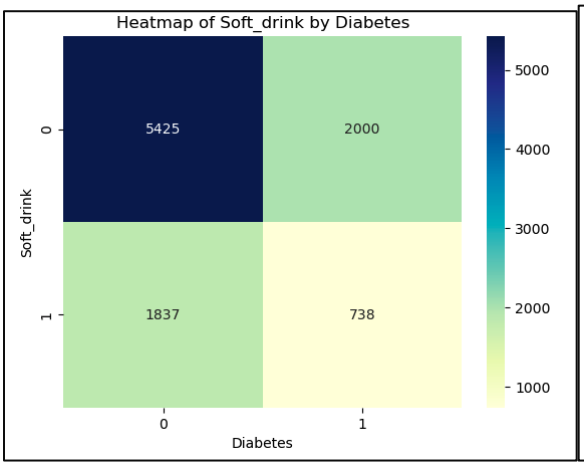

Heatmap of Soft_drink by comp_risk_low
Soft_drink
comp_risk_low
False
True
5837
1588
2038
537
5000
4000
3000
2000
1000

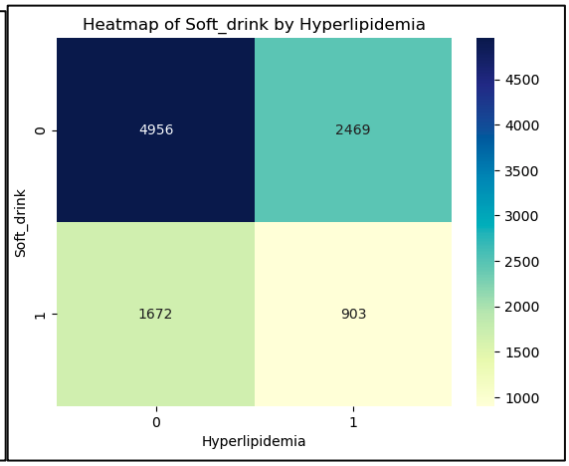

Heatmap of Soft_drink by comp_risk_medium
Soft_drink
comp_risk_medium
False
True
4087
3338
1396
1179
4000
3500
3000
2500
2000
1500

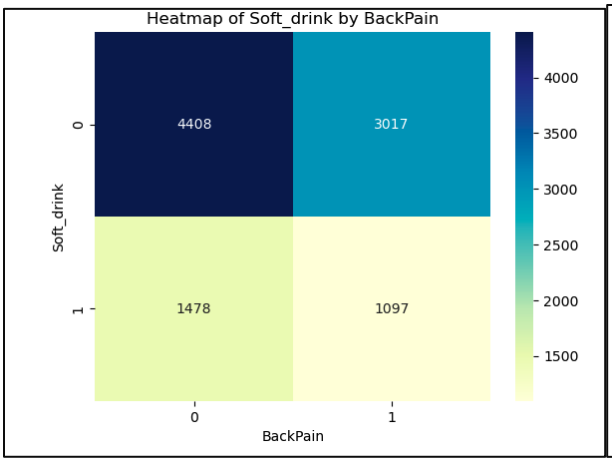

Heatmap of Soft_drink by Stroke
Soft_drink
Stroke
0
1
5948
1477
2059
516
5000
4000
3000
2000
1000

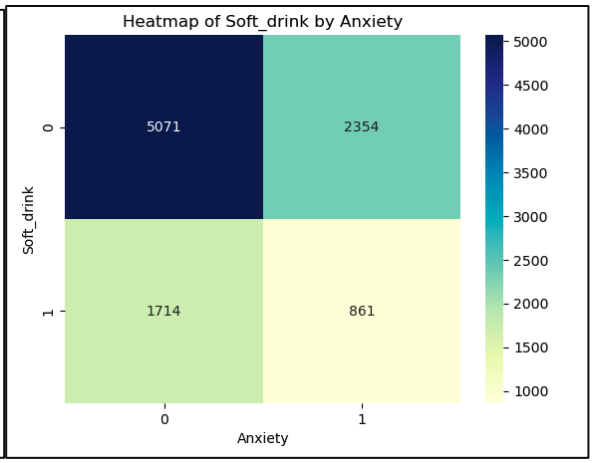

Heatmap of Soft_drink by Overweight
Soft_drink
Overweight
0
1
2148
5277
758
1817
5000
4000
3000
2000
1000

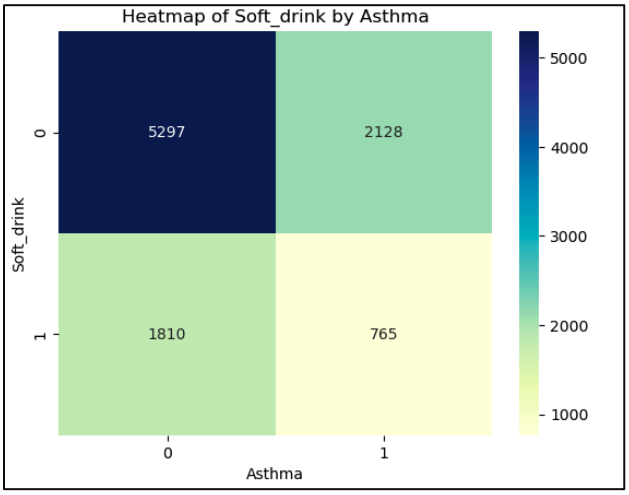

Heatmap of Soft_drink by Arthritis
Soft_drink
Arthritis
0
1
4756
2669
1670
905
4500
4000
3500
3000
2500
2000
1500
1000


Heatmap of Soft_drink by Diabetes
Soft_drink
Diabetes
0
1
5425
2000
1837
738
5000
4000
3000
2000
1000

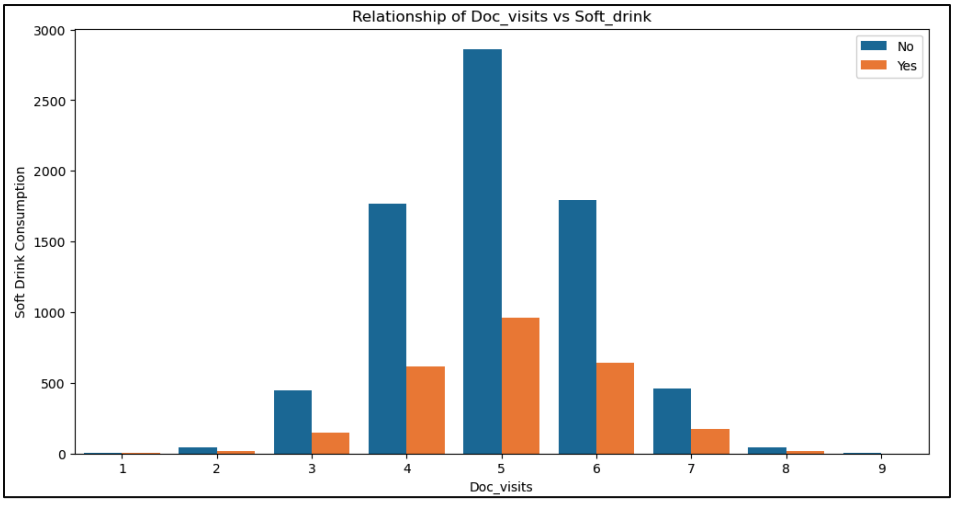

Heatmap of Soft_drink by Hyperlipidemia
Soft_drink
Hyperlipidemia
0
1
4956
2469
1672
903
4500
4000
3500
3000
2500
2000
1500
1000

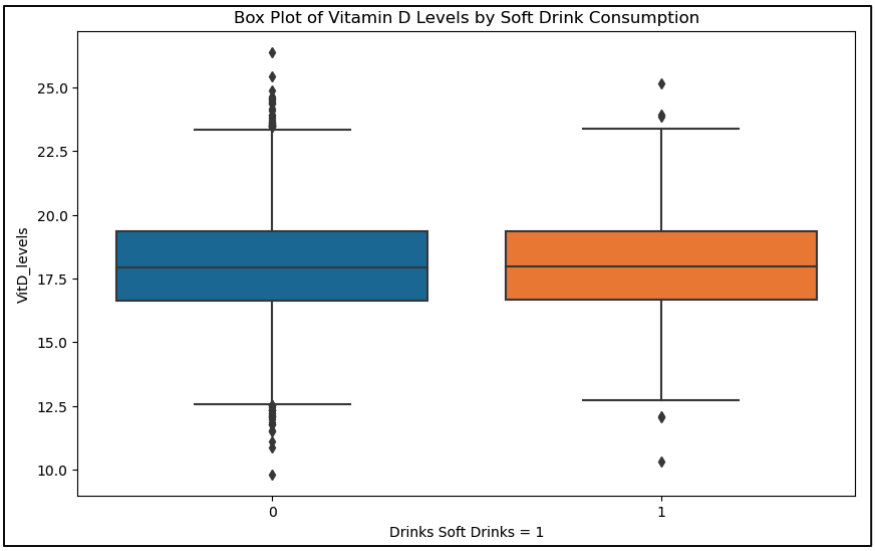


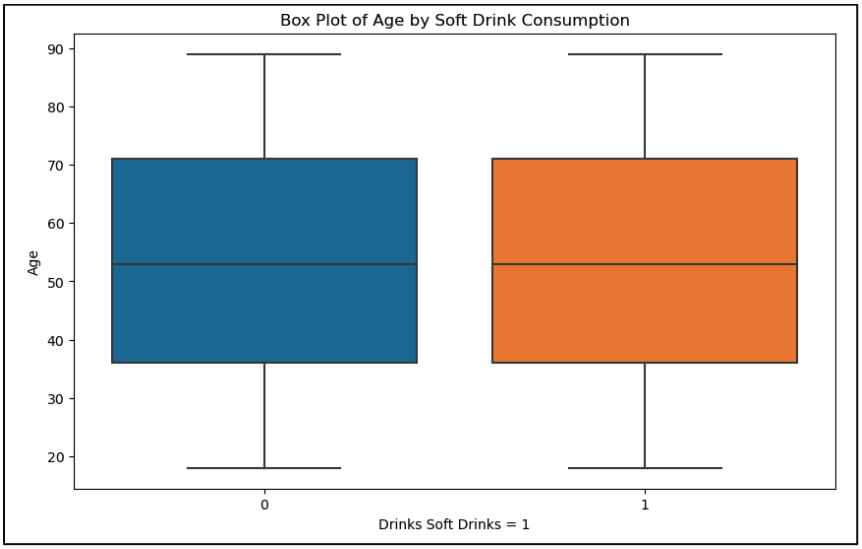


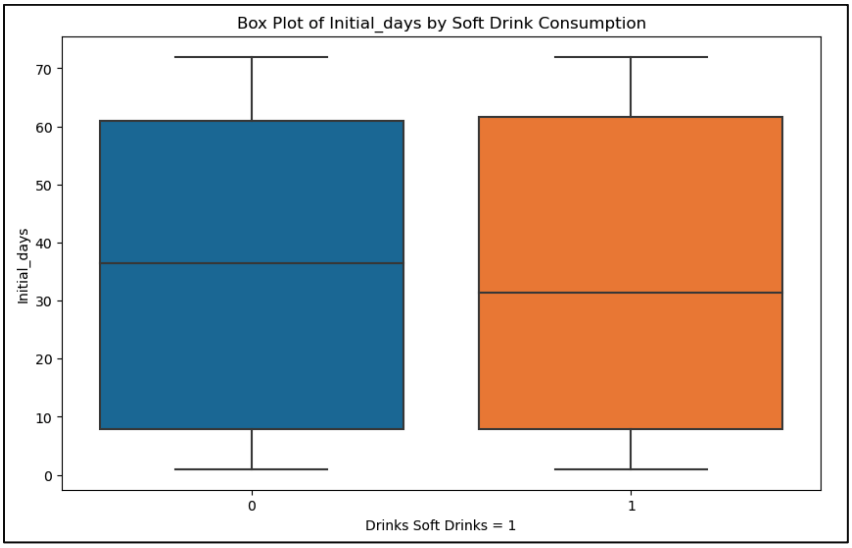


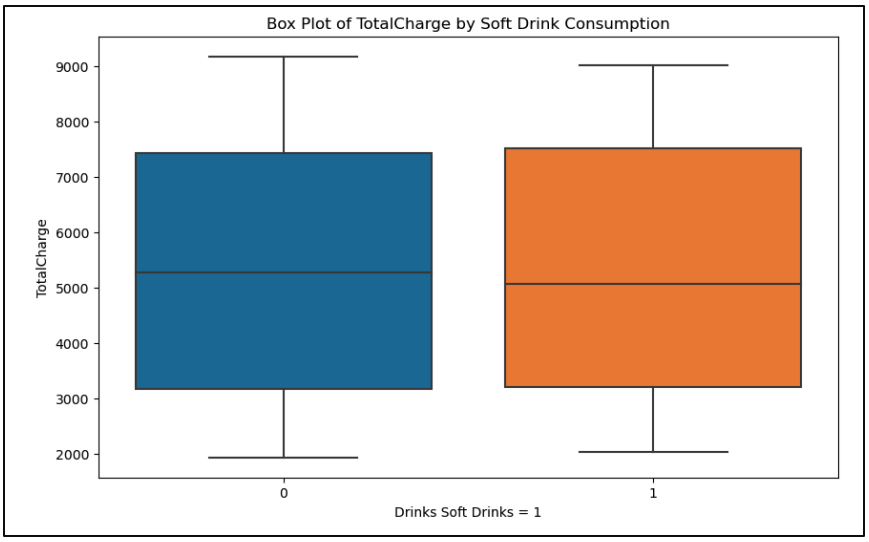
Bivariate Visualizations – Continuous Variables




Box Plot of Vitamin D Levels by Soft Drink Consumption
VitD_levels
25.0
22.5
20.0
17.5
15.0
12.5
10.0
0
1
Drinks Soft Drinks = 1


Box Plot of Age by Soft Drink Consumption
Age
90
80
70
60
50
40
30
20
0
1
Drinks Soft Drinks = 1


Box Plot of Initial_days by Soft Drink Consumption
Initial_days
70
60
50
40
30
20
10
0
0
1
Drinks Soft Drinks = 1



C.4. Describe your data transformation goals that align with your research question.

As stated in my previous task submission (Hosey 2024), the changes were to use numerical values to complete logistic regression. Therefore, several columns (the nominal categorical variables) needed to be remapped, so a Yes response would be worth a value of 1, and a No response would be worth a value of 0. To do this, the following code was used: `pd.get_dummies,` and `.insert`.

In addition, the need to change Items 1-8 (ordinal categorical variables) in the dataset to reflect how the patient felt about a specific metric was given a relationship. For example, a 1 response would be considered "most important" to the patient, and the computer needs to understand that. Without this change, typically, a 1 is less critical or weighted differently from an 8. This would mess up the value of the mean and standard error. The following code was used to create this relationship for the computer: numerical relationship for the Item 1 through Item 8 survey responses: `CategoricalDtype`, `.map`, and `.astype`.

The last thing used to verify that all data was numerical before creating the logistic regression model was to make a function for the computer to search through and change values to numeric if they were still unchanged after all those prior modifications. The following code was used to execute that function: `convert_to_numeric`, `.columns`, `.dtype`, `.loc`, `.astype`, and `.to_numeric`.

C.5. Provide the prepared data set as a CSV file.

See the attached files from my task submission.

Part IV: Model Comparison and Analysis

D.1. Construct an initial logistic regression model from all independent variables.

```
#Initial Log Model
#Set dependent variable for Y
y = log_regression_df.Soft_drink

#Set multiple independent variables for X and add constant
X = log_regression_df[["Age", "gender_male", "gender_nonbinary", "VitD_levels", "Doc_visits", "HighBlood",
               "initial_admit_observ", "initial_admit_emerg", "Stroke", "Overweight", "Arthritis", "comp_risk_low",
               "comp_risk_medium", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Asthma", "Initial_days", "TotalCharge"]]

X = sm.add_constant(X)  # This will add a constant column

#Double checking that all X columns are numeric
X = X.apply(pd.to_numeric, errors='coerce')

#Fit and print the model
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569213
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9980
Method:                           MLE   Df Model:                           19
Date:                Tue, 18 Jun 2024   Pseudo R-squ.:                0.002126
Time:                        18:05:00   Log-Likelihood:                -5692.1
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                    0.1865
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
const                        0.3312        nan        nan        nan         nan         nan
Age                         -0.0003      0.001     -0.228      0.819      -0.002       0.002
gender_male                 -0.0124   4.68e+15  -2.65e-18      1.000   -9.17e+15    9.17e+15
gender_nonbinary            -0.0124   4.68e+15  -2.65e-18      1.000   -9.17e+15    9.17e+15
VitD_levels                  0.0044      0.011      0.388      0.698      -0.018       0.027
Doc_visits                   0.0260      0.023      1.109      0.267      -0.020       0.072
HighBlood                    0.0538        nan        nan        nan         nan         nan
initial_admit_observ        -0.0917      0.065     -1.411      0.158      -0.219       0.036
initial_admit_emerg          0.4272        nan        nan        nan         nan         nan
Stroke                       0.0112      0.057      0.195      0.845      -0.101       0.123
Overweight                  -0.0239      0.051     -0.465      0.642      -0.125       0.077
Arthritis                    0.0193      0.046      0.423      0.672      -0.070       0.109
comp_risk_low               -0.3190        nan        nan        nan         nan         nan
comp_risk_medium            -0.2716        nan        nan        nan         nan         nan
Diabetes                     0.1407      0.086      1.634      0.102      -0.028       0.309
Hyperlipidemia               0.1490      0.052      2.866      0.004       0.047       0.251
BackPain                     0.1441      0.040      3.615      0.000       0.066       0.222
Anxiety                      0.1407      0.045      3.132      0.002       0.053       0.229
Asthma                       0.0503      0.051      0.994      0.320      -0.049       0.149
Initial_days                 0.0599        nan        nan        nan         nan         nan
TotalCharge                 -0.0007        nan        nan        nan         nan         nan
============================================================================================
```

```
log_regression_df.isnull().sum()
```

```
Age                     0
gender_male             0
gender_nonbinary        0
VitD_levels             0
Doc_visits              0
Soft_drink              0
HighBlood               0
initial_admit_observ    0
initial_admit_emerg     0
Stroke                  0
Overweight              0
Arthritis               0
comp_risk_low           0
comp_risk_medium        0
Diabetes                0
Hyperlipidemia          0
BackPain                0
Anxiety                 0
Asthma                  0
Initial_days            0
TotalCharge             0
dtype: int64
```

The initial run had several results with Nan values, so I checked for null values. There were no Null values in the data frame, so I moved to checking for multicollinearity. Initial Run had several results with Nan values, so I checked for null values. There were no Null values in the data frame, so I moved to checking for Multicollinearity.

```
#Checking for Multicollinearity using VIF due to having NaN values in the results and there is no columns with NaN values
X = log_regression_df[["Age", "gender_male", "gender_nonbinary", "VitD_levels", "Doc_visits", "HighBlood",
               "initial_admit_observ", "initial_admit_emerg", "Stroke", "Overweight", "Arthritis", "comp_risk_low",
               "comp_risk_medium", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Asthma", "Initial_days", "TotalCharge"]]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

```
/opt/conda/envs/anaconda-panel-2023.05-
ero encountered in scalar divide
  vif = 1. / (1. - r_squared_i)
                feature         VIF
0                   Age    7.719070
1           gender_male         inf
2      gender_nonbinary         inf
3           VitD_levels   78.252486
4            Doc_visits   23.867293
5             HighBlood    1.862913
6   initial_admit_observ    1.974016
7    initial_admit_emerg    6.633095
8                Stroke    1.250241
9            Overweight    3.445122
10            Arthritis    1.631781
11        comp_risk_low    2.198155
12     comp_risk_medium    3.602866
13             Diabetes    1.442638
14       Hyperlipidemia    1.627488
15             BackPain    1.816659
16              Anxiety    1.556567
17               Asthma    1.409513
18         Initial_days  283.631467
19          TotalCharge  716.952843
```

The results for that contained two variables with infinite VIF values. I removed the gender_nonbinary variable as it was the smaller sample set of the two with high VIF values.

```
#Checking VIF after removing gender_nonbinary
X = log_regression_df[["Age", "gender_male", "VitD_levels", "Doc_visits", "HighBlood",
               "initial_admit_observ", "initial_admit_emerg", "Stroke", "Overweight", "Arthritis", "comp_risk_low",
               "comp_risk_medium", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Asthma", "Initial_days", "TotalCharge"]]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Age	7.719070
1	gender_male	1.913295
2	VitD_levels	78.252486
3	Doc_visits	23.867293
4	HighBlood	1.862913
5	initial_admit_observ	1.974016
6	initial_admit_emerg	6.633095
7	Stroke	1.250241
8	Overweight	3.445122
9	Arthritis	1.631781
10	comp_risk_low	2.198155
11	comp_risk_medium	3.602866
12	Diabetes	1.442638
13	Hyperlipidemia	1.627488
14	BackPain	1.816659
15	Anxiety	1.556567
16	Asthma	1.409513
17	Initial_days	283.631467
18	TotalCharge	716.952843

I reran the VIF code and fixed the issue. I reran the initial model code to see if I had any Nan values in the results table. This time, I did not (as seen below), so I moved forward with reducing the model. fixed the issue. I reran the initial model code to see if I had any Nan values in the results table. This time I did not (as seen below), and I moved forward with reducing the model.

```
#Second run of Log Model - Without gender_nonbinary Variable do to high VIF hoping for no NaN values in results table.
#Set dependent variable for Y
y = log_regression_df.Soft_drink

#Set multiple independent variables for X and add constant
X = log_regression_df[["Age", "gender_male", "VitD_levels", "Doc_visits", "HighBlood",
               "initial_admit_observ", "initial_admit_emerg", "Stroke", "Overweight", "Arthritis", "comp_risk_low",
               "comp_risk_medium", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Asthma", "Initial_days", "TotalCharge"]]

X = sm.add_constant(X)  # This will add a constant column

#Double checking that all X columns are numeric
X = X.apply(pd.to_numeric, errors='coerce')

#Fit and print the model
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569213
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9980
Method:                           MLE   Df Model:                           19
Date:                Tue, 18 Jun 2024   Pseudo R-squ.:                0.002126
Time:                        18:05:02   Log-Likelihood:                -5692.1
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                    0.1865
==============================================================================================
                                coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                        0.3312      1.309      0.253      0.800      -2.234       2.897
Age                         -0.0003      0.001     -0.235      0.815      -0.002       0.002
gender_male                 -0.0248      0.046     -0.541      0.588      -0.115       0.065
VitD_levels                  0.0044      0.011      0.388      0.698      -0.018       0.027
Doc_visits                   0.0260      0.022      1.184      0.236      -0.017       0.069
HighBlood                    0.0538      0.078      0.689      0.491      -0.099       0.207
initial_admit_observ        -0.0917      0.066     -1.388      0.165      -0.221       0.038
initial_admit_emerg          0.4272      0.288      1.484      0.138      -0.137       0.992
Stroke                       0.0112      0.057      0.195      0.845      -0.101       0.123
Overweight                  -0.0239      0.050     -0.475      0.635      -0.123       0.075
Arthritis                    0.0193      0.063      0.307      0.759      -0.104       0.142
comp_risk_low               -0.3190      0.237     -1.345      0.178      -0.784       0.146
comp_risk_medium            -0.2716      0.235     -1.156      0.248      -0.732       0.189
Diabetes                     0.1407      0.066      2.142      0.032       0.012       0.269
Hyperlipidemia               0.1490      0.071      2.108      0.035       0.010       0.288
BackPain                     0.1441      0.067      2.164      0.030       0.014       0.275
Anxiety                      0.1407      0.068      2.068      0.039       0.007       0.274
Asthma                       0.0503      0.050      1.000      0.318      -0.048       0.149
Initial_days                 0.0599      0.045      1.322      0.186      -0.029       0.149
TotalCharge                 -0.0007      0.001     -1.317      0.188      -0.002       0.000
==============================================================================================
```

D.2. Justify a statistically based feature selection procedure or a model evaluation metric to reduce the initial model in a way that aligns with the research question.

Two metrics were used to evaluate the model before the reduction of variables. As my task 1 submission (Hosey 2024) stated, the Variance Inflation Factor and P-values were used to maintain no multilinearity and reliability in the model's results. If multilinearity existed, the variables would be too similar, skew the model results, and decrease reliability.

Variance Inflation Factor (VIF) was used to analyze how the independent variable changes the standard error (Hosey 2024). If an independent variable has a high value, it must be eliminated to maintain the model's standard error. Then, when the model's initial run occurred, there was an issue with the results computed as Nan values. So, I used VIF to investigate the problem; two independent variables, gender_male, and gender_nonbinary, had a VIF value of infinite. This would cause the standard error to be inflated if I were to continue. Since the gender_nonbinary variable had less than a 1 (Yes) response, I chose to eliminate it. Rerunning VIF fixed the issue, and I could run a second initial model to see if any Nan values were present. None showed up, so I reran VIF to start evaluating the model. Any VIF value above 10 was removed. Once all the VIF values were under 10, I used P-values from the model's results table to further eliminate independent variables that would decrease the reliability of the model's results. There were several eliminations and ended with one independent variable within the model.

D.3. Provide a reduced logistic regression model.

Second run of Initial Model – P-value of 0.1865

```
#Second run of Log Model - Without gender_nonbinary Variable do to high VIF hoping for no NaN values in results table.
#Set dependent variable for Y
y = log_regression_df.Soft_drink

#Set multiple independent variables for X and add constant
X = log_regression_df[["Age", "gender_male", "VitD_levels", "Doc_visits", "HighBlood",
               "initial_admit_observ", "initial_admit_emerg", "Stroke", "Overweight", "Arthritis", "comp_risk_low",
               "comp_risk_medium", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Asthma", "Initial_days", "TotalCharge"]]

X = sm.add_constant(X)  # This will add a constant column

#Double checking that all X columns are numeric
X = X.apply(pd.to_numeric, errors='coerce')

#Fit and print the model
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569213
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9980
Method:                           MLE   Df Model:                           19
Date:                Tue, 18 Jun 2024   Pseudo R-squ.:                0.002126
Time:                        18:05:02   Log-Likelihood:                -5692.1
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                    0.1865
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                          0.3312      1.309      0.253      0.800      -2.234       2.897
Age                           -0.0003      0.001     -0.235      0.815      -0.002       0.002
gender_male                   -0.0248      0.046     -0.541      0.588      -0.115       0.065
VitD_levels                    0.0044      0.011      0.388      0.698      -0.018       0.027
Doc_visits                     0.0260      0.022      1.184      0.236      -0.017       0.069
HighBlood                      0.0538      0.078      0.689      0.491      -0.099       0.207
initial_admit_observ          -0.0917      0.066     -1.388      0.165      -0.221       0.038
initial_admit_emerg            0.4272      0.288      1.484      0.138      -0.137       0.992
Stroke                         0.0112      0.057      0.195      0.845      -0.101       0.123
Overweight                    -0.0239      0.050     -0.475      0.635      -0.123       0.075
Arthritis                      0.0193      0.063      0.307      0.759      -0.104       0.142
comp_risk_low                 -0.3190      0.237     -1.345      0.178      -0.784       0.146
comp_risk_medium              -0.2716      0.235     -1.156      0.248      -0.732       0.189
Diabetes                       0.1407      0.066      2.142      0.032       0.012       0.269
Hyperlipidemia                 0.1490      0.071      2.108      0.035       0.010       0.288
BackPain                       0.1441      0.067      2.164      0.030       0.014       0.275
Anxiety                        0.1407      0.068      2.068      0.039       0.007       0.274
Asthma                         0.0503      0.050      1.000      0.318      -0.048       0.149
Initial_days                   0.0599      0.045      1.322      0.186      -0.029       0.149
TotalCharge                   -0.0007      0.001     -1.317      0.188      -0.002       0.000
==============================================================================================
```

Rechecking VIF values to start reducing the model – any VIF value above 10 will be removed.

```
#Now that the model results contain no NaN values lets check VIF again and start reducing the model
X = log_regression_df[["Age", "gender_male", "VitD_levels", "Doc_visits", "HighBlood",
               "initial_admit_observ", "initial_admit_emerg", "Stroke", "Overweight", "Arthritis", "comp_risk_low",
               "comp_risk_medium", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Asthma", "Initial_days", "TotalCharge"]]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Age	7.719070
1	gender_male	1.913295
2	VitD_levels	78.252486
3	Doc_visits	23.867293
4	HighBlood	1.862913
5	initial_admit_observ	1.974016
6	initial_admit_emerg	6.633095
7	Stroke	1.250241
8	Overweight	3.445122
9	Arthritis	1.631781
10	comp_risk_low	2.198155
11	comp_risk_medium	3.602866
12	Diabetes	1.442638
13	Hyperlipidemia	1.627488
14	BackPain	1.816659
15	Anxiety	1.556567
16	Asthma	1.409513
17	Initial_days	283.631467
18	TotalCharge	716.952843

Removed Total Charge VIF = 716.952843

```
#Removing TotalCharge (VIF=716.952843)
X = log_regression_df[["Age", "gender_male", "VitD_levels", "Doc_visits", "HighBlood",
               "initial_admit_observ", "initial_admit_emerg", "Stroke", "Overweight", "Arthritis", "comp_risk_low",
               "comp_risk_medium", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Asthma", "Initial_days"]]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Age	7.355059
1	gender_male	1.893716
2	VitD_levels	29.159622
3	Doc_visits	19.716755
4	HighBlood	1.688905
5	initial_admit_observ	1.950294
6	initial_admit_emerg	2.979739
7	Stroke	1.248229
8	Overweight	3.390835
9	Arthritis	1.553473
10	comp_risk_low	1.619017
11	comp_risk_medium	2.320652
12	Diabetes	1.372689
13	Hyperlipidemia	1.501689
14	BackPain	1.694892
15	Anxiety	1.471443
16	Asthma	1.405203
17	Initial_days	2.681348

Removed Vitamin D Levels VIF = 29.159622

```
#Removing VitD_levels (VIF=29.159622)
X = log_regression_df[["Age", "gender_male", "Doc_visits", "HighBlood",
                "initial_admit_observ", "initial_admit_emerg", "Stroke", "Overweight", "Arthritis", "comp_risk_low",
                "comp_risk_medium", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Asthma", "Initial_days"]]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Age	6.599388
1	gender_male	1.866637
2	Doc_visits	11.658344
3	HighBlood	1.675006
4	initial_admit_observ	1.900800
5	initial_admit_emerg	2.850068
6	Stroke	1.242692
7	Overweight	3.270227
8	Arthritis	1.543141
9	comp_risk_low	1.592940
10	comp_risk_medium	2.261173
11	Diabetes	1.369105
12	Hyperlipidemia	1.488771
13	BackPain	1.684707
14	Anxiety	1.460908
15	Asthma	1.395478
16	Initial_days	2.616080

Removed Doctor Visits VIF = 11.658344

```
#Removing Doc_visits is still above 10 (VIF=11.658344)
X = log_regression_df[["Age", "gender_male", "HighBlood", "initial_admit_observ", "initial_admit_emerg", "Stroke",
                       "Overweight", "Arthritis", "comp_risk_low", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma", "Initial_days"]]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Age	5.327377
1	gender_male	1.819802
2	HighBlood	1.648503
3	initial_admit_observ	1.792433
4	initial_admit_emerg	2.626581
5	Stroke	1.235400
6	Overweight	3.053095
7	Arthritis	1.526294
8	comp_risk_low	1.555918
9	comp_risk_medium	2.183598
10	Diabetes	1.353709
11	Hyperlipidemia	1.476697
12	BackPain	1.660268
13	Anxiety	1.445990
14	Asthma	1.387311
15	Initial_days	2.515542

All VIF values are below 10, so we will focus on p-values to further reduce the model. Any p-value above 0.10 will be eliminated.

```
#Backwards Elimination #1: Looking for p-value above 0.05
y = log_regression_df.Soft_drink
X = log_regression_df[["Age", "gender_male", "HighBlood", "initial_admit_observ", "initial_admit_emerg", "Stroke",
                       "Overweight", "Arthritis", "comp_risk_low", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma", "Initial_days"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569379
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9983
Method:                           MLE   Df Model:                          16
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:               0.001835
Time:                        14:43:50   Log-Likelihood:                -5693.8
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                    0.1809
==============================================================================================
                                coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
Age                         -0.0002      0.001     -0.214      0.831      -0.002       0.002
gender_male                 -0.0246      0.046     -0.537      0.592      -0.115       0.065
HighBlood                   -0.0279      0.047     -0.599      0.549      -0.119       0.064
initial_admit_observ        -0.0876      0.066     -1.327      0.185      -0.217       0.042
initial_admit_emerg          0.0569      0.056      1.019      0.308      -0.053       0.166
Stroke                       0.0114      0.057      0.199      0.842      -0.101       0.124
Overweight                  -0.0226      0.050     -0.448      0.654      -0.121       0.076
Arthritis                   -0.0341      0.048     -0.712      0.476      -0.128       0.060
comp_risk_low               -0.0193      0.064     -0.303      0.762      -0.144       0.106
comp_risk_medium             0.0295      0.052      0.566      0.571      -0.073       0.132
Diabetes                     0.0864      0.051      1.698      0.089      -0.013       0.186
Hyperlipidemia               0.0792      0.048      1.644      0.100      -0.015       0.174
BackPain                     0.0815      0.046      1.756      0.079      -0.009       0.173
Anxiety                      0.0782      0.049      1.605      0.108      -0.017       0.174
Asthma                       0.0491      0.050      0.976      0.329      -0.049       0.148
Initial_days                 0.0002      0.001      0.264      0.792      -0.001       0.002
const                       -1.1477      0.106    -10.803      0.000      -1.356      -0.939
==============================================================================================
```

Eliminated Stroke – p-value = 0.842

```
#Backwards Elimination #2: Removed Stroke variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["Age", "gender_male", "HighBlood", "initial_admit_observ", "initial_admit_emerg", "Overweight",
                       "Arthritis", "comp_risk_low", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma", "Initial_days"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569381
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9984
Method:                           MLE   Df Model:                           15
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001832
Time:                        14:44:03   Log-Likelihood:                -5693.8
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                    0.1401
==============================================================================================
                                coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
Age                          -0.0002      0.001     -0.211      0.833      -0.002       0.002
gender_male                  -0.0247      0.046     -0.537      0.591      -0.115       0.065
HighBlood                    -0.0279      0.047     -0.597      0.550      -0.119       0.064
initial_admit_observ         -0.0876      0.066     -1.326      0.185      -0.217       0.042
initial_admit_emerg           0.0568      0.056      1.018      0.309      -0.053       0.166
Overweight                   -0.0226      0.050     -0.448      0.654      -0.121       0.076
Arthritis                    -0.0343      0.048     -0.716      0.474      -0.128       0.060
comp_risk_low                -0.0193      0.064     -0.303      0.762      -0.144       0.106
comp_risk_medium              0.0295      0.052      0.566      0.571      -0.073       0.132
Diabetes                      0.0865      0.051      1.700      0.089      -0.013       0.186
Hyperlipidemia                0.0790      0.048      1.641      0.101      -0.015       0.173
BackPain                      0.0816      0.046      1.757      0.079      -0.009       0.173
Anxiety                       0.0781      0.049      1.603      0.109      -0.017       0.174
Asthma                        0.0491      0.050      0.977      0.329      -0.049       0.148
Initial_days                  0.0002      0.001      0.264      0.792      -0.001       0.002
const                        -1.1454      0.106    -10.846      0.000      -1.352      -0.938
==============================================================================================
```

Eliminated Age – p-value = 0.833

```
#Backwards Elimination #3: Removed Age variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["gender_male", "HighBlood", "initial_admit_observ", "initial_admit_emerg", "Overweight",
                       "Arthritis", "comp_risk_low", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma", "Initial_days"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569383
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9985
Method:                           MLE   Df Model:                           14
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001828
Time:                        14:44:08   Log-Likelihood:                -5693.8
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                    0.1054
==========================================================================================
                             coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------
gender_male               -0.0245      0.046     -0.534      0.593      -0.114       0.065
HighBlood                 -0.0280      0.047     -0.599      0.549      -0.119       0.064
initial_admit_observ      -0.0875      0.066     -1.324      0.186      -0.217       0.042
initial_admit_emerg        0.0570      0.056      1.020      0.308      -0.052       0.166
Overweight                -0.0225      0.050     -0.447      0.655      -0.121       0.076
Arthritis                 -0.0344      0.048     -0.718      0.473      -0.128       0.060
comp_risk_low             -0.0193      0.064     -0.302      0.762      -0.144       0.106
comp_risk_medium           0.0296      0.052      0.567      0.571      -0.073       0.132
Diabetes                   0.0865      0.051      1.699      0.089      -0.013       0.186
Hyperlipidemia             0.0790      0.048      1.640      0.101      -0.015       0.173
BackPain                   0.0814      0.046      1.753      0.080      -0.010       0.172
Anxiety                    0.0780      0.049      1.602      0.109      -0.017       0.174
Asthma                     0.0490      0.050      0.975      0.330      -0.050       0.148
Initial_days               0.0002      0.001      0.260      0.795      -0.001       0.002
const                     -1.1579      0.087    -13.249      0.000      -1.329      -0.987
==========================================================================================
```

Eliminated Initial Days – p-value = 0.795

```
#Backwards Elimination #4: Removed Initial_days variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["gender_male", "HighBlood", "initial_admit_observ", "initial_admit_emerg", "Overweight",
                       "Arthritis", "comp_risk_low", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569386
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9986
Method:                           MLE   Df Model:                           13
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001822
Time:                        14:44:13   Log-Likelihood:                -5693.9
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.07725
==============================================================================================
                                coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
gender_male              -0.0244      0.046     -0.532      0.594      -0.114       0.066
HighBlood                -0.0280      0.047     -0.600      0.548      -0.119       0.063
initial_admit_observ     -0.0875      0.066     -1.325      0.185      -0.217       0.042
initial_admit_emerg       0.0568      0.056      1.017      0.309      -0.053       0.166
Overweight               -0.0226      0.050     -0.450      0.653      -0.121       0.076
Arthritis                -0.0342      0.048     -0.713      0.476      -0.128       0.060
comp_risk_low            -0.0190      0.064     -0.299      0.765      -0.144       0.106
comp_risk_medium          0.0296      0.052      0.567      0.571      -0.073       0.132
Diabetes                  0.0864      0.051      1.698      0.090      -0.013       0.186
Hyperlipidemia            0.0789      0.048      1.639      0.101      -0.015       0.173
BackPain                  0.0816      0.046      1.758      0.079      -0.009       0.173
Anxiety                   0.0782      0.049      1.605      0.109      -0.017       0.174
Asthma                    0.0488      0.050      0.971      0.331      -0.050       0.147
const                    -1.1501      0.082    -14.022      0.000      -1.311      -0.989
==============================================================================================
```

Eliminated Low Complication Risk – p-value = 0.765

```
#Backwards Elimination #5: Removed comp_risk_low variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["gender_male", "HighBlood", "initial_admit_observ", "initial_admit_emerg", "Overweight",
                       "Arthritis", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569391
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9987
Method:                           MLE   Df Model:                           12
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001814
Time:                        14:44:22   Log-Likelihood:                -5693.9
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.05499
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
gender_male             -0.0244      0.046     -0.532      0.595      -0.114       0.066
HighBlood               -0.0277      0.047     -0.593      0.553      -0.119       0.064
initial_admit_observ    -0.0878      0.066     -1.330      0.184      -0.217       0.042
initial_admit_emerg      0.0566      0.056      1.013      0.311      -0.053       0.166
Overweight              -0.0226      0.050     -0.449      0.653      -0.121       0.076
Arthritis               -0.0343      0.048     -0.717      0.473      -0.128       0.060
comp_risk_medium         0.0369      0.046      0.803      0.422      -0.053       0.127
Diabetes                 0.0863      0.051      1.696      0.090      -0.013       0.186
Hyperlipidemia           0.0790      0.048      1.639      0.101      -0.015       0.173
BackPain                 0.0814      0.046      1.753      0.080      -0.010       0.172
Anxiety                  0.0782      0.049      1.605      0.108      -0.017       0.174
Asthma                   0.0487      0.050      0.969      0.333      -0.050       0.147
const                   -1.1572      0.079    -14.741      0.000      -1.311      -1.003
============================================================================================
```

Eliminated Overweight – p-value = 0.653

```
#Backwards Elimination #6: Removed Overweight variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["gender_male", "HighBlood", "initial_admit_observ", "initial_admit_emerg",
                       "Arthritis", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569401
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9988
Method:                           MLE   Df Model:                           11
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001797
Time:                        14:44:25   Log-Likelihood:                -5694.0
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.03898
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
gender_male    -0.0244      0.046     -0.531      0.595      -0.114       0.066
HighBlood      -0.0282      0.047     -0.605      0.545      -0.120       0.063
initial_admit_observ    -0.0879      0.066     -1.331      0.183      -0.217       0.042
initial_admit_emerg      0.0567      0.056      1.016      0.310      -0.053       0.166
Arthritis      -0.0344      0.048     -0.719      0.472      -0.128       0.059
comp_risk_medium      0.0366      0.046      0.795      0.427      -0.054       0.127
Diabetes       0.0865      0.051      1.699      0.089      -0.013       0.186
Hyperlipidemia      0.0791      0.048      1.642      0.101      -0.015       0.173
BackPain       0.0811      0.046      1.749      0.080      -0.010       0.172
Anxiety        0.0784      0.049      1.610      0.107      -0.017       0.174
Asthma         0.0484      0.050      0.963      0.336      -0.050       0.147
const         -1.1729      0.070    -16.666      0.000      -1.311      -1.035
==============================================================================
```

Eliminated Gender_Male – p-value = 0.595

```
#Backwards Elimination #7: Removed gender_male variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["HighBlood", "initial_admit_observ", "initial_admit_emerg",
                       "Arthritis", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569415
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9989
Method:                           MLE   Df Model:                           10
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001772
Time:                        14:44:29   Log-Likelihood:                -5694.2
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.02729
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
HighBlood                   -0.0283      0.047     -0.608      0.543      -0.120       0.063
initial_admit_observ        -0.0871      0.066     -1.319      0.187      -0.217       0.042
initial_admit_emerg          0.0574      0.056      1.028      0.304      -0.052       0.167
Arthritis                   -0.0346      0.048     -0.723      0.470      -0.129       0.059
comp_risk_medium             0.0364      0.046      0.790      0.429      -0.054       0.127
Diabetes                     0.0865      0.051      1.700      0.089      -0.013       0.186
Hyperlipidemia               0.0787      0.048      1.634      0.102      -0.016       0.173
BackPain                     0.0815      0.046      1.757      0.079      -0.009       0.172
Anxiety                      0.0786      0.049      1.614      0.107      -0.017       0.174
Asthma                       0.0483      0.050      0.961      0.337      -0.050       0.147
const                       -1.1849      0.067    -17.770      0.000      -1.316      -1.054
============================================================================================
```

Eliminated High Blood Pressure – p-value = 0.543

```
#Backwards Elimination #8: Removed HighBlood variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ", "initial_admit_emerg",
                       "Arthritis", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569433
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9990
Method:                           MLE   Df Model:                            9
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001739
Time:                        14:44:34   Log-Likelihood:                -5694.3
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.01890
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
initial_admit_observ          -0.0873      0.066     -1.323      0.186      -0.217       0.042
initial_admit_emerg            0.0573      0.056      1.027      0.305      -0.052       0.167
Arthritis                     -0.0348      0.048     -0.727      0.467      -0.129       0.059
comp_risk_medium               0.0359      0.046      0.782      0.434      -0.054       0.126
Diabetes                       0.0867      0.051      1.704      0.088      -0.013       0.186
Hyperlipidemia                 0.0790      0.048      1.640      0.101      -0.015       0.173
BackPain                       0.0814      0.046      1.755      0.079      -0.010       0.172
Anxiety                        0.0784      0.049      1.609      0.108      -0.017       0.174
Asthma                         0.0481      0.050      0.956      0.339      -0.050       0.147
const                         -1.1960      0.064    -18.651      0.000      -1.322      -1.070
==============================================================================================
```

Eliminated arthritis – p-value = 0.467

```
#Backwards Elimination #9: Removed Arthritis variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ", "initial_admit_emerg", "comp_risk_medium", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569460
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9991
Method:                           MLE   Df Model:                            8
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001693
Time:                        14:44:45   Log-Likelihood:                -5694.6
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.01326
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
initial_admit_observ          -0.0874      0.066     -1.323      0.186      -0.217       0.042
initial_admit_emerg            0.0573      0.056      1.027      0.305      -0.052       0.167
comp_risk_medium               0.0354      0.046      0.769      0.442      -0.055       0.126
Diabetes                       0.0864      0.051      1.697      0.090      -0.013       0.186
Hyperlipidemia                 0.0792      0.048      1.645      0.100      -0.015       0.174
BackPain                       0.0820      0.046      1.769      0.077      -0.009       0.173
Anxiety                        0.0780      0.049      1.601      0.109      -0.017       0.173
Asthma                         0.0483      0.050      0.962      0.336      -0.050       0.147
const                         -1.2083      0.062    -19.527      0.000      -1.330      -1.087
==============================================================================================
```

Eliminated Medium Complication Risk – p-value = 0.442

```
#Backwards Elimination #10: Removed comp_risk_medium variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ", "initial_admit_emerg", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety", "Asthma"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569490
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9992
Method:                           MLE   Df Model:                            7
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001641
Time:                        14:44:52   Log-Likelihood:                -5694.9
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                  0.009099
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
initial_admit_observ          -0.0863      0.066     -1.307      0.191      -0.216       0.043
initial_admit_emerg            0.0575      0.056      1.030      0.303      -0.052       0.167
Diabetes                       0.0863      0.051      1.696      0.090      -0.013       0.186
Hyperlipidemia                 0.0796      0.048      1.653      0.098      -0.015       0.174
BackPain                       0.0817      0.046      1.761      0.078      -0.009       0.173
Anxiety                        0.0781      0.049      1.604      0.109      -0.017       0.174
Asthma                         0.0486      0.050      0.967      0.333      -0.050       0.147
const                         -1.1927      0.058    -20.418      0.000      -1.307      -1.078
==============================================================================================
```

Eliminated Asthma – p-value = 0.333

```
#Backwards Elimination #11: Removed Asthma variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ", "initial_admit_emerg", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569536
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9993
Method:                           MLE   Df Model:                            6
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001559
Time:                        14:44:58   Log-Likelihood:                -5695.4
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                  0.006776
==============================================================================================
                                 coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
initial_admit_observ          -0.0864      0.066     -1.308      0.191      -0.216       0.043
initial_admit_emerg            0.0571      0.056      1.023      0.306      -0.052       0.166
Diabetes                       0.0871      0.051      1.711      0.087      -0.013       0.187
Hyperlipidemia                 0.0792      0.048      1.644      0.100      -0.015       0.174
BackPain                       0.0823      0.046      1.774      0.076      -0.009       0.173
Anxiety                        0.0787      0.049      1.615      0.106      -0.017       0.174
const                         -1.1788      0.057    -20.835      0.000      -1.290      -1.068
==============================================================================================
```

Eliminated Initial Admin Emergency – p-value = 0.306

```
#Backwards Elimination #12: Removed initial_admit_emerg variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ", "Diabetes", "Hyperlipidemia",
                       "BackPain", "Anxiety"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569589
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:              Soft_drink   No. Observations:                10000
Model:                           Logit   Df Residuals:                     9994
Method:                            MLE   Df Model:                            5
Date:                 Mon, 24 Jun 2024   Pseudo R-squ.:                0.001467
Time:                         14:45:09   Log-Likelihood:                -5695.9
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                  0.005019
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
initial_admit_observ        -0.1247      0.054     -2.301      0.021      -0.231      -0.019
Diabetes                     0.0865      0.051      1.701      0.089      -0.013       0.186
Hyperlipidemia               0.0800      0.048      1.662      0.096      -0.014       0.174
BackPain                     0.0826      0.046      1.782      0.075      -0.008       0.174
Anxiety                      0.0790      0.049      1.623      0.104      -0.016       0.174
const                       -1.1408      0.043    -26.809      0.000      -1.224      -1.057
============================================================================================
```

Eliminated anxiety – p-value = 0.104

```
#Backwards Elimination #13: Removed Anxiety variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ", "Diabetes", "Hyperlipidemia", "BackPain"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569720
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:              Soft_drink   No. Observations:                10000
Model:                           Logit   Df Residuals:                     9995
Method:                            MLE   Df Model:                            4
Date:                 Mon, 24 Jun 2024   Pseudo R-squ.:                0.001237
Time:                         14:45:18   Log-Likelihood:                -5697.2
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                  0.006934
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
initial_admit_observ        -0.1251      0.054     -2.308      0.021      -0.231      -0.019
Diabetes                     0.0863      0.051      1.698      0.090      -0.013       0.186
Hyperlipidemia               0.0789      0.048      1.640      0.101      -0.015       0.173
BackPain                     0.0833      0.046      1.797      0.072      -0.008       0.174
const                       -1.1149      0.039    -28.339      0.000      -1.192      -1.038
============================================================================================
```

Eliminated Hyperlipidemia – p-value = 0.101

```
#Backwards Elimination #14: Removed Hyperlipidemia variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ", "Diabetes", "BackPain"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569854
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9996
Method:                           MLE   Df Model:                            3
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:                0.001002
Time:                        14:45:22   Log-Likelihood:                -5698.5
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                  0.009585
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
initial_admit_observ    -0.1259      0.054     -2.323      0.020      -0.232      -0.020
Diabetes                 0.0873      0.051      1.716      0.086      -0.012       0.187
BackPain                 0.0832      0.046      1.796      0.073      -0.008       0.174
const                   -1.0879      0.036    -30.519      0.000      -1.158      -1.018
============================================================================================
```

Eliminated Diabetes – p-value = 0.086

```
#Backwards Elimination #15: Removed Diabetes variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ", "BackPain"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.570000
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9997
Method:                           MLE   Df Model:                            2
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:               0.0007457
Time:                        14:45:32   Log-Likelihood:                -5700.0
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.01421
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
initial_admit_observ    -0.1258      0.054     -2.322      0.020      -0.232      -0.020
BackPain                 0.0822      0.046      1.773      0.076      -0.009       0.173
const                   -1.0633      0.033    -32.700      0.000      -1.127      -1.000
============================================================================================
```

Eliminated BackPain – p-value = 0.076

```
#Backwards Elimination #16: Removed BackPain variable (p value > 0.05)
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.570157
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9998
Method:                           MLE   Df Model:                            1
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:               0.0004708
Time:                        14:45:40   Log-Likelihood:                -5701.6
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.02047
===========================================================================================
                              coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------
initial_admit_observ    -0.1248      0.054     -2.304      0.021      -0.231      -0.019
const                   -1.0293      0.026    -39.423      0.000      -1.080      -0.978
===========================================================================================
```

All P-values are below 0.05, so this is our final reduced model.

```
#Final Reduced Log Regression Model
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.570157
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9998
Method:                           MLE   Df Model:                            1
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:               0.0004708
Time:                        14:45:47   Log-Likelihood:                -5701.6
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.02047
===========================================================================================
                              coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------
initial_admit_observ    -0.1248      0.054     -2.304      0.021      -0.231      -0.019
const                   -1.0293      0.026    -39.423      0.000      -1.080      -0.978
===========================================================================================
```

E.1. Explain your data analysis process by comparing the initial logistic regression model and the reduced logistic regression model.

As stated in my task 1 submission (Hosey 2024), looking at the initial and final model, we can see that several independent variables were removed from the data to maintain homoscedasticity and decrease multicollinearity between variables. This increases the validity of the model and the accuracy of its predictions. The VIF used was to decrease multicollinearity; any VIF value above 10 was removed to

ensure that no two variables were statistically identical. In addition, P-values (Backwards Stepwise Elimination) were used to maintain the model's homoscedasticity. In this research project, a p-value of 0.05 was used, to backwards eliminate independent variables.

The screenshots below show that the initial Model's LLR p-value was 0.1865, and the final models were 0.02047. This indicates that the final model is more precise and accurate in its predictions due to those reductions and indicates a statistical significance between the variables.

As seen in F1's screenshots, the confusion matrix indicates that the final model would predict 2239 inaccurate predictions and 761 correct predictions. It would be worrisome if a company used my model to predict patient outcomes. However, this is not the case, and a company would use more than a single day's data to predict patient outcomes if they do not want a lawsuit.

Initial model (after removing the gender_nonbinary column as the very first model had results with Nan values)

```
#Second run of Log Model - Without gender_nonbinary Variable do to high VIF hoping for no NaN values in results table.
#Set dependent variable for Y
y = log_regression_df.Soft_drink

#Set multiple independent variables for X and add constant
X = log_regression_df[["Age", "gender_male", "VitD_levels", "Doc_visits", "HighBlood",
               "initial_admit_observ", "initial_admit_emerg", "Stroke", "Overweight", "Arthritis", "comp_risk_low",
               "comp_risk_medium", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Asthma", "Initial_days", "TotalCharge"]]

X = sm.add_constant(X)  # This will add a constant column

#Double checking that all X columns are numeric
X = X.apply(pd.to_numeric, errors='coerce')

#Fit and print the model
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.569213
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:             Soft_drink   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9980
Method:                           MLE   Df Model:                           19
Date:                Tue, 18 Jun 2024   Pseudo R-squ.:                0.002126
Time:                        18:05:02   Log-Likelihood:                -5692.1
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                    0.1865
==============================================================================
                             coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const                     0.3312      1.309      0.253      0.800      -2.234       2.897
Age                      -0.0003      0.001     -0.235      0.815      -0.002       0.002
gender_male              -0.0248      0.046     -0.541      0.588      -0.115       0.065
VitD_levels               0.0044      0.011      0.388      0.698      -0.018       0.027
Doc_visits                0.0260      0.022      1.184      0.236      -0.017       0.069
HighBlood                 0.0538      0.078      0.689      0.491      -0.099       0.207
initial_admit_observ     -0.0917      0.066     -1.388      0.165      -0.221       0.038
initial_admit_emerg       0.4272      0.288      1.484      0.138      -0.137       0.992
Stroke                    0.0112      0.057      0.195      0.845      -0.101       0.123
Overweight               -0.0239      0.050     -0.475      0.635      -0.123       0.075
Arthritis                 0.0193      0.063      0.307      0.759      -0.104       0.142
comp_risk_low            -0.3190      0.237     -1.345      0.178      -0.784       0.146
comp_risk_medium         -0.2716      0.235     -1.156      0.248      -0.732       0.189
Diabetes                  0.1407      0.066      2.142      0.032       0.012       0.269
Hyperlipidemia            0.1490      0.071      2.108      0.035       0.010       0.288
BackPain                  0.1441      0.067      2.164      0.030       0.014       0.275
Anxiety                   0.1407      0.068      2.068      0.039       0.007       0.274
Asthma                    0.0503      0.050      1.000      0.318      -0.048       0.149
Initial_days              0.0599      0.045      1.322      0.186      -0.029       0.149
TotalCharge              -0.0007      0.001     -1.317      0.188      -0.002       0.000
==============================================================================
```

Final Reduced Model

```
#Final Reduced Log Regression Model
y = log_regression_df.Soft_drink
X = log_regression_df[["initial_admit_observ"]].assign(const=1)
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.570157
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:              Soft_drink   No. Observations:                10000
Model:                           Logit   Df Residuals:                     9998
Method:                            MLE   Df Model:                            1
Date:                Mon, 24 Jun 2024   Pseudo R-squ.:               0.0004708
Time:                        14:45:47   Log-Likelihood:                -5701.6
converged:                       True   LL-Null:                       -5704.3
Covariance Type:            nonrobust   LLR p-value:                   0.02047
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
initial_admit_observ    -0.1248      0.054     -2.304      0.021      -0.231      -0.019
const                   -1.0293      0.026    -39.423      0.000      -1.080      -0.978
============================================================================================
```

E.2. Provide the output and all analysis calculations: confusion matrix and accuracy calculations.

Small Test Sample Set of 0.3

```
#Confusion Matrix on Test Size of 0.3
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test, y_test)))
final_matrix = confusion_matrix(y_test, y_pred)
print(final_matrix)
```

```
Accuracy of logistic regression classifier on test set: 0.75
[[2239    0]
 [ 761    0]]
```

Predicts 2239 inaccurate predictions and 761 correct predictions.

Bigger Test Sample Set of 0.8

```
#Confusion Matrix on Test Size of 0.8
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.8, random_state=42)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test, y_test)))
final_matrix = confusion_matrix(y_test, y_pred)
print(final_matrix)
```

```
Accuracy of logistic regression classifier on test set: 0.74
[[5940    0]
 [2060    0]]
```

Predicts 5940 inaccurate predictions and 2060 correct predictions.

Model Parameters

```
result.params

initial_admit_observ   -0.124824
const                  -1.029296
dtype: float64
```

```
#F1 Calculate odds ratios for each coefficient and the change in odds for each independent variable in the reduced model
#Odds ratio and change in odds for initial_admit_observ
initial_admit_observ_odds_ratio = np.exp(-0.124824)
initial_admit_observ_change_in_odds = (initial_admit_observ_odds_ratio - 1) * 100
print(f"The odds ratio for initial_admit_observ is {round(initial_admit_observ_odds_ratio, 4)}. "
      f"Given this, the change in odds for being a Soft_drink drinker is {round(initial_admit_observ_change_in_odds, 4)}.")
```

```
The odds ratio for initial_admit_observ is 0.8827. Given this, the change in odds for being a Soft_drink drinker is -11.7348.
```

E.3. Provide an executable error-free copy of the code.

See the attached files from my task submission.

Part V: Data Summary and Implications

F.1. Discuss the results of your data analysis.

- ***a regression equation for the reduced model***

Log Odds *(Soft_drink)* = - 1.029296 - 0.1259 *(initial_admit_observ)*

- ***an interpretation of the coefficients of the reduced model***

After completing this, I realized that predicting whether you will drink soft drinks doesn't make sense. Still, I continued through as it does allow you to see whether your chances for back pain, diabetes, or being admitted under observations if you do drink soft drinks.

- The odds ratio for your initial admission for observation is 0.8827. Given this, the chance in odds for also being a soft drink drinker is -11.7348.

- ***the statistical and practical significance of the reduced model***

As stated in E1, the initial model p-value was 0.1865, and the final model was 0.02047. This means the final model is below the recommended 0.05 value, so there is statistical significance between the dependent and independent variables.

As for practical significance (as I stated above), this research project would not be the best fit for predicting patient outcomes in an honest company's function. In addition to practicality, the number of incorrect predictions is relatively high, so the chances of patients who drink soft drinks being hospitalized for serious reasons are not seen or heard. This is due to several reasons that will be addressed in the next section. Also, the sample size is relatively small (yes, even at 10,000 entries). This data was from one day of a hospital operation; this model would work better under a massive data set with more than one day of information.

- ***the limitations of the data analysis***

Several limitations hinder the accurate analysis of this model. Again, this dataset is one day (24 hours) of hospital admissions. The dataset contains only patients aged 18-89 (due to HIPAA or other policies); this information would be available if I worked within the hospital company, and the sample size would be much more significant.

Using a p-value below 0.05 this eliminates all but one independent variable. So, the model is not the best to predict whether soft drinkers are actually doing harm to their health.

The sample size is small for creating a predictive model; if there were more data or the ability to use upsampling via SMOTE (Li 2017), the sample size would be better suited to predicting more accurate results.

F.2. Recommend a course of action based on your results.

Recommendations for the next steps would be to increase the dataset size and change the research question to something more meaningful to a hospital's insights (not my curiosities). This research question is not horrible and could be used to gain insights into whether soft drinks are detrimental to patient health, as there is a lot of mixed (and incorrect information) out in the media to spin a story.

A big recommendation would be to change the dataset, as it is from a single day of operation and is limited to what information can be given out about patients (due to specific policies and laws). The more detailed the dataset is, the higher the likelihood of better predictions with the model.

Part VI: Demonstration

G. Panopto video

See the submission attachments for the code video.

H. Code Sources

Li, Susan. (September 28, 2017). Building a logistic regression in Python: Step by step. Towards Data Science. https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8#:~:text=Over%2Dsampling%20using%20SMOTE

Sewell, William. (April 2023). D208 Predictive Modeling Webinar Episode 1. Panopto. Retrieved from https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william_sewell_wgu_edu/ER_vJMbYtxJGpxImpZ0DUQcBoVcORYKanFVKNKFcEXkRow?rtime=03cpw43G2kg

I. Additional Sources

Bobbitt, Z. (2020, October 13). The 6 assumptions of logistic regression (with examples). Statology. Retrieved June 21, 2024, from https://www.statology.org/assumptions-of-logistic-regression/

Bobbitt, Z. (2021, May 19). How to interpret an odds ratio less than 1. Statology. Retrieved June 21, 2024, from https://www.statology.org/interpret-odds-ratio-less-than-1/

Hosey, Jessica. (2024). Performance Assessment 1, D208 – Predictive Modeling. [Unpublished manuscript]. WGU.

IBM. (n.d.). What is logistic regression? Retrieved June 21, 2024, from https://www.ibm.com/topics/logistic-regression

Middleton, Keiona. (October 2022). D208 - Webinar: Getting Started with D208 Part II [Video]. Panopto. Retrieved from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=09b8fdbb-a374-452b-ba53-af39001ff3f3

Middleton, Keiona. (November 2022). D208 - Webinar: Getting Started with D208 Part I [Video]. Panopto. Retrieved from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=15e09c73-c5aa-439d-852f-af47001b8970

pandas. (2024). Returning a view versus a copy – Use of .iloc. In pandas documentation. Retrieved from https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

Tripathi, Ashutosh. (Jun 10, 2019). "Feature Selection Techniques in Regression Model." Towards Data Science. Retrieved from https://towardsdatascience.com/feature-selection-techniques-in-regression-model-26878fe0e24e