**Matt Boruff**

**MSDA D208 Task 2**

**Logistic Regression**

**Dr Middleton**

**Part I: Research Question**

A.  Describe the purpose of this data analysis by doing the following:

1.  Summarize **one** research question that is relevant to a real-world organizational situation captured in the data set you have selected and that you will answer using logistic regression.

Our research question at this time will be:

With the data provided, are there any possible predictors of a readmission?

2.  Define the goals of the data analysis.

With this analysis we are attempting to find possible predictors that may indicate if a patient may have to be readmitted to the hospital after an initial stay. Using a logistic regression, we can determine decent independent variables to go along with a strong dependent variable. Also, this will provide us with a Logistic Equation that will be at the end of this analysis.

**Part II: Method Justification**

B.  Describe logistic regression methods by doing the following:

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

There are several assumptions associated with logistic regression models. Here are four of the most important ones:

* Linearity of the logit: The relationship between the predictor variables and the log odds of the outcome variable should be linear. This assumption is similar to the linearity assumption in linear regression.
* Independence of observations: Each observation in the dataset should be independent of all other observations. In other words, the presence or absence of the outcome variable in one observation should not influence the presence or absence of the outcome variable in any other observation.
* Absence of multicollinearity: There should not be a high degree of correlation between the predictor variables. Multicollinearity can make it difficult to estimate the coefficients of the logistic regression model accurately and can lead to unstable estimates.
* The error terms should be independent and identically distributed: The error terms in logistic regression should be independent and identically distributed (i.e., have the same variance). This assumption is similar to the homoscedasticity assumption in linear regression.

It's important to note that these assumptions are idealistic, and it's unlikely that they will be met exactly in practice. However, they provide a useful framework for understanding the behavior of logistic regression models and can help identify potential issues that may arise when fitting the model.

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

We will be performing this regression using Python, which for myself is a familiar language and easier to use. However here are two more benefits common for Python usage.

* Accessibility and ease of use: Python has a large and active community of users who have developed a wide range of open-source libraries and packages for data analysis and machine learning. Some of the most popular libraries for logistic regression in Python include NumPy, Pandas, Scikit-learn, and Statsmodels. These libraries provide a wide range of functions and tools for preprocessing data, building models, and evaluating model performance. Additionally, Python has a relatively simple and intuitive syntax, making it accessible even to users with limited programming experience.
* Flexibility and scalability: Python is a versatile language that can be used for a wide range of tasks, from simple data analysis to complex machine learning applications. It is also highly scalable, meaning that it can be used to build models that can handle very large datasets. Additionally, Python is compatible with a wide range of hardware and software platforms, making it a good choice for building models that need to run on multiple devices or be integrated with other systems. Finally, because Python is an open-source language, it is easy to customize and extend using libraries and packages developed by the user community.

3.  Explain why logistic regression is an appropriate technique to analyze the research question summarized in part I.

Logistic regression is a popular choice for modeling binary outcomes, where the dependent variable can take on one of two possible values (e.g., yes/no, success/failure, etc.). Here are some reasons why logistic regression might be a good choice for a given problem:

* Interpretable results: Logistic regression models are relatively simple and easy to interpret, making them a good choice for situations where it is important to understand how the predictor variables are related to the outcome variable.
* Robustness to noise: Logistic regression models are robust to the inclusion of noisy or irrelevant variables, making them a good choice for situations where there may be many predictor variables or where the relationship between the predictor variables and the outcome variable is not well understood.
* Flexibility: Logistic regression can be used with a wide range of predictor variables, including continuous, categorical, and binary variables. This makes it a flexible choice for a wide range of problems.
* Data availability: Logistic regression requires relatively little data to produce reliable results. This makes it a good choice for situations where data are limited or where data collection is expensive or time-consuming.
* Computational efficiency: Logistic regression is a computationally efficient method, meaning that it can handle large datasets and produce results quickly.

Overall, logistic regression can be a good choice for situations where the goal is to model binary outcomes, and where interpretability, robustness to noise, and flexibility are important considerations. However, it may not be the best choice for all problems, and it is important to carefully consider the strengths and limitations of the method when selecting an appropriate modeling approach.

**Part III: Data Preparation**

C.  Summarize the data preparation process for logistic regression by doing the following:

1.  Describe your data cleaning goals and the steps used to clean the data to achieve the goals that align with your research question including the annotated code.

Prior to starting the regression, the data will need to be prepped and cleaned. We can start by checking for any null values inside the data and correcting them if any are present. Once that check is complete, we can then begin to look for and mitigate any duplicate values in the data set as well. After the data is “cleaned” and ready for use at this point, we can begin looking at the data and considering what can be combined and what can be converted to numerical data if needed.

Code:

#checking for nulls and duplicates in the dataframe

Df.isna().any()

Df.duplicated().any()

#dropping survey questions and dempgraphic columns

df.drop(['Item8', 'Item7', 'Item6', 'Item5', 'Item4', 'Item3', 'Item2', 'Item1', 'City', 'State', 'County', 'Lat', 'Lng', 'Population', 'TimeZone'], axis=1, inplace=True)

#dropping more unneeded columns

df.drop(['Additional\_charges', 'TotalCharge', 'Services', 'Interaction', 'UID', 'Customer\_id', 'Job', 'CaseOrder', 'Zip'], axis=1, inplace=True)

#consolidate Marital data to married/not married (divorce, separated, never married)

df['Marital'] = df['Marital'].replace(['Divorced','Widowed','Separated','Never Married'],'NotMarried')

2.  Describe the dependent variable and all independent variables using summary statistics that are required to answer the research question, including a screenshot of the summary statistics output for each of these variables.

Our target variable will be the column, ReAdmis. This is a categorical, yes/no, variable that we can use to base all of our independent variables off of. ReAdmis is also what is used to explain to us if the patient was readmitted to the hospital after an initial stay. To gather what may be good independent variables, we can use a summary statistic, keying in on the P-Value and coefficients to help identify the predictors for our regression analysis. Next are the variables we will be using for this regression as well as reducing, later on.

1. Area is categorical: ['Suburban' 'Urban' 'Rural'].

2. Children is numerical

3. Age is numerical

4. Income is numerical

5. Marital is categorical: ['NotMarried' 'Married'].

6. Gender is categorical: ['Male' 'Female' 'Nonbinary'].

7. VitD\_levels is numerical

8. Doc\_visits is numerical

9. Full\_meals\_eaten is numerical

10. vitD\_supp is numerical

11. Soft\_drink is categorical: ['No' 'Yes'].

12. Initial\_admin is categorical: ['Emergency Admission' 'Elective Admission' 'Observation Admission'].

13. HighBlood is categorical: ['Yes' 'No'].

14. Stroke is categorical: ['No' 'Yes'].

15. Complication\_risk is categorical: ['Medium' 'High' 'Low'].

16. Overweight is categorical: ['No' 'Yes'].

17. Arthritis is categorical: ['Yes' 'No'].

18. Diabetes is categorical: ['Yes' 'No'].

19. Hyperlipidemia is categorical: ['No' 'Yes'].

20. BackPain is categorical: ['Yes' 'No'].

21. Anxiety is categorical: ['Yes' 'No'].

22. Allergic\_rhinitis is categorical: ['Yes' 'No'].

23. Reflux\_esophagitis is categorical: ['No' 'Yes'].

24. Asthma is categorical: ['Yes' 'No'].

25. Initial\_days is numerical

The summary statistics are as follows:

A picture containing text, screenshot, font, number

Description automatically generated

Numerical Data Summary Statistics

A screenshot of a graph

Description automatically generated with low confidence

Our categorical data is visualized in Section 4 during our Univariate analysis. A summary of these variables indicate that Area, Reflux\_esophagitis, BackPain and Gender are normally distributed, not including the Non Binary response in Gender. As well as the number of responses equal to No for ReAdmis are more abundant than the answer of Yes, which keeps with the dispersion inline of the variable itself.

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

Our Univariate visualizations start with looking at the data for ReAdmis Yes/No

#Begin Univariate and Bivariate analysis

print(df[target].value\_counts())

sns.countplot(x=target, data=df, palette= 'hls')

plt.show()

A picture containing text, screenshot, diagram, plot

Description automatically generated

Next, we will checking the values of the categorical values

#Categorical graphs

fig=plt.figure(figsize=(10, 20))

for i, col in enumerate(C\_data):

if col != target:

plt.subplot(10, 3, i+1)

ax = sns.countplot(y=col, data=df)

fig.tight\_layout(h\_pad=4, w\_pad=4)

plt.title('Categorical Data')

plt.show()

A screenshot of a graph

Description automatically generated with low confidence

We can see that the ReAdmis variable has a skewed population towards not being readmitted.

We can now visualize the Bivariate analysis by comparing the target variable of ReAdmis to other categorical variables we have identified.

A picture containing text, screenshot, diagram, plot

Description automatically generatedA picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generatedA picture containing text, screenshot, diagram, line

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generatedA picture containing text, screenshot, plot, diagram

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generatedA picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generatedA picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generatedA picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generated

From this analysis visualizations all the variables, minus Area, maybe good predictors of the outcome. These should be included for our regression. We are now going to move out Target Variable to ReAdmis\_Yes as we are evaluating the impact on of the independent variables on the new target.

In the chart below we compare the outcomes of the variables against the Yes and No of the ReAdmis target variable.

A picture containing text, screenshot, font, number

Description automatically generated

We can see from the above, that for the variables listed, all but Initial\_days, are fairly equal and normally distributed. Income for the non-readmitted patients, is slightly higher. While those with a lower initial days were less likely to be readmitted. Below is the table for Numeric Statistics

A screenshot of a graph

Description automatically generated with low confidence

We can then visualize the histograms of the numerical data. This can help with identifying which numerical variables could have a higher chance of impact on the target variable

A picture containing diagram, plot, text, line

Description automatically generated

4.  Describe your data transformation goals that align with your research question and the steps used to transform the data to achieve the goals, including the annotated code.

The goal of our data transformation will be to make it easier to compare and contrast against the target variable of ReAdmis\_Yes. What we will need to transform at this point would be the categorical data to a binary data set. This is achieved by the following commands:

for c in C\_data:

if c in df.columns:

df = pd.get\_dummies(df, columns=[c], drop\_first=True)

pred\_vari = df.select\_dtypes(include='uint8').columns.tolist()

pred\_vari

At this point we can then remove some more unneeded data, mainly ReAdmis\_No

#kick unwanted data

col\_to\_be\_dropped = ['ReAdmis\_N']

#print the list of data to be removed

print('data to be removed: {}'.format(col\_to\_be\_dropped))

#loop the list if its in the current df drop the column

for c in col\_to\_be\_dropped:

if c in df.columns:

df.drop(columns=c, inplace=True)

print('Data named [{}] has been removed.'.format(c))

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

#Copy of prepped data set

df.to\_csv('medical\_clean\_update\_T2.csv')

**Part IV: Model Comparison and Analysis**

D.  Compare an initial and a reduced logistic regression model by doing the following:

1.  Construct an initial logistic regression model from all independent variables that were identified in part C2.

#import all libraries needed

import pandas as pd

import numpy as np

import scipy.stats as stats

import statsmodels.api as sm

import statsmodels.formula.api as smf

from IPython.core.display import HTML

from IPython.display import display

Import matplotlib.pyplot as plt

plt.rc("font", size=14)

from sklearn.metrics import confusion\_matrix, accuracy\_score

from sklearn import preprocessing

from sklearn.decomposition import PCA

from sklearn.linear\_model import LogisticRegression

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

from sklearn.metrics import roc\_auc\_score

from sklearn.metrics import roc\_curve

from sklearn.metrics import classification\_report

from sklearn import metrics

import seaborn as sns

from imblearn.over\_sampling import SMOTE

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import LogisticRegression

sns.set(style="white")

sns.set(style="whitegrid", color\_codes=True)

#set columns options to see all columns

pd.set\_option('display.max\_columns', None)

df = pd.read\_csv('medical\_clean.csv')

df

From our initial look at this, the data is imbalanced. To help with this, we will be running a SMOTE balance on data set. We will split the data into a training, test set to work with and prep for more evaluation. Once we have run the SMOTE analysis, we will be conducting an RFE(Recursive Feature Elimination) to rank the variables which will have the highest potential impact on the target variable.

# Rebalance of prepared data using SMOTE oversample

os = SMOTE(random\_state=0)

X = df.loc[:, df.columns != 'ReAdmis\_Yes']

y = df.loc[:, df.columns == 'ReAdmis\_Yes']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,test\_size=0.3, random\_state=888)

columns = X\_train.columns

os\_data\_X,os\_data\_y=os.fit\_resample(X\_train, y\_train)

os\_data\_X = pd.DataFrame(data=os\_data\_X,columns=columns )

os\_data\_y= pd.DataFrame(data=os\_data\_y,columns=['ReAdmis\_Yes'])

# Check the numbers of our data to see if new dataset is now balanced after traning and sampling

print("length of oversampled data is ",len(os\_data\_X))

print("Number of no ReAdmis in oversampleddata",len(os\_data\_y[os\_data\_y['ReAdmis\_Yes']==0]))

print("Number of ReAdmis",len(os\_data\_y[os\_data\_y['ReAdmis\_Yes']==1]))

print("Proportion of no ReAdmis data in oversampled data is",len(os\_data\_y[os\_data\_y['ReAdmis\_Yes']==0])/len(os\_data\_X))

print("Proportion of ReAdmis data in oversampled data is",len(os\_data\_y[os\_data\_y['ReAdmis\_Yes']==1])/len(os\_data\_X))

length of oversampled data is 8852

Number of no ReAdmis in oversampleddata 4426

Number of ReAdmis 4426

Proportion of no ReAdmis data in oversampled data is 0.5

Proportion of ReAdmis data in oversampled data is 0.5

Next we will run the RFE to prep the data for the Regression and rank the variables that may have the most impact. The reults are below.

The following features are selected:

Column: 7, Rank: 1.000, Feature: Initial\_days

Column: 8, Rank: 1.000, Feature: Area\_Suburban

Column: 12, Rank: 1.000, Feature: Gender\_Nonbinary

Column: 13, Rank: 1.000, Feature: Soft\_drink\_Yes

Column: 14, Rank: 1.000, Feature: Initial\_admin\_Emergency Admission

Column: 16, Rank: 1.000, Feature: HighBlood\_Yes

Column: 17, Rank: 1.000, Feature: Stroke\_Yes

Column: 18, Rank: 1.000, Feature: Complication\_risk\_Low

Column: 19, Rank: 1.000, Feature: Complication\_risk\_Medium

Column: 21, Rank: 1.000, Feature: Arthritis\_Yes

Column: 25, Rank: 1.000, Feature: Anxiety\_Yes

Column: 26, Rank: 1.000, Feature: Allergic\_rhinitis\_Yes

Column: 27, Rank: 1.000, Feature: Reflux\_esophagitis\_Yes

Column: 28, Rank: 1.000, Feature: Asthma\_Yes

Now we can begin to build our Initial Model.

# initial model

X=os\_data\_X[features] # from RFE above

Xc = sm.add\_constant(X) # reset

y=os\_data\_y[target]

logit\_model=sm.MNLogit(y,Xc)

result=logit\_model.fit(maxiter=1000,method='bfgs')

print(result.summary2())

A screenshot of a computer

Description automatically generated

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.

Now that the Initial Model is run and a ranking of potential impactors to the target, we can begin to look at possible ways to reduce the potential variables by looking at the P-Values, Coefficients, Standard Error values and Z-Values.

Now we can use a confusion matrix to compare potential pairs of variables.

# confustion matrix for initial model

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,test\_size=0.3, random\_state=0)

lgr = LogisticRegression(max\_iter=10000)

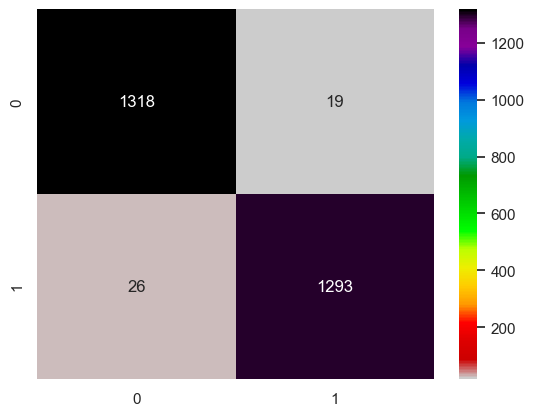
lgr.fit(X\_train, y\_train)

predicted = lgr.predict(X\_test)

expected = y\_test

confusion = pd.DataFrame(confusion\_matrix(y\_true=expected, y\_pred=predicted),index=range(2),columns=range(2))

axes = sns.heatmap(confusion, annot=True,cmap='nipy\_spectral\_r', fmt='g')



We then can provide a number and percentage for the predictions shown, helping the confidence level of the initial model.

# calculate number and percent of predictions

correct = sum(np.diagonal(confusion)) # on diag

total = confusion.values.sum()

incorrect = total - correct # off diag

print('Correct predictions on diagonal: {} ({:.0%})'.format( correct, correct / total ))

print('Incorrect predictions off diagonal: {} ({:.0%})'.format( incorrect, incorrect / total ))

Correct predictions on diagonal: 2611 (98%)

Incorrect predictions off diagonal: 45 (2%)

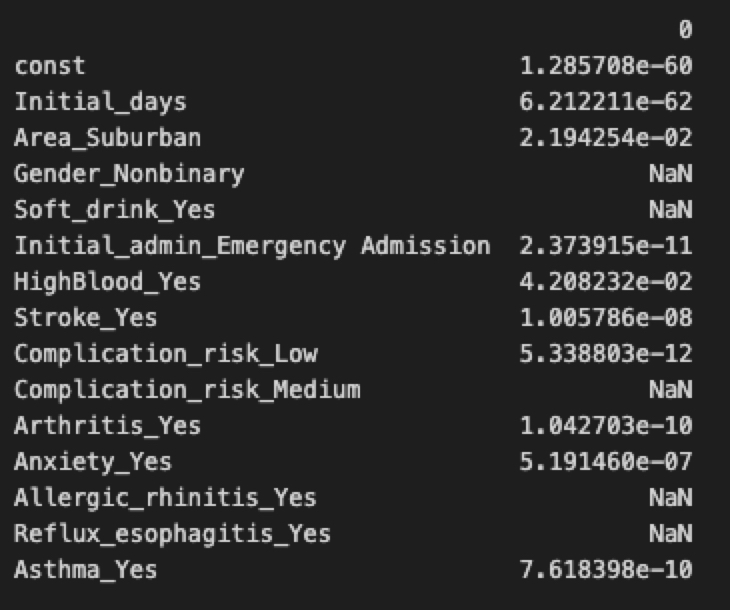
The next step, would be to compare result values to find the P-Values below .05

#Looking for P values <0.05

data=result.pvalues

d=data[data[:]<.05]

print(d)



Our next step and the one that will help us reduce the variable set down for the next logistic regression, running a VIF(Variance Inflation Factor) When we get the results, anything over the threshold of 5 will be removed. 5 is considered possibly too conservative but as you will see in this data set, 5 seems to be the appropriate number.

#Running a VIF test to check for multicollinearity

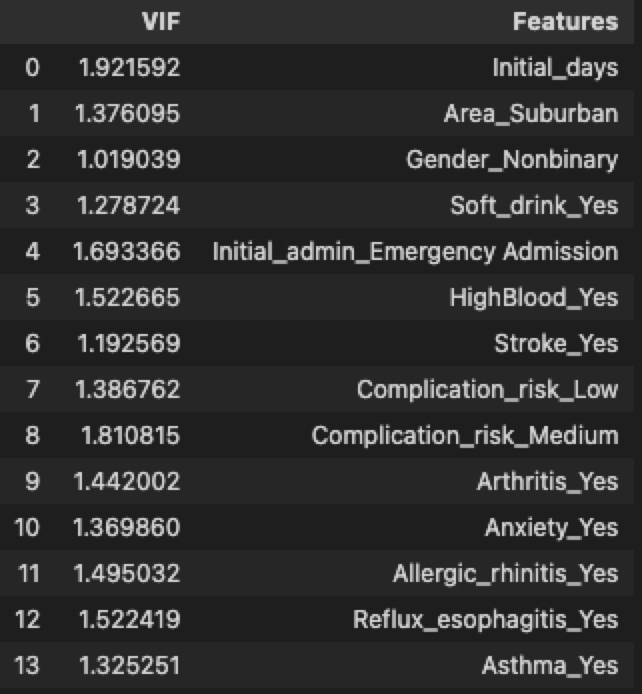
from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

vif=pd.DataFrame()

vif['VIF']=[variance\_inflation\_factor(X\_Train.values, i) for i in range(X\_Train.shape[1])]

vif['Features']=X\_Train.columns

vif[vif['VIF']<=5]



3.  Provide a reduced logistic regression model that follows the feature selection or model evaluation process in part D2, including a screenshot of the output for each model. Using the data from the VIF we can now condense the data frame down to only what we are going to be using for the reduced logistic regression.

We are still using the same training and test data for this model only altered to only include items with a lower VIF value than 5.

# Lets Reduce This Model

features.remove('Gender\_Nonbinary')

features.remove('Soft\_drink\_Yes')

features.remove('Complication\_risk\_Medium')

features.remove('Allergic\_rhinitis\_Yes')

features.remove('Reflux\_esophagitis\_Yes')

X=os\_data\_X[features]

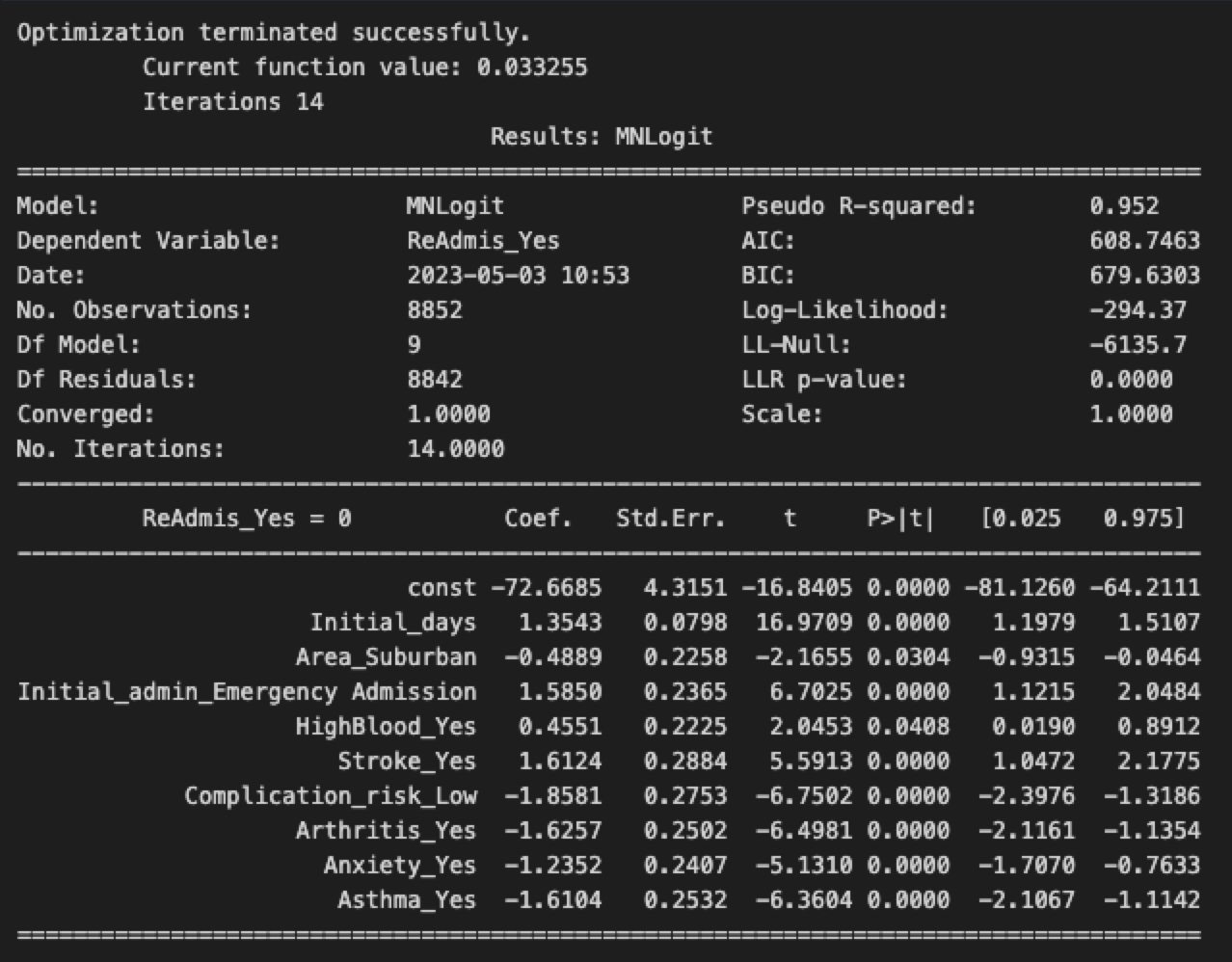
y=os\_data\_y[target]

Xc = sm.add\_constant(X) # reset

logit\_model=sm.MNLogit(y,Xc)

result=logit\_model.fit()

print(result.summary2())



Again, another confusion matrix to help see if our numbers have changed.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=.3, random\_state=0)

lgr = LogisticRegression(max\_iter=10000)

lgr.fit(X\_train, y\_train)

predicted = lgr.predict(X\_test)

expected=y\_test

confusion=pd.DataFrame(confusion\_matrix(y\_true=expected, y\_pred=predicted), index=range(2), columns=range(2))

axes = sns.heatmap(confusion, annot=True, cmap='nipy\_spectral\_r', fmt='g')

A picture containing text, screenshot, design

Description automatically generated

Now with the reduced set, we can check again with the predictions.

# calculate number and percent of predictions

correct = sum(np.diagonal(confusion)) # on diag

total = confusion.values.sum()

incorrect = total - correct # off diag

print('Correct predictions on diagonal: {} ({:.0%})'.format( correct, correct / total ))

print('Incorrect predictions off diagonal: {} ({:.0%})'.format( incorrect, incorrect / total ))

Correct predictions on diagonal: 2611 (98%)

Incorrect predictions off diagonal: 45 (2%)

E.  Analyze the data set using your reduced logistic regression model by doing the following:

1.  Explain your data analysis process by comparing the initial logistic regression model and reduced logistic regression model, including the following element:

•   a model evaluation metric

-Initial Model equation

-73.5828 = 1.3799(Initial\_days) - .5263(Area\_Suburban) - .5926(Gender Nonbinary) + .4780(Soft\_drink\_Yes) + 1.5945(Initial\_admin\_Emergengy\_admission) + .4572(HighBlood\_Yes) + 1.6797(Stroke) – 2.1864(Complication\_risk\_low) - .4854(Complication\_Risk\_Medium) – 1.6360(Arthrits\_Yes) – 1.2312(Anxiety\_Yes) - .3953(Allergic\_Rhinitis\_Yes) - .3852(Reflux\_esophagitis\_Yes) – 1.5865(Astham\_Yes)

Reduced Model equation

72.6685 = 1.3543(Initial\_days) - .4889(Area Suburban) + 1.5850(Initial\_admin\_emergency admission) + .4551(HighBlood\_yes) + 1.6124(Stroke\_Yes) – 1.8581(Complication\_risk\_low) – 1.6257(Arthritis\_yes) – 1.2352(Anxiety\_Yes) – 1.6104(Asthma\_Yes)

Metric evaluation

Receiver Operating Characteristic and Classification Report

# classification report

print(classification\_report(expected, predicted))# plot ROC Curve

logit\_roc\_auc = roc\_auc\_score(y\_test, lgr.predict(X\_test))

fpr, tpr, thresholds = roc\_curve(y\_test, lgr.predict\_proba(X\_test)

[:,1])

plt.figure()

plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' %logit\_roc\_auc)

plt.plot([0, 1], [0, 1],'r--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver operating characteristic')

plt.legend(loc="lower right")

plt.savefig('Log\_ROC')

plt.show()

A picture containing text, screenshot, font, black

Description automatically generated

A graph with a red line

Description automatically generated with low confidence

According to Li (2017), “The receiver operating characteristic(ROC) curve is another common tool used with binary classifiers. The dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top left corner). By the looks of this metric we have modeled this data fairly well.

Li, S. (2017, Sep 28). Building A Logistic Regression in Python, Step by Step. Retrieved from: https://towardsdatascience.com/building-a-logisticregression-in-python-step-by-step-becd4d56c9c8

2.  Provide the output and all calculations of the analysis you performed, including the following elements for your reduced logistic regression model:

•   confusion matrix:

#checking the correlation matrix

corr=df.corr()

plt.figure(figsize=(20,20))

sns.heatmap(corr, annot=True, cmap='coolwarm')

A screenshot of a graph

Description automatically generated with low confidence

•   accuracy calculation:

#accuracy of this model

print('Accuracy : ', accuracy\_score(y\_test, predicted))

Accuracy : 0.9830572289156626

3.  Provide an executable error-free copy of the code used to support the implementation of the logistic regression models using a Python or R file.

Filename is: Med208T2.ipynb

**Part V: Data Summary and Implications**

F.  Summarize your findings and assumptions by doing the following:

1.  Discuss the results of your data analysis, including the following elements:

•   a regression equation for the reduced model

So the logistic equation is as follows

-72.6685 = 1.3543(Initial\_days) - .4889(Area Suburban) + 1.5850(Initial\_admin\_emergency admission) + .4551(HighBlood\_yes) + 1.6124(Stroke\_Yes) – 1.8581(Complication\_risk\_low) – 1.6257(Arthritis\_yes) – 1.2352(Anxiety\_Yes) – 1.6104(Asthma\_Yes)

•   an interpretation of the coefficients of the reduced model:

The logistic regression model is using ReAdmis\_Yes (whether a patient was readmitted or not) as the dependent variable and the other variables as independent variables. The coefficients of the model represent the change in the log odds of being readmitted for each unit increase in the corresponding independent variable.

1. The constant coefficient represents the log odds of being readmitted when all other variables are equal to zero. It has a negative coefficient, which suggests that the baseline likelihood of being readmitted is low.
2. The coefficient for Initial\_days suggests that for each additional day that a patient is in the hospital at their initial admission, the log odds of being readmitted increase by 1.3543.
3. The coefficients for Area\_Suburban, HighBlood\_Yes, Arthritis\_Yes, Anxiety\_Yes, and Asthma\_Yes are negative, indicating that patients with these conditions are less likely to be readmitted than those without.
4. The coefficients for Initial\_admin\_Emergency Admission, Stroke\_Yes, and Complication\_risk\_Low are positive, suggesting that patients with these conditions are more likely to be readmitted.

Overall, the model indicates that the patient's initial hospital stay, their medical conditions, and the nature of their initial admission are important factors in predicting whether they will be readmitted.

•   the statistical and practical significance of the reduced model

The logistic regression model has a high pseudo R-squared value of 0.952, indicating that the model explains a large portion of the variability in the dependent variable. This value represents the proportion of variance explained by the model, compared to a model with only the intercept. However, it's important to note that pseudo R-squared values should be interpreted with caution, as they can be inflated by including too many variables in the model.

The LLR p-value of 0.0000 indicates that the overall model is statistically significant, meaning that at least one of the independent variables is significantly related to the dependent variable. The LL-Null value of -6135.7 represents the log-likelihood of a model with no predictors. The difference between the LL-Null and Log-Likelihood values (-294.37) indicates the improvement in fit obtained by including the predictors in the model.

Each of the independent variables in the model also has a coefficient, standard error, t-value, and p-value associated with it. The t-value and p-value for each variable are used to determine if the variable is statistically significant, that is, if its effect on the dependent variable is not likely due to chance. In this case, all variables except for Area\_Suburban have p-values less than 0.05, indicating that they are statistically significant.

Therefore, the logistic regression model as a whole is statistically and practically significant and provides insights into which variables are important predictors of the dependent variable, ReAdmis\_Yes.

•   the limitations of the data analysis

In general, here are some common limitations that could be relevant to any analysis:

1. Data quality: The analysis can be limited by the quality of the data used. Data may be incomplete, inaccurate, or not representative of the population of interest.
2. Assumptions: Certain statistical techniques and models rely on specific assumptions, which may not hold true in the data being analyzed. For example, linear regression assumes that the relationship between the independent and dependent variables is linear, which may not always be the case.
3. Causation vs correlation: Correlation does not necessarily imply causation. Even if two variables are strongly correlated, it may not mean that one causes the other.
4. Generalization: The analysis may not be generalizable to other populations or contexts. The findings may only be applicable to the specific sample or time period being studied.
5. Confounding variables: There may be other variables that influence the relationship being studied, which are not included in the analysis. These variables can create bias and limit the accuracy of the findings.

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

With some of the factors outside of the control of the patient and hospital, some readmissions may be unavoidable. However, the hospital could mitigate some of the factors we observed in this regression and lower that initial stay which does correlate to lower risk of readmission. The healthcare professionals then can focus on a more timely and accurate diagnosis the first time, could also impact how long the patient stays initially then if they may be readmitted. More analysis regressions should be performed with different questions and focuses to help find and reduce the predictors of a readmission.

Citations:

1. Books Ciaburro, G. (2018). Regression analysis with R : Design and develop statistical nodes to identify unique relationships within data at scale. Packt Publishing.
2. Massaron, L., & Boschetti, A. (2016). Regression analysis with python: Learn the art of regression analysis with python. Packt Publishing
3. <https://www.javatpoint.com/vif-in-python#:~:text=The%20Variance%20Inflation%20Factor%20is,with%20the%20other%20independent%20variables>.
4. <https://www.statology.org/statsmodels-linear-regression-p-value/>
5. <https://www.w3schools.com/python/python_ml_multiple_regression.asp>
6. <https://towardsdatascience.com/how-to-remove-multicollinearity-using-python-4da8d9d8abb2>
7. <https://www.youtube.com/watch?v=He74ncsUMbE>
8. <https://www.kaggle.com/code/ashishsaxena2209/step-by-step-regression-backward-elimination/notebook>
9. ht<tps://scikit-le>arn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html