**PERFORMANCE ASSESSMENT TASK 2:**

**D208 – Predictive Modeling**

Michael Lawson

Western Governors University

D208 – Predictive Modeling

Keiona Middleton

02/28/2022

Michael C. Lawson | Student ID: #001479557 | Program: MS Data Analytics (5/1/21) |

Program Mentor: Emil Stoica | 859-265-1888 (EST) | [mlaws77@wgu.edu](mailto:mlaws77@wgu.edu)

**Part I: Research Question**

**A.  Purpose of this data analysis:**

**1. Real world research question.**

Do medical conditions and hospital care influence whether a patient will have a high risk of being readmitted?

**2.  What are the objectives or goals of the analysis? Are they reasonable within the scope of the data dictionary and represented in the available data?**

The objective of the data analysis will be to determine whether a patient’s medical conditions will predict that they are more likely to be readmitted in the future. The hospital could use this information to make decisions on how to reduce readmissions by addressing the cause. The goal of this logistic regression will be to calculate the percentage of how likely a patient is to be readmitted according to the most influential variables. The data set does provide information on which patients were readmitted, and there is a large set of explanatory variables related to patient demographics and

**Part II: Method Justification**

**B.  Describe logistic regression methods:**

**1.  The assumptions of a logistic regression model.**

Logistic regression is based on Bernoulli (binomial or boolean) distribution rather than Gaussian because the dependent variable is binary. Predicted values are limited to a range of nominal binary variables such as “No, Yes”, “False, True”, “Fail, Pass”, etc… Logistic regression is the classifiction algorithm for machine learning. Logisitc regression classifies percentages of which binomial or boolean value the independent variable falls. Logisitc regression assumes there is no high correlation or multicollinearity. The maximum likelihood estimation method is used to estimate accuracy. Logistic regression is the logarithm of the odds of achieving 1 (Sewell, ep. 4, sl. 10)

**2.  The benefits of choosing Python.**

Python is the tool chosen for this task., Python can import all the tools needed to perform the same tasks built into R with a few lines of code. Python is more flexible for a multitude of uses and has thousands of packages that do so much more than statistics. Python can import and read a wider variety of file types than R. Python is faster at handling large multidimensional arrays using the numpy library and can create beautiful visualizations using packages like Matplotlib and Seaborn. Python is more powerful for data wrangling and web scraping in the future if obtaining more data would be helpful. Python is popular with data scientists because of its machine learning and deep learning capabilities. Logistic regression is easily calculated and visualized using various packet libraries.

**3.  Why is logistic regression an appropriate technique to analyze the question in Part I?**

The response variable is “ReAdmis”, which has binary nomial categories “Yes” and “No”. Logistic regression is the appropriate method for regression on a binary response variable. Rather than predicting an outcome like linear regression, logistic regression predicts the percentage of chance the explanatory variable will fall into one of two categories.

**Part III: Data Preparation**

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

1.  Describe your data preparation goals and the data manipulations that will be used to achieve the goals.

The data preparation goals will be to clean the data, visualize the data, try to find a relationship with the binary response variable “Admis”, verify a relationship statistically, reduce the variables, run a logistic regression model on all the variables selected, reduce the variables using the regression statistics, and then run a regression model on the final set of variables. The data will be cleaned by treating missing values, treating outliers, and checking the data for mistakes such as spelling errors that would create outliers. A gross group of variables will be selected from the data that will exclude variables that obviously wouldn’t contribute to answering the question.

A sample will be taken from the large population. The data will be sorted by the values “Yes” an. “No” in the “ReAdmis” column. The data will be visualized using scatterplots, boxplots, histograms, etc…The goal of the visualizations is to see if any relationship can be seen with the response variable before the model is run. Before making any decisions on reducing the data with graphs, the variation inflation factor (VIF) will be calculated on various groups of variables in the data to check for multicollinearity. VIF will help reduce the data further and prepare for principal component analysis (PCA). PCA will help see which variables have the most statistical effect on the data. It will help determine which variables to keep after running recursive feature elimination (RFE). The correlation of the remaining variables will be checked to understand the PCA and RFE results even further. A regression model will be run on all the variables that remained after RFE and PCA. The p-values of the explanatory variables and their relationships will be used to make the final selection of variables. A final linear regression model will be created along with visualizations of the model. The end goal is to find a significant explanation to predict the response variable.

2.  Discuss the summary statistics, including the target variable and all predictor variables that you will need to gather from the data set to answer the research question.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **dtype** | **Variable Type** | **Example** |
| ReAdmis | object | Binary categorical (target) | Yes, No |
| VitD\_levels | int64 | continuous | 19.141466 |
| Soft\_drink | object | Binary | 0,1 |
| HighBlood | object | binary categorical | Yes, No |
| Stroke | object | binary categorical | Yes, No |
| Complication\_risk | object | ordinal categorical | Low,Medium, High |
| Overweight | object | binary categorical | Yes, No |
| Arthritis | object | binary categorical | Yes, No |
| Diabetes | object | binary categorical | Yes, No |
| Hyperlipidemia | object | binary categorical | Yes, No |
| BackPain | object | binary categorical | Yes, No |
| Anxiety | object | binary categorical | Yes, No |
| Allergic\_rhinitis | object | binary categorical | Yes, No |
| Reflux\_esophagitis | object | binary categorical | Yes, No |
| Asthma | object | binary categorical | Yes, No |

Graphical user interface

Description automatically generated with low confidence

Table

Description automatically generated

Most of the variables are binary, so with an even number of 1 and 0, the mean should be 0.5. You can see that “Soft\_drink” has a mean of 0.26, which tells us there are probably more cases of 0 than 1. The minimum value in binary columns is 0 and the max is 1. We can say patients stay on average 34 days in “Initial\_days”, but the standard deviation of 26.3 tells us the mean has extreme cases pulling the average number in one direction. We can see that the minimum stay was 1 day and the maximum was 72 days.

3.  Explain the steps used to prepare the data for the analysis, including the annotated code.

1. Load Python libraries appropriate for data visualization and regression.
2. Load the data using read.csv()
3. Examine the header to see what column names and the values they contain using .head().
4. Examine the shape, dtype, and all column names using .info().
5. Change the ordinal categorical variable “Complication\_risk” to ranking numerical values.
6. Convert all binary categorical columns from “No,Yes” to “0,1”.
7. If null values exist, treat them.
8. Treat outliers as they are found in visualizations.
9. Create histplots of all variables to compare distribution.
10. Rather than delete any data, select data needed to create a new data frame of the variables that are most likely to influence the response variable “ReAdmis”.
11. Review the mean, median and quartiles of the data gross data set selected.
12. Create histograms of the “ReAdmis” of every explanatory variable.
13. Perform VIF on groups of variables to test for and drop columns with multicollinearity to help prevent overfitting in the model.
14. Perform PCA to examine which variables have the most effect on the data set.
15. Create a covariance matrix and keep the PCA results above 1.
16. Perform recursive feature extraction (RFE).
17. Compare PCA results and RFE to decide which variables actually have the most effect on the response variable.
18. Check the Pearson correlation of the selected variables and their relationships with each other and the response variable.
19. Use scatterplots with lineplots to visualize different combinations of the remaining variables.
20. Run the Logistic Regression model
21. Decide on the final variables for the finished model.
22. Create visualizations of the final model, using different combinations of the variables if needed.

**Annotated Code**

*# import all possible packages useful for multiple linear regression*

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

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

**import** sklearn

**from** pandas **import** Series, DataFrame

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

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

**from** pylab **import** rcParams

**from** sklearn **import** preprocessing

**from** sklearn.linear\_model **import** LogisticRegression

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

**from** sklearn **import** metrics

**from** sklearn.metrics **import** classification\_report, confusion\_matrix

**import** statsmodels.api **as** sm

**from** statsmodels.formula.api **import** logit

**from** sklearn.feature\_selection **import** RFE

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

**import** warnings

warnings**.**filterwarnings("ignore")

*#Import data set from hard drive*

med **=** pd**.**read\_csv(r"C:\Users\mlaws\OneDrive - Western Governors University\Documents\WGU\D208\medical\_clean.csv", skiprows**=**0, delimiter**=**",")

*#examine the list of variables, their data type and the shape of the data*

med**.**info()

*#print the header*

med**.**head()

*#change the values of the ordinal categorical variable to numeric ranks*

med['Complication\_risk']**.**unique()

med['Complication\_risk'] **=** med['Complication\_risk']**.**map({'Low':0,'Medium':1,'High':2})

*#change all binary categorical values to 0,1*

med['ReAdmis'] **=** med['ReAdmis']**.**map({'No':0, "Yes":1})

med['Soft\_drink'] **=** med['Soft\_drink']**.**map({'No':0, "Yes":1})

med['HighBlood'] **=** med['HighBlood']**.**map({'No':0, "Yes":1})

med['Stroke'] **=** med['Stroke']**.**map({'No':0, "Yes":1})

med['Overweight'] **=** med['Overweight']**.**map({'No':0, "Yes":1})

med['Arthritis'] **=** med['Arthritis']**.**map({'No':0, "Yes":1})

med['Diabetes'] **=** med['Diabetes']**.**map({'No':0, "Yes":1})

med['Hyperlipidemia'] **=** med['Hyperlipidemia']**.**map({'No':0, "Yes":1})

med['BackPain'] **=** med['BackPain']**.**map({'No':0, "Yes":1})

med['Anxiety'] **=** med['Anxiety']**.**map({'No':0, "Yes":1})

med['Allergic\_rhinitis'] **=** med['Allergic\_rhinitis']**.**map({'No':0, "Yes":1})

med['Reflux\_esophagitis'] **=** med['Reflux\_esophagitis']**.**map({'No':0, "Yes":1})

med['Asthma'] **=** med['Asthma']**.**map({'No':0, "Yes":1})

*#Create dummy columns for the Services column*

med **=** pd**.**get\_dummies(med, prefix**=**'Services', prefix\_sep**=**'\_', dummy\_na**=False**, columns**=**["Services"])

*#Check whether dummy columns were created*

med**.**columns

Index(['CaseOrder', 'Customer\_id', 'Interaction', 'UID', 'City', 'State',

'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job',

'Children', 'Age', 'Income', 'Marital', 'Gender', 'ReAdmis',

'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'vitD\_supp',

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

'Complication\_risk', 'Overweight', 'Arthritis', 'Diabetes',

'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic\_rhinitis',

'Reflux\_esophagitis', 'Asthma', 'Initial\_days', 'TotalCharge',

'Additional\_charges', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5',

'Item6', 'Item7', 'Item8', 'Services\_Blood Work', 'Services\_CT Scan',

'Services\_Intravenous', 'Services\_MRI'],

dtype='object')

*#Gross selection of data before PCA*

med\_df **=** med[['Doc\_visits',"Initial\_days", "ReAdmis","VitD\_levels", "Soft\_drink","HighBlood", "Stroke", "Complication\_risk", "Overweight", "Arthritis", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Allergic\_rhinitis", "Reflux\_esophagitis", "Asthma",'Services\_Blood Work', 'Services\_CT Scan',

'Services\_Intravenous', 'Services\_MRI']]

*#Standardize the data so that all the variables are uniform in comparison*

x **=** med\_df[["Initial\_days","VitD\_levels", "Soft\_drink", "HighBlood", "Stroke", "Complication\_risk", "Overweight", "Arthritis", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Allergic\_rhinitis", "Reflux\_esophagitis", "Asthma",'Services\_Blood Work', 'Services\_CT Scan',

'Services\_Intravenous', 'Services\_MRI']]

y **=** med\_df["ReAdmis"]

x\_data**=**x

target**=**y

scaled\_data**=**preprocessing**.**scale(x\_data)

*#Create histograms of every column to visualize*

sns**.**set(rc **=** {'figure.figsize':(4,3)})

sns**.**histplot(data**=**med\_df, x**=**"ReAdmis", bins**=**10)

sns**.**histplot(data**=**med\_df, x**=**"Initial\_days", bins**=**10)

<AxesSubplot:xlabel='Initial\_days', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"VitD\_levels", bins**=**10)

<AxesSubplot:xlabel='VitD\_levels', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"Soft\_drink", bins**=**10)

<AxesSubplot:xlabel='Soft\_drink', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"HighBlood", bins**=**10)

<AxesSubplot:xlabel='HighBlood', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"Stroke", bins**=**10)

<AxesSubplot:xlabel='Stroke', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"Complication\_risk", bins**=**10)

<AxesSubplot:xlabel='Complication\_risk', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"Overweight", bins**=**10)

<AxesSubplot:xlabel='Overweight', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"Arthritis", bins**=**10)

<AxesSubplot:xlabel='Arthritis', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"Diabetes", bins**=**10)

<AxesSubplot:xlabel='Diabetes', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"Hyperlipidemia", bins**=**10)

<AxesSubplot:xlabel='Hyperlipidemia', ylabel='Count'>

sns**.**histplot(data**=**med\_df, x**=**"BackPain", bins**=**10)

sns**.**histplot(data**=**med\_df, x**=**"Anxiety", bins**=**10)

sns**.**histplot(data**=**med\_df, x**=**"Allergic\_rhinitis", bins**=**10)

sns**.**histplot(data**=**med\_df, x**=**"Reflux\_esophagitis", bins**=**10)

sns**.**histplot(data**=**med\_df, x**=**"Asthma", bins**=**10)

<AxesSubplot:xlabel='Asthma', ylabel='Count'>

*#Create bivariate visualizations of data against "ReAdmis"*

sns**.**displot(data**=**med\_df, x**=**"ReAdmis", col**=**"Complication\_risk", col\_wrap**=**3, bins**=**10)

plt**.**show()

readmits **=** med\_df[med\_df['ReAdmis']**==**1]

admits **=** med\_df[med\_df['ReAdmis']**==**0]

sns**.**histplot(data**=**admits, x**=**"HighBlood", color**=**'violet')

sns**.**histplot(data**=**readmits, x**=**"HighBlood", color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Initial\_days", bins**=**10,color**=**'violet')

sns**.**histplot(data**=**readmits, x**=**"Initial\_days",bins**=**10, color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Soft\_drink", color**=**'green')

sns**.**histplot(data**=**readmits, x**=**"Soft\_drink", color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Stroke", color**=**'blue')

sns**.**histplot(data**=**readmits, x**=**"Stroke", color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Complication\_risk", color**=**'orange')

sns**.**histplot(data**=**readmits, x**=**"Complication\_risk", color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Overweight", color**=**'pink')

sns**.**histplot(data**=**readmits, x**=**"Overweight", color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Arthritis", color**=**'aqua')

sns**.**histplot(data**=**readmits, x**=**"Arthritis", color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Anxiety", color**=**'lightgreen')

sns**.**histplot(data**=**readmits, x**=**"Anxiety", color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Diabetes", color**=**'lightblue')

sns**.**histplot(data**=**readmits, x**=**"Diabetes", color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Hyperlipidemia", color**=**'purple')

sns**.**histplot(data**=**readmits, x**=**"Hyperlipidemia", color**=**'grey')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"BackPain", color**=**'grey')

sns**.**histplot(data**=**readmits, x**=**"BackPain", color**=**'white')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Allergic\_rhinitis", color**=**'lightblue')

sns**.**histplot(data**=**readmits, x**=**"Allergic\_rhinitis", color**=**'white')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Reflux\_esophagitis", color**=**'lightgreen')

sns**.**histplot(data**=**readmits, x**=**"Reflux\_esophagitis", color**=**'white')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Services\_Blood Work", color**=**'purple')

sns**.**histplot(data**=**readmits, x**=**"Services\_Blood Work", color**=**'white')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Services\_CT Scan", color**=**'lightgreen')

sns**.**histplot(data**=**readmits, x**=**"Services\_CT Scan", color**=**'white')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Services\_MRI", color**=**'lightblue')

sns**.**histplot(data**=**readmits, x**=**"Services\_MRI", color**=**'white')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

sns**.**histplot(data**=**admits, x**=**"Services\_Intravenous", color**=**'orange')

sns**.**histplot(data**=**readmits, x**=**"Services\_Intravenous", color**=**'white')

plt**.**legend(labels**=**["Initial admit", "Readmit"], bbox\_to\_anchor**=**(1,1))

*#Separate data into 30% test data and 70% training data*

X**=** med\_df[['Initial\_days','VitD\_levels','Soft\_drink','HighBlood','Stroke','Complication\_risk','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety', 'Allergic\_rhinitis','Reflux\_esophagitis','Asthma','Services\_Blood Work', 'Services\_CT Scan',

'Services\_Intravenous', 'Services\_MRI']]

y**=** med\_df[['ReAdmis']]

X\_train,X\_test,y\_train,y\_test **=** train\_test\_split(X,y, train\_size**=**0.7, test\_size**=**0.3, random\_state**=**944)

*#Verify train and test sets have the same percentage of ReAdmis*

train **=** pd**.**concat([X\_train, y\_train], axis**=**1)

test **=** pd**.**concat([X\_test, y\_test], axis**=**1)

print(round(sum(train['ReAdmis'])**/**len(train), 2))

print(round(sum(test['ReAdmis'])**/**len(test), 2))

0.37

0.35

*# VIF for demographic data. Adding gender columns returned an error when mixed with marital*

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

X **=** med\_df[["Initial\_days","VitD\_levels","Soft\_drink", "HighBlood","Stroke","Complication\_risk","Overweight","Arthritis","Diabetes", "Hyperlipidemia","BackPain", "Anxiety","Allergic\_rhinitis","Reflux\_esophagitis","Asthma",'Services\_Blood Work', 'Services\_CT Scan',

'Services\_Intravenous', 'Services\_MRI']]

vif\_demo\_data **=** pd**.**DataFrame()

vif\_demo\_data['Med\_cond'] **=** X**.**columns

vif\_demo\_data["VIF"] **=** [variance\_inflation\_factor(X**.**values, i)

**for** i **in** range(len(X**.**columns))]

print(vif\_demo\_data)

*#Perform PCA*

**from** sklearn.decomposition **import** PCA

med\_pca **=** X

med\_normalized **=** (med\_pca**-**med\_pca**.**mean())**/**med\_pca**.**std()

pca **=** PCA(n\_components **=** med\_pca**.**shape[1])

pca**.**fit(med\_normalized)

loadings **=** pd**.**DataFrame(pca**.**components\_**.**T,

columns **=** ['PCA1', 'PCA2', 'PCA3', 'PCA4','PCA5','PCA6', 'PCA7', 'PCA8','PCA9','PCA10','PCA11','PCA12','PCA13', "PCA14", "PCA15", "PCA16","PCA17","PCA18", "PCA19"],

index**=**med\_normalized**.**columns)

loadings

*#Create a covariance matrix to decide which PCAs to keep*

sns**.**set(rc **=** {'figure.figsize':(8,6)})

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

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

plt**.**plot(eigenvalues, 'ro-')

plt**.**xlabel('Number of Components')

plt**.**ylabel('Eigenvalues')

plt**.**show()

*#heatmap to view correlation*

sns**.**set(rc **=** {'figure.figsize':(15,8)})

sns**.**heatmap(med\_df**.**corr(), annot**=True**)

plt**.**show()

*#Check Pearson correlation*

med\_df**.**corr(method**=**'pearson')

*#perform RFE to pick top 3 explanatory variables*

logreg **=** LogisticRegression()

rfe **=** RFE(logreg,3)

rfe **=** rfe**.**fit(X\_train,y\_train)

list(zip(X\_train**.**columns, rfe**.**support\_, rfe**.**ranking\_))

*#1st Logostic Regression Model*

LogReg1 **=** sm**.**GLM(y\_train,(sm**.**add\_constant(X\_train)), family **=** sm**.**families**.**Binomial())

LogReg1**.**fit()**.**summary()

*#create reduced model*

col **=** X\_train**.**columns[rfe**.**support\_]

X\_train\_sm **=** sm**.**add\_constant(X\_train[col])

LogReg2 **=** sm**.**GLM(y\_train,X\_train\_sm, family **=** sm**.**families**.**Binomial())

res **=** LogReg2**.**fit()

res**.**summary()

*#Visualize logistic regression with stroke and initial days*

sns**.**set(rc **=** {'figure.figsize':(10,8)})

plt**.**xlim(**-**10,75)

plt**.**ylim(**-**0.25,1.25)

sns**.**regplot(x**=**'Initial\_days', y**=**"ReAdmis", data**=**med, color**=**"lightblue", logistic**=True**)

sns**.**scatterplot(x**=**'Initial\_days', y**=**"ReAdmis", hue**=**"Stroke", color**=**"black", data**=**med, ci**=None**)

*#Visualize logistic regression with MRI and initial days*

sns**.**set(rc **=** {'figure.figsize':(10,8)})

plt**.**xlim(**-**10,80)

plt**.**ylim(**-**0.2,1.2)

sns**.**regplot(x**=**'Initial\_days', y**=**"ReAdmis", data**=**med, color**=**"lightblue", logistic**=True**)

sns**.**scatterplot(x**=**'Initial\_days', y**=**"ReAdmis", hue**=**"Services\_MRI", color**=**"black", data**=**med, ci**=None**)

*#Define Classifier*

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

knn **=** KNeighborsClassifier(n\_neighbors **=** 5, metric **=** 'minkowski', p **=** 2)

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

*# Predicting the Test set results*

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

*# Making the Confusion Matrix*

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

print(conf\_matrix)

[[1892 46]

[ 36 1026]]

*#Visualize the confusion matrix*

**from** statsmodels.graphics.mosaicplot **import** mosaic

mosaic(conf\_matrix)

plt**.**show()

*#Export prepared data*

med\_df**.**to\_csv(r"C:\Users\mlaws\OneDrive - Western Governors University\Documents\WGU\D208\medical\_prepared\_D208P\_A2.csv")

4.  Generate univariate and bivariate visualizations of the distributions of variables in the cleaned data set. Include the target variable in your bivariate visualizations.

**Univariate**

A picture containing shape

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generated

A picture containing shape

Description automatically generatedA picture containing shape

Description automatically generatedChart, bar chart

Description automatically generated

Chart

Description automatically generated with low confidenceA picture containing shape

Description automatically generatedA picture containing shape

Description automatically generated

A picture containing shape

Description automatically generatedA picture containing shape

Description automatically generatedA picture containing shape

Description automatically generated

A picture containing shape

Description automatically generatedA picture containing shape

Description automatically generatedA picture containing shape

Description automatically generated

**Bivariate**

**A picture containing chart

Description automatically generated**

**Chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generated with low confidence**

Chart, waterfall chart

Description automatically generatedChart, waterfall chart

Description automatically generatedChart

Description automatically generated

Chart, waterfall chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generated

A picture containing chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generated

Chart

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generatedChart, histogram

Description automatically generated

5.  Provide a copy of the prepared data set.

med\_picks.to\_csv(r"C:\Users\mlaws\OneDrive - Western Governors University\Documents\WGU\D208\medical\_prepared\_D208P\_A2.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 predictors that were identified in Part C2

Graphical user interface

Description automatically generated with low confidence

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

The p-value should be lower that 0.05 on variables that are strong influencers, otherwise the influence of the variable could be due to random chance. The model had variables with a p-value lower than 0.05. The Variance Inflation Factor (VIF) revealed that the four dummy columns created to separate “Services” had extremely high multicollinearity. “Services\_MRI” was the only column in the group with a VIF lower than 10. “Services\_MRI” also made it through as one of the top 3 most influential columns in the recursive feature elimination (RFE) step, so the other 3 “Services” dummy columns were removed. PCA, RFE, and correlation were all examined to make sure the best variables were chosen for the final model. The outputs are shown below in E2. PCA showed the the “Services” columns have a strong influence on “ReAdmis”. RFE selected “Initial\_days”, “Services\_MRI” and “Stroke” in the top 3 most important variables in reference to the response variable “ReAdmis”. The final model used the response variable “ReAdmis” and the explanatory variables selected “Initial\_days”, “Services\_MRI” and “Stroke”.

3.  Provide a reduced logistic regression model.

Graphical user interface, application

Description automatically generated

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 and reduced logistic regression models, including the following elements:

•  the logic of the variable selection technique

Several statistical methods were used for the selection of variables. The variance inflation factor (VIF)was calculated to check for multicollinearity. VIF revealed that the various categories of “Services” that had been separated during preparation had high multicollinearity. VIF indicated that only “Services\_MRI” should remain from the group because it had a VIF lower than 10. The columns weren’t removed until further investigation.

The p-value should be lower that 0.05 on variables that are strong influencers, otherwise the influence of the variable could be due to random chance. The model in D1 shows several variables we can reliably say are strong not just due to chance, but p-values are known to swing wildly according to the other variables they are measured with, so PCA and RFE were run before removing any columns.

PCA, RFE, and correlation were all examined to make sure the best variables were chosen for the final model. The outputs are shown below in section E2. The PCA scree plot revealed that several PCAs stayed very close to 1 in the rubble, but the first 3 really stand above the rest. Correlation was checked with only the “Services” columns having high negative correlation. Finally, recursive feature elimination (RFE) revealed that “Services\_MRI” is indeed the correct column to keep out of the group with multicollinearity, and that the strongest variables in relation to “ReAdmis” are “Services\_MRI”, “Stroke” and “Initial\_days”.

•  the model evaluation metric

A confusion matrix was created to evaluate the model performance. The confusion matrix revealed that out of the 3000 rows separated for the test set, the model correctly predicted 2,918. The model predicted 1,892 true positives for the response variable “ReAdmis”. There were 1,026 false positives, meaning that the model predicted they would be false, and the actual data was false. The mode missed the target 82 times. It predicted a negative result 46 times when there was a positive, and the other 36 predictions said the result would be positive and it turned out to be negative.

The log-likelihood of the initial model with the variables from C2 is -274.86. The log-likelihood of the reduced model is -318.73. This means the initial model fits the data set better because it has a higher log-likelihood. The p-values of the final selection of variables were 0.0000 on all three accounts, meaning there is virtually no chance their relationship with the response variable is by random chance. The results of chi2 revealed the same thing. The chi2 values for both models are abnormally high, revealing the models are not a good fit for the data as a whole.

2.  Provide the output and any calculations of the analysis you performed, including a confusion matrix.

**Variance Inflation Factor**

Text

Description automatically generated with medium confidence

**Principal Component Analysis**

Graphical user interface, application, table

Description automatically generated

**PCA Scree Plot**

Chart, line chart

Description automatically generated

**Recursive Factor Extraction**

Text, letter

Description automatically generated

**Stats**

Table

Description automatically generated

**Correlation**

Graphical user interface, application, table

Description automatically generatedA picture containing chart

Description automatically generated

**Confusion Matrix**



Chart, treemap chart

Description automatically generated

**Final Model**

Chart, line chart

Description automatically generated

3.  Provide the code used to support the implementation of the logistic regression models.

logreg = LogisticRegression()

rfe = RFE(logreg,2)

rfe = rfe.fit(X\_train,y\_train)

list(zip(X\_train.columns, rfe.support\_, rfe.ranking\_))

LogReg1 = sm.GLM(y\_train,(sm.add\_constant(X\_train)), family = sm.families.Binomial())

LogReg1.fit().summary()

col = X\_train.columns[rfe.support\_]

X\_train\_sm = sm.add\_constant(X\_train[col])

LogReg2 = sm.GLM(y\_train,X\_train\_sm, family = sm.families.Binomial())

res = LogReg2.fit()

res.summary()

**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

•  an interpretation of coefficients of the statistically significant variables of the model

Using the coefficients from the final reduced model, we can see that every increase of x1 results in a 1.0480 increase of the logit(p) of “Initial\_days”, where p is the probablilty that y=1. Likewise, 1 increase of x will increase the logit(p) of “Stroke” by 1.3342 and “Services\_MRI” by 2.0310.

•  the statistical and practical significance of the model

The model and the intial visuals of the data reveal that the patients who had a longer initial visit to the hospital are more likely to be readmitted in the near future. The p-values reveal that the relationship of all of the independent variables is not by random chance. Thi chi2 and the Log-Likelihood results both indicate the models are not a good fit for the data. The bad fit of the data is probably due to a poor choice of initial variables or asking the wrong question for the data.

•  the limitations of the data analysis

The data was limited to mostly binary variables. The sigmoid line can look straight and limited when logistic models are created against binary independent variables. Fortunately, the binary variables that made it through RFE were easily incorporated into visuals with “Initial\_days” using different colors for the binary variables. No combination of variables produced a model with statistically great fit. “Initial\_days” is missing a significant portion of data between 20-45 days. It is highly unlikely that just as many people stayed 1-25 days as 45+. There is most likley a reason for the missing day values, and this may be one of the reasons the models were not good statistical fits.

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

The hospital should research patients who were readmitted to the hospital, and what caused the patients to stay more than 45 days on their initial visit. This could be because a longer stay indicates a more serious condition, which in turn raises the likelihood of returning to the hospital. The hospital may need to evaluate their procedures for sending long term cases home. They may need more after-care education for each patient. The hospital may need to ask staff about pressure from insurance companies to send people home early or what other factors might contribute to sending someone home before they are ready. The hospital should note that stroke patients and patients who had an MRI are more likely to be readmitted. They may need to be more careful in the initial evaluations to avoid return patients. Some of the factors may not have a resolution due to insurance or the need for space for new patients.

G. Panapto Video

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=e815287b-3190-4348-aecc-ae4b00329f7d

H.  List the web sources used to acquire data or segments of third-party code to support the application. Ensure the web sources are reliable.

Logistic Regression Analysis. Wayne W. LaMorte. May 31,2016. [*https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704-ep713\_multivariablemethods/bs704-ep713\_multivariablemethods5.html*](https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704-ep713_multivariablemethods/bs704-ep713_multivariablemethods5.html)

D208 Predictive Modeling Webinar Episode 4. Dr. William Sewell. *https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william\_sewell\_wgu\_edu/EYx20UarhiJLhlM0V5cTQ30BV0b2Qsmg2-sJrC9BzYQNmw?e=w81sHS*

D208 Predictive Modeling Webinar Episode 6. Dr. William Sewell. [*https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william\_sewell\_wgu\_edu/Ecby0spsQxRHj7KRsQk5L9UBYx9GqnaLwe94jv1moK-arw?e=dDZKbe&PreviousSessionID=235d6d97-ef99-0b21-789f-57b2d0450d4e*](https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william_sewell_wgu_edu/Ecby0spsQxRHj7KRsQk5L9UBYx9GqnaLwe94jv1moK-arw?e=dDZKbe&PreviousSessionID=235d6d97-ef99-0b21-789f-57b2d0450d4e)

sklearn.feature\_selection.RFE. Scikitlearn Developers (2021) [*https://scikit-learn.org/stable/modules/generated/sklearn.feature\_selection.RFE.html#sklearn.feature\_selection.RFE*](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html#sklearn.feature_selection.RFE)

A Simple Interpretation of Logistic regression Coefficients. Dina Jankovic. Sep 15, 2021. *https://towardsdatascience.com/a-simple-interpretation-of-logistic-regression-coefficients-e3a40a62e8cf*

1. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

Massaron, L., & Boschetti, A. (2016). *Regression analysis with python: Learn the art of regression analysis with python. Packt Publishing.*

J.  Demonstrate professional communication in the content and presentation of your submission.