**D208 Performance Assessment – Task 2**

NBM3 – NBM3 Task 2: Logistic Regression Modeling

Table of Contents

[**Part I: Research Question** 1](#_Toc131427479)

[A.  Describe the purpose of this data analysis by doing the following: 1](#_Toc131427480)

[**Part II: Method Justification** 2](#_Toc131427481)

[B.  Describe logistic regression methods by doing the following: 2](#_Toc131427482)

[**Part III: Data Preparation** 4](#_Toc131427483)

[C.  Summarize the data preparation process for logistic regression by doing the following: 4](#_Toc131427484)

[D.  Compare an initial and a reduced logistic regression model by doing the following: 12](#_Toc131427485)

[E.  Analyze the data set using your reduced logistic regression model by doing the following: 21](#_Toc131427486)

[**Part V: Data Summary and Implications** 30](#_Toc131427487)

[F.  Summarize your findings and assumptions by doing the following: 30](#_Toc131427488)

[**Part VI: Demonstration** 33](#_Toc131427489)

[G.  Provide a Panopto video recording that includes the presenter and a vocalized demonstration of the functionality of the code used for the analysis of the programming environment, including the following elements: **Video:** https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=0b424bb6-80ab-4b13-85ee-afd9016d84f9 33](#_Toc131427490)

[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. 34](#_Toc131427491)

[I.  Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized. 34](#_Toc131427492)

[J.  Demonstrate professional communication in the content and presentation of your submission 34](#_Toc131427493)

# **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.  
  
When working with a hospital it would be great to examine, “Which factors have a significant effect on readmission?” This is a topic that is hard to nail down as there are many factors that can contribute to re-admission with some categories that are unavoidable while others can be addressed. Helping the business understand which controllable factors may be contributing to readmission can help the hospital produce healthier patients while also reducing overall costs.

2.  Define the goals of the data analysis.

Note: Ensure that your goals are within the scope of your research question and are represented in the available data.  
  
Readmission to the hospital is a big factor because of the Hospital Readmissions Reduction Program (HRRP) act, which was established as part of the Affordable Care Act. Hospitals are required to report the data on their readmissions which, “… encourages hospitals to improve communication and care coordination to better engage patients and caregivers in discharge plans and, in turn, reduce avoidable readmissions.” (CMS 2023).

Understanding how various variables have an affect/correlation on hospital readmission even if they are not the specific cause is important to both the hospital as well as the patient. Understanding these relationships will help hospitals mitigate or remove aspects of their practice that increase readmission causing grief to the patient and higher costs for the hospital.

# **Part II: Method Justification**

## B.  Describe logistic regression methods by doing the following:

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

|  |  |
| --- | --- |
| Binary Response Variable | “Logistic regression assumes that the response variable only takes on two possible outcomes.” (Statology, 2020)  The total unique values of the response are two. Such as Yes/No, 1/0, etc. |
| Observations are Independent | “…the observations should not come from repeated measurements of the same individual or be related to each other in any way.” (Statology, 2020)  For example, if a dataset contains information about the heights of a group of people, each person's height would be considered an observation. These cannot be duplicated. |
| No Multicollinearity of Explanatory Variables | “Multicollinearity occurs when two or more explanatory variables are highly correlated to each other, such that they do not provide unique or independent information in the regression model.” (Statology, 2020)  Example:  Temperature and humidity: These variables are often highly correlated, as high temperatures tend to be associated with high humidity. |
| Linear Relationship Between Explanatory Variables and the Logit of the Response Variable | “…there exists a linear relationship between each explanatory variable and the logit of the response variable. Recall that the logit…” (Statology, 2020)  Logit Formula is: |
| There are No Extreme Outliers | This one is self-explanatory in that there are no extreme outliers because they could influence how the observations are perceived in the dataset. |
| Sample Size is Sufficiently Large | “Logistic regression assumes that the sample size of the dataset if large enough to draw valid conclusions from the fitted logistic regression model.” (Statology, 2020) |

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

1. A great benefit of using languages like Python and R (*I chose Python*) is that they are easy programming languages to pick up and have a wide scientific community support for packages geared toward analysis, science, and machine learning.
2. Another benefit of using these programming languages is reusability. Because your work is done in code form it can be reproduced and run again and again with small tweaks without having to start from scratch.

3.  Explain why logistic regression is an appropriate technique to analyze the research question summarized in part I.  
  
The reason that logical regression is the appropriate choice for this analysis over say linear regression is that the target/dependent variable we are analyzing is binary/categorical. ReAdmis has the values Yes/No only. It falls into the realm of either/or data collecting. We want to analyze the relationship between ReAdmis and multiple explanatory variables, logistical regression is designed to compare categorical variables with numerical or other categorial explanatory variables.

# **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.  
  
The preparation will be minor as this dataset is a continuation of the medical\_data that was cleaned in D206. Below will still be done to clean a little further.

The data preparation process steps used for MLR (multiple linear regression) included:

1. Importing dataset into Python as a Pandas Data Frame
2. Check for missing values.
3. Compare poorly named series (columns) with the WGU provided data dictionary. Rename ones that need clarity.
4. Convert chr and str datatypes to Category.
5. ReExpress logical yes/no binaries categorical data to 1,0 for processing
6. Calculate and remove Categoricals with high cardinality values as these can be explanatory variables that lead to over fitting of models.  
     
   **Code for Data Preparation:**

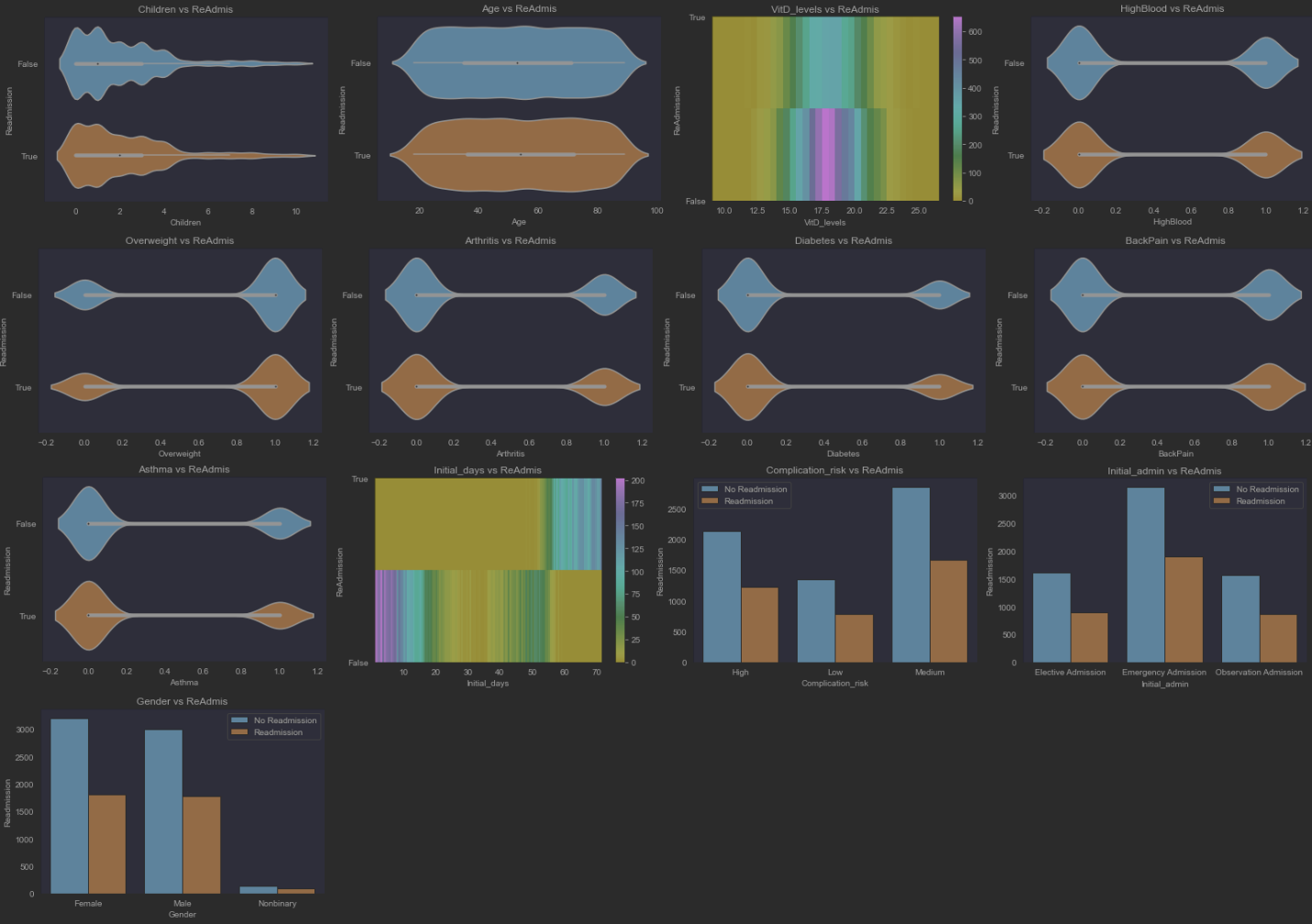
original\_medical = medical\_clean\_data = pd.read\_csv('./Data/Medical/medical\_clean.csv')  
#del medical\_clean\_data[medical\_clean\_data.columns[0]]  
  
any\_missing\_values = medical\_clean\_data.isna().values.any()  
if not any\_missing\_values:  
 print('Medical data does NOT contain any missing values\n')  
else:  
 print('Medical data CONTAINS missing values.\n')  
  
column\_renames = {  
 'Item1': 'Timely\_Admission'  
 ,'Item2': 'Timely\_Treatment'  
 ,'Item3': 'Timely\_Visits'  
 ,'Item4': 'Reliability'  
 ,'Item5': 'Options'  
 ,'Item6': 'Hours\_Of\_Treatment'  
 ,'Item7': 'Courteous\_Staff'  
 ,'Item8': 'Listening' #Evidence of active listening from Doctor  
}  
medical\_clean\_data.rename(columns=column\_renames, inplace=True)  
#medical\_clean\_data  
  
removal\_attributes = ['CaseOrder', 'Customer\_id', 'Interaction', 'UID',  
 'Zip', 'Lat', 'Lng', 'City', 'State', 'County',  
 'Area', 'Job', 'Marital', 'Population', 'TimeZone']  
  
medical\_clean\_data = medical\_clean\_data.drop(columns=removal\_attributes)  
  
category\_dtype = 'category'  
convert\_to\_category = {  
 'Gender': category\_dtype,  
 'ReAdmis': category\_dtype,  
 'Soft\_drink': category\_dtype,  
 'Initial\_admin': category\_dtype,  
 'HighBlood': category\_dtype,  
 'Stroke': category\_dtype,  
 'Complication\_risk': category\_dtype,  
 'Overweight': category\_dtype,  
 'Arthritis': category\_dtype,  
 'Diabetes': category\_dtype,  
 'Hyperlipidemia': category\_dtype,  
 'BackPain': category\_dtype,  
 'Anxiety': category\_dtype,  
 'Allergic\_rhinitis': category\_dtype,  
 'Reflux\_esophagitis': category\_dtype,  
 'Asthma': category\_dtype,  
 'Services': category\_dtype,  
 'Timely\_Admission': category\_dtype,  
 'Timely\_Treatment': category\_dtype,  
 'Timely\_Visits': category\_dtype,  
 'Reliability': category\_dtype,  
 'Options': category\_dtype,  
 'Hours\_Of\_Treatment': category\_dtype,  
 'Courteous\_Staff': category\_dtype,  
 'Listening': category\_dtype  
}  
  
medical\_clean\_data = medical\_clean\_data.astype(convert\_to\_category)  
  
#Logical categorical variables converted to numerical  
columns\_to\_reexpress = ['ReAdmis', 'Soft\_drink', 'HighBlood', 'Stroke',  
 'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia',  
 'BackPain', 'Anxiety', 'Allergic\_rhinitis', 'Reflux\_esophagitis',  
 'Asthma']  
for column in columns\_to\_reexpress:  
 medical\_clean\_data[column] = medical\_clean\_data[column].map({'Yes': 1, 'No': 0 }).astype(np.int64)  
  
categorical\_medical\_data = medical\_clean\_data[convert\_to\_category.keys()]  
high\_cardinalities = categorical\_medical\_data.nunique() > 3 #(> 3-5 Levels)  
high\_cardinalities = high\_cardinalities[high\_cardinalities == True]  
high\_cardinalities = list(high\_cardinalities.index.values)  
print('Categoricals with high cardinality to be removed:')  
print(high\_cardinalities)  
print('\n')  
  
medical\_clean\_data = medical\_clean\_data.drop(columns=high\_cardinalities)  
  
low\_cardinalities = [item for item in list(convert\_to\_category.keys()) if item not in high\_cardinalities]  
print('Categoricals remaining:')  
print(low\_cardinalities)  
print('\n')  
  
#Re-level ordinal/nominal categoricals  
complication\_risk\_dummies = pd.get\_dummies(data=medical\_clean\_data['Complication\_risk'], drop\_first=True)  
  
medical\_clean\_data['complication\_risk\_medium'] = complication\_risk\_dummies['Medium']  
medical\_clean\_data['complication\_risk\_low'] = complication\_risk\_dummies['Low']  
  
initial\_admission\_dummies = pd.get\_dummies(data=medical\_clean\_data['Initial\_admin'], drop\_first=True)  
  
medical\_clean\_data['initial\_admission\_emergency'] = initial\_admission\_dummies['Emergency Admission']  
medical\_clean\_data['initial\_admission\_observation'] = initial\_admission\_dummies['Observation Admission']  
  
gender\_dummies = pd.get\_dummies(data=medical\_clean\_data['Gender'], drop\_first=True)  
  
medical\_clean\_data['gender\_male'] = gender\_dummies['Male']  
medical\_clean\_data['gender\_non\_binary'] = gender\_dummies['Nonbinary']  
  
regression\_variables = ['Children', 'Age', 'VitD\_levels', 'HighBlood', 'Overweight', 'Arthritis', 'Diabetes', 'BackPain', 'Asthma', 'Initial\_days', 'ReAdmis', 'complication\_risk\_medium', 'complication\_risk\_low', 'initial\_admission\_emergency', 'initial\_admission\_observation', 'gender\_male','gender\_non\_binary']

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.  
  
Graphical user interface, text

Description automatically generated  
  
  
Description of the statistics in the above screenshot. This will just be basic checks against data like the min, max and averages.

|  |  |
| --- | --- |
| **Variable** | **Summary Description Analysis** |
| Children | The average seems to be 2 kids with an extreme of 10 kids. |
| Age | The age averages around 53. Age will likely be a contributing factor when checked against ReAdmis. |
| Income | This hospital seems to patients that are below the median household income consisting of $40490. |
| VitD\_levels | Vitamin D levels average is almost 18 with an extreme of 26. Vitamin D is an important health vitamin, it influence is expected to be present. |
| Doc\_visits | The number of times a patient was seen during their initial visit is averaged at 5 and a max of 9. During the regression analysis it will be interesting to see this influence on ReAdmis |
| Full\_meals\_eaten | Full meals eaten during hospitalization average 1 and go as high as 9. This has an initial sense of being a factor towards having a correlation to ReAdmis. |
| vitD\_supp | The average for supplying Vitamin D during visit is an average of 0.39 indicating that this isn’t something that happens very often. |
| Initial\_days | Initial days in the hospital averages 26 days. This seems extremely high for an average at the hospital. |
| TotalCharge | The total charges for the hospital visit have an average of $5312 with a high of $9180. These are lower than the additional charges which make the readmission feel extreme. |
| Additional\_charges | It has average additional charges of $12934 and as extreme as $30566. This is why this variable was chosen as the dependent variable for the linear regression mode. |
| ReAdmis | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Soft\_drink | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| HighBlood | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Stroke | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Overweight | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Arthritis | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Diabetes | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Hyperlipidemia | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| BackPain | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Anxiety | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Allergic\_rhinitis | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Reflux\_esophagitis | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to readmission. |
| Asthma | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to additional readmission. |
| complication\_risk\_medium | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to additional charges. |
| complication\_risk\_low | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to additional charges. |
| initial\_admission\_emergency | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to additional charges. |
| initial\_admission\_observation | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to additional charges. |
| gender\_male | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to additional charges. |
| gender\_non\_binary | This is a categorical variable that was converted to numerical to use as an explanatory variable. We are simply seeing if the existence of what it represents has any correlation to additional charges. |

3.  Generate univariate and bivariate visualizations of the distributions of the dependent and independent variables, including the dependent variable in your bivariate visualizations.  
  
Code is available in the Jupyter Notebook included in submission.   
File: `*D208 - nbm3-task2-logistic-regression-modeling.ipynb*`  
  
**Univariate Visualizations:**  
  
A picture containing text, scoreboard, crossword puzzle

Description automatically generated  
  
**Bivariate Visualizations:**  
  


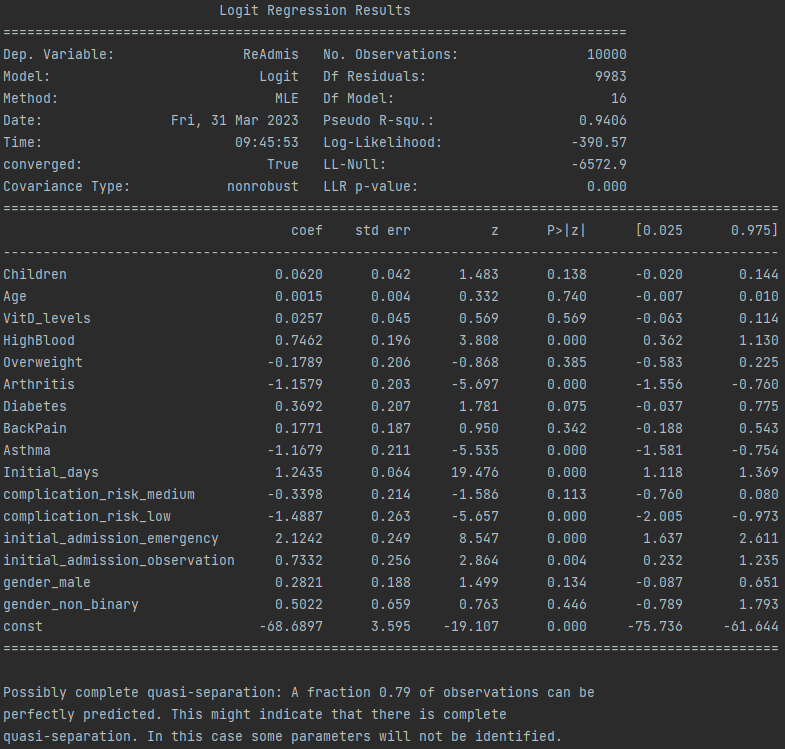
4.  Describe your data transformation goals that align with your research question and the steps used to transform the data to achieve the goals, including the annotated code.  
  
The transformation of the original dataset to the prepared dataset for the Logistic Regression modeling was done in a way to help reduce variables that could potentially lead to over fitting. The dataset is prepared so that it can be used for Logical Linear Regression. To be prepared for Logistic Regression we had to transform categorical variables into mapped numerical values and our dependent variable need to be binary in nature. ReAdmis was chosen as this is a Yes or No datatype which could easily be converted into 1 (Yes) or 0 (No).

5.  Provide the prepared data set as a CSV file.  
See submitted file: *‘initial-logistic-medical-model-dataset.csv’*

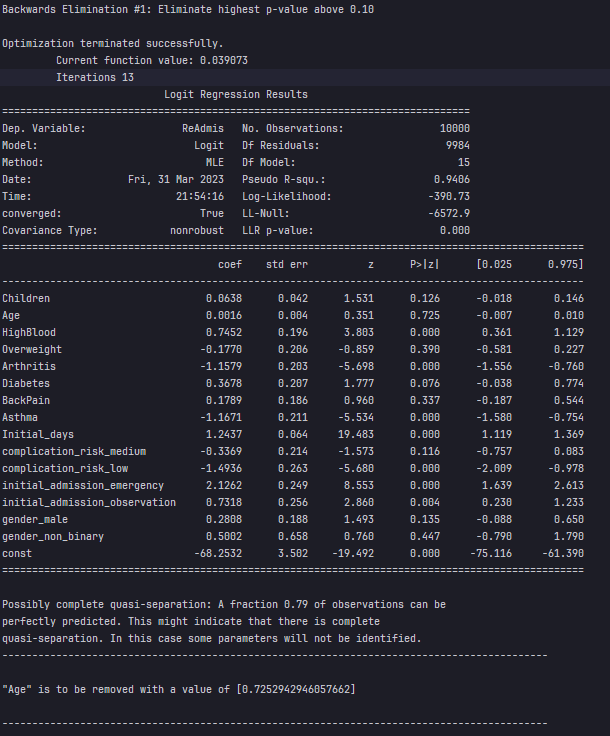
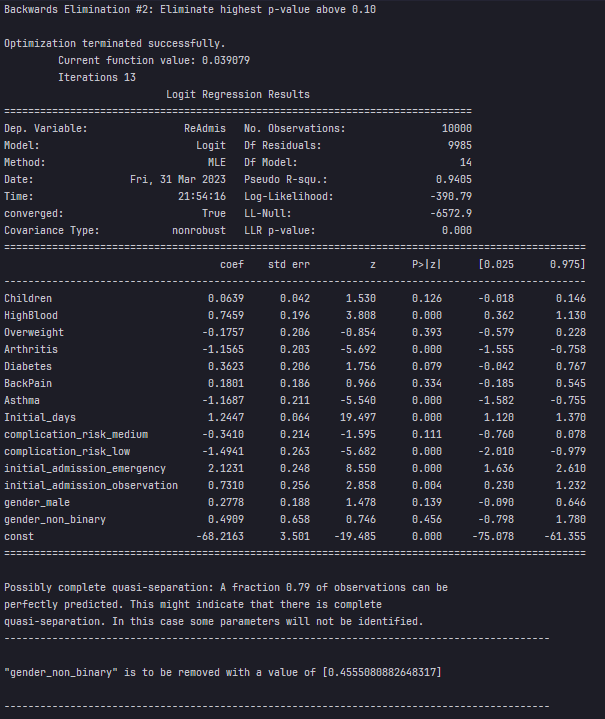
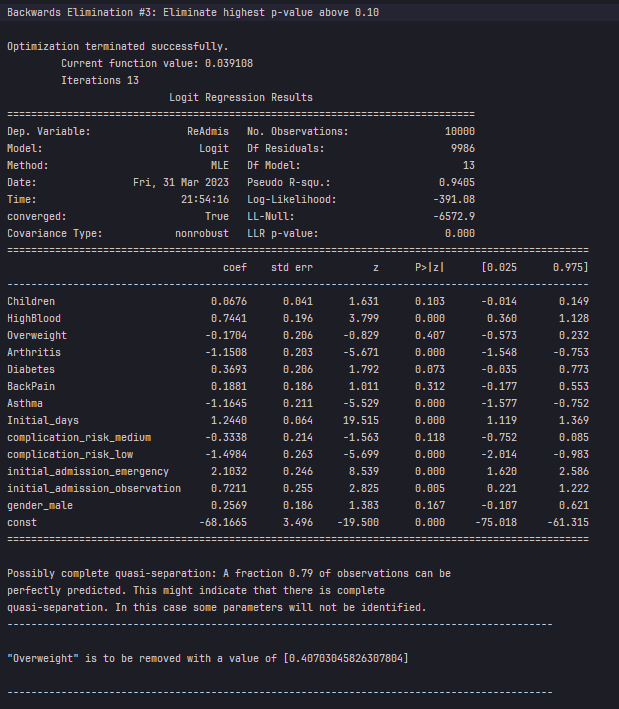
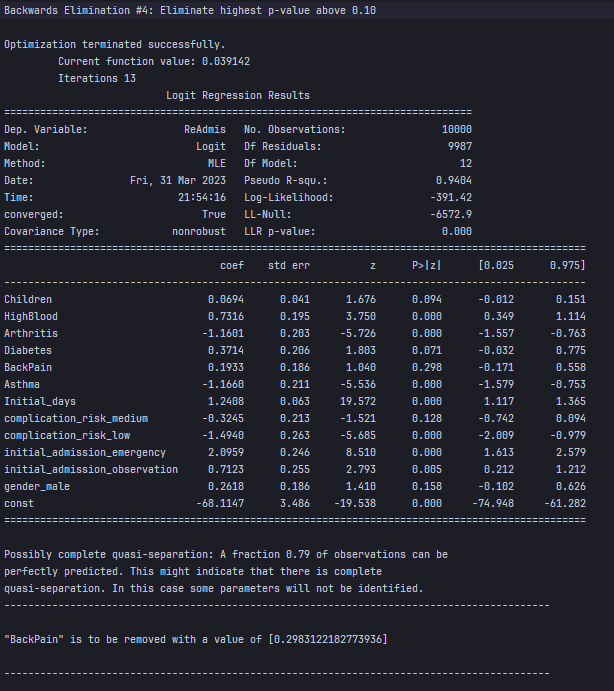
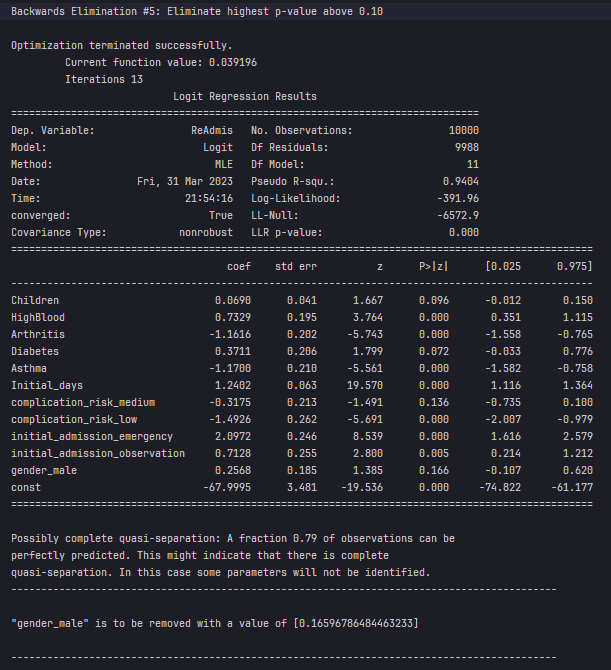
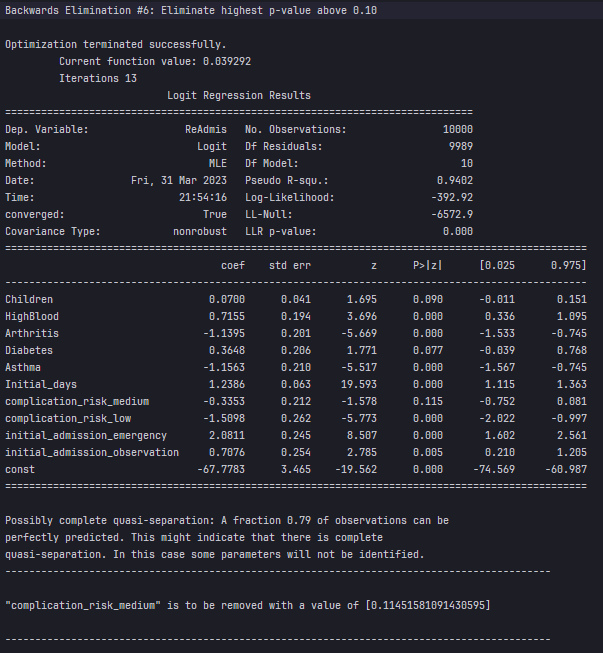
**Part IV: Model Comparison and Analysis**

## 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.  
  
The goal in constructing our initial logistic regression model is including as many explanatory variables as we can again ‘ReAdmis’ dependent variable to see if any of them show any signs of significant correlation.  
  
Code:  
dependent = prepared\_medical\_data['ReAdmis']  
  
#In logistic regression, adding a constant term to the predictor variables is important for estimating the intercept term in the logistic regression equation.  
explanatory\_variables = prepared\_medical\_data.drop('ReAdmis', axis=1).assign(const=1)  
  
initial\_logic\_model: Logit = Logit(dependent, explanatory\_variables)  
initial\_logit\_result = initial\_logic\_model.fit()  
print(initial\_logit\_result.summary())

The results of that initial regression model are seen in below screenshot:  
  


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 approaches were taken for the selecting which features to carry over to the reduced regression model.   
  
The first step was using Variance Inflation Facto (VIF) get rid of explanatory variables has high multicollinearity. “…high or near perfect multicollinearity not only often renders coefficient estimations impossible (matrix inversions is not work), but also, when it is feasible, it will be affected by imprecision in coefficient estimation, leading to large standard of errors of coefficients.” (Luca Massaron, 2016)  
  
The second approach was Backward Elimination (Backward Stepwise). We are using Backwards version as we already have a model filled with explanatory variables as opposed to adding them one at a time (Forward Stepwise).   
  
The Backward elimination required us to run our Logit function and review the summary results. The p-values above 0.10 were up for removal but for the current iteration the biggest p-value greater than 0.10 was removed. Once this explanatory variable was removed from the dataset the Logit function (Model) was run against the reduced explanatory variable dataset. This process continued until there were no more explanatory variables with a p-value greater than 0.10.  
  
Once these two methods were finished our explanatory variable selection was complete and we could create our Reduced Logistic Regression Model based on the reduced explanatory variables and our dependent variable ‘ReAdmis’

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.  
  
**Feature Selection per model during Backward Stepwise Process:**  
  
  
  
  
  
  
  
  
  
  
  
  
  
Backwards Elimination #7

Final Elimination process is complete.

## 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  
  
The evaluation metric we will be comparing is MSE (Mean Squared Error). This metric is the average squared difference between the predicted values and the actual values. MSE is an important metric as it allows the quantification of the regression model in a way that allows us understand if the model is accurately making predictions. In general a lower MSE score indicates that the regression model is more accurate.  
  
**Regression Model MSEs:**  
  
Initial Regression Model: 0.017  
Reduced Regression Model: 0.017

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

•   confusion matrix  
  
**Initial Model Confusion Matrix:**

|  |  |  |
| --- | --- | --- |
|  | ReAdmis | No ReAdmis |
| ReAdmis | 1912 | 22 |
| No ReAdmis | 29 | 1037 |

**Reduced Model Confusion Matrix:**

|  |  |  |
| --- | --- | --- |
|  | ReAdmis | No ReAdmis |
| ReAdmis | 1912 | 22 |
| No ReAdmis | 29 | 1037 |

•   accuracy calculation  
  
Accuracy Score of Initial Logistic Regression test datasets is: 0.98  
Accuracy Score of Reduced Logistic Regression test datasets is: 0.98

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.  
  
Please see submitted file: ‘*D208-nbm3-task-2-logistic-regression-modeling.ipynb*’, commented and runnable code.  
  
**Below is also the full projects code:**  
  
#!/usr/bin/env python

# coding: utf-8

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LogisticRegression

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

from statsmodels.api import Logit

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

from sklearn.metrics import confusion\_matrix

original\_medical = medical\_clean\_data = pd.read\_csv('./Data/Medical/medical\_clean.csv')

any\_missing\_values = medical\_clean\_data.isna().values.any()

if not any\_missing\_values:

print('Medical data does NOT contain any missing values\n')

else:

print('Medical data CONTAINS missing values.\n')

column\_renames = {

'Item1': 'Timely\_Admission'

,'Item2': 'Timely\_Treatment'

,'Item3': 'Timely\_Visits'

,'Item4': 'Reliability'

,'Item5': 'Options'

,'Item6': 'Hours\_Of\_Treatment'

,'Item7': 'Courteous\_Staff'

,'Item8': 'Listening' #Evidence of active listening from Doctor

}

medical\_clean\_data.rename(columns=column\_renames, inplace=True)

removal\_attributes = ['CaseOrder', 'Customer\_id', 'Interaction', 'UID',

'Zip', 'Lat', 'Lng', 'City', 'State', 'County',

'Area', 'Job', 'Marital', 'Population', 'TimeZone']

medical\_clean\_data = medical\_clean\_data.drop(columns=removal\_attributes)

category\_dtype = 'category'

convert\_to\_category = {

'Gender': category\_dtype,

'ReAdmis': category\_dtype,

'Soft\_drink': category\_dtype,

'Initial\_admin': category\_dtype,

'HighBlood': category\_dtype,

'Stroke': category\_dtype,

'Complication\_risk': category\_dtype,

'Overweight': category\_dtype,

'Arthritis': category\_dtype,

'Diabetes': category\_dtype,

'Hyperlipidemia': category\_dtype,

'BackPain': category\_dtype,

'Anxiety': category\_dtype,

'Allergic\_rhinitis': category\_dtype,

'Reflux\_esophagitis': category\_dtype,

'Asthma': category\_dtype,

'Services': category\_dtype,

'Timely\_Admission': category\_dtype,

'Timely\_Treatment': category\_dtype,

'Timely\_Visits': category\_dtype,

'Reliability': category\_dtype,

'Options': category\_dtype,

'Hours\_Of\_Treatment': category\_dtype,

'Courteous\_Staff': category\_dtype,

'Listening': category\_dtype

}

medical\_clean\_data = medical\_clean\_data.astype(convert\_to\_category)

#Logical categorical variables converted to numerical

columns\_to\_reexpress = ['ReAdmis', 'Soft\_drink', 'HighBlood', 'Stroke',

'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia',

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

'Asthma']

for column in columns\_to\_reexpress:

medical\_clean\_data[column] = medical\_clean\_data[column].map({'Yes': 1, 'No': 0 }).astype(np.int64)

categorical\_medical\_data = medical\_clean\_data[convert\_to\_category.keys()]

high\_cardinalities = categorical\_medical\_data.nunique() > 3 #(> 3-5 Levels)

high\_cardinalities = high\_cardinalities[high\_cardinalities == True]

high\_cardinalities = list(high\_cardinalities.index.values)

print('Categoricals with high cardinality to be removed:')

print(high\_cardinalities)

print('\n')

medical\_clean\_data = medical\_clean\_data.drop(columns=high\_cardinalities)

low\_cardinalities = [item for item in list(convert\_to\_category.keys()) if item not in high\_cardinalities]

print('Categoricals remaining:')

print(low\_cardinalities)

print('\n')

#Re-level ordinal/nominal categoricals

complication\_risk\_dummies = pd.get\_dummies(data=medical\_clean\_data['Complication\_risk'], drop\_first=True)

medical\_clean\_data['complication\_risk\_medium'] = complication\_risk\_dummies['Medium']

medical\_clean\_data['complication\_risk\_low'] = complication\_risk\_dummies['Low']

initial\_admission\_dummies = pd.get\_dummies(data=medical\_clean\_data['Initial\_admin'], drop\_first=True)

medical\_clean\_data['initial\_admission\_emergency'] = initial\_admission\_dummies['Emergency Admission']

medical\_clean\_data['initial\_admission\_observation'] = initial\_admission\_dummies['Observation Admission']

gender\_dummies = pd.get\_dummies(data=medical\_clean\_data['Gender'], drop\_first=True)

medical\_clean\_data['gender\_male'] = gender\_dummies['Male']

medical\_clean\_data['gender\_non\_binary'] = gender\_dummies['Nonbinary']

regression\_variables = ['Children', 'Age', 'VitD\_levels', 'HighBlood', 'Overweight', 'Arthritis', 'Diabetes', 'BackPain', 'Asthma', 'Initial\_days', 'ReAdmis', 'complication\_risk\_medium', 'complication\_risk\_low', 'initial\_admission\_emergency', 'initial\_admission\_observation', 'gender\_male','gender\_non\_binary']

prepared\_medical\_data = medical\_clean\_data[regression\_variables]

prepared\_medical\_data.describe(include='all')

print('Univariate Numerical Variables')

prepared\_medical\_data.hist(figsize=(15, 15))

plt.savefig('univariate-histograms.jpg')

plt.tight\_layout()

print('Bivariate Numerical Variables')

print(regression\_variables)

def bivariate\_count\_plot\_vs\_readmission(data: pd.DataFrame, x: str) -> None:

plt.title(f'{x} vs ReAdmis')

sns.countplot(data=data, x=x, hue='ReAdmis')

plt.legend(['No Readmission', 'Readmission'])

plt.xlabel(x)

plt.ylabel('Readmission')

plt.show()

def bivariate\_violin\_plot\_vs\_readmission(data: pd.DataFrame, x: str) -> None:

plt.title(f'{x} vs ReAdmis')

sns.violinplot(data=data, x=x, y='ReAdmis', orient='h')

plt.xlabel(f'{x}')

plt.ylabel('Readmission')

plt.yticks([0,1], ['False', 'True'])

plt.show()

def bivariate\_2d\_hist\_vs\_readmission(data: pd.DataFrame, x: str, bin\_increment: float = 1) -> None:

plt.rcParams['axes.grid'] = False #stops the deprecation warning

start = np.floor(prepared\_medical\_data[x].min())

end = np.ceil(prepared\_medical\_data[x].max())

x\_bins = np.arange(start, end, bin\_increment)

y\_logical\_bins = np.arange(0, 1.01, .5)

plt.title(f'{x} vs ReAdmis')

plt.hist2d(data=data, x=x, y='ReAdmis', bins=[x\_bins, y\_logical\_bins], cmap='viridis\_r')

plt.grid(False)

plt.colorbar()

plt.xlabel(x)

plt.ylabel('ReAdmission')

plt.yticks([0,1], ['False', 'True'])

plt.show()

bivariate\_violin\_plot\_vs\_readmission(data=prepared\_medical\_data, x='Children')

bivariate\_violin\_plot\_vs\_readmission(data=prepared\_medical\_data, x='Age')

bivariate\_2d\_hist\_vs\_readmission(data=prepared\_medical\_data, x='VitD\_levels', bin\_increment=.5)

bivariate\_violin\_plot\_vs\_readmission(data=prepared\_medical\_data, x='HighBlood')

bivariate\_violin\_plot\_vs\_readmission(data=prepared\_medical\_data, x='Overweight')

bivariate\_violin\_plot\_vs\_readmission(data=prepared\_medical\_data, x='Arthritis')

bivariate\_violin\_plot\_vs\_readmission(data=prepared\_medical\_data, x='Diabetes')

bivariate\_violin\_plot\_vs\_readmission(data=prepared\_medical\_data, x='BackPain')

bivariate\_violin\_plot\_vs\_readmission(data=prepared\_medical\_data, x='Asthma')

bivariate\_2d\_hist\_vs\_readmission(data=prepared\_medical\_data, x='Initial\_days', bin\_increment=.5)

bivariate\_count\_plot\_vs\_readmission(data=medical\_clean\_data, x='Complication\_risk')

bivariate\_count\_plot\_vs\_readmission(data=medical\_clean\_data, x='Initial\_admin')

bivariate\_count\_plot\_vs\_readmission(data=medical\_clean\_data, x='Gender')

prepared\_medical\_data.to\_csv('./initial-logistic-medical-model-dataset.csv')

dependent = prepared\_medical\_data['ReAdmis']

#In logistic regression, adding a constant term to the predictor variables is important for estimating the intercept term in the logistic regression equation.

initial\_explanatory\_variables = prepared\_medical\_data.drop('ReAdmis', axis=1).assign(const=1)

initial\_logic\_model: Logit = Logit(dependent, initial\_explanatory\_variables)

initial\_logit\_result = initial\_logic\_model.fit()

print(initial\_logit\_result.summary())

'''

Variance Inflaction Factor (VIF):

VIF Values:

1: No Collinearity

>1: Increasing levels of collinearity

>5-10: Severe Collinearity removal recommended.

'''

def check\_vif(data: pd.DataFrame) -> pd.DataFrame:

data = {

'Feature': data.columns,

'VIF': [variance\_inflation\_factor(data.values, i) for i in range(len(data.columns))]

}

results = pd.DataFrame(data)

return results

vif\_results = check\_vif(prepared\_medical\_data.drop('ReAdmis', axis=1))

print(vif\_results)

#VitD\_levels is well over a value of 10 coming in at 17+

#remove any items with a VIF score of 10 or greater.

vif\_to\_remove = list(vif\_results[vif\_results['VIF'] >= 10]['Feature'])

prepared\_medical\_data.drop(columns=vif\_to\_remove, inplace=True)

'''

Backwards Elimination Process:

The backward elimination process starts by fitting a logistic regression model with all the predictors, and then evaluating the significance of each predictor using its p-value.

The predictor with the highest p-value is removed from the model, and the new model is fitted with the remaining predictors.

The process is repeated until no more predictors can be removed without significantly affecting the model's performance.

Goal: Remove the highest p-value that remains above 0.10

'''

def backwards\_elimination(title: str, dependent: pd.Series, explanatory\_variables: pd.DataFrame) -> (bool, pd.DataFrame):

print(title, '\n')

removal\_limit = .10

more\_eliminations = False

#print(f'Remove the highest p-value that remains above {removal\_limit:.2f}')

logit\_model: Logit = Logit(dependent, explanatory\_variables)

model\_result = logit\_model.fit()

print(model\_result.summary())

p\_values: pd.Series = model\_result.pvalues

filtered\_p\_values = p\_values[p\_values > removal\_limit].sort\_values(ascending=False)

if len(filtered\_p\_values) > 0:

biggest\_value\_name = filtered\_p\_values.index[0]

biggest\_value = filtered\_p\_values[0]

print('-------------------------------------------------------------------------------------------')

print(f'\n"{biggest\_value\_name}" is to be removed with a value of [{biggest\_value}]\n')

print('-------------------------------------------------------------------------------------------')

print('\n')

explanatory\_variables.drop(biggest\_value\_name, axis=1, inplace=True)

more\_eliminations = True

return more\_eliminations, explanatory\_variables

else:

print('\nNo Backwards Eliminations are required.\n')

return more\_eliminations, explanatory\_variables

dependent = prepared\_medical\_data['ReAdmis']

#In logistic regression, adding a constant term to the predictor variables is important for estimating the intercept term in the logistic regression equation.

initial\_explanatory\_variables = prepared\_medical\_data.drop('ReAdmis', axis=1).assign(const=1)

elimination\_count = 1

continue\_running\_backwards\_elimination = True

while continue\_running\_backwards\_elimination:

title = f'Backwards Elimination #{elimination\_count}: Eliminate highest p-value above 0.10'

continue\_processing, filtered\_explanatory\_variables = backwards\_elimination(title, dependent, initial\_explanatory\_variables)

reduced\_explanatory\_variables = filtered\_explanatory\_variables

elimination\_count = elimination\_count + 1

continue\_running\_backwards\_elimination = continue\_processing

print(f'Remaining explanatory variables are [{reduced\_explanatory\_variables.columns}]')

dependent = prepared\_medical\_data['ReAdmis']

reduced\_logit\_model = Logit(dependent, reduced\_explanatory\_variables)

reduced\_logit\_result = reduced\_logit\_model.fit()

print(reduced\_logit\_result.summary())

'''

This will calculation values needed for Analysing a Logistic Regression Model

This will return the MSE (Mean Squared Error, Confusion Matrix and Accuracy Score based on a Dependent variable and Explanatory Variables

'''

def calculate\_evaluation\_info(target: pd.Series, explanatory\_variables: pd.DataFrame) -> (float, object, float):

explanatory\_train, explanatory\_test, dependent\_train, dependent\_test = train\_test\_split(explanatory\_variables, target, test\_size=0.3, random\_state=42)

logistic\_regression = LogisticRegression(max\_iter=1000)

logistic\_regression.fit(explanatory\_train, dependent\_train)

dependent\_predictions = logistic\_regression.predict(explanatory\_test)

print(f'"ReAdmis" Logistic Regression Predictions with test data. Top 10')

print(dependent\_predictions[:10])

mse = mean\_squared\_error(dependent\_test, dependent\_predictions)

c\_matrix = confusion\_matrix(dependent\_test, dependent\_predictions)

accuracy\_score = logistic\_regression.score(explanatory\_test, dependent\_test)

return mse, c\_matrix, accuracy\_score

initial\_mse, initial\_confusion\_matrix, initial\_accuracy\_score = calculate\_evaluation\_info(dependent, initial\_explanatory\_variables)

print(f'Initial Regression Models MSE: [{initial\_mse}]\n')

print('Initial Logic Regression Model Confusion Matrix:')

print(initial\_confusion\_matrix)

print(f'\nAccuracy Score of Initial Logistic Regression test datasets is: {initial\_accuracy\_score:.2f}\n')

print('\n')

reduced\_mse, reduced\_confusion\_matrix, reduced\_accuracy\_score = calculate\_evaluation\_info(dependent, reduced\_explanatory\_variables)

print(f'Reduced Regression Models MSE: [{reduced\_mse}]\n')

print('Reduced Logic Regression Model Confusion Matrix:')

print(reduced\_confusion\_matrix)

print(f'\nAccuracy Score of Reduced Logistic Regression test datasets is: {reduced\_accuracy\_score:.2f}\n')

'''

Odd Ration:

Compares the odds of an event occurring in one group to the odds of the same event occurring in another group.

Meanings:

1 - Event is likely in both groups

>1 - Event is likely in 1st group

<1 - Event is likely in 2nd group

Sigmoid(Logic Function):

Maps any input value to a value between 0 and 1.

This f(x) has an S-shaped curve, which allows it to be used to model probability of events occurring.

Interpretations:

The probability of the dependent(target) variable takin on the value 1, given the values of

independent variables.

Fun Factoid: This f(x) is widely used in neural networks. Output of a neuron is converted to 0 and 1 for

probability interpretation.

'''

def calculate\_odds\_ration\_and\_sigmoid(coefficient: float) -> (float, float):

odds\_ration = np.exp(coefficient)

sigmoid = 1 / (1 + np.exp(-coefficient))

return round(odds\_ration, 4), round(sigmoid, 4)

#print(reduced\_logit\_result.params.index)

print('Initial Model Odds Ration')

for index in initial\_logit\_result.params.index:

odds\_ratio, predicted\_probability = calculate\_odds\_ration\_and\_sigmoid(initial\_logit\_result.params[index])

print(f'Odd ratio for {index} is {odds\_ratio}. The change in odds for Readmission (ReAdmis) is {predicted\_probability}.')

print('\n')

print('Reduced Model Odds Ratios')

for index in reduced\_logit\_result.params.index:

odds\_ratio, predicted\_probability = calculate\_odds\_ration\_and\_sigmoid(reduced\_logit\_result.params[index])

print(f'Odd ratio for {index} is {odds\_ratio}. The change in odds for Readmission (ReAdmis) is {predicted\_probability}.')

## **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 the coefficients of the reduced model  
  
We can use odds ratio and logic function to make some interpretations about the reduced model coefficients.

* **Children**: For each additional child a patient has, the odds of readmission increase by 7.6%. The sigmoid function indicates that having children is associated with a 51.8% chance of readmission.
* **HighBlood**: Patients with high blood pressure are twice as likely to be readmitted compared to those without high blood pressure. The sigmoid function suggests that having high blood pressure is associated with a 66.9% chance of readmission.
* **Arthritis**: Patients with arthritis are about 68% less likely to be readmitted compared to those without arthritis. The sigmoid function indicates that having arthritis is associated with a 24.3% chance of readmission.
* **Diabetes**: Patients with diabetes are about 45% more likely to be readmitted compared to those without diabetes. The sigmoid function suggests that having diabetes is associated with a 59.2% chance of readmission.
* **Asthma**: Patients with asthma are about 68% less likely to be readmitted compared to those without asthma. The sigmoid function indicates that having asthma is associated with a 23.97% chance of readmission.
* **Initial\_days**: For each additional day a patient stays in the hospital initially, the odds of readmission increase by 243.1%. The sigmoid function indicates that staying in the hospital initially is associated with a 77.4% chance of readmission.
* complication\_risk\_low: Patients with a low risk of complications are about 73% less likely to be readmitted compared to those with a medium or high risk of complications. The sigmoid function suggests that having a low risk of complications is associated with a 21.2% chance of readmission.
* **initial\_admission\_emergency**: Patients who were admitted to the hospital as an emergency case are about 8 times more likely to be readmitted compared to those who were admitted as non-emergency cases. The sigmoid function indicates that being admitted as an emergency case is associated with an 88.9% chance of readmission.
* **initial\_admission\_observation**: Patients who were admitted to the hospital for observation are about twice as likely to be readmitted compared to those who were admitted for other reasons. The sigmoid function suggests that being admitted for observation is associated with a 66.7% chance of readmission.

•   the statistical and practical significance of the reduced model  
  
The logistic regression model summary shows that the model is significant with a p-value less than 0.05. This means that the model is statistically significant, and the independent variables have a significant relationship with the dependent variable.  
  
The pseudo R-squared value of 0.94 indicates that the model has good predictive power. The higher the R-squared value, the better the model is at predicting the dependent variable.  
  
The z-score and p-value for each variable determine the statistical significance of that variable. A p-value less than 0.05 indicates that the variable is statistically significant, while a p-value greater than 0.05 indicates that the variable is not statistically significant. In the final reduced Logistic Regression model nearly all the explanatory variables are statistically significant with Children and Diabetes 0.02 higher than statistically significant.

•   the limitations of the data analysis  
  
The major limitation to this and I suspect rest of the courses is that the datasets for educational purposes are kind of small. It would be interesting if the hospital itself had more historical data for decades and then run this Logistic Regression analysis again and see if the model holds.

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

1. Monitor patients with high blood pressure more closely: The analysis shows that patients with high blood pressure are more likely to be readmitted. Hospitals can develop protocols to monitor and manage blood pressure more closely, such as providing education on healthy lifestyle habits, offering medication management programs, and providing follow-up care after discharge.
2. Develop programs to manage chronic conditions: Patients with chronic conditions such as arthritis and asthma are at higher risk of readmission. Hospitals can develop programs to manage these conditions more effectively, such as providing patient education on self-management strategies, offering physical therapy or other rehabilitation services, and coordinating care with other healthcare providers.
3. Improve discharge planning: The number of initial days in the hospital is a significant factor in predicting readmission. This suggests that hospitals need to improve their discharge planning processes to ensure that patients are adequately prepared to manage their conditions at home. This can include providing clear instructions on medications, follow-up appointments, and warning signs of potential complications.
4. Increase resources for high-risk patients: Patients with low complication risk are more likely to be readmitted so as think about higher risk patients more resources are needed. Hospitals can consider offering additional resources to high-risk patients, such as home health services, telemedicine visits, or additional follow-up care after discharge.

# **Part VI: Demonstration**

## G.  Provide a Panopto video recording that includes the presenter and a vocalized demonstration of the functionality of the code used for the analysis of the programming environment, including the following elements: **Video:** https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=0b424bb6-80ab-4b13-85ee-afd9016d84f9

•   an identification of the version of the programming environment

* iMac Apple M1
* IDE DataSpell
  + Python 3
  + Jupyter Notebook

•   a comparison of the initial logistic regression model you used and the reduced logistic regression model you used in your analysis  
  
The initial model had 16 predictor variables, while the reduced model has only 9 predictor variables.  
  
The reduced model removed 7 predictor variables that were deemed not statistically significant in predicting the response variable 'ReAdmis'  
  
The removed variables are 'Age', 'VitD\_levels', 'Overweight', 'BackPain', 'complication\_risk\_medium', 'gender\_male', and 'gender\_non\_binary'.  
  
The reduced model's pseudo R-squared value of 0.94 is similar to the initial model's pseudo R-squared value of 0.9406, which indicates that the reduced model still provides a good fit for the data.  
  
The reduced model's coefficients and standard errors for the remaining predictor variables are largely similar to those in the initial model, indicating that the remaining variables still have a strong association with the response variable 'ReAdmis'.  
  
 Regression Model MSEs:

Initial Regression Model: 0.017

Reduced Regression Model: 0.017

Accuracy calculation:

Accuracy Score of Initial Logistic Regression test datasets is: 0.98

Accuracy Score of Reduced Logistic Regression test datasets is: 0.98

•   an interpretation of the coefficients of the reduced model  
  
Reference section F1 second bullet

Note: The audiovisual recording should feature you visibly presenting the material (i.e., not in voiceover or embedded video) and should simultaneously capture both you and your multimedia presentation.

Note: For instructions on how to access and use Panopto, use the "Panopto How-To Videos" web link provided below. To access Panopto's website, navigate to the web link titled "Panopto Access," and then choose to log in using the “WGU” option. If prompted, log in using your WGU student portal credentials, and then it will forward you to Panopto’s website.

To submit your recording, upload it to the Panopto drop box titled “Regression Modeling – NBM3 | D208.” Once the recording has been uploaded and processed in Panopto's system, retrieve the URL of the recording from Panopto and copy and paste it into the Links option. Upload the remaining task requirements using the Attachments option.

## 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.

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

* Zach. (2020, October 13). *The 6 Assumptions of Logistic Regression (With Examples)*. Statology. <https://www.statology.org/assumptions-of-logistic-regression/>
* Centers for Medicare and Medicaid Services. (2023, February 23). Hospital readmissions reduction program (HRRP). Cms.gov. <https://www.cms.gov/Medicare/Medicare-Fee-for-Service-Payment/AcuteInpatientPPS/Readmissions-Reduction-Program>
* Luca Massaron. (2016). Regression Analysis with Python : learn the art of regression analysis with Python. Packt Publishing. C.

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