**D208 Performance Assessment**

**LOGISTIC REGRESSION FOR MEDICAL DATA**

Fahim A. Akbar Student ID 001434895 Masters Data Analytics (January 1, 2021) Program Mentor: Lea Yoakem (877) 435-7948x6422 [fakbar3@wgu.edu](mailto:fakbar3@wgu.edu)

**Part I: Research Question**

**A.  Purpose of using Logistic Regression for Data Analysis**

**1.  Question**

The proposed research question is if the probability of a patient’s readmission is affected by factors such as age, gender, or present medical conditions.

**2.  Goal**

The objective of this data analysis is to predict a patient’s chance of readmission. To do so, we will test to see which independent variables in the dataset are good indicator and fit. Once we decide on which variables to use, we will be doing a logistic regression analysis and predict the probability of patient readmissions. We will also provide a logistic equation along with the statistics for the identified variables.

**Part II: Method Justification**

**B.  Logistic Regression Methods**

**1.  Logistic regression model assumptions**

According to Statology, the following assumptions must be met for the logistic regression model to be a good fit for the present data: (Zach 2020)

* The dependent variable is binary
* Observations are independent. Residual plots against time can be used to check if there is a random pattern in the order of observations. If there isn’t, then the observations do not satisfy this assumption
* No multicollinearity should be present among the variables.
* No extreme outliers should be present.
* Explains the relationship between one dependent binary variable and one or more independent nominal variables
* A linear relationship should be present between the independent variables and logit of the dependent variable. Logit is defined as Logit(p) = log(p / (1-p)), where p is the probability of a positive outcome.
* The dataset offers a large sample size. Our data set has over 10,000 entries and is therefore sufficient.

**2.  Benefits of using Python**

In this assessment, Python will be used for the logistic regression model of the medical data. Python provides many different packages that can help with statistical analyses, such as Pandas, Scipy, Seaborn, and Statsmodels. Python also can create visualizations of variables and the observations within them. These visualizations are easy to understand and useful when present data. It allows you to analyze data easily into separate groups without modifying the original dataset. Python also logistic regression specific commands. These include LogisticRegression, model.fit, and model.coef\_. These commands will help with performing the regression analyses and visualizing them.

**3. Why logistic regression is an appropriate technique to analyze the research question**

For this assessment we are not trying to predict specific values. Instead, we are trying to predict the probability of the event on interest, readmission, occurring. To analyze the probability of patient readmission, Logistic regression is an appropriate technique. This is because the patient readmission is a binary predictor to a categorical variable (Readmission = Yes or No.) The patient observations are the independent variables of the data set that will be tested.

**Part III: Data Preparation**

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

**1. Relevant data preprocessing goal and data manipulations**

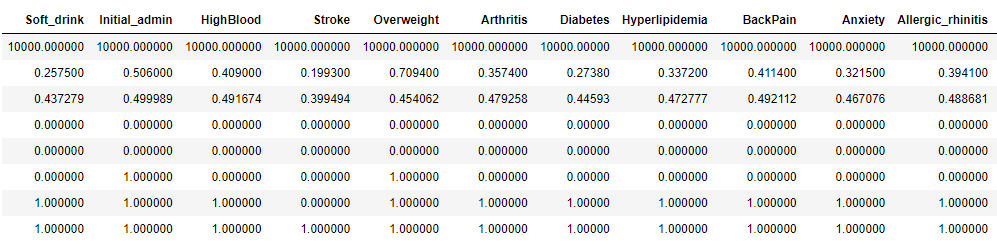
To prepare the data for our logistic regression analysis, we begin by first addressing null and missing values. We will either change the null values to zeros or populate them based on averages. Continuous variables will have histograms created to see if they have too many categories, and if we need combine categories. We will need to determine the ranges for other variables like HighBlood. Categorical variables will be converted into numerical values so that we can use them in our regression. Demographic data such as latitude and longitude will be removed since they do not pertain to our research question. If any duplicate data entries, rows, or columns are found, they will be removed. We will also be removing some of the categorical variables that have a high number of unique levels which cannot be condensed. In this case the variables that we will condense are Gender and Marital. We will be removing Complication Risks and Services since both of these variables have a high number of unique levels that cannot be condensed.

**2. Summary statistics**

P-values and coefficients for the independent variables will be identified to figure out which independent variables impact the probability of the dependent variable. For this analysis, the target dependent variable is ReAdmis. This variable represents whether a patient has been readmitted to the hospital. We believe that performing logistic regression will help predict which patients have a higher chance of being readmitted. This analysis could help the hospital plan to reduce readmissions, saving time and resources for the hospital and the patients. The predictor variables in this study are the patient observations, such as Children, Income, and Initial\_Admin. The tables below show us the standard deviations of each variable as well as the dispersion in the interquartile ranges. From here, we see that Children, Full\_meals\_eaten, Income, and vitD\_supp are not normally distributed variables, while Age, VitD\_Levels, Doc\_visits, and Initial\_days are normally distributed.

The summary statistics of all the variables in the cleaned data sheet are showcased below.





All of the categorical variables were converted to numerical variables so that we can conduct the logistic analysis and compare statistical data. The variable visualizations are shown in section 4. A summary of the statistics for these variables show that Age, BackPain, Reflux\_esophagitis, and Gender are normally distributed. Marital, Soft\_drink, Initial\_admin, Stroke, and Diabetes are not normally distribute and separated by large margins.

**3. Data Preparation Steps**

The first step in preparing the data is to make sure that there are no missing data entries in any of the columns. Next, we want to make sure there aren’t any duplicated data entries. We’ll also make sure that there aren’t any duplicated columns or rows to further prevent dealing with repeated entries. For the logistics analysis, several columns in the dataset were deemed irrelevant and were subsequently dropped from the dataset (i.e latitude, longitude). The “yes/no” entries for the categorical variables will be converted to 1 and 0, respectively. The marriage variable will also be condensed to fewer unique answers, “Married” and “Non-Married.” Gender will be condensed to “female” and “non-female” to reduce the levels. “Initial\_admin” will be condensed to "Emergency Admission/non-Emergency Admission" to reduce the levels. Services and complication risks will be dropped due to the high number of levels and upon further inspection, these levels could not be condensed to a binary set. Once all the necessary modifications to the data set are made, we will proceed with creating the initial logistics regression model.

**Code used for preparing data:**

# import packages that will be used for the logistics regression analysis

import pylab

import seaborn as sb

sb.set(style="white")

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

import sklearn

from sklearn.metrics import confusion\_matrix

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 matplotlib.pyplot as plt

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

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 pandas as pd

from pandas import Series, DataFrame

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

from imblearn.over\_sampling import SMOTE

# import data set that will be used for the logistics regression analysis

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

df = pd.read\_csv (r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\1medical\_clean.csv')

# rename the item columns accordingly

df.rename(columns={'Item1':'Timely\_admis','Item2':'Timely\_treat',

'Item3':'Timely\_visits','Item4':'Reliability',

'Item5':'Options','Item6':'Hrs\_treat',

'Item7':'Courteous','Item8':'Active\_listen'},inplace=True)

df.head()

df.info()

# drop all the demographic columns we don't need for this logistics regression analysis

df.drop(['City','State','County','Area','Zip','Lat','Lng','Population','TimeZone','Additional\_charges','TotalCharge','Interaction','UID','Customer\_id','Job','CaseOrder'],axis = 1,inplace=True)

# verify that all the columns were dropped before proceeding

df.info()

#check if there is any duplicate data entries present in columns

df[df.duplicated()]

# check if there are any duplicated columns in the data set - if there are none then the output should be False

df.columns.duplicated().any()

# check if there are any duplicated rows in the data set - if there are none then the output should be False

df.duplicated().any()

# convert categorical yes/no values to numeric 1/0 values

df = df.replace(to\_replace = ['Yes','No'],value = [1,0])

df

# convert the non-married Marital status values to "Married/Not Married", then convert "Married/Not Married" to "1/0"

#this will make the Marital variable easier to work with during regression analysis

df['Marital'] = df['Marital'].replace(['Divorced','Widowed','Separated','Never Married'],'Not Married')

df['Marital'] = df['Marital'].replace(['Married','Not Married'],[1,0])

df

# Showcase the unique values for the Services variable

df['Gender'].unique()

#convert the non-Female gender values to "Female/non-female", then convert "Female/non-female" to "1/0"

df['Gender'] = df['Gender'].replace(['Male','Nonbinary'],'non-female')

df['Gender'] = df['Gender'].replace(['Female','non-female'],[1,0])

df

# Showcase the unique values for the Services variable

df['Services'].unique()

# Drop the services variable since these values cannot be condensed

df.drop(['Services'],axis = 1,inplace=True)

# Showcase the unique values for the Complication\_risk variable

df['Complication\_risk'].unique()

# Drop the services variable since these values cannot be condensed

df.drop(['Complication\_risk'],axis = 1,inplace=True)

# Showcase the unique values for the Initial\_admin variable

df['Initial\_admin'].unique()

# convert the non-emergency admission status values to "Emergency Admission/non-Emergency Admission", then convert "Emergency Admission/non-Emergency Admission" to "1/0"

#this will make the Marital variable easier to work with during regression analysis

df['Initial\_admin'] = df['Initial\_admin'].replace(['Elective Admission','Observation Admission'],'non-Emergency Admission')

df['Initial\_admin'] = df['Initial\_admin'].replace(['Emergency Admission','non-Emergency Admission'],[1,0])

df

# describe the dataframe and showcase summary statistics of the variables

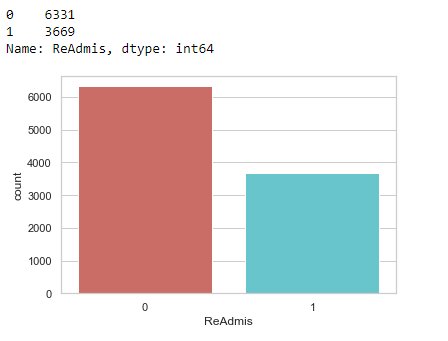
df.describe()

# now that all the modifications have been made, export the prepared dataset

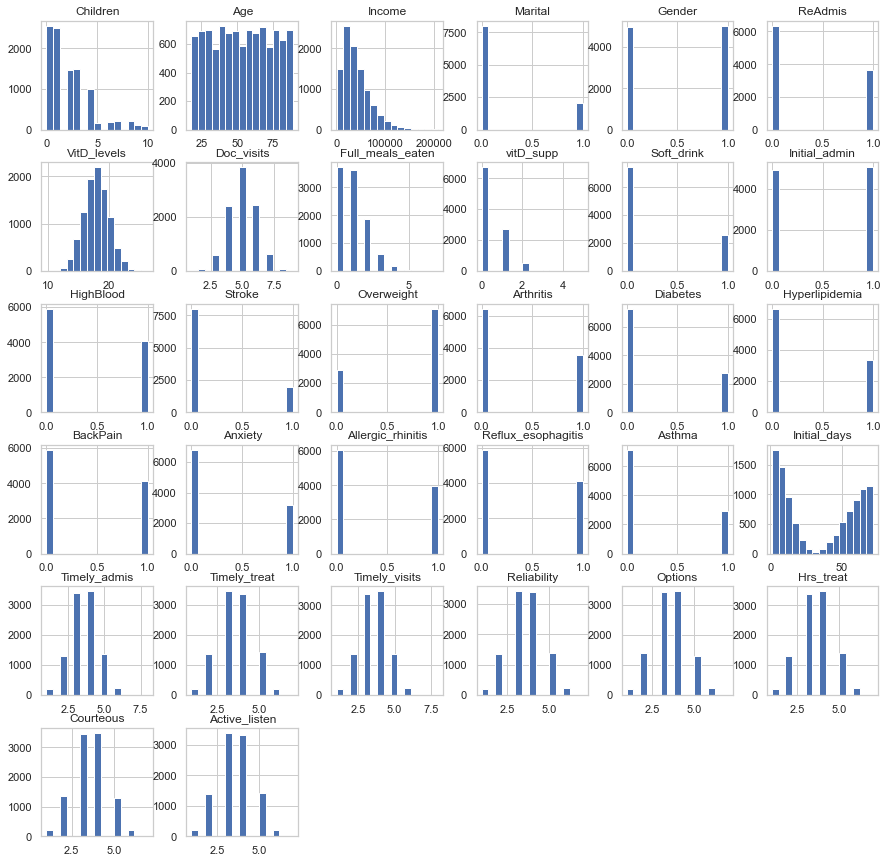
df.to\_csv(r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\2medical\_clean-PREPAREDTASK2\_12-24-2022.csv')

**4. Univariate and bivariate visualizations of the distributions of variables**

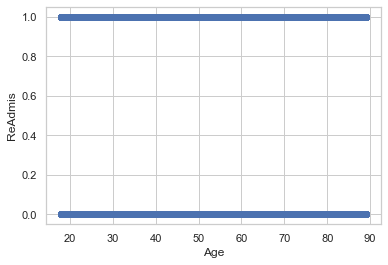
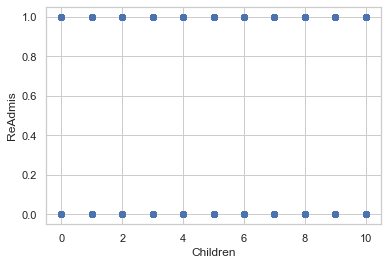
The cleaned data set was used to prepare visualizations of the variables we chose for the logistic regression. First we observed the target variable.

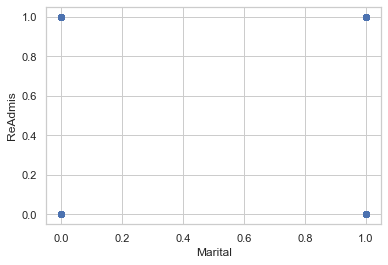
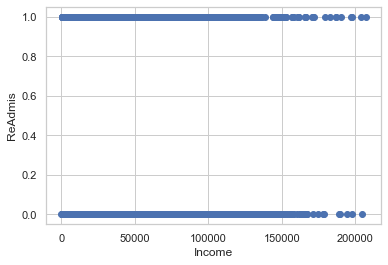


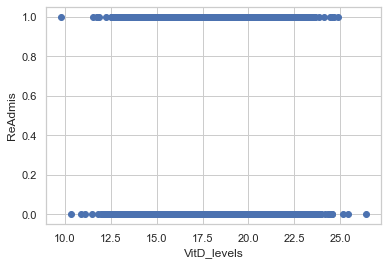
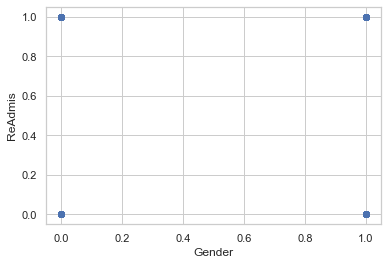
Here we see that the majority of patients were not readmitted. Next, we review the independent variables, using histograms.

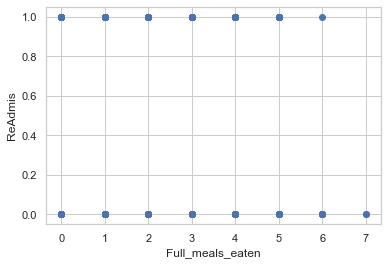
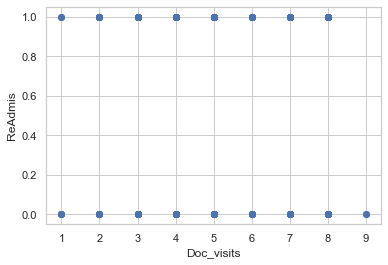


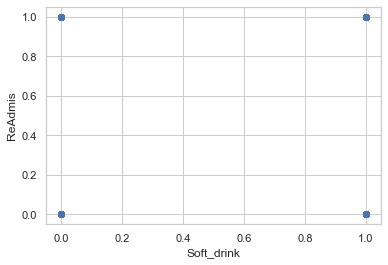
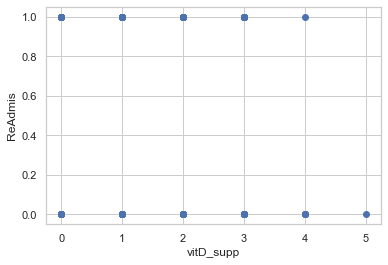
For bivariate visualizations, the target ReAdmis is compared with the potential predictors in the scatterplots shown below:

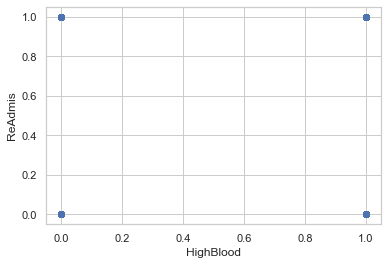
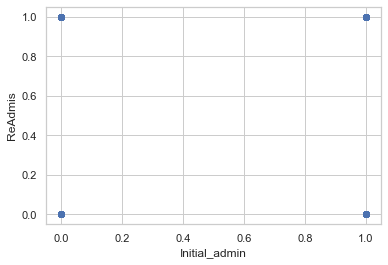


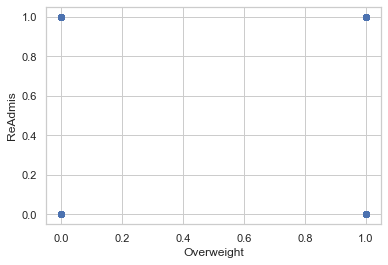
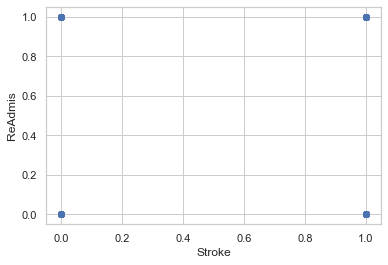


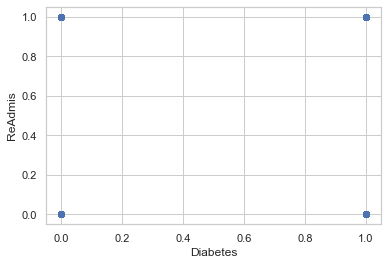
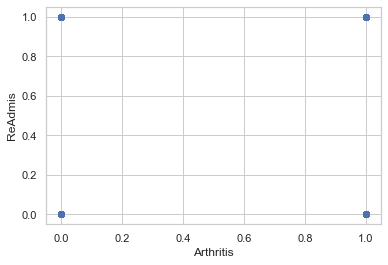


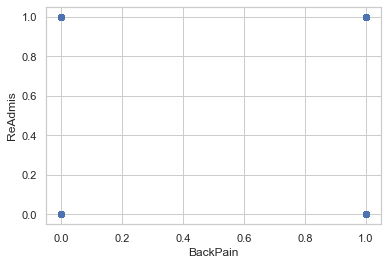
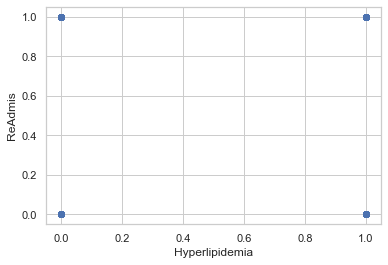


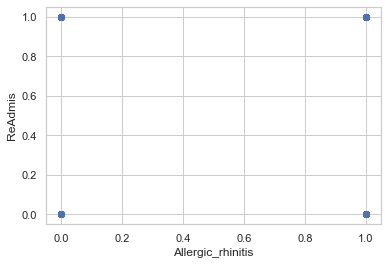
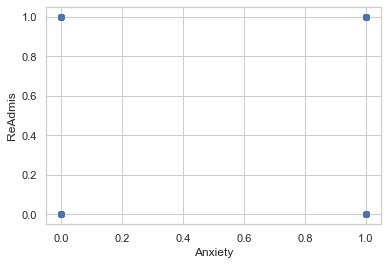


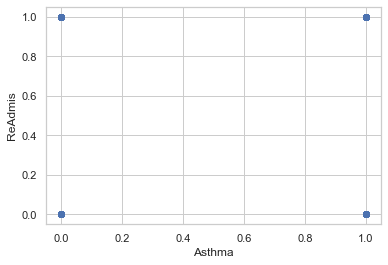
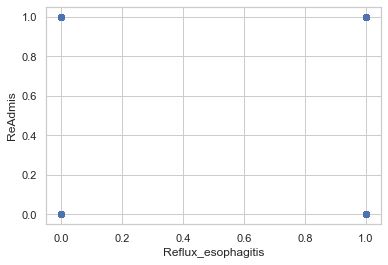


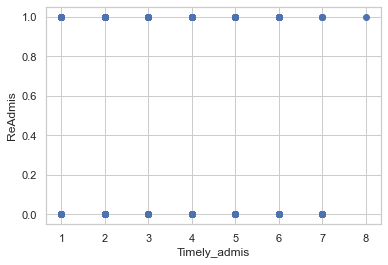
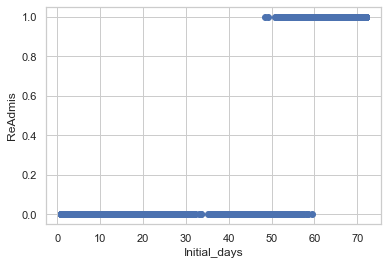


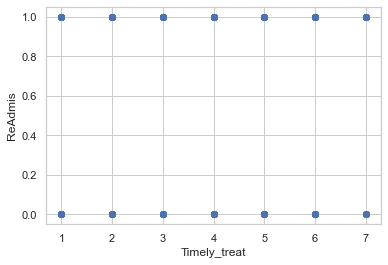


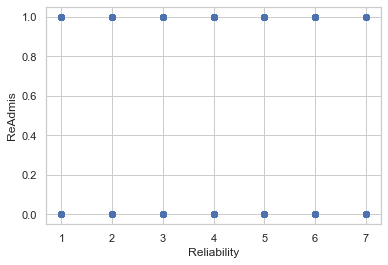
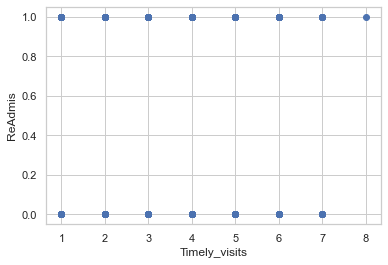


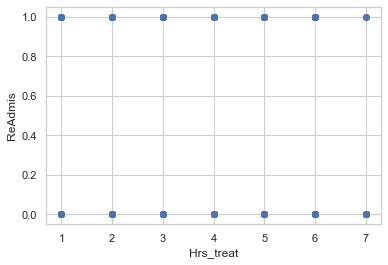
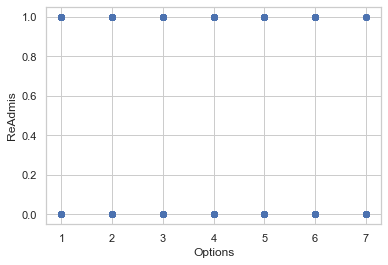


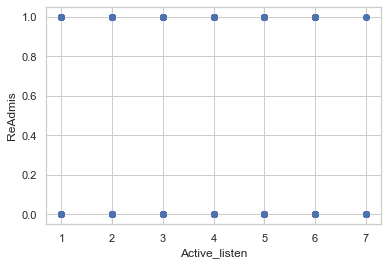
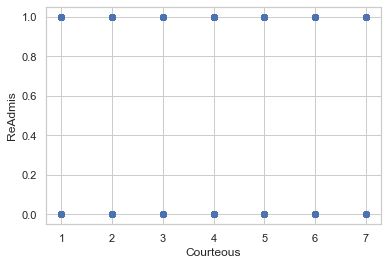












**5.  Provide a copy of the prepared data set.**

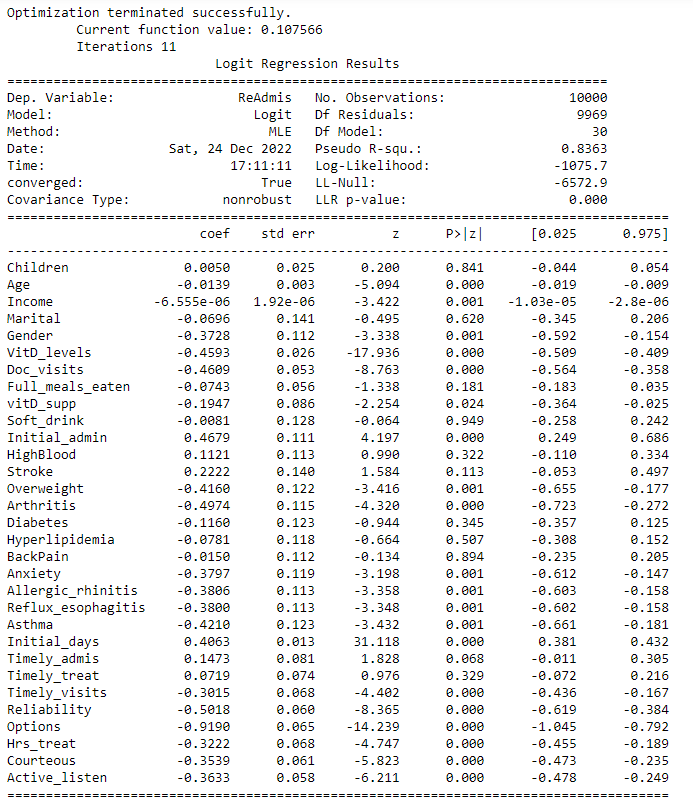
A copy of the cleaned data set, titled “2medical\_clean-PREPAREDTASK2\_12-24-2022.csv” is provided in the task submission.

**Part IV: Model Comparison and Analysis**

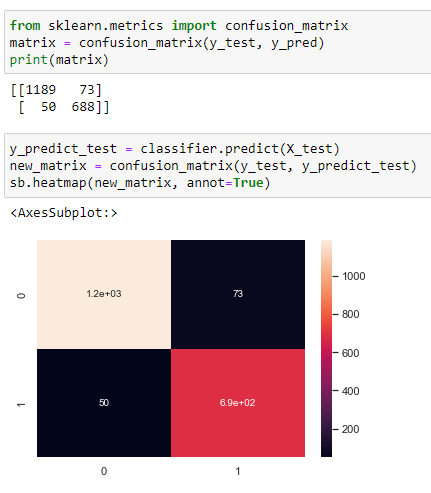
**D.  Comparison an initial and a reduced logistic regression**

**1. Initial logistic regression model**

Logit is used to run initial model. The output provided us with the P-value, coefficients, standard error values, and Z-value of the variables.

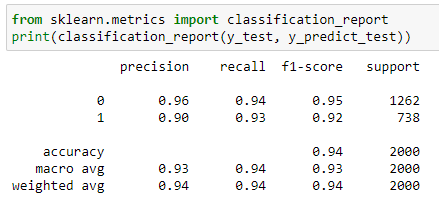
****

For us to better visualize the data, we also created the following confusion matrix:



**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**.

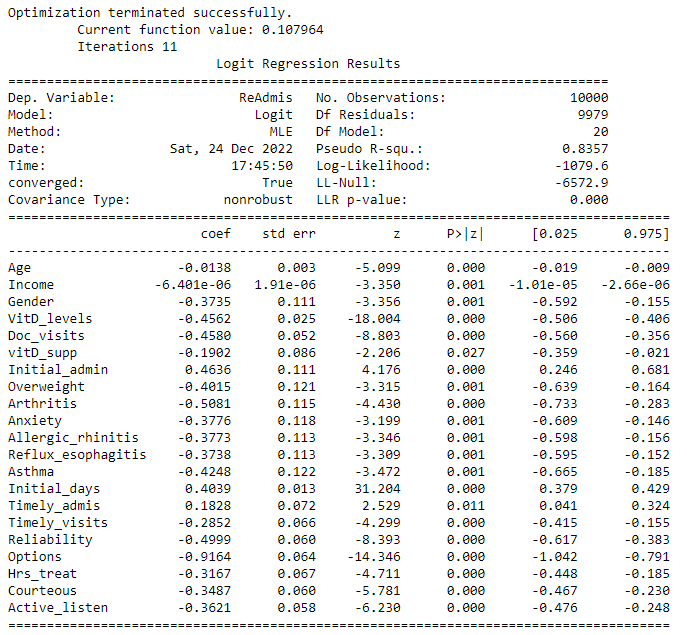
With the confusion matrix created, we generated a classification report to get the precision score of the initial model. This showed that the initial model predictions were 94% correct.



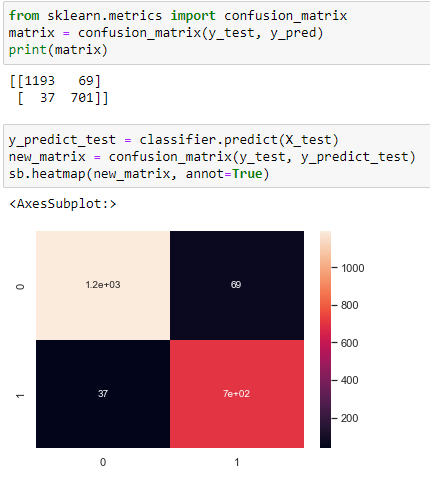
**Statistically based variable selection method:** To create our reduced model, the variables that had a P value below .05 statistical significance level were chosen. Variables with a P value above .05 statistical significance were dropped. The following variables were chosen for the reduced model: 'Age', 'Income', 'Gender', 'VitD\_levels', 'Doc\_visits', 'vitD\_supp','Initial\_admin', 'Overweight','Arthritis', 'Anxiety','Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Initial\_days','Timely\_admis', 'Timely\_visits', 'Reliability','Options', 'Hrs\_treat', 'Courteous', 'Active\_listen'.

**3.  Reduced logistic regression model.**

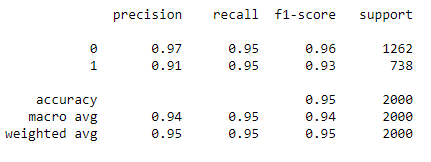
Now that we have our reduced model, we can proceed and run Logit again.



The new model provided us with this new confusion matrix and prediction percentage:

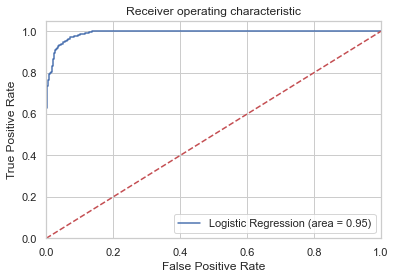


The classification report for the reduced model showed that the reduced model predictions were 95% correct, demonstrating a 1% increase in precision.



The Pseudo R squared value in the initial model was .8363 while the Pseudo R squared valuein the is .8357. This shows that the goodness of fit between the two models is almost the same. This also suggests that the variables we dropped may have statistically significant but only by a factor of .0006.

Using the reduced model and variables, we built a Receiver Operating Characteristic (ROC) graph. This linear slope is that of a random classifier and the blue curve is used to show classifiers true positive. If the blue arc is closer to the top left, then variables used have generated a greater confidence for the model. Because this is what we see with the graph produced, we can confirm our results with confidence.



**Code used to support the implementation of the logistic regression models.**

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

The full code for the project is provided at the end section **Part VI: Demonstration.** Furthermore, a pdf print of the Jupyter notebook used for running the python scripts is attached in task submission.

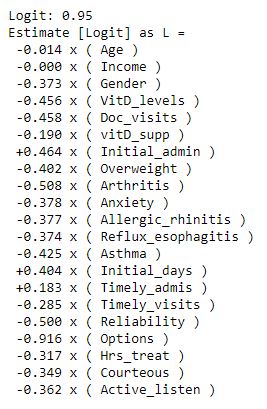
**Part V: Data Summary and Implications**

**F. Summary of findings and assumptions**

**Data analysis results:**

**Regression equation for the reduced model:**

After conducting the analysis, we created the following logistic regression equation for the reduced model:



**Interpretation of coefficients of the statistically significant variables of the model:**

A higher chance of readmission correlates with the positive coefficients, while a lower chance correlates with the negative coefficients. Based on our analysis, the following variables are linked to a patient’s higher chance of readmission: Initial\_admin, Initial\_days, and Timely\_admis. The remaining variables indicate a lower likelihood, but all variables used to create this model can potentially influence a patient’s chance of being readmitted.

**Statistical and Practical significance of the model:**

The final logistic regression model was both statistically and practically significant. We were able to determine which variables had the greatest impact on the likelihood of a patient being readmitted. We were also able to see the how when these variables increase or decrease, the chance of being readmitted also increases or decreases accordingly. The reduced model showed us that there is a strong correlation between a patient being readmittance and their initial administration, initial days spent at the hospital, and their timely admissions. The hospital can use our analysis to further investigate the correlation that a patient’s initial days, initial admission and timely admission has with their likelihood of being readmitted to the hospital.With the case of initial admissions, they could identify which of the readmitted patients were initial emergency admissions and which were not.These were strong predictor variables for a patient’s chance of being readmitted, so now the hospital could plan more efficient protocols involving these kinds of patients. They can also plan protocols based on the other strong predictor variables listed. By doing so, they could decrease the likelihood of patient readmission which will in turn help the hospital save time and resources. This can also benefit the patients because giving them the best treatment ahead of time based on their conditions will help to further reduce their chances of being readmitted. If any patient shows a positive result for any of the predictor variables, the hospital could plan ahead and anticipate their readmission in case preventive care or protocols aren’t enough. A patient who shows a combination of these variables would in turn have a higher chance of being readmitted. Knowing this could help the hospital further determine which patients are at the highest risk and they can plan treatments accordingly.

**Data analysis limitations:**

The first limitation we address is that the predictor variables are quite limited. They do not provide a diagnosis or solution to the higher readmission rates. Rather, they can be used to help predict if a patient has a high chance of readmission. These results are purely predictive and must be treated as such. Furthermore, the analysis is limited by the data set provided, as there could be additional variables that were not accounted for during the surveys. Our goal in this investigation was to see which variables provided could have correlated to a patient’s chance of readmission. Our investigation could be even more insightful if we had variables related to specific diseases and aliments that may have caused patients to return to the hospital, such as cancers. Additional variables related to patients could range from genetic history, recent exposure to certain viruses, additional health complications, and so on. The data provided is what the hospital deemed most important, so we are limited to work within that given framework. Lastly, as with any logistic regression, there is the potential of the model being overfit.

**Recommended course of action**

Improving the treatment process is the best thing to help decrease the likelihood of readmission. By making sure the patient receives the most accurate diagnosis and procedures as soon as possible, they in turn won’t have to spend too much time at the hospital getting treated. Reducing the time for their initial days spent at the hospital could also help to reduce their chances of being readmitted to the hospital. To address concerns of the model being overfit, we could run the regression multiple times or having additional analysts test for different variables could be beneficial to increase overall confidence. Lastly, we recommend a more detailed survey and adding more variables related to the specific reasons why patients were admitted into the hospital as well as what kind of treatment they’ll be receiving. For instance, patients who needed chemotherapy would need to be readmitted to the hospital since it’s a multi-appointment treatment. Knowing the relationship between why the patients are admitted to the hospital to begin with would prove valuable in further investigations and future models.

**Part VI: Demonstration**

**Link to the Panopto Video recording:**

[**https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=aba630d4-fffc-4db8-8660-af7600656d4a#**](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=aba630d4-fffc-4db8-8660-af7600656d4a)

**Sources for third party code:**

**Code for importing packages, preparing data, and performing logistic regression:**

<https://www.kaggle.com/code/prashant111/logistic-regression-classifier-tutorial/notebook>

<https://www.kaggle.com/code/miguelrodriguezolmos/plot-the-logistic-map-with-python-matplotlib/notebook>

<https://www.kaggle.com/code/vipulgandhi/linear-regression/notebook>

<https://www.kaggle.com/code/alexnystrom/hospital-data-logistic-regression>

<https://www.kaggle.com/code/neisha/heart-disease-prediction-using-logistic-regression>

<https://thepythonguru.com/python-string-formatting/>

<https://www.w3resource.com/pandas/dataframe/dataframe-itertuples.php>

<https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8>

<https://intellipaat.com/community/68715/converting-statsmodels-summary-object-to-pandas-dataframe>

**References**

Zach. (2020, October 13). The 6 Assumptions of Logistic Regression (With Examples). Statology. Statology.org/assumptions-of-logistic-regression/

Kohli, S. (2019) Understanding a classification report for your machine learning model, Medium. Medium. Available at: <https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397>.

**Full Code used for this Project**

~~Full Code used for D209 Task 2 Submission~~

~~DATA CLEANING AND PREPARATION CODE~~

# import packages that will be used for the logistics regression analysis

import pylab

import seaborn as sb

sb.set(style="white")

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

import sklearn

from sklearn.metrics import confusion\_matrix

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 matplotlib.pyplot as plt

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

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 pandas as pd

from pandas import Series, DataFrame

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

from imblearn.over\_sampling import SMOTE

# import data set that will be used for the logistics regression analysis

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

df = pd.read\_csv (r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\1medical\_clean.csv')

# rename the item columns accordingly

df.rename(columns={'Item1':'Timely\_admis','Item2':'Timely\_treat',

'Item3':'Timely\_visits','Item4':'Reliability',

'Item5':'Options','Item6':'Hrs\_treat',

'Item7':'Courteous','Item8':'Active\_listen'},inplace=True)

df.head()

df.info()

# drop all the demographic columns we don't need for this logistics regression analysis

df.drop(['City','State','County','Area','Zip','Lat','Lng','Population','TimeZone','Additional\_charges','TotalCharge','Interaction','UID','Customer\_id','Job','CaseOrder'],axis = 1,inplace=True)

# verify that all the columns were dropped before proceeding

df.info()

#check if there is any duplicate data entries present in columns

df[df.duplicated()]

# check if there are any duplicated columns in the data set - if there are none then the output should be False

df.columns.duplicated().any()

# check if there are any duplicated rows in the data set - if there are none then the output should be False

df.duplicated().any()

# convert categorical yes/no values to numeric 1/0 values

df = df.replace(to\_replace = ['Yes','No'],value = [1,0])

df

# convert the non-married Marital status values to "Married/Not Married", then convert "Married/Not Married" to "1/0"

#this will make the Marital variable easier to work with during regression analysis

df['Marital'] = df['Marital'].replace(['Divorced','Widowed','Separated','Never Married'],'Not Married')

df['Marital'] = df['Marital'].replace(['Married','Not Married'],[1,0])

df

# Showcase the unique values for the Services variable

df['Gender'].unique()

#convert the non-Female gender values to "Female/non-female", then convert "Female/non-female" to "1/0"

df['Gender'] = df['Gender'].replace(['Male','Nonbinary'],'non-female')

df['Gender'] = df['Gender'].replace(['Female','non-female'],[1,0])

df

# Showcase the unique values for the Services variable

df['Services'].unique()

# Drop the services variable since these values cannot be condensed

df.drop(['Services'],axis = 1,inplace=True)

# Showcase the unique values for the Complication\_risk variable

df['Complication\_risk'].unique()

# Drop the services variable since these values cannot be condensed

df.drop(['Complication\_risk'],axis = 1,inplace=True)

# Showcase the unique values for the Initial\_admin variable

df['Initial\_admin'].unique()

# convert the non-emergency admission status values to "Emergency Admission/non-Emergency Admission", then convert "Emergency Admission/non-Emergency Admission" to "1/0"

#this will make the Marital variable easier to work with during regression analysis

df['Initial\_admin'] = df['Initial\_admin'].replace(['Elective Admission','Observation Admission'],'non-Emergency Admission')

df['Initial\_admin'] = df['Initial\_admin'].replace(['Emergency Admission','non-Emergency Admission'],[1,0])

df

# describe the dataframe and showcase summary statistics of the variables

df.describe()

# now that all the modifications have been made, export the prepared dataset

df.to\_csv(r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\2medical\_clean-PREPAREDTASK2\_12-24-2022.csv')

~~CODE USED FOR CREATING UNIVARIATE AND BIVARIATE VISUALIZATIONS~~

# Start to visualize the data, including univariate and bivariate analyses

# begin by visualizing the target variable, ReAdmis

print(df['ReAdmis'].value\_counts())

sb.countplot(x='ReAdmis', data=df, palette='hls')

plt.show()

# identify the columns for variables

Variables = df.select\_dtypes(include = "number").columns

print (Variables)

# create histogram plots of the identified predictor variables

fig = plt.figure(figsize=(15, 15))

ax = df[Variables].hist(bins = 15, figsize=(15,15))

plt.title('Variables')

fig.tight\_layout(h\_pad=5, w\_pad=5)

plt.show()

#selecting the target varialbe and showcasing the bivariate statistics

X=df.drop('ReAdmis',inplace=False, axis=1)

y=df['ReAdmis']

for column in X.columns:

plt.scatter(X[column],y)

plt.xlabel(column)

plt.ylabel('ReAdmis')

plt.show()

~~CODE USED FOR THE INITIAL MODEL AND ANALYSIS OF THE INITIAL MODEL~~

# create the initial logistics model

log\_reg\_results = sm.Logit(df["ReAdmis"], df[['Children', 'Age', 'Income', 'Marital', 'Gender', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'vitD\_supp','Soft\_drink', 'Initial\_admin', 'HighBlood', 'Stroke', 'Overweight','Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain', 'Anxiety','Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Initial\_days','Timely\_admis', 'Timely\_treat', 'Timely\_visits', 'Reliability','Options', 'Hrs\_treat', 'Courteous', 'Active\_listen']]).fit()

print(log\_reg\_results.summary())

# create the correlation matrix

matrix\_df = pd.read\_csv(r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\2medical\_clean-PREPAREDTASK2\_12-24-2022.csv')

matrix\_df = matrix\_df[['Children', 'Age', 'Income', 'Marital', 'Gender', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'vitD\_supp','Soft\_drink', 'Initial\_admin', 'HighBlood', 'Stroke', 'Overweight','Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain', 'Anxiety','Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Initial\_days','Timely\_admis', 'Timely\_treat', 'Timely\_visits', 'Reliability','Options', 'Hrs\_treat', 'Courteous', 'Active\_listen','ReAdmis']]

X = matrix\_df.iloc[:, 1:-1].values

y = matrix\_df.iloc[:,-1].values

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state = 0)

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

y\_pred = classifier.predict(X\_test)

#now create the confusion matrix for the initial model

from sklearn.metrics import confusion\_matrix

matrix = confusion\_matrix(y\_test, y\_pred)

print(matrix)

#retrieve the classificaiton report for the initial model

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_predict\_test))

~~CODE USED FOR THE REDUCED MODEL AND ANALYSIS OF THE REDUCED MODEL~~

#Create the reduced model with the variables that had a P value below .05 statistical significance level

log\_reg\_results2 = sm.Logit(df["ReAdmis"], df[['Age', 'Income', 'Gender', 'VitD\_levels', 'Doc\_visits', 'vitD\_supp','Initial\_admin', 'Overweight','Arthritis', 'Anxiety','Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Initial\_days','Timely\_admis', 'Timely\_visits', 'Reliability','Options', 'Hrs\_treat', 'Courteous', 'Active\_listen']]).fit()

print(log\_reg\_results2.summary())

# create the correlation matrix for the reduced model

matrix\_df = pd.read\_csv(r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\2medical\_clean-PREPAREDTASK2\_12-24-2022.csv')

matrix\_df = matrix\_df[['Age', 'Income', 'Gender', 'VitD\_levels', 'Doc\_visits', 'vitD\_supp','Initial\_admin', 'Overweight','Arthritis', 'Anxiety','Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Initial\_days','Timely\_admis', 'Timely\_visits', 'Reliability','Options', 'Hrs\_treat', 'Courteous', 'Active\_listen','ReAdmis']]

X = matrix\_df.iloc[:, 1:-1].values

y = matrix\_df.iloc[:,-1].values

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state = 0)

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

y\_pred = classifier.predict(X\_test)

#now create the confusion matrix for the reduced model

from sklearn.metrics import confusion\_matrix

matrix = confusion\_matrix(y\_test, y\_pred)

print(matrix)

y\_predict\_test = classifier.predict(X\_test)

new\_matrix = confusion\_matrix(y\_test, y\_predict\_test)

sb.heatmap(new\_matrix, annot=True)

#retrieve the classificaiton report for the reduced model

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_predict\_test))

~~CODE USED FOR THE LOGISTICS REGRESSION EQUATION~~

# plot ROC Curve

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

fpr, tpr, thresholds = roc\_curve(y\_test, classifier.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()

# create an equation of the regression

print('Logit: {:.2f}'.format(logit\_roc\_auc))

equation = log\_reg\_results2.summary().tables[1]

print('Estimate [{}] as L = '.format(log\_reg\_results2.summary().tables[0][1][1]))

equation = pd.DataFrame(equation)

for i in equation.itertuples():

print(' {:+.3f} x ( {} ) '.format(i[0],i[1]))

print(equation)

updated\_equation = equation.drop(0)

print(updated\_equation)

# create an equation of the logistics regression

print('Logit: {:.2f}'.format(logit\_roc\_auc))

print('Estimate [{}] as L = '.format(log\_reg\_results2.summary().tables[0][1][1]))

for i in updated\_equation.itertuples():

print(' {:+.3f} x ( {} ) '.format(float(str(i[2])),i[1]))