Predictive Modeling

Jason Willis

College of Information Technology,

Western Governors University

Dr. Eric Straw

January 9th, 2021

Table of Contents for Each Rubric

Hospital Readmission Problem

For our chain of hospitals to lower readmission concerns, we need to identify patients who have increased risk of rehospitalization within a month of their release. According to Schuller (2020), non-obese adults were 21% less likely to be readmitted than obese adults. A readmission study by Gert, et. al. (2002) showed a correlation between longer initial hospital stays and readmission. Within the provided dataset, I'm leveraging these studies to help create my hypothetical question and shape my approach in finding potential patient groups with a statistically significant chance for readmission outcomes.

After viewing the provided medical_clean.csv data set and accompanying data dictionary, there seems to be some patient groupings which are aligned with the research mentioned above. For instance, the following patient data fields: Initial_days, Initial_admin and Diabetes seem like they should correlate with a patient's readmission. While my initial feelings towards these variables might make them feel related, are they?

A1 - Research Question

Do the following three predictors: Initial Days, Initial Admin, and Diabetes have an influence on the probability of a patient's readmission?

A2 - Objectives and Goals

The goal of our analysis is to logically investigate given data to support or reject the hypothesis. Some data will need to be converted from categorical to numerical data types in order to analyze and process. Our objective is to see how, if at all, a patient's initial inpatient

days, the reason for their initial admission or if they have diabetes correlate with their readmission within 30 days.

B1 – Summary of Assumptions

Applying logistic regression to our model will be based on some assumptions. This regression's dependent variable is binary and therefore based on the Bernoulli distribution. Predicted values are restricted to a range of nominal (categorical) values like yes, no, pass, fail, win, lose, small, medium, large. Predictions will be a probability of a particular outcome rather than the outcome itself. Predictors will not have a high correlation (multicollinearity). The logistic regression approach is the logarithm of odds in achieving 1. Our regression model will help understand the probability of patient readmission (the dependent variable) based on the following independent predictor variables:

- Initial Days (as inpatient)
- Initial Reason for Admission (emergency, elective or observational)
- If the patient has diabetes

B2 – Tool Benefits

Python and the chosen libraries (pandas, numpy, seaborn, matplotlib, statsmodels, sklearn) were all chosen to help structure, clean, transform, manipulate, analyze, and visualize the data set quickly, accurately, and efficiently.

B3 – Appropriate Technique

Logistic regression was chosen to predict a binary response, e.g. the probability of patient readmission. An equation of log $(\frac{p(y=1)}{1-(p=1)})$ will be utilized. Understanding chosen predictor variable relationships could help our hospital chain better predict potential patient care and readmissions.

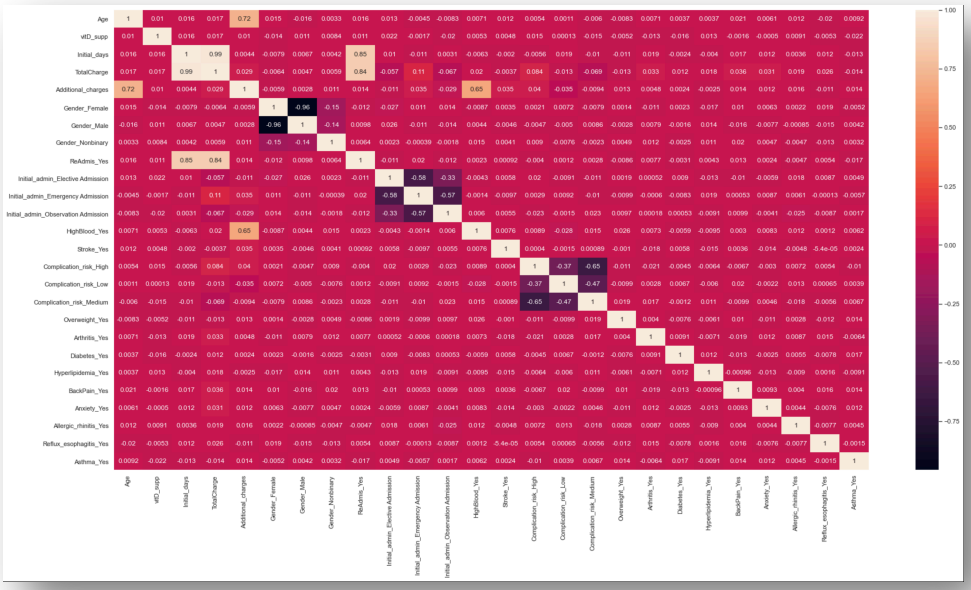


Figure 1 – Heat map of Data Field Correlation Tests (X)

As you can see above in the Figure 1 heatmap, the current set of data fields needs to be reduced. By viewing the display of significant correlation with other data fields, comparing to the hypothesis, and later viewing the series p-value, the data frame will by statistically reduced.

C1 – Data Goals

From the original data set of 10,000 rows and 50 columns, attention was focused on discarding fields which did not address and/or correlate with the hypothesis. To understand the

data, df.info() was run which provided range, shape, completeness (all columns contained Non-Null data) and field data types. After verifying data had no nulls (Figure 2), df.describe() was used to provide a descriptive statistical summary of the field's central tendency and dispersion. I split the data frame into numeric and categorical subsets (df_num and df_cat respectively), and used pandas describe() method to help me get a feel of the data.

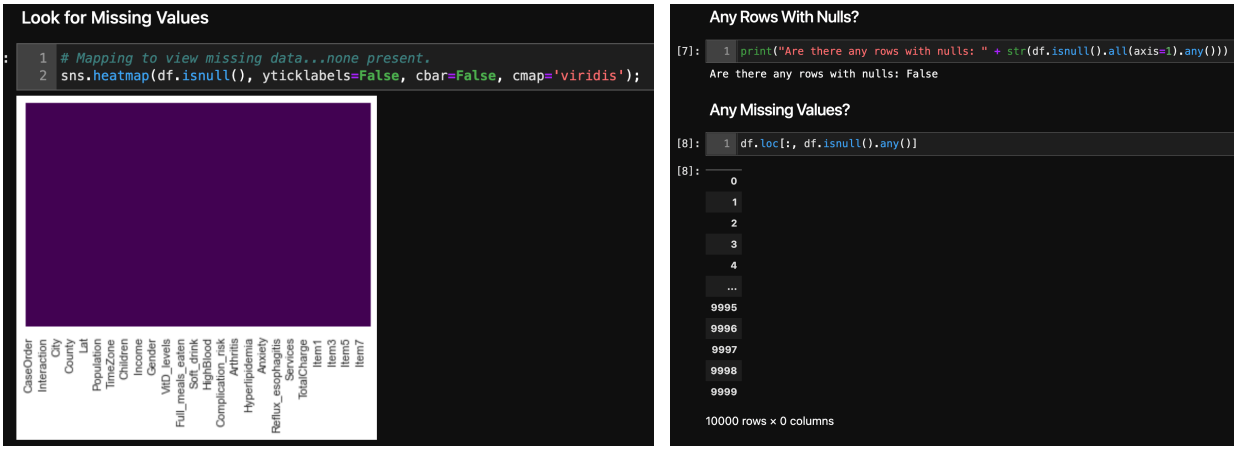


Figure 2 - Looking for Nulls

Next, I started to drop what I felt was obviously not helpful data and unrelated to the hypotheses. Next, pd.get_dummies() converted predictive categorical data to separate fields based on their category. For example, Diabetes became Diabetes_No and Diabetes_Yes, containing ones and zeros respectively. Then, for all binary categories, I dropped the '_No' columns to remove chance of multicollinearity (Figure 3).

Prune Numerical Fields

Add Columns to Quantify Boolean Fields

```
pruned_df_num = df_num.drop(['CaseOrder', 'Population', 'Children', 'Income', 'VitD_levels', 'Doc_visits',
# Transform & Add Quantified Data Fields As Needed:
pruned_df_num['Overweight_Num'] = df['Overweight'].eq('Yes').astype(int)
pruned_df_num['Diabetes_Num'] = df['Diabetes'].eq('Yes').astype(int)
pruned_df_num['ReAdmis_Num'] = df['ReAdmis'].eq('Yes').astype(int)
pruned_df_num['Gender_Num'] = df['Gender'].eq('Male').astype(int)

pruned_df_num
```

	Age	Initial_days	TotalCharge	Additional_charges	Overweight_Num	Diabetes_Num	ReAdmis_Num	Gender_Num
0	53	10.585770	3726.702860	17939.403420	0	1	0	1
1	51	15.129562	4193.190458	17612.998120	1	0	0	0
2	53	4.772177	2434.234222	17505.192460	1	1	0	0
3	78	1.714879	2127.830423	12993.437350	0	0	0	1
4	22	1.254807	2113.073274	3716.525786	0	0	0	0
...	...	...	...	...	...	...	...	...
9995	25	51.561220	6850.942000	8927.642000	0	0	0	1
9996	87	68.668240	7741.690000	28507.150000	1	1	1	1
9997	45	70.154180	8276.481000	15281.210000	1	0	1	0
9998	43	63.356900	7644.483000	7781.678000	1	0	1	1
9999	70	70.850590	7887.553000	11643.190000	1	0	1	0

10000 rows × 8 columns

Figure 3 - Pruning Data, Convert Categorical Fields to Numerical, Address Multicollinearity

```
df.describe()
```

	CaseOrder	Zip	Lat	Lng	Population	Children	Age	Income	VitD_levels	Doc_visits	...
count	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	...
mean	5000.50000	50159.323900	38.751099	-91.243080	9965.253800	2.097200	53.511700	40490.495160	17.964262	5.012200	...
std	2886.89568	27469.588208	5.403085	15.205998	14824.758614	2.163659	20.638538	28521.153293	2.017231	1.045734	...
min	1.00000	610.000000	17.967190	-174.209700	0.000000	0.000000	18.000000	154.080000	9.806483	1.000000	...
25%	2500.75000	27592.000000	35.255120	-97.352982	694.750000	0.000000	36.000000	19598.775000	16.626439	4.000000	...
50%	5000.50000	50207.000000	39.419355	-88.397230	2769.000000	1.000000	53.000000	33768.420000	17.951122	5.000000	...
75%	7500.25000	72411.750000	42.044175	-80.438050	13945.000000	3.000000	71.000000	54296.402500	19.347963	6.000000	...
max	10000.00000	99929.000000	70.560990	-65.290170	122814.000000	10.000000	89.000000	207249.100000	26.394449	9.000000	...

8 rows × 23 columns

Figure 4 - Summary Statistics for the Data Frame

C2 – Summary Statistics

As stated in the Data Goals, to answer the selected research question, the target variable and all predictor variables remained in the data frame to analyze. The pandas describe() method was used to provide summary statistics (Figure 4). From the describe() method, we can see

10,000 observations across all fields. For instance, when viewing the following target and predictor variable columns:

- Initial Days – 10,000 observations, a mean of 34.455299, a standard deviation of 26.309341, minimum value of 1.001981, 25% Percentile: 7.896215, 50% Percentile: 35.836244, 75% Percentile: 61.161020, and maximum value: 71.981490
- Diabetes – Count of 10,000 observations, 2 Unique elements (Yes/No), Top = 'No', Frequency of 7,262 No's
- Initial Admin – Count of 10,000 observations, 3 Unique elements (Emergency, Elective and Observation), Top = 'Emergency', Frequency of 5060

C3 – Steps to Prepare Data

In addition to some cleaning, wrangling, removing and transformation of data (Figure 2 and 3), I wanted to view each data series p-values (Figure 5).

```
Optimization terminated successfully.
         Current function value: 0.042612
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:            ReAdmis_Yes   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9984
Method:                           MLE   Df Model:                           15
Date:                Fri, 14 Jan 2022   Pseudo R-squ.:                 0.9352
Time:                        19:36:19   Log-Likelihood:                -426.12
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
const                    -62.5631      3.112    -20.103      0.000     -68.663     -56.463
Age                   -8.124e-05      0.004     -0.019      0.985      -0.008       0.008
vitD_supp                -0.0885      0.137     -0.647      0.518      -0.356       0.179
Initial_days              1.1569      0.057     20.150      0.000       1.044       1.269
Gender_Female            -0.1312      0.177     -0.741      0.458      -0.478       0.216
HighBlood_Yes             0.7724      0.186      4.146      0.000       0.407       1.137
Stroke_Yes                1.3843      0.232      5.965      0.000       0.929       1.839
Overweight_Yes           -0.1424      0.195     -0.730      0.465      -0.525       0.240
Arthritis_Yes            -0.8773      0.191     -4.598      0.000      -1.251      -0.503
Diabetes_Yes              0.4177      0.198      2.109      0.035       0.029       0.806
Hyperlipidemia_Yes        0.3820      0.189      2.022      0.043       0.012       0.752
BackPain_Yes              0.3203      0.180      1.781      0.075      -0.032       0.673
Anxiety_Yes              -0.7763      0.193     -4.024      0.000      -1.154      -0.398
Allergic_rhinitis_Yes    -0.2336      0.181     -1.293      0.196      -0.588       0.121
Reflux_esophagitis_Yes   -0.2059      0.182     -1.133      0.257      -0.562       0.150
Asthma_Yes               -1.0159      0.198     -5.135      0.000      -1.404      -0.628
============================================================================================

Possibly complete quasi-separation: A fraction 0.77 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

Figure 5 - Set up and Run the Logistic Regression Model, Compare P-Values to 0.05 Threshold

By comparing the p-value of each series to a standard cutoff threshold of 0.05, I removed the data fields that did not fit the model (Figure 5).

```
logit = sm.Logit(y.astype(float), X.astype(float))
result = logit.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.042633
         Iterations 13
                          Logit Regression Results
==============================================================================
Dep. Variable:            ReAdmis_Yes   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9986
Method:                           MLE   Df Model:                          13
Date:                Fri, 14 Jan 2022   Pseudo R-squ.:                0.9351
Time:                        20:17:27   Log-Likelihood:                -426.33
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
const                      -62.5942      3.110    -20.125      0.000     -68.690     -56.498
Initial_days                 1.1567      0.057     20.193      0.000       1.044       1.269
Gender_Female               -0.1261      0.177     -0.715      0.475      -0.472       0.220
HighBlood_Yes                0.7761      0.186      4.168      0.000       0.411       1.141
Stroke_Yes                   1.3774      0.231      5.953      0.000       0.924       1.831
Overweight_Yes              -0.1381      0.195     -0.709      0.478      -0.520       0.244
Arthritis_Yes               -0.8801      0.191     -4.619      0.000      -1.254      -0.507
Diabetes_Yes                 0.4191      0.198      2.121      0.034       0.032       0.806
Hyperlipidemia_Yes           0.3769      0.189      1.998      0.046       0.007       0.747
BackPain_Yes                 0.3214      0.180      1.789      0.074      -0.031       0.674
Anxiety_Yes                 -0.7702      0.192     -4.006      0.000      -1.147      -0.393
Allergic_rhinitis_Yes       -0.2348      0.181     -1.299      0.194      -0.589       0.119
Reflux_esophagitis_Yes      -0.2072      0.181     -1.147      0.251      -0.561       0.147
Asthma_Yes                  -1.0157      0.198     -5.133      0.000      -1.403      -0.628
============================================================================================
```

Figure 6 - Set up and Run the Logistic Regression Model, Only With Data Meeting the P-Values to 0.05 Threshold

Next, the pruned data set was ready to be set for the next phase of training and testing. A variable ‘X’ was set to the remaining predictor variables meeting the p-value threshold and a variable ‘y’ was set to the target variable. A constant variable was also added to the set of ‘X’ predictor variables, which creates a column with the value of 1.0. Then the logistic regression model was run (Figure 6) and split into a training size of 70% (7,000 rows) and a test size of 30% (3,000 rows).

```
1 print(result_test.summary2())
```

```
                           Results: Logit
================================================================================
Model:                    Logit                Pseudo R-squared:     0.934
Dependent Variable:       ReAdmis_Yes          AIC:                  286.0641
Date:                     2022-01-15 11:57     BIC:                  370.1532
No. Observations:         3000                 Log-Likelihood:       -129.03
Df Model:                 13                   LL-Null:              -1965.9
Df Residuals:             2986                 LLR p-value:          0.0000
Converged:                1.0000               Scale:                1.0000
No. Iterations:           13.0000
--------------------------------------------------------------------------------
                         Coef.   Std.Err.     z     P>|z|    [0.025   0.975]
--------------------------------------------------------------------------------
const                   -65.0820   5.9258 -10.9829 0.0000 -76.6963 -53.4677
Initial_days              1.2069   0.1096  11.0141 0.0000   0.9921   1.4217
Gender_Female             0.2888   0.3340   0.8649 0.3871  -0.3657   0.9433
HighBlood_Yes             0.7958   0.3472   2.2922 0.0219   0.1153   1.4762
Stroke_Yes                1.8764   0.4130   4.5436 0.0000   1.0670   2.6859
Overweight_Yes           -0.6074   0.3578  -1.6977 0.0896  -1.3086   0.0938
Arthritis_Yes            -0.7105   0.3551  -2.0009 0.0454  -1.4065  -0.0145
Diabetes_Yes              0.1118   0.3731   0.2997 0.7644  -0.6195   0.8432
Hyperlipidemia_Yes        0.2454   0.3507   0.6998 0.4840  -0.4419   0.9327
BackPain_Yes              0.3630   0.3335   1.0884 0.2764  -0.2907   1.0167
Anxiety_Yes              -0.9067   0.3550  -2.5540 0.0106  -1.6026  -0.2109
Allergic_rhinitis_Yes    -0.1182   0.3318  -0.3562 0.7217  -0.7685   0.5321
Reflux_esophagitis_Yes   -0.6672   0.3452  -1.9329 0.0533  -1.3438   0.0094
Asthma_Yes               -1.1635   0.3548  -3.2793 0.0010  -1.8589  -0.4681
================================================================================
```

Figure 7 - Pruned Test Data Results

Once run, test results were compared to p-values to a threshold of 0.05 cutoff. As you can see in Figure 7, not all test results met the cutoff as in the previous logistic regression model.

```
logit_hyp = sm.Logit(y_hyp.astype(float), X_hyp.astype(float))
result_hyp = logit_hyp.fit()
print(result_hyp.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.044923
         Iterations 13
                                  Results: Logit
================================================================================================
Model:                          Logit                       Pseudo R-squared:            0.932
Dependent Variable:             ReAdmis_Yes                 AIC:                         908.4613
Date:                           2022-01-15 12:21            BIC:                         944.5130
No. Observations:               10000                       Log-Likelihood:              -449.23
Df Model:                       4                           LL-Null:                     -6572.9
Df Residuals:                   9995                        LLR p-value:                 0.0000
Converged:                      1.0000                      Scale:                       1.0000
No. Iterations:                 13.0000
------------------------------------------------------------------------------------------------
                                   Coef.     Std.Err.      z    P>|z|      [0.025         0.975]
------------------------------------------------------------------------------------------------
const                            -43.6637 7158278.8267 -0.0000 1.0000 -14030012.3553 14029925.0278
Initial_days                       1.0665       0.0506 21.0624 0.0000         0.9673        1.1657
Initial_admin_Elective Admission -15.3276 7158278.8267 -0.0000 1.0000 -14029984.0192 14029953.3640
Initial_admin_Emergency Admission -13.5636 7158278.8267 -0.0000 1.0000 -14029982.2552 14029955.1280
Initial_admin_Observation Admission -14.7725 7158278.8267 -0.0000 1.0000 -14029983.4641 14029953.9190
Diabetes_Yes                       0.1728       0.1883  0.9174 0.3589        -0.1963        0.5418
================================================================================================
```

```
logit_hyp_test = sm.Logit(y_hyp_test.astype(float), X_hyp_test)  #[['const','Initial_days', 'Diabet
result_hyp_test = logit_hyp_test.fit()
print(result_hyp_test.summary2())

p_vals = dict(logit_hyp_test.fit().pvalues[1:])
```

```
Optimization terminated successfully.
         Current function value: 0.046360
         Iterations 13
                            Results: Logit
================================================================================
Model:                      Logit                  Pseudo R-squared:       0.929
Dependent Variable:         ReAdmis_Yes            AIC:                    288.1574
Date:                       2022-01-15 12:21       BIC:                    318.1892
No. Observations:           3000                   Log-Likelihood:         -139.08
Df Model:                   4                      LL-Null:                -1965.9
Df Residuals:               2995                   LLR p-value:            0.0000
Converged:                  1.0000                 Scale:                  1.0000
No. Iterations:             13.0000
--------------------------------------------------------------------------------
                                  Coef.   Std.Err.    z    P>|z|   [0.025 0.975]
--------------------------------------------------------------------------------
const                           -45.6564      nan     nan    nan     nan    nan
Initial_days                      1.1155   0.0975 11.4396 0.0000  0.9244 1.3066
Initial_admin_Elective Admission -16.0234      nan     nan    nan     nan    nan
Initial_admin_Emergency Admission -13.9644      nan     nan    nan     nan    nan
Initial_admin_Observation Admission -15.6687      nan     nan    nan     nan    nan
Diabetes_Yes                     -0.0272   0.3399 -0.0801 0.9362 -0.6934 0.6389
================================================================================
```

Figure 8 - Logistic Regression Run with Hypothesis Variables and Compared to P-Value Threshold

C4 – Visualizations

The following univariate (Table 1) and bivariate (Table 2) visualizations are from the cleaned and reduced data set. The bivariate visualization includes the target variable.

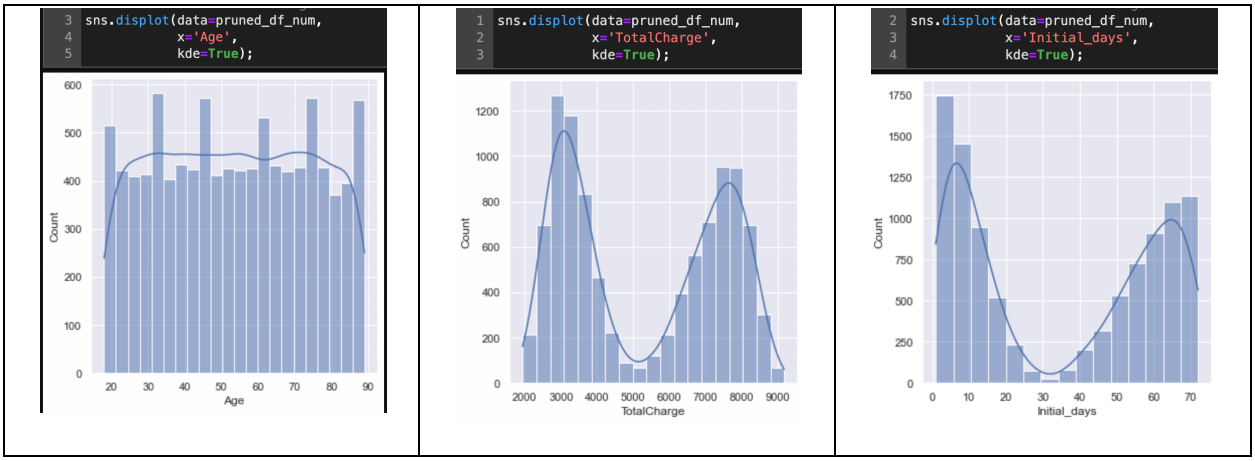
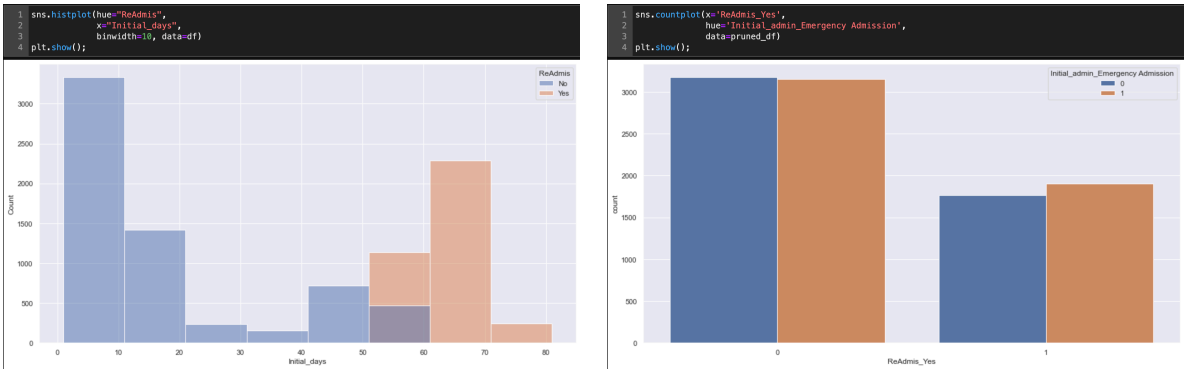


Table 1 - Univariate Visualizations



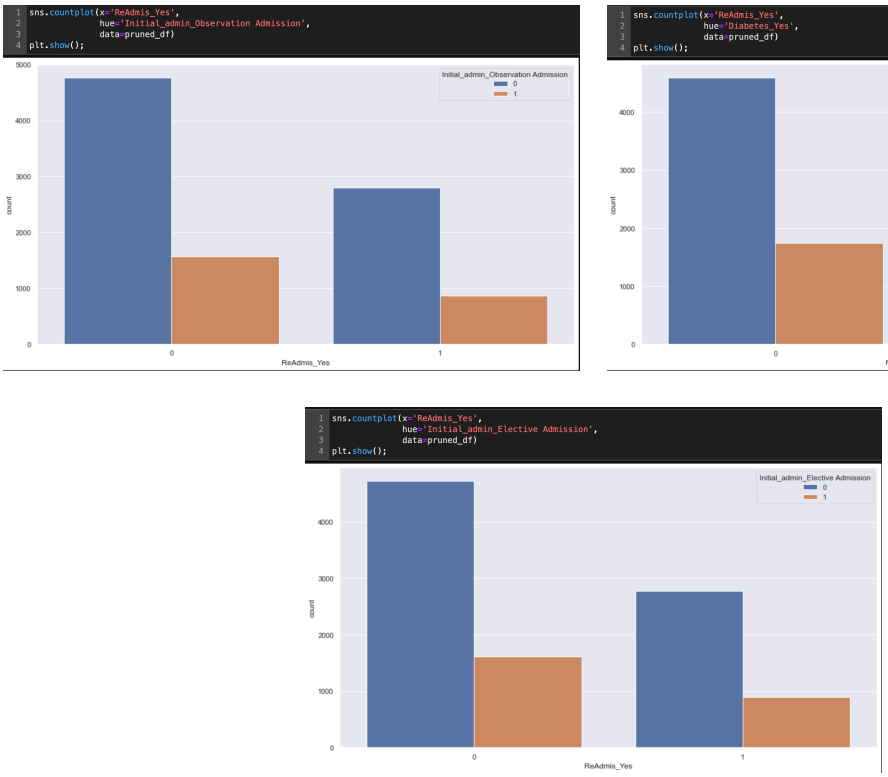


Table 2 - Bivariate Visualizations

C5: Prepared Data Set

Saved within the provided folder is the "JWillis_ D208_ PA2_LogReg.ipynb" file, demonstrating the steps taken to explore and analyze the provided data set from the "medical_clean.csv" file. Once Jupyter Notebook or JupyterLab is running, navigate to the open the "JWillis_ D208_ PA2_LogReg " folder. From there, you will find the JWillis_ D208_ PA2_LogReg.ipynb file and the "medical_clean.csv" file. Run all cells by navigating to Cells → 'Run All' in Jupyter Notebook or View → 'Run All Cells' from within JupyterLab. The first step is to import libraries and read the raw data from the CSV. Once the libraries are imported, the data is read into the notebook. Ensure the medical_raw_data.csv file is kept in the same folder or provide the correct path when reading in the data.

D1: Initial Model

The submission provides an accurate initial logistic regression model from all predictors identified in the hypothesis.

D2 – Justification of Model Reduction

The model's variables were further reduced (Figure 6 and 7) after initially running the model, comparing variable p-values to a threshold of 0.05, retaining values that were lower than the cutoff, and rerunning the model.

D3 –Reduced Logistic Regression Model

Submission provides a reduced logistic regression model, including categorical and continuous variables, aligned with the hypothesis.

E1 – Model Comparison

Using stepwise regression helped logically choose which variables to keep and which variables to remove. As seen in Figure 6 and 7, while looking at the "P > |t|" column and comparing coefficients against a p-value of 0.05, I kept significant variables that were less than this threshold. Additionally, only Initial_days made the training model threshold from our original hypothesis. Meaning, we would have to accept the null hypothesis as the other predictor variables (Initial_admin and Diabetes_Yes) were all above a p-value threshold of 0.05. Furthermore, I specifically ran and tested only variables from the hypothesis (Figure 8) showing out of the original variables, only Initial_days met the model criteria in both the original run and the training/testing run.

E2 – Output and Calculations

From Reading:

$$\mu = \frac{e^{X\beta}}{1 + e^{X\beta}}$$

The parametric form of the model:

$$p(y) = \frac{\exp\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p\right)}{1 + \exp\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p\right)} + \varepsilon$$

The descriptive form:

$$\hat{p}(y) = \frac{\exp\left(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p\right)}{1 + \exp\left(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p\right)}$$

From the file listed above and/or from Figure 9 or Figure 10, this data is inserted into the final regression equation here:

y' = dependent variable (estimated value of target y) = Total Charges (Total_charge)

X = independent variable(s) (predictors) = Initial Days (Initial_days), Readmission (transformed to ReAdmis_Num), Diabetes (transformed to Diabetes_Num)

exp =

$\beta = \beta_0$ = intercept, *while* $\beta_1, \beta_2, \beta_3 \ldots$ = known values of the regression coefficients

E3 – Code

Code is provided in the JWillis_D208_PA2_LogReg.ipynb file.

F1 – Results

From Data Science Using Python and R (reading):

$$\mu = \frac{e^{X\beta}}{1 + e^{X\beta}}$$

The parametric form of the model:

$$p(y) = \frac{\exp\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p\right)}{1 + \exp\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p\right)} + \varepsilon$$

The descriptive form:

$$\hat{p}(y) = \frac{\exp\left(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p\right)}{1 + \exp\left(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p\right)}$$

From the file listed above and/or from Figure 9 or Figure 10, this data is inserted into the final regression equation here:

y' = dependent variable (estimated value of target y) = Readmission

X = independent variable(s) (predictors) = Initial Days (Initial_days), Initial_admin (Elective, Emergency, Observation transformed using get_dummies()), Diabetes (transformed to Diabetes_Yes)

exp = exponential of the term

$\beta = \beta_0$ = intercept, *while* $\beta_1, \beta_2, \beta_3 \ldots$ = known values of the regression coefficients

Interpretation of most statistically significant coefficients: After running most variables through the model, p-values were compared to a commonly used cutoff threshold of 0.05. Fields with p-

values lower than the cutoff were kept in the model. Since the model utilizes coefficients with respective p-values providing a significant statistical relationship to the target variable and some not providing a significant statistical relationship, we must accept the null hypothesis. Meaning, the probability needed to show a significant relationship between the predictor and target variables was not observed.

Statistical and practical significance of the reduced model (as related to the research question): A reduced model containing most data fields: Initial_days, and Diabetes variables met the cutoff threshold in our initial model. Additionally, the category variable for Diabetes was transformed to numerical output and added to the model as they, too, passed the p-value cutoff threshold.

Limitations of study: While we have a mathematical cutoff threshold set for this logistic regression model, in reality, there may not be correlation between a, some or all predictor variables and the target variable. Correlation doesn't always equal causation, especially in the case of correlated predictor variables.

Chosen predictor variables within the hypothesis did not provide significant statistical relation to the target variable. What the logistic model shows, since we need to accept the null hypothesis, is that we should reframe the question and rerun the tests. For instance, if a hypothesis was created with predictor variables focused on Initial_days, Anxiety_Yes and Stroke_Yes, (Figure 9) then we could reject the null hypotheses since these variables do show a significant statistical relationship to a patient's readmission probability.

```
X_hyp = pruned_df[['Initial_days', 'Anxiety_Yes', 'Stroke_Yes']]
# X = df.select_dtypes(include='number')
X_hyp = sm.add_constant(X_hyp)

y_hyp = pd.DataFrame(pruned_df[['ReAdmis_Yes']])

X_hyp_train, X_hyp_test, y_hyp_train, y_hyp_test = \
train_test_split(X_hyp, y_hyp, train_size=0.70,test_size=0.30, random_state=123)
```

```
logit_hyp = sm.Logit(y_hyp.astype(float), X_hyp.astype(float))
result_hyp = logit_hyp.fit()
print(result_hyp.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.046492
         Iterations 13
                         Results: Logit
=================================================================
Model:              Logit            Pseudo R-squared: 0.929
Dependent Variable: ReAdmis_Yes      AIC:              937.8412
Date:               2022-01-15 13:04 BIC:              966.6825
No. Observations:   10000            Log-Likelihood:   -464.92
Df Model:           3                LL-Null:          -6572.9
Df Residuals:       9996             LLR p-value:      0.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     13.0000
-----------------------------------------------------------------
              Coef.   Std.Err.     z     P>|z|   [0.025   0.975]
-----------------------------------------------------------------
const        -56.7757   2.6668 -21.2901 0.0000 -62.0025 -51.5490
Initial_days   1.0445   0.0489  21.3430 0.0000   0.9486   1.1405
Anxiety_Yes   -0.6751   0.1818  -3.7137 0.0002  -1.0314  -0.3188
Stroke_Yes     1.2700   0.2201   5.7705 0.0000   0.8386   1.7013
=================================================================
```

```
logit_hyp_test = sm.Logit(y_hyp_test.astype(float), X_hyp_test)  #[['const','Init
result_hyp_test = logit_hyp_test.fit()
print(result_hyp_test.summary2())

p_vals = dict(logit_hyp_test.fit().pvalues[1:])
```

```
Optimization terminated successfully.
         Current function value: 0.047925
         Iterations 13
                         Results: Logit
=================================================================
Model:              Logit            Pseudo R-squared: 0.927
Dependent Variable: ReAdmis_Yes      AIC:              295.5511
Date:               2022-01-15 13:04 BIC:              319.5766
No. Observations:   3000             Log-Likelihood:   -143.78
Df Model:           3                LL-Null:          -1965.9
Df Residuals:       2996             LLR p-value:      0.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     13.0000
-----------------------------------------------------------------
              Coef.   Std.Err.     z     P>|z|   [0.025   0.975]
-----------------------------------------------------------------
const        -58.4097   4.9951 -11.6934 0.0000 -68.2000 -48.6195
Initial_days   1.0719   0.0915  11.7198 0.0000   0.8926   1.2511
Anxiety_Yes   -0.6909   0.3247  -2.1278 0.0334  -1.3273  -0.0545
Stroke_Yes     1.7352   0.3902   4.4476 0.0000   0.9706   2.4999
=================================================================
```

Figure 9 - Alternative Predictor Variables

F2 – Recommendations

Looking at the Logistic model, based on the coefficients, the three predictor variables (Initial_days, Initial_admin (type: elective, emergency and observation), and Diabetes_Yes) are not all positively and significantly (due to p-values) correlated with the response variable of Readmissions. Therefore, we did not observe there evidence where longer hospital stays, the initial reason for administration, and diabetes are all predictors of (and cannot determine) a patients readmission within 30 days of being discharged from the hospital.

Here is our recommended course of action: We should rework the hypothesis to focu on predictor variables that pass the threshold testing. For instance, Figure 9 shows an example of how a patients initial days, anxiety and history of stroke could help predict readmissions. Further subsets could also be explored to help predict readmissions, reduce the length of their hospital stays and overall costs while freeing up beds. While focusing on a new subset of patients, we should also recommend that patients reduce their inpatient stay in the hospital for as little time as possible, i.e. avoid getting readmitted if at all possible, and to manage, treat, or minimize their diabetes symptoms as much as possible. Prevention and continued/improved healthier living behaviors should be recommended as lifestyle implementations for this subset of patients in order to reduce additional charges through their initial inpatient days, diabetes, and the possibility of readmission.

G – Panopto Demonstration

Panopto video will be uploaded after this submission is reviewed

H – Sources for Third-Party Code

- Help using Markdown: https://www.markdownguide.org/basic-syntax/
- Help to see ALL columns: https://stackoverflow.com/questions/24524104/pandas-describe-is-not-returning-summary-of-all-columns
- Help to create a better histogram design: https://mode.com/example-gallery/python_histogram/
- Python Help: https://docs.python.org/3.9/library/index.html
- Pandas Help: https://pandas.pydata.org/docs/user_guide/index.html#user-guide
- Numpy Help: https://numpy.org/doc/stable/
- Seaborn Help: https://seaborn.pydata.org/
- Matplotlib Help: https://matplotlib.org/
- Sklearn Help: https://scikit-learn.org/

References

Larose, D., C., & Larose, D., T. (2019). Data Science Using Python and R. Wiley. https://www.wiley.com/en-us/Data+Science+Using+Python+and+R-p-9781119526810

Gert P Westert, Ronald J Lagoe, Ilmo Keskimäki, Alastair Leyland, Mark Murphy, An international study of hospital readmissions and related utilization in Europe and the USA, Health Policy, Volume 61, Issue 3, 2002, Pages 269-278, ISSN 0168-8510, https://doi.org/10.1016/S0168-8510(01)00236-6. (https://www.sciencedirect.com/science/article/pii/S0168851001002366)

Schuller K. A. (2020). Is obesity a risk factor for readmission after acute myocardial infarction? *Journal of healthcare quality research*, *35*(1), 4–11. https://doi.org/10.1016/j.jhqr.2019.09.002