­

**Performance Assessment for D208: Predictive Analytics  
Task 2: Logistic Regression**

Christopher Fischer - 011933891

College of Information Technology, Western Governors University

June 18, 2024

Performance Assessment for D208: Predictive Analytics – Task 2

This document contains the tasks and outputs required for the “NBM3 TASK 2: LOGISTIC REGRESSION MODELING” assessment. All work is original to the author unless otherwise indicated by a citation.

# A - Research Question

The company’s internal and external stakeholders are disappointed with the rising cost of readmittance penalties received from the Centers for Medicare and Medicaid Services (CMMS) over the last several years. While it is obvious that significant efforts need to be made to reduce the number of readmittances our patients experience, no reasonable plan can be formulated without knowing which patients are more likely to be readmitted. Management wants to know *if certain variables are associated with re-admittance being more likely*. This information will allow teams to zero in on the patient populations with the highest likelihood of re-admittance.

With this motivation, the data analytics team has been given the budget and scope to perform a study to determine if it is possible to predict which patients are most likely to be readmitted. Armed with a useful predictive model, management can identify methods for which groups of patients to focus additional care strategies and educational programs designed to minimize readmittance and reduce the costs of penalties received from CMMS.

# B – Method Justification

When attempting to predict the probability of an outcome, data analysts first turn to a classification technique called logistic regression. With this method, an analyst can define an event or outcome they wish to predict and a set of variables that may explain the probability of that outcome occurring. The regression algorithm then analyzes the data, developing a formula that estimates the relationship between the predictors and the target outcome. Unlike linear regression, the logistic regression formula will result in a sigmoid-shaped plot. (Logistic Regression in Machine Learning, 2024)

Logistic regression is a widely used method for conducting binary classification. However, there are several critical assumptions that, if unmet, may render the study results unusable. These assumptions include:

1. **Response variable is binary** – The target variable must have only two possible values, such as 0/1, True/False, or Yes/No.
2. **No excessive outliers** – Extreme values should be removed from the dataset before modeling
3. **Sufficiently large dataset** – The dataset should contain at least as many observations as indicated by the formula:  
    N = (10 \* number\_of\_predictor\_variables) / pct\_of\_positive\_cases
4. **Independent observations** – Each data record used in the model is independent of all the other records.
5. **Independent predictors** – The model's prediction variables are generally independent. In data science speak, this is called the absence of *multicollinearity*.
6. **Linear relationship between each predictor variable and the logit of the response variable** – Logit is the logarithm of the odds of the probability of an outcome. (Bobbitt, 2020)

Since this study aims to predict the probability of an event occurring (hospital readmittance) from a set of explanatory variables, logistic regression is the ideal statistical method.

The data science team will develop the code for this study using Python. In its nearly 35 years, Python has become one of the most widely used general-purpose programming languages. Python is backed by a massive community of developers and a vast collection of libraries that extend the core capabilities of the language. (Datacamp, 2022) These libraries provide functionality critical to performing the logistic regression analysis this study requires. While R provides similar capabilities, Python offers multiple libraries that implement logistic regression, allowing the team to choose the library with the ideal features. Another reason for selecting Python is its scalability. As a higher-level language, Python can more easily handle large datasets without running into the same constraints that may be faced when using R. (Python vs. R for Data Science 2023: Which is better?, 2024) This performance will help to ensure that the data cleansing and transformation steps go smoothly. Lastly, while R is an excellent choice for interactive studies, Python is better suited for deployment on production servers. (WGU Information Technology, n.d.) This will make it easier for the model resulting from this study to be productized, allowing it to be used as part of an unattended data pipeline.

# C – Data Preparation

The following sections outline the team’s plan to clean and prepare the input data for use in predictive modeling.

## C1 – Data Cleaning

Although the source data provided to the team for this study is labeled “clean,” we must ensure the data quality before beginning the predictive modeling activity. Poor data will negatively impact the model's accuracy, making it unusable for decision-making.

The cleaning steps which will be performed in this study include:

* Identify and treat missing data
* Identify and treat outlying values
* Identify and treat any invalid categorical values

### Cleaning code and results

# Check missing data

msno.matrix(df, fontsize = 12, labels=True)

plt.title('Missing Data Matrix')

plt.show()

A black and white striped background

Description automatically generated

No variables exhibit missing data.

#Check values in yes/no variables

print(df[yes\_no\_variables][~df[yes\_no\_variables].isin(['Yes','No'])].count())

A screenshot of a computer screen

Description automatically generated

Boolean variables all contain valid values.

#Detect potential outliers

df\_z = (df[numerical\_variables] - df[numerical\_variables].mean())/df[numerical\_variables].std(ddof=0)

outlier\_cols = df\_z.loc[: , (df\_z > 3.0).any()].columns

for col in outlier\_cols :

    cnt = len(df\_z[df\_z[col]>3])

    min, max = df[col].min(), df[col].max()

    print('Likely outlier for {0:<20}\t Count: {1:7d} ({2:5.2%} of observations)\tMin: {3:>9.2f}\tMax: {4:>9.2f}'.format(col,cnt,cnt/10000,min,max))

A screen shot of a computer

Description automatically generated

While a small percentage of values exceed the norms for several variables, only *Population* demonstrates values that require further investigation and cleaning.

# Deeper look at Population

sns.boxplot(data=df, x='Population')

plt.show()

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

A graph with a bar graph

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Since a population cannot logically be zero, these observations will be removed from the dataset.

# Drop rows with Population <= 0

df = df[df.Population > 0]

df['Population'].count()

This leaves 9,891 observations in the dataset.

## C2 – Summary Statistics for Target and Predictor Variables

This section provides a brief exploration of the predictor variables to be included in the study, as well as the target variable. Please note that while the table includes all the variables that will be included in the initial model, the final model will likely include fewer variables once tuning is complete.

| Variable | Statistics | Comment |
| --- | --- | --- |
| Target | | |
| ReAdmis | No 6262  Yes 3629  Mode: No | About 37% of patients are readmitted within one month. |
| Predictors - Numeric | | |
| VitD\_levels | count 9891.000000  mean 17.966051  std 2.017781  min 9.806483  25% 16.629179  50% 17.952513  75% 19.352843  max 26.394449 | Serum vitamin D levels below 20 ng/mL are considered insufficient for good health. (National Institutes of Health, Office of Dietary Supplements, 2023) Hypothesis: lower vitamin D levels are correlated with readmission. |
| Doc\_visits | count 9891.000000  mean 5.012537  std 1.046176  min 1.000000  25% 4.000000  50% 5.000000  75% 6.000000  max 9.000000 | Hypothesis: patients with more doctor visits on their first hospitalization are more likely to be readmitted. |
| Full\_meals\_eaten | count 9891.000000  mean 1.000506  std 1.008006  min 0.000000  25% 0.000000  50% 1.000000  75% 2.000000  max 7.000000 |  |
| VitD\_supp | count 9891.000000  mean 0.398847  std 0.628892  min 0.000000  25% 0.000000  50% 0.000000  75% 1.000000  max 5.000000 |  |
| Income | count 9891.000000  mean 40484.044689  std 28511.621864  min 154.080000  25% 19609.470000  50% 33773.250000  75% 54292.740000  max 207249.100000 | Hypothesis: lower-income patients are more likely to be readmitted. |
| Population | count 9891.000000  mean 10075.072086  std 14869.065326  min 1.000000  25% 730.000000  50% 2859.000000  75% 14161.000000  max 122814.000000 |  |
| Age | count 9891.000000  mean 53.555859  std 20.648187  min 18.000000  25% 36.000000  50% 54.000000  75% 71.000000  max 89.000000 | Patients in our dataset skew older. Hypothesis: older patients are more likely to be readmitted. |
| Children | count 9891.000000  mean 2.097563  std 2.164304  min 0.000000  25% 0.000000  50% 1.000000  75% 3.000000  max 10.000000 |  |
| Initial\_days | count 9891.000000  mean 34.469513  std 26.310525  min 1.001981  25% 7.896060  50% 36.266030  75% 61.157520  max 71.981490 | Patients in our dataset endured lengthy initial hospital stays. Hypothesis: the longer a patient is in hospital, the more likely they will require readmittance. |
| Predictors – Categorical | | |
| Initial\_admin | EmergencyAdmission 5014  ElectiveAdmission 2475  ObservationAdmission 2402  Mode: EmergencyAdmission | Half of the patients in our dataset entered the hospital via the emergency department. Hypothesis: emergency admittees are more likely to be readmitted. |
| Complication\_risk | Medium 4461  High 3324  Low 2106  Mode: Medium | Hypothesis: the higher the complication risk, the more likely to be readmitted. |
| Marital | Widowed 2016  Married 1998  Separated 1968  NeverMarried 1961  Divorced 1948  Mode: Widowed |  |
| Gender | Female 4969  Male 4710  Nonbinary 212  Mode: Female | Slightly more females than males. |
| Services | BloodWork 5215  Intravenous 3091  CTScan 1210  MRI 375  Mode: BloodWork |  |
| Area | Rural 3330  Suburban 3291  Urban 3270  Mode: Rural | Hypothesis: patients from a rural area are more likely to be readmitted. |
| State | TX 549  PA 543  CA 538  NY 505  IL 442  OH 380  MO 325  FL 302  VA 286  IA 275  MI 267  MN 265  NC 252  GA 244  KS 218  WI 214  KY 209  WV 207  OK 206  IN 195  TN 194  AL 191  AR 187  WA 187  NE 185  NJ 176  CO 176  LA 169  MA 149  MS 132  MD 130  SC 127  OR 122  SD 122  ME 118  MT 112  ID 108  NM 108  ND 108  AZ 105  NH 79  CT 79  AK 70  UT 68  VT 60  NV 49  WY 47  PR 42  HI 33  DE 16  RI 14  DC 6  Mode: TX |  |
| Soft\_drink | No 7340  Yes 2551  Mode: No | Hypothesis: drinking soda is correlated to readmittance. |
| HighBlood | No 5842  Yes 4049  Mode: No | Hypothesis: high blood pressure is correlated to readmittance. |
| Stroke | No 7922  Yes 1969  Mode: No | Hypothesis: having had a stroke previously makes a patient more likely to be readmitted. |
| Overweight | Yes 7021  No 2870  Mode: Yes | More than 70% of our patients are overweight. Hypothesis: overweight patients are more likely to be readmitted. |
| Hyperlipidemia | No 6549  Yes 3342  Mode: No | Hypothesis: patients with high cholesterol are more likely to be readmitted. |
| BackPain | No 5820  Yes 4071  Mode: No |  |
| Anxiety | No 6714  Yes 3177  Mode: No | Hypothesis: patients with anxiety are more likely to be readmitted. |
| Allergic\_rhinitis | No 5110  Yes 4890  Mode: No |  |
| Reflux\_esophagitis | No 5803  Yes 4088  Mode: No |  |
| Asthma | No 7024  Yes 2867  Mode: No | Hypothesis: patients with asthma are more likely to be readmitted. |
| Diabetes | No 7191  Yes 2700  Mode: No | Hypothesis: patients with diabetes are more likely to be readmitted. |

## C3 – Univariate and Bivariate Visualizations

The following graphs visualize the univariate and bivariate statistics for each study variable.

### Target Variable

A graph with blue rectangular bars

Description automatically generated

### Predictor Numeric Variables

A group of blue bars

Description automatically generated

### Predictor Categorical Variables

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Numeric Predictors versus Target

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Only *Initial\_days* shows a strong correlation to the target variable of these predictors. That is a discouraging start.

### Categorical Predictors versus Target

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

By looking at the difference in proportion between the *Yes* and *No* bars of each graph, the following predictors seem to show some correlation to the target variable: *Initial\_admin, Services, Soft\_drink, HighBlood, Stroke, Hyperlipidemia, BackPain, Anxiety,* and *Diabetes*. Nevertheless, further assessment will occur during the modeling phase of this study.

## C4 – Data Transformation

The approach taken in this study aims to create an initial model that includes as many of the candidate predictor variables as possible. Subsequently, the model will be reduced to eliminate variables that do not contribute materially to the model’s accuracy.

This requires the team to transform certain variables to permit their inclusion in the model. The transformations planned for this study include:

* **Re-express *Gender*, *Marital, Services,* and *Initial\_admin* categorical variables using one-hot encoding**  
  As logistic regression works only on variables with numerical values, categorical variables with textual values must be re-expressed into a numeric form. One-hot encoding involves creating a new variable for each category's values. For example, the variable *Gender* has three values: Male, Female, and Non-binary. Using one-hot encoding, new variables will be created for two values, *Gender\_Male* and *Gender\_Nonbinary*, each containing a zero or one to indicate which gender applies to the observation. Notice that only two new variables will be created, not three. This is known as the *k-1 rule,* which stipulates that one-hot encoding must always create one less new variable than there are category values. This is done to prevent multi-collinearity. (Mahto, 2019)
* **Re-express any Boolean (yes/no) variables as numeric**This is a more straightforward implementation of the previous transformation. To be used in the regression model, these yes/no values must be converted to one and zero, respectively.
* **Re-express *State* with frequency encoding**  
  The *State* variable has 52 unique values, so it is not a good candidate for one-hot encoding. Doing so would lead to the “curse of dimensionality, " leading to poor performance and model overfitting. (Brownlee, 2020) To avoid this, the *State* variable will be re-expressed with frequency encoding. This method will replace each two-letter state code with the count of how many times that code appears in the dataset. Encoding this way provides additional insight into that variable for the modeling algorithm. (Neural Ninja, 2023)
* **Re-express *Complication\_risk* and *Area* with simple ordinal encoding**These textual categorical variables each exhibit a natural ordering and will, therefore, be re-expressed as ordinal integer values.
* **Variables exhibiting high multicollinearity will be removed from consideration before producing the initial model**  
  Multicollinearity occurs when two or more predictor variables are strongly correlated. If such variables were included in a logistic regression model, it would be impossible for the model to determine which predictors were responsible for a change in the target variable. To detect problematic variables, the *Variance Inflation Factor* (VIF) will be calculated for each variable. The variable with the highest VIF over a threshold of 5.0 will be removed from consideration. This process will be repeated until variables with a VIF > 5.0 remain in the set to be modeled. (Frost, 2017)

### Transformation code and results

# Reexpress yes/no columns as numbers [In-Text Citation:(Eiler, 2017)]

yesno\_dict = {'No': 0, 'Yes': 1}

for col in yes\_no\_variables:

    df[col] = df[col].map(yesno\_dict)

print(df[yes\_no\_variables].info())

A screenshot of a computer

Description automatically generated

# Rexpress State with frequency encoding

df['State'] = df['State'].map(df['State'].value\_counts().to\_dict())

print(df['State'].info())

# Rexpress Complication\_risk with simple ordinal encoding

complication\_map = {'Low': 1, 'Medium': 2, 'High' : 3}

df['Complication\_risk'] = df['Complication\_risk'].map(complication\_map)

print(df['Complication\_risk'].info())

# Rexpress Area with simple ordinal encoding

area\_map = {'Rural': 1, 'Suburban': 2, 'Urban' : 3}

df['Area'] = df['Area'].map(area\_map)

print(df['Area'].info())

A computer screen shot of a computer program

Description automatically generated

# One-hot encoding

for feature in ['Gender','Marital','Services','Initial\_admin'] :

    dummies = pd.get\_dummies(df[feature], drop\_first=True, prefix=feature, dtype=np.int64)

    print(dummies.info())

    X\_full.remove(feature)

    for newcol in dummies.columns :

        X\_full.append(newcol)

    df = pd.concat([df, dummies], axis=1).drop(feature, axis=1)

<class 'pandas.core.frame.DataFrame'>

Index: 9891 entries, 0 to 9999

Data columns (total 2 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Gender\_Male 9891 non-null int64

1 Gender\_Nonbinary 9891 non-null int64

dtypes: int64(2)

memory usage: 231.8 KB

None

<class 'pandas.core.frame.DataFrame'>

Index: 9891 entries, 0 to 9999

Data columns (total 4 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Marital\_Married 9891 non-null int64

1 Marital\_NeverMarried 9891 non-null int64

2 Marital\_Separated 9891 non-null int64

3 Marital\_Widowed 9891 non-null int64

dtypes: int64(4)

memory usage: 386.4 KB

None

<class 'pandas.core.frame.DataFrame'>

Index: 9891 entries, 0 to 9999

Data columns (total 3 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Services\_CTScan 9891 non-null int64

1 Services\_Intravenous 9891 non-null int64

2 Services\_MRI 9891 non-null int64

dtypes: int64(3)

memory usage: 309.1 KB

None

<class 'pandas.core.frame.DataFrame'>

Index: 9891 entries, 0 to 9999

Data columns (total 2 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Initial\_admin\_EmergencyAdmission 9891 non-null int64

1 Initial\_admin\_ObservationAdmission 9891 non-null int64

dtypes: int64(2)

memory usage: 231.8 KB

for feature colliniarity and drop high VIFs [In-Text Citation: (Prashant, 2016)]

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

def calculate\_vif(X, thresh=5.0):

    X = X.assign(const=1)

    variables = list(range(X.shape[1]))

    dropped = True

    while dropped:

        dropped = False

        vif = [variance\_inflation\_factor(X.iloc[:, variables].values, ix)

               for ix in range(X.iloc[:, variables].shape[1])]

        vif = vif[:-1]  # always preserve the constant

        maxvif = np.max(vif)

        maxloc = vif.index(maxvif)

        if maxvif > thresh:

            print(f'dropping {X.iloc[:, variables].columns[maxloc]:30} at index: {str(maxloc)} with VIF: {maxvif:.5f}')

            del variables[maxloc]

            dropped = True

    return X.iloc[:, variables[:-1]], X.columns[variables[:-1]].tolist()

tmp\_df, X\_full = calculate\_vif(df[X\_full],5.0)

# Merge post VIF reduction df with the target variable from the df

df = pd.concat([tmp\_df, df[y]], axis=1)

No variables were dropped due to high VIF.

## C5 – Prepared Data File

This file contains the cleaned and prepared data used in subsequent modeling activities.



# D – Initial and Reduced Model

The initial model built for this study included the following predictor variables:

* VitD\_levels
* Doc\_visits
* Full\_meals\_eaten
* vitD\_supp
* Income
* Population
* Age
* Children
* Initial\_days
* Complication\_risk
* Area
* State
* Soft\_drink
* HighBlood
* Stroke
* Overweight
* Hyperlipidemia
* BackPain
* Anxiety
* Allergic\_rhinitis
* Reflux\_esophagitis
* Asthma
* Diabetes
* Gender\_Male
* Gender\_Nonbinary
* Marital\_Married
* Marital\_NeverMarried
* Marital\_Separated
* Marital\_Widowed
* Service\_CTScan
* Service\_Intravenous
* Service\_MRI
* Initial\_EmergencyAdmission
* Initial\_ObservationAdmission

Here is the summary of the initial model:

Logit Regression Results

==============================================================================

Dep. Variable: ReAdmis No. Observations: 9891

Model: Logit Df Residuals: 9856

Method: MLE Df Model: 34

Date: Sat, 22 Jun 2024 Pseudo R-squ.: 0.9465

Time: 12:52:09 Log-Likelihood: -347.57

converged: True LL-Null: -6501.2

Covariance Type: nonrobust LLR p-value: 0.000

================================================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------------------------

Intercept -78.5848 4.370 -17.983 0.000 -87.150 -70.020

VitD\_levels 0.0233 0.048 0.488 0.626 -0.070 0.117

Doc\_visits -0.0319 0.095 -0.336 0.737 -0.218 0.154

Full\_meals\_eaten 0.0213 0.101 0.212 0.832 -0.176 0.219

vitD\_supp -0.1423 0.157 -0.908 0.364 -0.449 0.165

Income 1.132e-06 3.56e-06 0.318 0.750 -5.84e-06 8.1e-06

Population 7.487e-06 7.07e-06 1.058 0.290 -6.38e-06 2.14e-05

Age 0.0006 0.005 0.124 0.901 -0.009 0.010

Children 0.0726 0.045 1.621 0.105 -0.015 0.160

Initial\_days 1.3748 0.075 18.263 0.000 1.227 1.522

Complication\_risk 0.8224 0.142 5.811 0.000 0.545 1.100

Area 0.0372 0.125 0.297 0.767 -0.209 0.283

State -0.0001 0.001 -0.221 0.825 -0.001 0.001

Soft\_drink 0.1504 0.236 0.637 0.524 -0.312 0.613

HighBlood 0.8177 0.211 3.873 0.000 0.404 1.231

Stroke 1.6211 0.265 6.110 0.000 1.101 2.141

Overweight -0.3990 0.225 -1.776 0.076 -0.839 0.041

Hyperlipidemia 0.2065 0.211 0.979 0.328 -0.207 0.620

BackPain 0.3403 0.202 1.686 0.092 -0.055 0.736

Anxiety -0.9664 0.220 -4.399 0.000 -1.397 -0.536

Allergic\_rhinitis -0.3357 0.207 -1.623 0.105 -0.741 0.070

Reflux\_esophagitis -0.4259 0.208 -2.044 0.041 -0.834 -0.018

Asthma -1.2327 0.227 -5.430 0.000 -1.678 -0.788

Diabetes 0.4001 0.224 1.784 0.074 -0.040 0.840

Gender\_Male 0.0331 0.204 0.163 0.871 -0.366 0.432

Gender\_Nonbinary 0.3754 0.644 0.582 0.560 -0.888 1.639

Marital\_Married 0.2649 0.317 0.836 0.403 -0.356 0.886

Marital\_NeverMarried 0.3343 0.329 1.016 0.310 -0.311 0.979

Marital\_Separated -0.0911 0.331 -0.275 0.783 -0.740 0.558

Marital\_Widowed 0.2415 0.321 0.752 0.452 -0.388 0.871

Service\_CTScan 1.6665 0.354 4.711 0.000 0.973 2.360

Service\_Intravenous -0.0072 0.227 -0.032 0.975 -0.451 0.437

Service\_MRI 2.4410 0.481 5.071 0.000 1.498 3.384

Initial\_EmergencyAdmission 2.3578 0.275 8.577 0.000 1.819 2.897

Initial\_ObservationAdmission 0.7884 0.281 2.809 0.005 0.238 1.339

================================================================================================

Possibly complete quasi-separation: A fraction 0.81 of observations can be

perfectly predicted. This might indicate that there is complete

quasi-separation. In this case some parameters will not be identified.

The initial model exhibits substantial accuracy, as measured by Pseudo-R2 and LLR p-value. However, there are many variables with high p-values. It would be best to remove these variables to improve computational efficiency and reduce the chance of model overfitting. (Tripathi, 2019)

To improve the initial model, backward feature elimination will be performed to remove the least significant features from the initial model. Backward elimination, one of the wrapper methods of feature selection, is an iterative process where the feature with the highest p-value above a given alpha threshold is removed, and the model is re-fitted. These steps are repeated until no further improvement in the model can be made. (Middleton, 2023) This feature selection method was selected because it is quite easy to explain to stakeholders.

For this study, the alpha threshold selected was 0.05. This resulted in 24 features being eliminated. After backward elimination was performed, a reduced model was built that included the following predictor variables:

* Initial\_days
* Complication\_risk
* HighBlood
* Stroke
* Anxiety
* Asthma
* Service\_CTScan
* Service\_MRI
* Initial\_EmergencyAdmission
* Initial\_ObservationAdmission

Here is the summary of the reduced model:

Logit Regression Results

==============================================================================

Dep. Variable: ReAdmis No. Observations: 9891

Model: Logit Df Residuals: 9880

Method: MLE Df Model: 10

Date: Sat, 22 Jun 2024 Pseudo R-squ.: 0.9446

Time: 12:58:28 Log-Likelihood: -360.26

converged: True LL-Null: -6501.2

Covariance Type: nonrobust LLR p-value: 0.000

================================================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------------------------

Intercept -75.1936 4.001 -18.792 0.000 -83.036 -67.351

Initial\_days 1.3249 0.070 18.887 0.000 1.187 1.462

Complication\_risk 0.7907 0.136 5.827 0.000 0.525 1.057

HighBlood 0.7628 0.200 3.811 0.000 0.371 1.155

Stroke 1.4927 0.254 5.881 0.000 0.995 1.990

Anxiety -0.8527 0.209 -4.074 0.000 -1.263 -0.443

Asthma -1.1964 0.220 -5.432 0.000 -1.628 -0.765

Service\_CTScan 1.6028 0.328 4.891 0.000 0.961 2.245

Service\_MRI 2.4229 0.461 5.256 0.000 1.519 3.326

Initial\_EmergencyAdmission 2.2458 0.261 8.597 0.000 1.734 2.758

Initial\_ObservationAdmission 0.7194 0.269 2.675 0.007 0.192 1.247

================================================================================================

Possibly complete quasi-separation: A fraction 0.80 of observations can be

perfectly predicted. This might indicate that there is complete

quasi-separation. In this case some parameters will not be identified.

The reduced model also exhibits strong accuracy, as measured by Pseudo-R2 and LLR p-value. Additionally, the warning observed on the initial model’s summary has been resolved.

# E – Model Evaluation

It is important to assess the changes to the model’s performance after feature selection has been performed. For this study, the results are as follows:

╒═════════╤═════════╤════════════╤═════════════╤══════════════╤═══════════╤════════════╕

│ model │ pr2 │ llf │ llr │ llr-pvalue │ aic │ bic │

╞═════════╪═════════╪════════════╪═════════════╪══════════════╪═══════════╪════════════╡

│ Full │ 0.94654 │ -347.56547 │ 12307.27569 │ 0.00000 │ 765.13094 │ 1017.10926 │

│ Reduced │ 0.94459 │ -360.25876 │ 12281.88911 │ 0.00000 │ 742.51752 │ 821.71071 │

╘═════════╧═════════╧════════════╧═════════════╧══════════════╧═══════════╧════════════╛

By all measures, the reduced model is almost identical in performance to the initial model despite containing 24 fewer predictor variables.

Confusion matrices were produced as an additional comparison between the initial and reduced models. Each model shows similar accuracy and precision results.

|  |  |
| --- | --- |
|  |  |

All code used to perform this study can be found in this Python notebook.



# F – Summary and Implications

## F1 – Analysis and Discussion

The logistic regression of the reduced model can be expressed in the following equation:

(Middleton, Getting Started with D208 - Part 2, 2023)

This regression equation can be understood more simply by saying that keeping all things constant:

* A one-unit increase in *InitialDays* changes the log odds of readmittance by 1.3249
* A one-unit increase in *Complication\_risk* changes the log odds of readmittance by 0.7907
* Having *HighBlood* pressure changes the log odds of readmittance by 0.7628
* Having had a *Stroke* changes the log odds of readmittance by 1.4927
* Having *Anxiety* changes the log odds of readmittance by -0.8527
* Having *Asthma* changes the log odds of readmittance by -1.1964
* Having had a *Service\_CTScan* during the initial hospitalization changes the log odds of readmittance by 1.6028
* Having had a *Service\_MRI* during the initial hospitalization changes the log odds of readmittance by 2.4229
* Having come into the hospital initially via *Initial\_EmergencyAdmission* changes the log odds of readmittance by 2.2458
* Having come into the hospital initially via *Initial\_ObservationAdmission* changes the log odds of readmittance by 0.7194

(Middleton, Getting Started with D208 - Part 2, 2023)

The results documented in [E – Model Evaluation](#_E_–_Model) clearly demonstrate the statistical significance of the reduced model. The model yields a high Pseudo R2 value, very low p-value, and improved AIC and BIC compared to the initial model. These are all helpful since a model with poor performance is useless in making practical applications.

Looking at the predictor variables that remained in the reduced model after feature selection, signs can be seen that indicate the practical significance of the model. Four of the predictors are categorical variables indicating whether a patient has certain comorbidities. Five more of the predictors are variables describing characteristics of a patient’s initial hospital stay. Knowing these variables are correlated to readmittance will provide management with the information they need to identify new programs and procedures aimed at reducing the likelihood of readmittance among certain patient populations.

The model also supports several hypotheses documented in [C2 – Summary Statistics for Target and Predictor Variables](#_C2_–_Summary). Namely:

* Patients with longer initial hospital stays are at increased likelihood of being readmitted
* Patients initially admitted through the emergency department have a significantly higher likelihood of being readmitted than those who enter via other channels
* Patients with higher complication risks are more likely to be readmitted
* Patients having high blood pressure or having had a stroke previously are more likely to be readmitted
* Patients who had a CT scan or MRI during their initial stay are more likely to be readmitted

Nevertheless, no model is perfect, and no modeling technique is perfect. There are implications to every decision made in a study. In this study, there are several important implications:

* **Most outliers were retained**.The suspected outlier values of six numeric variables were retained. While none of those six variables remained in the reduced model, it is possible that the outcome of feature selection would have differed if outliers had been removed,
* **Some outliers were removed.** Data cleansing revealed that 109 observations had population values of zero. Analysts removed these rows since this was just 1.1% of the observations. An alternative approach would have been to cleanse the zero values by substituting mean or median.
* **VIF removal was performed before the initial model.** The team attempted to use VIF to filter out variables with high multicollinearity before the initial model was created. No variables were removed. An alternative approach would have been to do so as part of the feature selection process. Doing so may have resulted in a different subset of variables retained in the reduced model.
* **Backward Feature Elimination was chosen for feature selection**. While wrapper methods are more computationally intensive than other feature selection methods, the impact of this limitation on our study is quite small because there are relatively few observations in the population. As an alternative, several feature selection methods could have been employed to compare the results and identify which method yielded the best model. (Oleszak, 2023)
* **Survey answers were excluded**. While selecting features to include in the initial model, the team consulted with subject matter experts in the business units. Together, the stakeholders decided to exclude the survey answers from consideration. It is possible the including those features would have yielded a different reduced model.

## F2 – Recommendations

The research question posed for this survey indicates that management is concerned that so many patients are being readmitted, costing the company more by doing so. The logistic regression model produced to address this concern can help the clinical team identify patient subgroups that could benefit from targeted treatment and educational programs aimed at reducing readmittance. Specifically, attention should be paid to patients with long initial stays, having certain comorbidities, or receiving advanced scanning during their initial stay.

Additional studies should be used to compare outcomes from multiple treatment and education strategies. This could take the form of A/B testing or simply repeating the logistic regression after enough new data are collected to perform a viable study.

# G – Recorded Code Review

A recording of the code review presentation was uploaded with this submission. For quick reference, that video may be found here: [Panopto Recording](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=88151503-3cdf-40b3-8e7e-b19701446334)

# H – Third-Party Code References

Andrews, T. (2018, 05 01). *Extract a subset of key-value pairs from dictionary*. Retrieved from StackOverflow: https://stackoverflow.com/questions/5352546/extract-a-subset-of-key-value-pairs-from-dictionary

Astinin, S. (2022, 11 25). *tabulate 0.9.0*. Retrieved from Python Package Index: https://pypi.org/project/tabulate/

Eiler, J. (2017, 01 16). *Remap values in pandas column with a dict, preserve NaNs*. Retrieved from Stackoverflow: https://stackoverflow.com/questions/20250771/remap-values-in-pandas-column-with-a-dict-preserve-nans

Prashant. (2016, 12 28). *How to systematically remove collinear variables (pandas columns) in Python?* Retrieved from Cross Validated: https://stats.stackexchange.com/questions/155028/how-to-systematically-remove-collinear-variables-pandas-columns-in-python

Ray, S. (2020, 10 21). *Show all colums of a pandas-dataframe in ".describe()"*. Retrieved from Stackoverflow: https://stackoverflow.com/questions/64455605/show-all-colums-of-a-pandas-dataframe-in-describe

Tuk, D. (2019, 07 25). *Confusion Matrix Visualization*. Retrieved from Medium: https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea

Vijayakumar, A. (2019, 02 11). *Stepwise Regression*. Retrieved from Github: https://github.com/AakkashVijayakumar/stepwise-regression

# I – Referenced Works

Bobbitt, Z. (2020, 10 13). *The 6 Assumptions of Logistic Regression (With Examples)*. Retrieved from Statology: https://www.statology.org/assumptions-of-logistic-regression/

Brownlee, J. (2020, 06 30). *Introduction to Dimensionality Reduction for Machine Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/dimensionality-reduction-for-machine-learning/

Datacamp. (2022). *Python vs R for Data Science: Which Should You Learn?* Retrieved 03 22, 2024, from Datacamp Blog: https://www.datacamp.com/blog/python-vs-r-for-data-science-whats-the-difference

Frost, J. (2017, 09 22). *Multicollinearity in Regression Analysis: Problems, Detection, and Solutions*. Retrieved from Statistics by Jim: https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/#comments

Larose, C. D., & Larose, D. T. (2019). *Data Science Using Python and R.* Hoboken, NJ, USA: Wiley. Retrieved 03 21, 2024, from https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2091371&site=eds-live&scope=site

*Logistic Regression in Machine Learning*. (2024, 06 20). Retrieved from Geeks For Geeks: https://www.geeksforgeeks.org/understanding-logistic-regression/

Mahto, K. K. (2019, 07 08). *One-Hot-Encoding, Multicollinearity and the Dummy Variable Trap*. Retrieved from Towards Data Science: https://towardsdatascience.com/one-hot-encoding-multicollinearity-and-the-dummy-variable-trap-b5840be3c41a

Middleton, K. D. (2023, 07 13). *Getting Started with D208 - Part 1*. Retrieved from Data Science Team - Student Facing Resources: https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?csf=1&web=1&e=9ccodm&cid=656002fd%2Da3e2%2D4286%2Da361%2D8300bc813a87&FolderCTID=0x01200022092E63FD85A64A8ABFB4F5AEA4839A&id=%2Fsites%2FDataScie

Middleton, K. D. (2023, 07 13). *Getting Started with D208 - Part 2*. Retrieved from Data Science Team - Student Facing Resources: <https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?csf=1&web=1&e=9ccodm&cid=656002fd%2Da3e2%2D4286%2Da361%2D8300bc813a87&FolderCTID=0x01200022092E63FD85A64A8ABFB4F5AEA4839A&id=%2Fsites%2FDataScie>

*Python vs. R for Data Science 2023: Which is better?*. (2024, 04 18). Retrieved from ProjectPro: https://www.projectpro.io/article/data-science-programming-python-vs-r/128#mcetoc\_1ght2bqoe88

National Institutes of Health, Office of Dietary Supplements. (2023, 09 18). *Vitamin D Fact Sheet for Health Professionals*. Retrieved from NIH Health Information: https://ods.od.nih.gov/factsheets/VitaminD-HealthProfessional/

Neural Ninja. (2023, 06 12). *Frequency Encoding: Counting Categories for Representation*. Retrieved from Let's Data Science: https://letsdatascience.com/frequency-encoding/

Oleszak, M. (2023, 08 01). *Feature Selection Methods and How to Choose Them*. Retrieved from Neptune.ai: https://neptune.ai/blog/feature-selection-methods

Tripathi, A. (2019, 06 10). *Feature Selection Techniques in Regression Model*. Retrieved from Towards Data Science: https://towardsdatascience.com/feature-selection-techniques-in-regression-model-26878fe0e24e

WGU Information Technology. (n.d.). *R or Python*. Retrieved 06 22, 2024, from https://www.wgu.edu/online-it-degrees/programming-languages/r-or-python.html