Performance Assessment D208 – Predictive Modeling  
Task II

Doug Haunsperger

College of IT, Western Governors University

November 3, 2023

# Part I. Rese**arch Questi**on

## A1. Question Description

I am choosing to examine the question, “What factors might cause a stroke?”. In other words, I will determine if any of the other data collected in our dataset have a correlation and possible causal relationship with whether the patient has had a stroke (as indicated by the Stroke variable).

## A2. Goals

The answer to this question could lead to patients and their caregivers being more mindful of looking for stroke symptoms if the patient already exhibits some of the correlating factors.

# Part II. Method Justification

## B1. Summary of Assumptions

A logistic regression model tries to predict the probability that a binary response variable **y** is 1, based on different predictor variables **xn** with different weights **βn**, and a random noise factor ε. This probability can be expressed as an equation:

(Larose & Larose, 2019, p. 189)

Bobbit (2020) gives six factors assumed to hold for logistic regression models to be valid. The first and last given are nearly self-evident or definitional – that one needs a binary response variable, and a sufficient sample size for drawing conclusions. The remaining four are:

1. Independence – each observation in the dataset is unrelated to the others.
2. No multicollinearity – the explanatory variables are not highly correlated amongst themselves.
3. No extreme outliers in the data set
4. Linearity – there exists a linear relationship between each predictor variable and the logit of the response variable.

## B2. Tool Benefits

Python, via the numpy, statsmodels, and scikit-learn packages, has several easy-to-use functions for fitting logistic regression models. Via matplotlib and seaborn, there are multiple ways to visualize model fit, residuals, errors, etc.

## B3. Appropriate Technique

Logistic regression is an appropriate technique to use for this research question because the task is to “understand the relationship between one or more predictor variables” and a binary response variable (Bobbit, 2020).

# Part III. Data Preparation

## C1. Data Cleaning

To clean my data in preparation for the linear regression analysis, I set out to do the following tasks:

1. Verify no null or missing data
2. Verify no duplicate records
3. Check for and remove outliers

Code and explanation of methods is in the attached Jupyter notebook, “PredMod-LogReg.ipynb” – see section C1.

## C2. Summary Statistics

Summary statistics given by the built-in pandas describe() method are shown in the attached Jupyter notebook, section C2.

## C3. Visualizations

See attached Jupyter notebook, section C3.

Univariate visualizations are shown as histograms for quantitative variables, and as bar charts (Seaborn countplot()) for categorical variables. Bivariate visualizations are shown as box plots for quantitative variables and crosstab heatmaps for categorical variables.

For those explanatory variables with large right-skew (Population and Income), I also show visualizations of their log transformation.

## C4. Data Transformation

I have two goals in further transforming the data prior to running the regression analysis. First, I need to transform all categorical values into numeric data. Yes/No values can easily be encoded as 1/0. Complication\_risk is ordinal and can be encoded as 1/2/3 for Low/Medium/High. Marital is nominal data, not ordinal, so needs to be transformed to a one-hot encoding. Code and references are given in the attached Jupyter notebook, section C4.

## C5. Prepared Data Set

The cleaned and transformed data set used for the regression analysis is attached as ‘clean\_xform\_medical\_data.csv’.

# Part IV. Model Comparison & Analysis

## D1. Initial Model

See attached Jupyter notebook, section D1. I created a “kitchen sink” model with the entire dataset prepared in Part III. I checked for multicollinearity with code provided by Dr. Sewell, which does show features with very high variance inflation factors, but there are no apparent strong pairwise correlations shown in the correlation matrix or heatmap.

## D2. Justification of Model Reduction

In PA1, I chose to use Backward Stepwise Elimination to reduce the feature set of my initial model. For PA2, I have chosen instead to use Forward Stepwise Variable Selection. I chose to select 3 variables to add, though in practice only one (Arthritis) would meet the threshold for statistical significance. I start with only the constant column and “an empty variable set and proceed[] in steps, where in each step the next best variable is added” (Verbeist, n.d.). The “next best” variable is chosen by trying to refit the regression model with each variable independently and choosing the model with the best AUC (Area Under the Curve) score. AUC is a “measure often used to quantify the performance of predictive models” (Verbeist, n.d.). See attached Jupyter notebook, section D2.

## D3. Reduced Linear Regression Model

See attached Jupyter notebook, section D3. Three explanatory variables remain in the model – Age, Arthritis, and Marital\_Separated. These variables are not correlated with each other and satisfy the non-multicollinearity requirement of logistic regression.

## E1. Model Comparison

My original “kitchen sink” model had 21 potential explanatory variables, including the 4 dummy one-hot variables for the Marital column. In the summary stats, the pseudo R-squared was 0.002, which indicates that the model is poor (Middleton, 2022). The LLR p-value of 0.835 is likewise not close to statistical significance at an alpha level of 0.05.

My reduced model contains only three explanatory variables. The adjusted R-squared value is slightly worse at 0.001, but the LLR p-value has improved to 0.113. This still fails to clear the statistical significance bar.

## E2. Output and Calculations

Refer to section E2 of the attached Jupyter notebook for the confusion matrix. The model predicts ‘No stroke’ for all records in the reduced model. Therefore, there are no true positives or false positives, only true negatives and false negatives. The accuracy is calculated as 80.2%, which is equal to the proportion of patients in the data set who did not report a stroke.

## E3. Code

Refer to attached Jupyter notebook.

# Part V. Data Summary & Implications

## F1. Results

Regression equation for the reduced model:

This means that, holding all else equal, the log odds of a patient having a stroke decrease by ~0.1 if the patient has arthritis, that they increase by 0.0015 for each year of age, and that they increase by ~0.07 if the patient is separated.

None of these explanatory variables is statistically significant, given a standard alpha=0.05. Arthritis nearly meets that threshold at a p-value of 0.066.

Taken as a whole, the model shows is not statistically significant due to the LLR p-value of 0.113. Practically, the model has no more predictive value than just predicting no one will ever have a stroke.

## F2. Recommendations

For future analysis, I would recommend additional data collection. It could be that other medical or lifestyle data could have a stronger correlation with stroke risk. The current data set does not lend itself to predicting stroke risk based on the logistic regression performed above.

# Part VI. Demonstration & Supporting Documentation

## G. Demonstration Video

## A video describing my methods and code can be found at: <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=7c42604e-18d6-46c6-b34c-b0b3016b4739>

## H. Third-party Code Sources

Chouinard, J. (2023, September 25). *Confusion Matrix in Python (Scikit-learn Example).* <https://www.jcchouinard.com/confusion-matrix-in-scikit-learn/>

Kassies, R. (2013, May 6). *Make more than one chart in same IPython Notebook cell.* StackOverflow. <https://stackoverflow.com/questions/16392921/make-more-than-one-chart-in-same-ipython-notebook-cell>

Kumar, A. (2022, March 22). *A Quick Guide to Bivariate Analysis in Python.* Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2022/02/a-quick-guide-to-bivariate-analysis-in-python/>

Sudheer, S. (2023, July 21). *12 Univariate Data Visualizations With Illustrations in Python.* Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2020/07/univariate-analysis-visualization-with-illustrations-in-python/>

Verbeist, N. (n.d.). Lesson 2: Forward stepwise variable selection for logistic regression. *Introduction to Predictive Analytics in Python* [MOOC]. DataCamp. <https://campus.datacamp.com/courses/introduction-to-predictive-analytics-in-python/forward-stepwise-variable-selection-for-logistic-regression?ex=6>

## I. References

Bobbit, Z. (October 13, 2020). *The 6 Assumptions of Logistic Regression.* Statology. <https://www.statology.org/assumptions-of-logistic-regression/>

Larose, C., & Larose, D. (2019). *Data Science Using Python and R.* Wiley.

Middleton, K. (November 2022). *Getting Started with D208, Part I.* Western Governors University. <https://westerngovernorsuniversity.sharepoint.com/:b:/r/sites/DataScienceTeam/Shared%20Documents/Graduate%20Team/D208/Student%20Facing%20Resources/Dr.%20Middleton%20Getting%20Started%20with%20D208(Part%20I)COIT.pdf?csf=1&web=1&e=CLRFMI>

*pandas.DataFrame.describe — pandas 2.1.2 documentation*. (n.d.). Pandas.pydata.org. <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.describe.html>

Ragan, A. (October 10, 2018). *Taking the Confusion Out of Confusion Matrices.* Medium – Towards Data Science. <https://towardsdatascience.com/taking-the-confusion-out-of-confusion-matrices-c1ce054b3d3e>