CLASSIFICATION ANALYSIS – D209

Gooden, Nina S. Gooden [ID #: 009823504]

Dr. Eric Straw

CLASSIFICATION ANALYSIS – D209

This analysis will explore the medical data of a theoretical real-world organization. By creating this logistic regression, I will address the business’ concern about readmission in a data-first evaluation. I will also provide visualizations to support my assessments, in tandem with code for my models. As previously requested, I will also discuss the limitations and potential course of action this data supports.

## **A. The Research Question:**

1. Research Question
   1. Using the *k*-nearest neighbor method, can we identify patients likely to contribute to readmission based on comorbidity factors such as age, gender, and noted health concerns?
2. Data Analysis Goals and Objective
   1. I will be reassessing the model I created in D208, with additional tools provided from this course. This evaluation is to further explore variables that were identified as having correlation with readmission. In this evaluation, we will look at variables that have been identified as patient notes—age, sex, and reported comorbidities—through a logistic regression in order to measure likelihood of readmission given said variables.

## **B. Justification for Method**

1. How KNN analyses the selected data set and hypothesized outcome:
   * 1. KNN finds the distance between my given parameter and all the available data, then selects the variables (k) closest to the query. Next, for classification, it votes for the most frequent label, as long as they are continuous.
     2. Once an algorithm is established, it can be trained and matched to the outcome of the primary variable—in this case, Yes/No for readmission. From this training step we gain a model accuracy score.
     3. It is my expectation that this classification will have an accuracy **of less than 95%. This is due to the previous assessments from D208.** My goal for this task is to glean insights as to which variables have value in the comorbidity question. Classification is a good fit for this question. I do not expect under- or overfitting.
2. KNN assumption summary:
   1. KNN assumes that each training data consists of variables and class levels that are associated with one another. “In the simplest case, it will be either + or – (for positive or negative classes).” (*A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm*, 2010)
3. Packages and libraries:
   1. I’ve opted to use Python for my analysis as this is the language, I am most comfortable with. Python offers packages and libraries that make visualizations and analysis easy and straightforward. I will be using pandas, Numpy, Matplotlib, and Seaborn in this evaluation, all of which are uniquely designed for data analysis and visualization. In addition, I will also be using scikit-learn, for splitting, training, testing, and fitting my data. This package is uniquely suited for predicting and classifying data in measurable models.

## **C. Data Preperation Process**

1. Data Preparation Goals and Manipulations
   1. I began my analysis using my standard method of:
      1. Dropping irrelevant columns
      2. Reordering the remaining column so that my target variable is first
      3. Use .isna() and .any to see if there are any missing values
      4. Check and group column types with .info and select\_dtypes()

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

1. Initial data set variables:
   1. Even though the *k-*neighbors evaluation looks at categorical variables, I kept the two continuous variables as placeholders to ensure my evaluations were on the right track—that I hadn’t messed up any of my scrubbing steps.
      1. Target variable: 'ReAdmis'
      2. Categorical/Predictor variables: 'Gender', 'HighBlood', 'Stroke', 'Complication\_risk', 'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma'
      3. Continuous/Other variables (sanity checks): 'Age', 'VitD\_levels'

A screenshot of a computer

Description automatically generated with medium confidence

1. Data Preparation:
   1. Steps and code for initial data cleaning have been provided via screenshots in C1 and C2.
   2. I went through the standard tests and checks for this data set as well, including:
      1. Checking the distribution for the target variable.

Graphical user interface

Description automatically generated

* + 1. Running bi- and univariate checks, as well as visualizations

|  |  |
| --- | --- |
| Chart, histogram  Description automatically generated | Chart, histogram  Description automatically generated |

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Diagram, polygon

Description automatically generated

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| Chart, bar chart  Description automatically generated | Chart, bar chart  Description automatically generated |
| Chart, bar chart  Description automatically generated | Chart, bar chart  Description automatically generated |
| Chart, bar chart  Description automatically generated | Chart, bar chart  Description automatically generated |
| Chart, bar chart  Description automatically generated | Chart, bar chart  Description automatically generated |
| **Chart, bar chart  Description automatically generated** | Chart, bar chart  Description automatically generated |
| Chart, bar chart  Description automatically generated | Chart, bar chart  Description automatically generated |
| Chart, bar chart  Description automatically generated |  |

* + 1. Converting the categorical variables to numerical ones in order to run evaluations.

A screenshot of a computer

Description automatically generated

* + 1. Creating scatterplots for the unchecked variables—while also looking for outliers

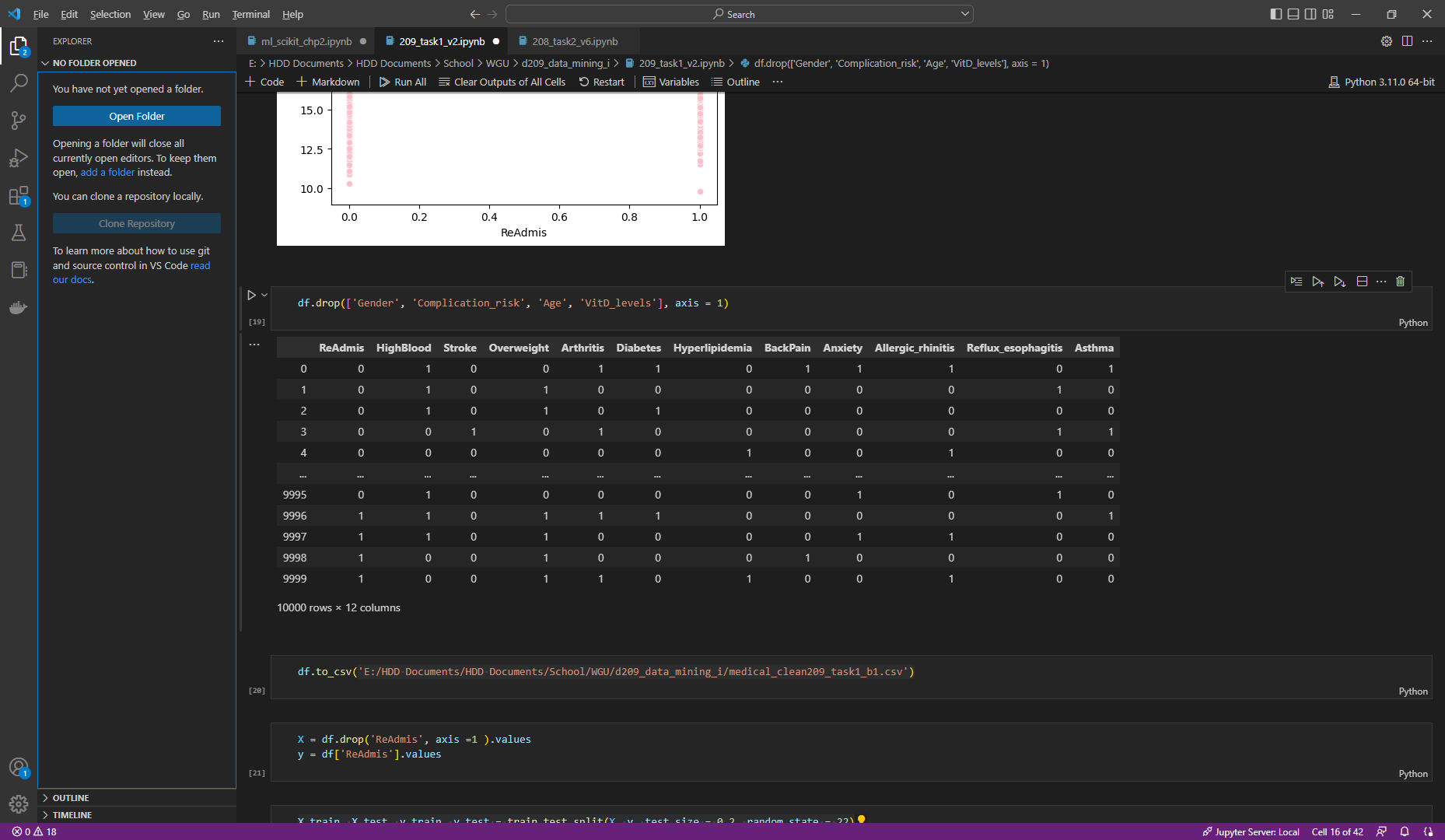
A screenshot of a computer

Description automatically generated

Graphical user interface

Description automatically generated

* + 1. After reviewing the analysis, I decided to drop 4 additional columns. I was ready to start my model.



1. Provide cleaned data set:
   1. Data provided.

A screenshot of a computer

Description automatically generated

## **D. MODEL COMP AND ANALYSIS**

1. Data analysis and report:
   1. I first created the X and y values by removing and creating exclusive frames with ReAdmis.

A screenshot of a computer

Description automatically generated

* 1. Next, I created a training and test set of the data using a 3:7 ratio for test and training respectively. I also set a seed value so I could replicate this model.

A screenshot of a computer

Description automatically generated

* 1. I exported the files, per the assignment.

A screenshot of a computer

Description automatically generated

* 1. Due to the work I did in D208 with almost the same data set, I wasn’t worried about underfitting my data. As such, I set the n\_neighbors to 14, telling the model to look at the nearest 14 values fot make the classification. I ran the .fit() method to create the prediction.

A screenshot of a computer

Description automatically generated

1. Analysis technique:
   1. I used the .accuracy\_score() function to check the initial accuracy for the test. It came back at a little less than 61%.

A screenshot of a computer

Description automatically generated

* 1. I also printed the classification report, so I would be able to compare the stats with the trained model.

A screenshot of a computer

Description automatically generated

* 1. Finally, I created a confusion matric so that I could see the data a little better.

Graphical user interface, application

Description automatically generated

1. I scaled the data using the Pipeline() function to see if it would turn over a similar result. The data was split and tested with the same parameters.

Graphical user interface, text, application

Description automatically generated

1. The accuracy score went down to 58% after scaling. In addition, there was a significant jump in False results.

A screenshot of a computer

Description automatically generated

1. I also created a confusion matrix for this result.

Graphical user interface

Description automatically generated

1. Ultimately, scaling the model made it perform worse. Both accuracy and percision suffered from the shift.
2. All code has been provided with detailed steps.

## **E. Data Set Analysis**

1. Explaining the accuracy and AUC of the model:
   1. I used cross validation to find the best parameters for the *k-*neighbors model while looking at the training data. I could have got the model as high as 63%, had I used n-neighbors = 3.

Graphical user interface, text

Description automatically generated

* 1. I calculated the Area Under Curve and landed on a score of .585, which means the model has predictions which are more often correct than incorrect. According to, the .59 is just shy of a successful model *(Bhandari, 2022)*. The 5-fold cross validation method returned the following result: 0.50233711, 0.48902818, 0.49008549, 0.49382455, 0.50025989

Graphical user interface, text, application

Description automatically generated

1. Discuss the results and implications of the analysis:
   1. The model, at its very best, only has an accuracy score of 63%. My initial model was close to the best-case scenario and has a score of .74. While that is not a complete failure, it is far from the best. This model shows that it is able to correctly predict whether or not a patient will be readmitted 63% of the time. The model does not have a strong classifier, based on the data and does not produce high-confidence results.
2. Data analysis limitations:
   1. The knn is a simple, easy to understand algorithm but it is also chunky. While running my analysis VSC ran slowly. As such, as more information is added to the data set, the slower and choppier analysis will be.
   2. In addition, with knn, the initial k-values are chosen at semirandom, which requires additional testing in order to avoid under- and over-fitting.
   3. knn is sensitive to outlier and missing data.
3. Recommendations:
   1. This analysis should be rerun with a n-neighbors value of 46 to examine the best possible parameters.
   2. This medical center should spend resources to house data in a more way that better designates comorbidity information for evaluation.
   3. In order to maintain the health of this analysis, the medical center should launch campaigns to gather as much comorbidity data as possible. I would also recommend keeping track of the variation KPIs in order to segment response efforts.
   4. This analysis should be run quarterly in order to ensure the value of this assessment.

## **F. Panopto**

[Data Mining I – NVM2| D209](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=49a2435a-3fe4-44ad-a4be-af5d0016dac7)

## **G. Reference web sources**

*A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm*. (2010, May 18). God, Your Book Is Great !! https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/

Bhandari, A. (2022, June 14). AUC-ROC Curve in Machine Learning Clearly Explained. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

## **H. Acknowledge sources**

sklearn.pipeline.Pipeline. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

T, D. (2021, December 11). Confusion Matrix Visualization - Dennis T. Medium. https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea