Part I: Research question

1. 1. The relevant question I am posing for this task is can we predict a patient’s complication risk based on the information provided?

2. One goal of my analysis is to identify someone as high, medium, or low risk for complications given just the information provided in this data set (age, history of stroke, location, etc).

Part II: Method Justification

1. 1. I chose k-nearest neighbors for my classification method. The knn algorithm classifies objects off of given attributes. The main idea is that like goes with like. It looks for similarities among training data and uses that to sort the testing data into groups (Harrison). In my case, knn will be predicting a patient’s complication risk given the other available data. Due to the scope of the data I expect to be able to predict complication risk with at least 80% accuracy.

2. One assumption of the KNN algorithm is that similar things are in close proximity to each other.

3. The libraries I will be using in this analysis and their benefits are as follows:

* Numpy - I used the numpy library to convert my lists into arrays
* SKlearn preprocessing - I used this library to binarize columns
* SKlearn neighbors - This library includes the k-nearest neighbors algorithm that sorts data based on similarities
* Pandas - I used the pandas library to read my csv
* SKlearn train\_test\_split - I used this package to split the data into X and Y and train and test groups
* SKlearn roc\_auc\_score - I used this package to calculate the area under the curve which judges accuracy for a model
* SKlearn precision score - I used this package to find the precision score which is the ratio of true positives to false positives
* SKlearn confusion matrix - This package allowed me to construct and visualize a confusion matrix

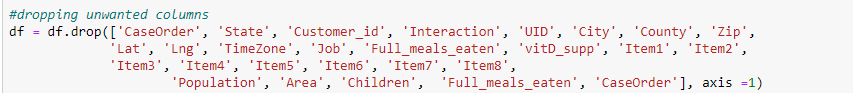
Part III: Data preparation

1. 1. One goal for the preprocessing step is to get dummies for the categorical variables before dropping the original columns.

2. Here are the variables I will be including in my first model:

* Age - continuous
* Income - continuous
* ReAdmis - categorical
* VitD\_levels - continuous
* Doc\_visits - categorical
* Soft\_drink - categorical
* HighBlood - categorical
* Stroke - categorical
* Overweight - categorical
* Arthritis - categorical
* Diabetes - categorical
* Hyperlipidemia - categorical
* BackPain - categorical
* Anxiety - categorical
* Allergic\_rhinitis - categorical
* Reflux\_esophagitis - categorical
* Asthma - categorical
* Initial\_days - continuous
* TotalCharge - continuous
* Additional\_charges - continuous
* Marital - categorica
* Gender - categorical
* Initial\_admin - categorical
* Complication\_risk - categorical
* Services - categorical

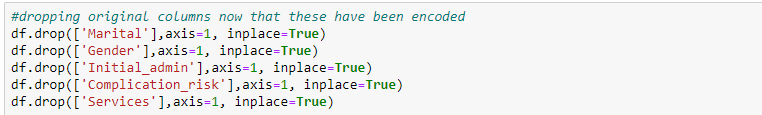
3. The first step I took for data preparation was to drop the columns I know I will not be using.



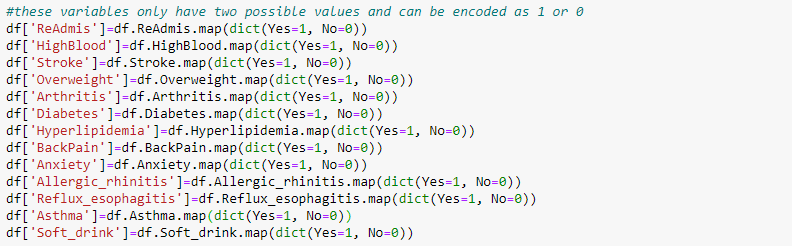
Next I got dummies for the categorical variables.



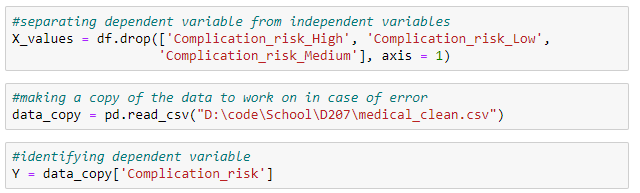
After that I dropped the original columns that have now been hot encoded.



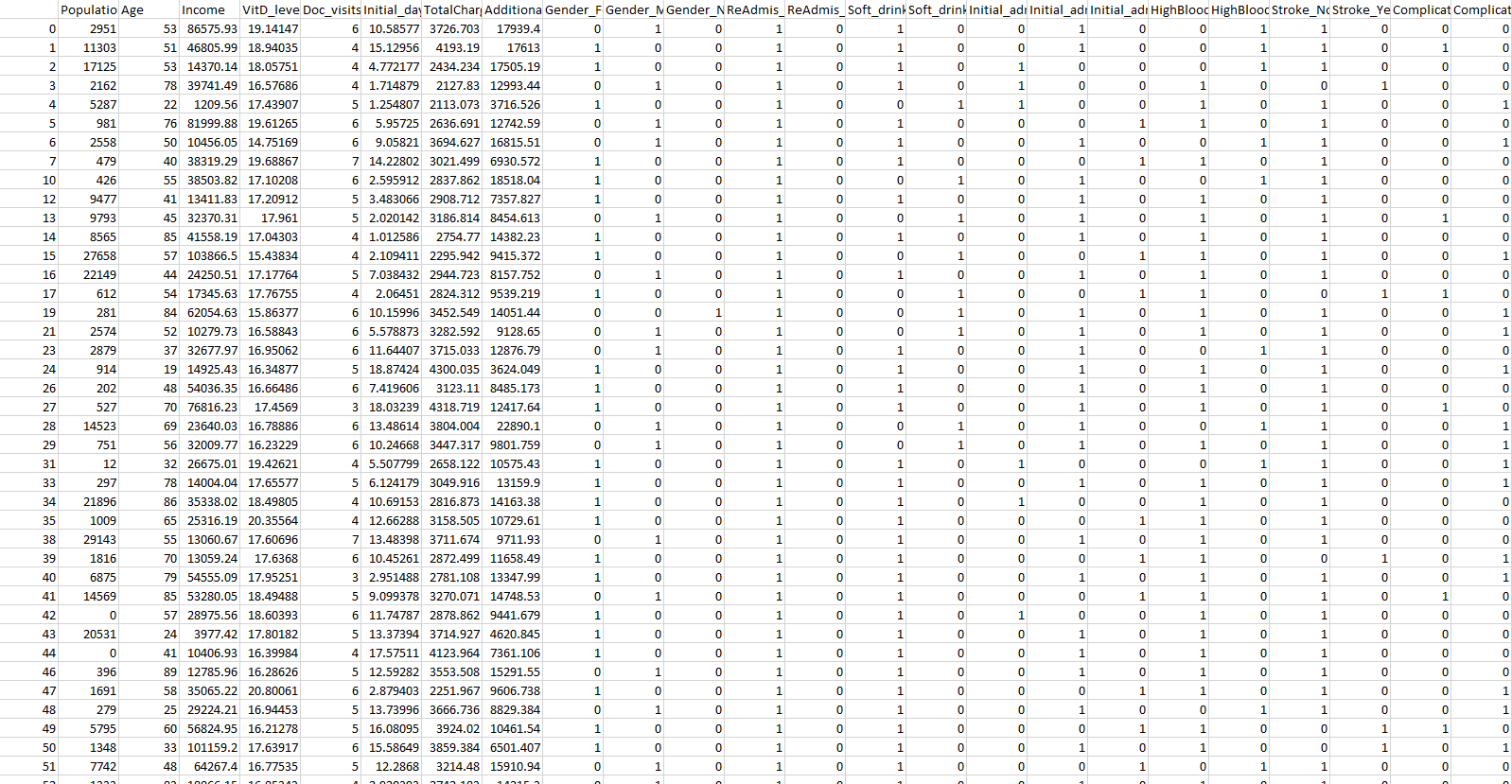
Then I encoded the binary variables to set 1 = Yes and No = 0.



Next I separated the dependent variable from the independent variables and made a copy of the data to work on in case of error.



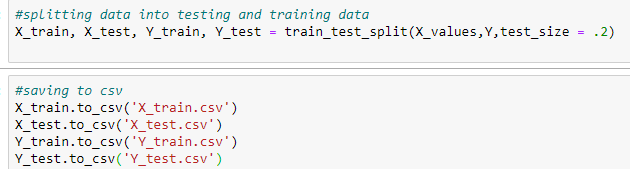
4. Here is a snippet of the cleaned dataset included in my project.



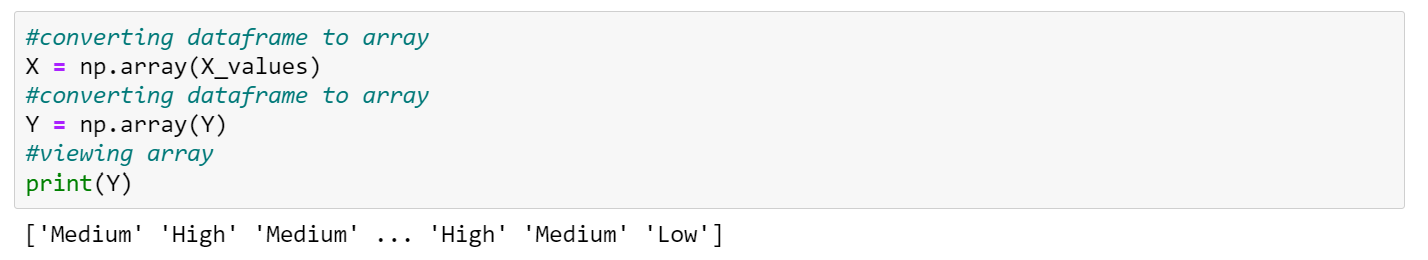
IV:Analysis

D.1.

Below, I split the data into testing and training data. The files are included in this submission.



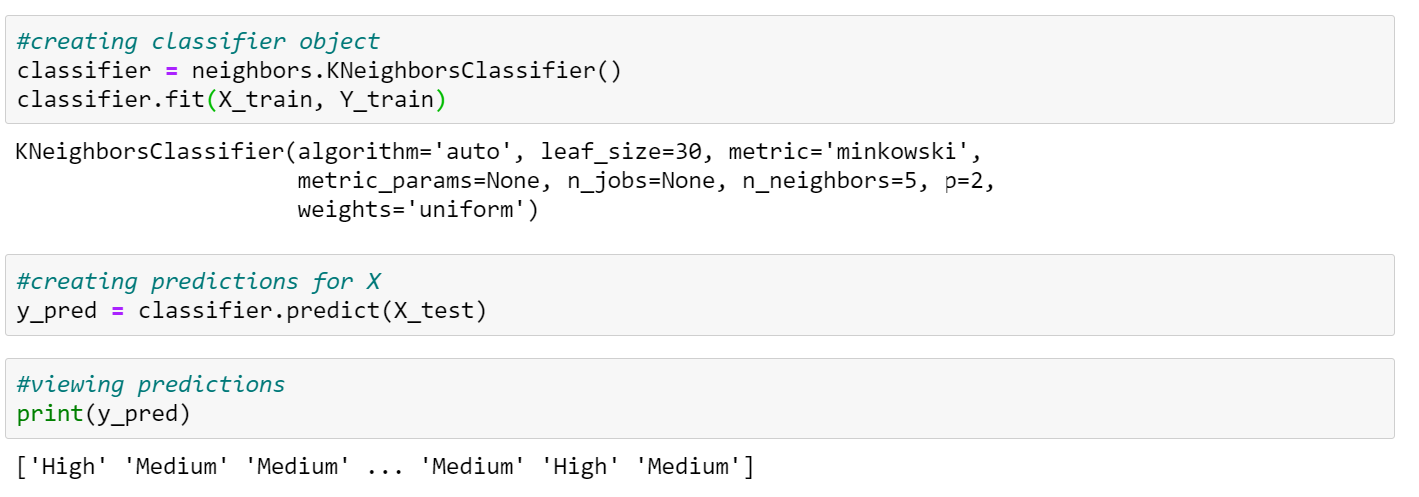
2. The analysis type I used on this data is the k-nearest-neighbors algorithm. Once my data was cleaned and split, I began by identifying the X and Y variables. Next I converted them to arrays to be used in the function.



Next I imported and used StandardScaler from SKlearn. This standardizes features by subtracting the mean and then dividing all values by the standard deviation.



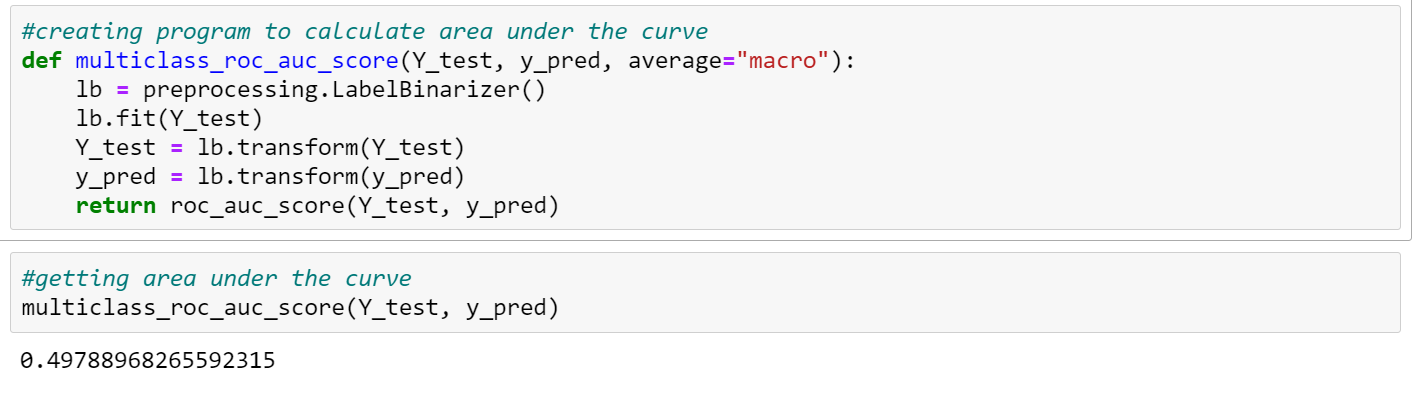
After that I created a knn object and fit it to the data. I then used it to predict Y values based off of the training data.



After that I created a confusion matrix to view true and false positive and negative counts.



Last but not least I created a program to calculate the area under the curve for my model.



3. The code used to perform this classification analysis is included in the .ipynb file of this submission.

Part V: Data Summary and Implications

E.1. My model is not very accurate. For the accuracy, I got a precision score of 0.34547. Since this equals about a third of the data, we see that this model is not much more accurate than selecting with random chance. For the Area Under the Curve I got 0.497. This again means that the model is not making any more distinctions than chance itself would.

2. The result of my analysis is that we cannot accurately predict a person’s complication risk given the information provided. The implication is that we need more data to be able to do this.

3. One major limitation of my analysis is the scope of the data. While this dataset has a significant amount of information, a lot of it is non medical. There are many conditions (e.g. cancer, Parkinson’s, autoimmune disease, etc) that are unaccounted for in this model that could drastically improve accuracy.

4. I recommend collecting more medical data as allowed by HIPPA. Hospitals could use this risk analysis to identify patients that may require extra care if it was more accurate.

G. Third party code referenced

Euskalduana. (2015, July 9). *Scikit-learn: How to obtain True Positive, True Negative, False Positive and False Negative*. Stack Overflow. https://stackoverflow.com/questions/31324218/scikit-learn-how-to-obtain-true-positive-true-negative-false-positive-and-fal.

sklearn. (n.d.). *Confusion\_matrix*. scikit. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\_matrix.html.

H. Works Cited

Harrison, O. (2019, July 14). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Medium. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.