**Part I: Research Question**

Question: *Can we predict which patients are at high risk of readmission?* We can answer this using k-nearest neighbor (KNN).

The objectives of this data analysis are to learn how we can predict the probability of whether a patient will be readmitted. Thus, if we know which patients are at higher risk of readmission, we can adjust our care accordingly and plan as such. This will allow us to be more prepared, avoid penalties for readmissions, and hopefully, provide better overall healthcare to patients.

**Part II: Method Justification**

The KNN method analyzes this dataset by predicting a datapoint’s label for ReAdmis (Yes/No) based on several independent variables. These independent variables effectively classify this datapoint based on its proximity to the other datapoints in the dataset. Ultimately, the nearest “k” number of datapoints “vote” on the datapoint’s label based on those labels (of whether ReAdmis is Yes or No). This method is appropriate because the question we are trying to answer focuses on predicting that ReAdmis label.

Expected outcomes of the first 15 patients (where 0 = Not readmitted and 1 = readmitted):

array([0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0], dtype=uint8)

The KNN method centers on the assumption that essentially, “similar things exist in close proximity” (Harrison, 2018). In other words, data points that are similar to each other are positioned close to each other. In this scenario, this means that patients with similar characteristics also have other similar characteristics.

**Python packages used to support analysis:**

**import pandas as pd:** *Reads in the data.*

**import seaborn as sb:** *Used for the confusion matrix heatmap.*

**from sklearn.metrics import confusion\_matrix, precision\_score, recall\_score, accuracy\_score:** *Used to calculate the precision score, recall score, and accuracy score, as well as generate the confusion matrix.*

**from sklearn.model\_selection import train\_test\_split:** *Used to split the dataset into training and testing sets.*

**from sklearn.metrics import classification\_report:** *Used to generate the classification report*

**from sklearn.neighbors import KNeighborsClassifier:** *To fit the classifier, as well as train the data and predict our outcomes.*

**from sklearn.metrics import roc\_auc\_score:** *Used to calculate the AUC score*

**from sklearn.preprocessing import StandardScaler:** *Used to scale the data*

**from sklearn.pipeline import Pipeline:** *Used to put the scaler in a pipeline*

**Part III: Data Preparation**

Prior to processing this data, I know I will need to make some manipulations – particularly, I will need to convert all categorical variables to numeric variables (using dummies), as well as drop any variables I’m not going to use. Another important preprocessing step will be to scale the data, since the ranges of our variables will largely vary. The below table shows the different scales the original data is on. We want to standardize this data so all features are on similar scales. To do this, we will use the StandardScaler and Pipeline packages in sklearn.

Summary statistics for all variables (prior to scaling) are below. The first table shows continuous variables, while the following table show categorical variables.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **Lat** | 10000 | 38.75 | 5.40 | 17.97 | 35.26 | 39.42 | 42.04 | 70.56 |
| **Lng** | 10000 | -91.24 | 15.21 | -174.21 | -97.35 | -88.40 | -80.44 | -65.29 |
| **Population** | 10000 | 9965.25 | 14824.76 | 0 | 694.75 | 2769 | 13945 | 122814 |
| **Children** | 10000 | 2.10 | 2.16 | 0 | 0 | 1 | 3 | 10 |
| **Age** | 10000 | 53.51 | 20.64 | 18 | 36 | 53 | 71 | 89 |
| **Income** | 10000 | 40490.50 | 28521.15 | 154.08 | 19598.78 | 33768.42 | 54296.4 | 207249 |
| **VitD\_levels** | 10000 | 17.96 | 2.02 | 9.81 | 16.63 | 17.95 | 19.35 | 26.39 |
| **Doc\_visits** | 10000 | 5.01 | 1.05 | 1 | 4 | 5 | 6 | 9 |
| **Full\_meals\_eaten** | 10000 | 1.00 | 1.01 | 0 | 0 | 1 | 2 | 7 |
| **vitD\_supp** | 10000 | 0.40 | 0.63 | 0 | 0 | 0 | 1 | 5 |
| **Initial\_days** | 10000 | 34.46 | 26.31 | 1.00 | 7.90 | 35.84 | 61.16 | 71.98 |
| **Additional\_charges** | 10000 | 12934.53 | 6542.60 | 3125.7 | 7986.488 | 11573.98 | 15626.49 | 30566.1 |
| **Timely\_admission** | 10000 | 3.52 | 1.03 | 1 | 3 | 4 | 4 | 8 |
| **Timely\_treatment** | 10000 | 3.51 | 1.03 | 1 | 3 | 3 | 4 | 7 |
| **Timely\_visits** | 10000 | 3.51 | 1.03 | 1 | 3 | 4 | 4 | 8 |
| **Reliability** | 10000 | 3.52 | 1.04 | 1 | 3 | 4 | 4 | 7 |
| **Options** | 10000 | 3.50 | 1.03 | 1 | 3 | 3 | 4 | 7 |
| **Hours\_of\_treatment** | 10000 | 3.52 | 1.03 | 1 | 3 | 4 | 4 | 7 |
| **Courteous\_staff** | 10000 | 3.49 | 1.02 | 1 | 3 | 3 | 4 | 7 |
| **Active\_listening** | 10000 | 3.51 | 1.04 | 1 | 3 | 3 | 4 | 7 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **count** | **unique** | **top** | **freq** |
| **Area** | 10000 | 3 | Rural | 3369 |
| **Marital** | 10000 | 5 | Widowed | 2045 |
| **Gender** | 10000 | 3 | Female | 5018 |
| **Soft\_drink** | 10000 | 2 | No | 7425 |
| **Initial\_admin** | 10000 | 3 | Emergency Admission | 5060 |
| **HighBlood** | 10000 | 2 | No | 5910 |
| **Stroke** | 10000 | 2 | No | 8007 |
| **Overweight** | 10000 | 2 | Yes | 7094 |
| **Arthritis** | 10000 | 2 | No | 6426 |
| **Diabetes** | 10000 | 2 | No | 7262 |
| **Hyperlipidemia** | 10000 | 2 | No | 6628 |
| **Anxiety** | 10000 | 2 | No | 6785 |
| **Allergic\_rhinitis** | 10000 | 2 | No | 6059 |
| **Reflux\_esophagitis** | 10000 | 2 | No | 5865 |
| **Asthma** | 10000 | 2 | No | 7107 |
| **Services** | 10000 | 4 | Blood Work | 5265 |
| **Complication\_risk** | 10000 | 3 | Medium | 4517 |
| **BackPain** | 10000 | 2 | No | 5886 |
| **ReAdmis** | 10000 | 2 | No | 6331 |

My data preparation goals begin with confirming the data is complete and accurate (i.e. clean), and if not, cleaning it as necessary. I will reduce the data to only the columns that I plan to use, as some columns will be irrelevant, and others will have too many categories to work with. Since I know I’ll be working with several categorical values, I will need to convert all of those to dummy variables so they will work with the model. This will create several additional columns - I will also need to remove the original converted column.

Cleaned dataset is attached separately.

**Part IV: Analysis**

I used the k-nearest neighbor (KNN) technique for this analysis. The KNN technique takes the nearest neighboring datapoints and uses them as references to predict a label. Here, the label that is being predicted is whether a patient will be readmitted or not.

Files for split datasets, as well as code for classification analysis are attached separately.

**Part V: Data Summary and Implications**

Our AUC (area under the ROC curve) is equal to approximately 0.859. The accuracy score is 0.7715 (which means our method is accurate about 77% of the time) and the precision score is about 0.8244.

precision recall f1-score support

0 0.76 0.94 0.84 1266

1 0.82 0.48 0.61 734

accuracy 0.77 2000

macro avg 0.79 0.71 0.72 2000

weighted avg 0.78 0.77 0.75 2000

The classification report above, coupled with the AUC of 0.859 tells us that our classification method is accurate to a high degree. In general, an AUC of over 0.85 is considered “high” classification accuracy, while a moderate accuracy is an AUC of 0.75 to 0.85 and low accuracy corresponds with an AUC of under 0.75 (Zhou, 2019).

The limitations of this data analysis are significant. I have considerable skepticism about the accuracy of this data, and the accuracy of this method is only as accurate as the data itself. Furthermore, there is no way to validate the data to confirm. It is also important that we do not mistake correlation for causation, and that we understand that just because these variables are correlated with readmission status does not mean they are the cause. Also, the fact that these have been trends in the past doesn’t automatically imply they will be trends in the future.

My recommendation at this point would be to first investigate this data to be sure it is accurate. If the data is indeed accurate, we should begin using this classification method to predict whether a patient will be readmitted. Despite the limitations outlined in the paragraph above, this method’s accuracy of above 75% can be a useful tool. Although it can’t be used as our only prediction method – the doctors and nurses must also be involved in that prediction – we have a solid starting point that can help us forecast from a statistical point of view.

**References**

Harrison, O. (2018, September 18). *Machine learning basics with the K-nearest neighbors algorithm*. Medium. Retrieved October 5, 2021, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

Zhou, X. (2019, May 14). *Receiver Operating Characteristic (ROC) Area Under the Curve (AUC): A Diagnostic Measure for Evaluating the Accuracy of Predictors of Education Outcomes*. Education Leadership Data Analytics Research Group. Retrieved October 12, 2021, from https://www.tc.columbia.edu/elda/blog/content/receiver-operating-characteristic-roc-area-under-the-curve-auc/.