**D209 Performance Assessment I**

Renzo Espinoza

College of Information Technology, Western Governors University

D209: Data Mining

Dr. Festus Elleh

October 14, 2022

**A1: Proposal of Question**

The research question that I will answer using k-nearest-neighbor is: can one predict which hospital patients are at risk of being readmitted within a month of their release?

**A2: Defined Goal**

Readmission of patients is such an issue for hospitals that a third-party hands out fines for excessive readmissions. For this scenario,the hospital chain that I am performing the analysis for is interested to know whether they have a readmission problem, likely due to the potential financial repercussions that comes with it. The goal of the analysis is to identify patients with a high risk of being readmitted within a month of their release, allowing the hospitals to implement whatever measures are needed to minimize the odds of readmission. The dataset contains many features of patients that I could potentially use as predictors.

**B1: Explanation of Classification Method**

The classification method I chose, KNN, analyzes the data set by predicting whether a new data point will have a ‘Re\_admis’ value of 1 or 0 based on which value its nearest neighbors have.

**B2: Summary of Method Assumption**

According to Bruce et al. (2020), one assumption of KNN is that all of the features must be numerical.

**B3: Packages or Libraries List**

|  |  |
| --- | --- |
| **Library** | **Usage** |
| Numpy and Pandas | Data storage and manipulation |
| Matplotlib and Seaborn | Data visualization |
| Scikit-learn | Implement KNN, split data into training and testing sets, feature scaling, and compute ROC and AUC |

**C1: Data Preprocessing**

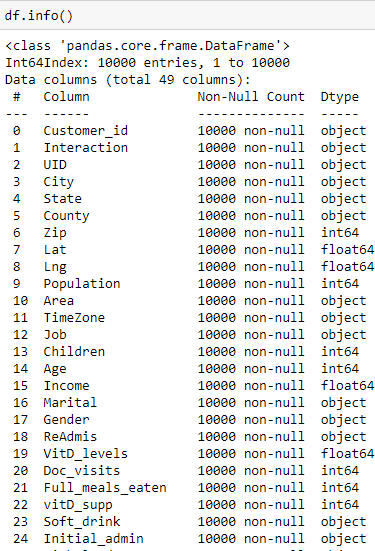
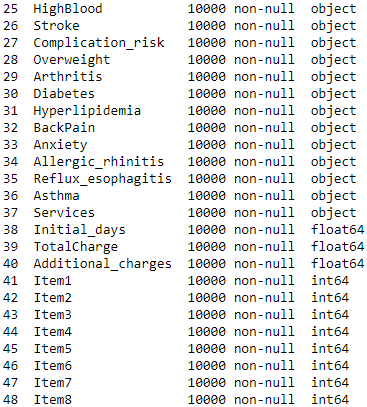
One data preprocessing goal that I have is to encode categorical data. As stated before, an assumption of KNN is that the predictors must be numerical, so by converting these categorical variables into integers, I have more potential features to use.

**C2: Data Set Variables**

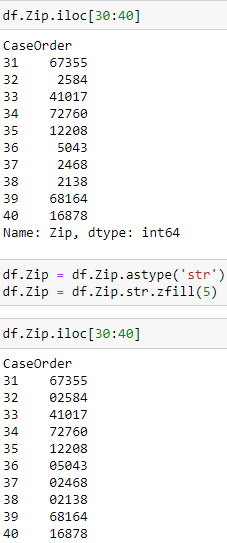
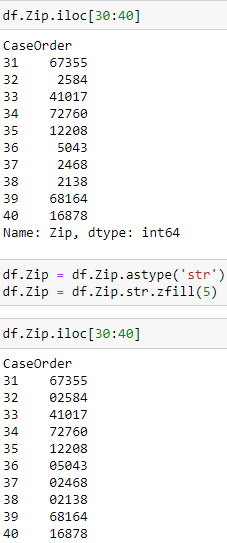
|  |  |
| --- | --- |
| **Variable** | **Type** |
| Re\_admis | Categorical |
| Total\_charge | Continuous |
| Initial\_days | Continuous |

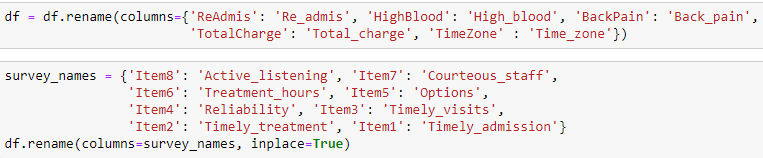
**C3: Steps for Analysis**

To prepare the data for analysis, I started off by using the ‘Dataframe.info’ method from the Pandas library to get a bird’s-eye view of the data that I imported into a Pandas Dataframe.

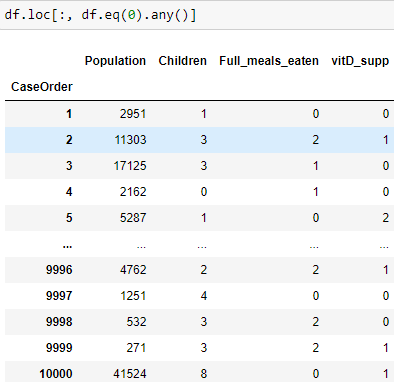


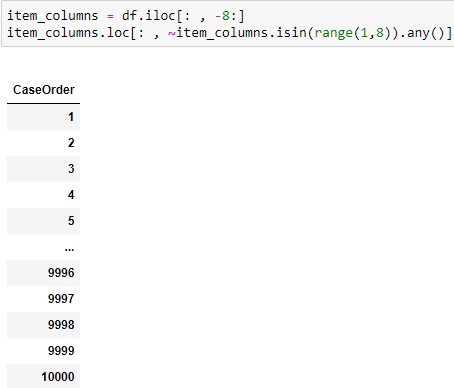
From there I see that the categorical field ‘Zip’ is being imported as an integer. To fix this, I convert the column to String and add the missing leading zeros.

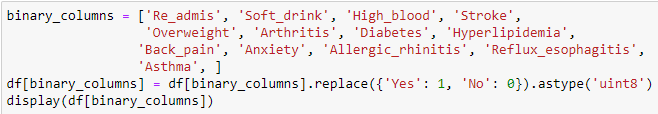


Next, I want to rename certain columns that don’t fit the standard naming conventions of the other columns. I also renamed the survey columns to more easily identify what each survey item pertains to.

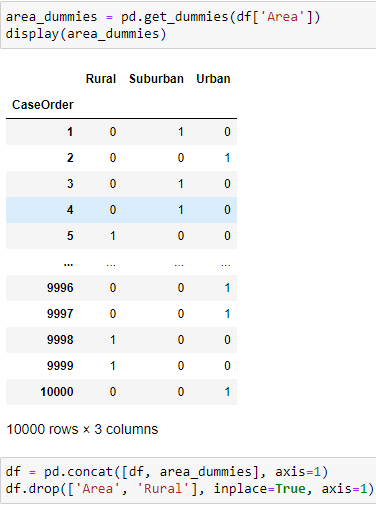
According to the summary of the dataset there are no nulls found in the data, but it is still possible for data to be missing. I queried the dataset for 0’s using the ‘Dataframe.loc’ method and the results showed that there are zero-values inside the ‘Population’ column. The rest of the columns in the results do not imply an input error.



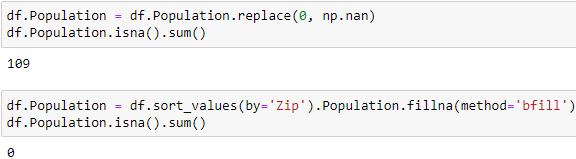
**** I also queried the ‘Item’ columns for values outside of the 1-8 range that the surveys range from.

**** Next I want to convert several categorical binomial variables to integers, represent ‘Yes’ and ‘No’ with 1 and 0, respectively.

I will perform one-hot encoding on suitable categorical variables. I used the ‘get\_dummies’ method from the Pandas library to create the dummy variables, I then concatenate dummy variables to the dataset and drop the original variable and one of the dummy variables to use as my reference group. Below I show the code for one-hot encoding the ‘Area’ column, I repeat this process for ‘Gender’, ‘Initial\_admin’, and ‘Services’.

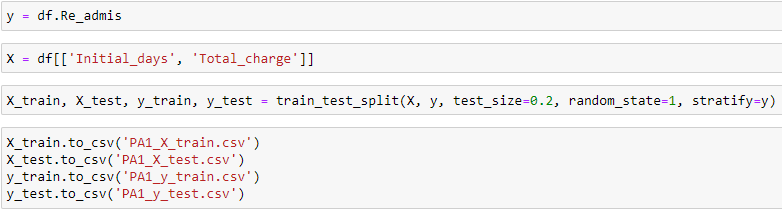


 The ‘Complicatoin\_risk’ variable has three different values: ‘Low’, ‘Medium’, and ‘High’. I replaced these three values with 1, 2, and 3, respectively.

 Finally, I want to impute the 0 values found in the ‘Population’ column. To do this I performed backwards fill after sorting the dataset by the ‘Zip’ column.

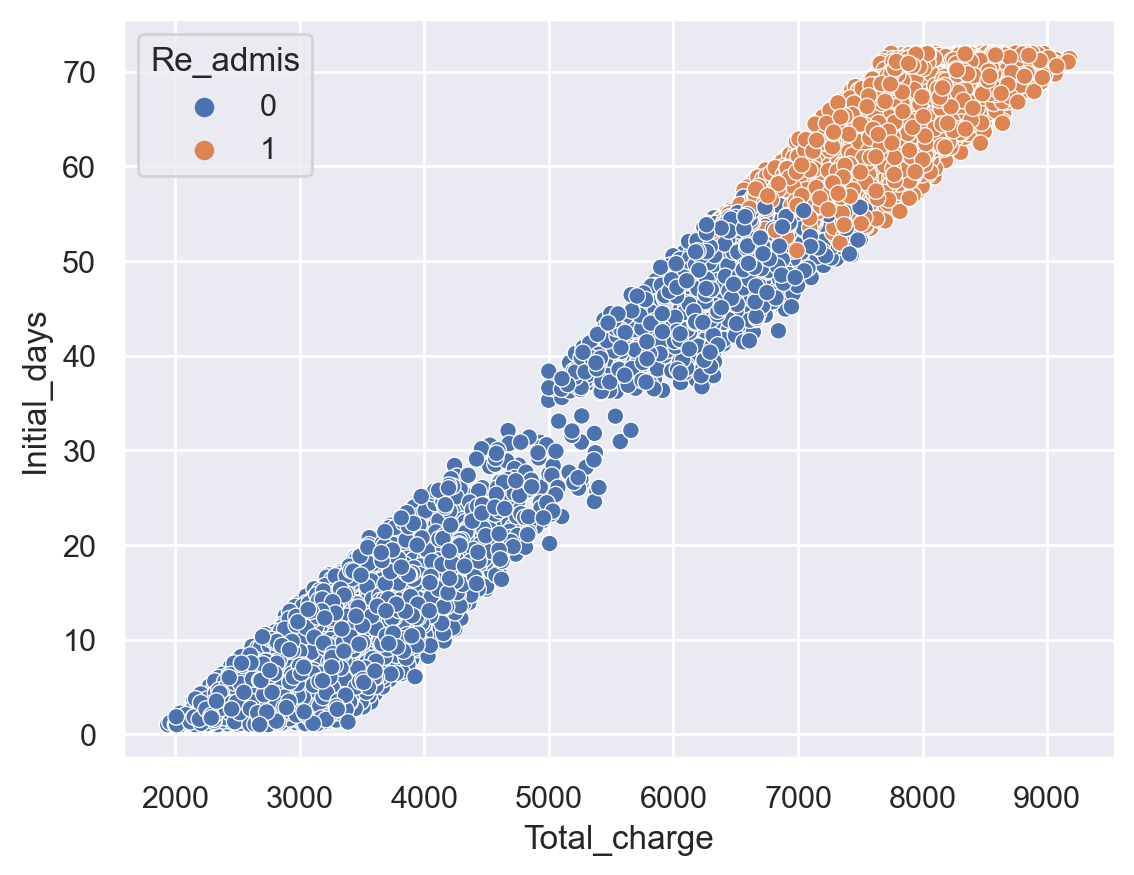
**C4: Cleaned Data Set**

The cleaned dataset will be attached to my submission under the name ‘PA1\_cleaned\_data.csv’.

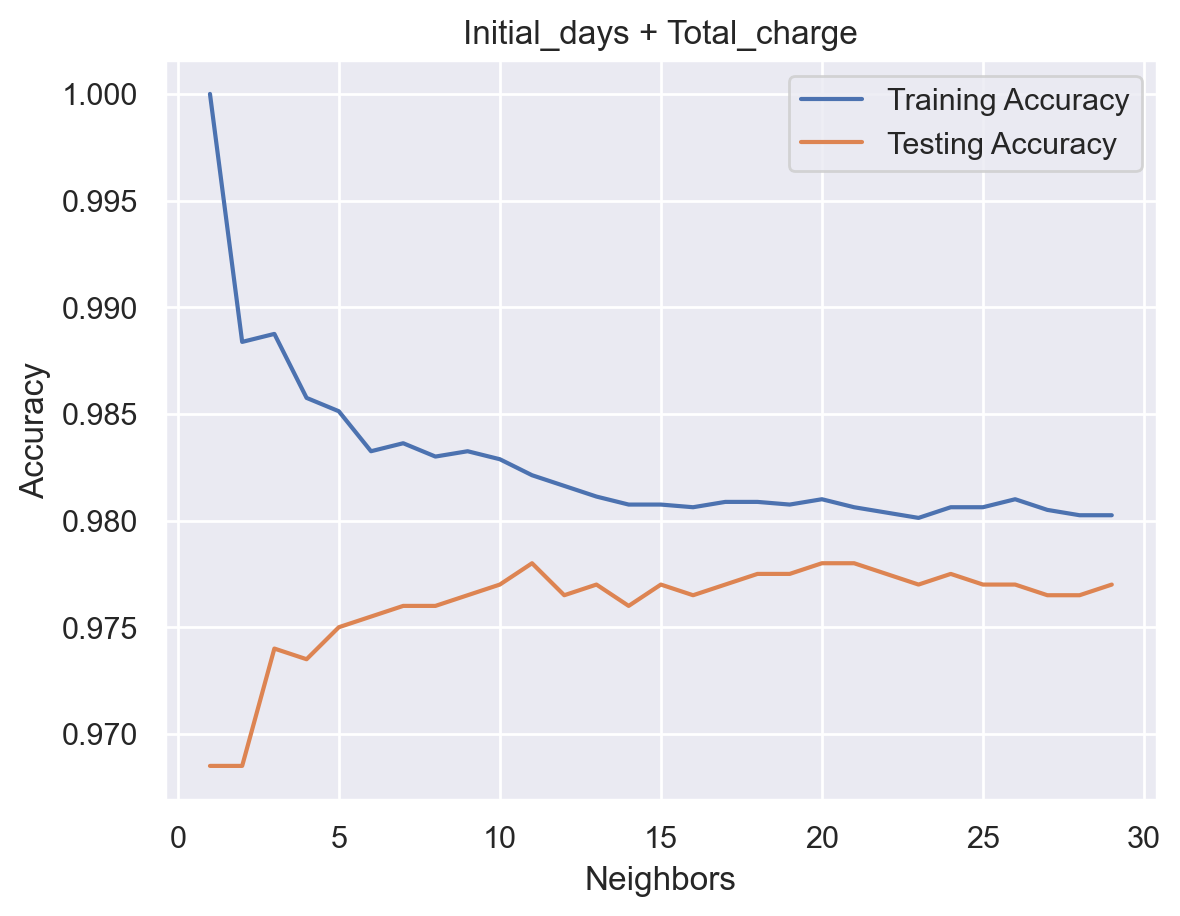
**D1: Splitting the Data**

**D2: Output and Intermediate Calculations**

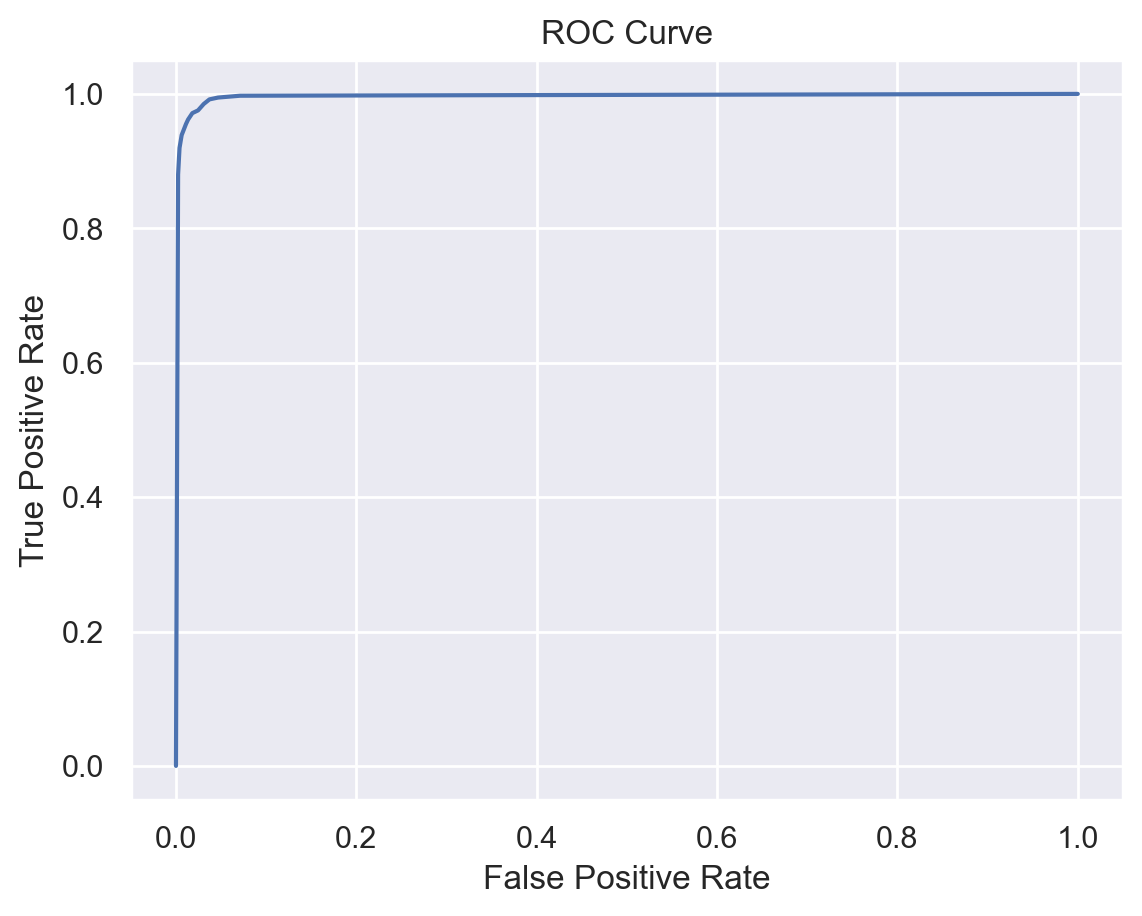
After cleaning the data, I explored it to find relationships with the target variable ‘Re\_admis’. The most promising features are ‘Initial\_days’ and ‘Total\_charge’ because a clear pattern emerges when plotted, showing that patients are much more likely to be readmitted when those features are very high.



Because the values of ‘Total\_charge’ are much larger than ‘Initial\_days’, I used ‘StandardScaler’ from Scikit-learn to scale the features before fitting the ‘KNeighborsClassifier’ onto the data. I chose an arbitrary value of 5 for K. The classifier performed fairly well with an accuracy score of 0.975, but there could still be room for improvement.

 To calculate the best performing value for K, I plotted the training and testing accuracy score for several neighbor values ranging from 1 to 30. The resulting model complexity curve shows that the classifier has the highest accuracy score when K is equal to 11.

Finally, I used the ‘roc\_curve’ method from the Scikit-learn library to compute the ROC for my classifier and the ‘auc’ method to compute the area under the ROC curve.

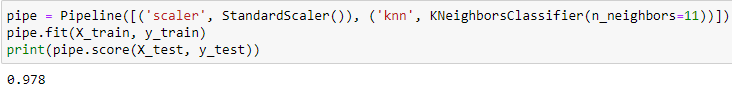


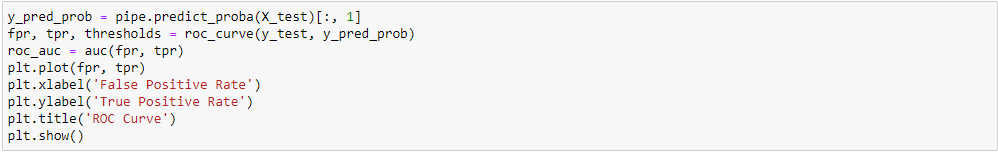
**D3: Code Execution**

Scatterplot:

Model Complexity Curve:



Accuracy score:

ROC Curve:

AUC:

**E1: Accuracy and AUC**

The accuracy of the classifier when fitted to the testing split is 97.8% when K is equal to 11. This score suggests that out of the 2,000 entries in the test set, the classifier correctly predicted the ‘Re\_admis’ label of 1,956 patients. To further evaluate the classifier’s performance, the area under the ROC curve of the classifier is 0.996. The high AUC score suggests that the model can accurately distinguish between a positive and negative class 99.6% of the time.

**E2: Results and Implications**

Altogether, the results show that given the amount of days of a patient’s initial stay and their total charge, you can quite accurately predict whether the patient will be readmitted within a month of release by using the KNN classifier.

**E3: Limitation**

A possible limitation of my analysis is that by default I chose Euclidean distance for my distance metric when performing KNN. It’s possible that other distance metrics perform better, such as Manhattan, Minkowski, Jaccard, etc.

**E4: Course of Action**

My recommended course of action for the hospital chain would be to closely monitor patients whose initial stay length and total charge are relatively highly. The analysis shows that there is a clear correlation between those two features and the odds of the patient being readmitted. Intuitively this is fairly obvious, patients who stay longer and received more expensive treatment likely suffered severe incidents where the odds of readmission are higher. Still, the analysis could prove to be useful by accurately identifying which patients will need further treatment before being released.

**F: Panopto Recording**

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=cddea162-5703-4394-b3b3-af2f00050a38>

**G & H: Sources**

Bruce, P., Bruce, A. G., & Gedeck, P. (2020, May). *Practical statistics for data scientists: 50+ essential concepts using r and python*. O'Reilly.