Performance Assessment 1 – Revision 3

D209 – Data Mining I

Jessica Hosey

MSDA, College of Information Technology

Western Governors University

July 25th, 2024

Part I: Research Question

A.1. Propose one question relevant to a real-world organizational situation.

What factors contribute and lead to patients being readmitted within a month of their last doctor's visit?

A.2. Define one goal of the data analysis.

Knowing why a patient returns to the hospital a month later might help public health officials plan and provide information about ailments affecting people in the local community. In addition, hospitals knowing what precursors may label a patient as needing more care later would allow them to be more thorough in their current hospital visit versus having the patient return with the same or similar issue. Obviously, in this dataset, there are people with children, so parents might visit the hospital more often if they have children than if they didn't. However, this dataset does not contain information about patients under 18. Ultimately, this research project aims to see if any factors (Diabetes, Stroke, High Blood Pressure, etc.) could label or predict whether a patient will be readmitted to the hospital within a month of their last visit.

Part II: Method Justification

B.1. Explain how the classification method you chose analyzes the selected data set. Include expected outcomes.

The method chosen was the K-Nearest Neighbors (KNN) classification, which allows us to see patterns of unknown data points if they are near k or "neighboring" data points. The reason for choosing KNN over Naive Bayes is that it is easy to use and understand (Vishalmendekarhere 2021). It is quick to complete and does not have specific assumptions that must be held to be valid. KNN uses the training dataset to make predictions (based on neighboring data points) for later use as new data points in the testing dataset. To make these predictions it uses a “major voting” system from neighboring data points to guide predictions of possible new data points (IBM, “Background of KNN”).

The expected outcomes are that any of these independent variables are close to others to help understand if these factors contribute to a patient's need to visit the hospital more than once a month (ReAdmis).

B.2. Summarize one assumption of the chosen classification method.

One assumption of the KNN classification method is that similar data points are close to one another (Vishalmendekarhere 2021). Therefore, a widely spaced set of points would not be a good dataset from which to use KNN classification to predict similar data points, as there are no neighboring points from which to gather information.

B.3. List the packages or libraries you have chosen for Python or R and justify how each item on the list supports the analysis.

- Pandas: CategoricalDtype and load the dataset to the anaconda jupyter environment.

- NumPy: to perform general mathematical calculations

- Matplotlib and Seaborn for visualizations

- Sklearn: train_test_split, Preprocessing (used MinMaxScaler() to standardize the numeric values), Preprocessing- OneHotEncoder (to encode the categorical data), SelectKBest, f_classif, KNeighborsClassifier, GridSearchCV, confusion_matrix, roc_auc_score, oc_curve, and classification_report for manipulation of the data to perform several Machine Learning Algorithms and to fine-tune/evaluate our model.

Part III: Data Preparation

C.1. Describe one data preprocessing goal relevant to the classification method from part A1.

Many preprocessing methods were used to prepare the dataset for analysis. One method needed to complete the KNN classification was to encode the categorical data to represent 1 for a Yes response and 0 for a No response. This allows the human response to be represented as a numerical value. This allows the KNN classifier to see the data and understand what a Yes or No response entails.

C.2. Identify the initial data set variables used to perform the analysis and classify each variable as numeric or categorical.

Variables	Classification
Area (Rural, Suburban, Urban)	Categorical
Number of Children	Numeric
Age	Numeric
Gender	Categorical
Readmission (Dependent Variable)	Categorical
Vitamin D levels	Numeric
Number of Dr Visits	Numeric
Meals Eaten Prior	Numeric
Soft Drink	Categorical
Initial Admission Reason (Observation, Elective, Emergency)	Categorical
High Blood Pressure	Categorical
Stroke	Categorical
Complication Risk (Low, Medium, High)	Categorical
Overweight	Categorical
Arthritis	Categorical
Diabetes	Categorical
Hyperlipidemia	Categorical
Anxiety	Categorical
Allergic Rhinitis	Categorical

Asthma	Categorical
Days hospitalized	Numeric

C.3. Explain each of the steps used to prepare the data for the analysis. Identify the code segment for each step.

The same steps were used to prepare the data for both D208 tasks for this dataset (Hosey 2024). Most of this preparation includes remapping the ordinal data and the binary categories to numerical values needed to complete the KNN classification method for this task.

Once the data types were changed and the ordinal and binary data remapped were completed, I explored the data. The use of `describe()` and `value_counts()` was to make sure that the data was ready to be used for the classification method that I chose (KNN). Then dummy columns will be created using sklearn's `OneHotEncoder()` this is a different method compared to my D208 submissions as `GetDummies` was not best for KNN Classification. These dummy values and the columns I am interested in will remain in the dataset, and the columns I am not interested in will be dropped. After this, the data frame will be rechecked to ensure the columns I selected remain the same and have the correct data type to move forward. After creating dummy values, I attempted to complete the next step, however the code brought back errors stating that there were Nan values present. This issue had to be resolved or the KNN Classification would not work. The use of `dropna()`, `iloc`, and `concat()` was used to drop those rows with Nan values to move forward. I did not use imputer as it would have messed with the data and created values that were not true to the dataset, so I choose to drop the rows instead.

Next was to rescale the feature set using `MinMaxScaler()`. This will put the features on the same scale to ensure that no one feature can dominate another and ruin the KNN classification method that I will be using later in this task. After the dataset is scaled, I must identify which factors are best for KNN classification. Using `SelectKBest()`, we will locate the x values with the best k value for the model (Elleh 2022).

C.4. Provide a copy of the cleaned data set.

See the attached files in my submission.

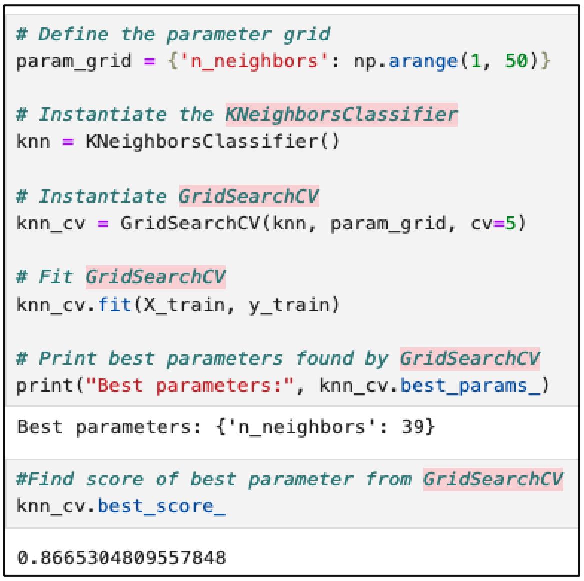
Part IV: Analysis

D.1. Split the data into training and test data sets and provide the file(s).

See the attached files in my submission.

D.2. Describe the analysis technique you used to appropriately analyze the data. Include screenshots of the intermediate calculations you performed.

Below is the screenshot of the intermediate calculations used to analyze the data before completing the KNN classification method. After the initial preparation of the data and selecting the correct factors to include in my analysis, the data was split into testing (30%) and training (70%) datasets. Now, we can use the k-nearest Neighbors classification to find the appropriate value of k. Using GridSearchCV will allow us to use a few different k values to see which helps create the most accurate model. Now that we know which k value is best, we can use KNN classification with our k-value of 39.



```
# Define the parameter grid
param_grid = {'n_neighbors': np.arange(1, 50)}

# Instantiate the KNeighborsClassifier
knn = KNeighborsClassifier()

# Instantiate GridSearchCV
knn_cv = GridSearchCV(knn, param_grid, cv=5)

# Fit GridSearchCV
knn_cv.fit(X_train, y_train)

# Print best parameters found by GridSearchCV
print("Best parameters:", knn_cv.best_params_)
```

```
Best parameters: {'n_neighbors': 39}
```

```
#Find score of best parameter from GridSearchCV
knn_cv.best_score_
```

```
0.8665304809557848
```

D.3. Provide the code used to perform the classification analysis from part D2.

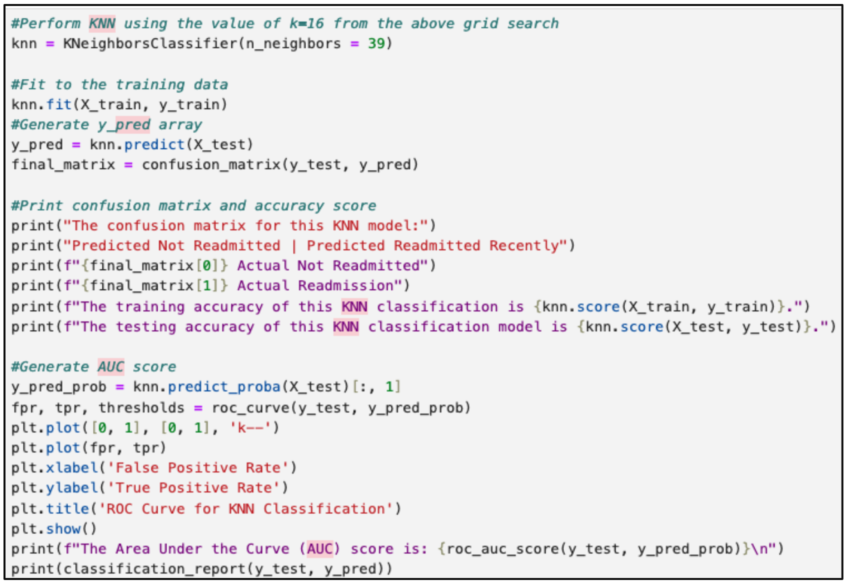
See the attached files in my submission.

Part V: Data Summary and Implications

E.1. Explain the accuracy and the Area under the curve (AUC) of your classification model.

Accuracy was not the greatest, as you could only get so much information from one day of the hospital operation. The screenshots below show that the model correctly predicts when a patient is not readmitted more than when a patient is readmitted based on Initial_days and Initial_admin_Emergency Admission features. The model, however, makes 2601 (1501 + 1100) correct predictions, which is decently high compared to 86% accuracy for the KNN classification model.

As the Receiver Operating Characteristic (ROC) Curve, the graph indicates two lines. The dashed line is the performance of the classification model at complete random. If there were a curve below the dashed line, the model would predict classifications worse than randomly picking the classifications. Since the second line on the graph is above the dashed line (random classifications), the model predicts well above random classifications. The curve for my model is not up near the top left corner, which means it is not 100% accurate. In addition to the graph, I computed the Area Under the Curve (AUC) score, which was 90.2%. With an AUC score and an ROC curve like this, the model outperforms the randomly classified dashed line (used as a control to compare to).

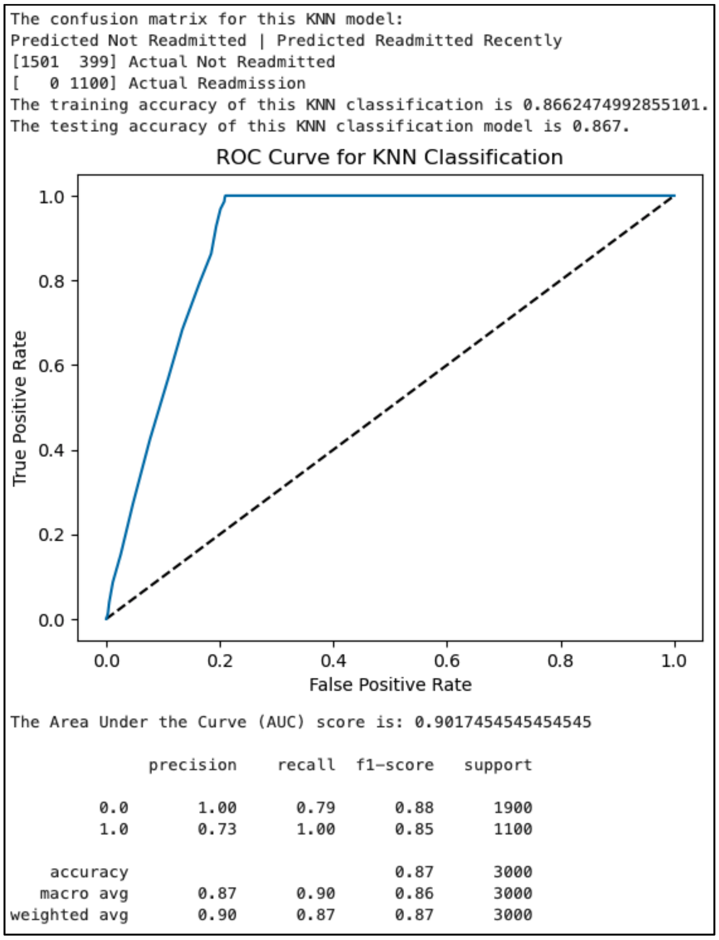


```
#Perform KNN using the value of k=16 from the above grid search
knn = KNeighborsClassifier(n_neighbors = 39)

#Fit to the training data
knn.fit(X_train, y_train)
#Generate y_pred array
y_pred = knn.predict(X_test)
final_matrix = confusion_matrix(y_test, y_pred)

#Print confusion matrix and accuracy score
print("The confusion matrix for this KNN model:")
print("Predicted Not Readmitted | Predicted Readmitted Recently")
print(f"{final_matrix[0]} Actual Not Readmitted")
print(f"{final_matrix[1]} Actual Readmission")
print(f"The training accuracy of this KNN classification is {knn.score(X_train, y_train)}.")
print(f"The testing accuracy of this KNN classification model is {knn.score(X_test, y_test)}.")

#Generate AUC score
y_pred_prob = knn.predict_proba(X_test)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
plt.plot([0, 1], [0, 1], 'k--')
plt.plot(fpr, tpr)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for KNN Classification')
plt.show()
print(f"The Area Under the Curve (AUC) score is: {roc_auc_score(y_test, y_pred_prob)}\n")
print(classification_report(y_test, y_pred))
```



```
The confusion matrix for this KNN model:
Predicted Not Readmitted | Predicted Readmitted Recently
[1501  399] Actual Not Readmitted
[   0 1100] Actual Readmission
The training accuracy of this KNN classification is 0.8662474992855101.
The testing accuracy of this KNN classification model is 0.867.
```

```
The Area Under the Curve (AUC) score is: 0.9017454545454545

              precision    recall  f1-score   support

         0.0       1.00      0.79      0.88      1900
         1.0       0.73      1.00      0.85      1100

    accuracy                           0.87      3000
   macro avg       0.87      0.90      0.86      3000
weighted avg       0.90      0.87      0.87      3000
```

E.2. Discuss the results and implications of your classification analysis.

The model has a high accuracy based on those two features; however, logically, those two factors might not be the reason for readmission. Unfortunately, back pain or diabetes was not selected for KNN classification. Again, the accuracy for those two factors is high and could be improved if the dataset is expanded to include more than a day of operation and all aged patients.

E.3. Discuss one limitation of your data analysis.

As stated above in E2, the dataset is incomplete and would be more helpful if there was more than one day of hospital operation. This would increase the sample size and the chances of finding insights beyond those selected features (Initial_days and Initial_admin_Emergency Admission). Since this is not the case, the AUC score is high but not helpful for physicians when providing patient care.

E.4. Recommend a course of action for the real-world organizational situation from part A1 based on your results and implications discussed in part E2.

To gain more insight into the given dataset, choosing a different dependent variable would be the best option to move forward. Selecting diabetes or high blood pressure might provide better insights into why patients are readmitted to the hospital. This may open the door for the model to select other features that this current model did not choose (Initial_days and Initial_admin_Emergency Admission). Again, as stated above repeatedly, increasing the sample size and variety will benefit the organization if they wish to find meaningful insights within this data.

Part VI: Demonstration

F. Panopto Video

See the attached files in my submission.

G. Code Sources

Bowne-Anderson, Hugo. DataCamp. (n.d.). Supervised Learning with Scikit-Learn: Classification [Online course]. DataCamp. https://campus.datacamp.com/courses/supervised-learning-with-scikit-learn/classification-1?ex=1

Bowne-Anderson, Hugo. DataCamp. (n.d.). Supervised Learning with Scikit-Learn: Classification [Online course]. DataCamp. https://campus.datacamp.com/courses/supervised-learning-with-scikit-learn/classification-1?ex=1

Elleh, Festus. Panopto. (2022, July 12). [Welcome to D209 Data Mining I Webinar Video]. Western Governors University. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=b73b6274-ef01-4d1b-a59f-aed100228a93

Glen, Stephanie. Data Science Central. (2019, June 19). Comparing classifiers: Decision trees, KNN, Naive Bayes. Data Science Central. https://www.datasciencecentral.com/comparing-classifiers-decision-trees-knn-naive-bayes/#:~:text=Naive%20Bayes%20is%20a%20linear,Naive%20Bayes%20over%20K%2DNN

Hosey, Jessica. (2024). Performance Assessment 2, D208 – Predictive Modeling. [Unpublished manuscript]. WGU.

Prashant. (n.d.). KNN classifier tutorial. Kaggle. https://www.kaggle.com/code/prashant111/knn-classifier-tutorial

scikit-learn developers. (n.d.). sklearn.neighbors.KNeighborsClassifier. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Stats Wire. (2021 March 13). Data Preprocessing 06: One Hot Encoding python | Scikit Learn | Machine Learning [Video]. YouTube. https://www.youtube.com/watch?v=InZ0n2knz1E

H. Content Sources

IBM. (n.d.). K-nearest neighbors (KNN). IBM. https://www.ibm.com/topics/knn

IBM. (n.d.). Background of KNN. Retrieved July 13, 2024, from https://www.ibm.com/docs/en/ias?topic=knn-background

Vishalmendekarhere. (2021, January 17). It’s all about assumptions: Pros & cons. Medium. https://medium.com/swlh/its-all-about-assumptions-pros-cons-497783cfed2d