Classification Analysis

Matthew E. Heino

Data Mining I

Classification Analysis

Introduction

The purpose of this paper is to explore a business question using a classification algorithm. The chosen algorithm will be the K-nearest Neighbors algorithm. This paper will walk through the steps to prepare the data for analysis. There will be justification offered as to why this algorithm is the best to answer the organizational question. There will be a discussion of the findings of the analysis. Each section within this paper will address them and offer insight as to how they apply to the organizational question and the chosen algorithm.

Background

The dataset is composed of 10,000 observations and 50 columns. The data stored within the provided CSV file is data about patients. The information contains the expected information like address and other personal details. The dataset is composed of both categorical and numeric data. Some of the data that is included in the CSV file will not used in this analysis. This will not offer any insight to the question.

Some of the code that will be used in this assessment will come from previous coursework. It is prudent to include it in this assessment since this code has been proven to work and accomplished some of the tasks that are required to complete certain sections of the assignment. Code that has been modified from online sources will have the appropriate citation either included in this document or the accompanying Jupyter Notebook.

The layout of the paper will follow the sections of the rubric and the assessment document. If there is a change in structure it will be noted.

Part I: The Organizational Business Question

This part of the document will be an introduction to an organizational question. This will be followed by what the goal of the analysis will hope to accomplish. This will be after the completion of creating, training, and utilizing the K-Nearest Neighbor algorithm.

A1. The Question. The question that is of interest to the organization is “Is it possible to predict whether or not a patient will be readmitted to the hospital using the k-nearest neighbors algorithm?” This is an important organizational question since medical organizations want to reduce the chances that their patients are returning to the hospital.

The answer to this question is good for both the hospital and the patient. It is best to keep repeat admissions to a bare minimum. This question's analysis can be used to help classify certain types of patients based on their previous medical information. For example, does the patient having high blood pressure mean that the patient would be a candidate for possible readmission in the future? While this is just one feature that could be looked at, it can pose a starting point for gathering other features that may be worth looking at.

The question will be addressed by using the K-Nearest Neighbor algorithm to group patients based on health features that will be discussed in subsequent sections of this paper.

A2. Goal of the Analysis. The goal of the analysis is to see what features in the patient's medical history can be used to help gauge whether or not they will be part of a group that will likely be the subject of being readmitted to the hospital. With these features or medical traits, there can be a course of action devised to help keep these "at risk" patients out of the hospital. This is both beneficial to the patient and the medical organization.

Less returns to the hospital means the patient does not incur any undue hardships and the hospital does not spend additional resources on the patients. If the patient does not return that means the patient's medical issues have been resolved successfully and their prior history was successfully acknowledged and addressed. This is the goal of the question and its subsequent analysis. Using the KNN model will allow for a prediction of whether a patient with a given set of features (medical history factors) will be in the group or cluster of patients being re-admitted.

The goal of the research question is to see if it is possible to build a KNN model that can faithfully classify patients as to whether they will be a candidate to be readmitted in the future. The caveat here is that re-admittance has to occur within the thirty days following discharge.

Part II: Justification of the Method

In this section, there is a discussion on how the K-nearest Neighbor algorithm works. The section will give a summary of the mechanics of the algorithm and why it will be utilized to answer the research question that was proposed in Section A of the document.

B1. Explanation of K-Nearest Neighbor Algorithm. The K-nearest neighbor algorithm is a modeling algorithm that is used to classify observations based on labels and features. Its main concept is data points that are close to each other must belong to the same group or cluster. The algorithm is based on the idea of proximity.

The KNN algorithm will look at neighbors that are within a certain distance of the given point. The most common distance that is used is the Euclidean distance. The Euclidean distance is the distance between two points. This is the default metric that is used in the sklearn library that will be used to create the classifier (Gokte, n.d.). The creation of the KNN classifier will be discussed in a later section of the paper.

The number of points that are looked at is determined by the ***k*** value. This ***k*** value is the number of points that will be polled – the neighbors that fall within a given distance of the new data point. Polling here means that based on the number of similar features the new data point has between itself and the polled points. Using the majority of votes the algorithm will assign the data point to that particular group.

B2. Assumption of K-Nearest Neighbor. The KNN assumption that is easiest to understand is the assumption that points that are close to each other are similar to each other. When a group of points are near each other it is safe to assume that they are members of the same group. The further a point lies from another point the likelihood that the two points belong to the same group diminishes. It can be said that the further a point is from the reference point the more dissimilar to the reference point the point is (Nelson, 2020).

B3. Python Packages Used. Python will be the language of choice in creating the model that will be used to create the KNN model. Python has many libraries that will be useful. The table that is given below will list the library and how it will be used to accomplish the task of creating and evaluating the model. Please note there is no specific order to the table.

	Packages & Libraries	Usage
1	pandas	Provides the ability to read in CSV, create a dataframe, and provide methods for manipulating the data within the frame.
2	missingno	Visualizes any of the missing values in the dataframe
3	seaborn	Provided the library for visualizations for data exploration.
4	matplotlib.pyplot	Provided additional means for plotting data.
5	from sklearn import preprocessing	Provided the function for scaling the data.
6	from sklearn.feature_selection import	Provided the SelectKBest for choosing the best

	SelectKBest, f_classif	features to include in the model.
7	from sklearn.model_selection import train_test_split	Provides the method to split the data into training and testing sets.
8	import numpy as np	Provided the arange function.
9	import math	Provided the math function square root.
10	from sklearn.metrics import confusion_matrix, roc_auc_score, roc_curve, classification_report	Provided functions for the creation of the confusion matrix, score function for ROC and AUC, and the function for the classification report

Part III. Preparing the Data for Analysis

In this section, there will be a discussion of what is the goal of data preprocessing as it pertains to the creation of the model. The section will also include a discussion about what variables will be included and which variables will be excluded from the model. There will be a discussion about the steps that will be employed to bring the data into a state that is conducive to the model creation and analysis.

C1. One Goal of Data Preprocessing. The main goal of data preprocessing is to get the dataset into a state that allows for its use in the model. The model of choice for answering the question is the KNN model. This model requires that the data that is used as inputs be in a certain state. The KNN model will need to have data that is free of missing data, outliers have been removed and the categorical values be represented in a numerical state (encoded). Looking at the values for the categorical values in the given CSV file it should be noted that the categorical values are strings or objects as they are stored when read into a dataframe in pandas.

These variables will need to be encoded into their numerical equivalents. How this will be accomplished will discussed and explained in Section C3 below.

C2. Description of the Variables Used. To answer the question that was proposed in Section A, we need to gather the right features from the given dataset. This analysis will focus on medical conditions that the patient may have. The features that do not reflect medical conditions will not be included in the list of candidate features that will be used to create the model. For example, the patient's personal information will not be included in the initial list of model candidates, e.g., Area.

The table below shows the feature candidates that will be used in the initial KNN model. Please note that the list may be reduced as the creation of the model enters different stages that are a part of the paper. If a feature is removed there will be a note in the appropriate section of the paper.

	Variable Name	Independent or Dependent	Data Type	Data Class
1	ReAdmis	Dependent	Categorical	Qualitative
2	Age	Independent	Continuous/ numeric	Quantitative
3	HighBlood	Independent	Categorical	Qualitative
4	Stroke	Independent	Categorical	Qualitative
5	Complication_risk	Independent	Categorical	Qualitative
6	Overweight	Independent	Categorical	Qualitative
7	Diabetes	Independent	Categorical	Qualitative

8	Hyperlipidemia	Independent	Categorical	Qualitative
9	Asthma	Independent	Categorical	Qualitative
10	Initial_days	Independent	Continuous/ numeric	Quantitative

Please note the quantitative variables are numeric and the categorical ones are not and these will need to be converted into a numeric representation.

C3. Steps Used to Prepare the Data. The data in its current state is not suitable for use in the KNN model. The steps below will be followed to create a dataset that can be used in the mode.

1. The model will be read into a pandas dataframe from the provided CSV file. At this time the only columns that will be read into the dataframe will be the features that were discussed in Section C2. All the other columns that are in the CSV file will be disregarded as they do not pertain to the question and are not needed to create the initial model. See Jupyter Notebook section Pre-assessment Tasks.
2. Duplicates in the dataset will looked for in the dataset. If there are instances of duplicate data. The data will be removed from the dataframe. See Jupyter Notebook Section C3 Step 1.
3. Missing data will be looked for in the data set. This will be accomplished using **.isnull().values.any()** and **isnull().sum()** and a visual will be created (see Notebook for visual). If missing data is found then it will be imputed using the appropriate method. See Jupyter Notebook Section C3 Step 2.

4. Outliers will be looked for in the data. The inclusion of the outliers will affect the model. The inclusion of an outlier can affect the decision boundaries that are employed in the KNN classification boundaries. This could lead to a misclassification of the new data point (Victor Lavrenko & Lavrenko, 2015). This in turn will affect the accuracy of the model. Outliers will be checked using the interquartile range method (IQR). (This method has been successfully used in other tasks and will be employed here.) If the observation falls outside of the accepted bounds the observation will be removed from the dataframe. See Jupyter Notebook Section C3 Step 3.
5. After outliers are handled, any missing data will be imputed. This particular dataset did not exhibit any missing data. So no imputation of values will be undertaken. (***Note****: This step was skipped in the Jupyter Notebook. No outliers were found.*)
6. There will be a look at the summary statistics this to look at values like counts and other data that may be of interest. There was nothing out-of-the-ordinary found in the candidate features. This code can be found in the section Summary Statistics for the Chosen Features.
7. The categorical variables will be transformed from the string values to a numeric. Two methods will be employed here. For categorical features, the **map** method will be used to encode the Yes/No to 1 or 0. For the categoricals that are composed of more than two levels, the pandas **get_dummies** method will be employed. For example, the Complication_risk will be the variable that the get_dummies method will be used on. See Jupyter Notebook Section Data Transformation.
8. The data will be scaled. This is to make sure that there are no features that unduly influence the distance calculation. This will occur in a separate section and will not be

done until it becomes necessary to begin to create the KNN model. The MinMaxScaler will be utilized to scale the data. The scale will be will range from 0 to 1. See Jupyter Notebook Section Scale Features.

9. The best features will be selected using SelectKBest. This class will select the best features to be used. The features that have been selected for inclusion were identified during this step. The features that are included in this model are the following: Initial_days, Asthma, and Age. This was after using the SelectKBest class to determine what features to select from the candidate features that were proposed in section C2.

The code to accomplish these steps can be found in Section C3 unless otherwise noted previously.

C4. Copy of Cleaned Dataset.

A copy of the clean data set can be found in the following file:

Heino_Cleaned_Medical_Task_1.csv

The code that was used to create the file can be found in section C4 of the Jupyter Notebook.

Part IV. The Analysis

D1. Creating the Training and Test Data Sets. The training and test datasets were created using the rule of thumb of 80/20. Where the dataset was split into 80% training and 20% testing. The method to accomplish this task can be found in the sklearn library. The method is the **train_test_split** method. The method used **stratify=y** to make sure that the same proportions of each class as they are observed in the original dataset.

After the creation of the split datasets you will find the data in the following files:

- **X_train**: Heino_X_train_Task_1.csv

- **X_tes**t: Heino_X_test_Task_1.csv
- **y_train**: Heino_y_train_Task_1.csv
- **y_test**: Heino_y_test_Task_1.csv

See Jupyter Notebook Section D1 for the code.

D2. Description of the Analysis Technique. The model that was chosen is the KNN model. There are a few parameters that will be used to create the model. The ***k*** value will be used to determine how many neighbors will have to vote on whether the new data point belongs to that particular group.

This ***k*** for the model should be greater than zero and less than or equal to the number of features in the original dataset. The ***k*** will be determined by running the model with various ***k*** values and looking at how well they perform. This concept is referred to as hyperparameter tuning. This will be implemented using **GridSearchCV** to find the optimal value of k from a list of chosen values.

The KNN will use the default metric of Euclidean distance to determine the distance. The rationale is that the KNN will be computing the distances of the points (Gokte, n.d.). The findings for the k value will be used as the value in the **n_neighbors** argument. The value that was determined during this stage was 42. The value 43 will be used. The rationale for using this odd number of ***k*** is to make sure that a point does not fall into the trap of "confusion" between two classes (Kumar, 2021).

After this tuning the model with these "best" values will be created. This new model can be found in the section "Create the KNN model with the best K value" in the cell [20]. This is where you find all the code for the creation of the model.

There were no other intermediate calculations used to create the model. All calculations were handled by the appropriate methods.

D3. Code for Used for Classification Analysis. The code that was used for the creation of the analysis can be found in the following section of the Jupyter Notebook.

Purpose of Code	What Cell in the Jupyter Notebook is it Found
Splitting the data into test and training sets	D1. Split the Cleaned Dataset. Splitting the data into the training and test sets.
Refining the Classifier	Found in the cell [18].
Creating the Classifier	Create the KNN model with the best K value.
Print Confusion Matrix	Print a confusion matrix for the model.
Accuracy of the Classifier	The accuracy of the model.
AUC score and Classification Report. [1]	Calculating the AUC score and classification report.

There is additional code that can be found in the Jupyter Notebook. (The code is extensive and it is better viewed in the Notebook. It is easier to view in that manner.)

Part V. Summary of Analysis and Its Implications

After completing the model and running the model with the test there will be an analysis of its accuracy. This analysis will be found in this section. There will be a discussion of the results of the models and the implications that this model may have. A brief discussion of the limitations of the analysis will be found in this section. The last section of the paper will go into

[1] This code is also used in Section E of this report.

a course of action that will be based on the analysis of the model and its predictions. How good a model is can be explained in numerous ways two are given below. A few others are discussed in the subsequent sections.

E1. Accuracy and AUC. Accuracy can be described as the number of correct predictions divided by the number of all the predictions (Nighania, 2021). This information can be visualized using a confusion matrix. The matrix below is for this KNN model.

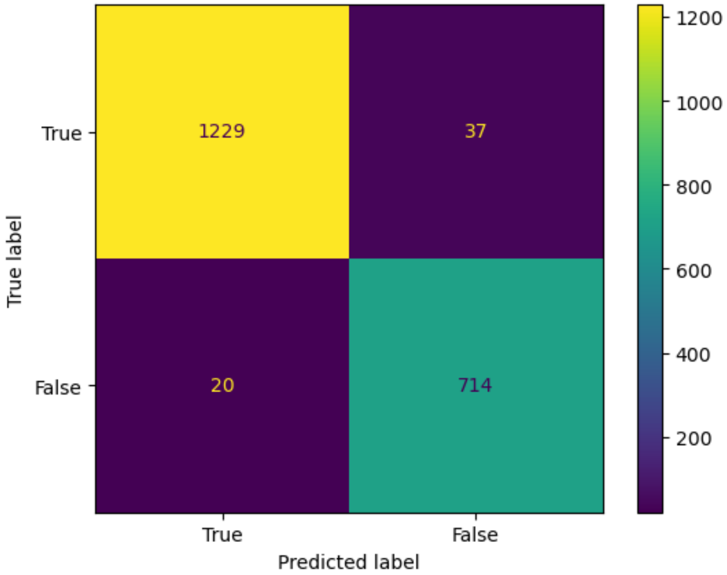

Figure 1. Confusion matrix for the KNN model

This matrix is composed of four quadrants. The top left corresponds to the true positives (TP). The top right is the false negatives (FN). The bottom left is the false positives (FP) and the bottom right is the true negatives (TN). The values within the boxes can be used to determine a few measures that judge how good a model is. The measure we are concerned with is accuracy. As stated earlier is the number of correct predictions, true positives plus true negatives. This is then divided by the total number of observations. The formula is the following (Nighania, 2021):

$$(TP + TN) / (TP + FP + TN + FN)$$

For the model that was created for this assessment the value that was arrived at was generated using the train and test sets generated the following scores:

- Training set: 0.978875 or ~97.8%
- Test set: 0.9715 or ~97.1%

Using this metric there seems to be the model performed equally with the data that it was not trained on. The percentages while not the same are reasonably close to indicate that the model did perform well. There is no evidence of overfitting based on this metric.

Another metric to gauge the validity of the model is the area under the curve (AUC). The AUC is an area that is bounded by the ROC curve. The graph below illustrates the curve for this KNN model. The x-axis is the false positive rate and the y-axis is the true positive rate.

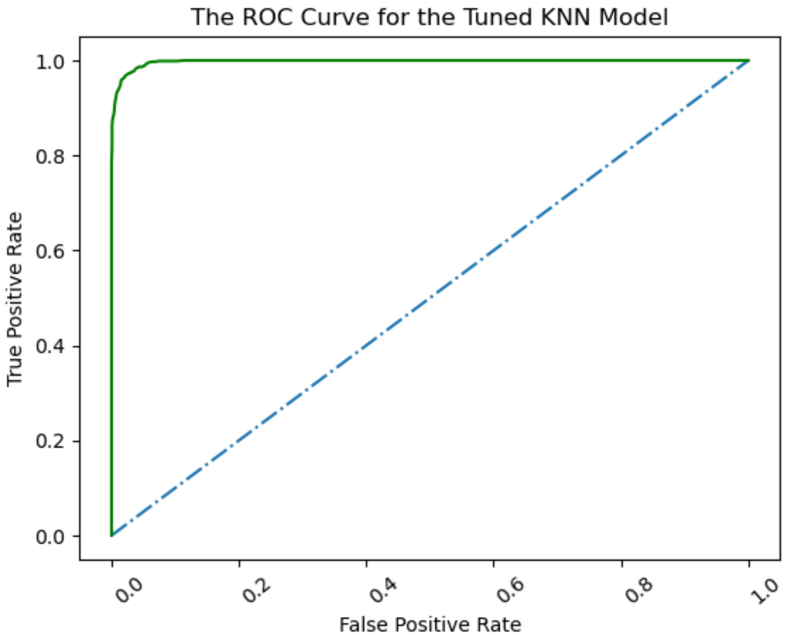

Figure 2. ROC Curve

This graph shows a model whose AUC is approaching 1 or almost perfect performance based on the model. Using the **roc_auc_score** method, the area under the curve is ~.9977, which means that as this score approaches one the model is almost perfect. If the rounded part of the graph were complete at the ***y*** point equal to 1 instead of approaching it the model would be a "perfect" predictor (*How to Explain the ROC AUC Score and ROC Curve?*, n.d.).

Based on these two metrics the model seems to be a valid one. The metrics used seem to indicate that the model will perform well on data that it has not seen.

E2. Results and Implications. The KNN was tasked with classifying patients into whether or not they would be readmitted to the hospital based on a select group of features. This model performed very well given the metrics that were discussed in an earlier section. Using the three features (Initial_days, Asthma, and Age) proved to be acceptable. Using SelectKBest with a few tweaks to the parameters yielded good results. The model's variables seemed to be scaled appropriately. It might be prudent to see what other variables could be included to increase the accuracy and to include a more diverse feature set.

Based on the accuracy of the model the model does not need any further tuning. The accuracy will be good enough to be used on data sets that the model was not trained as given by the training was 97.1%. So it did not overfit while being exposed to the training data. The model can make accurate predictions that were not part of the training set.

While tuning with another parameter setting may be utilized, it is probably not needed since the model offers good accuracy with the current features.

This model would be a valid one when using the feature. It would be a good model to make predictions about whether a patient will be a returning patient within the next 30 days.

E3. A Limitation of the Analysis. A limitation of this model is that it only handles a few of the many features of the dataset. It might be beneficial to look at other combinations to see if it is possible to obtain the same level of accuracy as the three that are utilized in this model. It might be beneficial to look at these features to see if they have any bearing on whether or not the patient will readmitted to the hospital within 30 days of being discharged. Adding the other features may improve the accuracy, but with the caveat that including more features will increase the complexity of the model. This will make the model computationally more complex and resource-intensive. This inclusion of additional features should only be undertaken if there is a reason and the appropriate resources are available.

E4. Course of Action. Based on the model's accuracy score this model can be used to predict whether a patient will be returning to the hospital. This model's models predictor features can be used as a way to improve the patient's odds of not returning to the hospital. The only feature that is outside the control of the hospital is the Age variable. Asthma can be dealt with by making sure that the patient has adequate care both within the hospital and when the patient returns home. For Initial days, it might be worth looking into what is the cause of these prolonged stays. This is something that may be worth looking into to see if there is an avenue to reduce the stay.

Part VI. Demonstration

F1. The Panopto Video. The link below will take you to the Panopto video that shows a demonstration of the working code. The link is the following:

References

G. Web Resources.

In this section, you will find references to resources that aided in the creation of the assessment. These resources were used in addition to the ones that were supplied in the course material section.

sklearn.feature_selection.SelectKBest. (n.d.). Scikit-learn. Retrieved December 2, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

sklearn.preprocessing.MinMaxScaler. (n.d.). Scikit-learn. Retrieved December 2, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

H. In-text Citations

In this section will find all the references that were used in the creation of the written document. These are outside references that were not included in the Web Sources section or provided by the university.

Gokte, S. A. (n.d.). *Most Popular Distance Metrics Used in KNN and When to Use Them - KDnuggets*. KDnuggets. Retrieved December 1, 2023, from https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html

Kumar, A. (2021, December 14). KNN Algorithm: When? why? how? - towards data science. *Medium*. Retrieved December 2, 2023, from https://towardsdatascience.com/knn-algorithm-what-when-why-how-41405c16c36f

Nelson, D. (2020, August 23). *What is a KNN (K-Nearest Neighbors)?* Unite. AI. Retrieved December 2, 2023, from https://www.unite.ai/what-is-k-nearest-neighbors/

Nighania, K. (2021, December 7). Various ways to evaluate a machine learning model's performance. *Medium*. Retrieved December 2, 2023, from https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15

Victor Lavrenko, & Lavrenko, V. (2015, September 15). *kNN.4 Sensitivity to outliers* [Video]. YouTube. Retrieved December 1, 2023, from https://www.youtube.com/watch?v=_yNLrPxG7PE