Saemi Ramirez

011926418

D209 Data Mining I

PA1 – Classification Analysis

7/27/2024

WGU

1. Describe the purpose of this data mining report
   1. My research question is “What factors contribute the most to the patient diagnosed with high blood pressure?” and I will be using KNN.
   2. The goal of the data analysis is to identify the key factors that most significantly contribute to patients being diagnosed with high blood pressure. This involves examining various potential predictors and determining their impact on the likelihood of a high blood pressure diagnosis. The insights gained from this analysis can help in understanding the underlying causes, improving patient risk assessments, and developing targeted interventions for prevention and management.
2. Reasons for the chosen classification method
   1. The K-Nearest Neighbors algorithm is a machine learning method for both classification and regression tasks. It operates on the principal data points with similar features are likely to have similar labels or values. The algorithm works by identifying the K closest neighbors to a given input data based on distance metrics. For classification tasks, it predicts the label of the input data point by choosing the most frequent class label among its K neighbors (Srivastava). The expected outcome is whether the patient has high blood pressure (Yes/True) or not(No/False).
   2. The core assumption of KNN is “the closer two given points are to each other, the more related and similar they are.” (Hachcham)

|  |  |
| --- | --- |
| **Packages/Libraries** | **Purpose** |
| pandas | To handle the dataset/dataframe |
| Numpy | To perform mathematical operations or values |
| Matplotlib.pyplot | To visualize the data via graphics |
| From sklearn.feature\_selection import SelectKBest | Evaluate how each feature relates to the target variable |
| From sklearn.feature\_selection import f\_classif | Captures any kind of relationship |
| From sklearn.model\_selection import train\_test\_set | To split the dataset into X, y, train and test |
| From sklearn.pipeline import Pipeline | To sequentially apply a list of transformers to process the data |
| From sklearn.preprocessing import MinMaxScaler | To normalize the numeric values by reducing the size of the data to match the original form |
| From sklearn.neighbors import KNeighborsClassifier | To implement learning based on the number of k-nearest neighbors |
| From sklearn.model\_selection import GridSearchCV | Performs exhaustive search over the specified parameter values for an estimator |
| From sklearn.metrics import roc\_curve | To plot the KNN model |
| From sklearn.metrics import roc\_auc\_score | To calculate the ROC AUC score |
| From sklearn.metrics import confusion\_matrix | Calculates the confusion matrix |
| From sklearn.metrics import ConfusionMatrixDisplay | To visualize the computed confusion matrix |
| From sklearn.metrics import classification\_report | Computes the accuracy summary of the model |

1. Data preparation for the chosen dataset
   1. One of the methods used during the data preprocessing for KNN classification is one hot encoding. The get\_dummies function was conducted.

|  |  |
| --- | --- |
| **Variable** | **Data Type** |
| Marital | Categorical |
| Gender | Categorical |
| Initial\_admin | Categorical |
| HighBlood | Categorical |
| Complication\_risk | Categorical |
| Overweight | Categorical |
| Arthritis | Categorical |
| Diabetes | Categorical |
| BackPain | Categorical |
| Anxiety | Categorical |
| Asthma | Categorical |
| Services | Categorical |
| Age | Numeric |
| Income | Numeric |
| VitD\_levels | Numeric |
| Initial\_days | Numeric |
| TotalCharge | Numeric |

* 1. Explain each of the steps to prepare the data for analysis
     1. df.info() – check the details on each column of the data set
     2. df.head() – check the first 5 rows of the dataset
     3. df.isnull().sum() – check the total number of missing values on each column
     4. df.duplicated() – check for any duplicated information in the data set
     5. df[‘Age’].describe() – check statistical information of the numeric values
     6. df[‘Marital’].value\_counts() – check the total number of each unique values
     7. initial\_model.head() – create another data frame with selected predictors and checking the first 5 rows of the information
     8. one\_hot = pd.get\_dummies(initial\_model) – conducted one hot encoding method using get\_dummies
     9. one\_hot.info() – check the details on each column of one\_hot
     10. one\_hot.head() – checking the first 5 rows of the information in one\_hot
     11. one\_hot.rename() – update the invalid column names with space to underscore(\_)
     12. one\_hot.info() – check the column names
     13. for loop – update the data type of the columns from bool to int64
     14. one\_hot.info() – check the details on each column after data type changed
     15. one\_hot.head() – check the first 5 rows of the updated data set
     16. pd.DataFrame(MinMaxScaler().fit\_transform(one\_hot), columns=one\_hot.columns) – to normalize the numeric values to set min = 0 and max = 1
     17. SelectKBest & fit\_transform(X, y) – to select the best fit predictors for the KNN model  
         A screenshot of a computer

         Description automatically generated
     18. DataFrame({‘Feature’:X.columns, ‘p\_value’:skbest.pvalues\_}).sort\_values(‘p\_value’) – sort the data by p\_value and filter the predictors that the p-value less than 0.05  
         A screenshot of a computer code

         Description automatically generated
     19. final\_data.to\_csv() – save the data frame to csv format in local drive
  2. Submitted “final.csv”

1. Data analysis and report
   1. One\_hot\_scaled dataset was split to X\_train, X\_test, y\_train, y\_test with 70% of training set and 30% of test set. Each set are submitted as ‘XTest.csv’, ‘XTrain.csv’, ‘yTest.csv’, and ‘yTrain.csv’.
   2. After splitting the data to X and y train and test sets, GridSearchCV was conducted with KNeighborsClassifier and the parameters with the ranges of neighbors between 1 and 49 to determine the best k.   
      A screenshot of a computer code

      Description automatically generated

The best number of parameters is determined to 34 with the best score of 0.579.  
A screenshot of a computer code

Description automatically generated

KNeighborsClassifier was conducted again with the n\_neighbors of 34, fitted the model, and predicted the y values. The confusion\_matrix was called to visualize the accuracy of the TP, FP, TN, and FN.  
A screenshot of a graph

Description automatically generated

Classificatin\_report was conducted to see the accuracy of the model which came out as 58%.   
A screenshot of a computer

Description automatically generated

The complexity curve was graphed to visualize the performance of both training and the test sets.   
A graph of a number of neighbors

Description automatically generated

The accuracy for the train set was approximately 60% and the test set was 58%.  
A computer code with numbers and symbols

Description automatically generated

Lastly, the AUC-ROC curve was graphed and calculated.

A graph of a logistic regression roc curve

Description automatically generated

The AUC-ROC score is approximately 51.4% which means the model classifies the patient with or without high blood pressure correctly only half of the time.

A screenshot of a computer code

Description automatically generated

* 1. Submitted ‘Saemi Ramirez D209 PA1 Classification Analysis – Submit.ipynb’

1. Summarize the data analysis
   1. The accuracy for the train set was approximately 60% and the test set was 58% which are not high numbers. The model is not classifying the patient diagnosed with high blood pressure or not correctly.   
      In the graph of the ROC curve above, the blue line above the dotted line which represents the chance model reflects a good prediction rate and the blue line below the dotted line reflects a poor prediction rate. When the threshold is set to 0, the model predicts 1 for all observations, correctly identifying all positive values but incorrectly classifying all negative values. Conversely, if the threshold is set to 1, the model predicts 0 for all data, resulting in both true and false positive rates being 0. AUC score for this case is approximately 51% which means the model is only predicting correctly half of the time.
   2. The SelectKBest approach with p-value less than 0.05 determined 6 predictors (Complication\_risk\_Low, Overweight\_No, Marital\_Married, and TotalCharge) and GridSearchCV determined the best k-NN number which was 34.   
      The confusion matrix indicates that the model predicts the true negative well (1595/3000), but not the true positive (138/3000). The test set performed lower (58%) than the training set (60%). AUC-ROC score with 51% also means the model is better than random guessing but not accurate.  
      The decision tree was not a best model for this dataset to determine whether the patient was diagnosed with the high blood pressure or not.
   3. The data set did not have enough information to predict the patient with the high blood pressure or not. The given predictors were more towards to other complications rather than the lifestyle such as eating habits, exercises, other medications, etc.
   4. The dataset needs to contain more diverse information in different points of view such as the patients’ life style including food and workouts in order to raise the prediction rate.
2. Panopto Link: <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=2d99acc5-eadd-4556-b4fb-b1b8015709d9>
3. Resources for third-party code  
     
   Keith, Mark. *Python: MLR, OLS, Standardization, normalization*. YouTube. (October 11, 2021). https://www.youtube.com/watch?v=QH\_elD\_JKuc&t=205s.
4. Resources for in-text citation

Hachcham, Aymane. *The KNN Algorithm – Explanation, Opportunities, Limitations*. Neptune.ai. (August 11, 2023). https://neptune.ai/blog/knn-algorithm-explanation-opportunities-limitations.  
  
Srivastava, Tavish. *Guide to K-Nearest Neighbors Algorithm in Machine Learning*. Analytics Vidhya. (May 22, 2024). https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/.