**Performance Assessment**

NVM2 TASK 1: CLASSIFICATION ANALYSIS

Bader Ale

Department of Information Technology, Western Governors University

D209 Data Mining I

August 2023

Table of Contents

[Part I: Research Question 3](#_Toc143866679)

[Part II: Method Justification 3](#_Toc143866680)

[Part III: Data Preparation 4](#_Toc143866681)

# Part I: Research Question

For this performance assessment, our research question is: **Given certain patient characteristics, can we classify whether a patient is hypertensive or not**.

Using the *k*-nearest neighbor (KNN) algorithm, the goal for this data analysis is to be able to classify whether a patient is hypertensive or not taking into consideration other patient-specific variables contained in the dataset (IBM, n.d.).

# Part II: Method Justification

The KNN algorithm was chosen for several reasons. First, the KNN algorithm is simple to implement – given the dataset’s overall shape, it is easily and readily implemented and used. Moreover, the KNN algorithm only requires the k-value (neighbor proximity) in order to classify the label. In this case, a value of 3 was used for *k.* The algorithm checks the closest data points (neighbors) to help it in determining the target variables (HighBlood). Comparing new data points to its neighbors, the model will classify the target variable as either HighBlood yes or no. An assumption of the KNN algorithm is that the target variable has discrete values – in this case, the target can either be Yes or No, meaning the patient is classified as hypertensive (high blood pressure) or not respectively (Vanderplas, n.d.).

The libraries used in the analysis were the following: Pandas was used as the main data manipulation tool, Seaborn and Matplotlib were used to visualize the data, Numpy was used as numerical computation tool, and SKLearn was used to pre-process the data for the algorithm (also from SKLearn). SciPy was briefly used to create the z-scores required for eliminating outliers.

# Part III: Data Preparation

In order to use the KNN algorithm, the dataset had to be preprocessed. One important step was to re-express the categorical explanatory variables. For this step, the Pandas functionality *.get\_dummies()* was used for the nominal variables such as Area, Marital, Gender and others. The *OrdinalEncoder* functionality from SKLearn was used for the only ordinal variable used in the model, *Complication\_risk.*

The following variables were used in the model

Works Cited

IBM. (n.d.). *K-Nearest Neighbors Algorithm*. Retrieved August 2023, from What is the k-nearest neighbors algorithm?: https://www.ibm.com/topics/knn

Vanderplas, J. (n.d.). *Classification Models*. Retrieved August 2023, from 1.6. Nearest Neighbors: https://scikit-learn.org/stable/modules/neighbors.html#classification